Acta Cryst D

STRUCTURAL
BIOLOGY

ISSN 2059-7983

Scaling and merging macromolecular diffuse scattering with *mdx*2

Steve P. Meisburger[a]* and Nozomi Ando[b]*

[a]Cornell High Energy Synchrotron Source, Cornell University, Ithaca, NY 14850, USA, and [b]Department of Chemistry and Chemical Biology, Cornell University, Ithaca, NY 14850, USA. *Correspondence e-mail: spm82@cornell.edu, nozomi.ando@cornell.edu

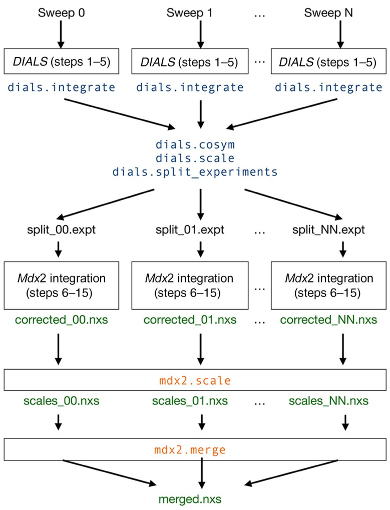
Received 22 January 2024
Accepted 25 March 2024

Edited by A. Warren, Diamond Light Source, United Kingdom

Keywords: *mdx*2; diffuse scattering; macromolecular crystallography; data processing; software; reciprocal-space mapping.

Supporting information: this article has supporting information at journals.iucr.org/d

Diffuse scattering is a promising method to gain additional insight into protein dynamics from macromolecular crystallography experiments. Bragg intensities yield the average electron density, while the diffuse scattering can be processed to obtain a three-dimensional reciprocal-space map that is further analyzed to determine correlated motion. To make diffuse scattering techniques more accessible, software for data processing called *mdx*2 has been created that is both convenient to use and simple to extend and modify. *mdx*2 is written in Python, and it interfaces with *DIALS* to implement self-contained data-reduction workflows. Data are stored in NeXus format for software interchange and convenient visualization. *mdx*2 can be run on the command line or imported as a package, for instance to encapsulate a complete workflow in a Jupyter notebook for reproducible computing and education. Here, *mdx*2 version 1.0 is described, a new release incorporating state-of-the-art techniques for data reduction. The implementation of a complete multi-crystal scaling and merging workflow is described, and the methods are tested using a high-redundancy data set from cubic insulin. It is shown that redundancy can be leveraged during scaling to correct systematic errors and obtain accurate and reproducible measurements of weak diffuse signals.



1. Introduction

Diffuse scattering is the continuous pattern in the background of X-ray diffraction images from crystals (Welberry & Weber, 2016). It occurs whenever disorder is present, and significant motion is possible in macromolecular crystals; a typical protein crystal has 30–70% solvent, which is comparable to the crowded environment of cells (Zimmerman & Trach, 1991). Recently, the recognition that experimental data on protein dynamics are needed to understand their function has led to renewed interest in diffuse scattering (Wall *et al.*, 2018). In particular, diffuse scattering encodes unique information on the correlated displacements of pairs of atoms in the crystal (Meisburger & Ando, 2017). Such correlated motions result, for instance, from the thermally excited breathing motions of proteins, which have long been implicated in mechanisms of allostery and catalysis, but are challenging to study experimentally (Xu *et al.*, 2021).

A quantitative analysis of diffuse scattering requires carefully measured and processed data. In the reciprocal-space mapping technique, diffraction data from one or more crystals in multiple orientations are combined to reconstruct the continuous, three-dimensional scattering pattern (Meisburger & Ando, 2023). The state of the art in reciprocal-space mapping has evolved rapidly in recent years, driven by several technological advances. The availability of direct X-ray detectors at synchrotrons (Förster *et al.*, 2019) has enabled new data-collection strategies to maximize data quality

https://doi.org/10.1107/S2059798324002705

through fine φ-slicing (Mueller *et al.*, 2012) and averaging many redundant observations (Winter *et al.*, 2019). The ideal properties of these detectors allow simultaneous measurement of Bragg and diffuse scattering in the same data set (Van Benschoten *et al.*, 2016; Meisburger & Ando, 2017). Software is then used to reconstruct the three-dimensional diffuse scattering patterns in reciprocal space. Building on the diffuse scattering methods used in materials science [exemplified by programs such as *XCAVATE* (Estermann & Steurer, 1998; Scheidegger *et al.*, 2000), *Mantid* (Arnold *et al.*, 2014) and *Meerkat* (Simonov *et al.*, 2020)], software tools including *Lunus* (Wall, 2009), *EVAL* (Schreurs *et al.*, 2010), *Diffuse* (Peck *et al.*, 2018) and *mdx-lib* (Meisburger *et al.*, 2020) have been developed to address unique challenges in macromolecular diffuse scattering. These programs produce patterns that can be compared quantitatively with various models (Wych & Wall, 2023; Case, 2023; Peck *et al.*, 2023).

By using these newly available tools, the field has reached a greater understanding of how different kinds of disorder contribute to diffuse patterns (Polikanov & Moore, 2015; Peck *et al.*, 2018; Wall, 2018; De Klijn *et al.*, 2019; Meisburger *et al.*, 2020, 2023). A key realization has been that atomic motions in protein crystals have correlations spanning a large range of distances, for instance from local atomic vibrations correlated over a few bond lengths to wave-like excitations of the crystal lattice that may be correlated over many unit cells. As a consequence of the large range of length scales, diffuse patterns must be measured on a fine scale (*i.e.* oversampled with respect to the reciprocal lattice) in order to account for all of the observed atomic motion (Meisburger *et al.*, 2020). Moreover, in reciprocal space the signals from the short-range and long-range correlations are superimposed, and thus precise measurements are required in order to disentangle the signals from protein motions of interest (Meisburger *et al.*, 2023). Techniques to improve accuracy and precision have been developed, including experimental background measurement and subtraction (Pei *et al.*, 2023), procedures to correct for systematic errors during scaling (Peck *et al.*, 2018; Meisburger *et al.*, 2020) and the development of new statistical indicators of data quality (Su *et al.*, 2021) and model–data agreement (Meisburger *et al.*, 2023).

As the macromolecular diffuse scattering field grows beyond the community of methods developers, it is important to build data-processing software that is easy to use, embodies best practices, promotes reproducibility in research and invites new communities of users and developers through documentation and tutorials. To address this community need, we developed *mdx*2, a Python package for processing macromolecular diffuse scattering data. A development version (0.3) has been available since 2022 with basic functionality suitable for education and preliminary data processing (Meisburger & Ando, 2023). *mdx*2 is the successor to *mdx-lib*, a MATLAB library for diffuse scattering that has been used by the authors since 2016, and it reimplements many of its successful algorithms. Unlike *mdx-lib*, *mdx*2 is designed to interface closely with the Bragg data-processing program *DIALS* (Winter *et al.*, 2022). It features a user-friendly command-line interface, also inspired by *DIALS*, and stores intermediate data and metadata in standardized, self-describing NeXus-formatted HDF5 files (Könnecke *et al.*, 2015). The NeXus format was chosen to facilitate interchange with other software, such as the *NeXpy* graphical user interface (The NeXpy Development Team, 2023), or Jupyter notebooks via the *nexusformat* package. Although successful, the development version of *mdx*2 lacked key features needed to handle the large data sets produced in modern, high-redundancy, fine φ-sliced data collection, including support for multi-crystal scaling and parallel processing.

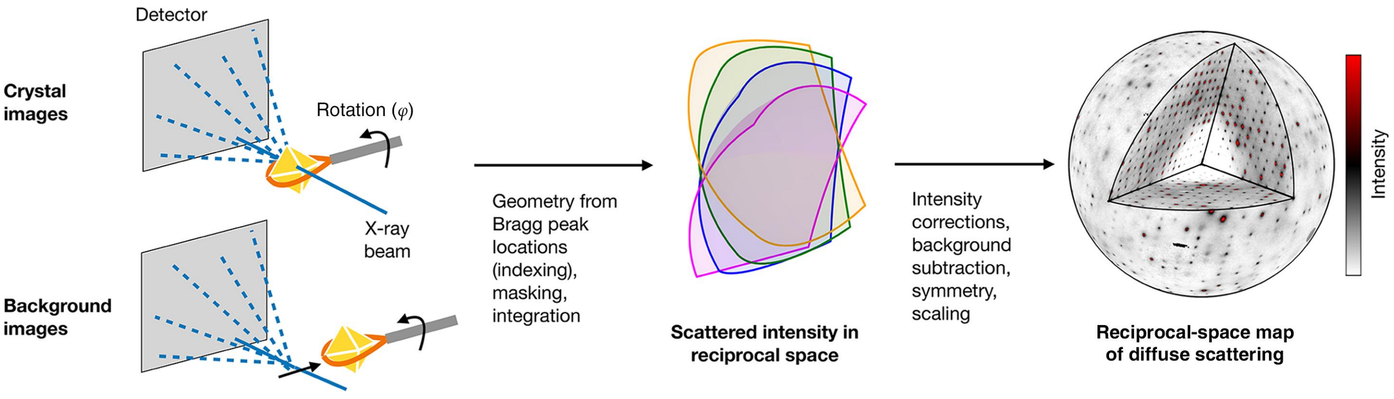
Here, we describe *mdx*2 version 1.0, the first release intended for research, which includes an implementation of the full scaling model from *mdx-lib* (Meisburger *et al.*, 2020) and features parallelized data-reduction tasks for multi-CPU architectures. This article is organized as follows. First, we provide an overview of data-processing workflows combining *DIALS* and *mdx*2 and describe the tasks performed by each command-line program. Next, we introduce the scaling model and refinement algorithm from *mdx-lib* and describe its reimplementation in Python. Finally, we demonstrate new capabilities in *mdx*2 by processing a large, multi-crystal data set from cubic insulin collected at room temperature. We take advantage of the ∼70-fold redundancy of this data set, which is unprecedented for macromolecular diffuse scattering, to examine alternative statistical measures of data quality, and demonstrate the reproducible detection of very faint diffuse signals at high resolution.

1.1. Overview of data processing in *mdx*2

*mdx*2 is a software package for processing diffraction data to reconstruct an accurate, three-dimensional map of reciprocal space. *mdx*2 breaks the reconstruction process into multiple steps that can be chained together to process data from one or more crystals. In developing the command-line interface for *mdx*2, we have focused on data collected using the traditional rotation method where the background scattering is measured at each rotation angle (Fig. 1*a*). This method, particularly when applied to large crystals, can robustly separate the scattering of the protein crystal from that of background sources, such as from air and mounting materials, and it has yielded high-quality maps in previous diffuse scattering studies (Meisburger *et al.*, 2020, 2023).

1.1.1. Diffraction geometry refinement and data import. The guiding philosophy of *mdx*2 is that individual processing steps should consist of numerical algorithms that do not depend on experimental or crystallographic details. In practice, this means that the 'import' steps in *mdx*2 use crystallographic libraries [*dxtbx* (Parkhurst *et al.*, 2014) and *cctbx* (Grosse-Kunstleve *et al.*, 2002)] in order to pre-compute all necessary information for subsequent processing steps. For example, the symmetry operators for the space group are retrieved and converted to matrix form (Fig. 1*b*, step 6), and raw diffraction-image data are re-compressed in a standard format (Fig. 1*b*, step 7). This approach has several advantages. Firstly, standardization of data formats makes it easier to optimize the performance of algorithms, particularly when parallel processing is considered. Secondly, custom algorithms can be added easily, by developers or users, because they do not require specialized libraries or expertise in crystallographic computing. Finally, the pre-computed correction factors and detector data can be inspected directly before any processing occurs, which has significant value for education and exploratory data analysis (Meisburger & Ando, 2023).

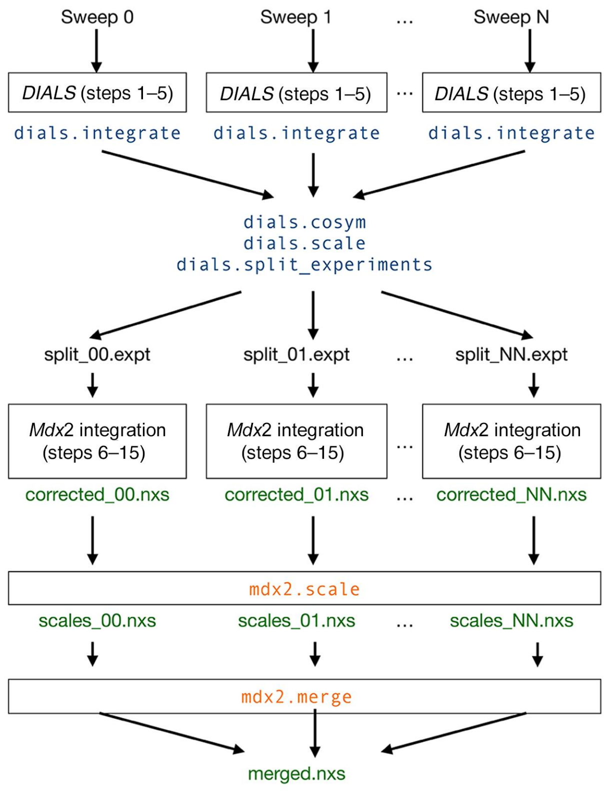
In a complete single-crystal workflow (Meisburger & Ando, 2023), *DIALS* is first used to index the diffraction patterns and refine the detector geometry (Fig. 1*b*, steps 1–5). This step is critical in order to accurately assign each pixel to a location in three-dimensional reciprocal space (fractional Miller indices *h*, *k* and *l*). A beamstop mask is also created (step 2 in Fig. 1*b*), which will be carried over into diffuse processing. Although further processing in *DIALS* is not a necessary condition


Detector
Crystal images
Rotation (φ)
X-ray beam
Background images
Geometry from Bragg peak locations (indexing), masking, integration
Scattered intensity in reciprocal space
Intensity corrections, background subtraction, symmetry, scaling
Reciprocal-space map of diffuse scattering
Intensity

(*a*)

Step	Program	Input files	Output files
1	dials.import	(diffraction images)	imported.expt
2	dials.generate_mask	imported.expt	pixels.mask, imported.expt
3	dials.find_spots	imported.expt	strong.refl
4	dials.index	imported.expt, strong.refl	indexed.expt
5	dials.refine	indexed.expt	refined.expt
6	mdx2.import_geometry	refined.expt	geometry.nxs
7	mdx2.import_data	refined.expt	data.nxs
8	mdx2.find_peaks	geometry.nxs, data.nxs	peaks.nxs
9	mdx2.mask_peaks	geometry.nxs, data.nxs, peaks.nxs	mask.nxs
10	mdx2.integrate	geometry.nxs, data.nxs, mask.nxs	integrated.nxs
11	dials.import	(background images)	bkg_imported.expt
12	dials.apply_mask	pixels.mask, bkg_imported.expt	bkg_masked.expt
13	mdx2.import_data	bkg_masked.expt	bkg_data.nxs
14	mdx2.bin_image_series	bkg_data.nxs	bkg_data_binned.nxs
15	mdx2.correct	geometry.nxs, integrated.nxs, bkg_data_binned.nxs	corrected.nxs
16	mdx2.scale	corrected.nxs	scales.nxs
17	mdx2.merge	corrected.nxs, scales.nxs	merged.nxs
18	mdx2.map	merged.nxs	map.nxs

(*b*)



(*c*)

Figure 1
Data-processing workflows. (*a*) Diffraction images are collected as the crystal rotates, and background images are taken with the crystal moved out of the beam to record scattering from materials in the beam path (*e.g.* air and capillaries). Measured intensities are mapped into reciprocal space and corrected for artifacts to produce a map of diffuse scattering. (*b*) Sequence of *DIALS* (blue) and *mdx2* (orange) command-line programs to process a single rotation data set with experimental background corrections. Diffraction geometry is refined by indexing Bragg peaks (steps 1–5) and the diffuse scattering is then integrated on a three-dimensional grid (steps 6–10). Background corrections are created (steps 11–14) and the integrated data are corrected, scaled and merged (steps 14–18). NeXus-formatted HDF5 files (`.nxs` extension, green text) are used by *mdx2* for data and metadata exchange. See Section 2 and Meisburger & Ando (2023) for details of each program. (*c*) Modified workflow for multi-sweep data. Each sweep is integrated independently in *DIALS* and then combined to resolve indexing ambiguities with *dials.cosym* (Gildea & Winter, 2018). Sweeps are processed independently in *mdx2* through the correct step; the scaling model is then refined globally and sweeps are merged.

for proceeding to *mdx*2, integration and scaling should be performed to assess the overall Bragg data quality and to detect potential issues, such as radiation damage. Ultimately, the structure solved from the Bragg data will be combined with diffuse data to build a self-consistent disorder model.

Once Bragg data processing is completed, the geometry model from *DIALS* is imported into *mdx*2 (step 6 in Fig. 1*b*). All geometric corrections are pre-computed on a grid of points, including solid angle, air absorption, polarization and detector efficiency. In addition, the fractional Miller indices are computed on a grid sampling detector position and rotation angle. Finally, symmetry information such as the Laue group operators, reciprocal-space asymmetric unit and reflection conditions are stored (Meisburger & Ando, 2023). The pre-computed data and metadata are saved in NeXus format (`geometry.nxs` in Fig. 1*b*).

The data-import step copies the detector image data from the raw format to NeXus format (step 7 in Fig. 1*b*). Because *mdx*2 and *DIALS* both use *dxtbx* (Parkhurst *et al.*, 2014) to represent experimental geometry and read detector data, any image format compatible with *DIALS* can be read by *mdx*2. During import, the masks used in *DIALS* are applied to the image data (for example to remove bad pixels and the beamstop shadow). The data are stored as a three-dimensional array, or image stack, in NeXus format (`data.nxs` in Fig. 1*b*). Data are compressed in three-dimensional chunks so that small segments of the array can be read from disk without decompressing the entire file, which is useful for certain algorithms and for parallel processing.

1.1.2. Integration. In order to integrate the diffuse pattern, it is important to exclude pixels that are potentially contaminated by Bragg peaks. The strategy used in *mdx*2 is to apply a mask wherever a Bragg peak is predicted to exist, regardless of its intensity. The mask has an ellipsoidal shape in reciprocal space (it is a function of the fractional coordinates Δhkl relative to the nearest Bragg peak). Because the extent of a Bragg peak depends on many factors, including crystal mosaicity, beam divergence and errors in the geometry model, the ellipsoidal shape is fitted empirically in order to encompass the most intense features (step 8 in Fig. 1*b*). Firstly, the images are searched for all pixels with counts above a set threshold. For the data set processed here, the threshold was set to ten times the background scattering level of ~2 photons per pixel. The threshold is tunable, and it can be increased to allow greater dynamic range in the final diffuse map (for instance to examine the halo intensity close to intense Bragg peaks). The detector location of each strong pixel is mapped to fractional Miller indices relative to the nearest whole integer. An ellipsoidal Gaussian probability distribution is then fitted to the resulting point cloud. A binary mask is generated for each image excluding the Bragg peak regions (step 9 in Fig. 1*b*). The spatial extent of the Bragg peak region is set by a cutoff value expressed as a multiple of the standard deviation of the Gaussian peak. The sigma-cutoff value is another tunable parameter that affects how much near-Bragg scattering is allowed in the final diffuse map. We find that the default cutoff of 3σ is a good compromise between the need to measure near-Bragg scattering (halos) while robustly masking all Bragg diffraction. The same cutoff was applied uniformly to all data sets. Finally, any strong pixels (those exceeding a count threshold) that are not covered by the ellipsoidal masks are also flagged as outliers and masked out. Such outliers could arise, for instance, from broken detector pixels or diffraction from small salt crystals.

The integration task accumulates photon counts in a three-dimensional grid of voxels (Fig. 1*b*, step 10). The axes of the grid are aligned with the reciprocal lattice and each voxel is assigned fractional Miller indices. When choosing a grid, the voxel dimensions must evenly subdivide the reciprocal unit cell. In *mdx*2, any integer subdivision is allowed (its predecessor *mdx-lib* allowed only odd integer subdivisions). During integration, fractional Miller indices are calculated for each pixel according to the geometric model and pixels are assigned to the nearest voxel. For each voxel, *mdx*2 keeps track of the number of photon counts accumulated, the number of pixels contributing and the mean position of the voxel in scan coordinates (the rotation angle and the location on the detector). The choice of grid spacing depends on the nature of the diffuse signal under investigation, as well as experimental considerations that potentially smear the diffuse pattern (see Meisburger & Ando, 2023).

1.1.3. Intensity corrections. After integration, the diffuse intensities are corrected for geometric effects and background scattering. Background corrections are prepared from a data set taken with the crystal moved out of the beam (Pei *et al.*, 2023). These measurements include air scattering, scattering from mounting materials (such as capillaries) and shadows cast by the sample pin. The background images are first imported using *DIALS* and the beamstop mask is applied (Fig. 1*b*, steps 11–12). The image stack is then coarsened (binned down, potentially in terms of both pixels and frames) to reduce noise (step 14 in Fig. 1*b*). Finally, the integrated photon counts from the crystal images are corrected using the background map and pre-computed geometric corrections. For each observation of a particular voxel, the correction factors are interpolated at the center position of the observation. The measured intensity and its uncertainty are estimated as follows:

$$I_{\text{measured}} = \frac{n/\Delta t - r_{\text{background}}}{\Delta\Omega E A P}, \tag{1}$$

$$\sigma_{\text{measured}} = \frac{n^{1/2}}{\Delta t \Delta\Omega E A P}, \tag{2}$$

where n is the number of accumulated photons, Δt is the cumulative exposure time per voxel (the number of pixels times the exposure time per image), $r_{\text{background}}$ is the background scattering rate (photons per second), $\Delta\Omega$ is the solid angle per pixel, E is the detector quantum efficiency, A is the transmission factor of air in the diffracted beam path and P is the polarization factor (Meisburger & Ando, 2023). The Poisson error due to background subtraction is reduced because of binning and is neglected in equation (2).

1.1.4. Scaling, merging and visualization. Generally, a diffraction data set contains redundant observations, either because the same region of reciprocal space is measured multiple times or because of symmetries in the diffraction pattern. However, the equivalent observations may disagree as a result of systematic errors in the measurement, such as changes in the illuminated volume of the crystal as it is rotated. In *mdx*2, the parameters of a scaling model are refined in order to minimize the discrepancy between equivalent observations (Fig. 1*b*, step 16), thereby correcting the systematic errors. The model describes a correction applied to each observation that depends on its scan coordinates (such as rotation angle). The scaling-model refinement procedures implemented in *mdx*2 are detailed in Section 2.2. Multi-crystal workflows can also be orchestrated using *mdx*2 (Fig. 1*c*). Scaling models for each rotation series are refined jointly.

Finally, the merging step outputs a table of fractional Miller indices, intensities and estimated errors. In order to visualize the data in three dimensions, *mdx*2 includes a mapping routine that symmetry-expands the data and places them in a three-dimensional array (Fig. 1*b*, step 18) suitable for plotting with *NeXpy*.

1.2. Scaling diffuse scattering data

The accuracy of diffuse scattering measurements may be limited by systematic errors. Artifacts commonly encountered during macromolecular crystallography experiments include detector inhomogeneities, scattering from air in the beam path, scattering from the liquid or solid mounting materials that intercept the beam, and shadows cast by the sample and mounting materials. When an irregularly shaped crystal is partially illuminated by a small beam, which is often the case, the diffracting volume may also change during the scan, leading to a change in overall intensity. Additionally, some of the diffracted X-rays may be absorbed by the sample itself, leading to variations in intensity across the detector.

In Bragg data processing, *scaling* is the determination of scale factors for each independent observation that bring equivalent observations into agreement. The process of scaling involves fitting a model for the scale factors. Typically, the model parameters are physically motivated and are parameterized to avoid overfitting. Common effects included in scaling models include rotation-dependent changes in the illuminated volume, the absorption of diffracted X-rays and *B*-factor decay due to radiation damage (Evans, 2006).

Scaling diffuse scattering data has unique challenges. The diffuse signal of interest is typically a small variation (<10%) on top of a largely homogeneous background (Wall *et al.*, 2014). Because of this, errors in scaling can very easily corrupt the small variations. In Bragg data, the local background is subtracted from each Bragg peak prior to integration and thus changes in background scattering do not need to be corrected. In contrast, for diffuse scattering the excess background scattering must be considered. Finally, since Bragg data and diffuse scattering come from the same crystal volume, the same scaling model should apply to both signals. In practice, it has not been possible to transfer the Bragg scaling model to diffuse scattering data, possibly because radiation damage-induced decay of Bragg intensitites is significant at ambient temperature. In the workflow presented here, Bragg and diffuse data are scaled independently.

In general, redundant observations are required in order to fit a scaling model. High redundancy (much greater than twofold) has significant advantages for diffuse scattering. Outliers can more easily be identified and eliminated, scaling-model refinement is more robust and systematic errors that are not included in the scaling model are more likely to be averaged out.

1.2.1. Theoretical background. The scaling process minimizes the discrepancy between the intensity predicted by the scaling model and each corresponding observation. The discrepancy can be written as a χ^2 statistic, which is minimized,

$$\chi^2 = \sum_i \left(\frac{I_{\rm measured}(i) - I_{\rm predicted}(i)}{\sigma_{\rm measured}(i)} \right)^2, \tag{3}$$

where the summation is over all observed voxels indexed by i, I is the intensity and σ is the standard error of the measurement. The predicted intensity ($I_{\rm predicted}$) is initially unknown. For linear scaling models, the predicted intensity can be written schematically as a linear transformation of the 'true' intensity I_0 as follows,

$$I_{\rm predicted}(i) = o_i + m_i I_0(i), \tag{4}$$

where o_i is the offset and m_i is the scale factor.

Combining equations (3) and (4), and rearranging terms,

$$\chi^2 = \sum_i \left(\frac{I_{\rm scaled}(i) - I_0(i)}{\sigma_{\rm scaled}(i)} \right)^2, \tag{5}$$

where $I_{\rm scaled}(i) = m_i^{-1}[I_{\rm measured}(i) - o_i]$ and $\sigma_{\rm scaled}(i) = m_i^{-1}\sigma_i$ are the inverse-scaled observations and uncertainties.

The equation for optimally merging scaled intensities can be derived from equation (5) as the value of I_0 that minimizes χ^2, with the following result,

$$I_{\rm merged}(\mathbf{h}) = \frac{\sum_j w_j I_{\rm scaled}(j)}{\sum_j w_j}, \tag{6}$$

where $w_j = [\sigma_{\rm scaled}(j)]^{-2}$ are the weights and the sums run over all equivalent reflections j with Miller index $\mathbf{h}$ in the asymmetric unit of reciprocal space. The uncertainty is as follows:

$$\Delta I_{\rm merged}(\mathbf{h}) = \left(\sum_j w_j \right)^{-1/2}. \tag{7}$$

Both the model parameters and the merged intensity are unknown and must be estimated simultaneously. This leads to a nonlinear optimization problem. An iterative method may be applied when the scale factors are linear functions of model parameters (Hamilton *et al.*, 1965). Repeated cycles of merging and model fitting converge toward the solution that minimizes χ^2.

Table 1
Scaling-model parameters.

	a	*b*	*c*	*d*
Correction	Absorption	Illuminated volume	Background scattering	Detector sensitivity
Type	Scale factor	Scale factor	Offset	Scale factor
Coordinates	x, y, φ	φ	s, φ	x, y
Parameterization	3D linear interpolation	1D linear interpolation	2D linear interpolation	2D linear interpolation
Regularizers	Laplacian (x, y), second derivative (φ)	Second derivative (φ)	Second derivative (φ), second derivative (s), $\lvert c\rvert^2$	Laplacian (x, y)
Restraints	None	None	$c > 0$	None
*mdx*2 object	`AbsorptionModel`	`ScalingModel`	`OffsetModel`	`DetectorModel`

1.2.2. Scaling-model refinement. The formalism for least-squares optimization can be expressed conveniently in matrix form (Press, 2007). Conventionally, χ^2 is written as the l^2-norm of a vector of residuals as follows,

$$\chi^2 = \|\mathbf{Ax} - \mathbf{b}\|_2^2, \tag{8}$$

where **x** is a vector of unknowns, **b** is a known vector and **A** is a matrix. The unknowns are the solution to the linear equation

$$\mathbf{A}^{\rm T}\mathbf{Ax} = \mathbf{A}^{\rm T}\mathbf{b}, \tag{9}$$

which can be solved directly by matrix inversion if the problem is not underdetermined. When the model itself is underdetermined by the data, or when there is a danger of overfitting, it is often useful to apply restraints to the model, such as smoothness. In the method of regularized least squares, an additional term is added:

$$(\mathbf{A}^{\rm T}\mathbf{A} + \lambda\mathbf{B}^{\rm T}\mathbf{B})\mathbf{x} = \mathbf{A}^{\rm T}\mathbf{b}. \tag{10}$$

The matrix **B** is an operator that acts on the parameter vector **x** to calculate the quantity to minimize in an L2-norm sense. For instance, **B** may calculate a discrete representation of the second derivative in order to obtain solutions that are smooth (Press, 2007). The pre-factor λ is known as the regularization parameter, and it sets the trade-off between satisfying the restraints and goodness of fit.

As a concrete example, consider the general scaling model (equation 4) where the scale factor m_i depends only on the φ angle of the crystal. Let the model be parameterized by a set of scale factors (the parameters) at regular intervals, where the scales for each observation are calculated by linear interpolation between the control points. The linear interpolation can be expressed as a linear operator acting on a vector of parameters as follows:

$$\mathbf{m} = \mathbf{Mx}. \tag{11}$$

By algebraic manipulation of equation (3), it can be shown that the least-squares solution for **x** is obtained from equation (3) with the following substitutions:

$$\mathbf{A} = {\rm diag}(\mathbf{I}_0/\sigma)\mathbf{M}, \tag{12}$$

$$\mathbf{b} = (1/\sigma) \circ (\mathbf{I}_{\rm measured} - \mathbf{o}). \tag{13}$$

In the above notation, the diag function creates a diagonal matrix from its vector argument, and the open-circle and forward-slash (/) operators denote element-wise multiplication and division of vectors, respectively.

Based on this formalism, we have applied the following general procedure to derive more complex scaling models.

(i) Write the complete scaling model in terms of its prediction for each observation given the 'true' intensity.

(ii) Choose a convenient parameterization (typically linear intepolation on a grid of dimension 1–3).

(iii) For each scale factor and offset, derive a linear operator that calculates the value for each observation given the relevant coordinates of the observation and a vector of parameters.

(iv) Rearrange equation (3) to find the **A** matrix and **b** vector to be used in least-squares minimization.

(v) Derive operators to use in regularization (typically enforcing a notion of smoothness).

2. Methods

2.1. Scaling model

We recently introduced a scaling model that accounts for common experimental artifacts in diffuse scattering data (Meisburger *et al.*, 2020). The model contains four physically motivated parameters, labeled *a*–*d*, that describe a linear transformation of the true intensity I_0 at each observation point i (equation 4), as follows:

$$I_{\rm predicted}(i) = a_i d_i[b_i I_0(i) + c_i]. \tag{14}$$

The three scale factors (*a*, *b* and *d*) and the offset term (*c*) are themselves functions of the coordinates for each observation: the rotation angle (φ), position on detector (x, y) and scattering-vector magnitude (s). The coordinate dependence of each parameter and its physical meaning are summarized in Table 1.

Numerically, the parameters are represented by discrete samples on a grid of dimension 1–3, and the value at the coordinates of each observation is computed by linear interpolation (Table 1). Interpolation is computed using a matrix operator (Meisburger *et al.*, 2021) acting on a vector of control points, as in equation (11). In one-dimensional interpolation, the vector is simply an ordered list of the discrete samples. In higher dimensions, the grid of control points is vectorized, and the interpolation operators are constructed in a corresponding manner.

The general procedure to refine a given parameter is to alternate between two linear least-squares minimizations: first, to find the 'true' intensity I_0 that minimizes χ^2 with the scaling model held constant (equation 6), and second, to find the scaling parameter values that minimize χ^2 plus the regularizing terms (equation 10) given the previous value for the merged intensity. The procedure is repeated for all four parameters in the scaling model. To simplify the implementation, each least-squares minimization step is expressed in a common notation, where the $\mathbf{A}$ matrix and $\mathbf{b}$ vector appearing in equation (10) are derived for each parameter, and $\mathbf{M}$ represents the linear interpolation operator (Table 2).

Each parameter has one or more associated regularizing operators to enforce a notion of smoothness (Table 1). We have chosen operators with straightforward $\mathbf{B}$-matrix implementations (Press, 2007): a discrete second-derivative operator regularizes the φ- and s-dependence and a two-dimensional discrete Laplacian operator regularizes the (x, y)-dependent parameters. Each operator is weighted independently by a regularization parameter during least-squares minimization (λ in equation 10). Reasonable values for the regularization parameters are estimated by the fitting program (see Section 2.2), but in general they can be tuned by the user to control the smoothness of the solution based on an understanding of the experiment.

The offset parameter (c) has additional restraints and regularizers in order to make the correction as small as the data allow. This is important because an arbitrary offset may be added and subtracted from the merged intensity and the scaling model, respectively, leading to the same χ^2 in equation (8). Physically, it is assumed that at least some of the data do not require offset corrections; *i.e.* the beam intersects only the crystal during at least some part of the scan, and the experimental background has been correctly subtracted. The offset correction is therefore always positive, and an additional regularization operator (an identity matrix) penalizes large values of the parameter. The positivity restraint is nonlinear and cannot be enforced using the regularized least-squares formalism of equation (10). Instead, the restraint is applied during iterative refinement of the offset parameter, an approach that is widely used for such restraints in the context of multivariate curve resolution (Cichocki & Zdunek, 2007; de Juan *et al.*, 2014). After fitting the parameter values, any negative values are set to zero. If all of the points are positive, a constant is subtracted to make at least one of the values equal to zero. The fit is then repeated until convergence is reached (described further in Section 2.2).

2.2. Implementation of scaling-model refinement in *mdx*2

The operations for scaling and merging described above were implemented in Python following the same modular approach that is used for other functions in *mdx*2. First, a library within *mdx*2 called *scaling* was created that is essentially a toolkit from which algorithms may be built. The algorithm itself is implemented in the command-line function `mdx2.scale`, which imports the *scaling* library, handles file inputs and outputs, sets regularization parameters and controls flow through the algorithm (such as the order of refinement operations and halting conditions). To maintain compabitility with existing tutorials (Meisburger & Ando, 2023), the default behavior of `mdx2.scale` is to run a primitive scaling model where only the b term is refined. The full scaling model is activated using the flag `--mca2020`, which tells the program to mimic the behavior and default parameters from *mdx-lib* (Meisburger *et al.*, 2020). The full scaling model with default parameters is designed to work well for data sets collected in the manner described here (*i.e.* rotation data with background subtraction and high redundancy). However, to maintain flexibility the algorithm can be customized at the command line using non-default parameters and individual terms in the scaling model can be disabled entirely if required. As in all *mdx*2 command-line programs, the options and their default values are printed using the `--help` flag (Section S2).

Table 2
Least-squares problem for each scaling parameter.

	$\mathbf{A}$	$\mathbf{b}$
a	$\mathrm{diag}[\mathbf{d} \circ (\mathbf{b} \circ \mathbf{I}_0 + \mathbf{c})/\sigma]\mathbf{M}$	$\mathbf{I}/\sigma$
b	$\mathrm{diag}(\mathbf{d} \circ \mathbf{a} \circ \mathbf{I}_0/\sigma)\mathbf{M}$	$(\mathbf{I} - \mathbf{a} \circ \mathbf{c} \circ \mathbf{d})/\sigma$
c	$\mathrm{diag}(\mathbf{d} \circ \mathbf{a}/\sigma)\mathbf{M}$	$(\mathbf{I} - \mathbf{a} \circ \mathbf{c} \circ \mathbf{d} \circ \mathbf{I}_0)/\sigma$
d	$\mathrm{diag}[\mathbf{a} \circ (\mathbf{b} \circ \mathbf{I}_0 + \mathbf{c})/\sigma]\mathbf{M}$	$\mathbf{I}/\sigma$

The *scaling* library contains a hierarchy of objects (Python classes) that build on one another to execute scaling operations for each type of parameter with minimal duplication of code. At the lowest level of the hierarchy are the linear interpolation objects (`InterpLin1`, `InterpLin2` and `InterpLin3`) that compute sparse-matrix representations of the linear interpolation operators in 1–3 dimensions, as well as the Laplacian and second-derivative operators used in regularization. Corresponding objects were previously implemented in the MATLAB library *mdx-lib* and were translated directly into Python following the example in our *REGALS* package for small-angle scattering data analysis (Meisburger *et al.*, 2021), which has both MATLAB and Python implementations of `InterpLin1`.

Each type of scaling parameter is stored using a corresponding Python object with a consistent interface (Table 1). The objects store parameter values on a grid of scan coordinates, and these values can be converted to/from NeXus data arrays for input and output.

The `ScaledData` object stores the unscaled observations, their reciprocal-space indices, the scan coordinates and the current value of the overall scale and offset for each observation. It includes functions to facilitate the selection of batches (*i.e.* single sweeps of data) via the Python iterator syntax and for merging equivalent observations using equation (6).

Each scaling-model object (Table 1) has a corresponding `ModelRefiner` object with a common interface. The function `calc_problem` returns the matrix–matrix and matrix–vector products used in least-squares minimization (equation 10 and Table 2), and the `fit` method performs the least-squares minimization step given the current merged intensity and scaling parameters. Regularized least squares is used with regularization parameters passed as arguments to the `fit` method. Each regularization parameter is re-normalized following the approach used in *REGALS* (Meisburger *et al.*, 2021),

Table 3
Crystallization.

Method	Sitting-drop vapor diffusion
Plate type	Cryschem 24-well (Hampton Research)
Temperature (K)	295
Protein concentration (mg ml^{-1})	20
Buffer composition of protein solution	20 m*M* sodium phosphate dibasic
Composition of reservoir solution	300 m*M* sodium phosphate pH 10.0, 10 m*M* sodium EDTA
Volume and ratio of drop	16 µl, 1:1
Volume of reservoir (ml)	500

$$\lambda = \alpha\,\mathrm{trace}(\mathbf{A}^{\mathrm{T}}\mathbf{A})/\mathrm{trace}(\mathbf{B}^{\mathrm{T}}\mathbf{B}),$$

where α is the regularization parameter specified by the user and λ is the pre-factor of the regularization term $\mathbf{B}^{\mathrm{T}}\mathbf{B}$ (see equation 10). This re-normalization means that a certain value of α will produce similar results despite different grid spacings, data-set sizes and noise levels in the data. In general, $\alpha \ll 1$ will favor minimization of χ^2, while $\alpha \gg 1$ will favor minimization of the regularizer. For default values of α, see the supporting information.

2.3. Parallel processing in *mdx*2

Computationally intensive processing steps in *mdx*2 can be efficiently distributed over multiple processors if available. The image stack is divided among the workers in three-dimensional segments that are aligned with the HDF5 chunks specified in `mdx2.import_data`. In *mdx*2 version 1.0, multiprocessing is available in `mdx2.import_data`, `mdx2.find_peaks`, `mdx2.mask_peaks` and `mdx2.integrate`. This feature is activated by specifying the number of processors with the `--nproc` flag. Note that in `mdx2.import_data`, the performance is currently limited by the single-threaded HDF5 output file writer. In other cases, performance scales with the number of processors.

2.4. Experimental methods

Insulin crystals in the zinc-free cubic form were prepared following published protocols (Faust *et al.*, 2008). Insulin from bovine pancreas was purchased as a lyophilized powder (Sigma, catalogue No. I5500) and used without further purification. Drop volume, pH and precipitant concentration were optimized to favor the growth of large single crystals (Table 3).

X-ray diffraction data were collected at ambient temperature on beamline F1 at the Cornell High Energy Synchrotron Source (CHESS) following diffuse scattering protocols introduced previously (Meisburger *et al.*, 2020; Pei *et al.*, 2023). The beamline and data-collection parameters are summarized in Table 4. At the synchrotron, two large insulin crystals were harvested from the same drop of a crystallization tray using Kapton loops. The loops were immediately placed in plastic capillaries pre-loaded with reservoir solution to maintain hydration. A total of 17 data sets were collected from distinct locations on the two crystals using fine φ-slicing and low dose per frame. A total exposure time of 50 s for each location was chosen to limit radiation damage to tolerable levels, as judged by the *B*-factor decay obtained during scaling. For each crystal, a set of background scattering measurements were made by translating the crystal out of the beam along the rotation axis and collecting 360° of data at 1° s^{-1} while acquiring images at 1 Hz (1 s and 1° per frame).

Table 4
Data collection.

X-ray source	CHESS beamline F1
Wavelength (Å)	0.9768
Energy bandwidth ($\Delta E/E$) (%)	∼0.05
Beam size, profile	100 µm diameter round, flat-top
Detector	PILATUS 3 6M, 1.0 mm silicon sensor
Rotation rate (° s^{-1})	1.0
Rotation per image (°)	0.1
Temperature (K)	295
Crystal mount	800 µm diameter loop (MiTeGen DT) inside PET capillary (MiTeGen Micro-RT) with 10 µl reservoir solution
No. of crystals	2
No. of diffraction images	8500

3. Results and discussion

3.1. A high-redundancy room-temperature data set for diffuse scattering

To validate and test the new capabilities of *mdx*2, we chose to process a room-temperature multi-crystal data set from cubic insulin. Diffraction data were collected from two nominally identical crystals (Table 4) using a multi-sweep strategy to distribute the dose (Fig. 2). To determine whether all frames collected are of consistent quality, we processed the Bragg data using a *DIALS* multi-crystal workflow (Fig. 1*c*). According to the *B*-factor decay model fit during scaling, the onset of radiation damage was immediate and it progressed in a similar manner at each location (Fig. 2, bottom axes). The change in overall *B* factor during exposure ranged from 2 to 4 Å^2 per 50° of rotation, which is comparable to previously reported diffuse scattering data sets from lysozyme (Meisburger *et al.*, 2020, 2023). The data were processed to a resolution of 1.20 Å (Table 5). With each crystal processed separately, statistics such as $R_{\mathrm{p.i.m.}}$, mean $I/\sigma(I)$ and CC$_{1/2}$ indicate that the data are of excellent quality (Crystal 1 and Crystal 2, Table 5). These statistics improve further when the two crystal data sets are processed together (Crystals 1 and 2, Table 5), indicating that the crystals are highly isomorphous and of comparable quality. The final merged data set has a multiplicity of 68.5, which far exceeds that of any macromolecular diffuse scattering data set reported to date.

3.2. Data reduction in *mdx*2

After initial processing in *DIALS*, each rotation data set (sweep) was imported into *mdx*2 (Fig. 1*c*). In the data-import step (`mdx2.import_data`), the diffraction images were re-compressed with three-dimensional chunks chosen to encompass each detector panel and 2° of rotation, or 20 images (see supporting information). Because the subsequent

Table 5
Bragg data-collection statistics.

Values in parentheses are for the highest resolution shell. Unit-cell uncertainty refers to the standard deviation among data sets (sweeps).

	Crystals 1 and 2	Crystal 1	Crystal 2
Crystal parameters			
Space group	$I2_13$	$I2_13$	$I2_13$
$a = b = c$ (Å)	79.48±0.09	79.55±0.11	79.517±0.005
Data-set statistics			
Resolution (Å)	28.11–1.20 (1.22–1.20)	28.11–1.20 (1.22–1.20)	28.11–1.20 (1.22–1.20)
$\mathrm{CC}_{1/2}$	1.000 (0.276)	1.000 (0.115)	1.000 (0.226)
R_{merge}	0.054 (2.236)	0.058 (2.539)	0.046 (1.758)
$R_{\mathrm{p.i.m.}}$	0.006 (0.769)	0.009 (1.113)	0.007 (0.868)
Mean $I/\sigma(I)$	44.8 (0.6)	28.9 (0.3)	37.2 (0.6)
Completeness (%)	99.81 (96.32)	98.38 (77.68)	99.08 (83.97)
Multiplicity	68.5 (8.7)	36.6 (5.7)	32.5 (4.8)
Observations	1795925 (10983)	944563 (5742)	846333 (5241)
Unique reflections	26218 (1256)	25842 (1013)	26026 (1095)

processing steps distribute data among workers (CPU cores) according to chunk shape, the choice can affect performance. We have not systematically evaluated different chunk sizes; however, the initial choice worked well in tests. The background images were also imported and re-binned every 10° and 20 pixels in each direction, reducing each stack of 360 images with 2463 × 2527 pixels to an array of 36 × 124 × 127 (`mdx2.bin_image_series`).

The next task in data processing was to accumulate the diffuse scattering on a three-dimensional grid. In order to mask out Bragg peaks, the coordinates of all pixels recording more than 20 photons (see Section 1.1.2) were fitted to a three-dimensional Gaussian distribution in reciprocal space (`mdx2.find_peaks`). A peak mask was created at each Bragg peak location by thresholding the Gaussian distribution at three standard deviations (`mdx2.mask_peaks`). For integration, a grid was chosen that oversampled the reciprocal lattice by a factor of three in each direction (for example, each voxel region spans Miller indices −1/6 to 1/6, 1/6 to 1/2, 1/2 to 5/6 *etc.*). Because space group $I2_13$ has the reflection condition $h + k + l = 2n$, only half of the voxels with all-integer Miller indices contain Bragg peaks, and thus for every Bragg peak there are $3^3 \times 2 - 1 = 17$ voxels of diffuse scattering. Finally, the integrated intensities were corrected for the experimentally measured background scattering as well as for polarization, air absorption, detector efficiency and solid angle (`mdx2.correct`).

3.3. Scaling and analysis of systematic errors

In order to correct for systematic errors, a multi-crystal scaling model was refined (`mdx2.scale` with the flag `--mca2020`). As described in Section 2.1, this model accounts for changes in illuminated volume, excess background scattering, absorption of diffracted X-rays and detector flat-field errors (equation 14 and Table 1). Default values were used for the grid dimensions, regularization parameters, outlier rejection thresholds and stopping conditions (see Section 2.2 and the supporting information).

In general, the refined scaling-model parameters offer insight into the types of systematic errors that are present during data collection. This is particularly true of the highly redundant insulin data set, because the model parameters are expected to be robustly determined. We examined each correction visually; representative sets are presented in Fig. 3. The overall scale factor for each frame varies in a similar manner to the corresponding Bragg scale factor refined in *DIALS* (Supplementary Fig. S1), which is significant because Bragg intensities are not included in the diffuse data set refined in *mdx*2. Offset corrections varied greatly among the different wedges of data (Fig. 3*a*, Offset column). This variation is consistent with expectations; the large crystals were mounted with very little excess liquid on the surface (crystals in Fig. 2) and thus for most of the rotation range the beam should pass only through the crystal without encountering the loop or excess solvent. The absorption corrections were relatively large for this data set (Fig. 3*a*, Absorption column), with a variation across the detector of as much as ±10%, which is expected given the exceptionally large crystals used.

The detector flat-field correction was fitted globally to all data sets. In contrast to the detector gain correction implemented previously in *mdx-lib*, where a single scale factor was applied for each detector chip (96 parameters), the flat-field correction in *mdx*2 interpolates a grid covering the detector surface (200 × 200 = 40 000 parameters); the *mdx*2 correction

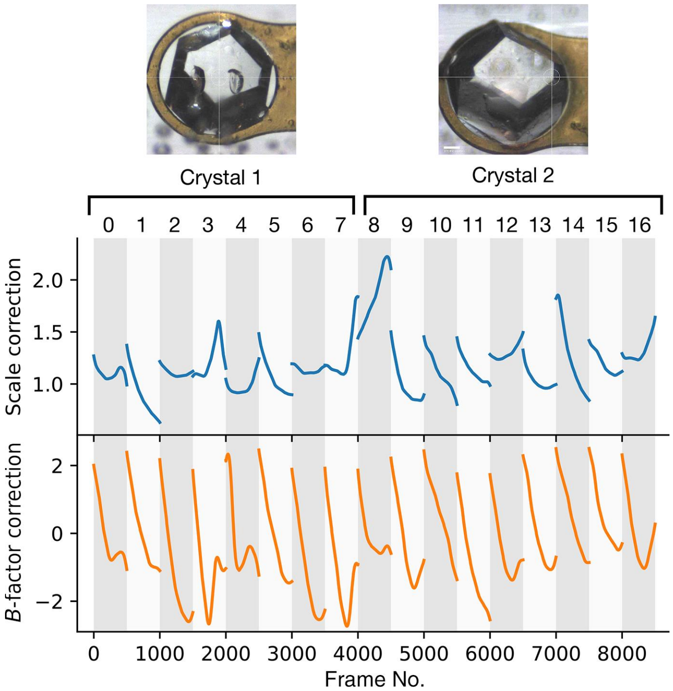
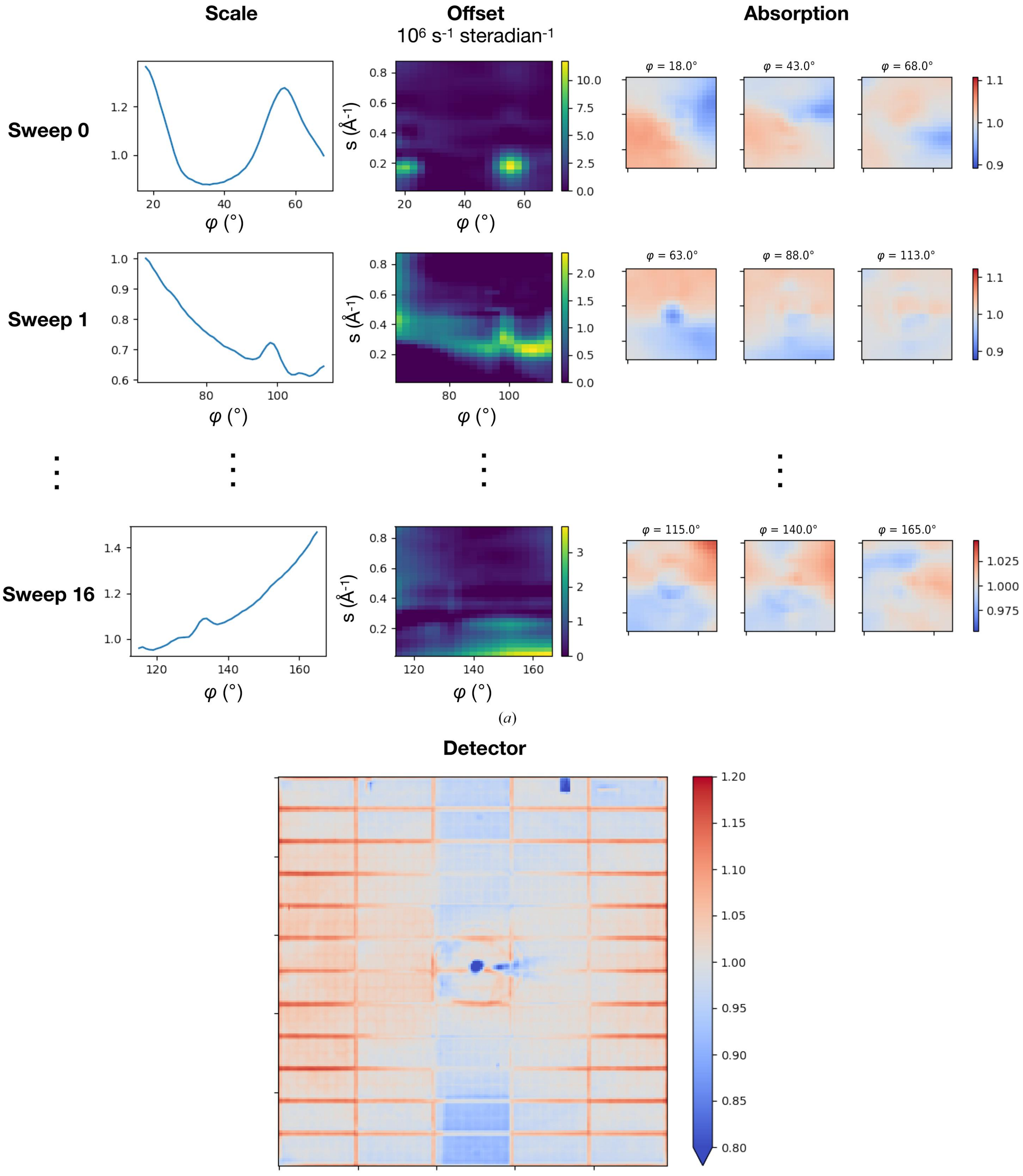


Figure 2
Bragg intensity scaling results from a multi-crystal insulin data set. Rotation data sets of 50° (500 frames each) were collected from 17 locations on two large insulin crystals (pictured). Each 50° wedge (sweep) spans the width of the white/gray vertical bars. Correction factors from global scaling in *DIALS* are shown for each wedge. The illuminated volume changes during rotation, as reflected in the overall scale factor (top axes, blue curves). *B* factors show a characteristic linear decay with exposure time due to radiation damage (bottom axes, orange curves). The overall extent of radiation damage is consistent among the data sets.



(*a*)

(*b*)

Figure 3
Diffuse scattering scaling model after refinement. The model was refined globally in *mdx*2 to all 17 wedges in the multi-crystal insulin data set. (*a*) Columns show each model parameter as a function of observation coordinates (rotation angle *φ*, detector position and scattering-vector magnitude *s*). Each row corresponds to a 50° wedge (sweep) of data (only three are shown for clarity). Panels, from left to right, show the scale correction (*b* parameter), the offset correction (*c* parameter) and the absorption correction (*a* parameter) at the beginning, middle and end of the rotation range. (*b*) The detector correction (*d* parameter) refined globally to all 17 wedges.

thus contains more potential detail about detector response than was available previously. Many of the detector errors were known in advance; for instance, one of the chips along the top row of the detector had malfunctioned and was reading lower values than the others. This chip is clearly visible in the corrections (blue rectangle in Fig. 3*b*, Detector). Another striking feature is the vertical stripe of approximately one panel width through the middle of the image. This feature results from absorption by the strip of Kapton film that suspends the beamstop. Finally, we note that data recorded at the edges of the detector panels have consistently lower intensity, and are boosted as much as ~20% by the scaling correction (red outlines around each panel in Fig. 3*b*). After scaling, the redundant observations were merged according to the Laue symmetry (`mdx2.merge`).

3.4. Statistical analysis of intensities and data-quality indicators

In all previously collected data sets from lysozyme (Meisburger *et al.*, 2020, 2023), the voxels near the Bragg peaks contain more intense diffuse scattering (halos) compared with voxels further away. This phenomenon is expected to be a general feature of protein crystal diffuse scattering as it relates to the long-range coupling of subtle lattice motions (Peck *et al.*, 2018; Polikanov & Moore, 2015; Wall *et al.*, 1997; De Klijn *et al.*, 2019). Halo features can be broad and anisotropic depending on the correlation length of lattice motions. In lysozyme, halos were found to decay gradually according to the inverse square of the distance from the Bragg peak (Meisburger *et al.*, 2020, 2023). Thus, it is not possible in general to separate the signals from internal molecular motion versus lattice disorder by filtering the data. Instead, a model that includes lattice disorder can be fitted to the total scattering, for instance using the *GOODVIBES* and *DISCOBALL* methods (Meisburger *et al.*, 2023). However, for the purpose of analyzing data quality, it is good practice to compute statistics separately for the smoothly varying and halo-containing parts of the signal, as the intense halos tend to dominate variances and correlation coefficients.

To roughly quantify the halo scattering present in the insulin data set, we split the merged data into two parts: the 'halo' part consisting of only those voxels with integer Miller indices satisfying the reflection condition ($h + k + l = 2n$), and

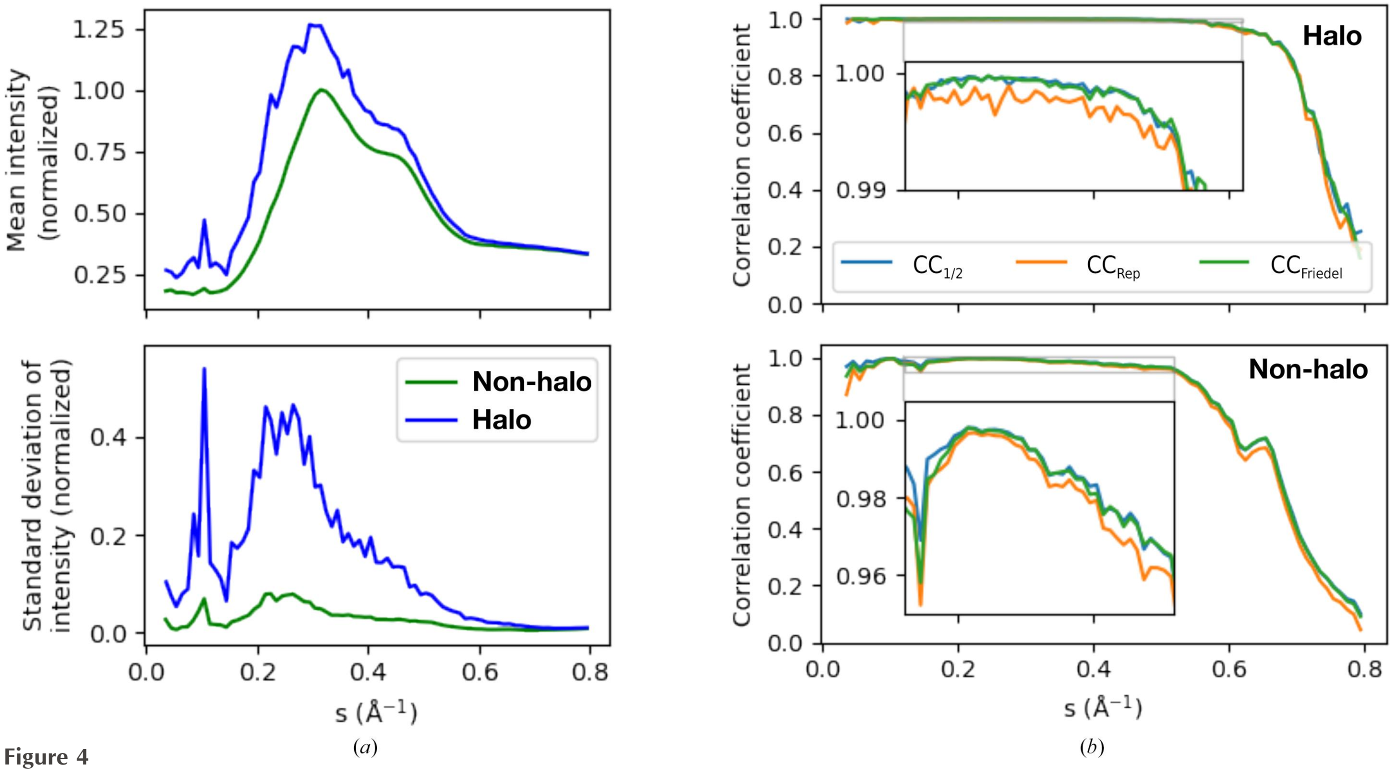


Figure 4
Diffuse intensity statistics and data-quality indicators. (*a*) Merged intensities were split between halo and non-halo voxels (see main text) and binned in shells of constant scattering vector *s*, where 1/*s* is the resolution. The mean intensity (top panel) rises to a peak at ~3 Å resolution (~0.3 Å^{-1}). The intensities are normalized to 1 at the non-halo maximum. The halo-containing voxels have higher intensity on average than non-halo voxels. The intensity variations of interest are quantified by the standard deviation within each resolution shell (bottom panel). The halo-containing voxels have an approximately fivefold greater signal than non-halo voxels. Within each resolution shell, the non-halo variations are less than 10% of the mean scattering, indicating that this signal is much more subtle. (*b*) The correlation coefficient for random half-data sets (CC$_{1/2}$) between crystals 1 and 2 scaled independently (CC$_{\rm Rep}$) and between half-data sets split according to Friedel symmetry (CC$_{\rm Friedel}$) are compared for halo-containing voxels (top panel) and non-halo voxels (bottom panel). Correlation coefficients are close to 1 for much of the resolution range (insets) and decay at high resolution as the signal-to-noise ratio decreases. The correlations are higher overall for halo-containing voxels, as expected from the higher signal strength. There is no significant difference between random selection of observations (CC$_{1/2}$) and grouping observations based on symmetry considerations (CC$_{\rm Friedel}$), which is expected for successful scaling. The CC$_{\rm Rep}$ statistic closely follows CC$_{1/2}$, showing that both the measurement and data-processing procedures are reproducible.

the rest, which we call 'non-halo'. Note that we do not expect the 'non-halo' fraction for insulin to be completely free of scattering related to lattice disorder, as halos tend to decay gradually in protein crystals, as noted above. The 'halo' voxels in this case correspond to a cubic region of reciprocal space with each side having Miller indices −1/6 to 1/6 with the central Bragg peak removed. The 'halo' voxels include fractional scattering vectors from the (mean) radius of the peak mask at 0.102 to the corner of the voxel at $3^{1/2}/6 = 0.289$, corresponding to wave-like lattice displacements with wavelengths between 275 and 780 Å.

To better understand the components of the diffuse scattering signal, we computed the mean and standard deviation of the merged intensity within shells of constant resolution (Fig. 4*a*). The mean intensity rises to a peak at ∼3 Å resolution (Fig. 4*a*, top panel), a common feature of diffuse scattering from protein crystals originating from multiple sources: disordered solvent present in the pores of the crystal, uncorrelated atomic motion and some short-range correlated motion (Meinhold & Smith, 2005; Wall *et al.*, 2014). The mean intensity for the halo fraction is greater than that of the non-halo fraction (Fig. 4*a*, top panel), which is consistent with the presence of diffuse scattering from lattice disorder. The intensity variations, which contain the signal of greatest interest for studies of protein dynamics, can be quantified by the standard deviation of the intensity in each resolution shell. The variations are significantly larger for the halo fraction (Fig. 4*a*, bottom panel) compared with the non-halo part. The non-halo variations are small – less than 10% of the mean diffuse scattering in each resolution shell – underscoring the importance of careful measurement and scaling to determine this signal accurately.

The precision of diffraction data may be quantified by statistics reporting the agreement between equivalent observations. For diffuse scattering, we have previously quantified precision within each resolution shell using $CC_{1/2}$, the Pearson correlation coefficient of intensity merged from random half-data sets (Meisburger *et al.*, 2020, 2023). To facilitate this

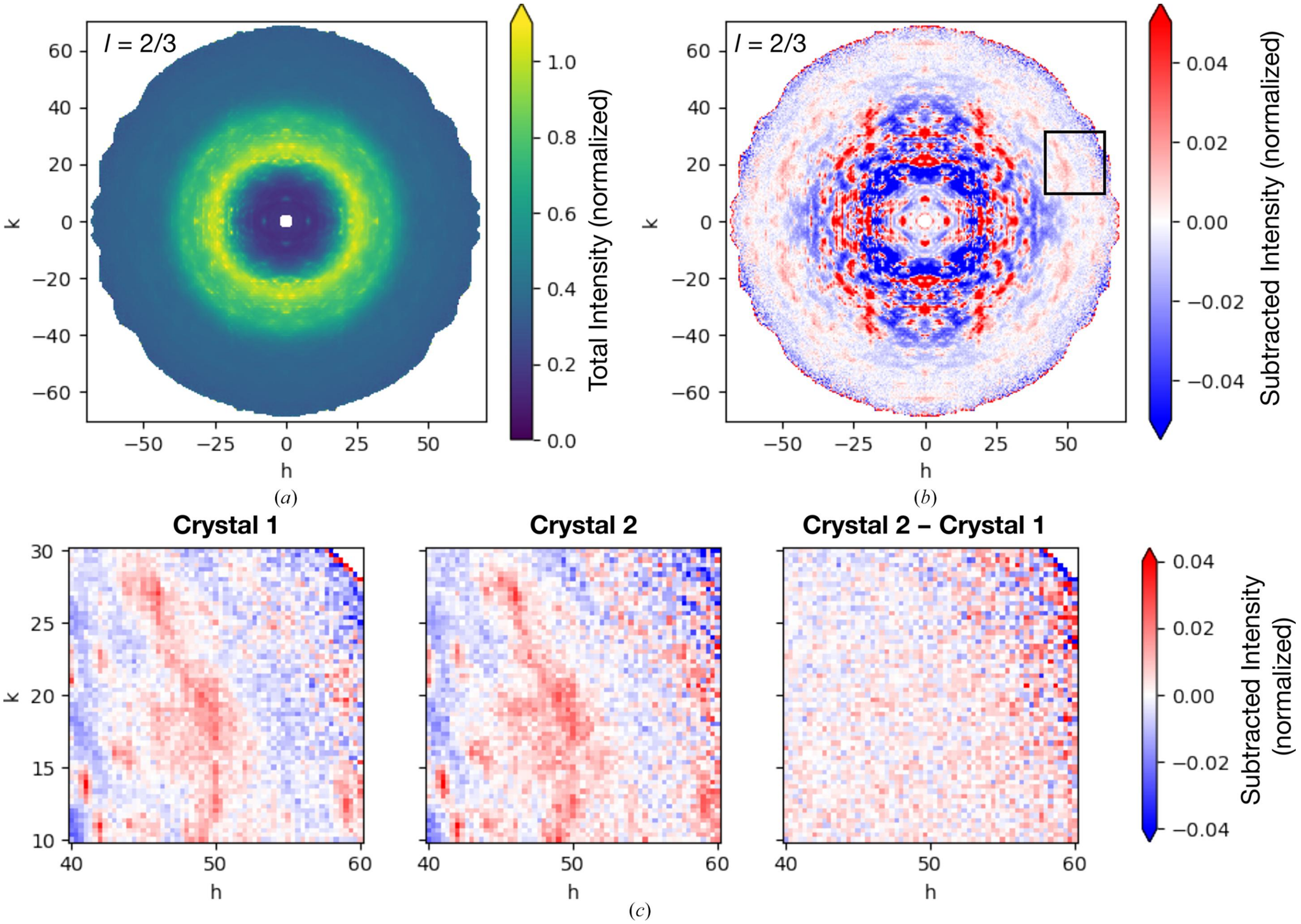


Figure 5
Visualization of weak diffuse scattering features. (*a*) A slice through the symmetry-expanded, three-dimensional map from cubic insulin in a non-halo plane (*l* = 2/3). The intensity is normalized as in Fig. 4. (*b*) The same slice with the mean subtracted from each resolution shell to reveal subtle intensity variations. Features are visible at high resolution. (*c*) Comparison of the boxed region in (*b*) for two maps obtained by independently scaling and merging data from the two crystals. The pattern is reproduced in both maps and the difference between the two appears to be random (right panel). The code used to generate this figure is provided in the supporting information.

calculation, a data-splitting feature has been implemented in `mdx2.merge`, where various criteria may be used. To compute $CC_{1/2}$, we split the data using a random-shuffling algorithm that distributes equal numbers of equivalent observations between the two half-data sets (`--split randomHalf` option). The merged half-data sets are present as separate columns in the data table output by `mdx2.merge`. Because the scattering in halo voxels exceeds that in non-halo voxels, they will tend to dominate statistics such as $CC_{1/2}$. Thus, the halo and non-halo parts were analyzed separately. Correlation coefficients were close to 1 for both halo and non-halo parts (Fig. 4*b*, blue lines in top and bottom panels, respectively). In general, halo voxels have a slightly higher correlation than non-halo voxels at all resolutions, consistent with their greater signal-to-noise ratio. The signals are measured with very high precision in regions of high signal to noise: within all resolution shells up to 2 Å, $CC_{1/2}$ exceeds 0.95 for non-halo voxels and 0.99 for halo voxels. In both cases, $CC_{1/2}$ decays at high resolution as the signal diminishes. However, it remains significantly greater than zero up to 1.25 Å resolution ($s < 0.8$ Å^{-1}), even for the more subtle non-halo signal (Fig. 4*b*, bottom panel), suggesting that meaningful diffuse signal is present throughout reciprocal space.

A potential limitation of using the correlation between half-data sets to quantify precision is that such statistics may be inflated by certain systematic errors. A recent study compared various statistical measures of precision to address this possibility (Su *et al.*, 2021). Here, we follow a similar strategy. One alternative to the random split used in $CC_{1/2}$ is to split according to Friedel symmetry (*i.e.* whether or not the symmetry operator mapping the observation to the asymmetric unit contains a center of inversion). Because equivalent observations differing by a center of inversion tend to be measured far apart in terms of their scan coordinates (for example detector position or sample rotation angle), the correlation coefficients from such a split will emphasize systematic differences in the data. To test this idea, we implemented such a split in `mdx2.merge`, which is activated by the `--split Friedel` option.

For the insulin data set, we find that $CC_{\rm Friedel}$ and $CC_{1/2}$ are indistinguishable across all resolution shells, both for the halo-containing voxels (Fig. 4*b*, top panel) and for the more subtle patterns in the non-halo voxels (Fig. 4*b*, bottom panel). The equivalence of $CC_{\rm Friedel}$ and $CC_{1/2}$ would be expected if the scaling model corrects for all of the significant differences between equivalent observations, and thus it can be used to verify that a particular scaling model is sufficiently realistic.

A second, alternative splitting scheme is possible if multiple crystals are measured. In general, crystals have different shapes and are mounted in different orientations, and thus when comparing these independent data sets there is less chance of spurious correlations arising from the chance alignment of a crystal symmetry with a particular geometric effect, such as directional differences in absorption (Su *et al.*, 2021). The correlation coefficient between independent data sets measures reproducibility, both of the measurement itself as well as the scaling procedure, and is called $CC_{\rm Rep}$ (Su *et al.*, 2021).

We repeated the scaling and merging steps for each insulin crystal separately and computed $CC_{\rm Rep}$, the correlation between independent measurements. We find that $CC_{\rm Rep}$ is very close to $CC_{1/2}$ in all resolution shells. When examined closely (Fig. 4*b*, insets), we find that $CC_{\rm Rep}$ is always slightly below $CC_{1/2}$, suggesting that $CC_{1/2}$ overestimates reproducibility by a small margin. The reproducibility is still excellent; we attribute this result to the high redundancy of the insulin data and the highly symmetric Laue group, which ensures that the scaling model is tightly constrained by the data. Next, we investigated how data quality depends on the number of data sets that are merged. As expected when the data sets are equivalent, $CC_{1/2}$ improves at all resolutions as more data sets are added (Supplementary Fig. S2). The high multiplicity of the data was necessary in order to obtain good signal to noise for the non-halo data beyond ~2 Å resolution ($s > 0.5$ Å^{-1}). In summary, this comparison of intensity statistics suggests that meaningful diffuse signal is present at all resolutions and that systematic errors have been sufficiently corrected.

3.5. Visualizing weak features

To visualize the diffuse patterns, we symmetry-expanded and exported the merged data as a three-dimensional array (`mdx2.map`). A slice through this map containing only non-halo voxels (l = 2/3) is shown in Fig. 5(*a*). The pattern is dominated by the mostly isotropic scattering. To better visualize the variational features, we subtracted the isotropic scattering. Here, we defined the isotropic part as the mean non-halo scattering from each resolution shell interpolated at the reciprocal-space coordinates of each voxel (a complete script is provided in the supporting information). In this subtracted map, the variations are visible throughout (Fig. 5*b*).

Our analysis of the data-quality metrics, discussed above, suggests that the diffuse pattern at high resolution contains significant signal despite the measurement noise, and that it is uncorrupted by systematic errors. To verify that the patterns are indeed accurately determined, we compared maps that were scaled and merged separately from the two insulin crystals. As an illustrative example, we chose a small region containing a recognizable pattern (the boxed region in Fig. 5*b*). Agreement between the reciprocal-space maps of the two crystals is excellent, both in terms of the precise pattern and its overall magnitude (Fig. 5, left and middle panels). Moreover, when one crystal map is subtracted from the other, the residual appears to be random (Fig. 5*c*, right panel). In general, such visual tests can be important to build confidence in the accuracy of a data set, which is especially important when very subtle diffuse features such as these are used to support models of correlated motion.

4. Conclusions

We have described *mdx2*, a user-friendly software package for processing macromolecular diffuse scattering that incorporates state-of-the-art algorithms for data processing. A detailed scaling model, newly available in *mdx*2 version 1.0, fully corrects for systematic errors in multi-crystal experiments. The complete processing of a multi-crystal data set was demonstrated using the *mdx*2 and *DIALS* command-line interfaces. This application is a clear example of how the close connection between *mdx*2 and *DIALS* can be utilized to build complete workflows through simple scripting. At the same time, *mdx*2 enables interactive data exploration through its commitment to the NeXus format for data interchange. *mdx*2 can also be imported as a Python toolkit for implementing custom algorithms. While most of the data-processing functionality of *mdx-lib* is now available in *mdx*2, a few methods have not yet been reimplemented. In particular, Compton scattering corrections and absolute intensity scaling (Meisburger *et al.*, 2020) are planned for future versions.

The high-quality diffuse scattering maps generated for insulin highlight the potential benefits of high-redundancy data collection for diffuse scattering. When paired with the detailed scaling model in *mdx*2, redundancy can be leveraged to reduce the impact of systematic errors. Here, we achieve ~70-fold redundancy by collecting data from two large crystals in a high-symmetry space group. If the crystals are low symmetry, or cannot be obtained with sufficient size, it may be necessary to collect data serially (*i.e.* from a large number of small crystals). Our results suggest that the high redundancy inherent to serial crystallography may be a useful feature for diffuse data processing, so long as background scattering can be minimized. We expect future versions of *mdx*2 to include methods optimized for serial data.

The macromolecular diffuse scattering field is currently small, in part because substantial technical expertise has been required to process the data. However, based on the rapid progress enabled by modern detectors and computational methods, we can envision a future where diffuse scattering is part of the standard macromolecular crystallography toolkit. Ultimately, diffuse scattering promises to enhance our understanding of protein dynamics and answer fundamental questions in biochemistry and biophysics. To achieve this goal, it is important to make diffuse scattering techniques accessible to researchers addressing important biological questions. *mdx*2 represents a step in this direction.

6. Data and software availability

The *mdx*2 software package is available on GitHub (https://github.com/ando-lab/mdx2) and is free to use (GNU General Public License version 3). Version 1.0 as described here has been archived with Zenodo (https://doi.org/10.5281/zenodo.10519719). The diffraction images are available for download from Zenodo (https://doi.org/10.5281/zenodo.10515006). Bash scripts for reprocessing the data are included in the supporting information. The figures can be reproduced by executing the Jupyter notebooks in the `insulin-multi-crystal` example included with the *mdx*2 source code.

Acknowledgements

We thank Neti Bhatt, Haoyue Wang and Xiaokun Pei for testing the software and providing early feedback. The translation of sparse-matrix interpolation code from MATLAB to Python followed the example performed by Darren Xu in the *REGALS* software package. We thank Kevin Dalton for getting us started with *dxtbx*, and the *DIALS* development team for answering questions when they arose during development. The software was run on shared computing resources available at CHESS, and we are grateful to the CLASSE computing group, especially Devin Bougie, for assistance with Python environments and for installing and maintaining a Jupyter Hub server that was used extensively for this project.

Funding information

Experiments were performed at the Center for High-Energy X-ray Sciences (CHEXS), which is supported by the National Science Foundation (BIO, ENG and MPS Directorates) under award DMR-1829070, and the Macromolecular Diffraction at CHESS (MacCHESS) facility, which is supported by award P30-GM124166 from the National Institute of General Medical Sciences, National Institutes of Health and by New York State's Empire State Development Corporation (NYSTAR). This work was supported by National Institutes of Health grant GM124847 (to NA).

References

Arnold, O., Bilheux, J., Borreguero, J., Buts, A., Campbell, S., Chapon, L., Doucet, M., Draper, N., Ferraz Leal, R., Gigg, M., Lynch, V., Markvardsen, A., Mikkelson, D., Mikkelson, R., Miller, R., Palmen, K., Parker, P., Passos, G., Perring, T., Peterson, P., Ren, S., Reuter, M., Savici, A., Taylor, J., Taylor, R., Tolchenov, R., Zhou, W. & Zikovsky, J. (2014). *Nucl. Instrum. Methods Phys. Res. A*, **764**, 156–166.

Case, D. A. (2023). *Methods Enzymol.* **688**, 145–168.

Cichocki, A. & Zdunek, R. (2007). *Advances in Neural Networks – ISNN 2007*, edited by D. Liu, S. Fei, Z. Hou, H. Zhang & C. Sun, pp. 793–802. Berlin, Heidelberg: Springer.

Estermann, M. A. & Steurer, W. (1998). *Phase Transit.* **67**, 165–195.

Evans, P. (2006). *Acta Cryst.* D**62**, 72–82.

Faust, A., Panjikar, S., Mueller, U., Parthasarathy, V., Schmidt, A., Lamzin, V. S. & Weiss, M. S. (2008). *J. Appl. Cryst.* **41**, 1161–1172.

Förster, A., Brandstetter, S. & Schulze-Briese, C. (2019). *Philos. Trans. R. Soc. A*, **377**, 20180241.

Gildea, R. J. & Winter, G. (2018). *Acta Cryst.* D**74**, 405–410.

Grosse-Kunstleve, R. W., Sauter, N. K., Moriarty, N. W. & Adams, P. D. (2002). *J. Appl. Cryst.* **35**, 126–136.

Hamilton, W. C., Rollett, J. S. & Sparks, R. A. (1965). *Acta Cryst.* **18**, 129–130.

Juan, A. de, Jaumot, J. & Tauler, R. (2014). *Anal. Methods*, **6**, 4964–4976.

Klijn, T. de, Schreurs, A. M. M. & Kroon-Batenburg, L. M. J. (2019). *IUCrJ*, **6**, 277–289.

Könnecke, M., Akeroyd, F. A., Bernstein, H. J., Brewster, A. S., Campbell, S. I., Clausen, B., Cottrell, S., Hoffmann, J. U., Jemian, P. R., Männicke, D., Osborn, R., Peterson, P. F., Richter, T., Suzuki, J., Watts, B., Wintersberger, E. & Wuttke, J. (2015). *J. Appl. Cryst.* **48**, 301–305.

Meinhold, L. & Smith, J. C. (2005). *Phys. Rev. Lett.* **95**, 218103.

Meisburger, S. P. & Ando, N. (2017). *Acc. Chem. Res.* **50**, 580–583.

Meisburger, S. P. & Ando, N. (2023). *Methods Enzymol.* **688**, 43–86.

Meisburger, S. P., Case, D. A. & Ando, N. (2020). *Nat. Commun.* **11**, 1271.

Meisburger, S. P., Case, D. A. & Ando, N. (2023). *Nat. Commun.* **14**, 1228.

Meisburger, S. P., Xu, D. & Ando, N. (2021). *IUCrJ*, **8**, 225–237.

Mueller, M., Wang, M. & Schulze-Briese, C. (2012). *Acta Cryst.* D**68**, 42–56.

Parkhurst, J. M., Brewster, A. S., Fuentes-Montero, L., Waterman, D. G., Hattne, J., Ashton, A. W., Echols, N., Evans, G., Sauter, N. K. & Winter, G. (2014). *J. Appl. Cryst.* **47**, 1459–1465.

Peck, A., Lane, T. J. & Poitevin, F. (2023). *Methods Enzymol.* **688**, 169–194.

Peck, A., Poitevin, F. & Lane, T. J. (2018). *IUCrJ*, **5**, 211–222.

Pei, X., Bhatt, N., Wang, H., Ando, N. & Meisburger, S. P. (2023). *Methods Enzymol.* **688**, 1–42.

Polikanov, Y. S. & Moore, P. B. (2015). *Acta Cryst.* D**71**, 2021–2031.

Press, W. H. (2007). *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press.

Scheidegger, S., Estermann, M. A. & Steurer, W. (2000). *J. Appl. Cryst.* **33**, 35–48.

Schreurs, A. M. M., Xian, X. & Kroon-Batenburg, L. M. J. (2010). *J. Appl. Cryst.* **43**, 70–82.

Simonov, A., De Baerdemaeker, T., Boström, H. L., Ríos Gómez, M. L., Gray, H. J., Chernyshov, D., Bosak, A., Bürgi, H.-B. & Goodwin, A. L. (2020). *Nature*, **578**, 256–260.

Su, Z., Dasgupta, M., Poitevin, F., Mathews, I. I., van den Bedem, H., Wall, M. E., Yoon, C. H. & Wilson, M. A. (2021). *Struct. Dyn.* **8**, 044701.

The NeXpy Development Team (2023). *NeXpy*. https://github.com/nexpy/nexpy.

Van Benschoten, A. H., Liu, L., Gonzalez, A., Brewster, A. S., Sauter, N. K., Fraser, J. S. & Wall, M. E. (2016). *Proc. Natl Acad. Sci. USA*, **113**, 4069–4074.

Wall, M. E. (2009). *Methods Mol. Biol.* **544**, 269–279.

Wall, M. E. (2018). *IUCrJ*, **5**, 172–181.

Wall, M. E., Clarage, J. B. & Phillips, G. N. (1997). *Structure*, **5**, 1599–1612.

Wall, M. E., Van Benschoten, A. H., Sauter, N. K., Adams, P. D., Fraser, J. S. & Terwilliger, T. C. (2014). *Proc. Natl Acad. Sci. USA*, **111**, 17887–17892.

Wall, M. E., Wolff, A. M. & Fraser, J. S. (2018). *Curr. Opin. Struct. Biol.* **50**, 109–116.

Welberry, T. R. & Weber, T. (2016). *Crystallogr. Rev.* **22**, 2–78.

Winter, G., Beilsten-Edmands, J., Devenish, N., Gerstel, M., Gildea, R. J., McDonagh, D., Pascal, E., Waterman, D. G., Williams, B. H. & Evans, G. (2022). *Protein Sci.* **31**, 232–250.

Winter, G., Gildea, R. J., Paterson, N., Beale, J., Gerstel, M., Axford, D., Vollmar, M., McAuley, K. E., Owen, R. L., Flaig, R., Ashton, A. W. & Hall, D. R. (2019). *Acta Cryst.* D**75**, 242–261.

Wych, D. C. & Wall, M. E. (2023). *Methods Enzymol.* **688**, 115–143.

Xu, D., Meisburger, S. P. & Ando, N. (2021). *Biochemistry*, **60**, 2331–2340.

Zimmerman, S. B. & Trach, S. O. (1991). *J. Mol. Biol.* **222**, 599–620.