

# *darfix* – data analysis for dark-field X-ray microscopy

Júlia Garriga Ferrer,\* Raquel Rodríguez-Lamas,\* Henri Payno, Wout De Nolf, Phil Cook, Vicente Armando Solé Jover, Can Yildirim and Carsten Detlefs

ESRF– The European Synchrotron, 71 Avenue des Martyrs, CS40220, 38043 Grenoble Cedex 9, France.

\*Correspondence e-mail: julia.garriga@esrf.fr, raquel.rodriguez-lamas@esrf.fr

Received 8 June 2022

Accepted 23 February 2023

Edited by A. Bergamaschi, Paul Scherrer Institut, Switzerland

**Keywords:** X-ray optics; software; data analysis.

A Python package for the analysis of dark-field X-ray microscopy (DFXM) and rocking curve imaging (RCI) data is presented. DFXM is a non-destructive diffraction imaging technique that provides three-dimensional maps of lattice strain and orientation. The *darfix* package enables fast processing and visualization of these data, providing the user with the essential tools to extract information from the acquired images in a fast and intuitive manner. These data processing and visualization tools can be either imported as library components or accessed through a graphical user interface as an Orange add-on. In the latter case, the different analysis modules can be easily chained to define computational workflows. Operations on larger-than-memory image sets are supported through the implementation of online versions of the data processing algorithms, effectively trading performance for feasibility when the computing resources are limited. The software can automatically extract the relevant instrument angle settings from the input files' metadata. The currently available input file format is EDF and in future releases HDF5 will be incorporated.

## 1. Introduction

Dark-field X-ray microscopy (DFXM) is a novel full-field imaging technique that non-destructively maps the 3D structure, orientation and strain of deeply embedded crystalline elements, such as grains or domains (Simons *et al.*, 2015; Poulsen *et al.*, 2017; Poulsen, 2020; Yildirim *et al.*, 2020). Direct-space images are formed by placing an X-ray objective lens along the diffracted beam, affording a spatial resolution of the order of 100 nm, while maintaining a working distance between the sample and X-ray objective lens that is in the centimetre-range.

The first implementation of a dedicated dark-field X-ray microscope was recently installed on beamline ID06-HXM of the European Synchrotron Radiation Facility (Kutsal *et al.*, 2019). Since its installation, this instrument has been used to investigate a variety of scientific subjects, including the domain evolution in ferroelectrics (Simons *et al.*, 2018), the austenitic transformation in shape memory alloys (Bucsek *et al.*, 2019), recovery in metals (Mavrikakis *et al.*, 2019; Ahl *et al.*, 2020), embedded particles in steel (Hlushko *et al.*, 2020), visualization of dislocation structures (Jakobsen *et al.*, 2019; Dresselhaus-Marais *et al.*, 2021a) and the structure of biominerals (Cook *et al.*, 2018; Schoeppler *et al.*, 2022).

DFXM is conceptually similar to dark-field electron microscopy in transmission electron microscopy, which is used to selectively image strain and orientation across materials science, physics, geoscience and numerous other fields (Williams & Carter, 1996; Nellist & Pennycook, 2000; Morones *et al.*, 2005).



While the *darfix* Python package has been specifically designed as a tool to facilitate the data treatment of DFXM data in particular, it can be used in a larger scope for imaging and diffraction data obtained from other techniques. So far, *darfix* fully supports the EDF file format. Data treatment of files in HDF5 format is enabled for datasets containing a stack of images as a function of one external parameter. Full support for HDF5 is planned for a future release.

## 2. Dark-field X-ray microscopy

The geometry of DFXM is illustrated in Fig. 1; further details are given by Poulsen *et al.* (2017) and Kutsal *et al.* (2019). A nearly monochromatic and nearly collimated X-ray beam illuminates the sample. This beam may be condensed in the vertical and horizontal directions to increase the beam intensity in the field of view, or to selectively illuminate a thin layer within the sample. The energy bandwidth is of the order  $\Delta E/E \simeq 1.4 \times 10^{-4}$ .

The goniometer is designed to access diffraction angles in the vertical scattering plane (where  $\mathbf{k}_0$  and  $\mathbf{k}_d$  lie in the  $x$ - $z$  plane) and probe reciprocal space only in the immediate vicinity of a given Bragg reflection ( $h, k, l$ ). The current implementation of DFXM at ID06-HXM, ESRF, achieves this by moving the sample along a combination of  $\mu$ ,  $\omega$ ,  $\chi$  and  $\phi$  rotation stages, see Fig. 1. A detailed description of the stacking order of the motors in the goniometer is given by Poulsen *et al.* (2017). The direction of the diffracted beam is

characterized by the scattering angle,  $2\theta$ , and the azimuthal angle,  $\eta$ . In most experiments the sample is aligned such that  $\eta \simeq 0$ .

The optical axis of an X-ray objective is aligned to the diffracted beam to produce a magnified image (inverted in both directions) on the 2D detector. The magnification can be calculated from the distances  $d_1$  (sample to objective) and  $d_2$  (objective to detector), and experimentally verified by observing the displacement of the image upon small calibrated translations of the sample.

In addition to the magnified DFXM images, there are further 2D detectors to record non-magnified images, *e.g.* a ‘near-field camera’ positioned directly downstream of the sample (Kutsal *et al.*, 2019). This can be used for classical diffraction topography and its extension, rocking curve imaging (Tran Thi *et al.*, 2017).

### 2.1. Scan types

The 2D images recorded as a function of the different rotation angles form the main data sets of DFXM. *darfix* facilitates the treatment of the individual raw images and the systematic analysis of scans.

The following recurring scan types are commonly used in DFXM:

**2.1.1. Rocking curve imaging.** Rocking curve imaging is an extension of classical diffraction topography. The diffraction topograph is measured as a function of the ‘rocking’ angle,  $\mu$  (see Fig. 1). Rocking curve imaging is typically performed without the magnifying objective lens, using the near-field camera. In this case the resolution in the  $2\theta$  and transverse ( $\chi$ ) directions is relatively low (Tran Thi *et al.*, 2017). It can, however, also be carried out in DFXM mode.

**2.1.2. Mosaicity scans.** Mosaicity scans can be seen as a generalization of rocking curve imaging, taking advantage of the improved resolution in the transverse ( $\chi$ ) direction due to the limited angular acceptance of the objective lens (Poulsen *et al.*, 2017).

**2.1.3. Strain scans.** In strain scans, the scattering angle  $2\theta$  is varied in order to probe spatial variations in the  $d$ -spacing of selected Bragg reflections (Poulsen *et al.*, 2017, 2021). Typically, strain maps are constructed from a series of rocking scans performed consecutively over a given range of scattering angles.

**2.1.4. Combined mosaicity-strain scans.** By recording a (2D) mosaicity map instead of a (1D) rocking curve scan at each  $2\theta$  position, all three directions in reciprocal space are probed. This is the most complete type of DFXM scan. However, as a 3D mesh of motor positions must be scanned, this scan type also requires the longest data acquisition time and yields the largest raw data volumes.

**2.1.5. Reciprocal space mapping.** This type of scan is similar to rocking curve imaging, without the objective lens. The images are recorded on a large-field-of-view camera positioned in the far-field [*i.e.*  $\sim 5$  m from the sample (Kutsal *et al.*, 2019)]. This type of scan provides angular information in three dimensions,  $\mu$  (scan axis),  $\eta$  and  $2\theta$  [derived from the

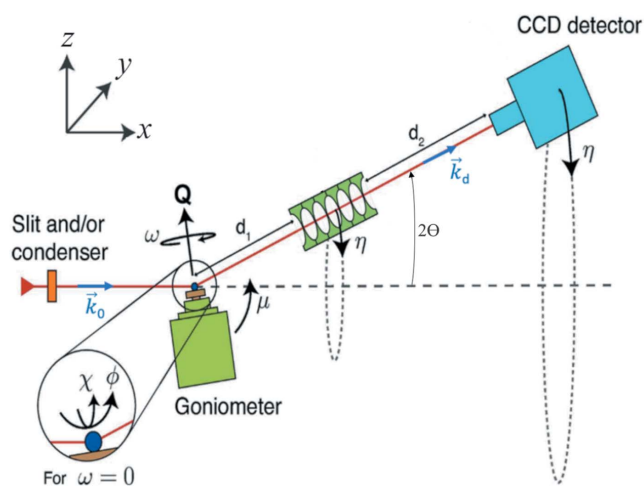


Figure 1

Geometry of the dark-field X-ray microscope at ID06-HXM at the ESRF. The incident beam  $\mathbf{k}_0$  travels along the laboratory  $x_1$  axis. The optical axis of the objective lens is aligned to the direction of the diffracted beam  $\mathbf{k}_d$ . The pivot point of the goniometer and sample is coincident with the intersection of these two optical axes. Vector  $\mathbf{Q}$  defines the local scattering vector at a given point  $\mathbf{r}(x, y, z)$  within the sample, and may be parameterized by the scattering angle  $2\theta$ , the azimuthal angle  $\eta$  and the length of the vector  $|\mathbf{Q}|$ . The value of  $|\mathbf{Q}|$  is related to the spacing of the lattice plane being measured,  $d_{hkl}$ , and the X-ray wavelength,  $\lambda$ , by Bragg’s law. The goniometer is associated with a base tilt,  $\mu$ , an  $\omega$ -rotation around  $\mathbf{Q}$  and two tilts,  $\chi$  and  $\phi$ .  $d_1$  is the distance from the sample to the entry point of the objective and  $d_2$  is the distance from the exit point of the objective to the detector. The positive directions of the angles are indicated. This figure is adapted from Poulsen *et al.* (2021).

pixel position, see Fig. 1 and Poulsen *et al.* (2017)]. Similar to rocking curve real-space imaging, the analysis includes integrated intensity and center-of-mass visualization. Reciprocal space mapping can be used, for example, to determine twin relationships between domains in ferroelastic materials (Gorfman *et al.*, 2022).

**2.1.6. Layer scans.** All of the scans listed above can be performed with a line-focused beam illuminating a single  $\sim 150$  nm-thick layer within the sample. Layers recorded at a series of heights within the sample can then be combined into 3D volume maps.

### 3. The *darfix* codebase

*darfix* is a Python library that provides a set of computer vision techniques for the analysis of dark-field X-ray microscopy data. It is split into two modules: (1) a back-end that contains the core processing utilities, algorithm implementations and abstractions, and (2) a graphical user interface (GUI). The typical user would interact with the core module through the GUI, but it is possible and straightforward to access the core methods directly by importing that part of the library. The central code object is `Dataset`, which encapsulates the properties of a stack of images (multiple rotation angles can vary within a single stack of images, each axis adds a dimension to the image stack). Operations on *darfix* (e.g. blind source separation, hot pixel removal, shift correction) are implemented as isolated tasks acting on a dataset and returning a dataset. This allows computational workflows to be defined by chaining several operations together. Furthermore, as datasets are commonly of the order of several hundred gigabytes, the software provides online versions for many of the data processing algorithms: the output is built incrementally as more data are brought in. The modules can then be processed sequentially in chunks if memory usage is limited. This is particularly relevant for dataset sizes larger than the available memory. If the option ‘use data from disk’ is selected when loading the data, these online versions will be automatically used throughout the workflow.

*darfix* is not offered as a stand-alone application, but includes an Orange (Demšar & Zupan, 2013) add-on for the definition of workflows. Orange is a platform for performing data analysis and visualization, which allows for the creation of workflows using the *darfix* GUI. A typical workflow comprises tasks such as raw data selection, background removal, region-of-interest selection, scan variable identification, scan processing (e.g. rocking curve fitting), display of the results, *etc.* (see Fig. 3). All the tasks in *darfix* are independent (in the GUI as well as in the back-end part), so that for every task there is a different dedicated widget. These widgets are based on *Qt* (The Qt Company, 2021) and *silx* (Vincent *et al.*, 2021). They can be linked through the workflow created in Orange. In addition, workflows in Orange can be called from the command line with different input datasets. The algorithms that were used in the original workflow will be then reproduced without the need of the GUI. Moreover, a workflow defined from the GUI can be saved with its settings

and reused using *ewoks* (De Nolf *et al.*, 2022). This allows users to launch the same workflow (algorithms and settings) but with a different input data set. This is particularly useful when the same workflow has to be carried out for a series of datasets, e.g. the same analysis has to be applied to every layer in a 3D volume map.

*darfix* contains a series of image processing methods for the analysis of DFXM scans. These methods can be organized into three groups: the selection of the data, the pre-processing algorithms, and the analysis and visualization of the results. To provide a complete overview of the current modules and their functionalities, a workflow exported from the GUI is represented in Fig. 2.

The following sections provide an overview of the used techniques. A simplified version of a *darfix* workflow, focused on data pre-processing, is presented in Fig. 3(a).

#### 3.1. File inputs: data selection

Currently, *darfix* accepts the ESRF data format (.edf) of the ID06 detectors, and is able to automatically characterize the relevant instrument angle settings by analyzing the input files’ metadata. HDF5 (The HDF Group, 1997–2022; Collette, 2013) is accepted, but only without dimension definition as the analysis of metadata is still under development.

Output data can be saved from the GUI in different formats: EDF, HDF5 [NeXus data format (Könnecke *et al.*, 2015)], CSV, TIFF, NumPy, ASCII and PNG. In the analysis there is usually the possibility to export specific data in a HDF5 file, which can then be used for any required post-processing. Alternatively, when working without the GUI, NumPy arrays can be extracted from the `Dataset` object for further processing.

Modified data are automatically saved onto disk under a folder chosen by the user. By default, only the output dataset of the most recent task of the workflow is saved, but there exists the possibility to copy the output of every task by appending a *Data copy* widget to it.

Exporting data in a format compatible with 3D visualization and processing software, e.g. *Paraview* (Ahrens *et al.*, 2005; Ayachit, 2015), is planned for a future release.

#### 3.2. Pre-processing

**3.2.1. Noise removal.** Noise in the data is practically unavoidable, and it can either come from the lens, the environment or the diffraction of the sample. The first step of the pre-analysis, after defining the dimensions of a dataset (or second if we apply a region of interest to the data) is to detect and remove this noise from the sequence of images. *darfix* provides the following tools for noise removal.

**Background subtraction.** This is a widely used approach that calculates the foreground mask of an image by subtracting it from a background model containing the static part of an image sequence. In *darfix*, the background is an image that can either be calculated using the pixel-wise mean or the median of a set of data images (for example, dark images from the scan). This background image is then subtracted from each of

the original images to obtain the foreground mask. While both mean and median subtractions can be used depending on the input dataset, the pixel-wise mean is less robust to outliers. On the other hand, the pixel-wise median is more computationally

expensive when data are not in memory. In that last case, the user has two options:

- (i) Select chunks of a certain shape so that the median is computed consecutively in all of them. Although this method

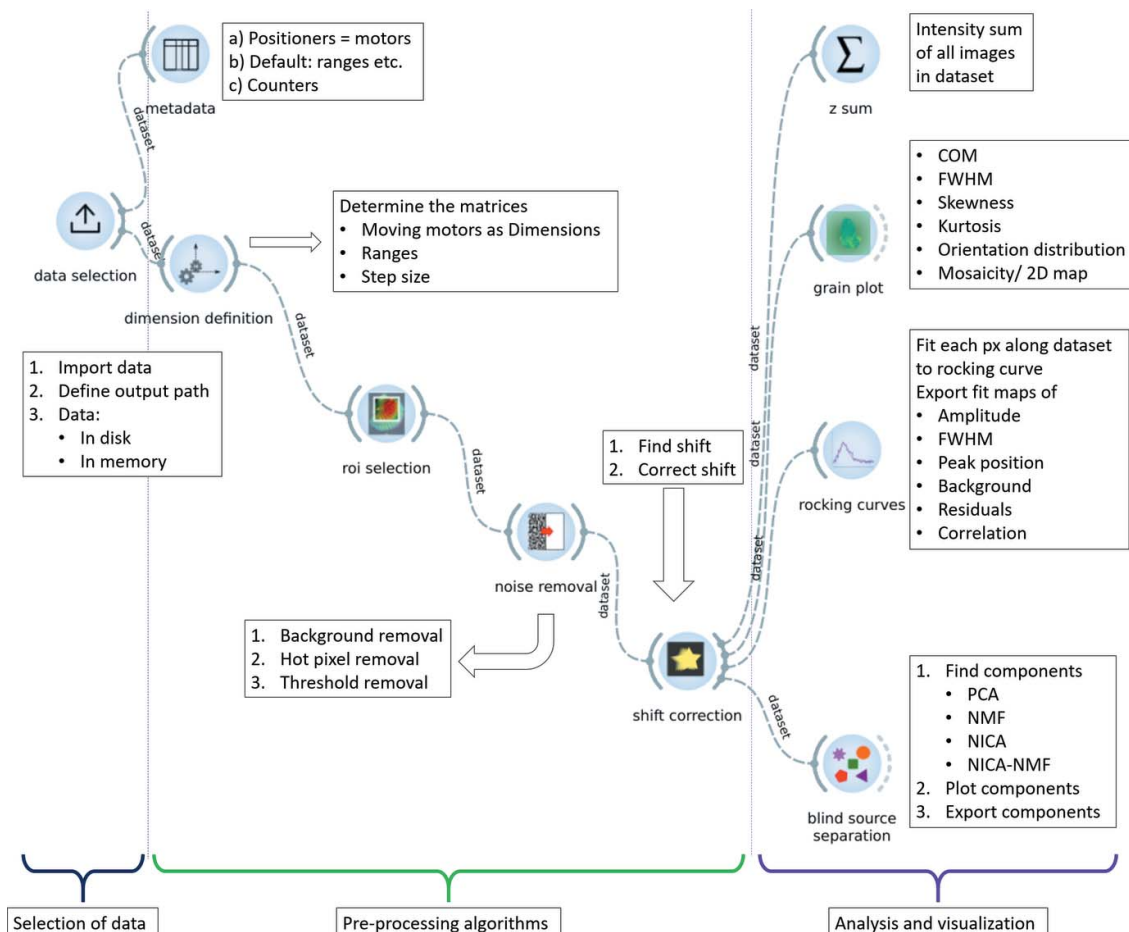


Figure 2 *darfix* GUI where a full workflow is shown. The three main modules – selection of data, pre-processing, and analysis and visualization – are indicated by brackets. A description of each widget’s functionalities is given in the labels.

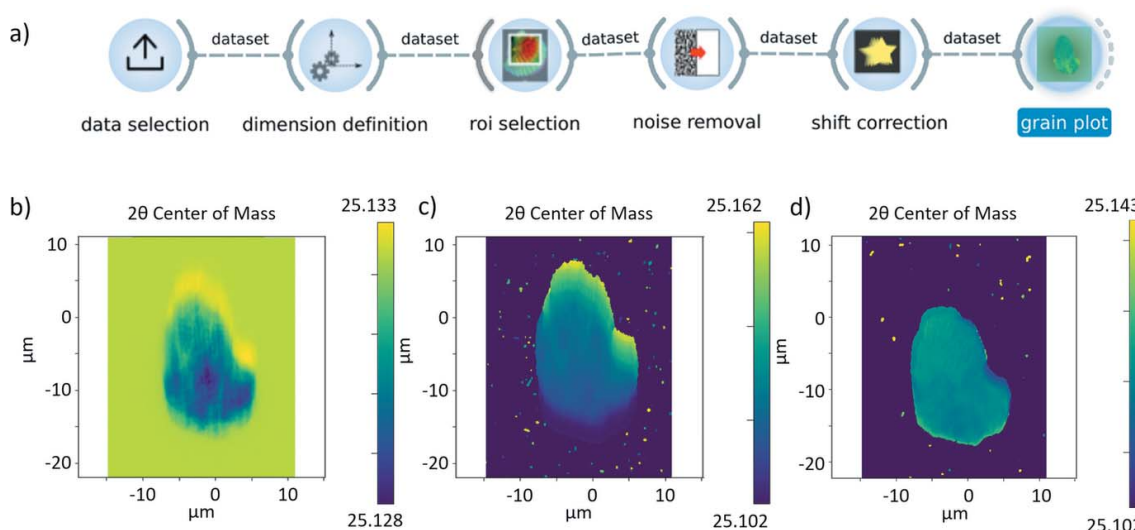


Figure 3 (a) *darfix* GUI where a data preprocessing thread is shown, finalized by the *Grainplot* widget. (b) Raw data plotted as COM. (c) After background removal and threshold removal. (d) After hot pixel removal and shift correction. The color bars represent scattering angles ( $2\theta$ ).

obtains a median of all the images, it is still time consuming as the input/output operations slow the process considerably.

(ii) Only use a subgroup of the stack of images to calculate the median: the user inputs a step  $k$  value so that the algorithm selects the images with index  $i, i + k, i + 2k, \dots$

**Hot pixel removal.** Sometimes, after performing background subtraction on the images, isolated groups of pixels appear at some or all of the images in the stack. These pixels are called ‘hot’ because they have higher intensity than the other pixels around them, and they are usually not part of the object we want to analyze but noise that needs to be treated. As the hot pixels are independent from one image to the other, this technique can be applied sequentially to the stack:

(i) Apply a median filter to every image of the stack and subtract it from the original image.

(ii) The hot pixels are identified as the ones with higher value than the standard deviation of this subtraction.

(iii) The hot pixels intensity values are replaced by the values of their corresponding pixels in the image with the median filter applied.

**Threshold removal.** Pixels with values lower than specified will be set to 0.

The use of any of these noise removal tools is optional.

**3.2.2. Image registration.** Alignment errors in the experimental hardware frequently cause a shift of the image as a function of rotation angle. *darfix* contains a module for the detection and correction of such shifts. We assume that the shift is a linear function of rotation angle, *i.e.*  $\mathbf{v}(\alpha) = \alpha \mathbf{v}_0$ , where  $\alpha$  is the rotation angle relative to the scan center, and  $\mathbf{v}_0$  is the displacement vector in pixels/degree. The algorithm used to find the shift, which has proven to give acceptable results in a considerably fast manner, is the following:

(i) Two images are obtained from the sum over the images taken at the motor positions of the first and second half of the dataset, respectively.

(ii) The shift vector  $\mathbf{v}'$  between the two resulting images is determined using the *scikit-image* (van der Walt *et al.*, 2014) function ‘`registration.phase_cross_correlation`’.

(iii) The linear shift is in the direction of the normalized shift:  $\mathbf{v}_0 = h \mathbf{v}'/|\mathbf{v}'|$ .

(iv) The scaling factor  $h$  remains to be determined – it depends on how the intensity of the images varies as function of  $\alpha$ :

$$\mathbf{v}_0^* = \frac{2\mathbf{v}_0}{\text{num\_images}}; \quad \varepsilon = 3\|\mathbf{v}_0^*\|_2$$

**for**  $h^* \leftarrow 0$  to  $3\varepsilon$  **do**

**for**  $i \leftarrow 0$  to  $\text{num\_images}$  **do**

$$\mathbf{v}^*(i) = ih^* \mathbf{v}_0$$

  Shift image  $i$  using  $\mathbf{v}^*(\alpha)$  via an affine transformation

**end for**

  Compute score using normalized variance of the images pixel-wise sum

**end for**

$h$  is the  $h^*$  with maximum score.

Once  $\mathbf{v}$  has been determined, the images are shifted with subpixel accuracy using the Discrete Fourier Transform (DFT) algorithm. The library OpenCV (Bradski, 2000) is used both for the affine transformations and for the DFT algorithm.

One could use other strategies, *e.g.* a non-linear fit, to optimize  $h$ . In some cases, for example when summing the uncorrected images leads to blurring, better results are obtained when the shift correction is run several times.

It is possible to find and apply the shift along a chosen dimension. In this case, a different shift is detected for every value of the chosen dimension. These shifts can then be applied to their corresponding images.

### 3.3. Scan analysis

After pre-processing, data can be further analyzed according to the type of scan performed during the experiment, see above in Section 2.1.

**3.3.1. Rocking curves fitting.** As described in Section 2.1.1, *darfix* provides a rocking curve analysis.

The rocking curve imaging module provides functions similar to the *RCIA* code (Tran Thi *et al.*, 2017) within the workflow of *darfix*. Its main function is to analyze the rocking curve of each pixel by fitting to a peak shape, *e.g.* a Gaussian.

After the fit, maps of the fit parameters (constant background, integrated intensity, peak position and peak width) as a function of pixel position are generated (Tran Thi *et al.*, 2017), as shown in Fig. 4.

As a computationally less intensive alternative to fitting, moments of the intensity as a function of the angle can be computed. For an ideal Gaussian distribution, the zeroth-order moment corresponds to the integrated intensity, the first-order moment (center of mass) to the peak position, and the second-order moment (variance) to the square of the r.m.s. peak width. These can be used as starting values for the Gaussian fit. *darfix* also provides the third (skewness) and fourth (kurtosis) moments. The moments can also be presented as color maps.

The intensity of a pixel,  $I_{\text{meas},xy}$  (where  $xy$  indicates pixel position), as a function of motor position, typically the rocking angle  $\mu$ , is fitted to a Gaussian,

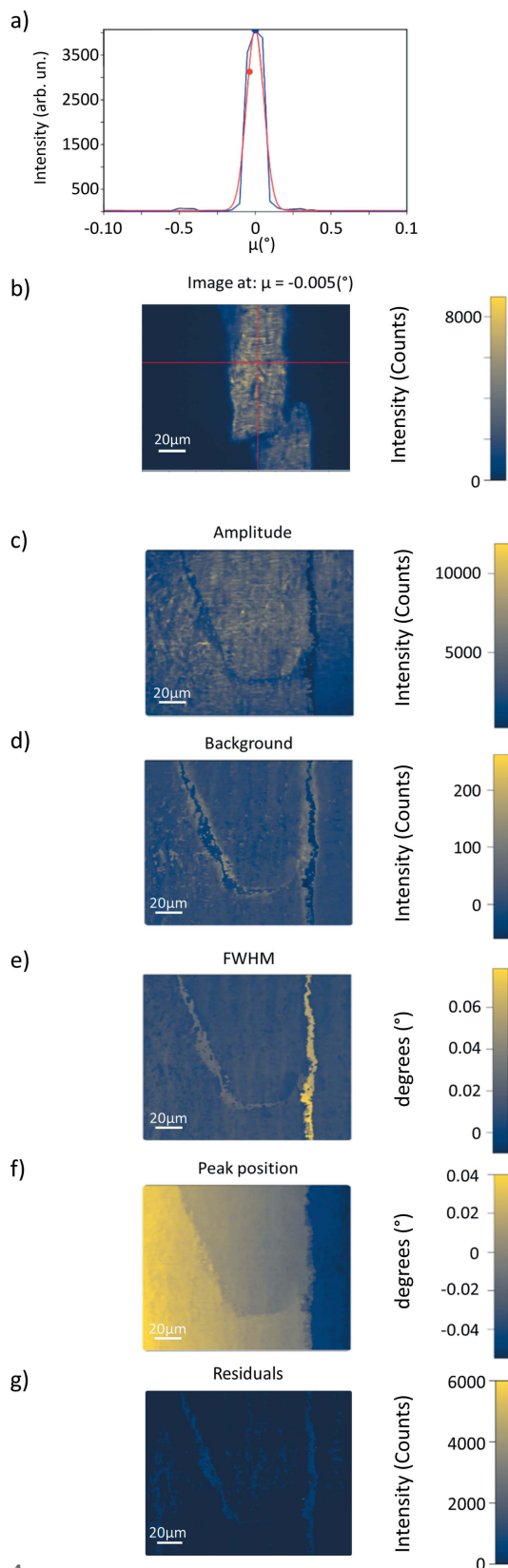
$$I_{\text{fit},xy}(\mu) = b_{xy} + A_{xy} \exp\left[-\frac{1}{2} \frac{(\mu - p_{xy})^2}{\sigma_{xy}^2}\right]. \quad (1)$$

Fitting all pixels in this way results in maps of the background ( $b_{xy}$ ), amplitude ( $A_{xy}$ ), peak position ( $p_{xy}$ ) and peak width [ $\sigma_{xy}$ , full width at half-maximum (FWHM) =  $2.355\sigma_{xy}$ ]. Furthermore, the fit generates a map of the  $\chi^2$  values,

$$\chi_{xy}^2 = \sum_{\text{images}} (I_{\text{fit},xy} - I_{\text{meas},xy})^2, \quad (2)$$

that can be used to assess deviations from a simple Gaussian profile. Note that  $\chi_{xy}^2$  in this context is *not* the goniometer motor  $\chi$ .

For faster results, pixels with low intensities can be omitted from the fit.



**Figure 4** Example of 1D rocking curve analysis. (a) Fit of the curve corresponding to the pixel indicated in (b). (b) Image at  $\mu = -0.005^\circ$  with a cross marking the pixel of interest. Maps of the full stack of images composing the rocking scan, generated by the pixel by pixel fit, of (c) amplitude, (d) background, (e) full width at half-maximum, (f) peak position and (g) residuals.

Two-dimensional scans, where, for example, two motors  $\mu$  and  $\chi$  are varied, can be fitted to a bivariate Gaussian,

$$I_{\text{fit},xy}(\mu, \chi) = b_{xy} + A_{xy} \exp \left\{ -\frac{1}{2(1-\rho_{xy}^2)} \left[ \left( \frac{\mu - p_{\mu,xy}}{\sigma_{\mu,xy}} \right)^2 - 2\rho_{xy} \left( \frac{\mu - p_{\mu,xy}}{\sigma_{\mu,xy}} \right) \left( \frac{\chi - p_{\chi,xy}}{\sigma_{\chi,xy}} \right) + \left( \frac{\chi - p_{\chi,xy}}{\sigma_{\chi,xy}} \right)^2 \right] \right\}. \quad (3)$$

Here,  $p_{\mu,xy}$  and  $\sigma_{\mu,xy}$  are maps of the peak position and peak width of motor  $\mu$ ,  $p_{\chi,xy}$  and  $\sigma_{\chi,xy}$  are maps of the peak position and peak width of motor  $\chi$ , and  $\rho_{xy}$  is the map of the Pearson correlation coefficient.

Additional line shapes are planned for a future release.

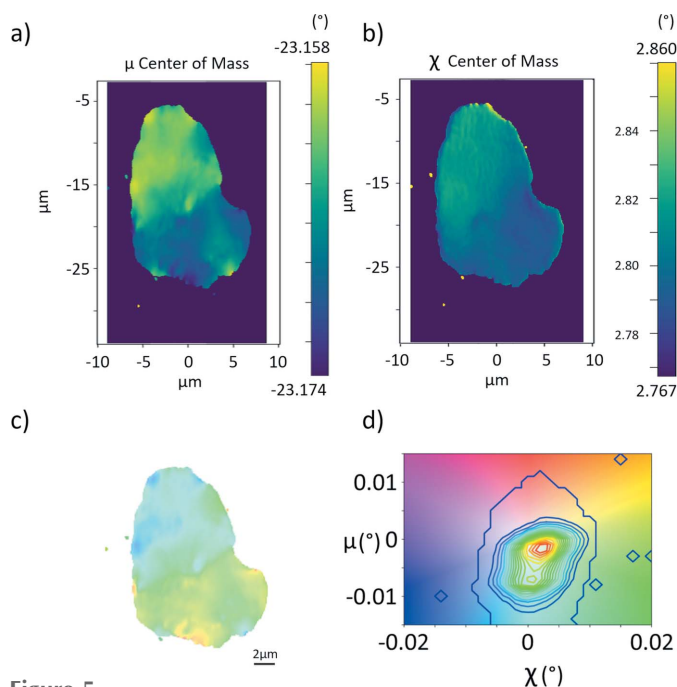
**3.3.2. Grainplot.** For scans where more than one motor varies, such as mosaicity ( $\mu$ - $\chi$  scans), the *Grainplot* tool can calculate moments, e.g. the integrated intensity, center-of-mass and standard deviation. This analysis works similarly to rocking curve imaging; the dependence of the intensity on two rotation angles, the ‘rocking’ and ‘rolling’ angles,  $\mu$  and  $\chi$ , respectively, is analyzed for each pixel. In particular, for mosaicity scans, the center of mass (COM) of the two scan motors are interpreted as components of a 2D vector in the color plane. Both components can then be displayed in a single graph. Intensity contours in the corresponding color map represent a local pole figure of the volume of interest. The first-order moments (COM) correspond to the local orientation of the diffracting planes. Due to the narrow acceptance of the objective lens in the  $2\theta$  direction, the magnitude of the scattering vector remains constant (Poulsen *et al.*, 2017) such that only shear strains and lattice rotations are measured (Poulsen *et al.*, 2021). Again, the results can be represented in a color map, using a 2D color plane that encodes both rotation angles. Some examples are shown in Fig. 5 (Section 4.4), where the COM color maps for  $\mu$  [Fig. 5(a)] and  $\chi$  [Fig. 5(b)] are shown.

**3.3.3. Blind source separation.** Blind source separation (BSS) comprises all techniques that try to decouple a set of source signals from a set of mixed signals with unknown (or very little) information (Herault *et al.*, 1985). Depending on the assumptions on the data, different BSS techniques can be used.

In DFXM, diffracting elements such as (sub)grains or ferroelastic domains can be interpreted as source signals that contribute to the images. BSS can then be used to identify these elements and extract the corresponding rocking curves, reciprocal space maps, *etc.* in a given dataset.

Specifically, we wish to reconstruct a matrix of observed signals ( $\mathbf{X}$ ) from a linear combination of unknown sources, encoded as rows in a matrix  $\mathbf{H}$ , so that  $\mathbf{X} \simeq \mathbf{WH}$ .  $\mathbf{W}$  is the so-called mixing matrix, and, in our case, we stack the flattened dataset images in rows to form the matrix  $\mathbf{X}$ . These matrices are obtained by optimizing a distance given by a matrix norm (e.g. the Frobenius norm).

The BSS techniques implemented in *darfix* are:



**Figure 5**

Center of mass maps for (a)  $\mu$  and (b)  $\chi$ , showing the relative motor values in angles. (c) Mosaicity map and (d) orientation distribution color key of the mosaicity map with an overlaid contour map of the integrated intensity, portraying the centered orientation distribution, indicating the angular spread for  $\mu$  and  $\chi$ .

(i) **Principal Component Analysis (PCA)**. PCA is a BSS technique to recover a collection of orthogonal vectors  $\mathbf{H}$ , called principal components, and perform a change of basis on the data which usually only uses the principal components that better reconstruct the data (in the  $L_2$ -norm sense).

If the data are in memory, a PCA implementation based on Martinsson *et al.* (2011) and Tipping & Bishop (1999) is included. Otherwise we use the incremental PCA model from Ross *et al.* (2008). In both cases the *scikit-learn* implementation (Pedregosa *et al.*, 2011) is used.

As the principal components are orthogonal, they do not satisfy the non-negativity criterion for image intensity. Therefore, they cannot be interpreted as diffracting crystal elements.

Nevertheless, the eigenvalues of the principal components are very useful to guess the number of true components present in the data. This number can then be used as input parameter for the other BSS methods described below.

(ii) **Non-negative Independent Component Analysis (NICA)**. Since the data are images of an intensity pattern we can impose the non-negativity of the sources as a constraint. NICA finds the components by assuming that they are non-Gaussian and statistically independent from each other [a technique known as independent component analysis, ICA (Hyvärinen *et al.*, 2001)], while restricting  $\mathbf{H}$  to be non-negative. The solution to this problem can be approached with several methods such as described by Yuan & Oja (2004) and Ouedraogo *et al.* (2010). *darfix* implements the method described by Oja & Plumbley (2004). However, NICA does not require the mixing coefficients to be non-negative. Nega-

tive contributions of diffracting elements to the image intensity are non-physical.

(iii) **Non-negative Matrix Factorization (NMF)**. We can constrain both the sources and the mixing elements ( $\mathbf{H}$  and  $\mathbf{W}$ ) to be non-negative. This constraint is called non-negative matrix factorization and is another widely used technique for solving BSS (Lee & Seung, 1999). *darfix* uses the implementation in *scikit-learn* to compute NMF when the data are in memory. Otherwise our own implementation is used based on the multiplicative update rule (Lee & Seung, 2001). This method is applied in chunks to avoid having all the data in memory.

(iv) **NICA–NMF**. The non-uniqueness (non-convexity) property of NMF implies that the solution depends on the initial factor matrices. To solve this problem we implement the idea presented by Kitamura & Ono (2016) which suggests that a good initialization is based on the factorization given by non-negative ICA.

A detailed example of the blind source separation techniques implemented in *darfix* is presented in Appendix A.

### 3.4. GUI

The GUI is programmed in *Qt* (The Qt Company, 2021) and *silx* (Vincent *et al.*, 2021) and it is structured to have a different widget for each step in the data processing workflow. Orange (Demšar & Zupan, 2013) is used to link all these widgets into a single workflow and to pass information between them. Every widget returns a new *Dataset* object which is the input to the next step of the workflow. Fig. 3(a) shows an example of a typical workflow comprising data selection, different pre-processing steps and shift correction. Figs. 3(b), 3(c) and 3(d) show examples of *Grainplot* of the intermediate and final results of the workflow (see below for details).

### 3.5. Open source, documentation and tutorials

*darfix* is open source under MIT license. The software can be installed using pip, see <https://pypi.org/project/darfix/>. User documentation can be found at [https://gitlab.esrf.fr/XRD/darfix/-/blob/main/doc/tutorials/darfix\\_guide.pdf](https://gitlab.esrf.fr/XRD/darfix/-/blob/main/doc/tutorials/darfix_guide.pdf) [https://gitlab.esrf.fr/XRD/darfix/-/blob/main/doc/tutorials/darfix\\_guide.pdf](https://gitlab.esrf.fr/XRD/darfix/-/blob/main/doc/tutorials/darfix_guide.pdf).

## 4. Examples

Fig. 3(a) shows an example of a typical workflow comprising data selection, different pre-processing steps, and shift correction. Figs. 3(b), 3(c) and 3(d) show examples, plotted as a *Grainplot* COM, of the intermediate and final results of the workflow (see below for details). The represented data correspond to a (200) reflection of a grain in an Fe–3%Si sample that was produced as explained by Mavrikakis *et al.* (2019).

### 4.1. Data selection and dimension definition

The first step of the workflow is to select the input data. Next, the dimensions are defined, *i.e.* the motor(s) varying

during the scan and the number of points along each scan axis is determined (see Fig. 3). *darfix* will attempt an automatic dimension definition. This will result in the automatic identification of the scan motors, the angular ranges, the step size and the number of steps of a scan. In the current version, dimension definition is only available for data in EDF format, where motor positions are automatically extracted from the metadata. Data in HDF5 can be analyzed, but at present the metadata are not used. Afterwards, a region of interest can be defined in order to reduce the data volume and speed up processing.

#### 4.2. Noise removal and shift correction

Next, noise removal and shift correction, as detailed above, are applied. The effect of these steps on the data is illustrated in Fig. 3. Fig. 3(b) shows the raw data, followed by Fig. 3(c) where a threshold removal had been executed and Fig. 3(d) where hot pixel removal was also applied.

#### 4.3. Rocking curve imaging

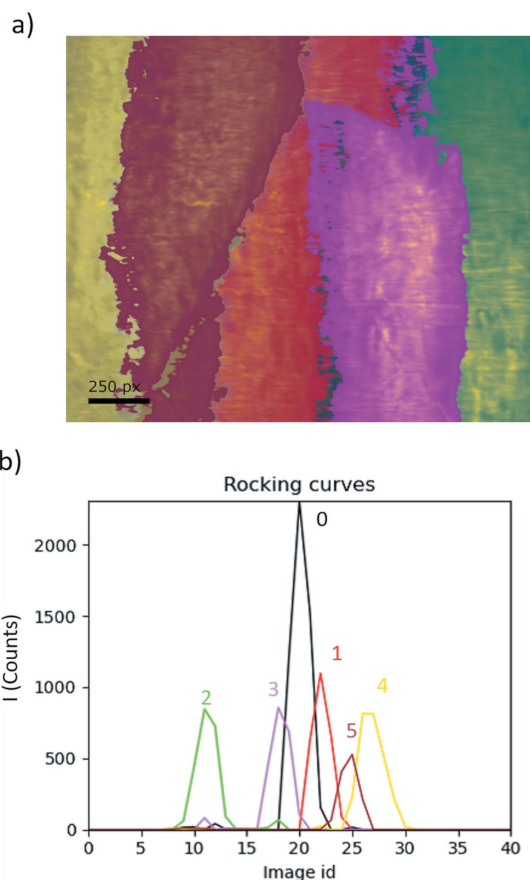
The rocking curve widget performs the fit as described in Sections 2.1.1 and 3.3.1 at each pixel of the data obtained from a rocking scan. Moreover, *darfix* allows to input a 2D scan into the rocking curve widget and performs a 2D fit of the mosaicity data (where the moving motors correspond to angular motions in  $\mu$  and  $\chi$ ).

Fig. 4 shows an example of the maps obtained from the rocking curve analysis of the (200) reflection of an Al sample (Dresselhaus-Marais *et al.*, 2021b), as well as the local rocking curve of a selected pixel. While Fig. 4(b) presents the image obtained at a certain  $\mu$ , the fit maps Figs. 4(c)–4(g) provide information extracted from the fit of each pixel over the whole rocking curve. In this example it can be seen that the ‘boundaries’ between two regions that diffract at different  $\mu$  angle are the hardest to fit, as shown by the residual ( $\chi^2$ ) maps. These boundaries show larger FWHM possibly related to a slightly gradual change in orientation. The peak position map shows a constant gradient, that can be related to a homogeneous deformation of the crystal.

#### 4.4. Mosaicity and strain scan

Figure 5 shows a mosaicity map obtained from the projection of a ferrite grain from an Fe–3%Si alloy (Mavrikakis *et al.*, 2019), in this case the (200) reflection. Fig. 5(a) shows the peak position map of the pitch (rocking angle  $\mu$ ), whereas Fig. 5(b) shows the peak position in roll (angle  $\chi$ ). The two COM angles can be combined into a color vector [Fig. 5(c)]. This type of map represents the local crystallographic orientation around the chosen Bragg reflection. The contours in the color key [Fig. 5(d)] thus represent a local pole figure of the (200) reflection.

Strain scans (varying  $\mu$  and  $2\theta$ ) can be analyzed in the same way. The interpretation, however, is different, as these scans measure the relative axial strain along a given crystallographic plane. In contrast to mosaicity scans where the lattice distortion and orientation are measured, strain scans provide



**Figure 6** Example of blind source separation obtained from a 1D rocking scan of a (200) reflection of an Al single crystal (Dresselhaus-Marais *et al.*, 2021b). (a) Rocking curve image of  $0.2^\circ$  in  $\mu$ . Colored regions correspond to blind source separated components as shown in the rocking curves (b) for each independent component.

information about the variation of the  $d$ -spacing of a given  $hkl$  plane of a crystal.

#### 4.5. Blind source separation

In Fig. 6 the result of the blind source separation is shown for the rocking curve ( $\mu$ ) obtained for a single crystal of Al in a range of  $0.2^\circ$  for a (200) reflection. In Fig. 6(a) the components resulting from the separation are identified by different colors. Fig. 6(b) follows the same color code to show the rocking curves corresponding to each component. The blind source separation operation can be performed on 2D maps; the output can in that case also be plotted as a reciprocal space map for each component. Once the components are obtained they can be exported as HDF5 files.

*darfix* offers the possibility to link components from two different datasets. This allows tracing features from one layer to the next in multilayer scans or the evolution of features under different external conditions.

### 5. Future evolution

Due to its modular structure, it is easy to extend *darfix* with additional functionalities and widgets, for example additional



pre-processing tools such as gradient-based threshold removal for the detection of low-intensity features (Gonzalez *et al.*, 2020).

The functionalities implemented so far are relatively low-level. They mostly implement statistical methods and do not include analysis of the diffraction physics. This is an obvious field for future developments. In particular, we envisage implementing the transformation from angle-space to reciprocal-space, including the transformation of peak shifts in mosaicity and strain scans to strain components (Poulsen *et al.*, 2021).

Furthermore, modules could be developed for the identification of characteristic features in the sample, *e.g.* isolated dislocations (Jakobsen *et al.*, 2019). Development is in progress for the automatic tracking of mobile dislocations in time sequences (Gonzalez *et al.*, 2020). Bayesian inference methods can be used to improve the accuracy of the dislocation core position to  $\sim 5$  nm (Brennan *et al.*, 2022). Due to the long range of strain fields from dislocations, these techniques are directly applicable to classical diffraction topography and rocking curve imaging.

Fine intra-granular defect features of dislocation arrangements can be highlighted by plotting the local orientation gradient of the tilt angles  $\mu$  and  $\chi$  for neighboring voxels within each layer using the following relation:  $\Delta\gamma = [(\Delta\mu)^2 + (\Delta\chi)^2]^{1/2}$  (Mavrikakis *et al.*, 2019; Ahl *et al.*, 2017). Here,  $\Delta\mu$  and  $\Delta\chi$  are the differences between the local sample tilt COM and their grain averages. This is achieved by taking the spatial derivatives of the COM maps, providing information on the dislocation density (Pantleon, 2008; Simons *et al.*, 2019).

Another field for future development is the transformation of a series of 2D scans into a 3D volume model of the sample. At present this is done manually for layer scans (Yildirim *et al.*, 2022), but stacking and registration could be automated. An alternative measurement strategy would be topo-tomo scans (Ludwig *et al.*, 2009), where projections are recorded while the sample is rotated about the scattering vector. Results should be saved in a format compatible with dedicated 3D analysis software such as *Paraview* (Ahrens *et al.*, 2005; Ayachit, 2015). Work for the 3D segmentation of dislocation networks is ongoing (Huang *et al.*, 2022).

## 6. Conclusions

*darfix* is a Python package for the analysis of dark-field X-ray microscopy and diffraction topography data. It provides data processing and visualization tools that can be used either as library components or via a graphical user interface as an add-on to an Orange workflow. *darfix* includes analysis functions specific to common scan types used in DFXM, such as rocking curve imaging, mosaicity and strain scans.

Through blind source separation, different diffracting elements present on the same map or rocking curve can be identified as distinct sources of diffraction, associating each separated unit with their corresponding rocking curve and reciprocal space map.

## APPENDIX A

### Example for blind source separation

In the framework of X-ray science, blind source separation has been used, for example, to identify and quantify the contribution of different elements to fluorescence spectra (Anitha *et al.*, 2013), the components to account for variance in FTIR spectra (Cotte *et al.*, 2016), and the contribution of different shells in EXAFS spectra (Wasserman *et al.*, 1999). For dark-field X-ray microscopy, we assume that different crystalline domains such as grains of ferroelastic domains form independent sources with overlapping diffraction signals. They can be distinguished by their orientation, which will lead to different rocking curves that are shifted in mosaicity maps.

As a simple example with known ground truth, we have simulated the rocking curve images of simple geometric forms as shown in Fig. 7. The resulting stack of images was then analyzed using the blind source separation functionality of *darfix*.

The blind source separation methods available in *darfix* are, as explained in Section 3.3.3: PCA, NICA, NMF and a combined method NICA–NMF. While PCA provides an indication of how many sources to expect, it does not properly identify the sources (geometric shapes) as the PCA components are orthogonal (therefore necessarily include negative intensities), whereas the intensity contributions of the components are semi-definite positive (non-negative). The same property is true for fluorescent intensities. The attribution of negative values can be seen in the example figure (Fig. 7) as the negative values (white) that are found in three of the four components.

Hence the non-negative independent component analysis (NICA) can be implemented, where the constraint for the sources ( $\mathbf{H}$ ) to be non-negative (as defined in Section 3.3.3) is imposed. (N)ICA identifies the sources by imposing the constraint that they are statistically independent (Hyvärinen *et al.*, 2001).

Alternatively, the non-negative matrix factorization (NMF) method, a parts-based representation, imposes the constraint of both the sources and the mixing coefficients (*i.e.* the contributions of the corresponding source to the final image intensities) to be non-negative. But this method yields non-unique solutions dependent on the initialization of the matrices. The combination of NICA–NMF solves the non-uniqueness issue for the NMF approach. The choice of BSS separation method is left to the user according to the goodness of the source identification for a given dataset.

## Acknowledgements

We acknowledge the European Synchrotron Radiation Facility for provision of synchrotron radiation facilities and we would like to thank Thomas Dufrane for assistance in using beamline ID06-HXM. We thank Melanie Gauvin for providing the sample shown in Fig. 6. Leora Dresselhaus-Marais is acknowledged for providing the unpublished data shown in Fig. 4 and critical reading of the manuscript. We thank Thu Nhi Tran Caliste for stimulating discussions on



**Figure 7**  
Example of blind source separation. Each column corresponds to a BSS method: PCA, NICA, NMF, NICA–NMF. The found components are shown for each method.

rocking curve imaging. Sonja R. Ahl and Hugh Simons contributed the original data analysis scripts for mosaicity maps that are now implemented in *darfix*.

**References**

Ahl, S., Simons, H., Detlefs, C., Jensen, D. J. & Poulsen, H. F. (2020). *Acta Mater.* **185**, 142–148.  
 Ahl, S., Simons, H., Zhang, Y., Detlefs, C., Stöhr, F., Jakobsen, A., Juul Jensen, D. & Poulsen, H. (2017). *Scr. Mater.* **139**, 87–91.  
 Ahrens, J., Geveci, B. & Law, C. (2005). *ParaView: An End-User Tool for Large Data Visualization, Visualization Handbook*. Elsevier.  
 Anitha, A., Brasoveanu, A., Duarte, M., Hughes, S., Daubechies, I., Dik, J., Janssens, K. & Alfeld, M. (2013). *Signal Process.* **93**, 592–604.  
 Ayachit, U. (2015). *The ParaView Guide: A Parallel Visualization Application*. Kitware.  
 Bradski, G. (2000). *Dr Dobb's J. Software Tools*, **120**, 122–125.  
 Brennan, M. C., Howard, M., Marzouk, Y. & Dresselhaus-Marais, L. E. (2022). *J. Mater. Sci.*, **57**, 14890–14904.  
 Bucsek, A., Seiner, H., Simons, H., Yildirim, C., Cook, P., Chumlyakov, Y., Detlefs, C. & Stebner, A. P. (2019). *Acta Mater.* **179**, 273–286.  
 Collette, A. (2013). *Python and HDF5*. O'Reilly.  
 Cook, P. K., Simons, H., Jakobsen, A. C., Yildirim, C., Poulsen, H. F. & Detlefs, C. (2018). *Microsc. Microanal.* **24**, 88–89.  
 Cotte, M., Fabris, T., Agostini, G., Motta Meira, D., De Viguierie, L. & Solé, V. A. (2016). *Anal. Chem.* **88**, 6154–6160.  
 Demšar, J. & Zupan, B. (2013). *Informatica*, **37**, 55–60.

De Nolf, W., Payno, H., Svensson, O. & Koumoutsos, G. (2022). *ewoks*, <https://doi.org/10.5281/zenodo.6075053>.  
 Dresselhaus-Marais, L. E. *et al.* (2021b). Private communication.  
 Dresselhaus-Marais, L. E., Winther, G., Howard, M., Gonzalez, A., Breckling, S. R., Yildirim, C., Cook, P. K., Kutsal, M., Simons, H., Detlefs, C., Eggert, J. H. & Poulsen, H. F. (2021a). *Sci. Adv.* **7**, eabe8311.  
 Gonzalez, A., Howard, M., Breckling, S. & Dresselhaus-Marais, L. E. (2020). *arXiv*: 2008.04972.  
 Gorfman, S., Spirito, D., Zhang, G., Detlefs, C. & Zhang, N. (2022). *Acta Cryst. A* **78**, 158–171.  
 Herault, J., Jutten, C. & Ans, B. (1985). *10eme Colloque sur le traitement du signal et des images*, 20–24 May 1985, Nice, France, pp. 1017–1022.  
 Hlushko, K., Keckes, J., Ressel, G., Pörnbacher, J., Ecker, W., Kutsal, M., Cook, P. K., Detlefs, C. & Yildirim, C. (2020). *Scr. Mater.* **187**, 402–406.  
 Huang, P.-H., Coffee, R. & Dresselhaus-Marais, L. (2022). *arXiv*:2211.05247.  
 Hyvärinen, A., Karhunen, J. & Oja, E. (2001). *Independent Component Analysis*. Wiley.  
 Jakobsen, A. C., Simons, H., Ludwig, W., Yildirim, C., Leemreize, H., Porz, L., Detlefs, C. & Poulsen, H. F. (2019). *J. Appl. Cryst.* **52**, 122–132.  
 Kitamura, D. & Ono, N. (2016). *2016 IEEE International Workshop on Acoustic Signal Enhancement (IWAENC)*, 13–16 September 2016, Xi'am, China, pp. 1–5.  
 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.  
 Kutsal, M., Bernard, P., Berruyer, G., Cook, P. K., Hino, R., Jakobsen, A. C., Ludwig, W., Ormstrup, J., Roth, T., Simons, H., Smets, K., Sierra, J. X., Wade, J., Wattecamp, P., Yildirim, C., Poulsen, H. F. & Detlefs, C. (2019). *Mater. Sci. Eng.* **580**, 012007.  
 Lee, D. & Seung, H. (1999). *Nature*, **401**, 788–791.  
 Lee, D. & Seung, H. S. (2001). In *Advances in Neural Information Processing Systems*, edited by T. Leen, T. Dietterich & V. Tresp, Vol. 13. MIT Press.  
 Ludwig, W., King, A., Reischig, P., Herbig, M., Lauridsen, E., Schmidt, S., Proudhon, H., Forest, S., Cloetens, P., Roscoat, S. R., Buffière, J., Marrow, T. & Poulsen, H. (2009). *Mater. Sci. Eng. A*, **524**, 69–76.  
 Martinsson, P.-G., Rokhlin, V. & Tygert, M. (2011). *Appl. Comput. Harmon. Anal.* **30**, 47–68.  
 Mavrikakis, N., Detlefs, C., Cook, P., Kutsal, M., Campos, A., Gauvin, M., Calvillo, P., Saikaly, W., Hubert, R., Poulsen, H., Vaugeois, A., Zapolsky, H., Mangelinck, D., Dumont, M. & Yildirim, C. (2019). *Acta Mater.* **174**, 92–104.  
 Morones, J. R., Elechiguerra, J. L., Camacho, A., Holt, K., Kouri, J. B., Ramírez, J. T. & Yacaman, M. J. (2005). *Nanotechnology*, **16**, 2346–2353.  
 Nellist, P. D. & Pennycook, S. (2000). *Advances in Imaging and Electron Physics*. Springer.  
 Oja, E. & Plumbley, M. (2004). *Neural Comput.* **16**, 1811–1825.

- Ouedraogo, W. S. B., Souloumiac, A. & Jutten, C. (2010). *Latent Variable Analysis and Signal Separation*, edited by V. Vigneron, V. Zarzoso, E. Moreau, R. Gribonval & E. Vincent, pp. 522–529. Berlin, Heidelberg: Springer.
- Pantleon, W. (2008). *Scr. Mater.* **58**, 994–997.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. & Duchesnay, E. (2011). *J. Mach. Learn. Res.* **12**, 2825–2830.
- Poulsen, H. F. (2020). *Curr. Opin. Solid State Mater. Sci.* **24**, 100820.
- Poulsen, H. F., Dresselhaus-Marais, L. E., Carlsen, M. A., Detlefs, C. & Winther, G. (2021). *J. Appl. Cryst.* **54**, 1555–1571.
- Poulsen, H. F., Jakobsen, A. C., Simons, H., Ahl, S. R., Cook, P. K. & Detlefs, C. (2017). *J. Appl. Cryst.* **50**, 1441–1456.
- Ross, D., Lim, J., Lin, R.-S. & Yang, M.-H. (2008). *Int. J. Comput. Vis.* **77**, 125–141.
- Schoeppler, V., Cook, P. K., Detlefs, C., Demichelis, R. & Zlotnikov, I. (2022). *Adv. Mater.* **34**, 2200690.
- Simons, H., Haugen, A. B., Jakobsen, A. C., Schmidt, S., Stöhr, F., Majkut, M., Detlefs, C., Daniels, J. E., Damjanovic, D. & Poulsen, H. F. (2018). *Nat. Mater.* **17**, 814–819.
- Simons, H., Jakobsen, A. C., Ahl, S. R., Poulsen, H. F., Pantleon, W., Chu, Y.-H., Detlefs, C. & Valanoor, N. (2019). *Nano Lett.* **19**, 1445–1450.
- Simons, H., King, A., Ludwig, W., Detlefs, C., Pantleon, W., Schmidt, S., Stöhr, F., Snigireva, I., Snigirev, A. & Poulsen, H. F. (2015). *Nat. Commun.* **6**, 6098.
- The HDF Group, (1997–2022). *Hierarchical Data Format*, version 5, <https://www.hdfgroup.org/HDF5/>.
- The Qt Company, (2021). *Qt*, <https://www.qt.io>.
- Tipping, M. E. & Bishop, C. M. (1999). *Neural Comput.* **11**, 443–482.
- Tran Thi, T. N., Morse, J., Caliste, D., Fernandez, B., Eon, D., Härtwig, J., Barbay, C., Mer-Calfati, C., Tranchant, N., Arnault, J. C., Lafford, T. A. & Baruchel, J. (2017). *J. Appl. Cryst.* **50**, 561–569.
- Vincent, T., Valls, V., Payno, H., Kieffer, J., Solé, V. A., Paleo, P., De Nolf, W., Knobel, P. & Garriga, J. (2021). *silx-kit/silx: 1.0.0*, <https://www.silx.org/>. (Accessed 2021/12/06).
- Walt, S. van der, Schönberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J. D., Yager, N., Gouillart, E., Yu, T. & the scikit-image contributors (2014). *PeerJ*, **2**, e453.
- Wasserman, S. R., Allen, P. G., Shuh, D. K., Bucher, J. J. & Edelstein, N. M. (1999). *J. Synchrotron Rad.* **6**, 284–286.
- Williams, D. & Carter, C. (1996). *Transmission Electron Microscopy*. Plenum Press.
- Yildirim, C., Cook, P., Detlefs, C., Simons, H. & Poulsen, H. F. (2020). *MRS Bull.* **45**, 277–282.
- Yildirim, C., Poulsen, H., Winther, G., Detlefs, C., Huang, P.-H. & Dresselhaus-Marais, L. E. (2022). *Sci. Rep.* **13**, 3834.
- Yuan, Z. & Oja, E. (2004). *International Conference on Independent Component Analysis and Signal Separation ICA 2004: Independent Component Analysis and Blind Signal Separation*, Vol. 3195 of *Lecture Notes in Computer Science*, pp. 1–8. Berlin, Heidelberg: Springer.