Book of Tutorials and Abstracts

European Microbeam Analysis Society

EMAS 2025

18th

EUROPEAN WORKSHOP

on

MODERN DEVELOPMENTS AND APPLICATIONS IN MICROBEAM ANALYSIS

11 to 15 May 2025
at the
TecnoCampus
Mataró (Barcelona), Spain

Organized in collaboration with the
Universitat de Barcelona, Spain

EMAS
European Microbeam Analysis Society eV
www.microbeamanalysis.eu/

This volume is published by:
European Microbeam Analysis Society eV (EMAS)
EMAS Secretariat
c/o Eidgenössische Technische Hochschule, Institut für Geochemie und Petrologie
Clausiusstrasse 25
8092 Zürich
Switzerland

ISBN 978 90 8227 6985
NUR code: 972 – Materials Science

ETSPY: AN OPEN SOURCE SOFTWARE PACKAGE FOR NANOSCALE 3D SPECTROSCOPIC IMAGING IN THE SCANNING TRANSMISSION ELECTRON MICROSCOPE

Andrew A. Herzing

National Institute of Standards and Technology (NIST), Material Measurement Laboratory
100, Bureau Drive, MS 8371, US-20899 Gaithersburg, MD, U.S.A.
e-mail: andrew.herzing@nist.gov

Andrew Herzing is a materials research engineer at the National Institute of Standards and Technology in Gaithersburg, MD. He received his PhD in Materials Science & Engineering from Lehigh University in 2007 where he studied the use of advanced microscopy techniques for the characterisation of gold-based catalysts. He is currently focussed on advancing the capabilities of strain measurement in semiconductors via 4D-STEM methods and the 3D characterisation of complex semiconductor device architectures using electron tomography. Andrew has been active as a reviewer for ISO TC 202 and has recently begun to champion an ISO standard for electron tomography of metal nanoparticle shape and size determination. He is currently serving as the President of the US Microanalysis Society.

1. INTRODUCTION TO STEM AND ELECTRON TOMOGRAPHY

The modern transmission electron microscope (TEM) is a powerful tool for the characterisation of materials at the nano- and atomic scales. One of the operating modes available on most instruments is scanning (S)TEM wherein a very fine probe is scanned over a region of interest in the sample and one or more signals is collected at each point in the digitised raster. STEM mode offers several advantages over the more commonly employed conventional (C)TEM mode. Not least of these is the ability to collect spectroscopic signals simultaneously with multiple imaging signals which makes STEM the operating mode of choice for analytical experiments.

Advances in optics over the past few decades have pushed the spatial resolution of STEM well below 0.1 nm and made atomic-resolution imaging and analysis nearly routine. However, all STEM signals are projected through the thickness of the specimen at each scan point and, therefore, do not readily provide information about the three-dimensional (3D) structural and chemical distributions within. Electron tomography (ET) is a process whereby these 3D aspects of the specimen object are reconstructed from a series of two-dimensional (2D) projections in the TEM or STEM [1]. ET has proven to be a valuable technique in both biological [2-7] and materials sciences [8-13]. Recent advances have even pushed the spatial resolution to the atomic-scale [14-23] and have integrated spectroscopic methods such as energy-dispersive X-ray (EDS) and electron energy-loss (EELS) spectroscopies with the more traditional bright-field or annular dark-field imaging techniques [24-37]. Further improvements in reconstruction resolution and quality have been demonstrated using a multi-modal approach combining imaging and spectral imaging during the tomographic acquisition [38].

Though the technique is quite flexible and has many modalities, the basic concept remains the same throughout. In ET, we seek to recover an image of an object, $f_x(y,z)$, from a series of projections collected over a range of angles, $P_x(\theta,y)$. This can be accomplished in several ways, but often begins with the inverse Radon transform ($\boldsymbol{RT^{-1}}$) such that:

$$\hat{\boldsymbol{f}}_{\boldsymbol{n}}(\boldsymbol{y},\boldsymbol{z}) = \boldsymbol{RT^{-1}}(\boldsymbol{P}_{\boldsymbol{n}}(\boldsymbol{\theta},\boldsymbol{y})) \tag{1}$$

$$\boldsymbol{\theta} \in \boldsymbol{\theta_1}, \boldsymbol{\theta_2}, \boldsymbol{\theta_3}, \ldots, \boldsymbol{\theta_k} \tag{2}$$

$$\mathbf{0} \leq \boldsymbol{n} \leq \boldsymbol{n_x} \tag{3}$$

where: $\boldsymbol{\theta_n}$: tilt angle for each projection image, $\boldsymbol{k}$: number of projection images, $\boldsymbol{n_x}$: number pixels in the horizontal image axis, x, and $\boldsymbol{y}$: vertical image axis.

In the convention we will use for this discussion and for the software package we use for data processing, the tilt axis of the dataset is assumed to be parallel to the horizontal x-axis of the images (see Fig. 1). Note that in the above equation, the reconstructed image is given as $\hat{\boldsymbol{f}}$ rather

than the true object, $\boldsymbol{f}$. This is a result of the finite tilt increment between projections, artefacts arising from data processing errors, the noise content of the projection data, and, in many cases, a hardware/specimen limit on the portion of the full tilt range (0 to π radians) that is accessible.

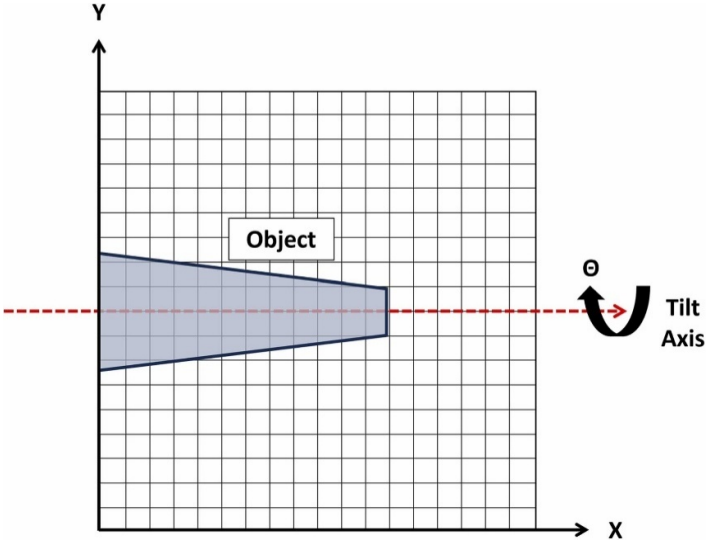

Figure 1. Schematic illustration of the axes configuration for an electron tomography experiment. The object is rotated about the tilt axis which is parallel to the horizontal x-axis of the image.

2. ET WORKFLOW

In practice, the workflow begins with the collection of projection images over a range of specimen tilt orientations. At each step, the stage goniometer is advanced by a given tilt increment, one or more new images are collected, and the data is stored for offline processing. If necessary, the stage and/or beam shift can be adjusted after tilting in order to account for mechanical movement of the specimen during tilting and the image can also be refocused. The entire process of tilting, adjusting shifts, focussing, and collecting data can be fully automated, semi-automated, or fully manual. The degree of user interaction that is necessary is typically set by the spatial resolution requirements of the analysis with higher resolution calling for more intervention.

Next, several pre-processing steps can be performed on an as-needed basis. The full dataset may need to be truncated with some images being discarded due to image quality issues or because the specimen had shifted during acquisition. Often, the dataset will be down sampled in the spatial domain both to decrease computational requirements in the subsequent steps or to reduce noise. Image filters may also be applied during pre-processing in order to remove noise or image artifacts (e.g., cosmic rays, camera defects, etc.). Prior to reconstruction, the projections must be spatially aligned to a common coordinate system and tilt axis. The former involves the

calculation and application of a transformation matrix for each projection in the series, while the latter is accomplished via a global rotation and/or shift of the entire image stack. Failure to either properly register the individual projections to each other or to align the stack to the tilt axis will result in artifacts of varying severity and appearance in the reconstructed volume.

Once pre-processing is complete, the stack is provided as input to a reconstruction algorithm. The goal of the reconstruction process is to convert the input slices along x, also known as the sinograms, ($\boldsymbol{P_x(\theta, y)}$) from the tilt series to two-dimensional images of the original object $\boldsymbol{\hat{f}_n(y, z)}$. These algorithms fit into two broad classes: analytical methods and iterative methods. The most commonly used analytical method is filtered-backprojection (FBP) which applies a weighting filter to the projections prior to calculating the inverse Radon transform of the projection data. The filter is designed to counteract the blurring effect, which results from the finite tilt increment used in acquiring the tilt series data and the nonuniform sampling of the frequency domain this produces. Iterative methods seek to improve upon an initial reconstruction by re-projecting it and comparing to the actual projection data. The two most widely used iterative methods are the simultaneous iterative reconstruction technique (SIRT) [39] and the simultaneous algebraic reconstruction technique (SART) [40]. In both cases, the goal is to minimise the difference between the measured projection data and the reprojection of the current reconstruction. In SIRT, the update step is carried out on the full reconstructed image by forward projecting it and comparing to the full input projection data. By contrast, in SART the update step is performed sequentially by forward projecting the current reconstruction estimate for each projection angle and comparing to the input projection data for just that angle. Many other reconstruction methods have been developed more recently for specific applications. These include (among others): (*i*) the discrete algebraic reconstruction technique (DART) [41] where the intensity levels in the reconstructed image are iteratively assigned to one of several distinct grayscale values provided by the analyst, (*ii*) total-variation minimisation (TVMin) [42] which is an iterative method that includes a regularisation step to promote smoothness within distinct regions of the image, and (*iii*) compressed sensing electron tomography (CSET) [43], which produces high-fidelity reconstructions while remaining robust to undersampling conditions in both the spatial and tilt-axis domains.

Finally, the reconstructed volume must be analysed in some fashion depending on the nature of the features or properties being measured. In some cases, simply visualising the 3D volume is sufficient to reveal the necessary detail and isosurface generation or volumetric rendering can meet these needs. More often, some features such as interfacial area or the volume fraction of a given phase needs to be measured. To do this, the reconstructed data is typically segmented so each voxel of the tomogram is assigned a discrete label, allowing for the quantification of various features.

3. SOFTWARE FOR DATA PROCESSING

Many software packages, both commercial and open-source, are available for handling some or all of these steps; from data acquisition to preprocessing and on to alignment and eventually reconstruction. It is beyond the scope of this paper to discuss all of these, but, limiting our focus to just the non-commercial open source options, the most widely used package for data acquisition is SERIALEM [44, 45] which can be paired with the IMOD package [46, 47] for preprocessing, alignment, and reconstruction. Another widely used open-source tool for tomography data processing and visualisation is TOMVIZ [48, 49], which offers a fully developed UI with intuitive workflow options for all stages of the tomography pipeline. Another well-developed open-source option is TOMOPY [50] which originated for X-ray based tomography work but is readily adapted to ET as well. Finally, a number of plugins for IMAGEJ (too many to list here) are available, each of which handles some portion of the tomography workflow.

In this paper, we describe the ETSPY package [51] which is an ET-focused extension of HYPERSPY [52, 53], a popular open-source, Python-based package for the processing of multidimensional data, including electron microscopy and spectroscopy data. ETSPY offers a compact, command line driven interface for pre-processing, alignment, reconstruction, and basic 2D visualisation of ET data. Additionally, when used through the Jupyter Notebook or Jupyter Lab interfaces54, ETSPY promotes scientific reproducibility through automatic documentation of all processing steps used to achieve a result. These Jupyter documents can also be easily annotated and shared with collaborators or included alongside any resulting publication. As a HYPERSPY extension, ETSPY offers a number of attractive options in terms of the ET workflow. Since the parent package already offers high quality widgets for plotting and visualisation of higher dimensional datasets and has embedded a large number of relevant data processing methods, ETSPY can draw on these existing capabilities and the efforts of the large network of active HYPERSPY users and developers. In this way, ETSPY joins the growing number of HYPERSPY extensions available to analysts all over the world, which as of this publication includes tools for X-ray energy-dispersive and electron energy loss spectroscopies (EXSPY [53]), diffraction and 4D-STEM (PYXEM [55]), EBSD (KIKUCHIPY [56]), cathodoluminescence (LUMISPY [57]), atomic resolution imaging (ATOMAP [58]), holography (HOLOSPY [59]), and particle analysis (PARTICLESPY [60]).

Finally, it is very straightforward to run Jupyter on a headless server which enables users to leverage remote high-performance computing resources which have large amounts of memory for handling larger datasets as well as GPUs and high CPU core counts for accelerated tomographic reconstruction. By accessing the Jupyter interface through a web browser on a local laptop or desktop, data processing and visualisation can be carried out in a convenient, interactive fashion without being constrained by the limits of the hardware of the local machine.

4. *ETSPY OVERVIEW*

ETSPY creates two new Python classes both of which are derived from the HYPERSPY `Signal2D` parent class and inherit all the latter's functionality. The first of these new classes is the `TomoStack`, which contains the tilt series data, metadata, and all the methods used for data manipulation, alignment, and reconstruction. As in the `Signal2D` parent class, the `TomoStack` has navigation axes and signal axes. For an ET image tilt series, the signal axes will be the x and y image axes, while the navigation axis will be the θ tilt angle dimension. The data can be visualised readily using the HYPERSPY plot functionality and, when using the Jupyter widget backend for MATPLOTLIB61, the stack can be interactively viewed in the Jupyter interface as a function of tilt angle. In addition to the `TomoStack`, there is a second class called a `RecStack`, which is used to handle the reconstructed data.

ETSPY contains a load function for reading data into memory. HYPERSPY [62] and the associated ROSETTASCIIO package [63] already offer robust data reading capabilities for a wide array of ET relevant file formats. ETSPY relies on this functionality to access image data and the required metadata for several formats such as Gatan Digital Micrograph (DM3 or DM4), HDF5, and MRC. When loaded, HYPERSPY reader functions handle the calibration data and ETSPY collects the projection angle data for inclusion in the resulting `TomoStack`. Alternatively, if a tilt series already resides in memory as a NumPy array or a HYPERSPY SIGNAL2D it can be directly converted to a `TomoStack`.

Any `TomoStack` or `RecStack` can be saved at any point during data processing and analysis. This is done by using the save method and is best accomplished using the HYPERSPY hspy format, which is a custom version of the HDF5 standard format. When using the `hspy` format, the data and all of the metadata is stored and easily accessed in the future using the `etspy.load()` method.

Both the `TomoStack` and `RecStack` classes inherit the plotting functionality of the HYPERSPY `Signal2D` class. When used interactively with ipywidgets a slice navigator can be used to navigate through the stack in the tilt dimension and update the display in real time. By employing the `swap_axes()` method, navigation can also be done over the x-axis which will show the sinogram view.

5. *CONVENTIONAL STEM TOMOGRAPHY*

To demonstrate the functionality of ETSPY for conventional image based tomography data processing, a dataset collected from a sample of NIST SRM-2135c [64] will be used. This SRM consists of a silicon substrate with alternating layers of nickel and chromium deposited on the free surface. A pillar-shaped sample was fabricated from this SRM using focussed ion beam milling. The tomographic tilt series consists of 91 STEM high-angle annular dark field

(HAADF) images collected over a 180° tilt range with a constant tilt increment of 2°. The primary beam energy was 300 keV and the probe current was approximately 50 pA. Each image was 1,024 pixels square and was collected using a per pixel dwell time of 32 µs and a pixel size of 1 nm. The full stack was collected in an automated manner using SerialEM.

The first task confronted by a microanalyst who wants to reconstruct electron tomography data is image registration. ETSPY provides multiple options for calculating and applying translational shifts for aligning a `TomoStack`, all which are accessed using the `register_stack()` method. The available options are based on phase correlation (PC) [65], the `StackReg` [66] method, centre of mass tracking (COM) [67], and an approach that combines centre of mass tracking with the so-called common-line method (COM-CL) [14]. All of these methods have advantages and disadvantages. PC is straightforward and fast, although it can be challenged by noisy data, by samples which exhibit regularly repeating features, or when the differences in image features between two images is large (e.g., when a large tilt increment is used). `StackReg` will often outperform PC in these cases but is computationally more expensive. Finally, both COM and COM-CL incorporate the geometry of the tilt series acquisition and are very effective for particulate or pillar shaped samples since little or no additional material enters the field of view during tilting in these cases. However, they will often fail when slab type specimen geometries are used or when significant amounts of extraneous material enters and leaves the field of view over the course of the series. Further detail about all these methods can be found in the provided citations.

Regardless of the method chosen, translation shifts are calculated between each set of images in the stack. The shifts are then composed such that they are made relative to the previous shift in the stack starting from a user defined projection number. The results of this process for the SRM-2135c pillar sample is shown in Fig. 2.

Finally, the shifts can be calculated using any of these methods and then applied to another stack using the `align_other()` method. This approach is useful when multiple image signals are collected during a single tilt series acquisition and when one of the signals has superior characteristics for calculating the alignment shifts compared to another. For example, shifts can be calculated on a high signal-to-noise ratio medium-angle annular dark field (MAADF) image stack and applied to a noisier high-angle annular dark field (HAADF) stack. Since both signals are acquired simultaneously, the shifts for each will be the same. This will also be very useful when we turn to hyperspectral tomography data, since the shifts can be calculated using an image signal then used to register the spectral images extracted from the simultaneously acquired hyperspectral dataset.

Once the stack has been translationally registered, it is then important to ensure that the tilt axis of the dataset is centred and made as close to horizontal as possible prior to reconstruction. Failure to do so will result in severe artefacts in the reconstructed sinograms and make interpretation and quantification of the data difficult or impossible. To perform the tilt axis

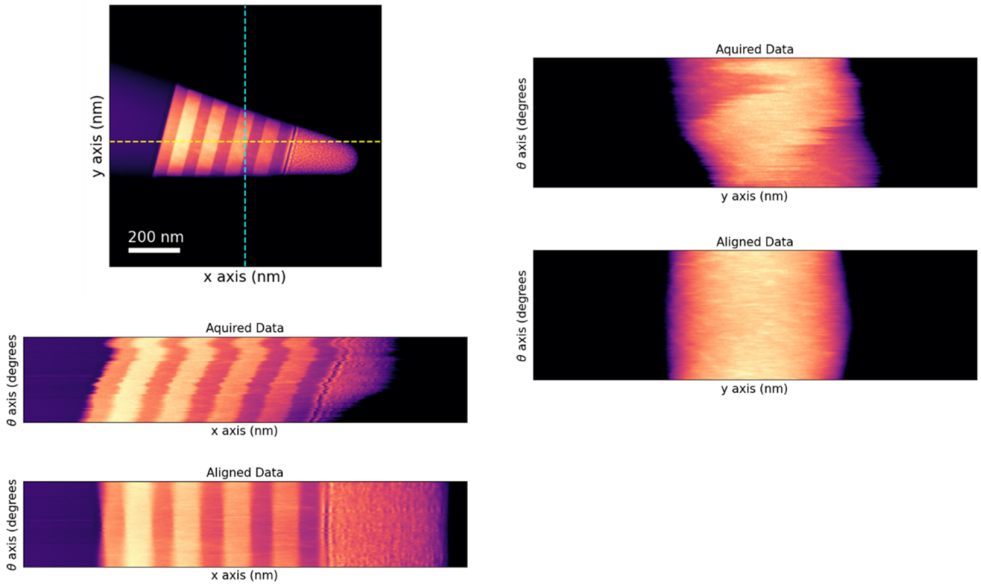

Figure 2. Central projections extracted from the NIST SRM-2135c tilt series. The *y*-axis projections (sinograms) and the *x*-axis projections were extracted from the tilt series data along the positions denoted by the cyan and yellow dashed lines. Projections are shown as acquired and after image registration and fine alignment.

alignment, two options are available: A method which uses centre of mass tracking [68] and another using the intensity present in the maximum projection image of the dataset. The latter is particularly effective when particles are present in the sample which trace linear paths in the projected maximum image. In both cases, the goal is to calculate a global shift and rotation of the stack which places the tilt axis along the horizontal axis of the stack. The effects of the stack registration and tilt axis alignment process on the reconstruction quality for the pillar-shaped SRM-2135c sample are shown in Fig. 3. Prior to tilt axis correction, both the as acquired data and the spatially registered data suffer from severe degradation in the form of the C-shaped artifact surrounding the sample region in each. Only after tilt axis alignment is the expected circular shape of the conical cross-section recovered.

Once the alignment process is complete, ETSPY offers five algorithms to perform the reconstruction. These are simple back projection (BP), filtered back projection (FBP, default), SIRT, SART, and discrete algebraic reconstruction technique (DART). Each of these algorithms relies upon the implementations provided in the ASTRA toolbox [69] to perform the reconstruction.

Reconstruction is initiated by calling the `reconstruct()` method of the `TomoStack`. The ASTRA toolbox offers CPU-based and GPU-accelerated options for the BP, FBP, SIRT, and SART methods. ETSPY will attempt to determine whether a CUDA-compatible GPU is available and use it for faster reconstruction when possible. Alternatively, the reconstruction

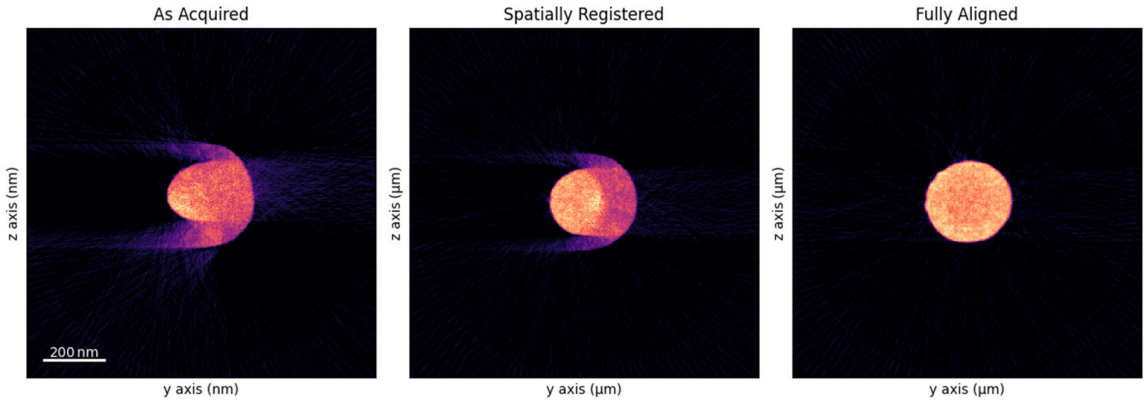

Figure 3. Reconstructed slices from the NIST SRM-2135c tilt series at various points in the alignment workflow. Reconstructions using the as acquired data, the spatially registered data, and the fully aligned data are shown.

will be carried out using parallel computation on the CPU using the Python multiprocessing package. The reconstruction returns a `RecStack` object, which contains the 3D reconstructed image data. Results generated from the NIST SRM-2135c sample are shown in Fig. 4.

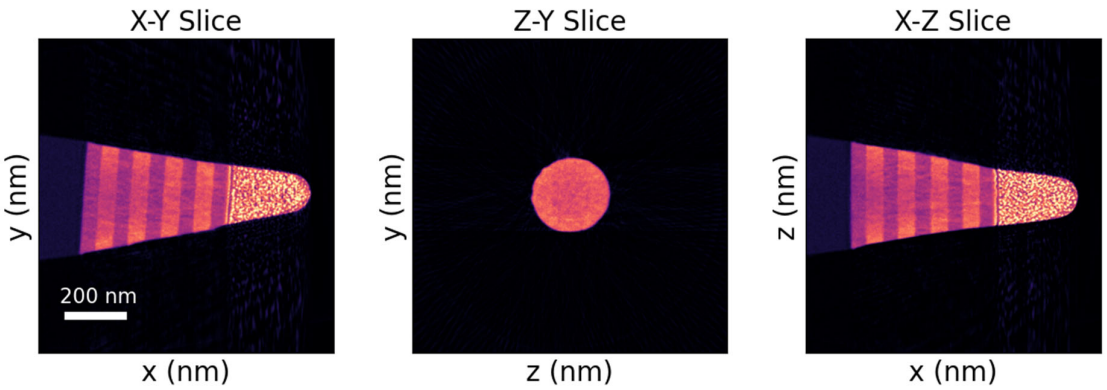

Figure 4. Reconstruction of STEM-HAADF tomography data of NIST SRM-2135c. Depicted are the central slices extracted from the reconstruction in the '*X*-*Y*', '*Z*-*Y*', and '*X*-*Z*' orientations.

6. HYPERSPECTRAL STEM TOMOGRAPHY

To demonstrate the use of ETSPY for processing hyperspectral tomography, data was collected from another needle-shaped sample of the NIST SRM-2135c depth-profiling standard. EDS spectrum images (SIs) were collected over a 180° tilt range using a tilt increment of 5°. The data were collected using an EDAX Octane-T silicon drift detector and the acquisition was controlled by the EDAX TEAM software. The SIs were 162 × 128 pixels in size with

3 nm square pixels. The probe current was approximately 0.5 nA and the per pixel dwell time was 200 ms. At each new tilt, the sample was manually aligned to an image at the previous tilt using a combination of beam and stage shifting. The sum spectrum for the entire spectral tomography dataset is shown along with the sum image calculated from a single projection angle in Fig. 5.

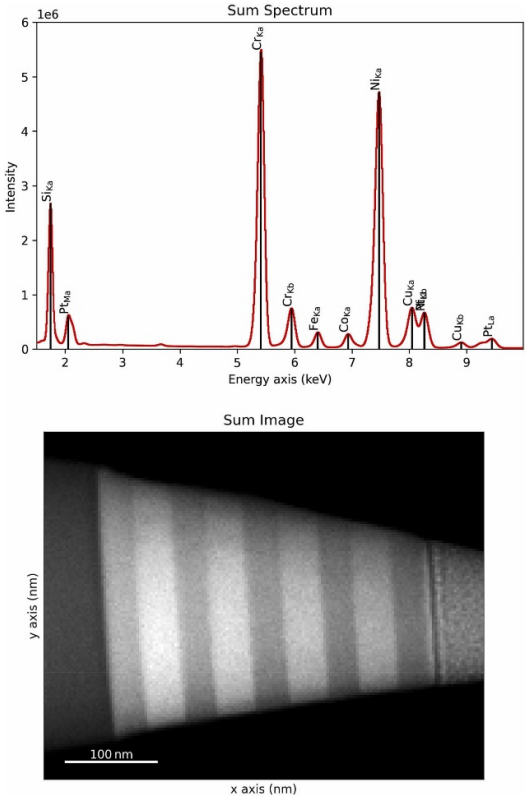

Figure 5. Sum spectrum of the full spectral tomography dataset of the SRM 2135c sample and sum image extracted from a single spectrum image projection.

To analyse this dataset using ETSPY, the TEAM SPD files were converted to HDF5 format using HyperSpy. The resulting dataset consists of 37 HYPERSPY `Signal2D` instances each of which is 4,000 channels deep and 162 × 128 pixels in size. HYPERSPY was used to extract the required 2D elemental maps at each tilt and converting each of the resulting HYPERSPY `Signal2D` instances to a `TomoStack` with a shape of (37 | 162 × 128). Using HYPERSPY, this can be done either via straightforward background subtraction methods, curve fitting, or via machine learning algorithms. In the present case, we have employed NMF decomposition and the resulting component images for one projection of the full hyperspectral tomography dataset are shown in Fig. 6. In this case, the first four components show the spatial distribution of the nickel, silicon, chromium, and platinum signals, respectively. Components five and six (not shown), were related to the absorption of the nickel and chromium L-lines.

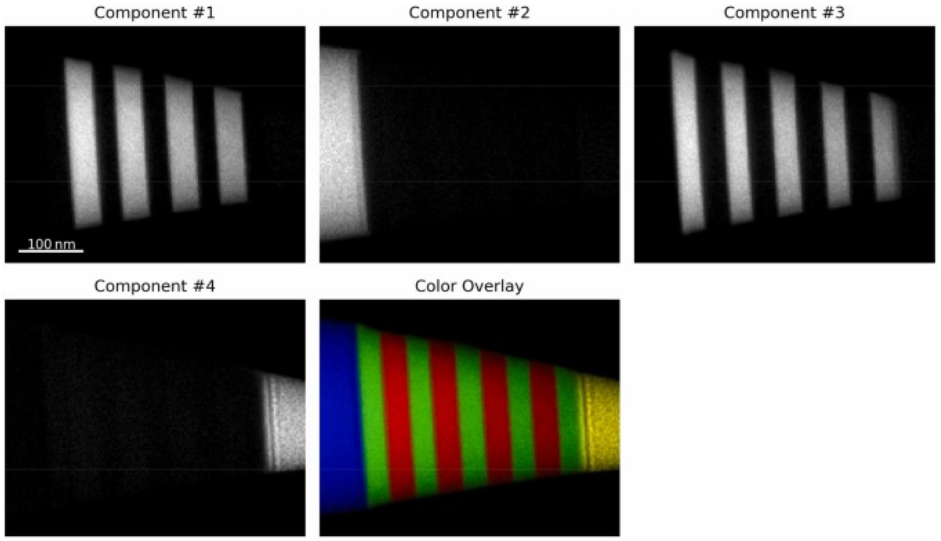

Figure 6. Results of NMF decomposition of spectral tomography dataset. Components 1 through 4 are shown along with a colour overlay emphasizing their spatial extent.

Once the `TomoStack`'s have been created, it is then just a matter of following the usual tomographic workflow of stack registration, tilt-axis alignment and reconstruction. Since the elemental maps are often not well suited for calculating alignment transformations, we can either use a tilt series of an image signal collected simultaneously (e.g., HAADF, etc.) or by the spectral sum images for alignment followed by using the `align_other()` method of the `TomoStack` to apply the calculated alignment to each spectral tilt series. For the present case we have used the spectral sum images to calculate the alignment. Finally, the individual NMF image series are reconstructed independently and slices from each are displayed in Fig. 7 (example code for the full process is included in the Appendix).

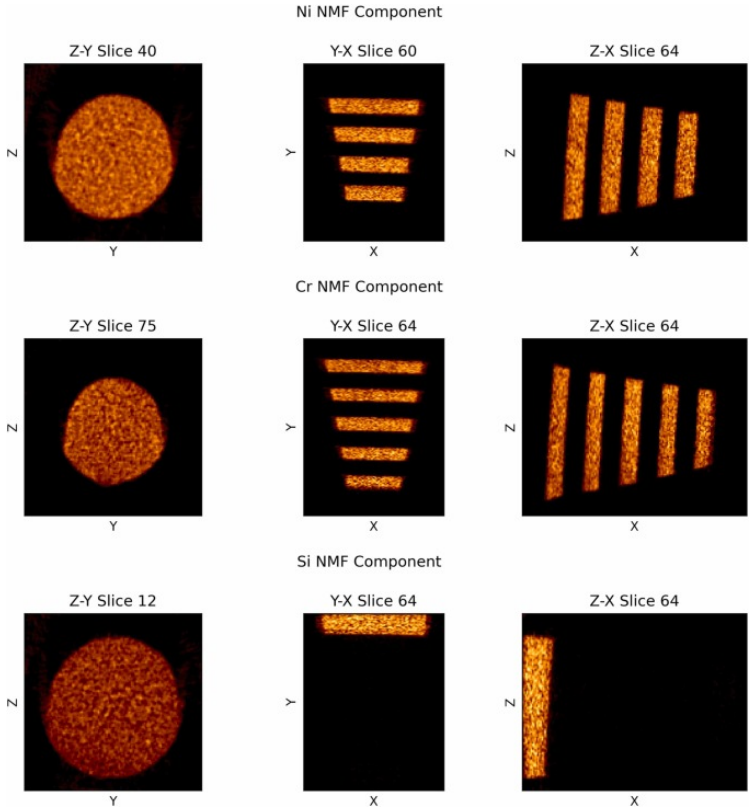

Figure 7. Slices extracted from reconstructions of the NMF components associated with nickel, chromium, and silicon.

At this point, the reconstructions can be quantified using packages such as NUMPY, SCIPY, SCIKIT-IMAGE, etc., all of which can be accessed directly from the same Jupyter Notebook used for ETSPY processing. For example, it is very straightforward to binarize the reconstructions of the nickel and chromium layers and calculate volumetric ratios of each:

```
### Code block showing a basic approach to quantification of
### tomographic reconstruction

pixel_size =Ni_rec.axes_manager[0].scale

# Copy Ni reconstruction and binarize it
Ni_binary =Ni_rec.deepcopy()
Ni_binary[Ni_binary>0] = 1

# Copy Cr reconstruction and binarize it
Cr_binary =Cr_rec.deepcopy()
Cr_binary[Cr_binary>0] = 1

# Calculate the calibrated volume of each component and the
# volumetric ratio of Cr to Ni

Ni_volume = pixel_size**3 * Ni_binary.sum()
Cr_volume = pixel_size**3 * Cr_binary.sum()
volume_ratio = Ni_volume/Cr_volume
```

This is just one simple example to illustrate the way in which tomographic reconstructions generated using ETSPY can be directly interrogated within the Jupyter environment. Finally, 3D visualisation can be carried out in one of two ways. First, the reconstructions can be saved to disk and then read into dedicated visualisation software. Alternatively, for those who wish to remain in the Python environment used for the rest of the data processing, packages such as IPYVOLUME [70] can be used for some basic visualisation. An example of using IPYVOLUME is shown in Fig. 8.

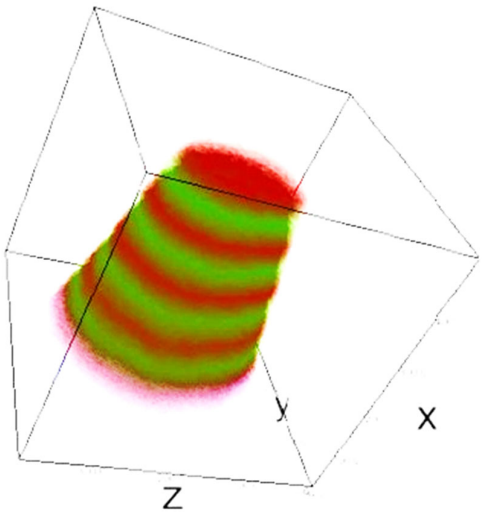

Figure 8. Volumetric rendering using IPYVOLUME of the Cr-Kα, Ni-Kα, and Si-Kα tomographic reconstructions (red, green, and blue, respectively).

7. *REFERENCES*

[1] Frank J 2006 *Electron tomography*. [New York, NY: Springer]
[2]) McEwen B F and Marko M 2001 *J. Histochem. Cytochem.* **49** 553-563
[3] Lučić V, Förster F and Baumeister W 2005 *Ann. Rev. Biochem.* **74** 833-865
[4] Downing K H, Sui H and Auer M 2007 *Anal. Chem.* **79** 7949-7957
[5] Gan L and Jensen G J 2012 *Quart. Rev. Biophys.* **45** 27-56
[6] Turk M and Baumeister W 2020 *FEBS Lett.* **594** 3243-3261
[7] Young L N and Villa E 2023 *Ann. Rev. Biophys.* **52** 573-595
[8] Midgley P A and Weyland M 2003 *Ultramicroscopy* **96** 413-431
[9] Weyland M and Midgley P A 2004 *Mater. Today* **7**(12) 32-40
[10] Ziese U, de Jong K P and de Koster A J 2003 *Appl. Catal. A: Gen.* **260** 71-74
[11] Kübel C, Voigt A, Schoenmakers R, Otten M, Su D, Lee T-C, Carlsson A and Bradley J 2005 *Microsc. Microanal.* **11** 378-400
[12] Midgley P A and Dunin-Borkowski R E 2009 *Nature Mater.* **8** 271-280

[13] Grenier A, Duguay S, Barnes J P, Serra R, Haberfehlner G, Cooper D, Bertin F, Barraud S, Audoit G, Arnoldi L, Cadel E, Chabli A and Vurpillot F 2014 *Ultramicroscopy* **136** 185-192
[14] Scott M C, Chen C-C, Mecklenburg M, Zhu C, Xu R, Ercius P, Dahmen U, Regan B C and Miao J 2012 *Nature* **483** 444-447
[15] Chen C-C, Zhu C, White E R, Chiu C-Y, Scott M C, Regan B C, Marks L D, Huang Y and Miao J 2013 *Nature* 496 74-77
[16] Xu R, Chen C-C, Wu L, Scott M C, Theis W, Ophus C, Bartels M, Yang Y, Ramezani-Dakhel H, Sawaya M R, Heinz H, Marks L D, Ercius P and Miao J 2015 *Nature Mater.* **14** 1099-1103
[17] Miao J, Ercius P and Billinge S J L 2016 *Science* **353** aaf2157
[18] Yang Y, Chen C-C, Scott M C, Ophus C, Xu R, Pryor A, Wu L, Sun F, Theis W, Zhou J, Eisenbach M, Kent P R C, Sabirianov R F, Zeng H, Ercius P and Miao J 2017 *Nature* **542** 75-79
[19] Zhang Q, Kusada K, Wu D, Yamamoto T, Toriyama T, Matsumura S, Kawaguchi S, Kubota Y and Kitagawa H 2018 *Nature Commun.* **9** 510
[20] Wang C, Ding G, Liu Y and Xin H L 2020 *Adv. Intell. Systems* **2** 2000152
[21] Wang, C, Duan H, Chen C, Wu P, Qi D, Ye H, Jin H-J, Xin H L and Du K 2020 *Matter* **3** 1999-2011
[22] Yang Y, Zhou J, Zhu F, Yuan Y, Chang D J, Kim D S, Pham M, Rana A, Tian X, Yao Y, Osher S J, Schmid A K, Hu L, Ercius P and Miao J *Nature* **592** 60-64
[23] Pelz P M, Groschner C, Bruefach A, Satariano A, Ophus C and Scott M C 2022 *ACS Nano* **16** 588-596
[24] Möbus G, Doole R C and Inkson B J 2003 *Ultramicroscopy* **96** 433-451
[25] Gass M H, Koziol K K, Windle A H and Midgley P A 2006 *Nano Lett.* **6** 376-379
[26] Yaguchi T, Konno M, Kamino T and Watanabe M 2008 *Ultramicroscopy* **108** 1603-1615
[27] Genc A, Kovarik L, Gu M, Cheng H, Plachinda P, Pullan L, Freitag B and Wang C 2013 *Ultramicroscopy* **131** 24-32
[28] Goris B, De Backer A, Van Aert S, Gómez-Graña S, Liz-Marzán L M, Van Tendeloo G and Bals S 2013 *Nano Lett.* **13** 4236-4241
[29] Bals S, Goris B, Liz-Marzán L M and Van Tendeloo G 2014 *Angewandte Chem. Int. Ed.* **53** 10600-10610
[30] Haberfehlner G, Orthacker A, Albu M, Li J and Kothleitner G 2014 *Nanoscale* **6** 14563-14569
[31] Pfannmöller M, Heidari H, Nanson L, Lozman O, Chrapa M, Offermans T, Nisato G and Bals S 2015 *Nano Lett.* **15** 6634-6642
[32] Goris B, Meledina M, Turner S, Zhong Z, Batenburg K J and Bals S 2016 *Ultramicroscopy* **171** 55-62
[33] Zanaga D, Altantzis T, Polavarapu L, Liz-Marzán L M, Freitag B and Bals S 2016 *Particle & Particle Syst. Characteriz.* **33** 396-403
[34] Zhong Z, Goris B, Schoenmakers R, Bals S and Batenburg J K 2017 *Ultramicroscopy* **174** 35-45

[35] Bender H, Richard O, Kundu P, Favia P, Zhong Z, Palenstijn W J, Batenburg K J, Wirix M, Kohr H and Schoenmakers R 2019 *Semiconductor Sci. Technol.* **34** 114002
[36] Baumann F H, Popielarski B, Lu Y and Mitchell T 2020 *IEEE Trans. Semiconductor Manuf.* **33** 346-351
[37] Han Y, Jang J, Cha E, Lee J, Chung H, Jeong M, Kim T-G, Chae B G, Kim H G, Jun S, Hwang S, Lee E and Ye J C 2021 *Nature Mach. Intell.* **3** 267-274
[38] Schwartz J, Di Z W, Jiang Y, Manassa J, Pietryga J, Qian Y, Cho M G, Rowell J L, Zheng H, Robinson R D, Gu J, Kirilin A, Rozeveld S, Ercius P, Fessler J A, Xu T, Scott M and Hovden R 2024 *Nature Commun.* **15** 3555
[39] Gilbert P 1972 *J. Theor. Biology* **36** 105-117
[40] Andersen A H and Kak A C 1984 *Ultrasonic Imaging* **6** 81-94
[41] Batenburg K J and Sijbers J 2011 *IEEE Trans. Image Process.* **20** 2542-2553
[42] Rudin L I, Osher S and Fatemi E 1992 *Physica D: Nonlin. Phenom.* **60** 259-268
[43] Saghi Z, Holland D J, Leary R, Falqui A, Bertoni G, Sederman A J, Gladden L F and Midgley P A 2011 *Nano Lett.* **11** 4666-4673
[44] Mastronarde D N 2005 *J. Struct. Biology* **152** 36-51
[45] Mastronarde D N *SerialEM: A program for automated electron microscope tomography*. https://bio3d.colorado.edu/SerialEM/index.html
[46] Mastronarde D N *IMOD: Image processing and 3D reconstruction*. https://bio3d.colorado.edu/imod/
[47] Kremer J R, Mastronarde D N and McIntosh J R 1996 *J. Struct. Biology* **116** 71-76
[48] Schwartz J, Harris C, Pietryga J, Zheng H, Kumar P, Visheratina A, Kotov N A, Major B, Avery P, Ercius P, Ayachit U, Geveci B, Muller D A, Genova A, Jiang Y, Hanwell M and Hovden R 2022 *Nature Commun.* **13** 4458
[49] Anon. *Tomviz: Open-source 3D visualization for electron tomography*. https://tomviz.org/
[50] Gürsoy D, De Carlo F, Xiao X and Jacobsen C 2014 *J. Synchrotron Rad.* **21** 1188-1193
[51] Herzing A 2024 *Usnistgov/Etspy: ETSpy Package*. https://github.com/usnistgov/etspy
[52] de la Pena F, Ostasevicius T, Tonaas Fauske V, Burdet P, Jokubauskas P, Nord M, Sarahan M, Prestat E, Johnstone D N, Taillon J, *et al.* 2017 *Microsc. Microanal.* **23** (Suppl. 1) 214-215
[53] de la Peña F, Prestat E, Burdet P, Lähnemann J, MacArthur K E, Fauske V T, Sarahan M, Francis C, Johnstone D N, Ostasevicius T, Migunov V, Furnival T, Nord M, Mazzucco S, Eljarrat A, Caron J, Aarholt T, Poon T, Jokubauskas P, actions-user; Winkler F, Taillon J, Slater T, pquinn-dls, Guzzinati G, Myers J C, Tappy N and Garmannslund A 2024 *Hyperspy/Exspy: V0.2*. https://doi.org/10.5281/zenodo.10953535
[54] Beg M,; Taka J, Kluyver T, Konovalov A, Ragan-Kelley M, Thiéry N M and Fangohr H 2021 *Computing Sci. Eng.* **23** 36-46
[55] Johnstone D, Crout P, Francis C, Nord M, Laulainen J, Høgås S, Opheim E, Prestat E, Martineau B, Bergh T, Cautaerts N, Ånes H W, Smeets S, Femoen V J, Ross A, Broussard, J, Huang S, Collins S, Furnival T, Jannis D, Hjorth I, Jacobsen E, Danaie M, Herzing A, Poon T, Dagenborg S, Bjørge R, Iqbal A, Morzy J, Doherty T, Ostasevicius T, Thorsen T I, von Lany M, Tovey R and Vacek P 2024 *Pyxem/Pyxem: V0.19.1*. https://doi.org/10.5281/zenodo.11585254

[56] Ånes H W, Lervik L A H, Natlandsmyr O, Bergh T, Prestat E, Bugten A V, Østvold E M, Xu Z, Francis C and Nord M 2024 *Pyxem/Kikuchipy: Kikuchipy 0.10.0*. https://doi.org/10.5281/zenodo.11432173
[57] Lähnemann J, Orri J F, Prestat E, Ånes H W, Johnstone D N, Migrator L and Tappy N 2023 *LumiSpy/Lumispy: V0.2.2*. https://doi.org/10.5281/zenodo.7747350
[58] Nord M, Vullum P E, MacLaren I, Tybell T and Holmestad R 2017 *Adv. Structural Chem. Imaging* **3** 9
[59] Prestat E, de la Peña F, Migunov V, Lähnemann J, Caron J, Winkler F, Francis C, Slater T, MacArthur K E, Ostasevicius T, Taillon J, pquinn-dls, Furnival T, Johnstone D N, Aarholt T, Poon T, Nord M and Jokubauskas P 2024 *Hyperspy/Holospy: V0.3*. https://doi.org/10.5281/zenodo.11099344
[60] Slater T, CameronGBell and Mohsen *ePSIC-DLS/Particlespy: V0.6.0*. https://doi.org/10.5281/zenodo.5094360
[61] Hunter J D 2007 *Computing Sci. Eng.* **9** 90-95
[62] de la Peña F, Prestat E, Fauske V T, Lähnemann J, Burdet P, Jokubauskas P, Furnival T, Francis C, Nord M, Ostasevicius T, MacArthur K E, Johnstone D N, Sarahan M, Taillon J, Aarholt T, pquinn-dls, Migunov V, Eljarrat A, Caron J, Nemoto T, Poon T, Mazzucco S, actions-user, Tappy N, Cautaerts N, Somnath S, Slater T, Walls M, pietsjoh and Ramsden H 2024 *Hyperspy*. https://doi.org/10.5281/zenodo.11148112
[63] Prestat E, de la Peña F, Lähnemann J, Jokubauskas P, Fauske V T, pietsjoh, Ostasevicius T, Nemoto T, Francis C, Johnstone D N, Furnival T, Cautaerts N, Somnath S, pquinn-dls, Caron J, MacArthur K E, Nord M, Burdet P, Tappy N, Aarholt T, Poon T, Taillon J, Slater T, Migunov V, DENSmerijn, Sarahan, M and Ånes H W 2024 *Hyperspy/Rosettasciio: V0.5*. https://doi.org/10.5281/zenodo.12208017
[64] NIST 1999 *NIST SRM 2135c Certificate of Analysis Ni/Cr Thin Film Depth Profile Standard*. [Gaitherburg, MD: National Institute of Standards and Technology]
[65] Guizar-Sicairos M, Thurman S T and Fienup J R 2008 *Optics Lett.* **33** 156-158
[66] Thevenaz P, Ruttimann U and Unser M A 1998 *IEEE Trans. Image Process.* **7** 27-41
[67] Sanders T, Prange M, Akatay C and Binev P 2015 *Adv. Structural Chem. Imaging* **1** 4
[68] Wolf D 2012 *Accurate tilt series alignment for single axis tomography by sinogram analysis*. in: Proc. 15th Eur. Microscopy Congress.
[69] van Aarle W, Palenstijn W J, Beenhouwer J D, Altantzis T, Bals S, Batenburg K J and Sijbers J 2015 *Ultramicroscopy* **157** 35-47
[70] Breddels M *Maartenbreddels/Ipyvolume: 3d plotting for Python in the Jupyter notebook based on IPython widgets using WebGL*. https://github.com/maartenbreddels/ipyvolume

```
### Code block showing workflow for conventional tomographic
### reconstruction

# Import ETSpy
import etspy.api as et

# Define path to tilt series file
datapath = '/path/to/tilt/series/filename.mrc'

# Load data
haadf = et.load(datapath)

# Rebin spatially by a factor of 2
haadf = haadf.rebin(scale=[1,2,2])

# Register the images in the stack using pystackreg
reg = haadf.stack_register("StackReg")

# Determine the tilt axis rotation and rotate stack so that
# tilt axis is horizontal
ali = reg.tilt_align("CoM", slices=[160,256,380])

# Manually inspect reconstruction quailty and shift tilt axis
# along the y axis (i.e., perpendicular to the tilt axis)
_ = ali.test_align(method='FBP', iterations=100)
ali = ali.trans_stack(yshift=6)

# Reconstruct data
rec = ali.reconstruct('FBP')
```


```
### Code block showing workflow for conventional tomographic
### reconstruction

# Import ETSpy
import etspy.api as et

# Define path to tilt series file
datapath = '/path/to/tilt/series/filename.mrc'

# Load data
haadf = et.load(datapath)

# Rebin spatially by a factor of 2
haadf = haadf.rebin(scale=[1,2,2])

# Register the images in the stack using pystackreg
reg = haadf.stack_register("StackReg")

# Determine the tilt axis rotation and rotate stack so that
# tilt axis is horizontal
ali = reg.tilt_align("CoM", slices=[160,256,380])

# Manually inspect reconstruction quailty and shift tilt axis
# along the y axis (i.e., perpendicular to the tilt axis)
_ = ali.test_align(method='FBP', iterations=100)
ali = ali.trans_stack(yshift=6)

# Reconstruct data
rec = ali.reconstruct('FBP')
```

```
### Code block showing workflow hyperspectral tomographic
### reconstruction using ETSpy

## Calculate alignment on the spectral sum image series

# Calculate sum image. This results in a HyperSpy
# Signal2D class:
edx_sum = edx.sum(3).as_signal2D((0,1))

# Convert to a Signal2D to TomoStack
edx_sum = etspy.TomoStack(si_sum, tilts)
edx_reg = edx_sum.stack_register("StackReg")
edx_ali = edx_reg.tilt_align("CoM")

# Apply alignments to the phase maps
Ni_NMF = edx_ali.align_other(Ni_NMF)
Si_NMF = edx_ali.align_other(Si_NMF)
Cr_NMF = edx_ali.align_other(Cr_NMF)
Pt_NMF = edx_ali.align_other(Pt_NMF)

# Reconstruct the datasets
Ni_rec = Ni_NMF.reconstruct()
Si_rec = Si_NMF.reconstruct()
Cr_rec = Cr_NMF.reconstruct()
Pt_rec = Pt_NMF.reconstruct()
```