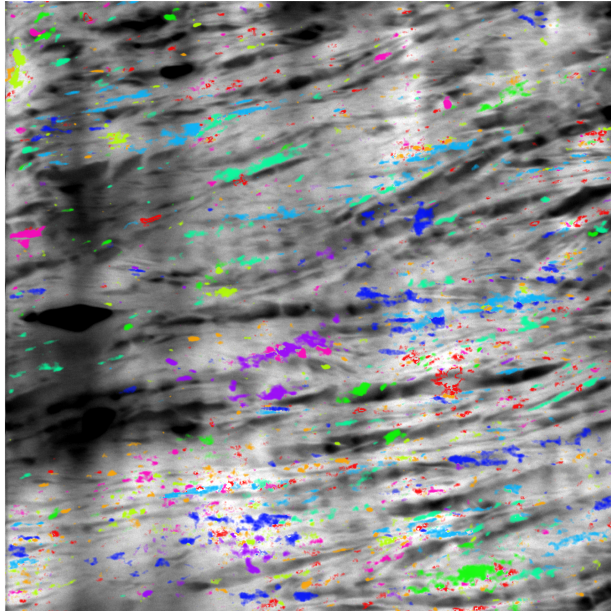




CHALMERS
UNIVERSITY OF TECHNOLOGY



Python-Based Analysis of Multidimensional Electron Diffraction Data in Human Bone

Master Thesis Project

HAYDAR BURAK ERKIZAN

DEPARTMENT OF PHYSICS AND ASTRONOMY, DIVISION OF MICROSTRUCTURE PHYSICS

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2026
www.chalmers.se

MASTER'S THESIS 2026

Python-Based Analysis of Multidimensional Electron Diffraction
Data in Human Bone

HAYDAR BURAK ERKIZAN



CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Physics and Astronomy
Division of Microstructure Physics
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2026

Python-Based Analysis of Multidimensional
Electron Diffraction Data in Human Bone
HAYDAR BURAK ERKIZAN

© HAYDAR BURAK ERKIZAN, 2026.

Supervisor: Chiara Micheletti
Examiner: Chiara Micheletti

Master's Thesis 2026
Department of Physics and Astronomy
Microstructure Physics
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Apatite *c*-axis orientation map in human bone obtained by 4D-STEM virtual dark-field imaging and overlaid on a HAADF-STEM image.

Typeset in L^AT_EX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2026

Python-Based Analysis of Multidimensional
Electron Diffraction Data in Human Bone
HAYDAR BURAK ERKIZAN
Department of Physics and Astronomy
Chalmers University of Technology

Abstract

The main focus of this thesis is to study the nanoscale crystallographic organization of human cortical bone using detector-based four-dimensional scanning transmission electron microscopy (4D-STEM). Because conventional electron diffraction averages over relatively large areas, it is difficult to relate diffraction features directly to local real-space structure in bone. This limitation is addressed with a previously acquired 4D-STEM dataset and a simultaneously recorded HAADF-STEM image, where py4DSTEM and Dragonfly-based real-space segmentation were used. The analysis was based on ROI selection, mean diffraction pattern (DP) construction, detector shape and position selection, virtual dark-field imaging, real-space masking, thresholding, and diffraction reconstruction from masked and thresholded pixels. The broad diffraction arc commonly associated with the apatite (002) reflection was used to guide the detector placement. Bright-band, dark-band, and extra-fibrillar mineral regions were segmented in Dragonfly using the HAADF-STEM image for real-space masking purposes. The results reveal that the diffraction signal along the selected arc is spatially heterogeneous, with different detector locations highlighting different local mineralized regions. An aggressive positive thresholding analysis separates the pixels that have strong diffraction contributions from a broader weakly contributing background, and the resulting reconstructed DPs of positive- and negative-thresholded DF images display either strong Bragg-like disk features or a broader ring geometry, respectively. Comparisons of different mask-selected regions show that the revealed structural regions do not separate into sharply distinctive diffraction classes, given the detector conditions used. However, this work suffers from some limitations, such as sample thickness, heterogeneity, diffuse scattering, low signal-to-noise ratio (SNR), and beam sensitivity. Nonetheless, the thesis demonstrates that masking 4D-STEM data in both real and reciprocal space can be used to connect local diffraction features to nanoscale structural variations in bone. Further work is needed to improve the above-mentioned limitations and the overall interpretation of our findings.

Keywords: 4D-STEM, py4DSTEM, virtual dark-field imaging, diffraction analysis, electron diffraction, human bone.

Acknowledgements

I would like to express my sincere gratitude to my supervisor and examiner, Chiara Micheletti, for her guidance, encouragement, and valuable feedback throughout this thesis work. Her support and insight were essential in shaping both the scientific direction of the project and my development during this work.

I would also like to extend my appreciation to Kathryn Grandfield, Peter Ercius, Alexandre Pofelski, and Andrea Vera for their collaboration, helpful and insightful discussions, and contributions to the research context of this thesis. Their expertise and support were greatly appreciated.

My thanks also go to Aqila Naseri and Erica Anderson for their help and encouragement during this project.

In addition, I would like to thank the members of the Microstructure Physics group for creating a welcoming and stimulating research environment.

Finally, I am grateful to everyone who supported me during this thesis work.

Haydar Burak Erkizan, Gothenburg, July 2026

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis, listed in alphabetical order:

4D-STEM	Four-Dimensional Scanning Transmission Electron Microscopy
BF	Bright Field
DF	Dark Field
DM4	DigitalMicrograph 4 file format
DPC	Differential Phase Contrast
DP	Diffraction Pattern
FIB	Focused Ion Beam
HAADF	High-Angle Annular Dark-Field
HAADF-STEM	High-Angle Annular Dark-Field Scanning Transmission Electron Microscopy
HDF5	Hierarchical Data Format version 5
MCF	Mineralized Collagen Fibril
NCEM	National Center for Electron Microscopy
NCP	Non-Collagenous Proteins
NERSC	National Energy Research Scientific Computing Center
ROI	Region of Interest
SAED	Selected Area Electron Diffraction
SNR	Signal-to-Noise Ratio
STEM	Scanning Transmission Electron Microscopy
TEM	Transmission Electron Microscopy
VDF	Virtual Dark Field

Contents

List of Acronyms	ix
List of Figures	xiii
1 Introduction	1
1.1 Background	1
1.2 Aim and research questions	2
1.3 Scope	2
1.4 Outline of the thesis	2
2 Literature	3
2.1 The Structure of Bone	3
2.1.1 Bone Multiscale Organization	3
2.1.2 Bone at the nanoscale level	3
2.1.3 General principles of TEM and STEM imaging	6
2.1.4 General principles of conventional electron diffraction (SAED)	7
2.1.5 4D-STEM: principles, advantages, and information that can be extracted	8
3 Materials and Methods	13
3.1 Experimental dataset and specimen preparation	13
3.2 Python analytical setup and preprocessing	13
3.3 General diffraction analysis	14
3.3.1 ROI selection and mean DPs	14
3.3.2 Radial and angular characterization of diffraction features . .	14
3.3.3 Masking and thresholding	15
3.3.3.1 Real-space segmentation and mask generation	15
3.3.3.2 Threshold-based selection of masked DF image . . .	15
3.3.4 VDF imaging	15
3.3.4.1 Detector geometries	16
3.3.4.2 Masked DF images	16
3.3.4.3 Multiple-detector analysis	16
3.3.5 Diffraction reconstruction from selected real-space populations	17
3.4 Limitations	17
3.5 AI usage statement	19
4 Results and Discussion	21

4.1	Real-space overview and segmentation	21
4.2	DP of the selected ROI	21
4.2.1	Masked real-space and resulting DP	24
4.3	{002} Arc detection	25
4.4	Off-axis DF imaging	25
4.5	Masked off-axis DF image	27
4.6	Threshold-based analysis	29
4.6.1	Thresholded regions	29
4.6.1.1	Positive-threshold regions	29
4.6.1.2	Negative-threshold regions	31
4.6.1.3	DPs reconstructed from threshold-selected pixels	31
4.7	Alternative detector geometry: moon-shaped detector	32
4.8	Orientation of the c-axis in multiple-detector analysis	34
4.9	Additional results for specific ROIs	35
5	Conclusion and outlook	41
5.1	Future research	41
	Bibliography	43
A	Appendix 1	I
A.1	Other results	I
B	Appendix 2	V
B.1	Example of Jupyter notebook	V

List of Figures

2.1	Schematic representation of the multiscale hierarchical organization of bone. The structure spans from the macroscopic whole bone down to microstructural osteons and lamellar motifs, highlighting the integration of both ordered and disordered arrangements. At the fundamental nanoscale, the composite consists of the MCF, non-collagenous proteins (NCP), and water. Adapted from Reznikov et al. [7].	4
2.2	MCF have a characteristic pattern of alternating gap and overlap zones along their length. These fibrils have two distinct mineralization sites, the intra-fibrillar and extra-fibrillar compartments. Adapted from [6].	5
2.3	Schematic model of bone ultrastructure illustrating the spatial distribution of bone mineral relative to the collagen fibrils. This model highlights the substantial presence of extra-fibrillar mineral—often described as a "deck of cards" arrangement—where curved mineral plates surround and coat the exterior of the fibrils, complementing the traditional intra-fibrillar mineral found within the gap zones. Adapted from McNally and Schwarcz [11].	6
2.4	Effect of section orientation on the HAADF-STEM appearance of mineralized collagen in bone. Parallel sections show fibrillar banding, whereas perpendicular sections show circular fibril cross-sections surrounded by mineralized structures. Adapted from Grandfield <i>et al.</i> [13].	7
2.5	BF TEM image and corresponding SAED pattern of a human femur section cut parallel to the long axis. The broad $00l$ arcs, especially the (002) reflection, indicate preferential alignment of the apatite c -axis parallel to the collagen fibrils. Adapted from McNally <i>et al.</i> [11]. . . .	8
2.6	BF TEM image and corresponding SAED pattern of a human femur cross-section. The more ring-like DP and spotty $00l$ reflections are consistent with apatite c -axes oriented largely normal to the section plane. Adapted from McNally <i>et al.</i> [11].	9
2.7	Visualization of a 4D-STEM experiment. The main component of this measurement is the convergent probe, and the acquisition is made with rastering the probe across the sample in real space while recording a full DP at each scan position, which generates a four-dimensional dataset.. Adapted from [15].	10

2.8	Real-space (x, y) and reciprocal-space (k_x, k_y) visualization of 4D-STEM data structure showing a four-dimensional data set. Reproduced from [16].	11
3.1	Dragonfly segmentation wizard interface. A manually annotated ROI (bottom) is used to train the model, which is then applied to the full image to generate automated pixel-wise segmentation (top) of mineral, bright-band, and dark-band regions.	16
3.2	Schematic overview of the 4D-STEM workflow, from HAADF image and ROI selection to VDF imaging, masking, thresholding, and diffraction reconstruction.	18
4.1	Schematic of the 4D-STEM analysis workflow. The workflow begins with a simultaneously acquired HAADF-STEM image and ends with DPs of both positive and negative thresholded masked DF images. In this specific example, the bright band mask is used in the masked DF image.	22
4.2	Real-space overview of the selected ROI (a) and the segmentation (b) used for masked analysis. The segmentation defines the structural regions used in the subsequent detector-based diffraction workflow. Individual Dragonfly segmentation masks used to define the main real-space classes in the HAADF-STEM image. The masks correspond to bright-band regions (c), dark-band regions (d), and extra-fibrillar mineral (d), respectively.	23
4.3	Binary masks of mineral (a), dark-band (b), and bright-band (c). DPs of mineral (d), dark-band (e), and bright-band masks (f). Mineral DP histogram (g), dark-band DP histogram (h), and bright-band DP histogram (i).	24
4.4	Mean DP obtained by averaging all DPs within the selected ROI. The pattern is dominated by the central transmitted beam and broad concentric scattering rings, and it serves as the reference for beam-center determination and detector placement in the subsequent analyses. 002 diffraction arcs are indicated by two arrows.	25
4.5	Workflow for identifying the broad {002}-associated arc using radial and angular analysis of the ROI-averaged DP.	26
4.6	Off-axis DF imaging of the selected ROI. A circular detector was placed on a localized sector of the broad diffraction ring, and the corresponding DF image was computed from the 4D-STEM datacube. The resulting image highlights discrete real-space patches associated with the selected diffraction sector.	27
4.7	Masked off-axis DF image obtained by applying the bright-band binary mask from Figure 4.2c to the off-axis detector response. The image retains the overall banded morphology of the selected bright-band region while showing non-uniform intensity along the masked structure.	28

4.8	Thresholded versions of the masked off-axis DF images ((a) and (b)) derived from the bright-band mask shown in Figure 4.2c. The positive-threshold image (a) isolates the strongest detector-response regions, while the negative-threshold image (b) retains the weaker-intensity component of the masked bright-band structure. DPs ((c) and (d)) reconstructed from threshold-selected pixels within the bright-band mask. The positive-threshold reconstruction (c) shows localized bright maxima on the diffraction ring, whereas the negative-threshold reconstruction (d) retains the same overall ring geometry but with a more diffuse intensity distribution.	30
4.9	DF imaging using a moon-shaped detector geometry placed on a broad sector of the diffraction ring (a). If interpreted as sampling part of the {002} diffraction arc, the resulting DF image (b) highlights regions contributing strongly to that sector of reciprocal space. DPs reconstructed from threshold-selected pixels within the dark-band mask. The positive-threshold reconstruction emphasizes localized azimuthal intensity maxima on the diffraction ring, whereas the negative-threshold reconstruction retains the same overall ring geometry but with a more diffuse intensity distribution.	33
4.10	Multiple-detector analysis of the selected ROI. A set of off-axis circular detectors was placed along the broad diffraction arc visible in the ROI-averaged DP, consistent with the approximately 30°-wide arc reported for the {002} reflection of bone mineral in samples with collagen fibrils oriented in-plane [14]. The resulting color-coded DF overlay reveals localized patches of similarly oriented crystals, with many of these patches appearing in inter-fibrillar regions.	34
4.11	Orientation of the c-axis of the selected disks and their resulting DF images. Colored arrows show the preferred direction of minerals that grow along the c-axis.	36
4.12	Schematic of the 4D-STEM analysis workflow in a smaller selected ROI. The binary mask is the darker region inside the ROI. The threshold value is 50 in the thresholded DF image.	37
4.13	Schematic of the 4D-STEM analysis workflow in a smaller selected ROI. The binary mask is the bright-band inside the ROI. The threshold value is 50 in the thresholded DF image.	38
4.14	Schematic of the 4D-STEM analysis workflow in a smaller selected ROI. The binary mask is the extra-fibrillar mineral inside the ROI. The threshold value is 50 in the thresholded DF image.	39
A.1	Dark bands results. The same detector geometry and position were used as in the bright bands analysis presented in the Results.	II
A.2	Extra-fibrillar mineral results. The same detector geometry and position were used as in the bright bands analysis presented in the Results.	III

1

Introduction

1.1 Background

Bone is a mineralized composite material, and its organized structure spans across multiple length scales, from mineralized fibrils to larger lamellar structures. One of its important levels is a collagen-rich organic matrix reinforced by calcium phosphate mineral. Since these mineral arrangements and orientations directly affect stiffness, strength, and toughness, bone is quite a challenging material, and to understand local structural organizations, it requires characterization methods that reveal these nano-level organizations rather than relying on bulk-average measurements. The nanostructure of bone is highly heterogeneous, and this directly affects diffraction behaviors, making it difficult to link its local morphology to crystallographic information using conventional imaging methods. Because of this, we need a method that can connect real-space structures to reciprocal-space signals at nanoscale levels. Thanks to the new technological development, four-dimensional scanning electron microscopy (4D-STEM) is one of the best available candidates that can address this challenge. In 4D-STEM, a focused electron probe is rastered across the sample in real-space while a full two-dimensional diffraction pattern (DP) is recorded at every scan position. This results in a four-dimensional dataset that has 2D real-space and 2D reciprocal-space coordinates. Since the full DP is preserved at each scan position, it is possible to generate virtual images after the acquisition [1, 2]. This is the flexibility needed for complex and heterogeneous materials like bone. This flexibility enables us to apply different virtual detector placements in diffraction space in post-acquisition analysis, allowing diffraction-space information to be mapped back into real space. With this advanced technique, the difficulties of conventional fixed detector STEM imaging can be overcome, and orientation-dependent and structure-dependent scattering can be resolved at high accuracy [1, 2]. Despite all of these advantages that 4D-STEM has, extracting meaningful structural information requires a careful computational analysis. Since the data volume is relatively larger compared to more traditional methods, the variation in DPs across even small regions is considerably higher; thus, calibration, detector geometry, sample preparation, and analysis choices play important roles in the success of the experiment. Thankfully, 4D-STEM has open-source tools such as py4DSTEM that provide a practical framework in the construction of Python notebooks, such as virtual imaging, diffraction-space masking, calibration of the data, and reconstruction of selected real-space or reciprocal-space subpopulations [2].

1.2 Aim and research questions

This thesis aims at developing and applying a Python-based workflow for analyzing multidimensional electron diffraction data in human bone. The main focus of this work is to reveal spatially resolved crystallographic information obtained in 4D-STEM measurements. Based on this, detector-based virtual imaging and diffraction-space selections are investigated to see how scattering behavior changes across lamellar and banded bone regions. More specifically, this thesis tries to address the following questions:

1. Can the selection of detector-based analysis in diffraction-space reveal differences in real-space images in human bone?
2. What is the sensitivity of the resulting virtual dark-field (VDF) images and reconstructed DPs in detector position, geometry, and the threshold values?
3. What is the effect of crystallographic heterogeneity on DPs and virtual real-space images across selected regions using Python-based 4D-STEM analysis?

1.3 Scope

Previously collected 4D-STEDM datasets are the main source in this thesis, and post-acquisition analysis was made based on these datasets using the py4DSTEM libraries in Python. The main streamline of this work is region-of-interest (ROI) masking, VDF imaging, localized detector selection, threshold-based real-space segmentation, and DP reconstruction from selected and highlighted pixel populations. The goal is to establish and evaluate a practical analysis framework to analyze local diffraction behavior and spatial nano-organizations in heterogeneous bone structure.

1.4 Outline of the thesis

The remainder of this thesis is organized as follows. Chapter 2 reviews the relevant background on bone nanostructure, 4D-STEM, and computational diffraction analysis. Chapter 3 describes the materials, datasets, and analysis methods used in this work. Chapter 4 presents the results of the detector-based 4D-STEM analysis, including comparisons across regions, detector positions, and threshold choices. Finally, Chapter 5 summarizes the main conclusions and suggests possible directions for future work.

2

Literature

2.1 The Structure of Bone

2.1.1 Bone Multiscale Organization

Bone is a highly organized tissue and has many critical functions in our bodies, ranging from mechanical functions to complex biological functions. It's responsible for mechanical support, body dynamics, and protection of vital organs, and serves as the reservoir for minerals, more specifically for calcium and phosphorus [3]. Bone has many roles, and to perform these roles, it requires stiffness, strength, and toughness, properties that are obtained thanks to bone structure as a complex, hierarchical nanocomposite that spans across many length scales [4, 5]. At the macroscopic level, we can see bone as being partitioned into two distinctive architectures: cortical bone forming the dense outer shell of the skeleton, and cancellous bone forming a porous, interconnected network inside the bone [3]. At the microstructural level, cortical bone has a characteristic cylindrical structural unit known as osteons composed of concentric lamellae (typically 3–7 μm thick). In the case of cancellous bone, it consists of a lattice of struts and plates [4]. If we want to see the fundamental mechanical properties of the bone, we need to consider the nanoscale level. At this sub-micron level, bone displays a biphasic material trait comprised of a compliant, collagen-rich organic matrix reinforced by stiff, nanoscale calcium phosphate mineral crystals, combined with non-collagenous proteins and water [4, 6]. Importantly, this nanostructural organization is extremely heterogeneous. We know from recent studies that three-dimensional imaging studies have shown that bone displays a fractal-like structural hierarchy [7]. Although it is debatable, it is made of a major "ordered" material that contains very well-aligned bundles of mineralized collagen fibrils (MCF) and relatively low "disordered" material, in which less fibril orientation is observed, thus it shows high variability [7, 8]. Since mechanical behavior is affected by the spatial arrangement and orientation of the mineral phase, we need characterization methods that enable us to resolve local structural organization, rather than purely depending on bulk-averaged measurements [5].

2.1.2 Bone at the nanoscale level

One of the most important building blocks of the bone nanostructure unit is the MCF. The fibrils are triple-helical type 1 collagen molecules, and they are built into a highly ordered, staggered array. The observed characteristic periodic banding pattern, continuing along the length of the fibril, stems from this specific arrange-

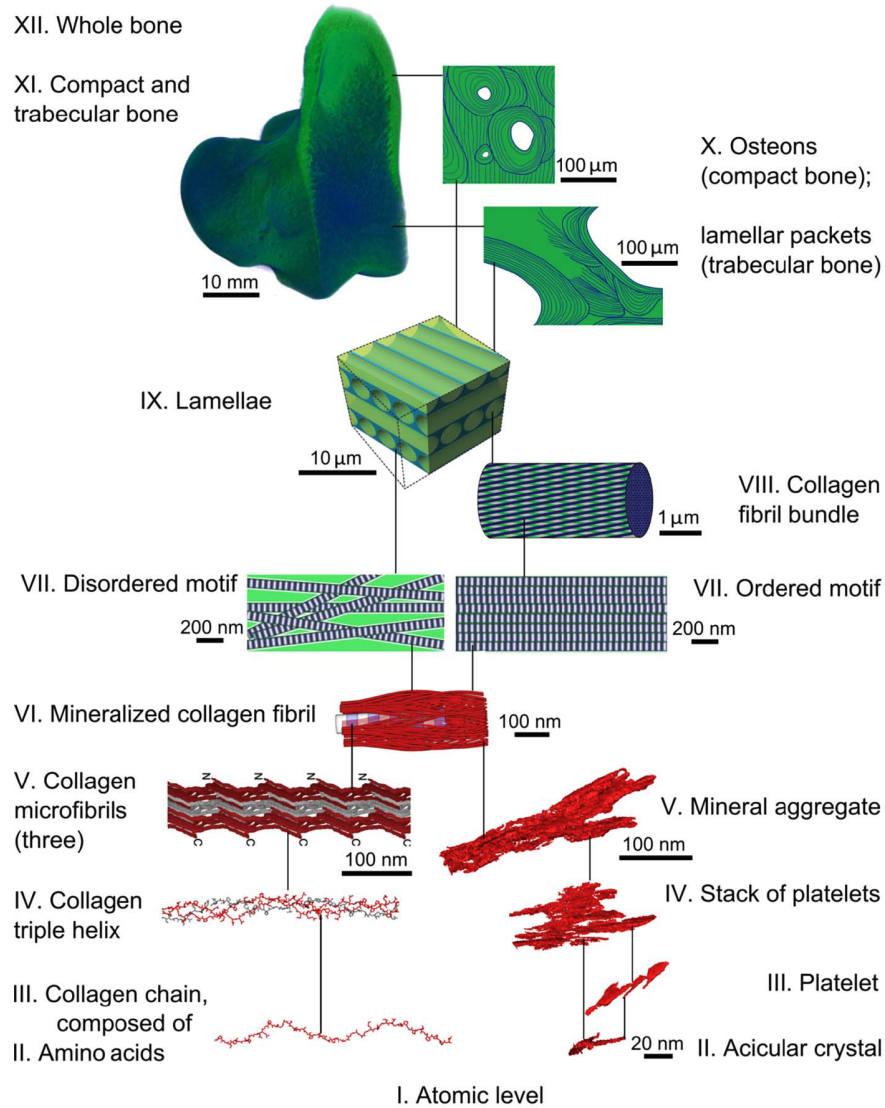


Figure 2.1: Schematic representation of the multiscale hierarchical organization of bone. The structure spans from the macroscopic whole bone down to microstructural osteons and lamellar motifs, highlighting the integration of both ordered and disordered arrangements. At the fundamental nanoscale, the composite consists of the MCF, non-collagenous proteins (NCP), and water. Adapted from Reznikov et al. [7].

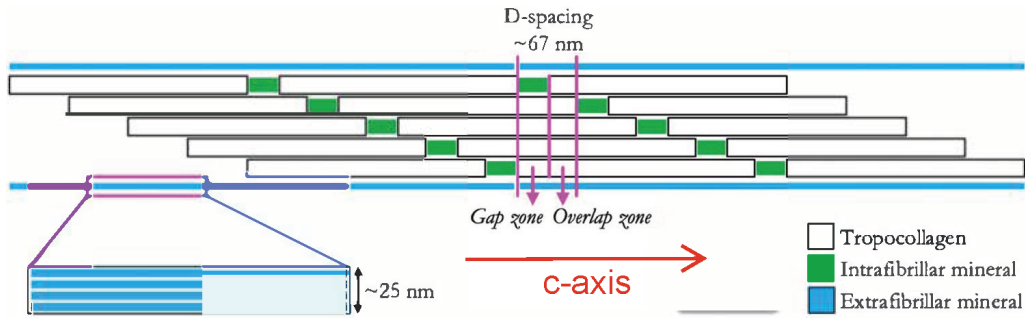


Figure 2.2: MCF have a characteristic pattern of alternating gap and overlap zones along their length. These fibrils have two distinct mineralization sites, the intra-fibrillar and extra-fibrillar compartments. Adapted from [6].

ment, giving rise to a 67 nm D-spacing [9]. As a result, we see these alternating regions consisting of high and low protein density known as the overlap and gap zones [4, 9]. Gap zones act as the primary sites for initial mineral nucleation and deposition. Bone mineral is considered an apatite-like calcium phosphate in the shape of small plates, ranging from 2 nm to 5 nm in thickness and around a few tens of nm in length and width [6, 7]. From the early classical models, it is observed that bioapatite crystals preferentially nucleate within these 40 nm gap channels, creating intra-fibrillar mineral arrays that have their c-axes aligned parallel to the longitudinal axis of the collagen fibril [10]. This model shows that the mineral fibril displays an isolated composite unit. Having said that, developments in electron microscopy techniques, particularly the imaging of ion-milled sections, have changed the way we understand mineral distribution beyond the fibril. It is now broadly accepted that a substantial amount of bone mineral is extra-fibrillar mineral [11]. As opposed to the belief that minerals are confined uniquely to the intra-fibrillar gap zones, these extra-fibrillar crystals are extended, plate-like structures that are found outside of the collagen fibrils and fill the inter-fibrillar regions, and they are often seen as a "deck of cards" arrangement [11]. DF transmission electron microscopy (TEM) enables us to study this gap zone model. Observations made in using DF imaging show that these strongly diffracting apatite crystals are often related to the extra-fibrillar structures being found around the fibrils, other than being completely localized within the gap zones [12]. Based on the studied DF images, the authors came to the conclusion that the majority of crystalline minerals are extra-fibrillar [12]. From the perspective of other authors, bone has a fractal-like architecture, and the bone mineral has a hierarchical structure where intra- and extra-fibrillar minerals are in the form of a continuous, interconnected network [6, 7]. Therefore, this hierarchical bone mineral is not simply viewed as isolated intra- and extra-fibrillar particles. According to Reznikov *et al.*, the mineral aggregates surround adjacent fibrils as a means of continuous cross-fibrillar mineralization and, as described by Shah *et al.*, these intra- and extra-fibrillar minerals are distinctive compartments merged in a way that they show a cohesive continuum phase [6, 7].

When dealing with bone, the orientation of the section in the bone lamella is very important because the same collagen-based structure appears differently in electron microscopy images. In parallel sections (or in-plane view), fibrils display the

characteristic 67 nm banding, and when the section plane is tilted, the resulting TEM image shows a lacy-like pattern, whereas in perpendicular sections (or out-of-plane view), the fibrils are seen as dark circular features surrounded by mineralized structures. These different sections of view are visualized in Figure 2.4.

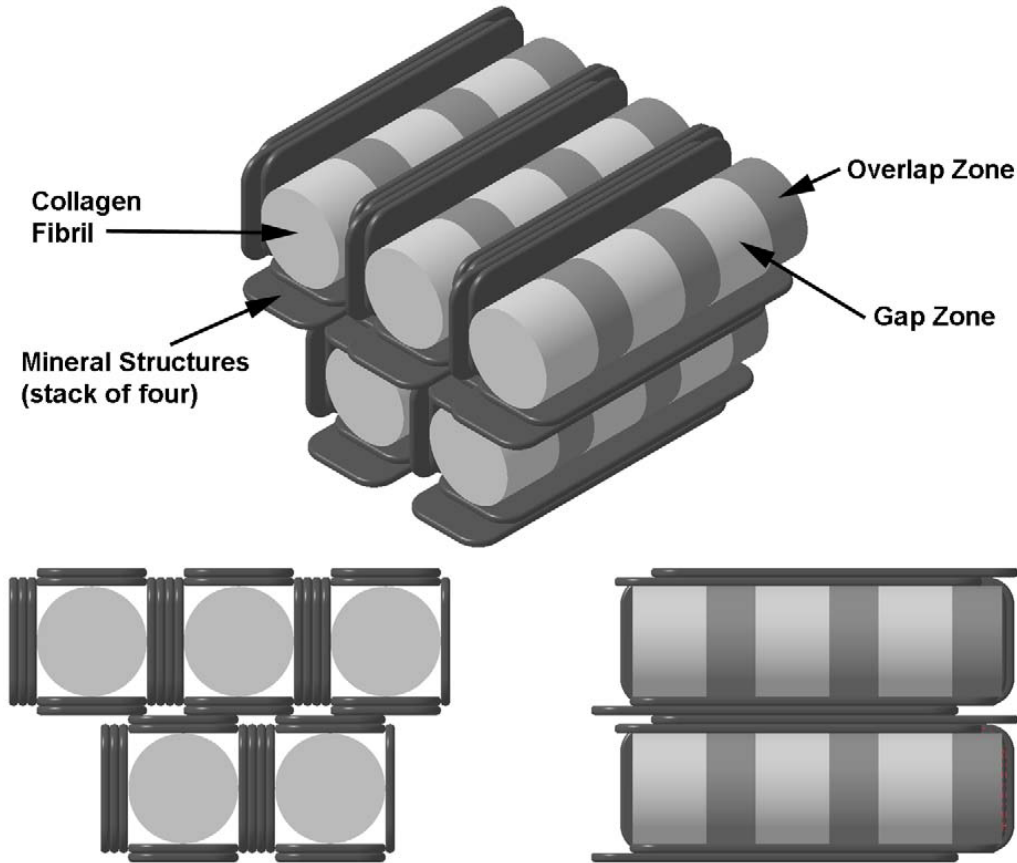


Figure 2.3: Schematic model of bone ultrastructure illustrating the spatial distribution of bone mineral relative to the collagen fibrils. This model highlights the substantial presence of extra-fibrillar mineral—often described as a "deck of cards" arrangement—where curved mineral plates surround and coat the exterior of the fibrils, complementing the traditional intra-fibrillar mineral found within the gap zones. Adapted from McNally and Schwanerz [11].

2.1.3 General principles of TEM and STEM imaging

TEM and scanning transmission electron microscopy (STEM) are among the most suitable techniques for studying bones at the nanoscale. But there are some difficulties to be overcome, especially in the sample preparation process, because this first step is crucial for the entire experiment, and it directly affects the results. In these methods, the sample needs to be thin enough to enable electron transfer. One of the limitations in earlier work is the process of obtaining thin, distortion-free electron-transparent sections of mineralized tissue for ultrastructural studies. With the recent development in technology, we can use advanced techniques like focused

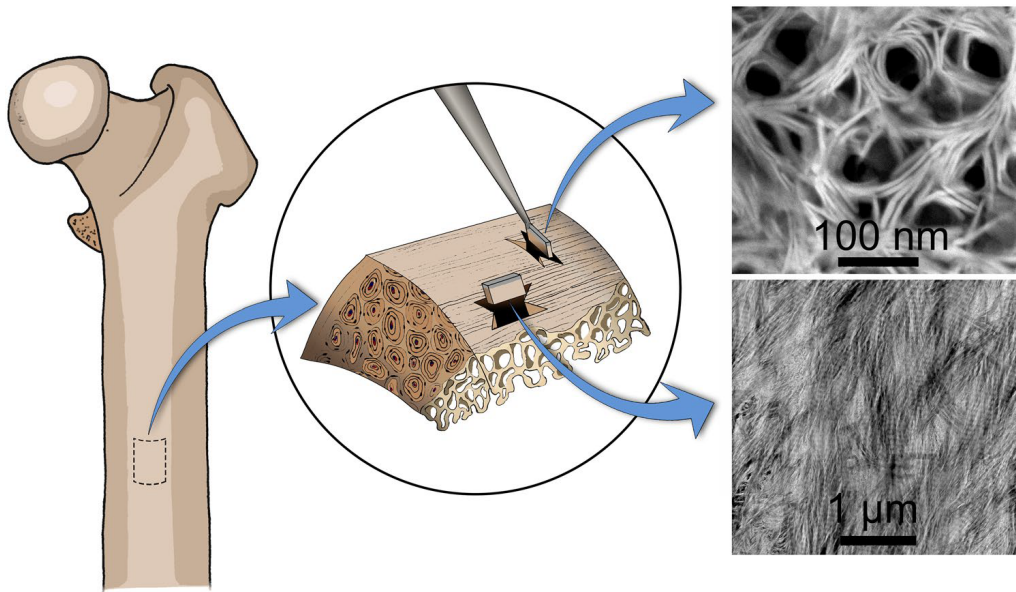


Figure 2.4: Effect of section orientation on the HAADF-STEM appearance of mineralized collagen in bone. Parallel sections show fibrillar banding, whereas perpendicular sections show circular fibril cross-sections surrounded by mineralized structures. Adapted from Grandfield *et al.* [13].

ion beam (FIB) milling, enabling site-specific preparation of electron-transparent lamellae [3, 13]

Traditional bright-field TEM uses an image-contrast technique that preserves intact mineral plates, as opposed to earlier methods like ultramicrotomy, formed mainly by electrons transmitted through the section, whereas DF TEM captures the scattering electrons (Bragg-scattered) from a selected lattice plane [11]. On the other hand, STEM uses a focused electron probe, which is rastered across the sample in real space, and the contrast of the image results from the detector geometry in diffraction space. A fixed detector on the optic axis captures bright-field-like contrast, while annular detectors collect scattered electrons coming from selected angular ranges. Since the image intensity in high-angle annular dark-field (HAADF) STEM image is approximately proportional to the projected atomic number, mineral-rich regions appear as bright spots [2, 13].

2.1.4 General principles of conventional electron diffraction (SAED)

Conventional selected area electron diffraction (SAED) in bone is usually performed over a selected region defined by the aperture that captures the DP, and this reflects the lattice spacings and preferred orientation of the mineral phase within that region. If we inspect bone from the longitudinal section (See Figure 2.5), then the BF TEM image shows the characteristic collagen banding, and the corresponding SAED pattern often contains broad 002 and 004 arcs, showing preferential alignment of the apatite c-axis parallel to the collagen fibrils [11]. In transverse or out-of-plane views (See Figure 2.6), the diffraction exhibits more ring-like patterns, and 002-related

reflections are not expected to be seen, and this is consistent with the apatite c-axis being tilted out of the image plane [7, 11].

When it comes to identifying preferred mineral orientation in bone, SAED is extremely important, but it comes with an important limitation: each DP is the result of averaged diffraction over a relatively large region defined by the selected-area aperture, which is typically tens to hundreds of nm and can therefore collect diffraction information coming from several MCF [14]. As a result, the overall DP cannot be assigned point-by-point to the local structure. DF imaging allows us to select a specific reflection or arc segment from the SAED pattern to obtain a real-space image, but the detector choice cannot be changed after the acquisition, thus the full diffraction information is not collected for each image pixel [11].

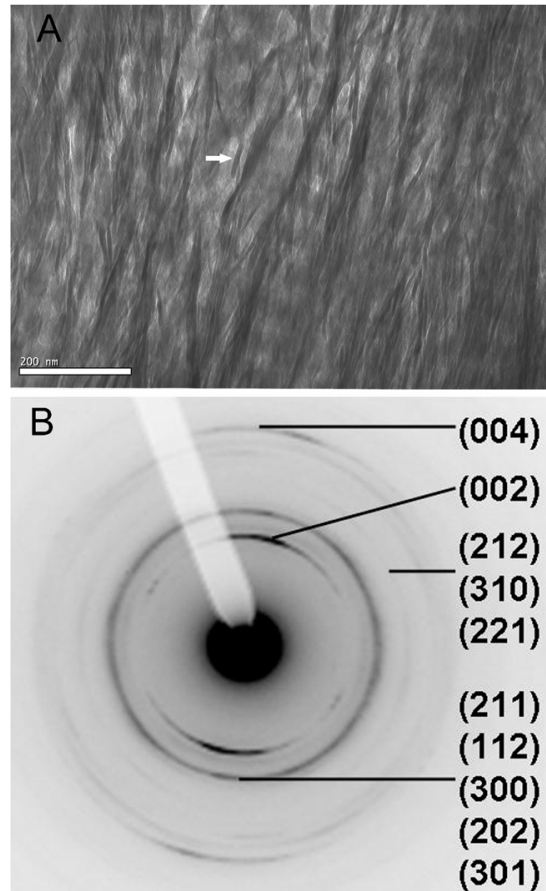


Figure 2.5: BF TEM image and corresponding SAED pattern of a human femur section cut parallel to the long axis. The broad $00l$ arcs, especially the (002) reflection, indicate preferential alignment of the apatite c-axis parallel to the collagen fibrils. Adapted from McNally *et al.* [11].

2.1.5 4D-STEM: principles, advantages, and information that can be extracted

4D-STEM improves the limitations of SAED by recording a full DP at every scan position. Thus the resulting dataset consists of two real-space coordinates (x, y) and

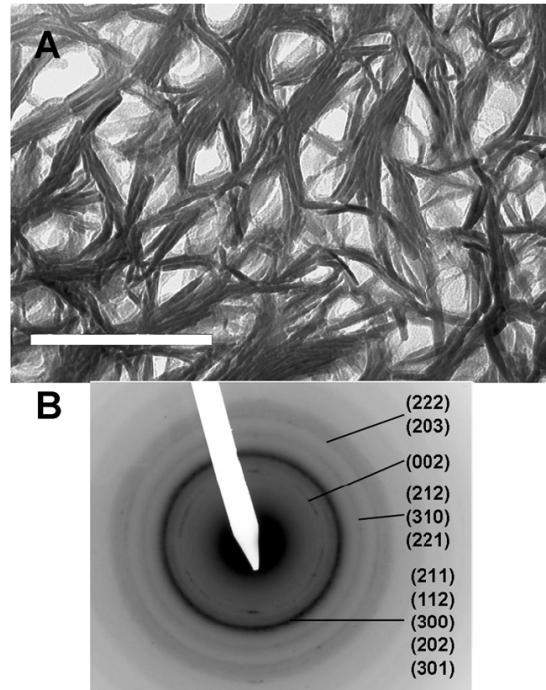


Figure 2.6: BF TEM image and corresponding SAED pattern of a human femur cross-section. The more ring-like DP and spotty $00l$ reflections are consistent with apatite c -axes oriented largely normal to the section plane. Adapted from McNally *et al.* [11].

two diffraction-space coordinates (k_x, k_y) [1, 2]. The main advantage of 4D-STEM over conventional STEM is the fact that it preserves the full angular distribution of the scattered electrons. This advantage enables defining virtual detectors after acquisition and reconstructing virtual BF, DF, annular, arc-shaped, or other customizations of the detector shapes [1, 2].

With this major advantage of the 4D-STEM, we can use a single dataset with different analysis modes, including virtual imaging, orientation mapping, phase mapping, radial and angular diffraction analysis, and, in appropriate materials, even strain mapping [1, 2]. When it comes to bone, the most relevant benefit is the fact that 4D-STEM allows crystallographic information to be connected to the local real-space structure with high accuracy. Since it is really important for the bone studies to be studied down to individual nm-sized pixels, this method makes it possible.

On the other hand, 4D-stem has its own challenges, like a substantial increase in analytical complexity. Recording a DP at every probe position comes with a price for extracting meaningful measurements, and these are mainly as follows: it requires several computational steps, including data handling, calibration, preprocessing, detector definition, and visualization. As Savitzky *et al.* emphasized, this complexity was the main motivation towards the development of py4DSTEM as an open-source analysis framework, which in fact is designed to make 4D-STEM workflows more understandable, accessible, reproducible, and most importantly capable of multimodal analysis from a single dataset, enabling post-acquisition analysis [2]. In this context of the framework, structured data objects such as `DataCube`, `DiffractionSlice`,

and `RealSlice` are the core that allow real-space and diffraction-space information to be processed smoothly and consistently while also supporting calibration, virtual imaging, orientation, and phase mapping, radial analysis, and other kinds of post-acquisition measurements [1, 2].

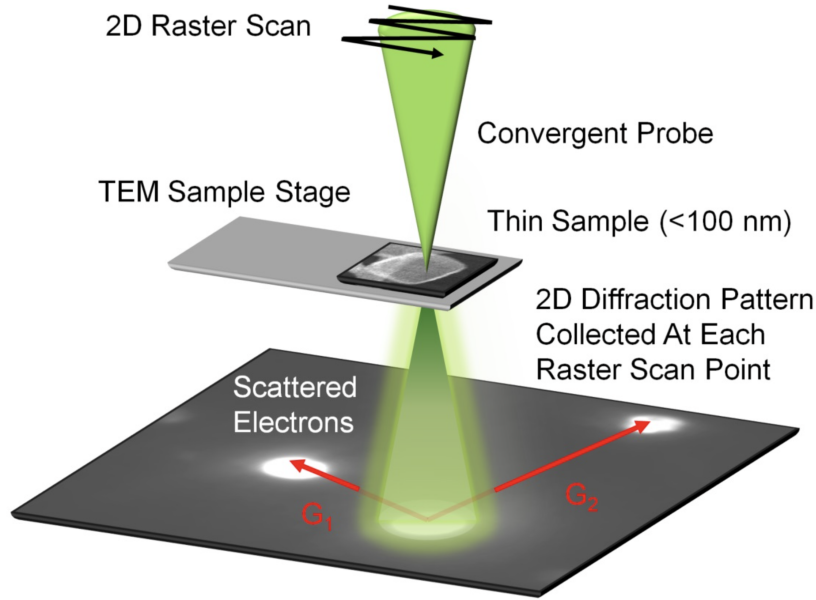


Figure 2.7: Visualization of a 4D-STEM experiment. The main component of this measurement is the convergent probe, and the acquisition is made with rastering the probe across the sample in real space while recording a full DP at each scan position, which generates a four-dimensional dataset.. Adapted from [15].

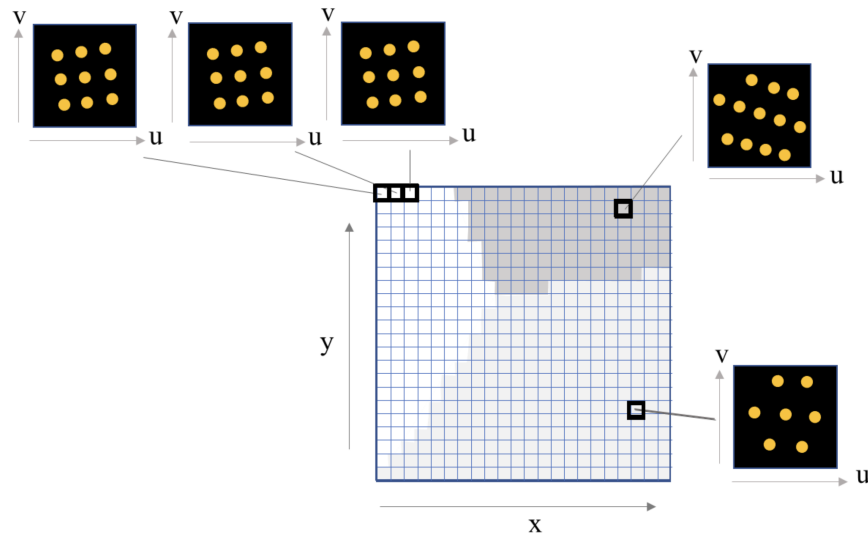


Figure 2.8: Real-space (x, y) and reciprocal-space (k_x, k_y) visualization of 4D-STEM data structure showing a four-dimensional data set. Reproduced from [16].

3

Materials and Methods

3.1 Experimental dataset and specimen preparation

The present study was performed with a previously acquired 4D-STEM dataset from human femoral cortical bone and based on a simultaneously acquired HAADF-STEM image that is used for real-space visualization and segmentation. The specimen (electron-transparent lamellae) was prepared with FIB milling in orientation relative to the long axis of the femur. With this advanced technique, collagen fibrils were analyzed in an in-plane view [14].

The experiment was conducted at the National Center for Electron Microscopy (Lawrence Berkeley National Laboratory, USA), and the data were acquired in scanning mode on a TEAM 0.5 S/TEM instrument, operated at 300 kV with a convergence semi-angle of 0.7 mrad and equipped with an ultrafast 4D camera [14]. Over one million DPs were recorded from scan areas of approximately $2 \times 2 \mu\text{m}^2$, using a probe step size below 2 nm and an acquisition speed of approximately 87,000 frames per second [14].

After acquiring the data, electron counting (data reduction) was applied, and the reduced data were stored in sparse HDF5 (Hierarchical Data Format version 5) format on the NERSC community file system. The simultaneously acquired HAADF-STEM image was stored as a DigitalMicrograph 4 (DM4) file containing acquisition metadata such as probe step size and stage values. 4D-STEM data and HAADF-STEM files were the starting point for the computational workflow described in the following sections [17].

3.2 Python analytical setup and preprocessing

All of the data was analyzed in Python using Jupyter notebooks. py4DSTEM was the main analysis framework, and stempy, ncempy, tiffio, and imageio were used for additional data handling processes. In addition to that, the official py4DSTEM software repository and tutorials repository were used as a guide for coding implementation during the development of the notebooks [18]. The notebook started with a pre-centered sparse HDF5 dataset produced by an upstream vacuum-reference centering workflow. Since beam-position shifts had already been corrected, there was no need for the correction in the notebook used in the present thesis.

The first preprocessing step of the notebook includes loading the STEM file and

the corresponding HAADF-STEM DM4 image. After that, the scan dimensions and diffraction-frame dimensions were checked, and then the real-space ROI was defined by using the HAADF image. After that, ROI coordinates were applied to both the HAADF image and the sparse 4D-STEM dataset to ensure later steps in real-space masking and virtual images were within the same cropped field of view. Next, the cropped sparse dataset was wrapped into a `py4DSTEM DataCube`, which acted as the main object for diffraction analysis, virtual imaging, and diffraction reconstruction in the later workflow.

3.3 General diffraction analysis

In this section, the diffraction analysis follows the workflow shown in Figure 3.2, and each step is explained in detail as follows.

3.3.1 ROI selection and mean DPs

At the beginning of the analysis, the ROI was defined in the HAADF image. After selecting ROI, the same coordinates were used in the 4D-STEM dataset to ensure they reflect the same coordinates between the real-space and the diffraction data. For each selected ROI, a mean DP was obtained by averaging the diffraction intensity over all probe positions within the region. These ROI-averaged DPs were the base for reciprocal-space images to identify the main scattering rings and arc, to find the diffraction center, and to define the detector geometries. Moreover, they were employed in the later virtual-imaging workflow.

3.3.2 Radial and angular characterization of diffraction features

After obtaining ROI-averaged DPs, radial and angular analysis were performed for the detector placement in diffraction space. In the next step, the center of the DP was found based on the central intensity maximum, and if needed, manual correction could be applied. Radial intensity profiles were then computed by azimuthally averaging the mean DP as a function of radius in detector-pixel coordinates.

In order to identify radii with enhanced scattering intensity, the radial profile was used. In the radial profile graph, after a high-fluctuation region shown in Figure 4.5, each peak represents a circle in the DP. Although there were many candidates for annulus, the first broad annular maximum outside the central beam was selected, which is commonly associated with the broad $\{002\}$ arc in in-plane fibrils. An annulus with inner radius r_{in} and outer radius r_{out} was then defined around this feature.

After selecting the annulus, the azimuthal angle θ was used in the calculation of the mean diffraction intensity to generate a ring-angular-intensity profile. The ring angular intensity figure shows where the scattering is strongest inside the selected annulus, and then this angular profile was used to locate the sector of enhanced scattering around the ring. As a practical guide, local maxima in the angular intensity were used to place off-axis virtual detectors to obtain DF images.

3.3.3 Masking and thresholding

3.3.3.1 Real-space segmentation and mask generation

The HAADF-STEM image was used in real-space segmentation in Dragonfly to define structural regions. Dragonfly (Object Research Systems, Canada) is a commercial company that focuses on scientific image-processing and analysis for 2D, 3D, and 4D imaging data. It has extensive tools for visualization, segmentation, quantification, measurement, and deep-learning-assisted image analysis. To identify structural classes visible in the HAADF-STEM image, a relatively small region was selected in which the features of the classes were especially visible, and then this small region was manually painted with three classes, namely, bright bands, dark bands, and extra-fibrillar mineral (Figure 3.1 bottom image). After that, these painted regions were used as a training set to teach the deep learning algorithm tool in Segmentation Wizard. This module provides a guided workflow for deep-learning-based segmentation from a limited set of manually labeled regions. After the deep-learning algorithm was trained, it applied the segmentation to the whole HAADF image (Figure 3.1 top image). Since only one slice was available in the data, the machine learning algorithm did not capture the overall segmentation; thus, it did not produce high-quality work. As a result, the automated segmentation output was manually revised. The final masks were saved as binary images and imported into Python as boolean arrays.

Each pixel in the bright-band, dark-band, and extra-fibrillar mineral was exclusively painted, meaning these painted pixels belong to only one class and were not simultaneously assigned to any other mask. This distinction is important because it ensured that the resulting mask-selected DPs were based on non-overlapping scan positions. However, since the sample is relatively thick, it may still have captured signals from hidden crystals in the deeper layers of the lamella.

3.3.3.2 Threshold-based selection of masked DF image

To detect strongly and weakly contributing subsets of pixels in a given detector-based response, thresholding was applied to the DF images. Then, positive and negative thresholds were applied to the DF images to retain the strongest and weakest contributions, respectively, in the resulting DP. Threshold values were chosen manually for each workflow and then tested to adjust the robustness of the results. With more aggressive threshold values, it is possible to test and see a sparse subset of the strongest detector-selected pixels, and less aggressive threshold values were tested to observe the expected broader fraction of the masked structural population.

3.3.4 VDF imaging

VDF images were obtained by integrating the intensity within the detector region in diffraction space in each probe position of the 4D-STEM datacube. Unlike TEM DF imaging, which has a fixed detector geometry during the experiment, in 4D-STEM, this is performed computationally in post-acquisition analysis. The virtual imaging workflow based on off-axis detectors in this study was implemented in py4DSTEM,

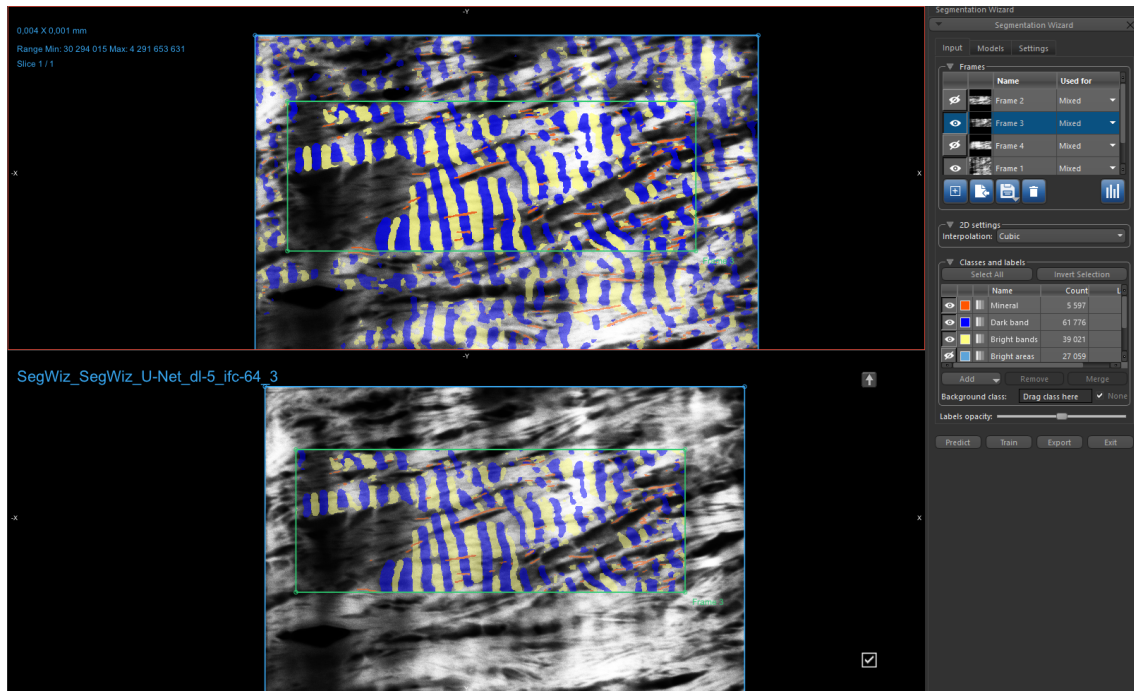


Figure 3.1: Dragonfly segmentation wizard interface. A manually annotated ROI (bottom) is used to train the model, which is then applied to the full image to generate automated pixel-wise segmentation (top) of mineral, bright-band, and dark-band regions.

using the DataCube as the main analysis object [1, 2].

3.3.4.1 Detector geometries

Several different detector geometries were explored and compared to probe different diffraction-space features. At first, annular detectors were used to see the overall distribution of the mineralized domains in DF. Then, the single disk detector position was defined based on the ring angular intensity profile, and lastly, for exploratory comparison purposes, crescent- or moon-shaped detectors were tested.

3.3.4.2 Masked DF images

A VDF image was constructed from the chosen detector geometry. To analyze specific regions in the DF image, the HAADF-derived real-space masks were applied as boolean masks as described in Section 3.3.3.1. In this process, each mask (the bright-band, dark-band, and extra-fibrillar mineral) was assigned to only one structural class and analyzed individually. This yielded masked DF images corresponding to non-overlapping subsets of scan position for each structural category.

3.3.4.3 Multiple-detector analysis

To analyze the orientation of the c-axis, several circular detectors were placed along the 002 arc region. Each circular detector is assigned to a color and constructs its

own DF image. Then these DF images were combined to create a color map, and then it was overlaid onto the HAADF image.

3.3.5 Diffraction reconstruction from selected real-space populations

After thresholding, VDF images were applied to the datacube (the original data in 4D-STEM) to reconstruct DPs from those selected scan positions, and in this way, mask-specific thresholded mean DPs were obtained for positive thresholded and negative thresholded DF images individually. This procedure was applied to each mask, which was exclusive in image space, thus DPs of different classes were computed from non-overlapping pixels. With this approach, the reciprocal-space signals of different real-space pixel populations could be analyzed by thresholding and evaluated consistently in a detector-based workflow, as shown in Figure 3.2. In this flow chart, at first, DPs are constructed from selected ROI with an applied binary mask. After inspection of these DPs, the workflow continues with DPs of unmasked ROI, and it follows the chart, which is also shown visually in Figure 4.1.

3.4 Limitations

The workflow is designed to compare detector-based DPs with real-space masking within a selected ROI in an HAADF-STEM image of human bone. The analysis notebook did not implement fully verified physical calibration in either real-space or reciprocal-space, and for this reason, the reconstructed DP and detector geometry were analyzed in pixel coordinates. The main motivation of this thesis is to study real-space masking in selected regions, relative differences between detector geometries and thresholds, rather than quantitative scattering-vector measurements. When it comes to the bone study, two of the important limiting factors are the thickness of the specimen and sample preparation. No matter how well the sample is prepared (e.g., focused ion-milled bone), the thickness still remains high on the order of $\sim 200\text{-}300$ nm, and what this means for two-dimensional projection is that the resulting 2D images includes the features of 3D structures along the entire sample thickness and this affects the DP and complicates direct assignment of image contrast to a specific regions in bone. As Grandfield *et al.* pointed out, gap- and overlap-banding could be directly affected and become poorly defined, and this may lead to the creation of invisible parts of a section due to the thickness of the sample. A further crucial point is the presence of empty holes or low-contrast circular regions because these abnormalities can also affect the contrast and create artifacts, and they are mostly stemming from preferential erosion of collagen-rich regions during ion milling or FIB preparation [11–13].

Another important limitation, and maybe the most important one, is bone heterogeneity. Since the variation of the mineralized collagen nanocomposite is strong across lamellae and with fibril orientation, it can be easily anticipated that even neighboring scan positions can produce different structural motifs, and depending on this, different crystallographic responses. Grandfield *et al.* reported that there

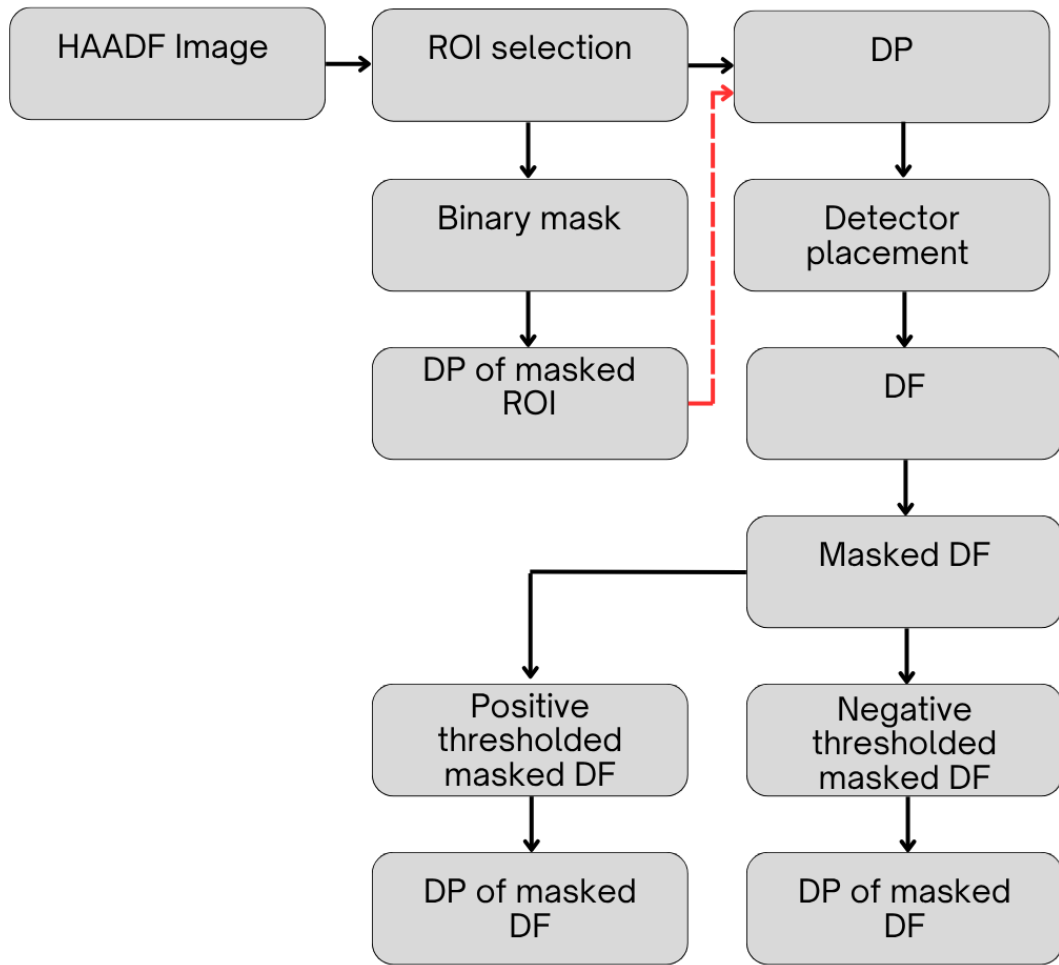


Figure 3.2: Schematic overview of the 4D-STEM workflow, from HAADF image and ROI selection to VDF imaging, masking, thresholding, and diffraction reconstruction.

are some abrupt changes in fibril orientation between adjacent osteonal lamellae. Beam sensitivity to the electron beam and, because of this, dose limitations also affect the experiment. It is a well-known fact that bone, particularly collagen-rich regions, is prone to radiation damage. McNally *et al.* reported that beam damage is inevitable after long TEM exposures, and thankfully, since 4D-STEM bone experiments have a short dwell time with an ultrafast camera (87,000 frames per second) that allowed us to mitigate this issue [11, 19]. However, this reduction comes with a price and introduces a trade-off: weaker probe current and low total electron counts have resulted in a low SNR and weak reflections, diffuse scattering, and making interpretation more difficult. Similar limitations have been pointed out in fibril-level studies because of the low electron flux, which led to limited reflections and reduced SNR [20].

Further limiting factors may arise from other parameters, such as preprocessing and calibration. Savitzky *et al.* especially put more emphasis on the requirement of a robust 4D-STEM analysis workflow like diffraction shifts, elliptical distortions, the rotational offset between real and diffraction space, and accurate pixel-size calibration [2]. This thesis did not fully employ all of these calibrations, but detector placements were refined manually. Detector shapes, positions, and threshold values were chosen manually; the resulting masked and reconstructed DPs should be evaluated as workflow-dependent rather than unique or detailed representations of the mineral phase. Moreover, the HAADF-based segmentation was done with a single slice using a limited training set, so manual inspection was applied.

3.5 AI usage statement

The output of the generative AI was used as a means of gaining a rapid overview of a topic or document and helped me identify which areas to focus on when reading. Also, it was used to support understanding of coding-related issues and for debugging. In the writing process, it was used to check and correct the grammar, spelling, and typographical errors. The final content, interpretations, analysis, and conclusions are my own, and no direct output was used.

4

Results and Discussion

This chapter includes the overall results of the detector-based 4D-STEM analysis of human bone and it follows the same order as shown in Figure 4.1: acquisition of the HAADF-STEM image, selection of the ROI, derivation of the mean DP from the ROI, detector placement on the DP, and reconstruction of the DF image from the off-axis detector in DP, application of the selected mask to DF image, implementation of positive and negative thresholds and obtaining DP image from positive and negative thresholded DF image respectively. Additionally, the detection of $\{002\}$ reflection is also studied because of the broad diffraction arc commonly associated with the apatite $\{002\}$ reflection. It is known from previous studies that bone mineral crystals have a preferential alignment parallel to the collagen fibril axis, and this orientation is represented by the c-axis that gives origin to the $\{002\}$ reflections in the DF, hence this is informative for orientation-sensitive DF analysis [8, 12, 14]. On the other hand, when it comes to the nearby stronger disk, it is a composite of several unresolved reflections, and because of this, the interpretation of the features is more difficult. For this unique reason, it is safe to say that in detector-based DF analysis, the $\{002\}$ plays an important role [12, 21].

4.1 Real-space overview and segmentation

Figure 4.2a shows the acquired HAADF-STEM image, and Figure 4.2b shows the segmentation work done in Dragonfly, and the latter was used for the real-space masking that includes bright bands, dark bands, and extra-fibrillar masks for diffraction analysis. This first step is the basis for the detector-based workflow, and the HAADF-STEM image provides the overall structure of the bone visually.

4.2 DP of the selected ROI

In this step, the mean DP (Figure 4.4) is obtained from the selected ROI in the HAADF-STEM image. The importance of this step is that it represents all of the scattering electrons coming from the selected ROI, and it provides an overall idea of how the $\{002\}$ reflections appear as a pair of arcs instead of forming a complete ring. This stems from the fact that c-axes of the apatite crystals have preferred orientations parallel to the fibrils, which is the long axis of bones [22].

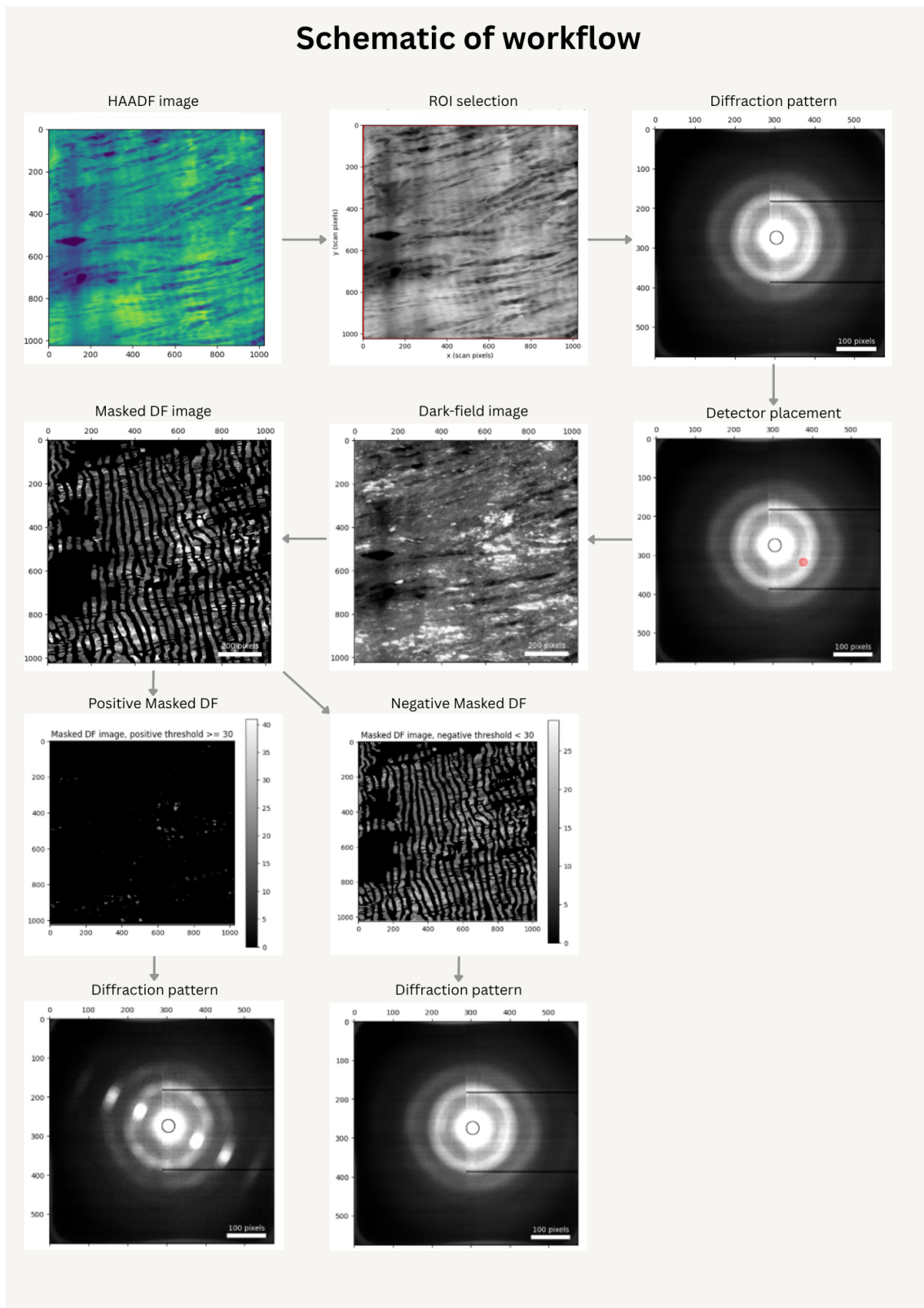


Figure 4.1: Schematic of the 4D-STEM analysis workflow. The workflow begins with a simultaneously acquired HAADF-STEM image and ends with DPs of both positive and negative thresholded masked DF images. In this specific example, the bright band mask is used in the masked DF image.

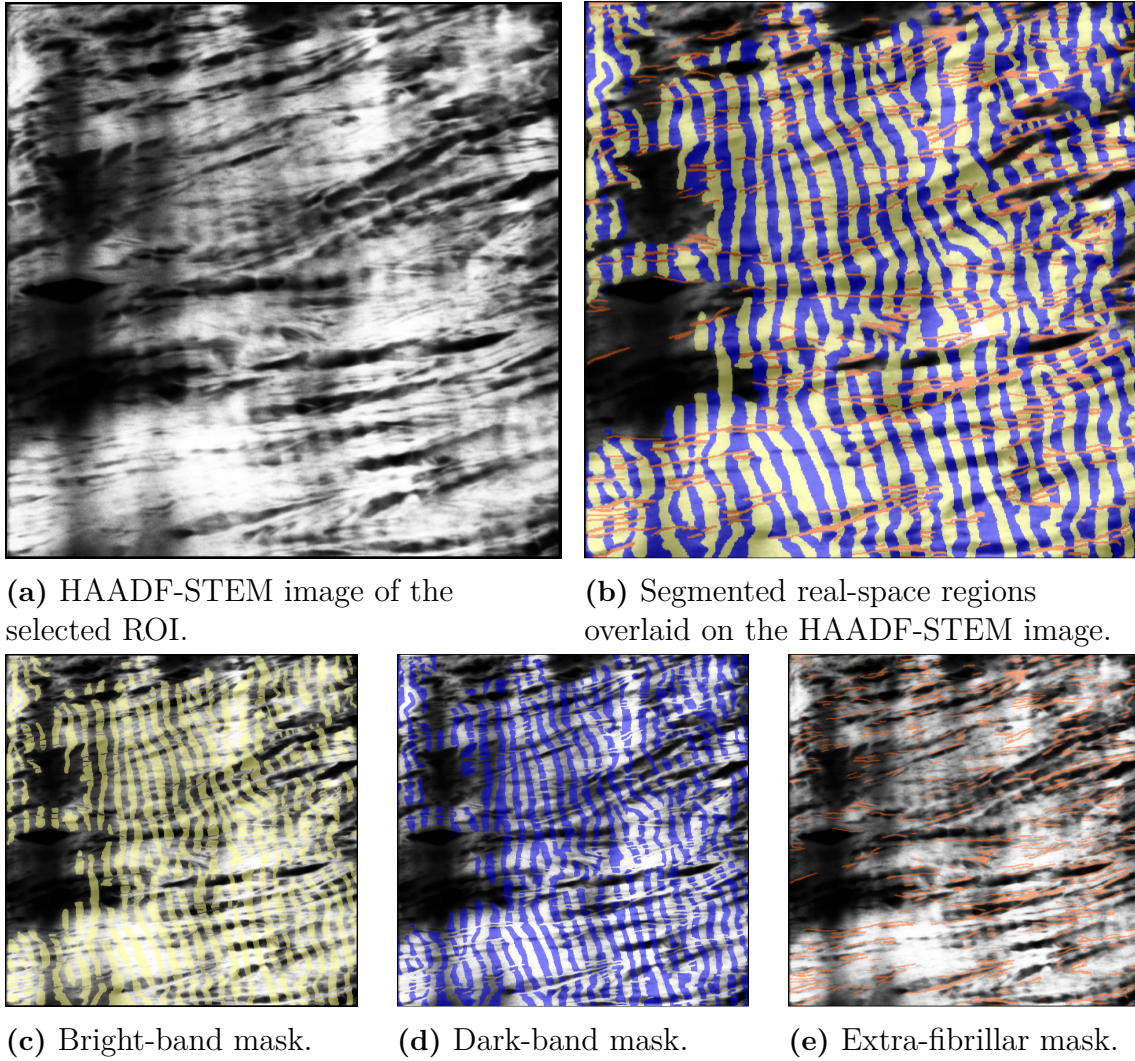


Figure 4.2: Real-space overview of the selected ROI (a) and the segmentation (b) used for masked analysis. The segmentation defines the structural regions used in the subsequent detector-based diffraction workflow. Individual Dragonfly segmentation masks used to define the main real-space classes in the HAADF-STEM image. The masks correspond to bright-band regions (c), dark-band regions (d), and extra-fibrillar mineral (e), respectively.

4.2.1 Masked real-space and resulting DP

Figure 4.3 shows the comparison of the DP of each mask over the full ROI. As seen from the DPs, each of them looks quite similar, and their histograms are slightly different, so it is really difficult to tell the differences between them. Therefore, this motivates the need for other approaches like the one this thesis employed: the workflow that uses threshold analysis. Another contributing factor to why these DPs are indistinguishable is the sample thickness, which causes unwanted scattering from deeper layers to be captured by the probe, and this makes the rings wider and more diffuse. However, with threshold analysis and a smaller ROI, this contribution can be minimized.

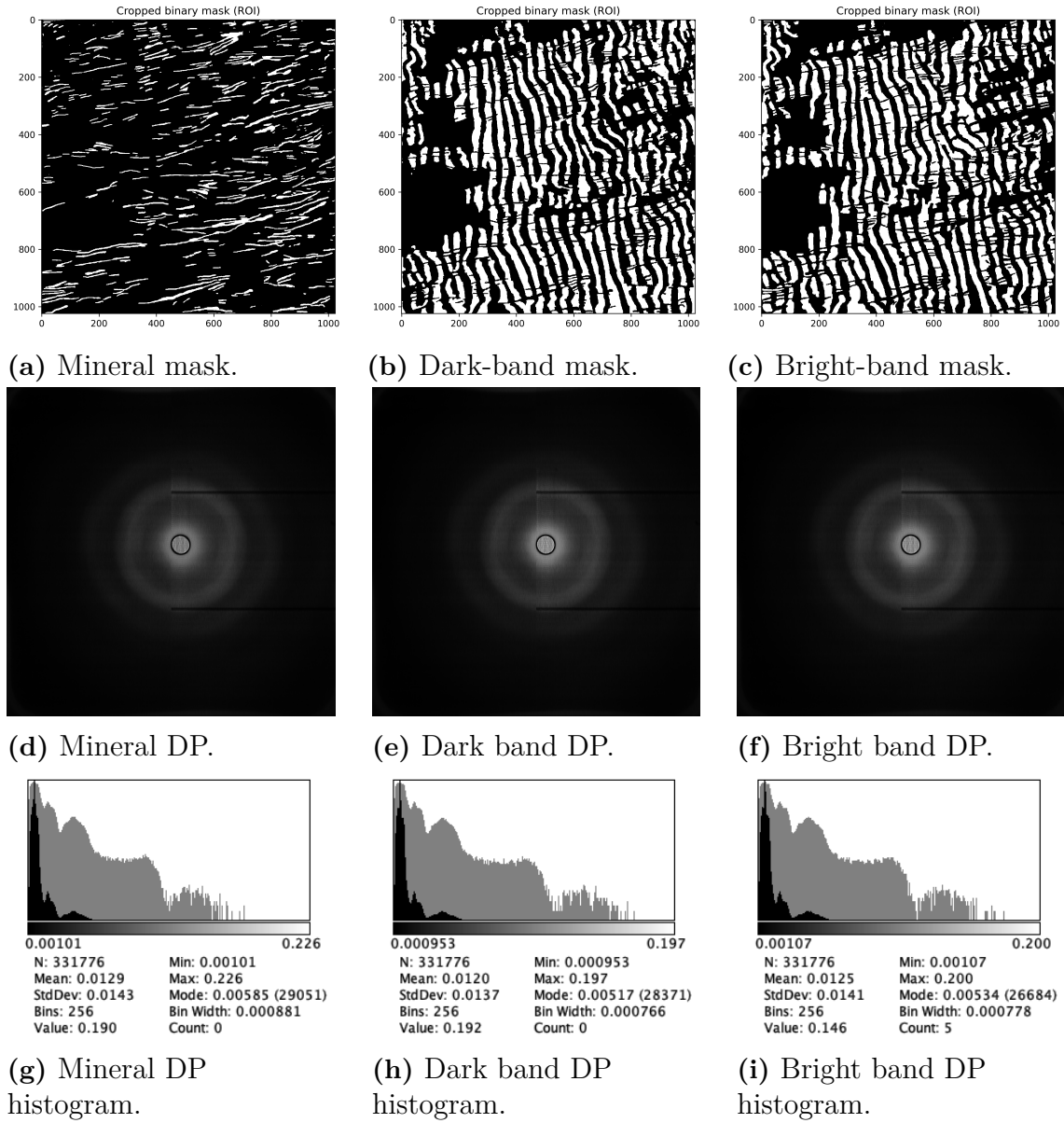


Figure 4.3: Binary masks of mineral (a), dark-band (b), and bright-band (c). DPs of mineral (d), dark-band (e), and bright-band masks (f). Mineral DP histogram (g), dark-band DP histogram (h), and bright-band DP histogram (i).

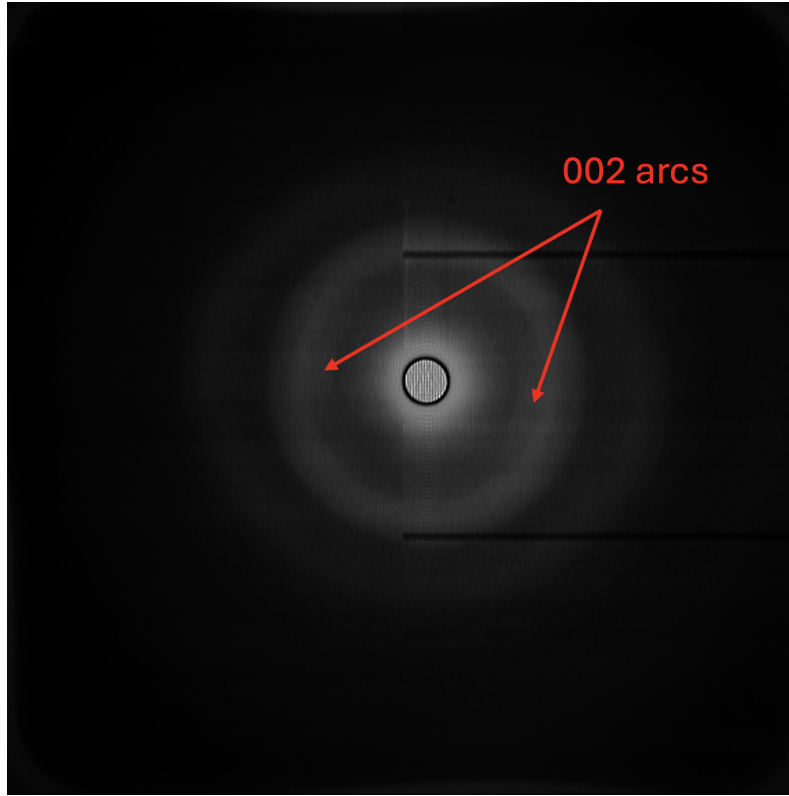


Figure 4.4: Mean DP obtained by averaging all DPs within the selected ROI. The pattern is dominated by the central transmitted beam and broad concentric scattering rings, and it serves as the reference for beam-center determination and detector placement in the subsequent analyses. 002 diffraction arcs are indicated by two arrows.

4.3 $\{002\}$ Arc detection

As shown in Figure 4.5, at first, a radial profile from the mean DP is formed, and this figure shows where the main ring intensity lies. After the high-fluctuation region, every peak indicates a ring intensity in DP, and the first peak region (red rectangle) shows the outer and inner radius of the annulus. After defining an annulus, the ring angular intensity profile is formed from the sector inside the annulus, clockwise. This figure indicates where along that ring the scattering is strongest. This is the systematic way of placing circular detectors based on the mean intensity in the ring sector.

4.4 Off-axis DF imaging

Following the $\{002\}$ arc detection, one off-axis circular disk detector was placed along the $\{002\}$ arc where the scattering was strongest, as shown in Figure 4.6a. The red disk detector captures the signal of scattered electrons by isolating a restricted reciprocal-space contribution, and since the diameter of the detector was

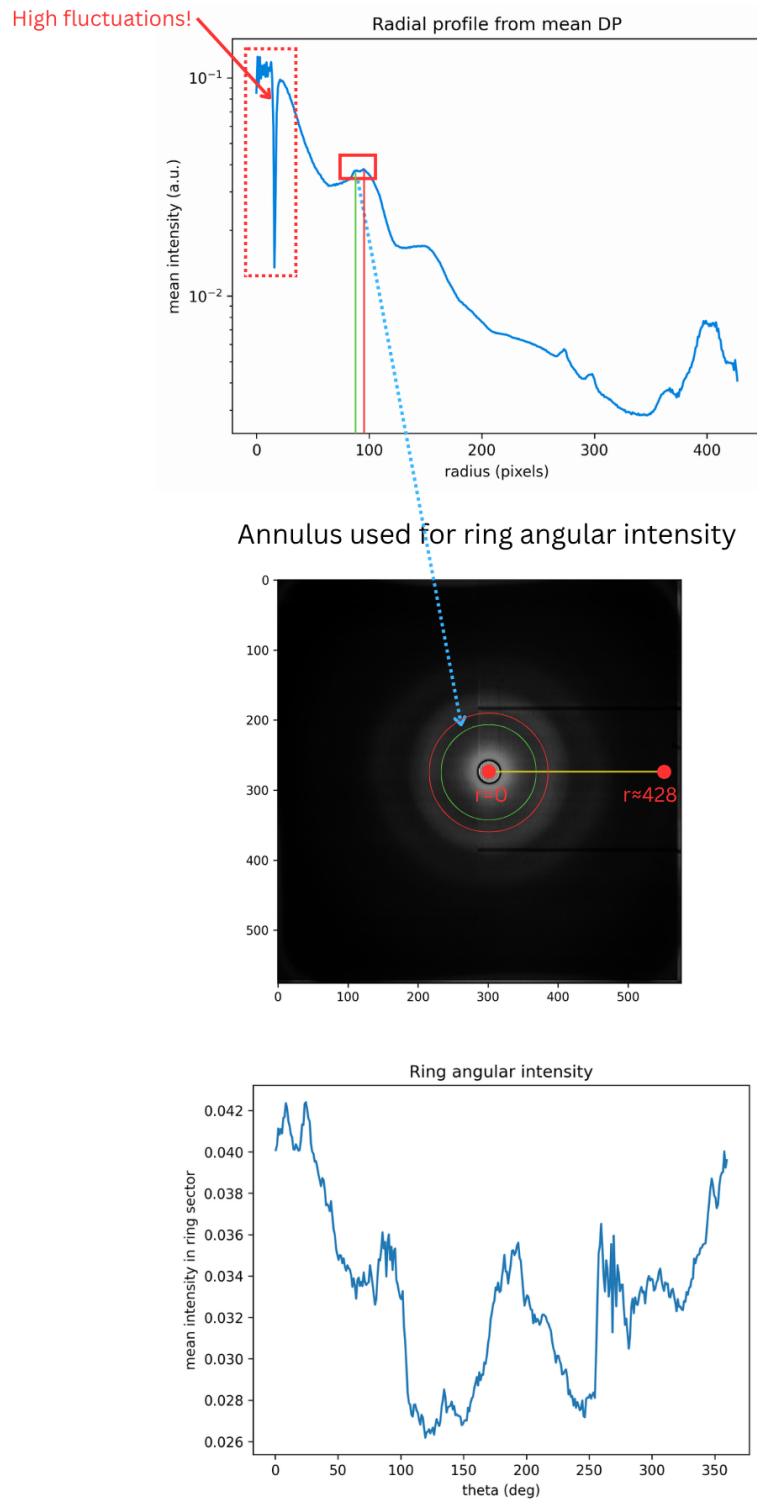
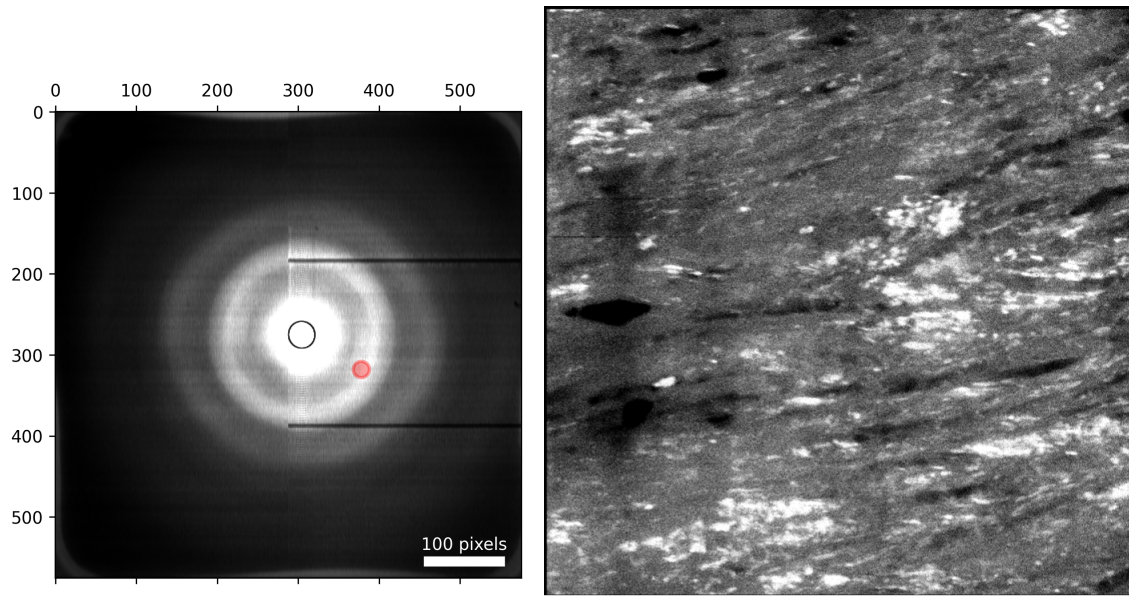


Figure 4.5: Workflow for identifying the broad $\{002\}$ -associated arc using radial and angular analysis of the ROI-averaged DP.

relatively small, the scattered electrons had similar scattering angles, and the orientations of the crystals had similar c-axis angles. This resulted in the DF image, which is shown in Figure 4.6b. The DF image shows the similarly oriented apatite crystals as bright spots. These localized bright patches are the regions that have the highest contributions to the selected reciprocal-space sector. When the thickness of the sample is accounted for, it can be argued that they are not the direct images of isolated crystals because the signals of crystals in the deeper layers of the bone sample could contribute to the DP, and thus they may appear as localized patches of apatite crystals in the resulting DF image. It can also be noticed that the characteristic banding pattern visible in the HAADF-STEM image is not present in the DF images. [11, 12, 21]. However, these banding patterns can be presented in the DF image by using the masks derived from the HAADF-STEM image.



(a) Placement of the off-axis circular detector on the ROI-averaged DP.

(b) Resulting off-axis DF image in real space. Bright regions correspond to pixels whose DPs contribute strongly to the selected reciprocal-space sector.

Figure 4.6: Off-axis DF imaging of the selected ROI. A circular detector was placed on a localized sector of the broad diffraction ring, and the corresponding DF image was computed from the 4D-STEM datacube. The resulting image highlights discrete real-space patches associated with the selected diffraction sector.

4.5 Masked off-axis DF image

The off-axis dark-field image presented in 4.4 was masked with the bright-band binary mask defined from the Dragonfly segmentation work shown in Figure 4.2c. In this way, the banding pattern, which is normally absent in the DF image, can be represented. Every pixel inside the "1" region of the binary mask survives, and they retain their original intensity. On the other hand, the rest of the pixels outside of

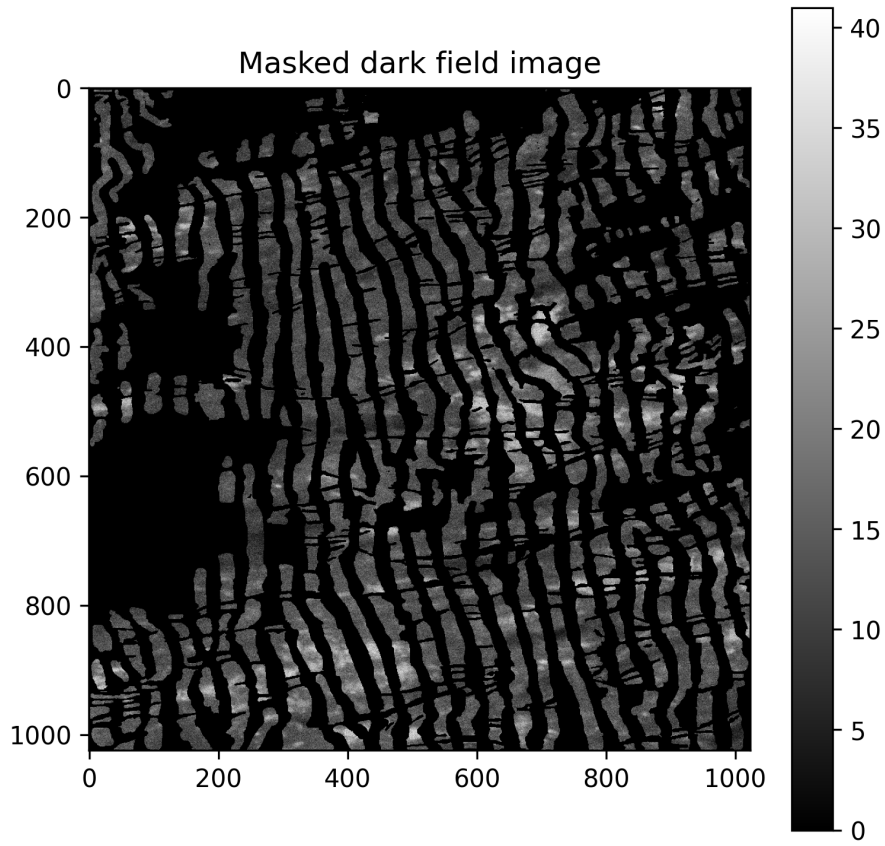


Figure 4.7: Masked off-axis DF image obtained by applying the bright-band binary mask from Figure 4.2c to the off-axis detector response. The image retains the overall banded morphology of the selected bright-band region while showing non-uniform intensity along the masked structure.

the banded region were blocked with "0" values in the binary mask, and they appear as a black region. This approach makes it possible to study the crystal apatites inside these banded regions. It should be noted that, as argued by Schwarcz *et al.*, DF images should not be treated as ground truth to prove that the diffracting mineral is physically located inside those specific gap or overlap zones, because it is argued that extra-fibrillar crystals may also appear in the zones that are over gap zones in projection [12]. The resulting masked DF image is shown in Figure 4.7, and the bright band mask is used for the representation because the end result, which is the thresholded DP, looks quite similar in every mask, and for the purpose of presenting the results, the bright band mask is chosen in this work. It can be said that the masked image represents the overall banded morphology, but the intensity of these regions is not distributed uniformly, and this leads to the observation of brighter and darker segments. This is an indication of the selected off-axis detector capturing a spatially heterogeneous response in real space. This result shows that the reciprocal-space disk detector does not capture the signal equally for all pixels within the bright-band mask. Moreover, the detector is able to isolate a subset of the segmented bright-band structure in which the contribution is the strongest to that azimuthal annulus of the DP. Therefore, the bright-band masked off-axis DF image can produce a real-space map of the detector-selected responses inside those bright-band regions and is the basis for the thresholded analysis described below.

4.6 Threshold-based analysis

As discussed in the previous section, the off-axis masked DF image (Figure 4.7) has strongly and weakly appearing pixels inside the banded region, and this threshold-based analysis aims to separate these pixels. This approach separates the DF image into positive and negative thresholded DF images, and the applied threshold value was 30 based on the pixel intensity. This value can be chosen arbitrarily depending on the maximum signal intensity present in the masked DF image. Each resulting thresholded DF image was exclusive to itself, meaning surviving pixels were only present in the subjected thresholded DF image, and positive-threshold pixels are greater than or equal to the threshold, and negative-threshold pixels are those less than the threshold.

Another important point about the threshold analysis is that the aggressive thresholding approach on the positive threshold enables testing a sparse subset of the strongest detector-selected pixels, and with less aggressive threshold values, it makes it possible to test the expected broader fraction of the masked structural pixels.

4.6.1 Thresholded regions

4.6.1.1 Positive-threshold regions

The positive thresholded image is shown in 4.8a. The threshold value was chosen as 30 in the previous section, and this can be interpreted as aggressive thresholding. This can be done when only strongly aligned mineralized patches are to be observed, because these are the regions where the strongest signals were captured in

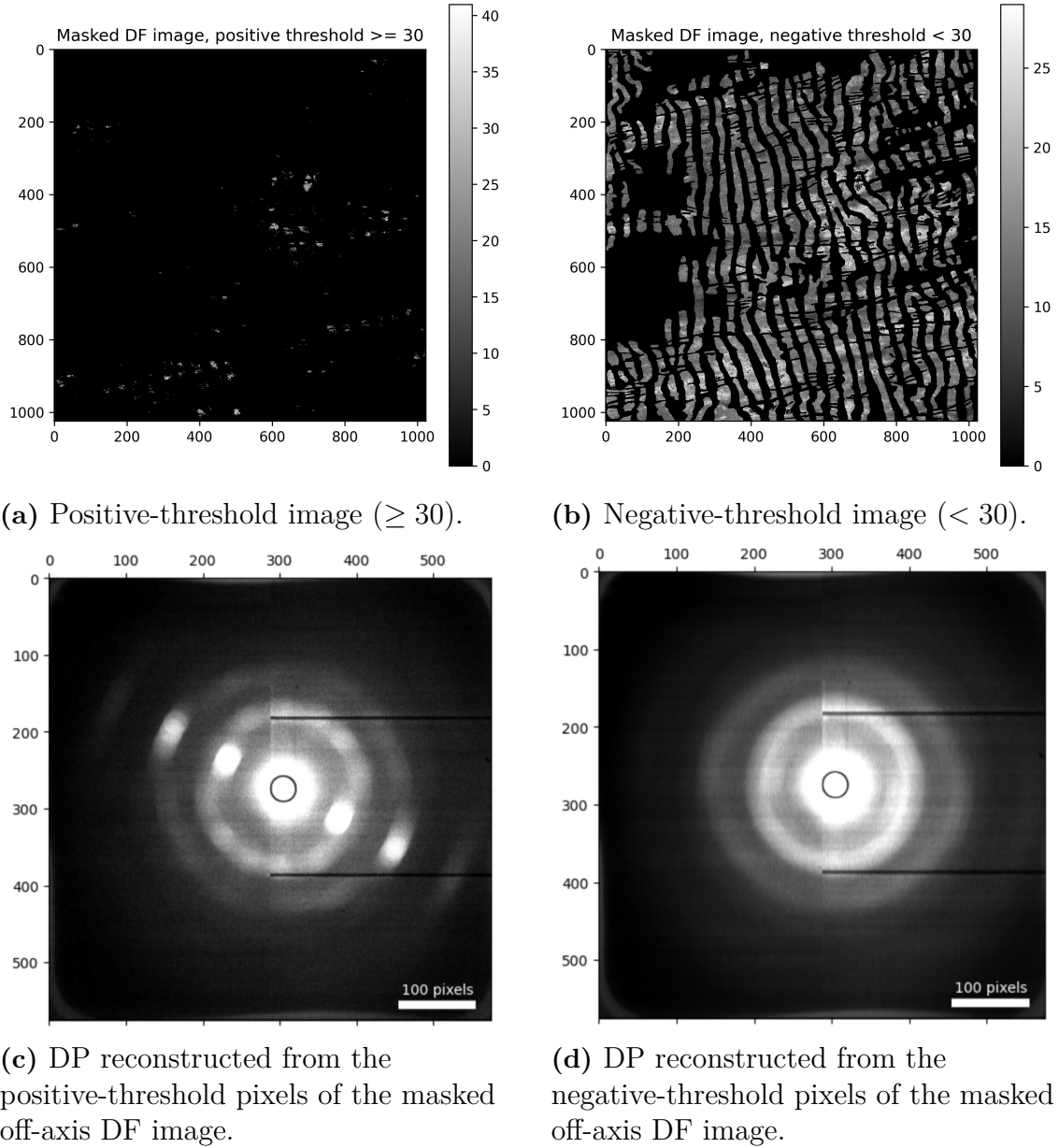


Figure 4.8: Thresholded versions of the masked off-axis DF images ((a) and (b)) derived from the bright-band mask shown in Figure 4.2c. The positive-threshold image (a) isolates the strongest detector-response regions, while the negative-threshold image (b) retains the weaker-intensity component of the masked bright-band structure. DPs ((c) and (d)) reconstructed from threshold-selected pixels within the bright-band mask. The positive-threshold reconstruction (c) shows localized bright maxima on the diffraction ring, whereas the negative-threshold reconstruction (d) retains the same overall ring geometry but with a more diffuse intensity distribution.

the off-axis diffraction sector. This positive threshold image is the indicator of the highest detector response, where it is concentrated in a limited subset of the masked bright-band mask structure, and the distribution of these mineralized structures is nonuniform across the full banded region.

4.6.1.2 Negative-threshold regions

The negative-threshold image is shown in Figure 4.8b. When the aggressive threshold value was applied for positive threshold regions, unaggressive threshold values resulted in negative-threshold regions, and the resulting thresholded masked DF image populated much of the bright-band morphology, where only the strongest hotspots were excluded. Meaning of this is that the contribution of the large fraction of the bright-band mask comes from weakly selected signals in the selected off-axis detector. Taken into consideration together, this indicates the bright-band mask pixel population is strongly non-uniform and can be separated into two classes consisting of a sparse high-intensity subpopulation and a more widespread low-intensity background.

4.6.1.3 DPs reconstructed from threshold-selected pixels

When the positive- and negative-threshold masked DF images obtained in previous steps were applied to the datacube in the 4D-STEM database, the DP of each DF image was reconstructed and shown in Figure 4.8c and 4.8d. The positive threshold of DP displays bright localized maxima on the broad diffraction arc. This suggests that a more anisotropic scattering focused in certain azimuthal sectors comes from the regions where the DF response is strongest. In contrast, the DP of the negative-threshold DP captures the signals from relatively wide areas and various angles of weakly scattering electrons; this makes the DP display a similar broad ring geometry as in the original DP and more diffuse intensity. As discussed in Savitzky *et al.*, this is a consistent behavior in weaker diffracting material than a fully distinct crystalline subpopulation [2]. However, the present data by no means prove the possibility of a separate amorphous phase, because complete or diffuse-like ring geometries in bone may also stem from the fact that the presence of poorly ordered or nanocrystalline apatite may influence scattering, and moreover, a true discrete amorphous solid phase in mammalian bone is yet to be discovered and remains difficult to establish [6].

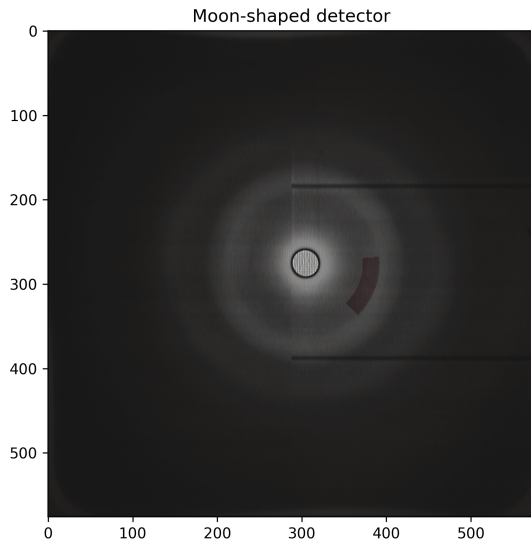
This interpretation is parallel to the recent electron tomography work where the intra-fibrillar bone mineral is inspected and described as small stacks of apatite nanoplatelets, which have a predominantly uniaxial orientation as opposed to a perfect co-aligned deck-of-cards arrangement. That framework shows the wide {002} diffraction arc having the same preferential orientation of the apatite c-axis along the axis of the collagen fibril. Therefore, the present relatively more localized maxima in the positive-threshold DP display consistency with a subset of more strongly aligned crystalline regions, and in the case of the negative-threshold pattern that shows a more diffuse ring that displays weaker angular localization and local nanoscale disorder in overall apatite-based mineral populations, it lacks direct evidence for a separate amorphous phase [23].

Importantly, it can be observed that the two DPs have the same overall main position, suggesting the possibility that the positive- and negative-threshold pixels are influenced by roughly the same scattering family, rather than being two different diffraction classes. In light of this observation, the thresholding can separate subpopulations in a way that they mostly differ in the strength and angular localization of their participation in the selected diffraction space. It has been shown that off-axis detector-selected DF images reveal localized bright patches other than a spatially uniform response, and the present study's results are in line with this [12, 21]. This is an expected behavior as explained by Schwarcz *et al.*, who claim strongly highlighted DF patches are the representation of well-diffracting apatite crystals related to mineral structures that are found around collagen fibrils instead of being crystals dwelling in gap zones [12, 21]. All things considered, this study is in line with the view that the strongest diffraction signals come from heterogeneous mineralized regions. However, as pointed out by Schwarcz *et al.*, all in all, it is not true to expect DF images to prove that the diffracting mineral is physically located in the anticipated gap or overlap zone, because of the thickness of the sample, there may be a deeper layer contributing to the scattering. Also, other accounted and unaccounted factors, such as the possibility of secondary unexpected scattered electrons, can influence DP; therefore, extra-fibrillar crystals may appear over gap zones in projection [12].

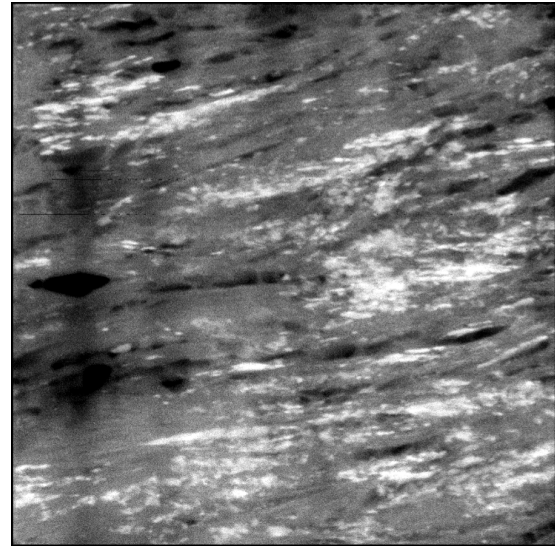
4.7 Alternative detector geometry: moon-shaped detector

To test the different detector shapes, a crescent- or moon-shaped virtual detector was created manually and placed on an azimuthal sector where the main intensity of $\{002\}$ arc lies based on the ring angular intensity profile in Figure 4.5. When compared to the single-disk detector, this geometry integrates intensity over a relatively wider angular range and therefore captures more scattering signals, including a wider range of oriented crystals. The resulting DF image is shown in Figure 4.9(b). The image shows localized bright patches that are distributed non-uniformly, and this indicates the selected reciprocal-space region has information about a heterogeneous real-space mineral orientation.

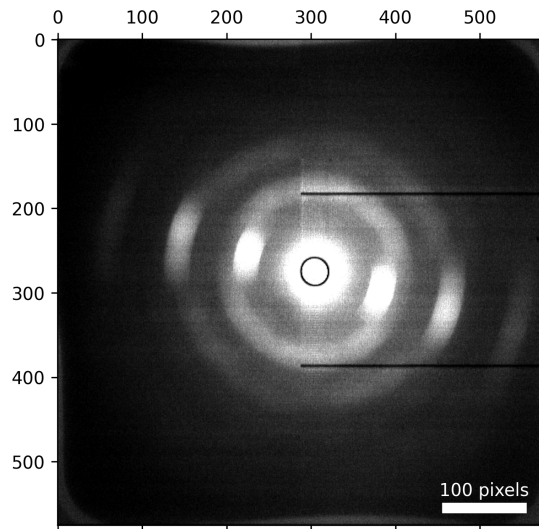
The rest of the results of the workflow are similar to those in single-disc detector-based geometry analysis, and the interpretation is the same as discussed in single-disc detector-based analysis. Only, the results of DPs reconstructed from threshold-selected pixels are different, and they are shared in Figures 4.9c and 4.9d. As seen in this figure, the positive thresholded pixels produced wider Bragg-like disk features because of the wider detector geometry in DP. It highlighted more pixels in the DF image (Figure 4.9b), which led to revealing more orientation variations in the mineralized regions.



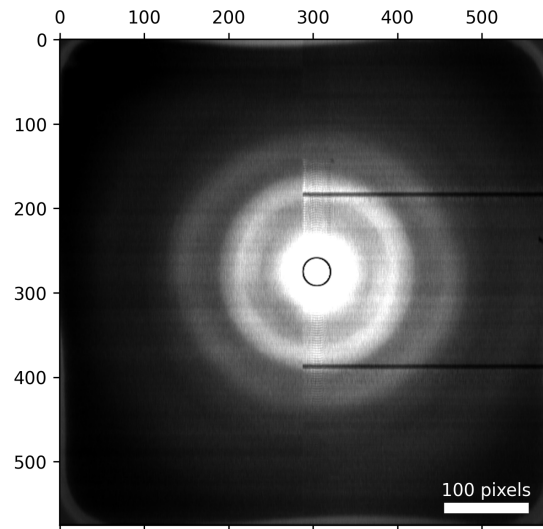
(a) Placement of the moon-shaped detector on the ROI-averaged DP.



(b) DF image obtained using the moon-shaped detector. Bright regions correspond to pixels contributing strongly to the selected reciprocal-space sector.



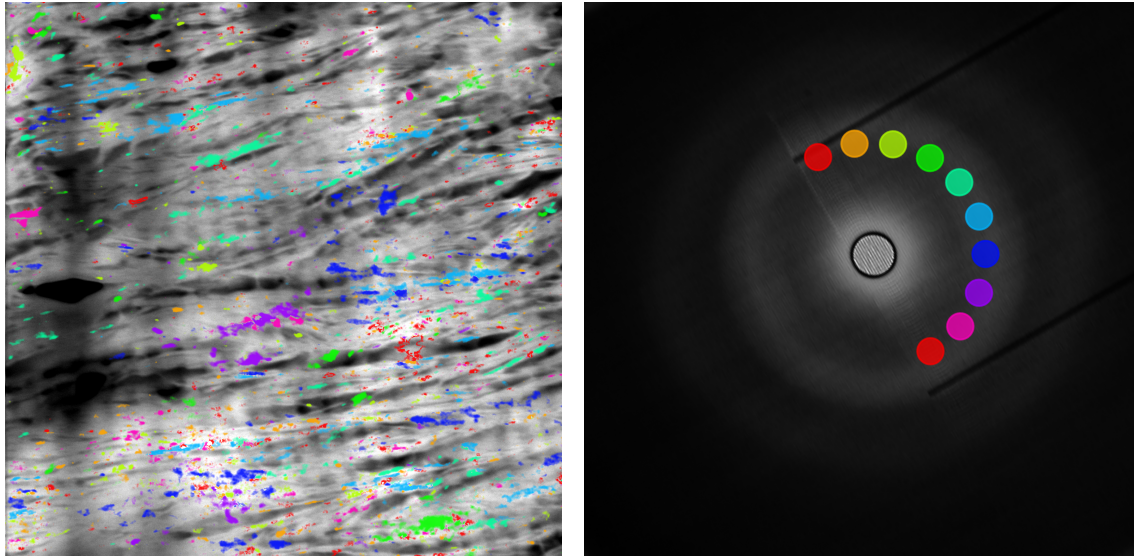
(c) DP reconstructed from the positive-threshold pixels of the masked DF image.



(d) DP reconstructed from the negative-threshold pixels of the masked DF image.

Figure 4.9: DF imaging using a moon-shaped detector geometry placed on a broad sector of the diffraction ring (a). If interpreted as sampling part of the $\{002\}$ diffraction arc, the resulting DF image (b) highlights regions contributing strongly to that sector of reciprocal space. DPs reconstructed from threshold-selected pixels within the dark-band mask. The positive-threshold reconstruction emphasizes localized azimuthal intensity maxima on the diffraction ring, whereas the negative-threshold reconstruction retains the same overall ring geometry but with a more diffuse intensity distribution.

4.8 Orientation of the c-axis in multiple-detector analysis



(a) Color-coded DF overlay on the HAADF-STEM image obtained from multiple off-axis circular detectors placed along the diffraction arc. Localized coloured patches indicate regions that contribute most strongly to different detector positions.

(b) Detector positions on the full ROI-averaged DP. The detectors were distributed along the broad diffraction arc; the first and last detector positions were assigned the same color to emphasize their approximate symmetry about the diffraction center.

Figure 4.10: Multiple-detector analysis of the selected ROI. A set of off-axis circular detectors was placed along the broad diffraction arc visible in the ROI-averaged DP, consistent with the approximately 30° -wide arc reported for the $\{002\}$ reflection of bone mineral in samples with collagen fibrils oriented in-plane [14]. The resulting color-coded DF overlay reveals localized patches of similarly oriented crystals, with many of these patches appearing in inter-fibrillar regions.

To investigate the orientation of the c-axis, multiple disks were placed along the $\{002\}$ reflection in the DP. In order to represent the direction of the mineral growth (along the c-axis), the DP in Figure 4.10b is rotated 30° . However, the exact value of the rotation angle is yet to be determined; this is just for the demonstration. Figure 4.10b shows the corresponding disk detectors' positions in reciprocal space, and each disk produces its own DF image. In DF images, minerals appear as bright spots, and each disk was assigned to a color. When these color-coded DF images were combined and overlaid on HAADF-STEM images, the resulting color-coded HAADF-STEM image displays localized patches of similarly oriented crystals, as shown in Figure 4.11 and 4.10a. It is reported that when the bone minerals display $\{002\}$ arc, it is approximately 30° -wide for the samples that were viewed in-plane [14]. In this color-coded DP, the disks symmetric to the center were painted the same color because they capture electrons that are scattered with relatively similar

angles, and the small variations in symmetric disks were taken into consideration; thus, only half of the DP was used.

DF overlay image shows that the signals of the highlighted regions come from different detector positions in reciprocal space, and it shows a spatially non-uniform response in real-space, indicating that the intensity along the diffraction $\{002\}$ arc is spatially heterogeneous. It is also observed that the signals that populate the HAADF-STEM images are the ones coming from the region along the $\{002\}$ arc where the signal is strongest, as expected [14]. It is also true that the neighboring detectors produced partially overlapping responses in real space, which suggests the scattering intensity is not confined to discrete orientations. The multiple detector analysis can reveal information about the rotation of the c-axis, thus can help determine a set of related, but not identical, crystallographic organizations within the reflected ROI. Figure 4.11 also shows selected examples of resulting DF images in grayscale.

4.9 Additional results for specific ROIs

Figure 4.12 shows the workflow of a small ROI. Inside this ROI, a dark region was selected to verify whether it corresponded to a vacuum. Note the difference between the DP of the masked ROI and the DP of the selected ROI. In this case, since ROI is smaller and the binary mask has fewer pixels, the resulting DPs are distinguishable, and the DP of the masked ROI displays more interpretable features. Also note that this darker region in the HAADF image was not expected to include mineral formation, but when inspected with the masked DF image, it displayed bright spots of a mineralized region. Of course, thickness may be affecting the scattering, and this is subject to future work.

Other selective results are shown in Figures 4.13 and A.2. We observed that when we inspect a smaller ROI, the SNR is worse, but also clearer patterns emerge.

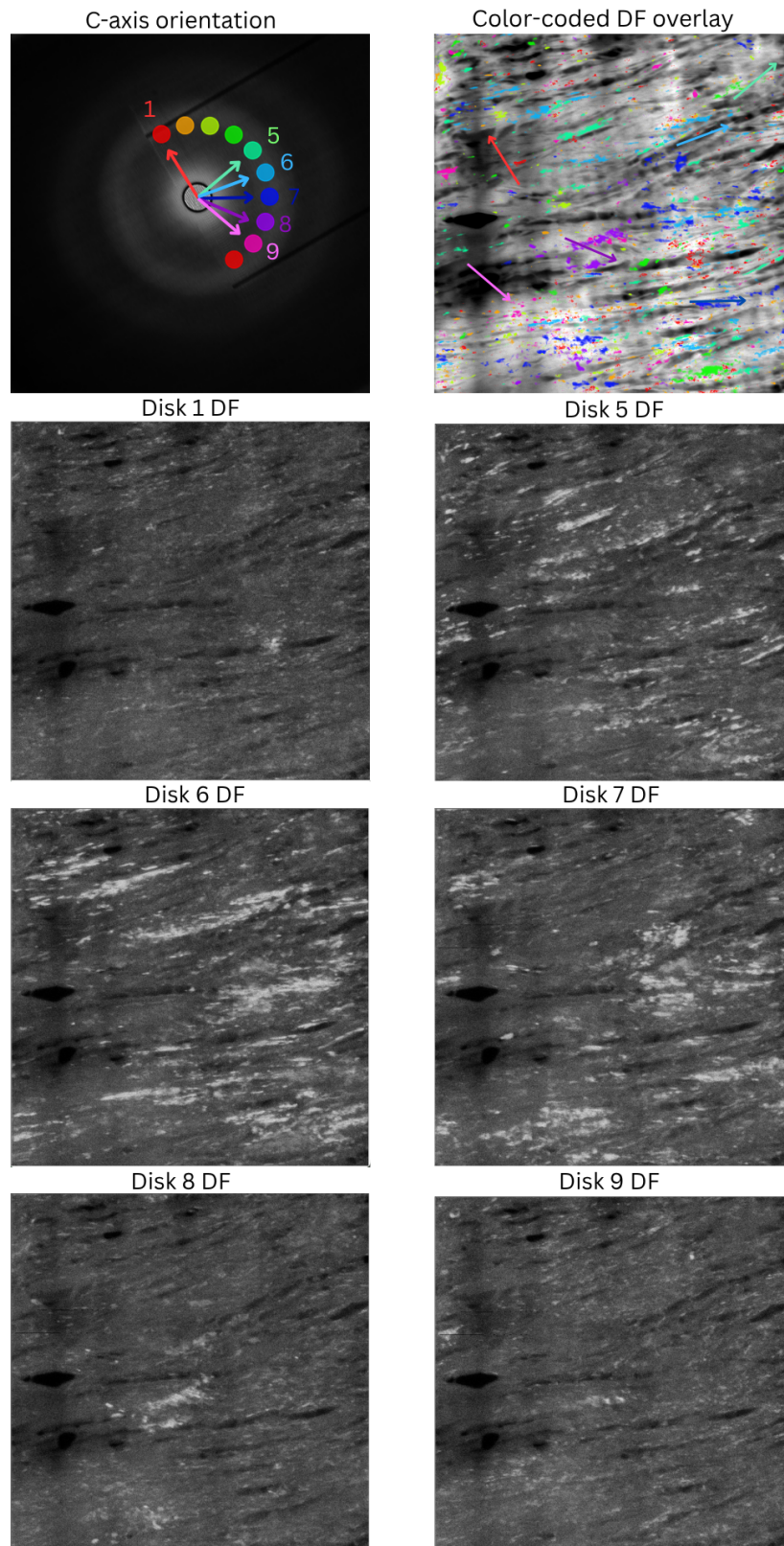


Figure 4.11: Orientation of the c-axis of the selected disks and their resulting DF images. Colored arrows show the preferred direction of minerals that grow along the c-axis.

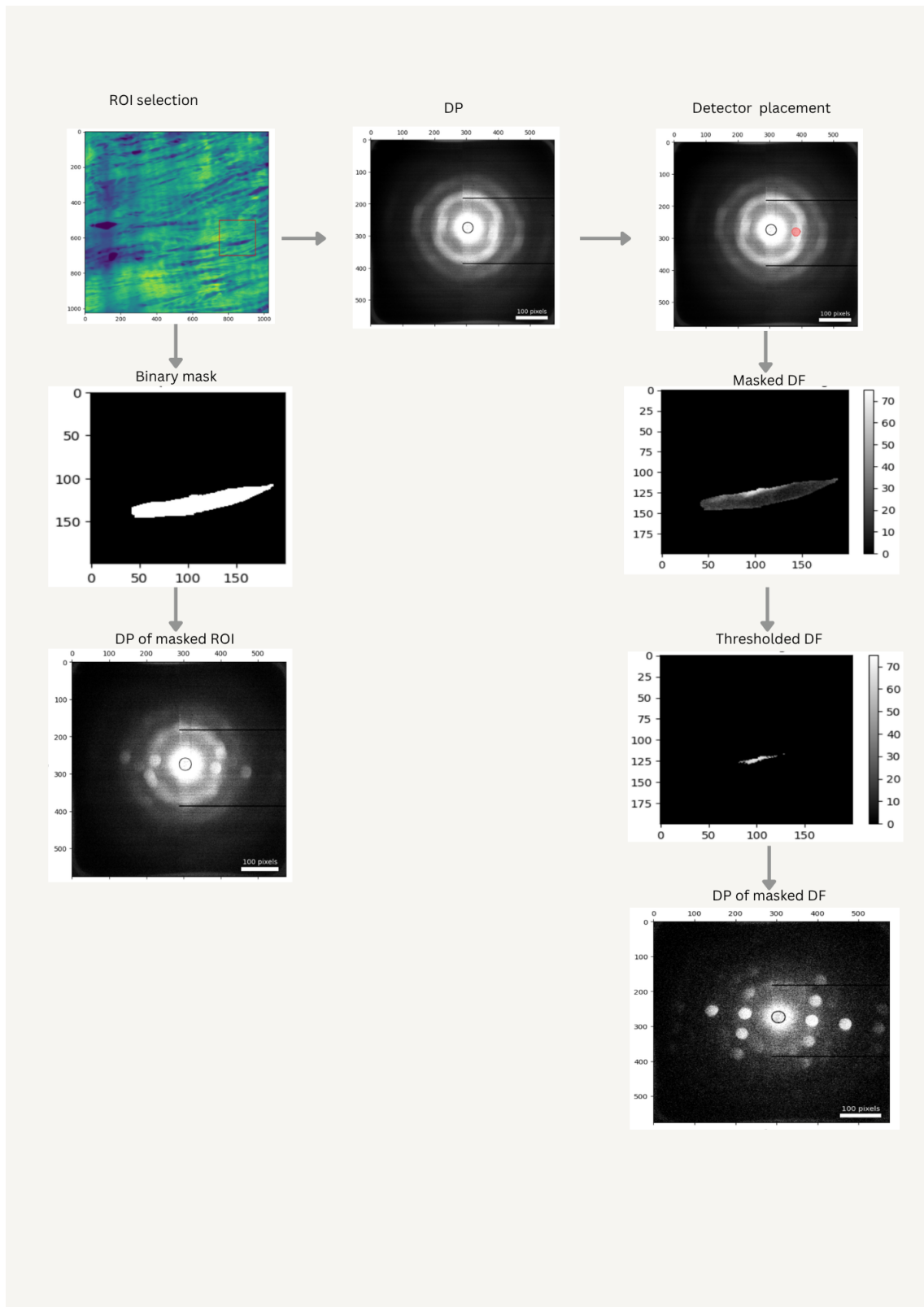


Figure 4.12: Schematic of the 4D-STEM analysis workflow in a smaller selected ROI. The binary mask is the darker region inside the ROI. The threshold value is 50 in the thresholded DF image.

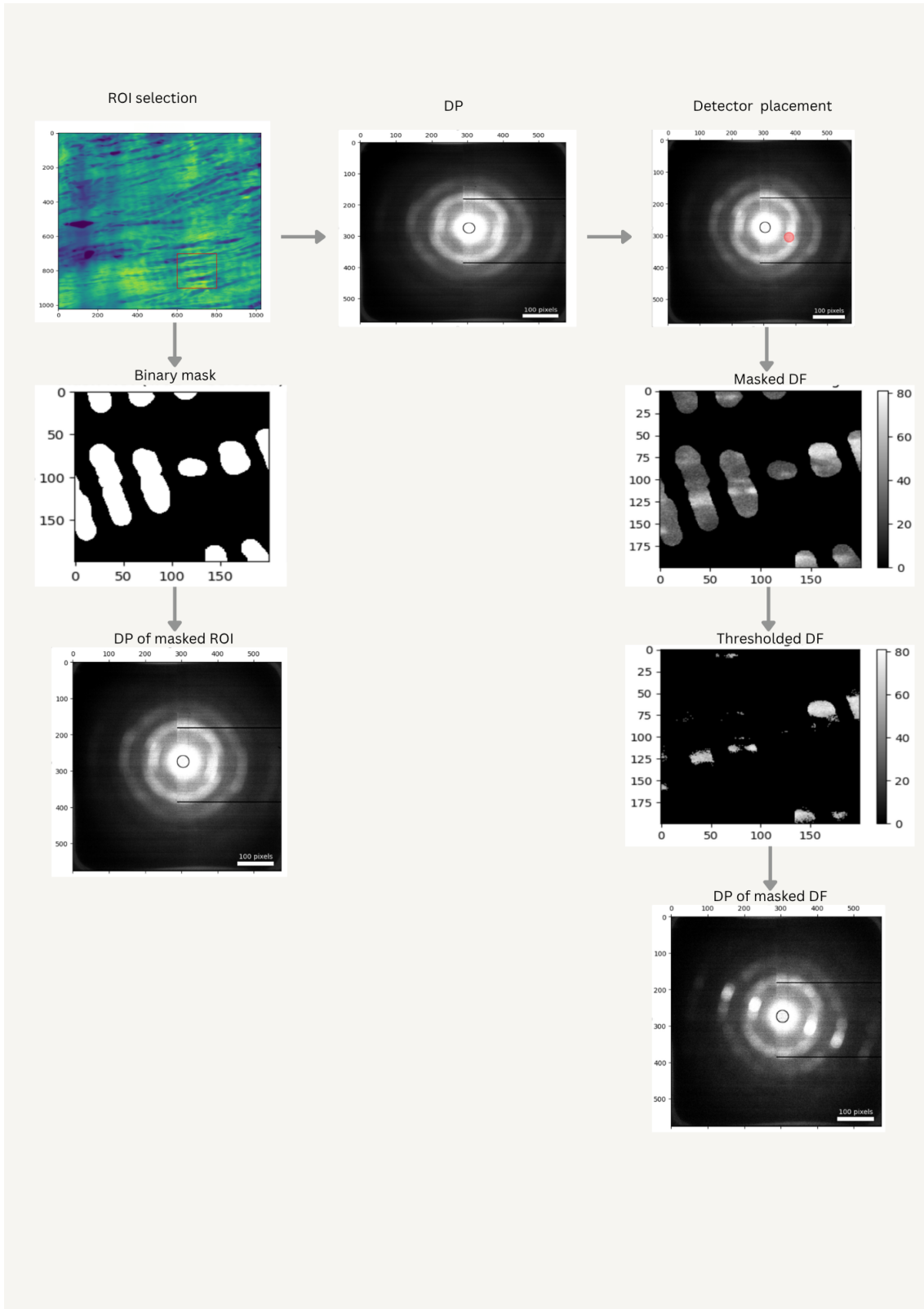


Figure 4.13: Schematic of the 4D-STEM analysis workflow in a smaller selected ROI. The binary mask is the bright-band inside the ROI. The threshold value is 50 in the thresholded DF image.

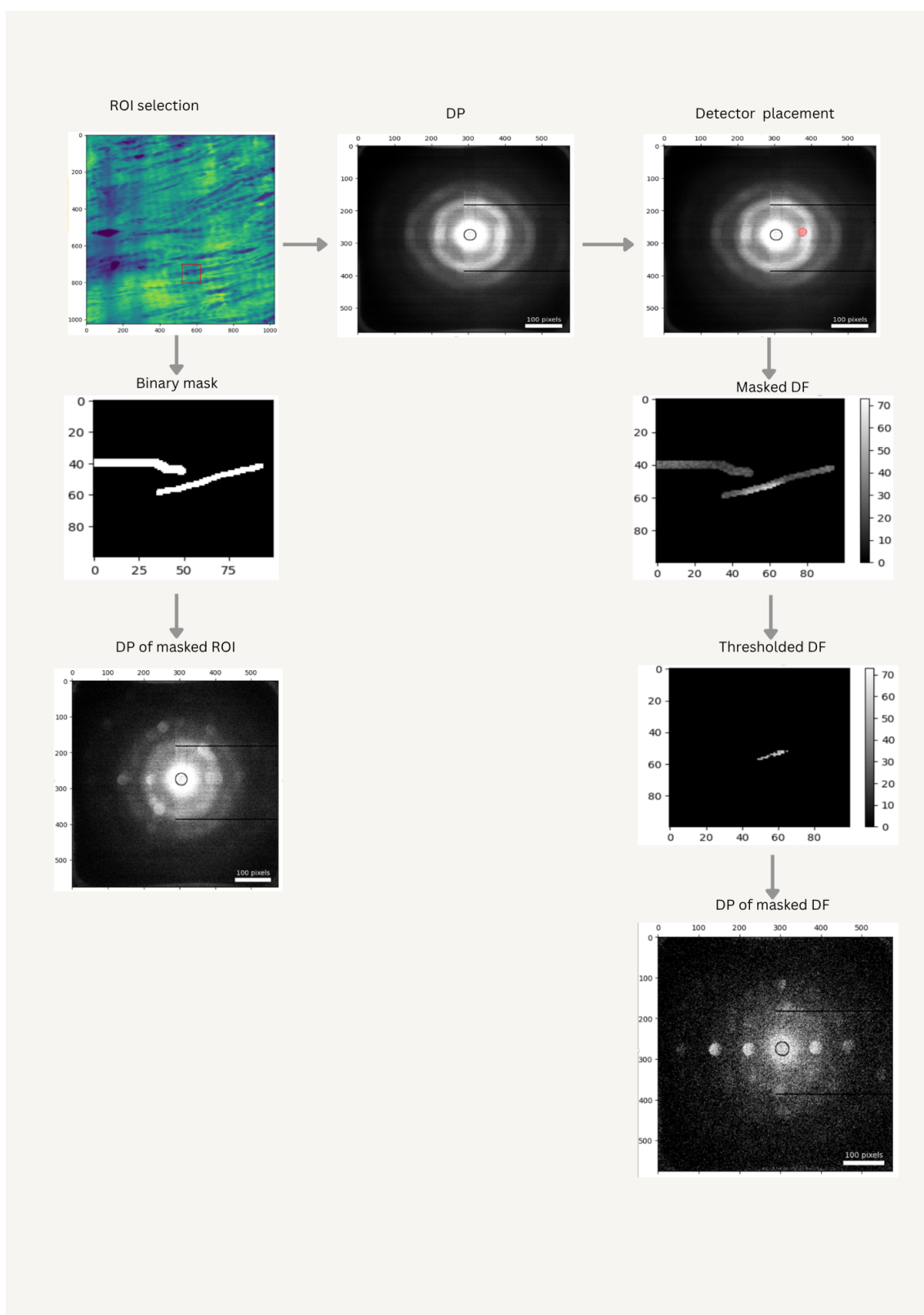


Figure 4.14: Schematic of the 4D-STEM analysis workflow in a smaller selected ROI. The binary mask is the extra-fibrillar mineral inside the ROI. The threshold value is 50 in the thresholded DF image.

5

Conclusion and outlook

This thesis studied the nanoscale crystallographic organization of human cortical bone using 4D-STEM with the main aim of developing and applying a workflow that consists of local real-space structures, selected diffraction-space features, mean DPs, virtual DF imaging, HAADF-based segmentation, and thresholding analysis. Real-space binary masking was able to reveal similarly oriented crystals with relatively broad 002 arc, but with further refinement using DF imaging, real-space masking, and threshold analysis, the local differences were analyzed extensively and resulted in a relatively less broad 002 arc that showed Bragg-like disk features. Overall, extra-fibrillar minerals were not differentiated from banded regions, meaning DF images should not be expected to distinguish the mineral physically found in the gap and overlap zones, because of the thickness of the sample and bone heterogeneity, there might be other contributing deeper layers to the scattering.

5.1 Future research

The possible future work can include more datasets available from the 4D-STEM experiment by increasing the number of ROIs and applying the workflow to additional datasets to test its reproducibility. With these additional datasets, the deep-learning algorithm will be trained more efficiently, and segmentation will be further refined based on the information of the c-axis rotation. Also, other analysis modes available in py4DSTEM, such as structural classification, can be applied alongside virtual imaging, which is the only mode explored in this thesis.

Bibliography

- [1] Colin Ophus. “Four-Dimensional Scanning Transmission Electron Microscopy (4D-STEM): From Scanning Nanodiffraction to Ptychography and Beyond”. In: *Microscopy and Microanalysis* 25.3 (2019), pp. 563–582. DOI: 10.1017/S1431927619000497.
- [2] Benjamin H. Savitzky et al. “py4DSTEM: A Software Package for Four-Dimensional Scanning Transmission Electron Microscopy Data Analysis”. In: *Microscopy and Microanalysis* 27 (2021), pp. 712–743. DOI: 10.1017/S1431927621000477.
- [3] Furqan A. Shah, Peter Thomsen, and Anders Palmquist. “Osseointegration and Current Interpretations of the Bone-Implant Interface”. In: *Acta Biomaterialia* 84 (Jan. 2019), pp. 1–15. DOI: 10.1016/j.actbio.2018.11.018.
- [4] J. Y. Rho, L. Kuhn-Spearing, and P. Zioupos. “Mechanical properties and the hierarchical structure of bone”. In: *Medical Engineering & Physics* 20.2 (Mar. 1998), pp. 92–102. DOI: 10.1016/S1350-4533(98)00007-1.
- [5] Henry P. Schwarcz, Diab Abueidda, and Iwona Jasiuk. “The Ultrastructure of Bone and Its Relevance to Mechanical Properties”. In: *Frontiers in Physics* 5 (2017), p. 39. DOI: 10.3389/fphy.2017.00039.
- [6] Furqan A. Shah. “Revisiting the physical and chemical nature of the mineral component of bone”. In: *Acta Biomaterialia* 196 (Apr. 2025), pp. 1–16. DOI: 10.1016/j.actbio.2025.01.055.
- [7] Natalie Reznikov et al. “Fractal-like Hierarchical Organization of Bone Begins at the Nanoscale”. In: *Science* 360.6388 (2018), eaao2189. DOI: 10.1126/science.aao2189.
- [8] Natalie Reznikov, Ron Shahar, and Steve Weiner. “Bone Hierarchical Structure in Three Dimensions”. In: *Acta Biomaterialia* 10.9 (2014), pp. 3815–3826. DOI: 10.1016/j.actbio.2014.05.024.
- [9] Atul S. Deshpande and Elia Beniash. “Bio-inspired Synthesis of Mineralized Collagen Fibrils”. In: *Crystal Growth & Design* 8.8 (Aug. 2008), pp. 3084–3090. DOI: 10.1021/cg800252f.
- [10] William J. Landis and Min J. Song. “Early Mineral Deposition in Calcifying Tendon Characterized by High Voltage Electron Microscopy and Three-Dimensional Graphic Imaging”. In: *Journal of Structural Biology* 107.2 (Oct. 1991), pp. 116–127. DOI: 10.1016/1047-8477(91)90015-0.
- [11] Elizabeth A. McNally et al. “A Model for the Ultrastructure of Bone Based on Electron Microscopy of Ion-Milled Sections”. In: *PLOS ONE* 7.1 (2012), e29258. DOI: 10.1371/journal.pone.0029258.

- [12] Henry P. Schwarcz et al. “A Search for Apatite Crystals in the Gap Zone of Collagen Fibrils in Bone Using Dark-Field Illumination”. In: *Bone* 135 (2020), p. 115304. DOI: 10.1016/j.bone.2020.115304.
- [13] K. Grandfield, V. Vuong, and H. P. Schwarcz. “Ultrastructure of Bone: Hierarchical Features from Nanometer to Micrometer Scale Revealed in Focused Ion Beam Sections in the TEM”. In: *Calcified Tissue International* 103.6 (July 2018), pp. 606–616. DOI: 10.1007/S00223-018-0454-9.
- [14] Chiara Micheletti et al. “Advanced Electron Microscopy to Study the Collagen–Mineral Nanocomposite in Human Bone: From On-Axis Electron Tomography to 4D-STEM”. In: *Microscopy and Microanalysis* 31.Supplement_1 (2025), pp. 1880–1881. DOI: 10.1093/mam/ozaf048.948.
- [15] California NanoSystems Institute. *Advanced scan generator unlocks new possibilities in 4DSTEM at UCLA user facility*. California NanoSystems Institute, UCLA. Feb. 7, 2024. URL: <https://cnsi.ucla.edu/february-7-2024-advanced-scan-generator-unlocks-new-possibilities-in-4dstem-at-ucla-user-facility/> (visited on 05/03/2026).
- [16] JEOL Ltd. *4 dimensional-scanning transmission electron microscopy, 4D-STEM*. JEOL Ltd. 2025. URL: <https://www.jeol.com/words/emterms/20250826.php> (visited on 05/03/2026).
- [17] NCEM / Molecular Foundry. *Working with 4D Camera Data and with NERSC*. Internal documentation provided by supervisor. n.d.
- [18] Savitzky et al. *py4DSTEM Tutorials Repository*. https://github.com/py4dstem/py4DSTEM_tutorials. Accessed: 2026-06-23.
- [19] Peter Ercius et al. “The 4D Camera: An 87 kHz Direct Electron Detector for Scanning/Transmission Electron Microscopy”. In: *Microscopy and Microanalysis* 30.5 (2024), pp. 903–912. DOI: 10.1093/mam/ozae086. URL: <https://doi.org/10.1093/mam/ozae086>.
- [20] Tatiana Kochetkova et al. “Structural and Mechanical Analysis of Individual Mineralized Collagen Fibrils Using In Situ Transmission Electron Microscopy”. In: *ACS Nano* 20.12 (2026), pp. 10127–10137. DOI: 10.1021/acsnano.6c00964.
- [21] Henry P. Schwarcz, Elizabeth A. McNally, and Gianluigi A. Botton. “Dark-Field Transmission Electron Microscopy of Cortical Bone Reveals Details of Extrafibrillar Crystals”. In: *Journal of Structural Biology* 188.3 (2014), pp. 240–248. DOI: 10.1016/j.jsb.2014.10.005.
- [22] V. Ziv, H.D. Wagner, and S. Weiner. “Microstructure-microhardness relations in parallel-fibered and lamellar bone”. In: *Bone* 18.5 (1996), pp. 417–428. ISSN: 8756-3282. DOI: [https://doi.org/10.1016/8756-3282\(96\)00049-X](https://doi.org/10.1016/8756-3282(96)00049-X). URL: <https://www.sciencedirect.com/science/article/pii/S875632829600049X>.
- [23] YiFei Xu et al. “Intermolecular channels direct crystal orientation in mineralized collagen”. In: *Nature Communications* 11.1 (Oct. 2020), p. 5068. DOI: 10.1038/s41467-020-18846-2.

A

Appendix 1

A.1 Other results

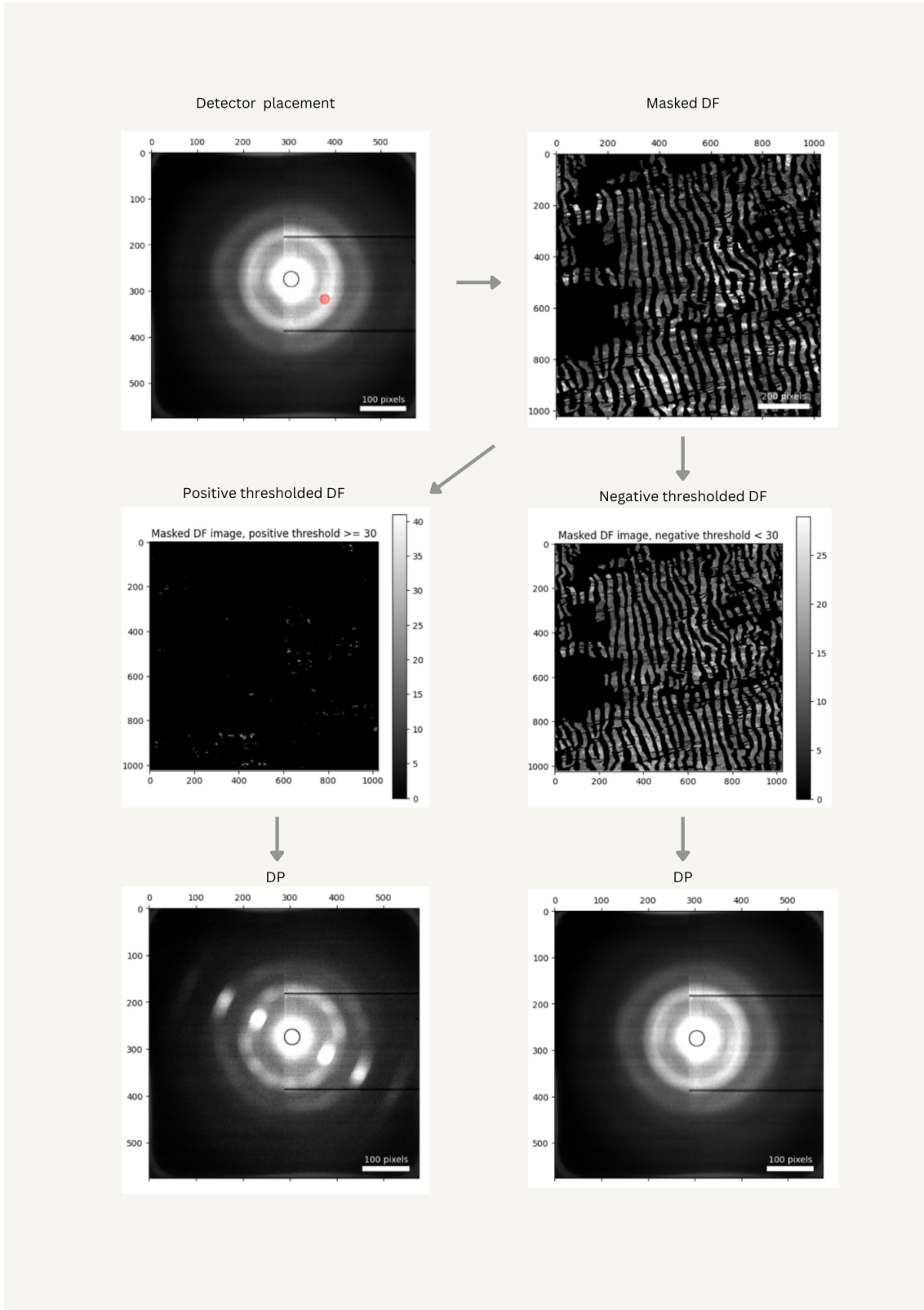


Figure A.1: Dark bands results. The same detector geometry and position were used as in the bright bands analysis presented in the Results.

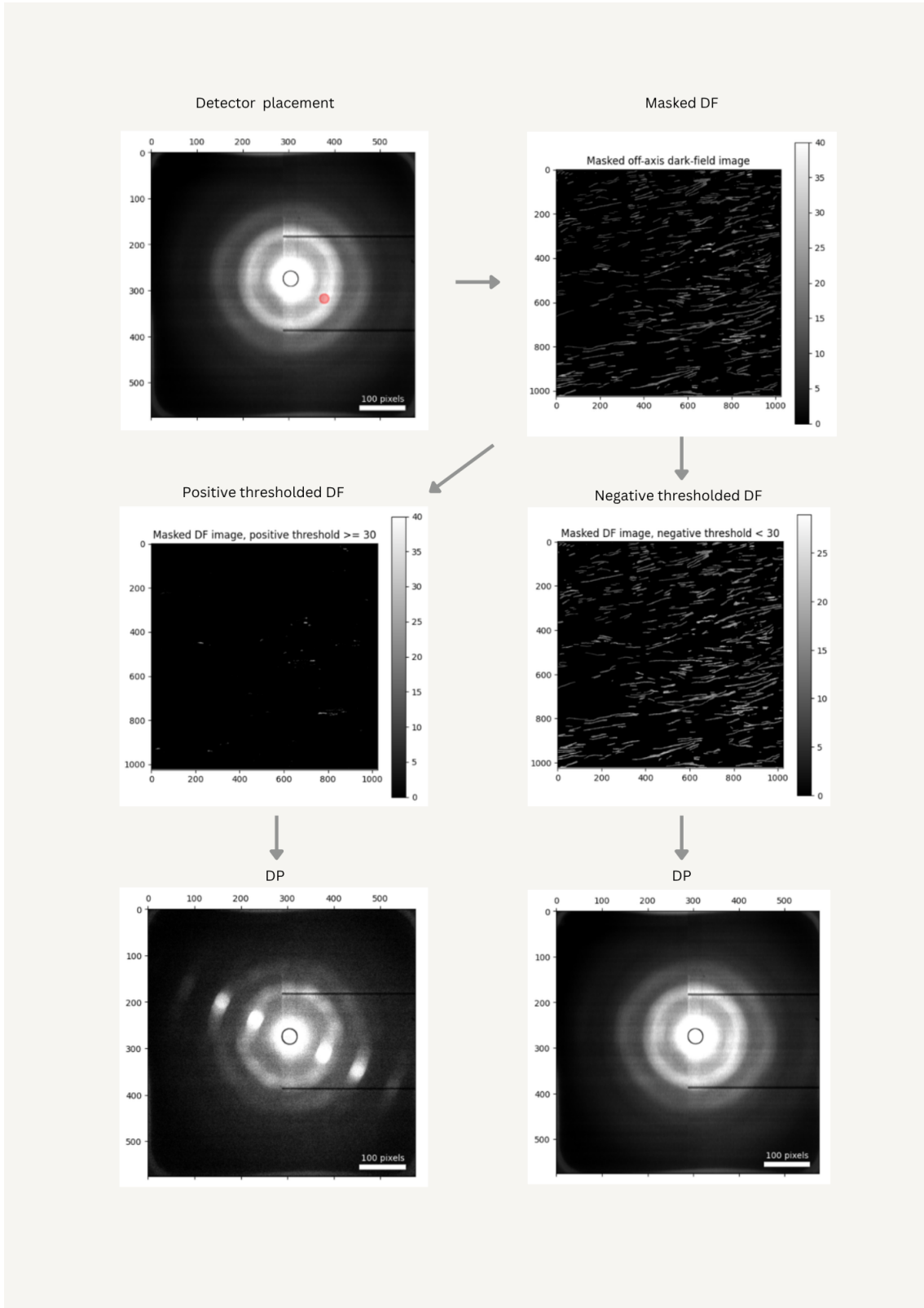


Figure A.2: Extra-fibrillar mineral results. The same detector geometry and position were used as in the bright bands analysis presented in the Results.

B

Appendix 2

B.1 Example of Jupyter notebook

```
In [1]: # Set up the environment

import numpy as np
import matplotlib.pyplot as plt
import matplotlib.patches as patches
from matplotlib.patches import Circle
import tiffio
import tiffio as tiff
import imageio.v3 as iio
import ncempy.io as nio
from pathlib import Path
from stempy.io.sparse_array import SparseArray
from py4DSTEM import DataCube
import py4DSTEM
from scipy.ndimage import gaussian_filter

print('Imports OK')
py4DSTEM.__version__
```

```
In [2]: #Load the data
```

```
In [2]: plt.close("all")

# -----
# User settings
# -----
date = "2022.03.04"
scan_num = 7
scan_id = 1428
threshold = 4.5

distiller_path = Path("/global/cfs/projectdirs/ncemhub/distiller/")

# counted sparse 4D-STEM file
dPath = distiller_path / "counted" / date
fPath = Path(f"data_scan{scan_num}_id{scan_id}_th{threshold}_electr
fpath_4d = dPath / fPath

# HAADF dm4 (if available)
haadf_path = distiller_path / "dm4" / date / f"scan{scan_num}.dm4"

print("4D file :", fpath_4d)
print("HAADF file:", haadf_path)

# -----
# Check 4D file exists
# -----
if not fpath_4d.exists():
    raise FileNotFoundError(f"4D-STEM file not found: {fpath_4d}")

# -----
# Load HAADF (optional)
# -----
haadf = None
```



```

haadf_data = None

if haadf_path.exists():
    haadf = nio.dm.dmReader(str(haadf_path))
    haadf_data = np.asarray(haadf["data"], dtype=float)

    fig, ax = plt.subplots(1, 1, figsize=(5, 5))
    ax.imshow(haadf_data, cmap="gray")
    ax.set_title("Simultaneously acquired HAADF")

    save_dir = Path("/global/homes/b/burak_er/Results/S07_R09_Mif")
    save_dir.mkdir(parents=True, exist_ok=True)

    out_png = save_dir / "S07_R09_HAADF_full.png"
    out_tiff = save_dir / "S07_R09_HAADF_full.tiff"

    fig.savefig(out_png, dpi=300, bbox_inches="tight")
    fig.savefig(out_tiff, dpi=300, bbox_inches="tight")

    print("Saved figure PNG to :", out_png)
    print("Saved figure TIFF to:", out_tiff)

    plt.show()
else:
    print("No HAADF available")

#Saving the raw data (image array)
save_dir = Path("/global/homes/b/burak_er/Results/S07_R09_Mif")
save_dir.mkdir(parents=True, exist_ok=True)

# --- full HAADF raw TIFF ---
out_tiff_raw = save_dir / "S07_R09_HAADF_full_raw.tiff"
tifffile.imwrite(out_tiff_raw, np.asarray(haadf_data, dtype=np.float))

# --- full HAADF PNG (normalized for viewing) ---
arr = np.asarray(haadf_data, dtype=float)
arr_norm = arr - arr.min()
if arr_norm.max() > 0:
    arr_norm = arr_norm / arr_norm.max()
arr_uint8 = (255 * arr_norm).astype(np.uint8)

out_png_raw = save_dir / "S07_R09_HAADF_full_raw.png"
iio.imwrite(out_png_raw, arr_uint8)

print("Saved raw TIFF to:", out_tiff_raw)
print("Saved PNG to      :", out_png_raw)

# -----
# Load sparse 4D dataset
# -----
sa = SparseArray.from_hdf5(fpath_4d)

print("Initial scan_shape:", sa.scan_shape) # (Ny, Nx)
print("Frame shape      :", sa.frame_shape) # (Qy, Qx)

# -----

```

```

# Optional: mean electrons per pattern
# -----
num_elec = 0
for hits in sa.data:
    num_elec += len(hits)

avg_elec = num_elec / (sa.scan_shape[0] * sa.scan_shape[1])
print("Mean electrons per pattern:", avg_elec)

# Only a rough estimate; keep with caution
if haadf is not None and "pixelSize" in haadf and "pixelUnit" in haadf:
    try:
        dose_est = avg_elec / haadf["pixelSize"][0]**2
        print(f"Approx. dose (using DM4 pixel size) = {dose_est:.3f}")
    except Exception:
        pass

# -----
# Define ROI (scan-space crop)
# NOTE: x = horizontal scan index, y = vertical scan index
# -----
x0, x1 = 0, 1024
y0, y1 = 0, 1024

Ny_scan, Nx_scan = sa.scan_shape
if not (0 <= x0 < x1 <= Nx_scan and 0 <= y0 < y1 <= Ny_scan):
    raise ValueError("ROI bounds are outside the scan dimensions.")

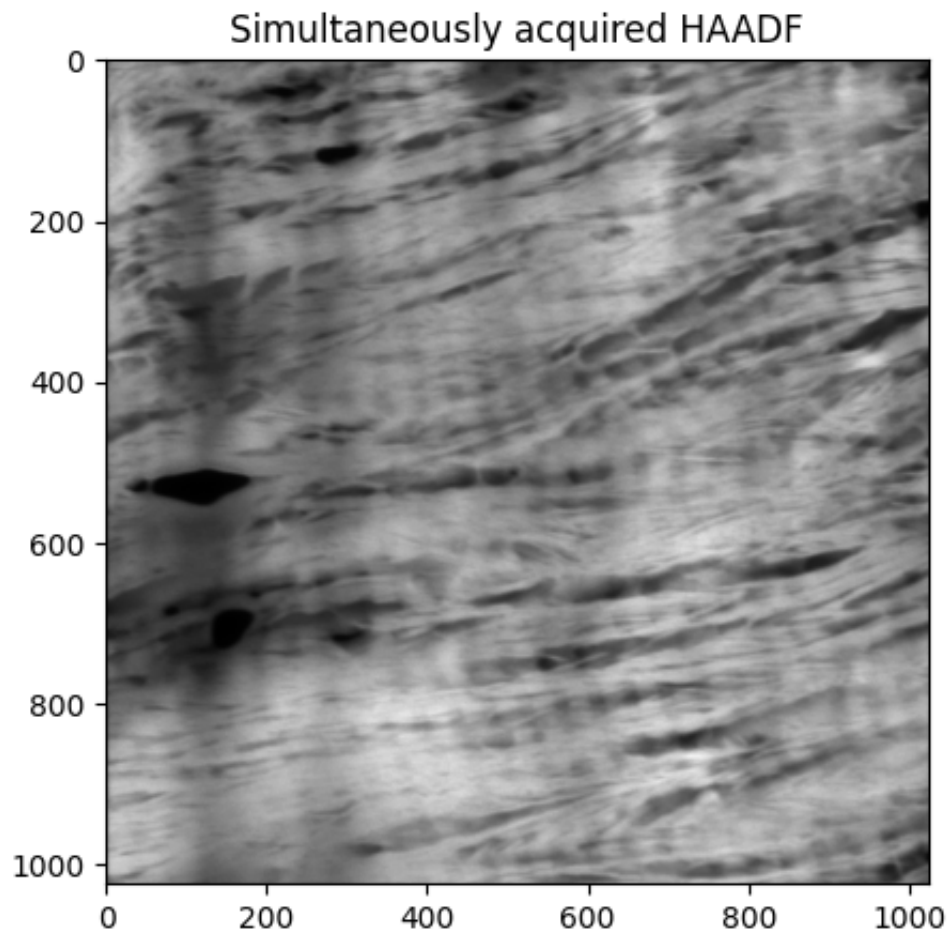
sa_roi = sa[y0:y1, x0:x1]
print("ROI scan_shape:", sa_roi.scan_shape, "(Ny, Nx)")

# Matching HAADF crop for later use
haadf_roi = None
if haadf_data is not None:
    haadf_roi = haadf_data[y0:y1, x0:x1]
    print("HAADF ROI shape:", haadf_roi.shape)

# -----
# Wrap into py4DSTEM DataCube
# -----
dc = DataCube(sa_roi)
print("dc.shape:", dc.shape) # (Ny, Nx, Qy, Qx)

```

4D file : /global/cfs/projectdirs/ncemhub/distiller/counted/2022.03.04/data_scan7_id1428_th4.5_electrons_centered.h5
 HAADF file: /global/cfs/projectdirs/ncemhub/distiller/dm4/2022.03.04/scan7.dm4
 Saved figure PNG to : /global/homes/b/burak_er/Results/S07_R09_Mif/S07_R09_HAADF_full.png
 Saved figure TIFF to: /global/homes/b/burak_er/Results/S07_R09_Mif/S07_R09_HAADF_full.tiff



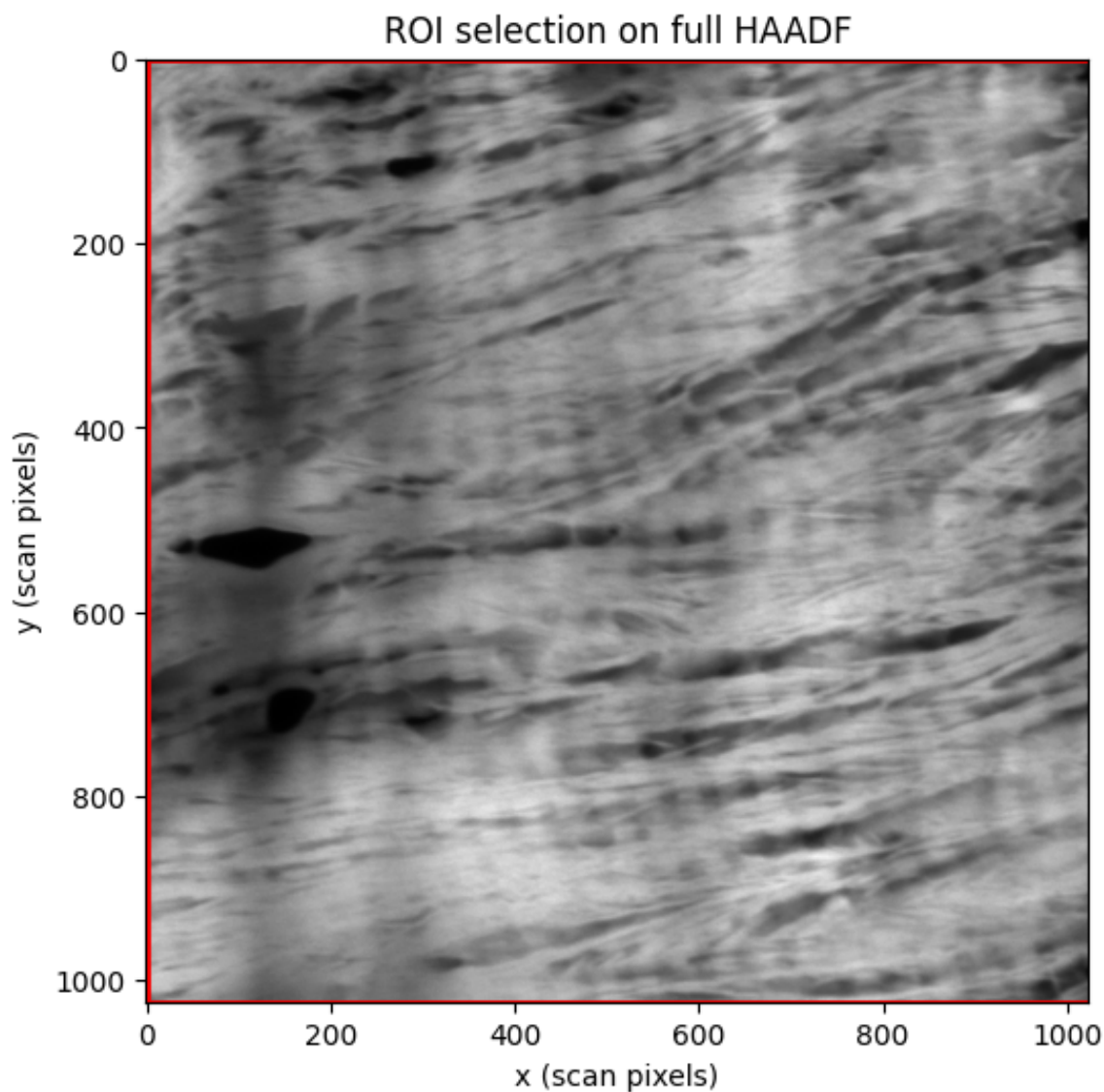
Saved raw TIFF to: /global/homes/b/burak_er/Results/S07_R09_Mif/S07_R09_HAADF_full_raw.tiff
 Saved PNG to : /global/homes/b/burak_er/Results/S07_R09_Mif/S07_R09_HAADF_full_raw.png
 Initial scan_shape: (1024, 1024)
 Frame shape : (576, 576)
 Mean electrons per pattern: 1.0
 Approx. dose (using DM4 pixel size) = 1.610 e-/nm^2
 ROI scan_shape: (1024, 1024) (Ny, Nx)
 HAADF ROI shape: (1024, 1024)
 dc.shape: (1024, 1024, 576, 576)

In [3]: *# Visualize ROI rectangle on full HAADF image*

```
if haadf_data is None:
    print("No HAADF image available")
else:
    fig, ax = plt.subplots(figsize=(6, 6))
    ax.imshow(haadf_data, cmap="gray")

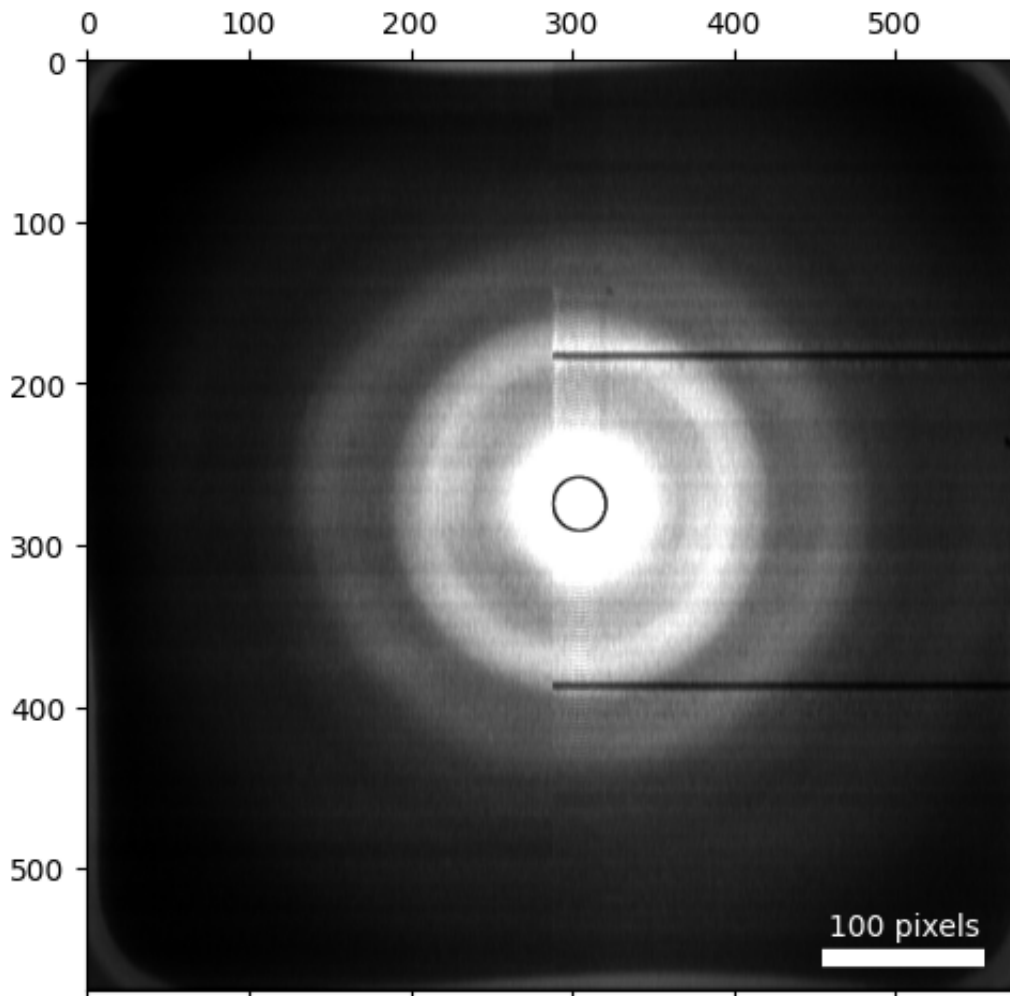
    rect = patches.Rectangle(
        (x0, y0),
        x1 - x0,
        y1 - y0,
        linewidth=2,
        edgecolor="red",
        facecolor="none"
    )
    ax.add_patch(rect)
```

```
ax.set_title("ROI selection on full HAADF")  
ax.set_xlabel("x (scan pixels)")  
ax.set_ylabel("y (scan pixels)")  
plt.show()
```



In [6]: *# Diffraction pattern from the selected ROI*

```
vd = dc.get_virtual_diffraction(  
    method="mean",  
    name="vd_roi"  
)  
  
py4DSTEM.visualize.show(vd, scaling="none")
```



```
In [7]: #Load the binary mask and crop it to the same ROI

mask_path = Path("/global/homes/b/burak_er/Minerals_final.tiff") #

# Read TIFF
try:
    mask_full = tiff.imread(str(mask_path))
except ImportError:
    mask_full = iio.imread(mask_path)

# Convert to boolean: True where mask is nonzero
mask_full = np.asarray(mask_full)
mask_full_bool = mask_full > 0

# Crop the mask using the SAME ROI as the DataCube
mask_roi = mask_full_bool[y0:y1, x0:x1]

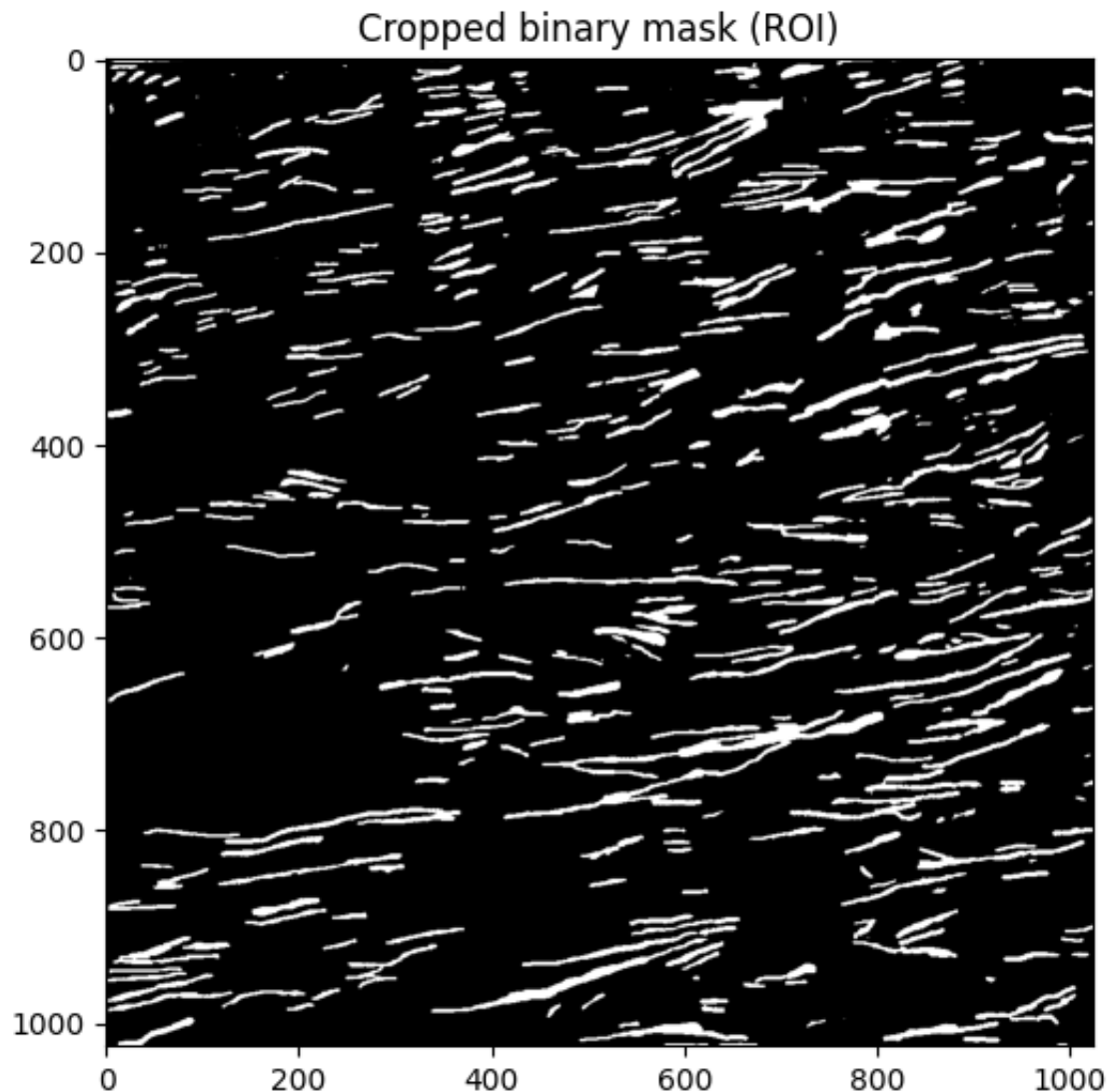
print("dc.Rshape:", dc.Rshape)
print("mask_roi.shape:", mask_roi.shape)
print("mask_roi dtype:", mask_roi.dtype)

assert mask_roi.shape == dc.Rshape, "Mask shape must match dc.Rshap
assert mask_roi.dtype == bool, "Mask must be boolean"
```

```
dc.Rshape: (1024, 1024)
mask_roi.shape: (1024, 1024)
mask_roi dtype: bool
```

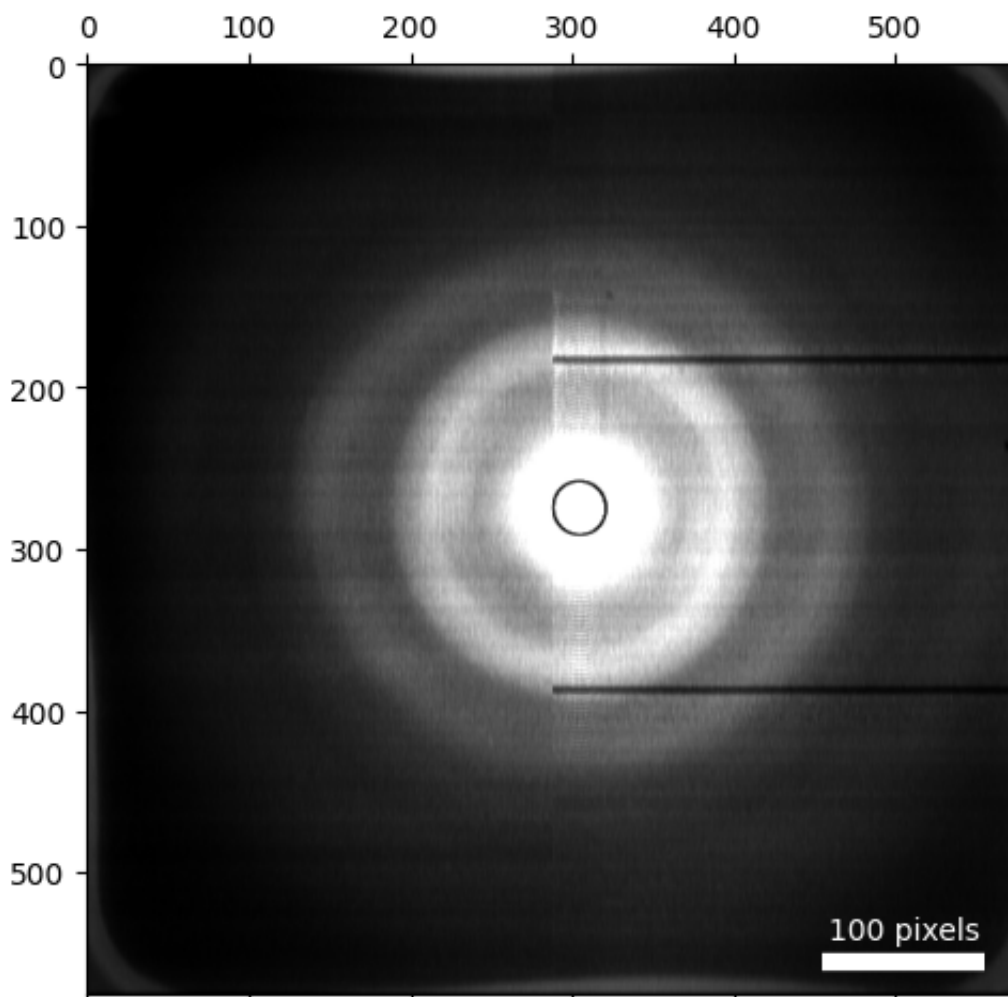
In [8]: *#Visualizing the cropped mask*

```
plt.figure(figsize=(6,6))
plt.imshow(mask_roi, cmap="gray")
plt.title("Cropped binary mask (ROI)")
plt.show()
```



```
In [9]: dp_mean = dc.get_dp_mean()
py4DSTEM.visualize.show(dp_mean, scaling="none") # or scaling = log

# crude peak near middle (good enough to start)
cy0, cx0 = dp_mean.data.shape[0]//2, dp_mean.data.shape[1]//2
w = 80
sub = dp_mean.data[cy0-w:cy0+w, cx0-w:cx0+w]
iy, ix = np.unravel_index(np.argmax(sub), sub.shape)
cy_peak = (cy0-w) + iy
cx_peak = (cx0-w) + ix
print("beam center (cx, cy):", cx_peak, cy_peak)
```



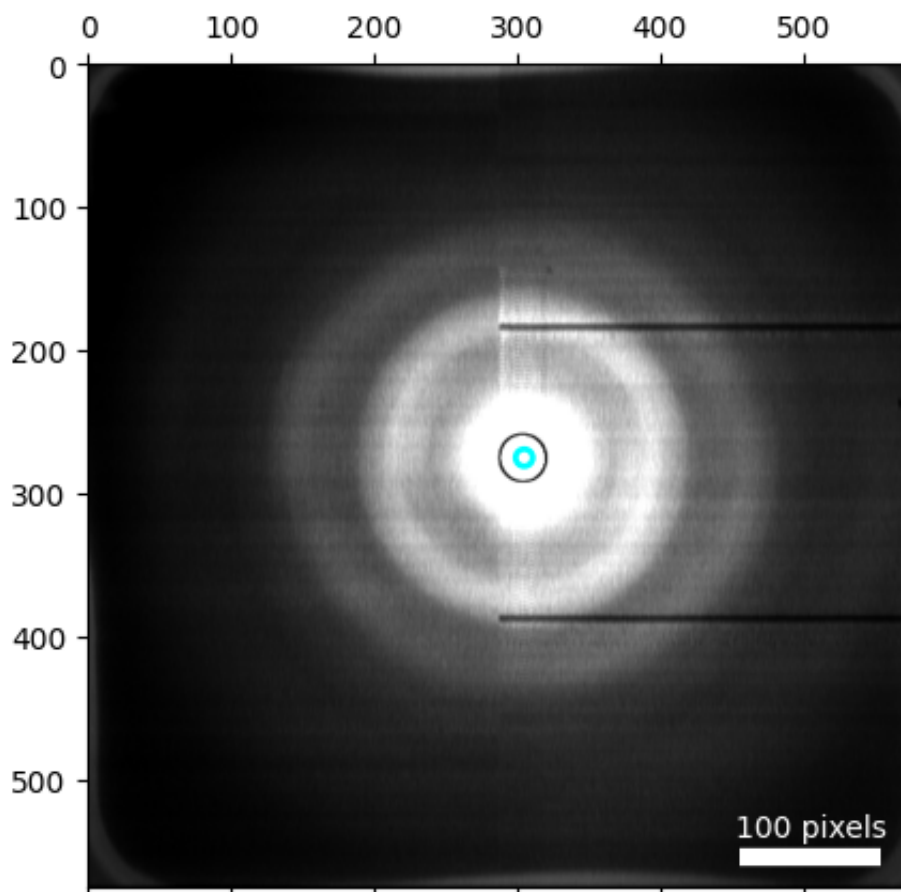
beam center (cx, cy): 314 286

```
In [10]: a = -11 #-11
b = -9 # -9
center = (cy_peak + a, cx_peak + b) # IMPORTANT: (y, x)
r_inner = 65 # just outside central disk 65
r_outer = 85 # covers a ring region 85, 119
radii = (r_inner, r_outer)
```

```
In [11]: # Confirm the center
cx_center = cx_peak + b
cy_center = cy_peak + a

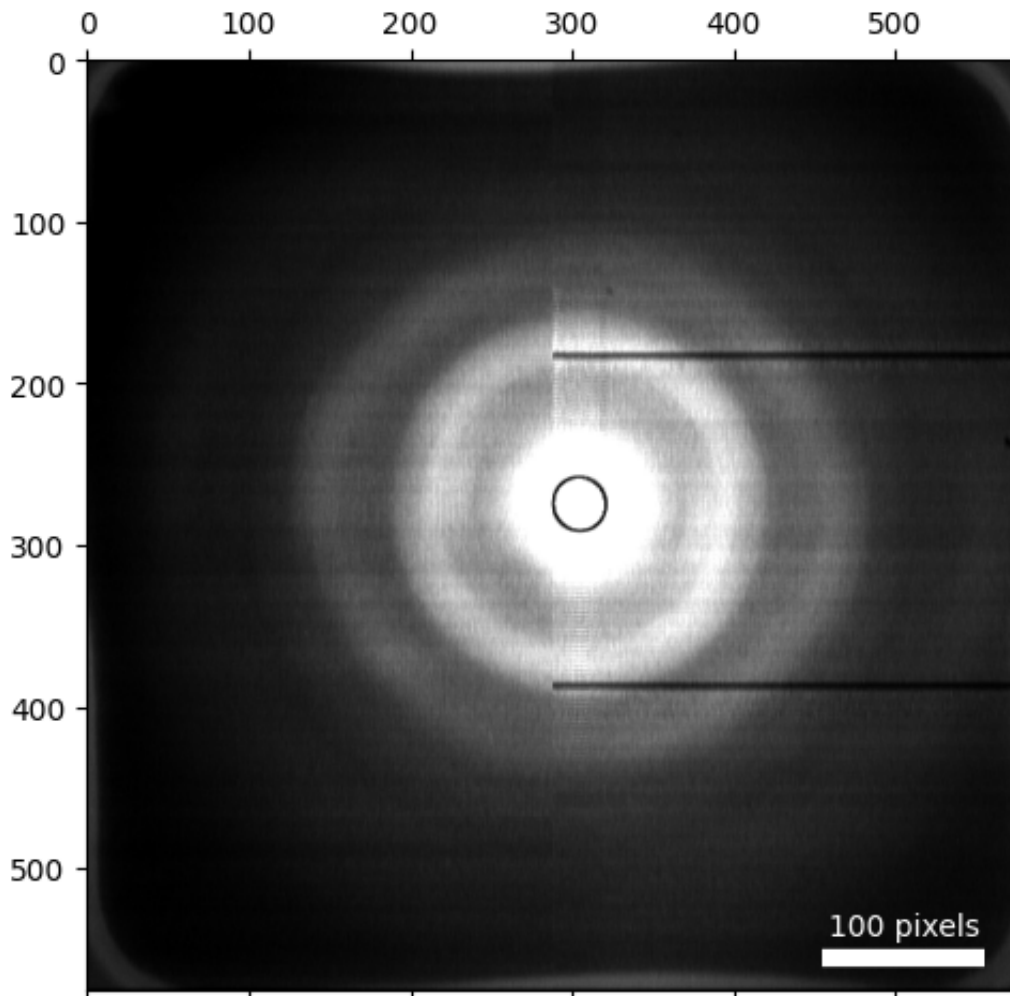
fig, ax = py4DSTEM.visualize.show(
    dp_mean, #vd_masked
    scaling="none",
    returnfig=True
)

ax.add_patch(Circle((cx_center, cy_center), 6, fill=False, linewidth=6))
plt.show()
```

In [12]: *#Compute virtual diffraction from only the masked pixels*

```
vd_masked = dc.get_virtual_diffraction(  
    method="mean",  
    mask=mask_roi,  
    name="vd_masked"  
)  
  
py4DSTEM.visualize.show(vd_masked, scaling="none")
```

```
In [13]: #vd = S07_R09_Dp01_Dbf_DpDbf # from dc.get_virtual_diffraction(...)

if hasattr(vd, "data"): #vd_masked
    dp = np.asarray(vd.data)
elif hasattr(vd, "get_data"):
    dp = np.asarray(vd.get_data())
else:
    dp = np.asarray(vd)

print("dp:", type(dp), dp.shape, dp.dtype, "min/max:", float(dp.min
dp: <class 'numpy.ndarray'> (576, 576) float32 min/max: 0.0009851455
688476562 0.1978759765625
```

```
In [14]: # If you already have cx,cy from COM or manual work, use those:
# cx_peak, cy_peak = cx, cy

# Otherwise: find peak near the center area (robust-ish)
cy0, cx0 = dp.shape[0] / 2, dp.shape[1] / 2
w = 60 # 30, search half-window, adjust 30-120
y2 = y0
y3 = y1
x2 = x0
x3 = x1
y2 = int(max(0, cy0 - w)); y3 = int(min(dp.shape[0], cy0 + w + 1))
x2 = int(max(0, cx0 - w)); x3 = int(min(dp.shape[1], cx0 + w + 1))
```

```

sub = dp[y2:y3, x2:x3]
iy, ix = np.unravel_index(np.argmax(sub), sub.shape)

cy_peak = y2 + iy
cx_peak = x2 + ix

print("Peak center (cx, cy):", cx_peak, cy_peak)

```

Peak center (cx, cy): 314 286

```

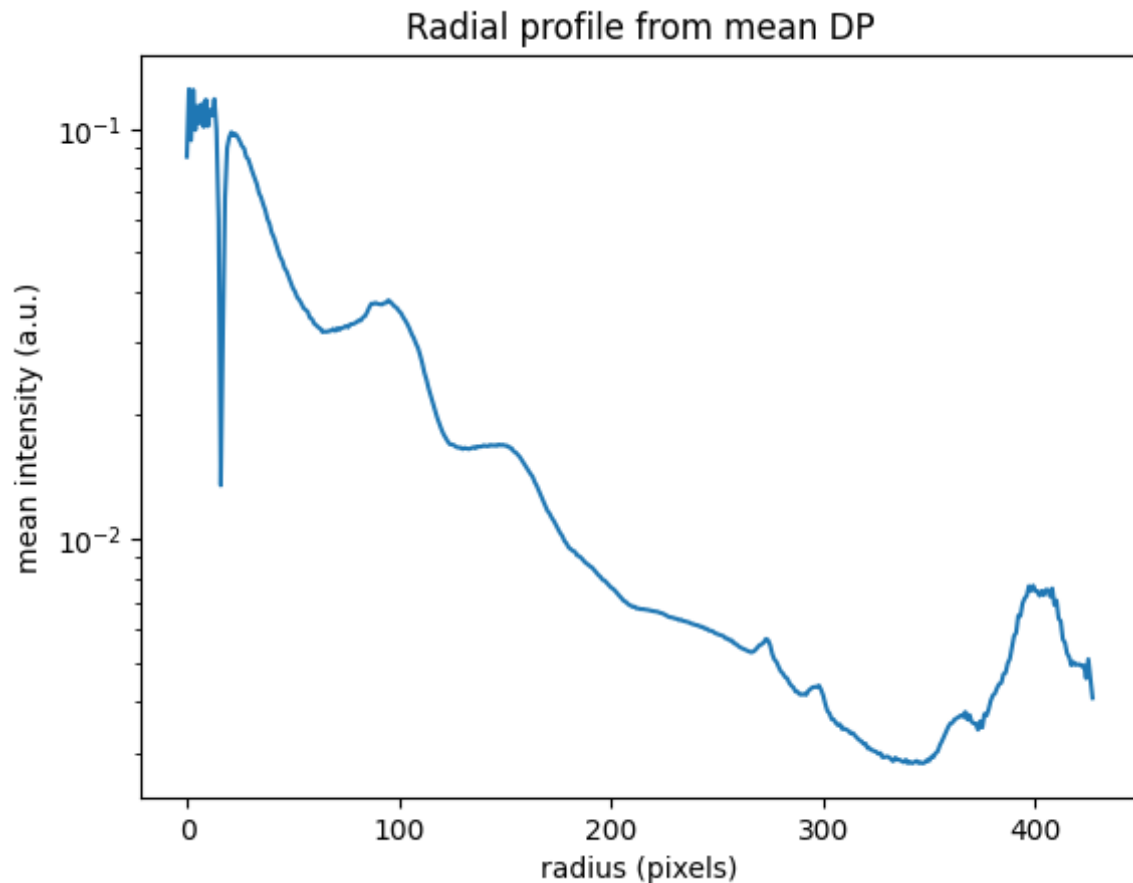
In [15]: #Peak center. Brute force it if it's off the center, otherwise keep
cx_peak=305
cy_peak=275

def radial_profile(img, cx, cy, nbins=None):
    # img must be 2D numpy array
    y, x = np.indices(img.shape)
    r = np.sqrt((x - cx)**2 + (y - cy)**2)
    r_int = r.astype(int)
    if nbins is None:
        nbins = r_int.max()+1
    sums = np.bincount(r_int.ravel(), weights=img.ravel(), minlength=nbins)
    counts = np.bincount(r_int.ravel(), minlength=nbins)
    with np.errstate(invalid='ignore', divide='ignore'):
        prof = sums / np.maximum(counts, 1)
    return prof, counts

# get dp array using the code in section 2
# dp, cx, cy must be defined
prof, counts = radial_profile(dp, cx=cx_peak, cy=cy_peak) # adjust

plt.figure()
plt.semilogy(prof) # log y is often useful for diffraction rings
plt.xlabel("radius (pixels)")
plt.ylabel("mean intensity (a.u.)")
plt.title("Radial profile from mean DP")
plt.show()

```



```
In [17]: r_in = 65 # 65 Adjust these parameters in the related cell/cells bel
r_out = 85 # 119, 85
```

```
In [18]: def ring_angular_intensity(img, cx, cy, r_in, r_out, ntheta=360):
    y, x = np.indices(img.shape)
    X = x - cx
    Y = y - cy
    R = np.sqrt(X**2 + Y**2)
    Theta = np.arctan2(Y, X) # range -pi..pi

    # mask ring
    ring_mask = (R >= r_in) & (R < r_out)
    thetas = Theta[ring_mask]
    vals = img[ring_mask]

    # map -pi..pi to 0..2pi
    thetas = np.mod(thetas, 2*np.pi)

    # bin in theta
    bins = np.linspace(0, 2*np.pi, ntheta+1)
    I_theta, _ = np.histogram(thetas, bins=bins, weights=vals)
    counts, _ = np.histogram(thetas, bins=bins)
    with np.errstate(invalid='ignore', divide='ignore'):
        I_theta_norm = I_theta / np.maximum(counts, 1)

    # compute nematic A,B,S,phi
    theta_centers = 0.5*(bins[:-1] + bins[1:])
    A = np.sum(I_theta * np.cos(2*theta_centers))
    B = np.sum(I_theta * np.sin(2*theta_centers))
```

```

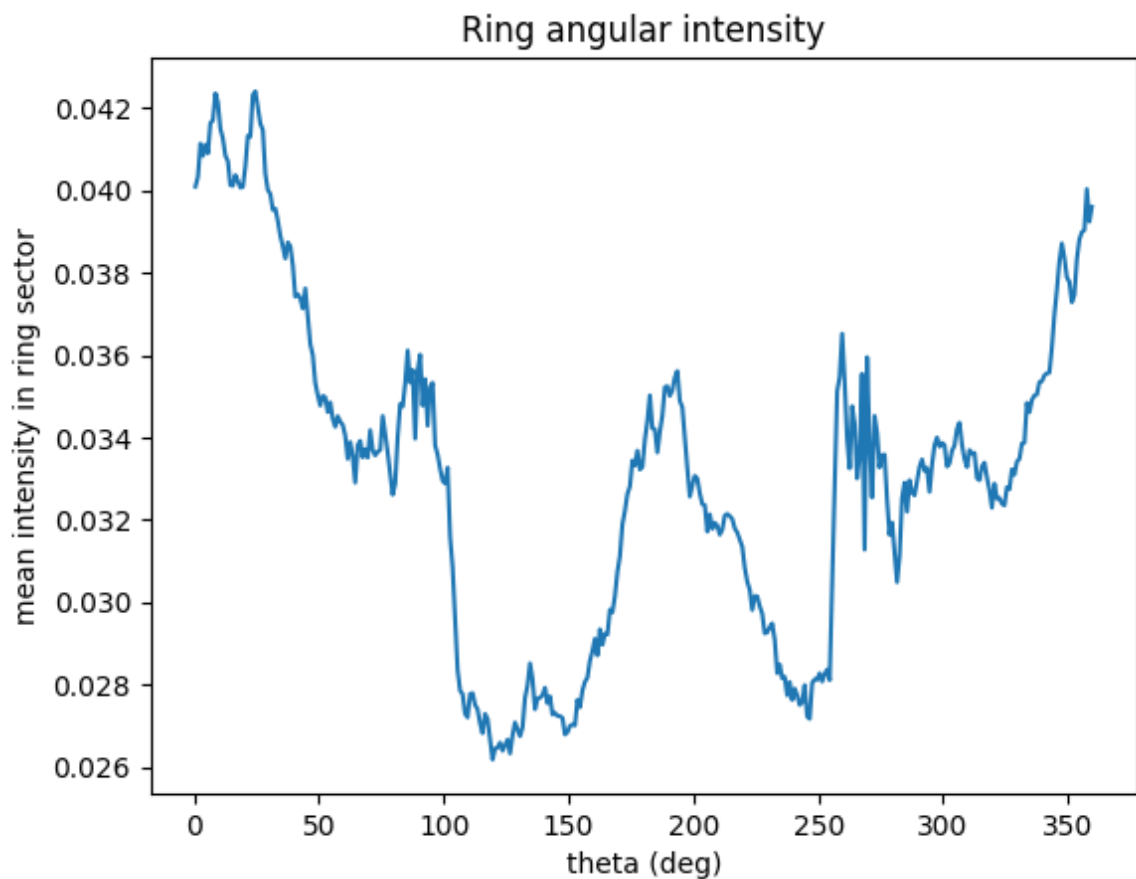
total = np.sum(I_theta)
S = np.sqrt(A**2 + B**2) / (total + 1e-12)
phi = 0.5 * np.arctan2(B, A) # radians

return theta_centers, I_theta, I_theta_norm, S, phi

# example usage on your mean DP:
theta_centers, I_theta, I_theta_norm, S_ring, phi_ring = ring_angular_intensity(
    dp, cx=cx_peak, cy=cy_peak, r_in=r_in, r_out=r_out, ntheta=360
)

plt.figure()
plt.plot(np.degrees(theta_centers), I_theta_norm)
plt.xlabel("theta (deg)")
plt.ylabel("mean intensity in ring sector")
# plt.title(f"Ring angular intensity, S={S_ring:.3f}, phi={np.degrees(phi_ring):.3f}")
plt.title(f"Ring angular intensity")
plt.show()

```



```

In [21]: fig, ax = plt.subplots(figsize=(6, 6))
im = np.log1p(dp)

r_in = 65 #After adjusting the annulus, don't forget to update the
r_out = 85 # 119

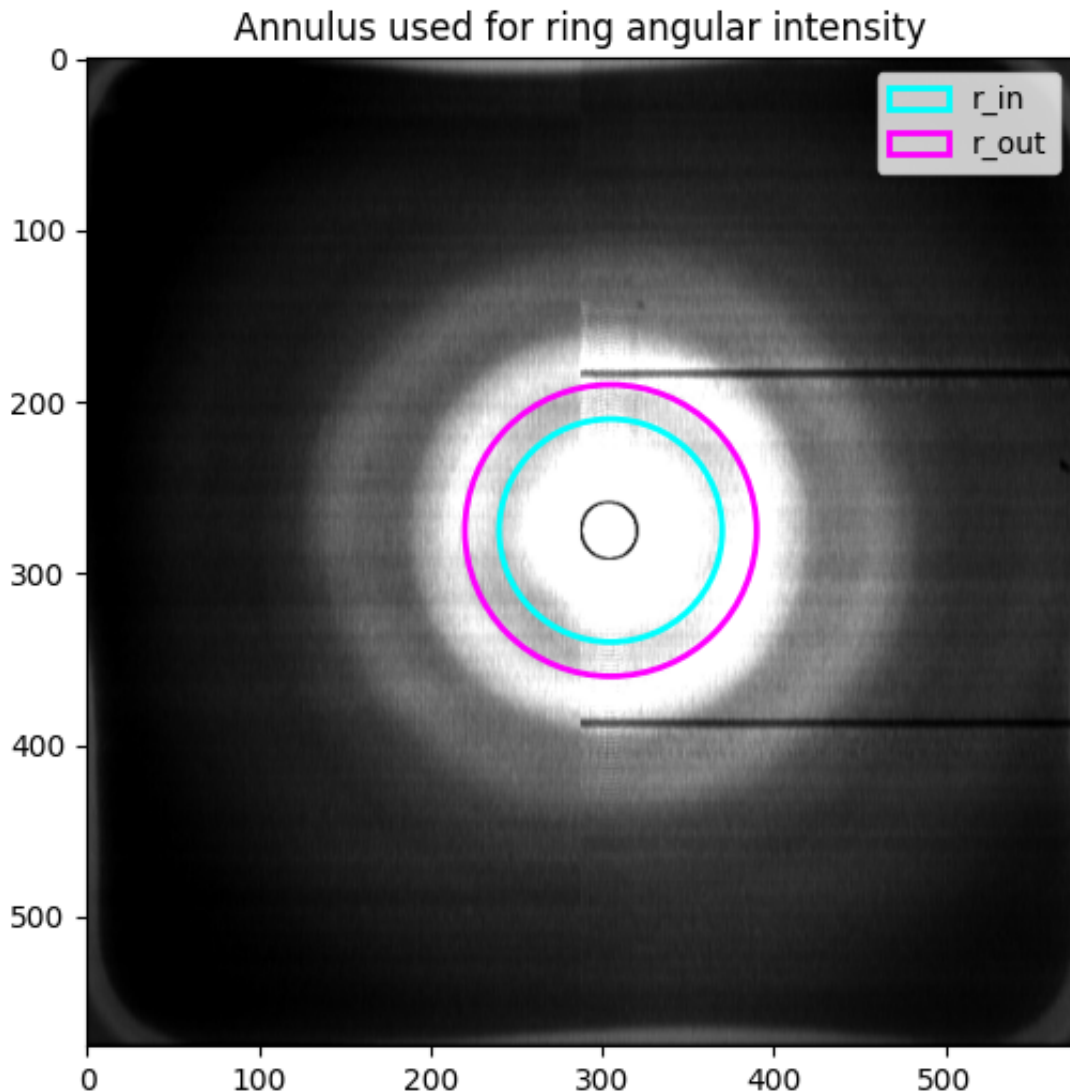
ax.imshow(
    im,
    cmap="gray",
    vmin=np.percentile(im, 5),
    vmax=np.percentile(im, 95) #99.8
)

```

```
)

circle_in = Circle((cx_peak, cy_peak), r_in, fill=False, edgecolor=
circle_out = Circle((cx_peak, cy_peak), r_out, fill=False, edgecolo

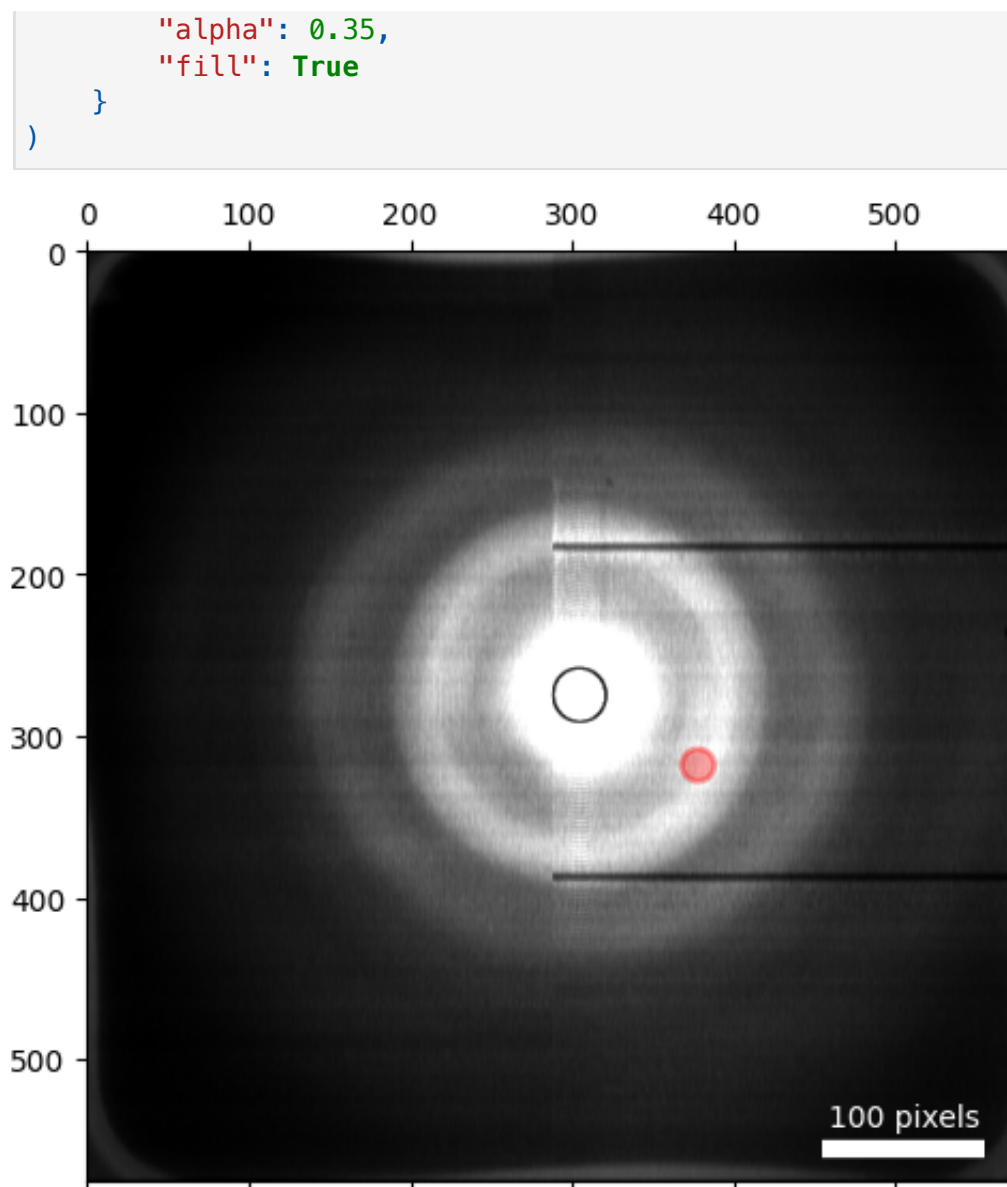
ax.add_patch(circle_in)
ax.add_patch(circle_out)
ax.set_title("Annulus used for ring angular intensity")
ax.legend()
plt.show()
```



```
In [27]: #Place the off-axis detector on the mean diffraction pattern

# choose detector center in diffraction coordinates
# write them as (x, y) for readability
qx0_DF, qy0_DF = 378, 318
r_DF = 10

py4DSTEM.visualize.show(
    vd,
    scaling="none",
    circle={
        "center": (qy0_DF, qx0_DF), # display convention used in
        "R": r_DF,
```



```

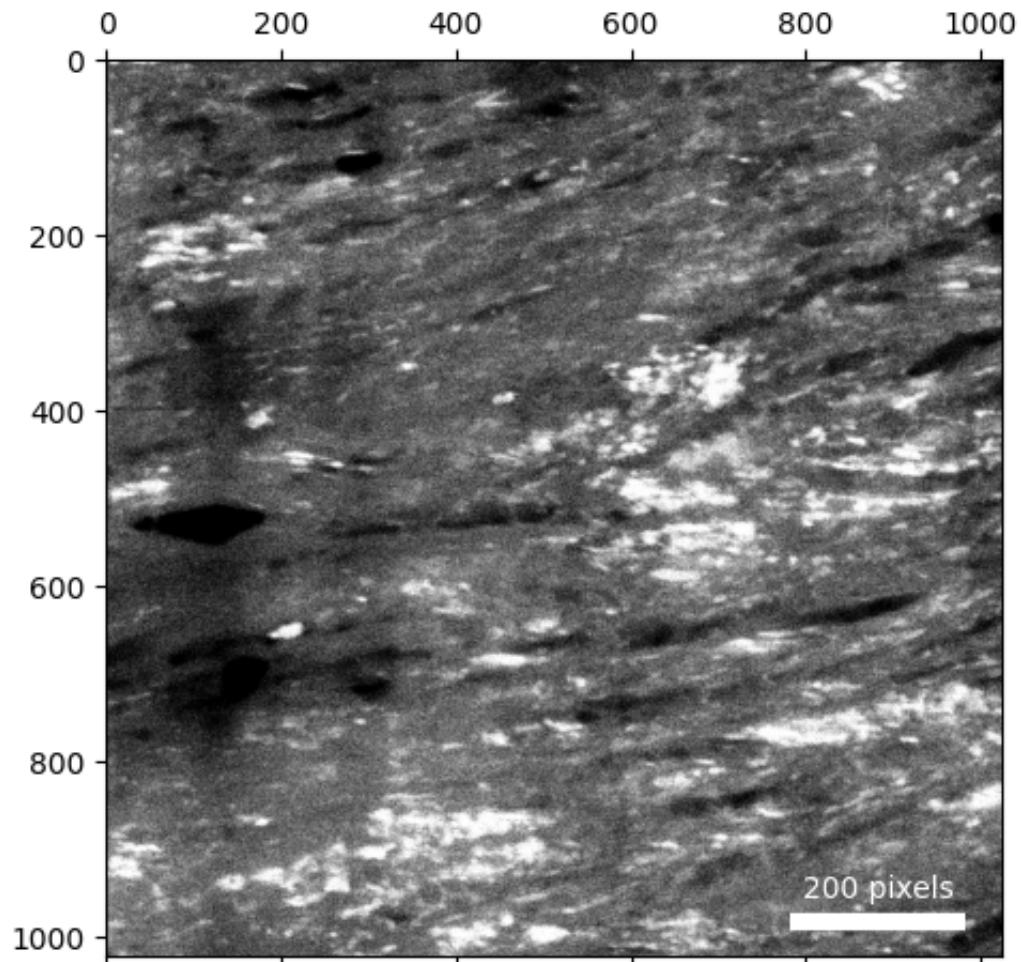
In [33]: # Compute off-axis dark-field image from the 4D dataset
geometry = (
    (qy0_DF, qx0_DF), # IMPORTANT: use (y, x)
    r_DF
)

df_offaxis = dc.get_virtual_image(
    mode="circle",
    geometry=geometry,
    name="DF_offaxis"
)

py4DSTEM.visualize.show(df_offaxis, scaling="none")

```

100%|██████████| 1048576/1048576 [17:29<00:00, 999.23it/s]



```
In [ ]: #Moon shaped off axis df
```

```
In [56]: #Moon shaped red disk
```

```
# center
cx = cx_center
cy = cy_center

# diffraction-space shape
Qy, Qx = dc.data.shape[2], dc.data.shape[3]
yy, xx = np.indices((Qy, Qx))

dx = xx - cx
dy = yy - cy
rr = np.sqrt(dx**2 + dy**2)
theta = np.arctan2(dy, dx)

def wrap_angle(a):
    return (a + np.pi) % (2*np.pi) - np.pi

# -----
# PARAMETERS TO PLAY WITH
# -----

r0 = 75          #75 distance from center
theta0_deg = 20  #5 angular position
dr_inner = 8     #8 # inner radial thickness
```

```
dr_outer = 12      #8      # outer radial thickness
dtheta_deg = 25    #18      # half angular width

# -----
# BUILD MOON MASK
# -----

theta0 = np.deg2rad(theta0_deg)
dtheta = np.deg2rad(dtheta_deg)

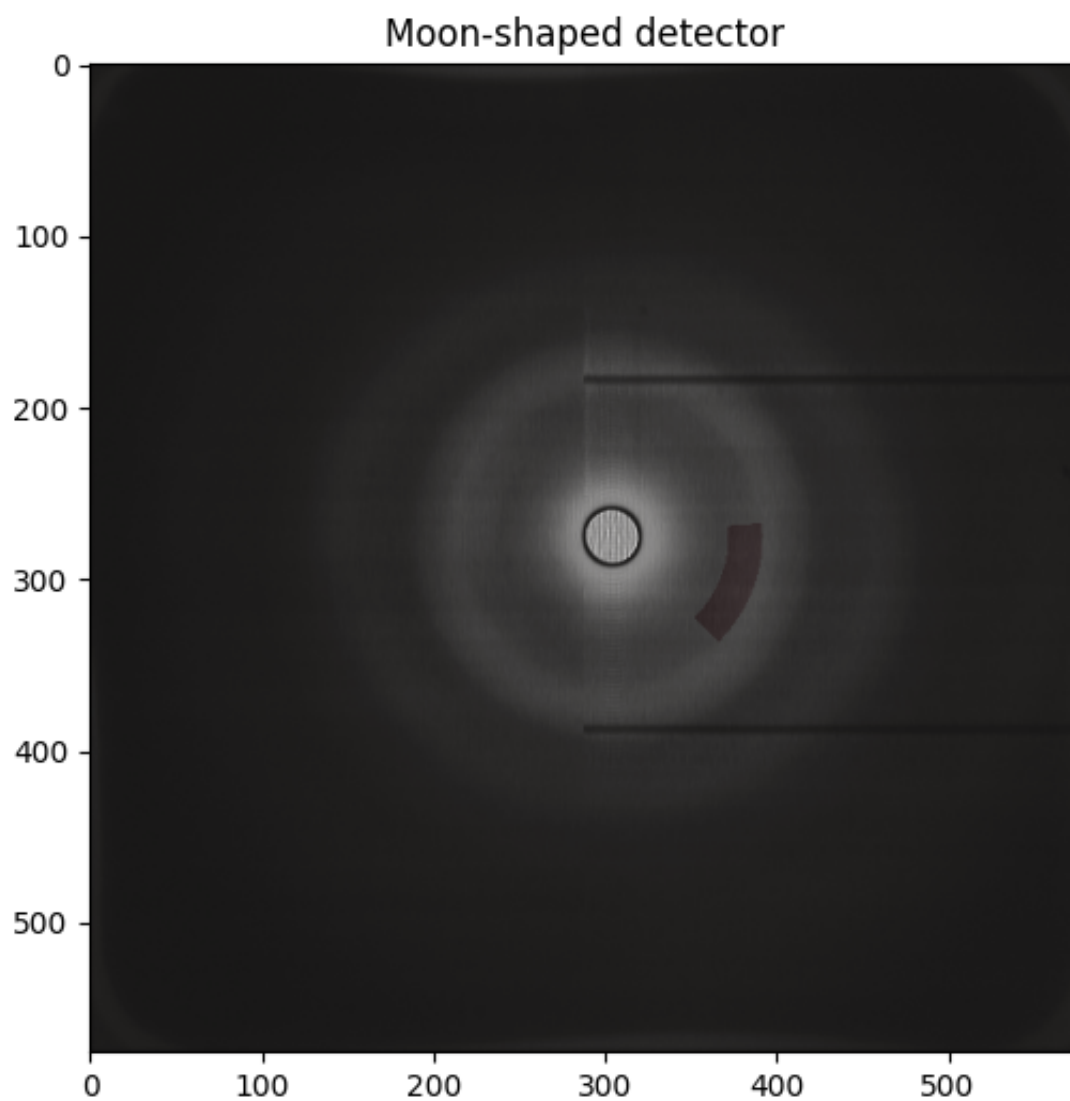
theta_diff = wrap_angle(theta - theta0)

moon_mask = (
    (rr >= (r0 - dr_inner)) &
    (rr <= (r0 + dr_outer)) &
    (np.abs(theta_diff) <= dtheta)
)

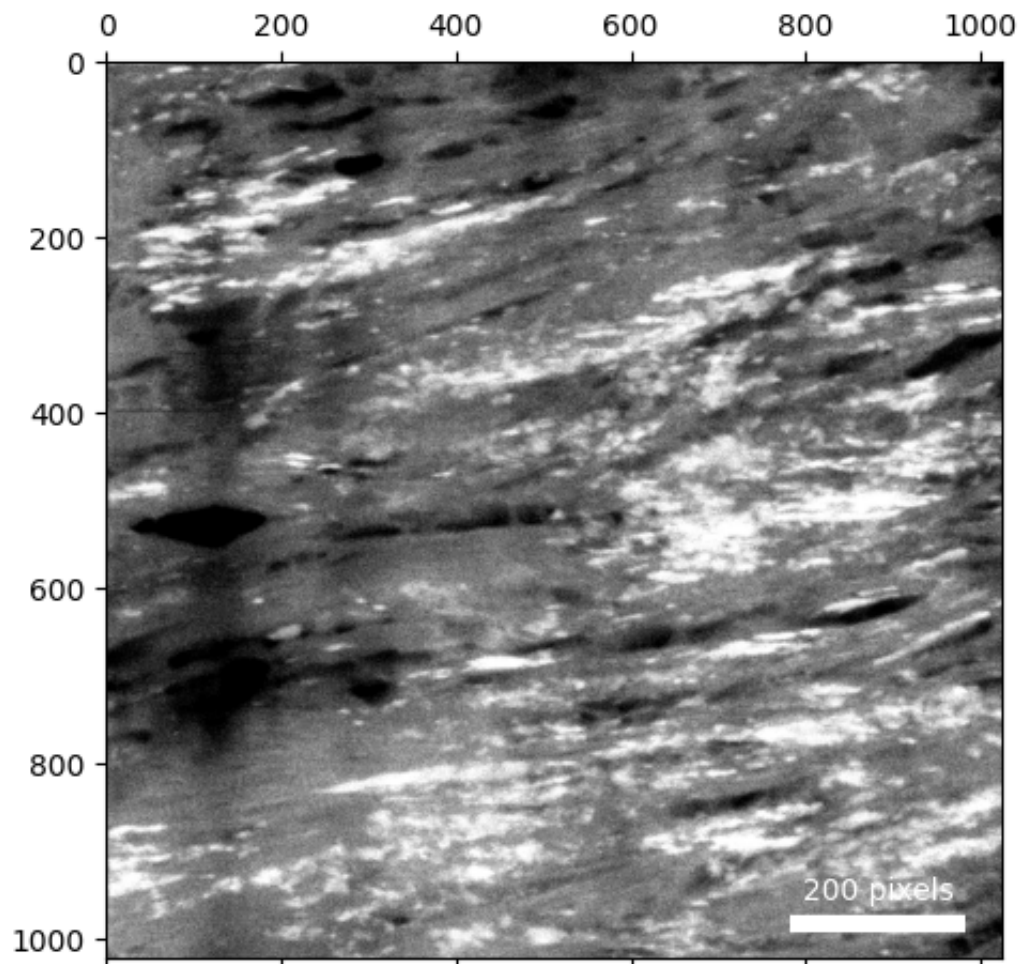
# show
plt.figure(figsize=(6, 6))
plt.imshow(vd.data, cmap="gray")
plt.imshow(moon_mask, cmap="Reds", alpha=0.1)
plt.title("Moon-shaped detector")
plt.show()

# compute DF image
df_moon = dc.get_virtual_image(
    mode="mask",
    geometry=moon_mask,
    name="DF_moon"
)

py4DSTEM.visualize.show(df_moon, scaling="none")
```

100% | 1048576/1048576 [31:20<00:00, 557.65it/s]



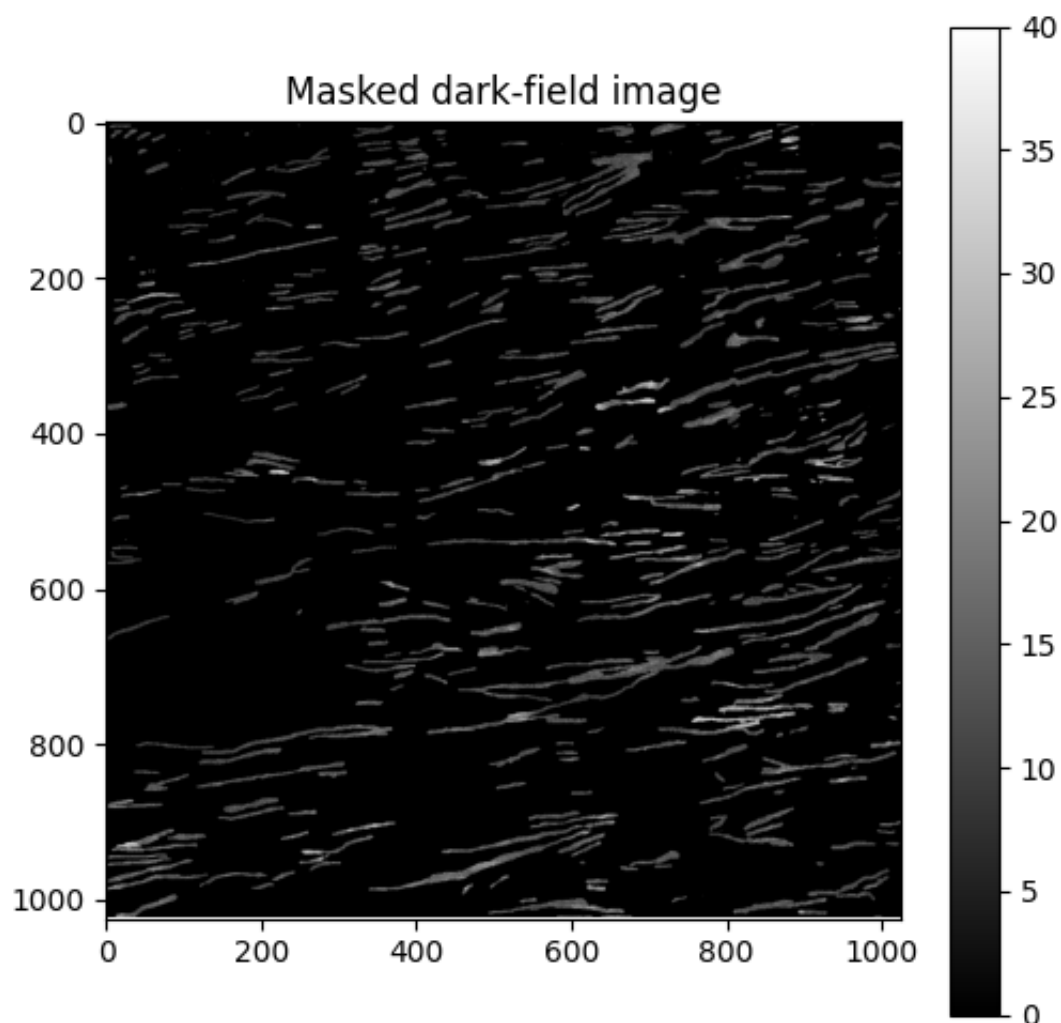
```
In [40]: #Choose the DF image you want to threshold
# Choose the DF image to threshold
# Examples:
# df_source = df_sum
# df_source = df_offaxis.data - if you are using a single off-axis
# df_source = df_moon.data

df_source = df_offaxis.data
```

```
In [41]: #Apply the binary real-space mask

df_masked = np.where(mask_roi, df_source, 0)

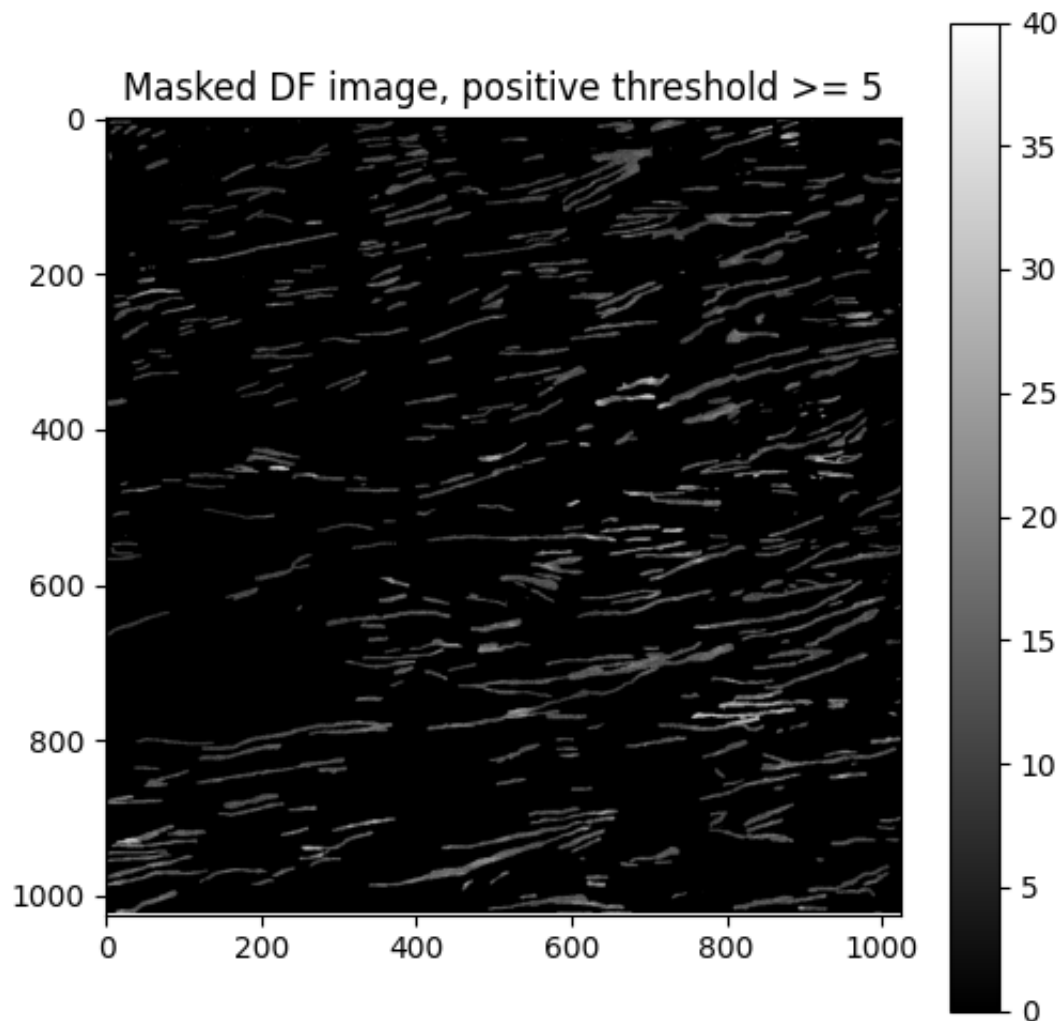
plt.figure(figsize=(6, 6))
plt.imshow(df_masked, cmap="gray")
plt.title("Masked dark-field image")
plt.colorbar()
plt.show()
```



```
In [104... #Positive thresholded DF image
threshold = 5

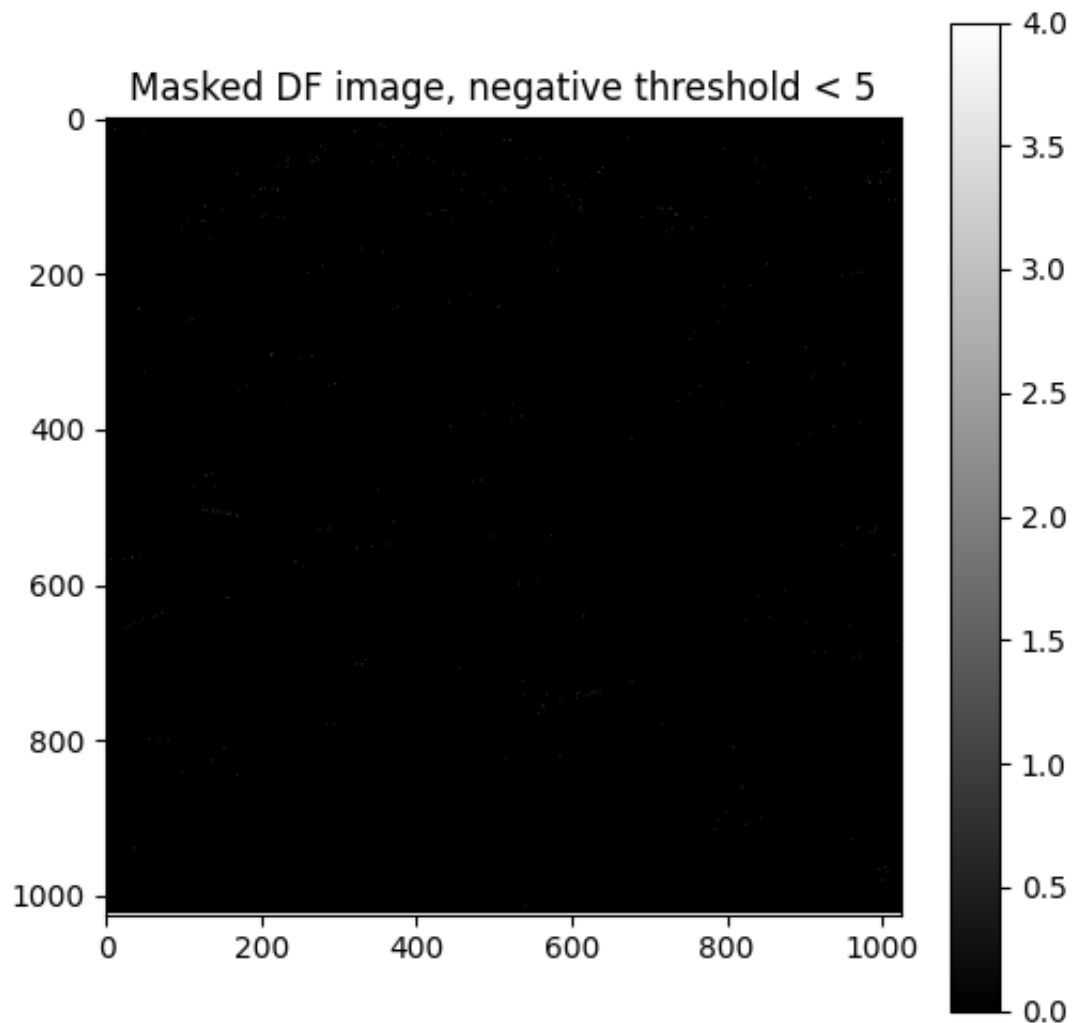
df_masked_thr_pos = np.where(df_masked >= threshold, df_masked, 0)

plt.figure(figsize=(6, 6))
plt.imshow(df_masked_thr_pos, cmap="gray")
plt.title(f"Masked DF image, positive threshold >= {threshold}")
plt.colorbar()
plt.show()
```



```
In [107... #Negative thresholded DF image
df_masked_thr_neg = np.where((df_masked > 0) & (df_masked < thresho

plt.figure(figsize=(6, 6))
plt.imshow(df_masked_thr_neg, cmap="gray")
plt.title(f"Masked DF image, negative threshold < {threshold}")
plt.colorbar()
plt.show()
```



```
In [110... #Boolean mask for positive-threshold region
thr_mask_pos = df_masked_thr_pos > 0

print("thr_mask_pos.shape:", thr_mask_pos.shape)
print("dc.Rshape:", dc.Rshape)
print("selected pixels (positive):", thr_mask_pos.sum())

assert thr_mask_pos.shape == dc.Rshape
```

```
thr_mask_pos.shape: (1024, 1024)
dc.Rshape: (1024, 1024)
selected pixels (positive): 110716
```

```
In [111... #Fraction selected for positive-threshold region
n_selected_pos = thr_mask_pos.sum()
n_total_roi = thr_mask_pos.size
n_total_masked = mask_roi.sum()

print("fraction selected in whole ROI:", n_selected_pos / n_total_r
print("fraction selected within binary mask:", n_selected_pos / n_t
```

```
fraction selected in whole ROI: 0.10558700561523438
fraction selected within binary mask: 0.996750002250691
```

```
In [112... #Diffraction pattern from positive-threshold region

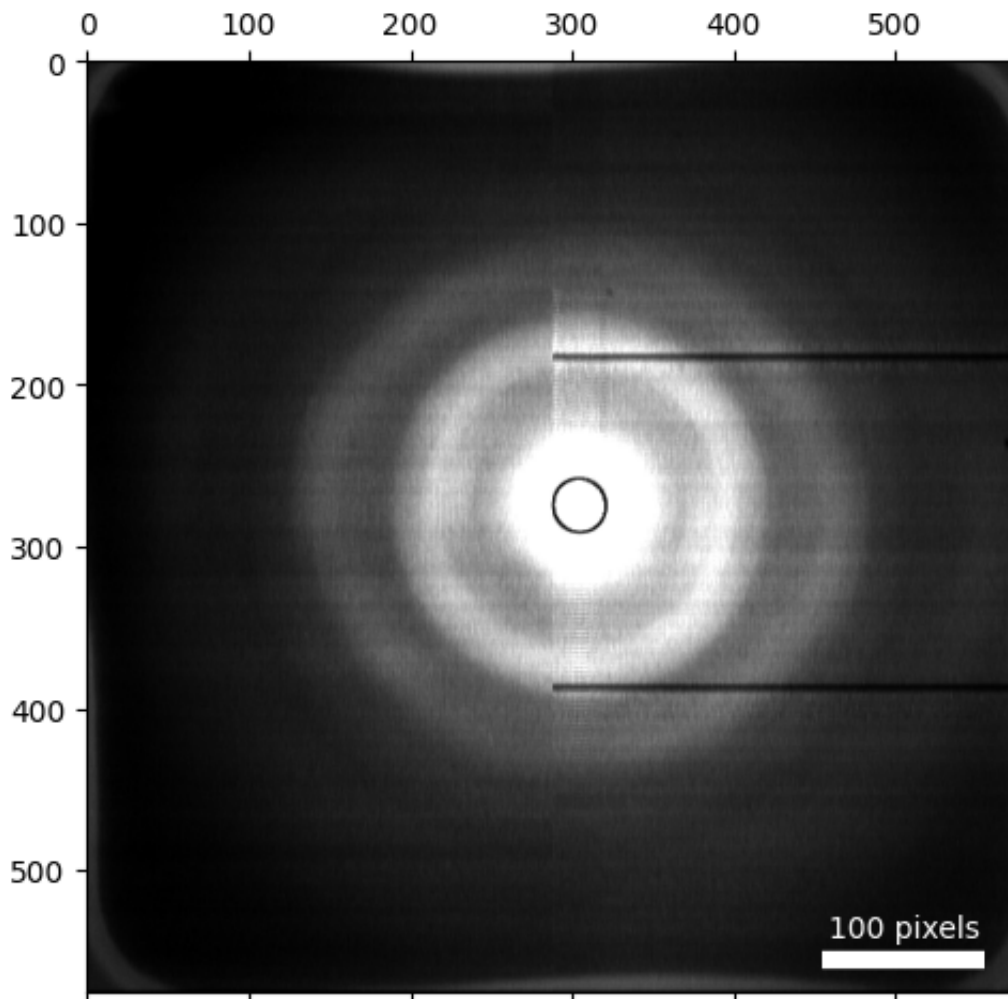
vd_thr_pos = dc.get_virtual_diffraction(
```

```

        method="mean",
        mask=thr_mask_pos,
        name="SS07_R09_Dp01_0FFA01_Df01_DfMMif_Th04_Dp04"
    )

py4DSTEM.visualize.show(vd_thr_pos, scaling="none")

```



```

In [115... #Boolean mask for negative-threshold region
thr_mask_neg = df_masked_thr_neg > 0

print("thr_mask_neg.shape:", thr_mask_neg.shape)
print("dc.Rshape:", dc.Rshape)
print("selected pixels (negative):", thr_mask_neg.sum())

assert thr_mask_neg.shape == dc.Rshape

```

```

thr_mask_neg.shape: (1024, 1024)
dc.Rshape: (1024, 1024)
selected pixels (negative): 308

```

```

In [116... #Fraction selected for negative-threshold region
n_selected_neg = thr_mask_neg.sum()
n_total_roi = thr_mask_neg.size
n_total_masked = mask_roi.sum()

print("fraction selected in whole ROI:", n_selected_neg / n_total_r
print("fraction selected within binary mask:", n_selected_neg / n_t

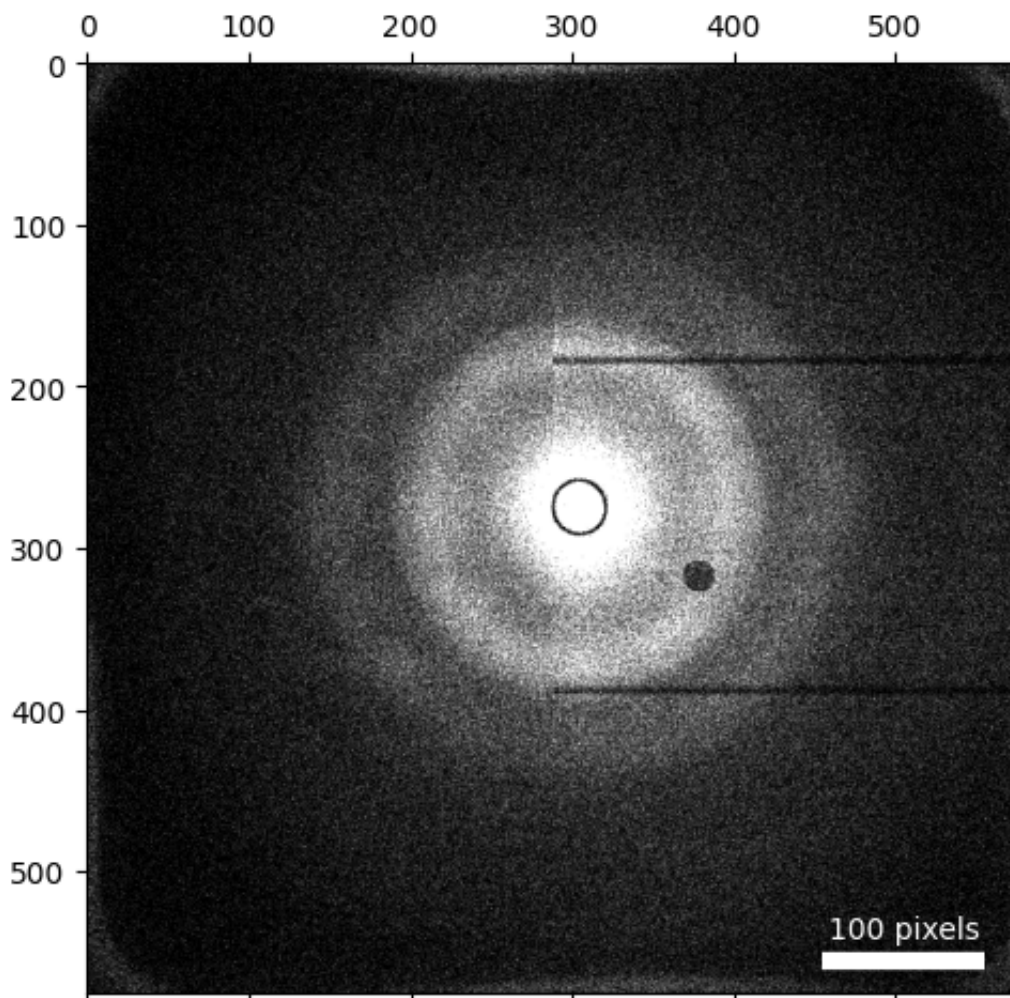
```

fraction selected in whole ROI: 0.000293731689453125

fraction selected within binary mask: 0.002772851265338459

```
In [117]: #Diffraction pattern from negative-threshold region
vd_thr_neg = dc.get_virtual_diffraction(
    method="mean",
    mask=thr_mask_neg,
    name="SS07_R09_Dp01_0FFA01_Df01_DfMMif_Th04_Min04_Dp04"
)

py4DSTEM.visualize.show(vd_thr_neg, scaling="none")
```



```
In [27]: #Multiple Disks
```

```
In [22]: #Define the center and the reference disk

# Recover standard diffraction center coordinates from the py4DSTEM
# center was previously defined as (qy_center, qx_center)
qy_center, qx_center = center

print("Diffraction center (qx, qy):", (qx_center, qy_center))

# Reference disk (standard x, y convention)
qx0_DF, qy0_DF = 378, 318
r_DF = 10
```

Diffraction center (qx, qy): (305, 275)


```
In [47]: #Generate multiple disks on the same ring
# Distance from center to the reference disk

r_ring = np.sqrt((qx0_DF - qx_center)**2 + (qy0_DF - qy_center)**2)

# Angles for the right half of the ring
#angles_deg = [-60, -35, -10, 15, 40, 65]
angles_deg = [-60, -30, 0, 30, 60, 90]

disk_centers = []
for ang in angles_deg:
    theta = np.deg2rad(ang)
    qx = int(round(qx_center + r_ring * np.cos(theta)))
    qy = int(round(qy_center + r_ring * np.sin(theta)))
    disk_centers.append((qx, qy))

print("Disk centers (qx, qy):")
for i, pt in enumerate(disk_centers, start=1):
    print(f"disk {i}: {pt}")
```

```
Disk centers (qx, qy):
disk 1: (347, 202)
disk 2: (378, 233)
disk 3: (390, 275)
disk 4: (378, 317)
disk 5: (347, 348)
disk 6: (305, 360)
```

```
In [26]: #Show the disks on the diffraction pattern

# diffraction image
dp_img = np.asarray(vd.data, dtype=float)

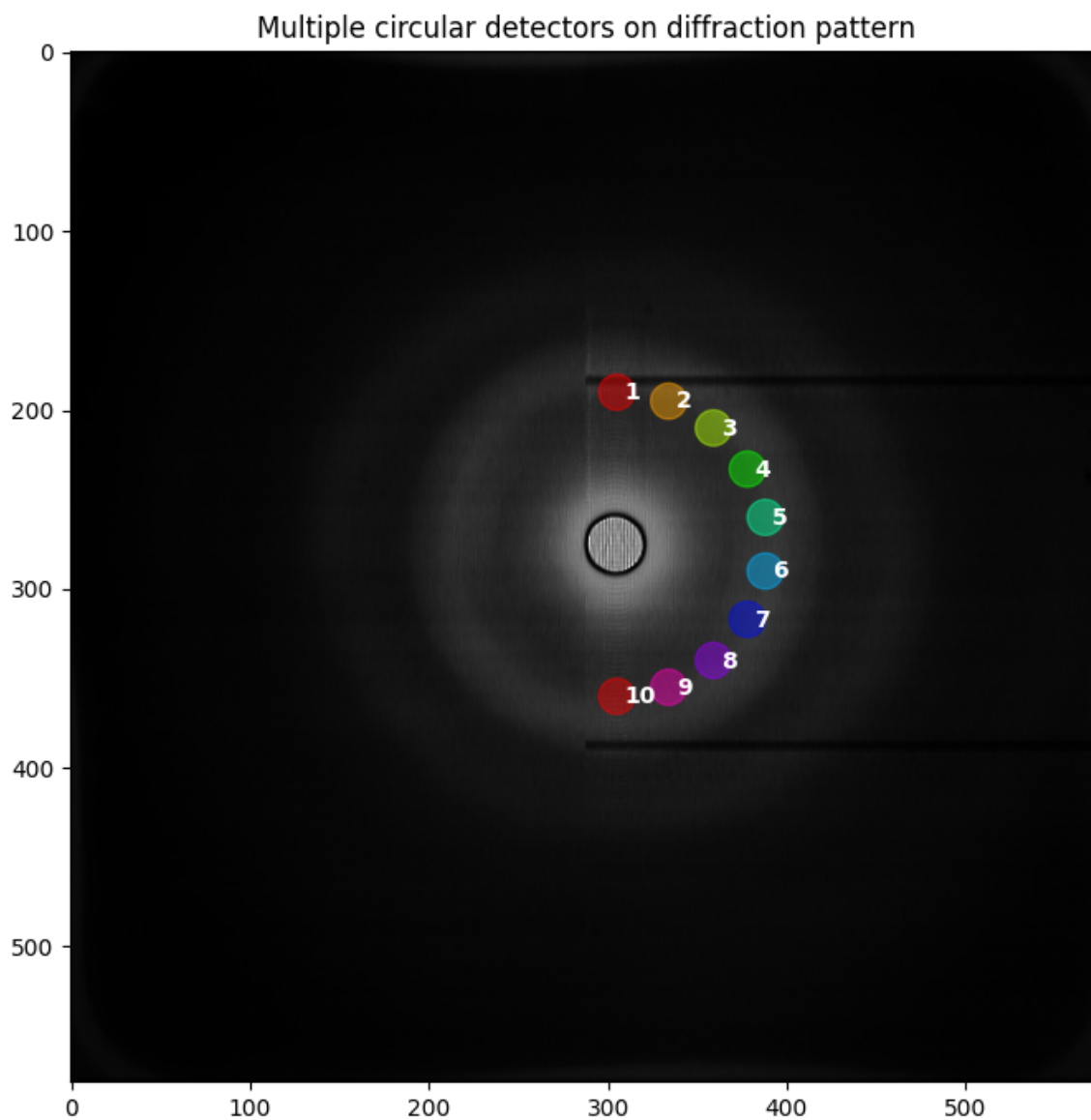
n_disks = len(disk_centers)

# generate colors automatically
# make n_disks-1 unique colors, then repeat the first for the last
cmap = plt.get_cmap("hsv") # cyclic colormap
# cmap = plt.get_cmap("turbo") # maybe nicer colors
# cmap = plt.get_cmap("gist_rainbow")
base_colors = cmap(np.linspace(0, 1, n_disks - 1, endpoint=False))
colors = np.vstack([base_colors, base_colors[0]]) # last = first

fig, ax = plt.subplots(figsize=(7, 7))
ax.imshow(dp_img, cmap="gray")

for i, ((qx, qy), color) in enumerate(zip(disk_centers, colors), start=1):
    circ = Circle((qx, qy), r_DF, fill=True, alpha=0.45, color=colors[i])
    ax.add_patch(circ)
    ax.text(qx + 4, qy + 4, str(i), color="white", fontsize=10, weight='bold')

ax.set_title("Multiple circular detectors on diffraction pattern")
plt.tight_layout()
plt.show()
```

```
In [27]: #Compute one DF image per disk

dfs = []

for i, (qx, qy) in enumerate(disk_centers, start=1):
    geometry = (
        (qy, qx), # py4DSTEM circle geometry uses (y, x)
        r_DF
    )

    df_i = dc.get_virtual_image(
        mode="circle",
        geometry=geometry,
        name=f"DF_disk_{i}"
    )
    dfs.append(df_i)
```

```

100%|██████████| 1048576/1048576 [20:57<00:00, 833.73it/s]
100%|██████████| 1048576/1048576 [21:37<00:00, 808.44it/s]
100%|██████████| 1048576/1048576 [23:32<00:00, 742.54it/s]
100%|██████████| 1048576/1048576 [24:09<00:00, 723.25it/s]
100%|██████████| 1048576/1048576 [25:46<00:00, 678.19it/s]
100%|██████████| 1048576/1048576 [26:20<00:00, 663.24it/s]
100%|██████████| 1048576/1048576 [26:39<00:00, 655.54it/s]
100%|██████████| 1048576/1048576 [25:25<00:00, 687.33it/s]
100%|██████████| 1048576/1048576 [27:48<00:00, 628.49it/s]
100%|██████████| 1048576/1048576 [28:00<00:00, 623.98it/s]

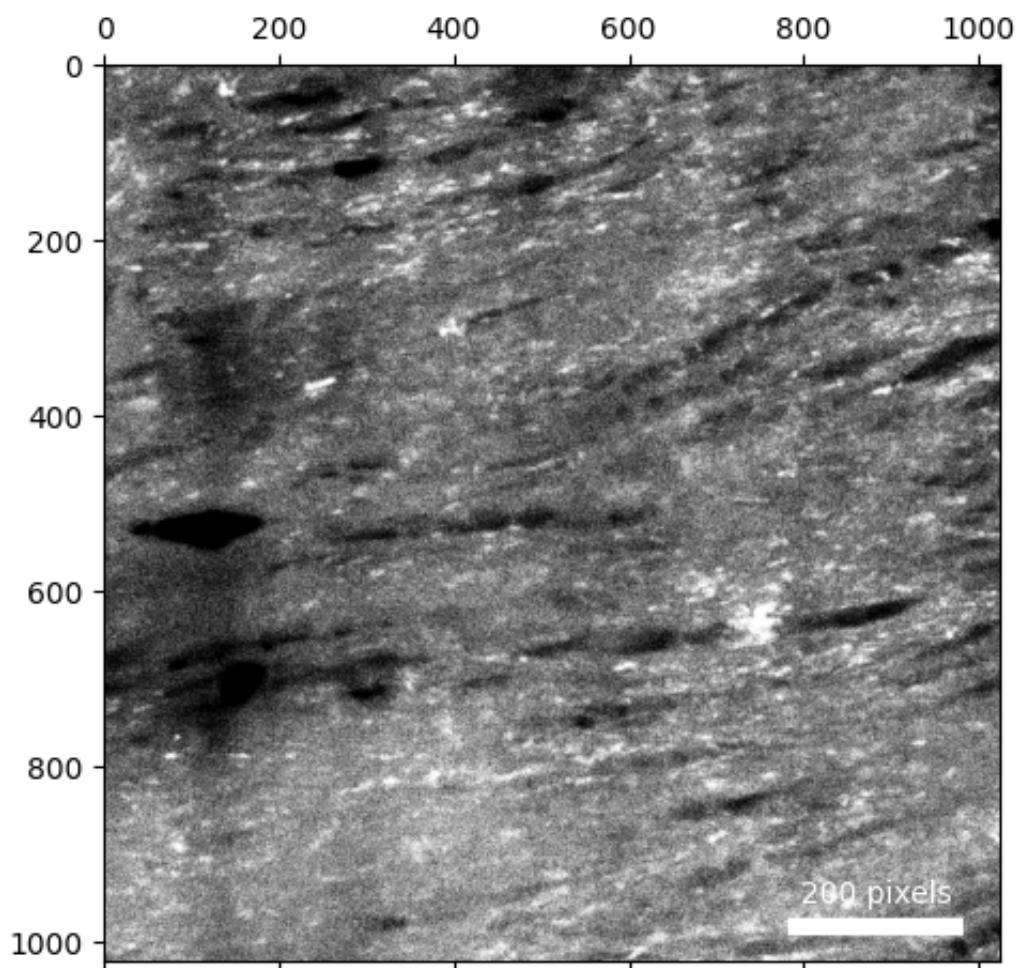
```

```

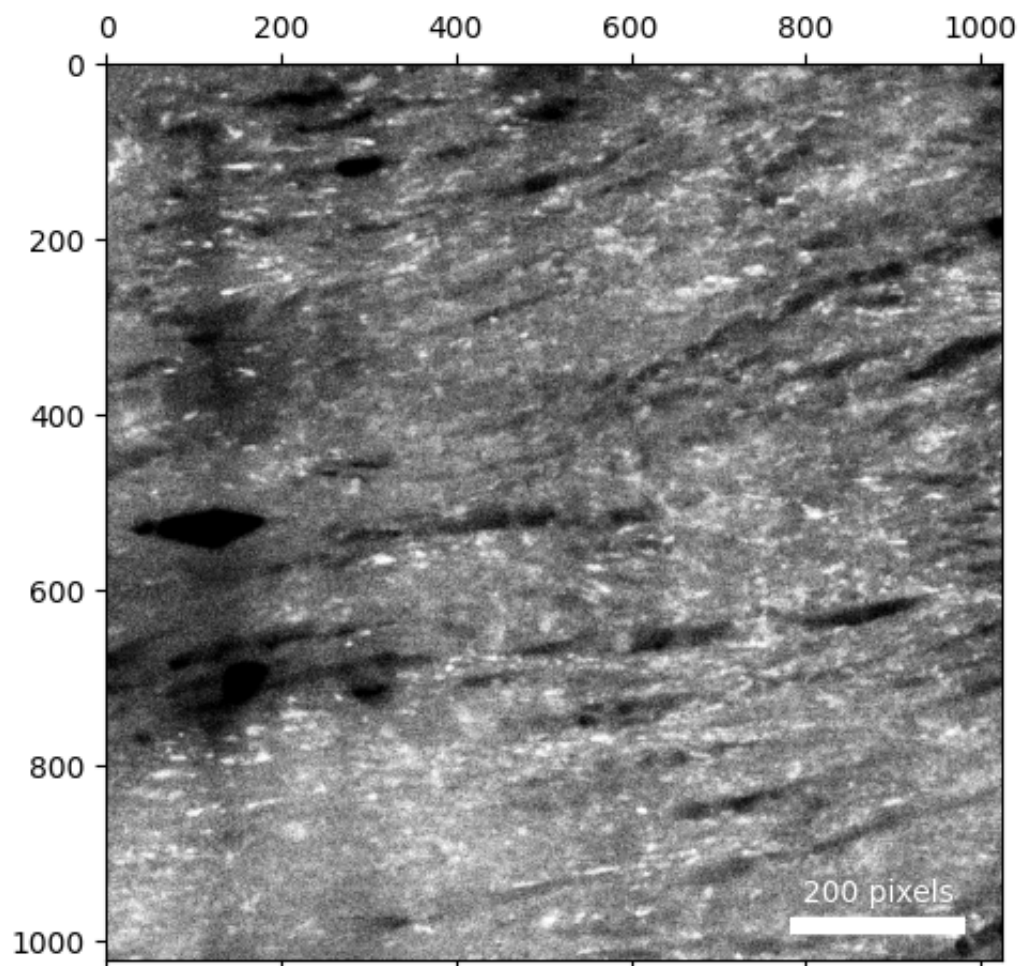
In [28]: #Display each DF image
for i, df_i in enumerate(dfs, start=1):
    print(f"Dark-field image for disk {i} at center {disk_centers[i]}")
    py4DSTEM.visualize.show(df_i, scaling="none")

```

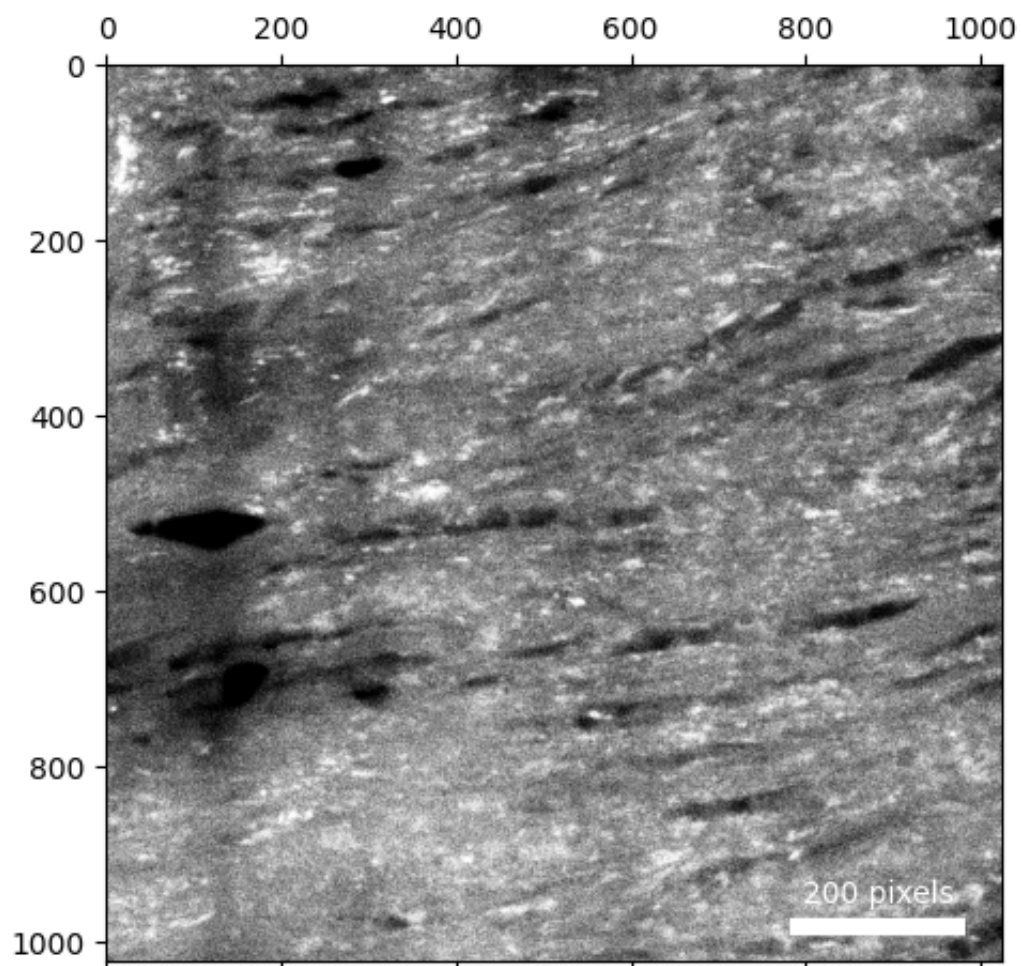
Dark-field image for disk 1 at center (305, 190)



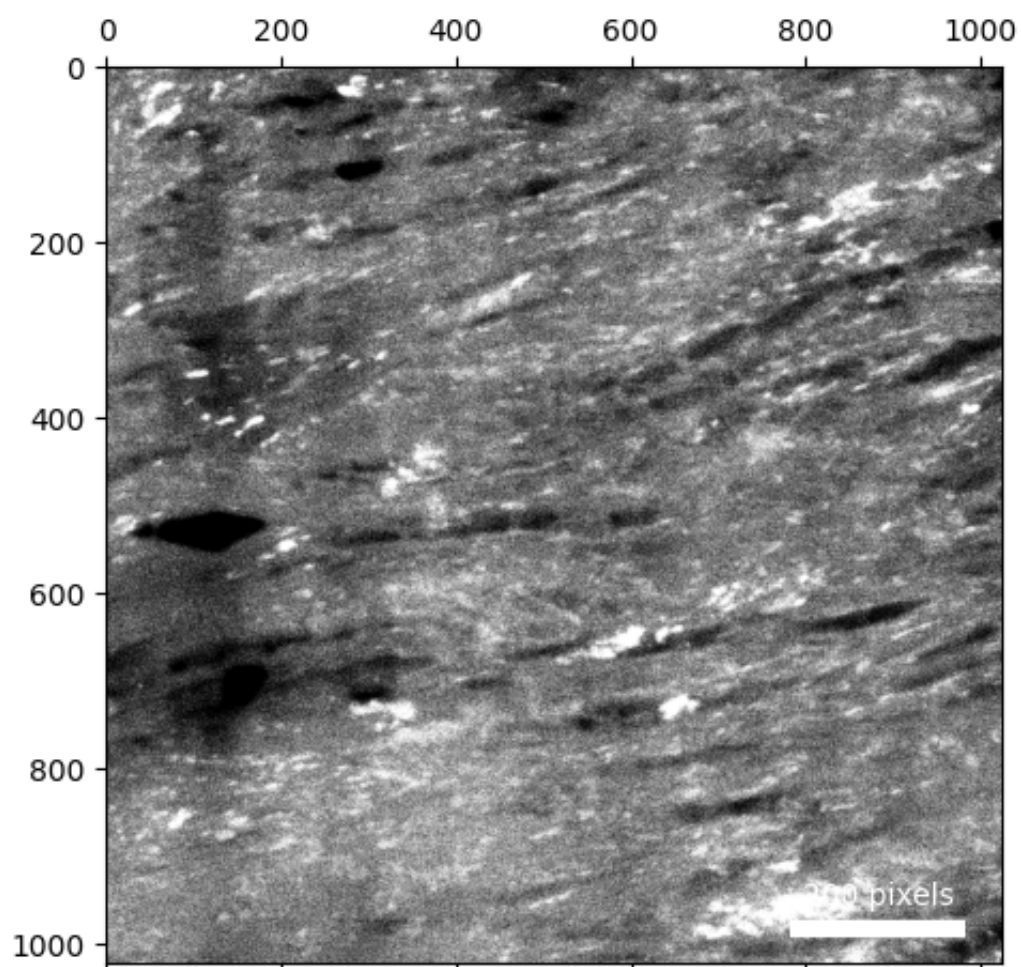
Dark-field image for disk 2 at center (334, 195)



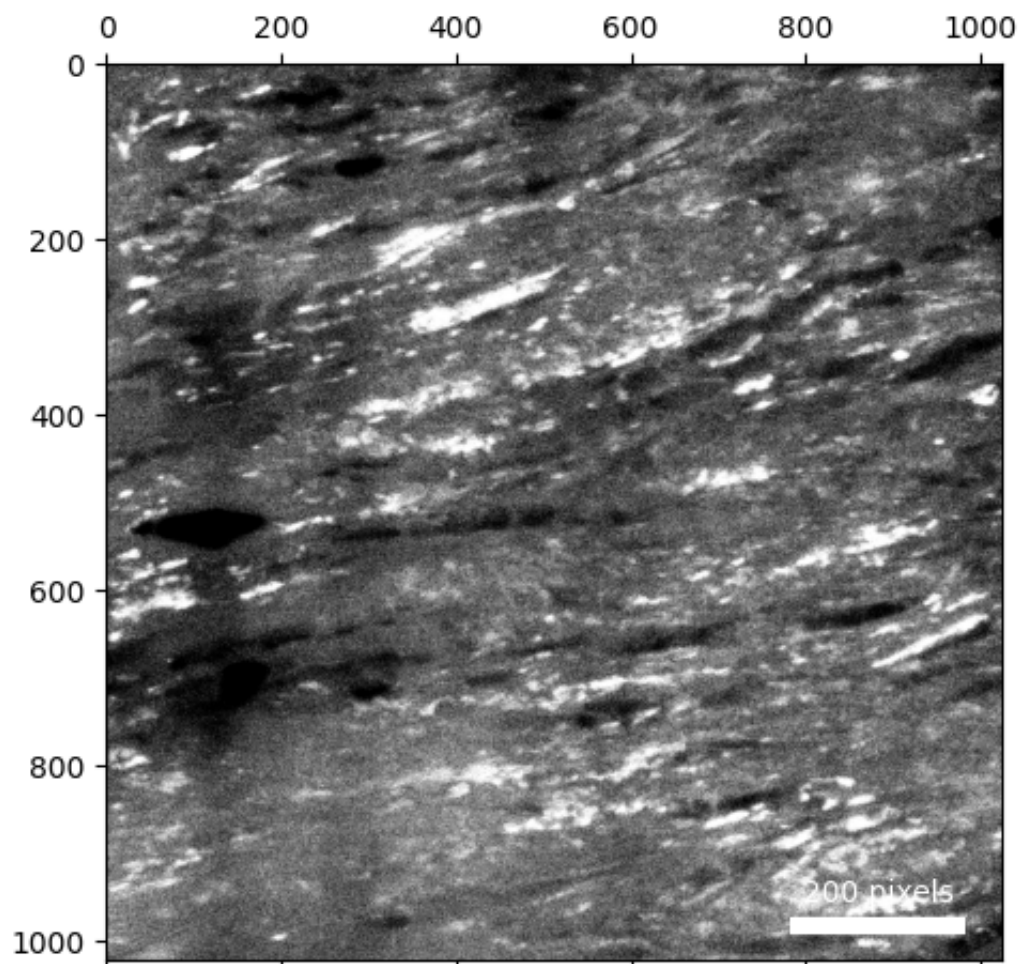
Dark-field image for disk 3 at center (359, 210)



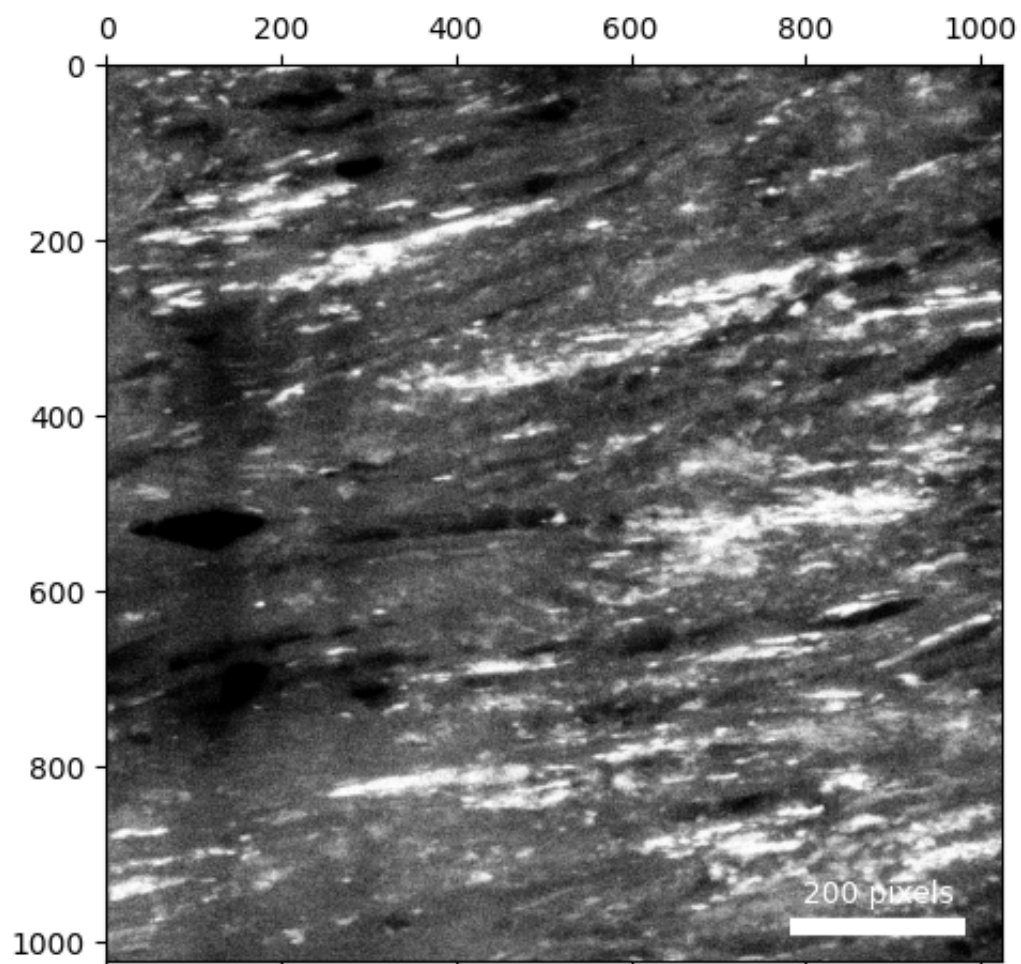
Dark-field image for disk 4 at center (378, 233)



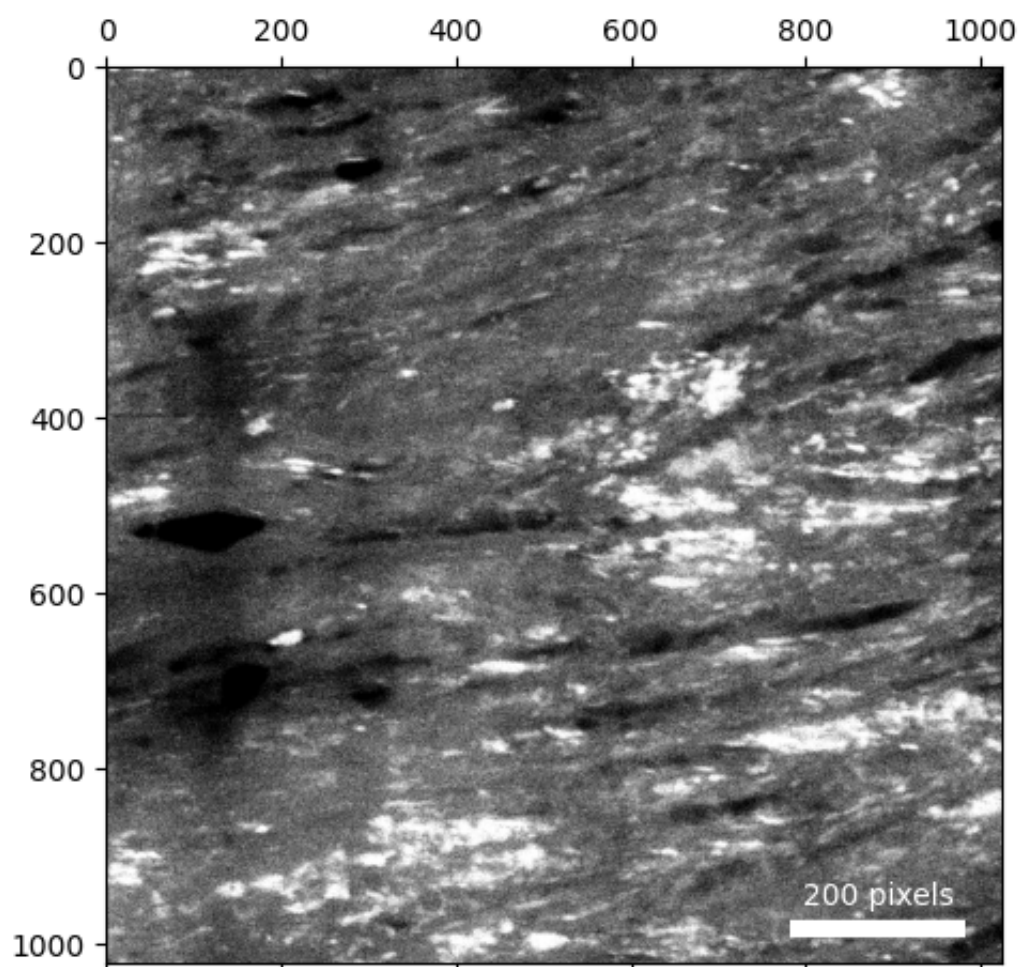
Dark-field image for disk 5 at center (388, 260)



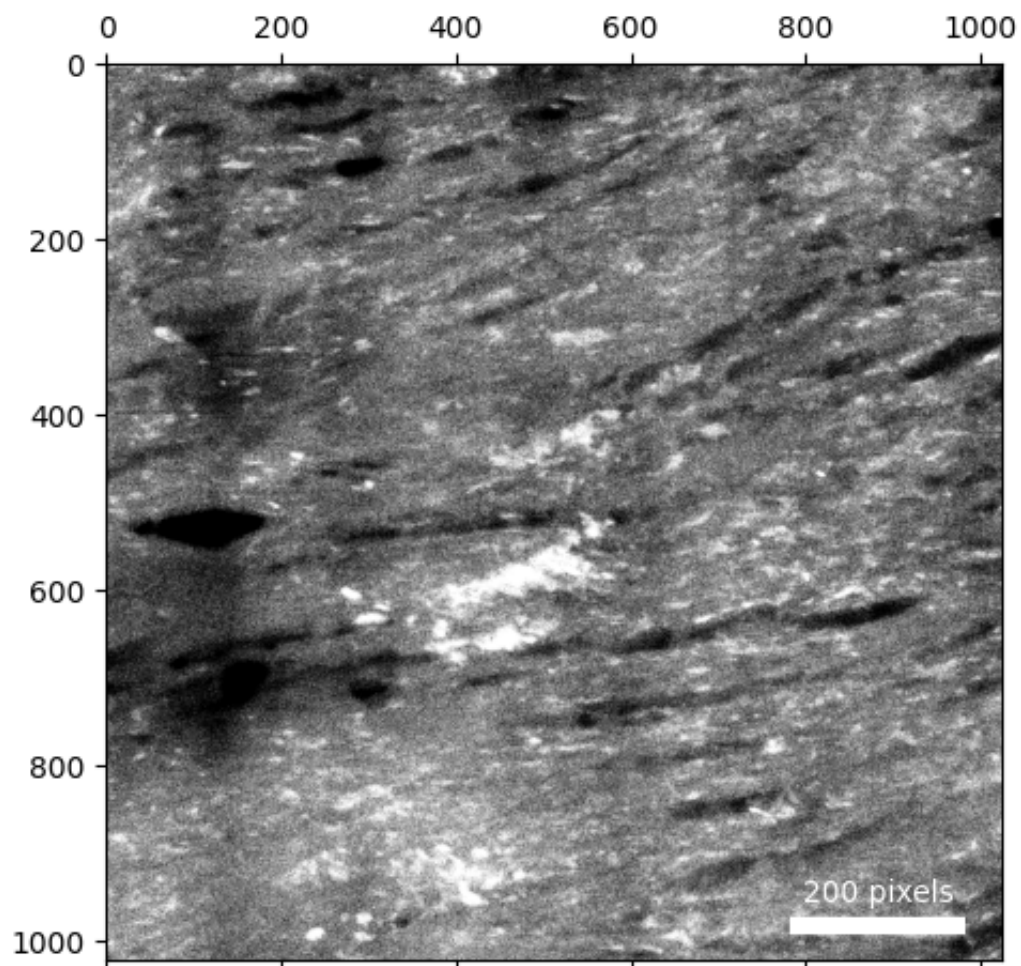
Dark-field image for disk 6 at center (388, 290)



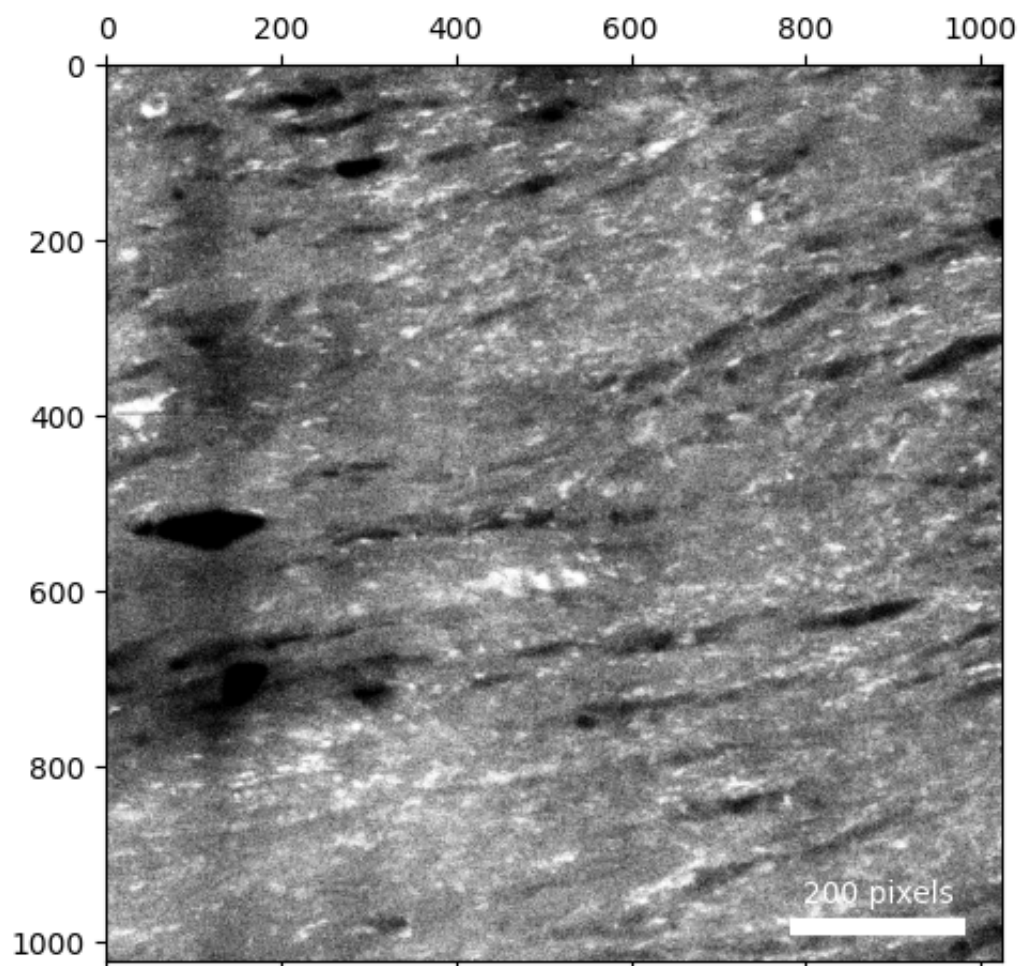
Dark-field image for disk 7 at center (378, 317)



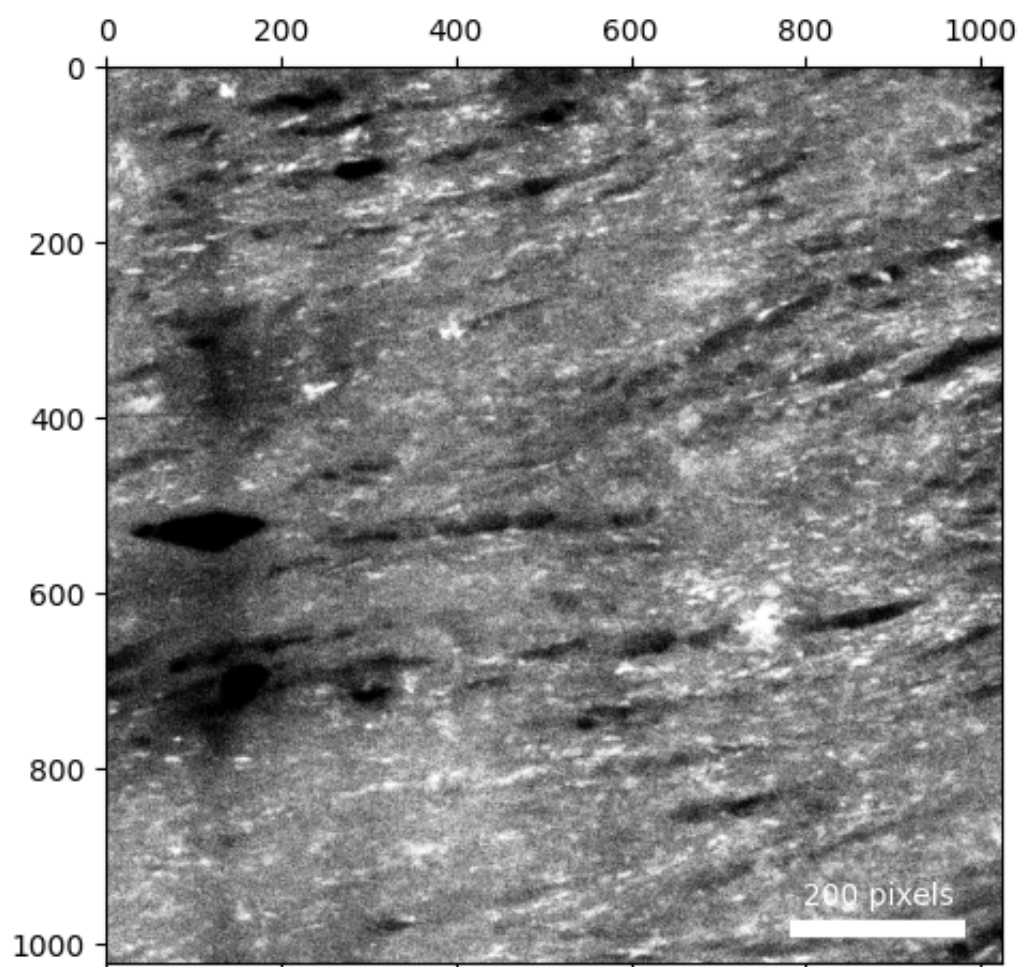
Dark-field image for disk 8 at center (359, 340)



Dark-field image for disk 9 at center (334, 355)



Dark-field image for disk 10 at center (305, 360)

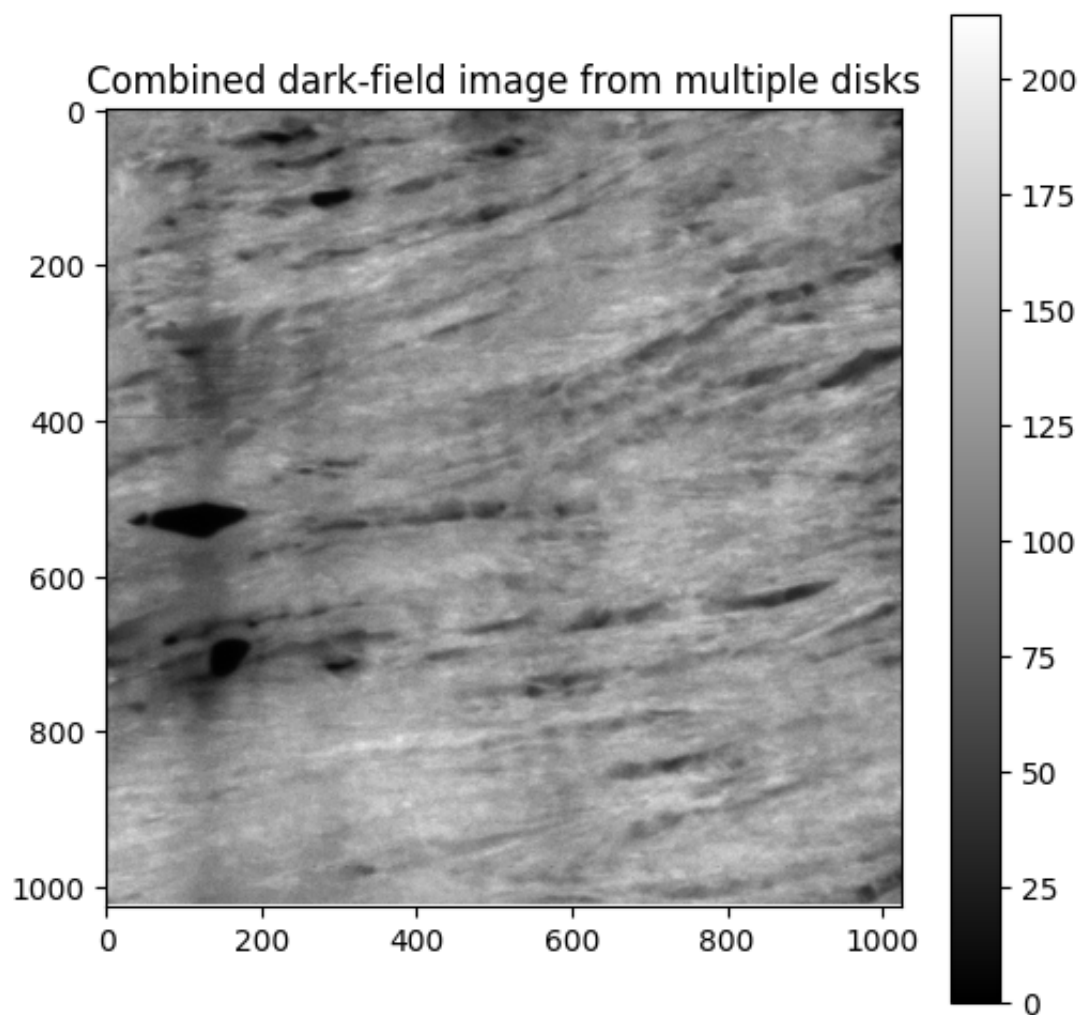


```
In [31]: #Sum all DF images
df_sum = np.zeros_like(np.asarray(dfs[0].data), dtype=float)

for df_i in dfs:
    df_sum += np.asarray(df_i.data, dtype=float)
```

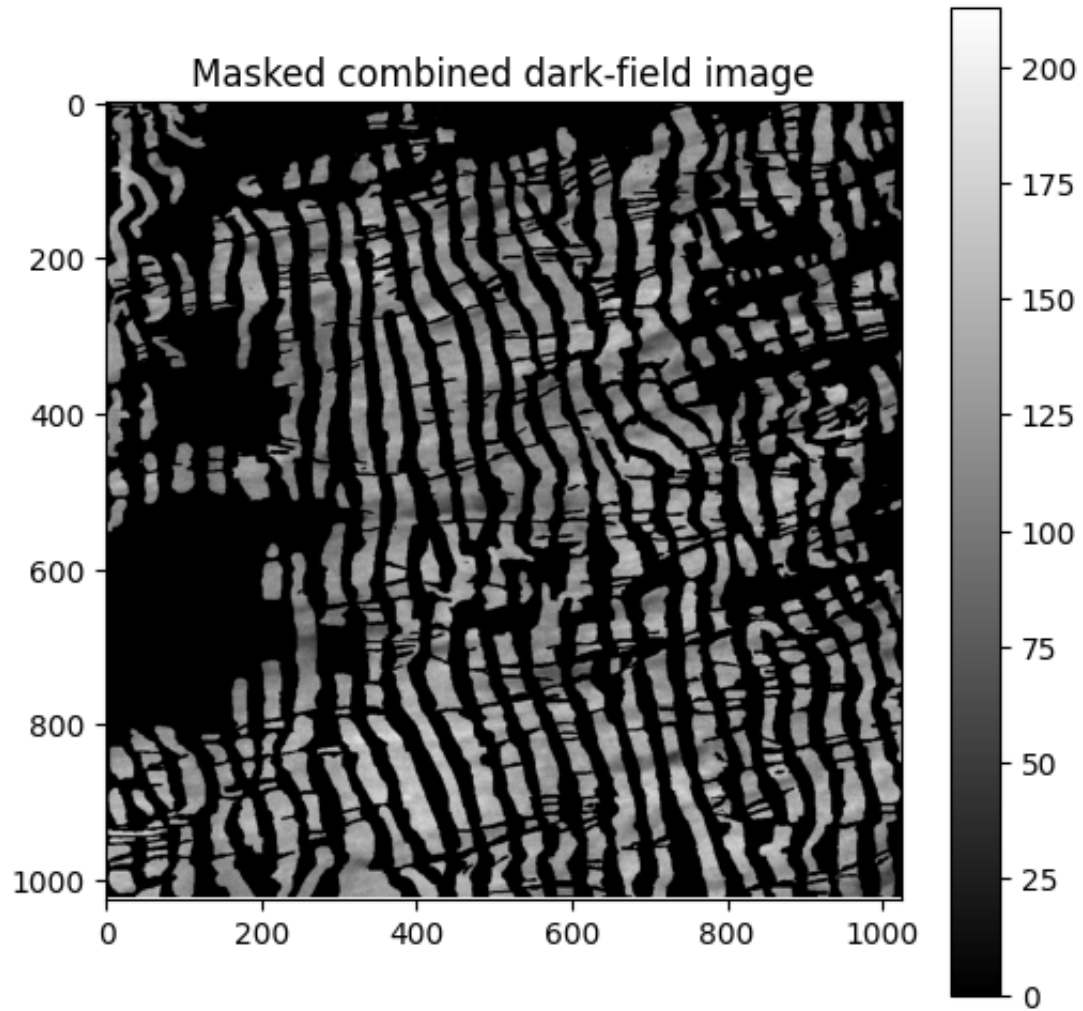
```
In [32]: #Display the summed DF image

plt.figure(figsize=(6, 6))
plt.imshow(df_sum, cmap="gray")
plt.colorbar()
plt.title("Combined dark-field image from multiple disks")
plt.show()
```

```
In [34]: #Apply the real-space binary mask (optional but likely useful)
df_sum_masked = np.where(mask_roi, df_sum, 0)

plt.figure(figsize=(6, 6))
plt.imshow(df_sum_masked, cmap="gray")
plt.colorbar()
plt.title("Masked combined dark-field image")
plt.show()
```



```
In [35]: #Adaptive colors - winner-takes-all
         # maybe a better version of (less noisier)

         # =====
         # INPUTS
         # =====

         # Better if you replace this with your HAADF/ADF-like grayscale image
         base_img = np.asarray(df_sum, dtype=float) # or haadf_roi if you

         # Diffraction pattern for panel C
         dp_img = np.asarray(vd.data, dtype=float) # or vd_masked.data

         # Stack DF images
         df_stack = np.stack([np.asarray(df_i.data, dtype=float) for df_i in
                                n_det, H, W = df_stack.shape

         # =====
         # COLORS: HSV, with first and last disk having the same color
         # =====
         cmap = plt.get_cmap("hsv")

         if n_det >= 2:
             base_colors = cmap(np.linspace(0, 1, n_det - 1, endpoint=False))
             colors = np.vstack([base_colors, base_colors[0]]) # last color
         else:
```

```

        colors = cmap(np.linspace(0, 1, n_det, endpoint=False))[:, :3]

# =====
# STEP 1: LIGHT SMOOTHING
# =====
# Helps connect patches and reduce salt-and-pepper noise
sigma = 1.2
df_smooth = np.stack([gaussian_filter(arr, sigma=sigma) for arr in

# =====
# STEP 2: ROBUST NORMALIZATION PER DETECTOR
# =====
norm_stack = np.zeros_like(df_smooth)

for i in range(n_det):
    arr = df_smooth[i]
    lo = np.percentile(arr, 95.0)
    hi = np.percentile(arr, 99.8)
    if hi <= lo:
        hi = lo + 1e-12
    norm_stack[i] = np.clip((arr - lo) / (hi - lo), 0, 1)

# =====
# STEP 3: DOMINANT DETECTOR + CONFIDENCE
# =====
order = np.argsort(norm_stack, axis=0)

label_map = order[-1] # strongest detector index
best = np.take_along_axis(norm_stack, order[-1:, :, :], axis=0)[0]
second = np.take_along_axis(norm_stack, order[-2:-1, :, :], axis=0)

# Difference between strongest and second strongest detector
margin = best - second

# Only keep pixels that are both strong and clearly assigned
strength_thresh = 0.22
margin_thresh = 0.08

valid = (best >= strength_thresh) & (margin >= margin_thresh)

# =====
# STEP 4: BUILD RGB OVERLAY
# =====
base_lo = np.percentile(base_img, 1.0)
base_hi = np.percentile(base_img, 99.5)
if base_hi <= base_lo:
    base_hi = base_lo + 1e-12

base_norm = np.clip((base_img - base_lo) / (base_hi - base_lo), 0,
base_rgb = np.dstack([base_norm, base_norm, base_norm])

overlay_rgb = np.zeros((H, W, 3), dtype=float)
for i in range(n_det):
    overlay_rgb[label_map == i] = colors[i]

# Alpha depends on strength, but only where valid

```

```

overlay_alpha = np.zeros((H, W), dtype=float)
overlay_alpha[valid] = np.clip((best[valid] - strength_thresh) / (1
overlay_alpha *= 0.90

# Blend
blended = (1 - overlay_alpha)[..., None] * base_rgb + overlay_alpha

# =====
# STEP 5: SHOW A, B, C
# =====
fig, axes = plt.subplots(1, 3, figsize=(18, 6))

# Panel A
ax = axes[0]
ax.imshow(base_img, cmap="gray")
ax.set_title("A")
ax.axis("off")

# Panel B
ax = axes[1]
ax.imshow(blended)
ax.set_title("B")
ax.axis("off")

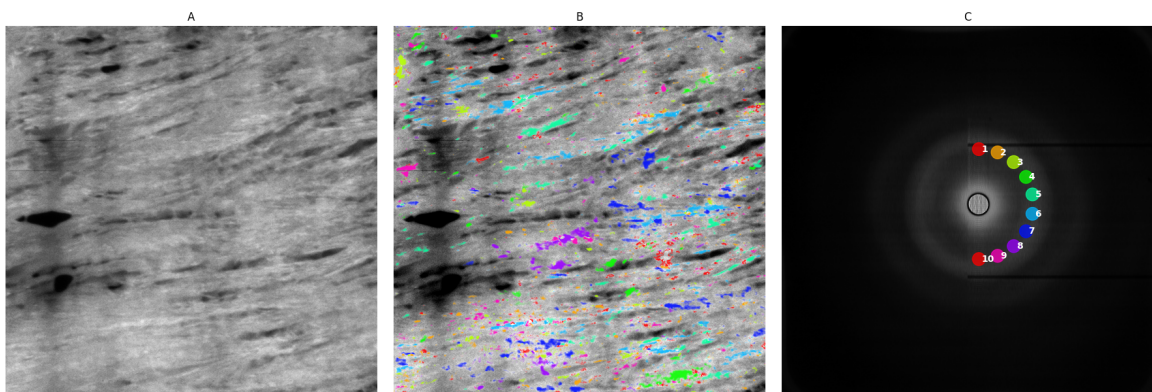
# Panel C
ax = axes[2]
ax.imshow(dp_img, cmap="gray")

for i, ((qx, qy), color) in enumerate(zip(disk_centers, colors), st
    ax.add_patch(Circle((qx, qy), r_DF, fill=True, alpha=0.75, colo
    ax.text(qx + 4, qy + 4, str(i), color="white", fontsize=10, wei

ax.set_title("C")
ax.axis("off")

plt.tight_layout()
plt.show()

```



In [38]: #Adaptive color HAADF overlay image
HAADF base image

```

# =====
# INPUTS
# =====

```

```

# If no HAADF ROI is available, fall back to df_sum
if base_img is None:
    base_img = np.asarray(df_sum, dtype=float)

# Diffraction pattern for panel C
dp_img = np.asarray(vd.data, dtype=float) # or vd_masked.data

# Stack DF images
df_stack = np.stack([np.asarray(df_i.data, dtype=float) for df_i in
n_det, H, W = df_stack.shape

# =====
# HSV COLORS, with first and last detector having same color
# =====
cmap = plt.get_cmap("hsv")

if n_det >= 2:
    base_colors = cmap(np.linspace(0, 1, n_det - 1, endpoint=False))
    colors = np.vstack([base_colors, base_colors[0]]) # last = fi
else:
    colors = cmap(np.linspace(0, 1, n_det, endpoint=False))[:, :3]

# =====
# STEP 1: LIGHT SMOOTHING
# =====
sigma = 1.2
df_smooth = np.stack([gaussian_filter(arr, sigma=sigma) for arr in

# =====
# STEP 2: ROBUST NORMALIZATION PER DETECTOR
# =====
norm_stack = np.zeros_like(df_smooth)

for i in range(n_det):
    arr = df_smooth[i]
    lo = np.percentile(arr, 95.0)
    hi = np.percentile(arr, 99.8)
    if hi <= lo:
        hi = lo + 1e-12
    norm_stack[i] = np.clip((arr - lo) / (hi - lo), 0, 1)

# =====
# STEP 3: DOMINANT DETECTOR + CONFIDENCE
# =====
order = np.argsort(norm_stack, axis=0)

label_map = order[-1]
best = np.take_along_axis(norm_stack, order[-1:, :, :], axis=0)[0]
second = np.take_along_axis(norm_stack, order[-2:-1, :, :], axis=0)

margin = best - second

strength_thresh = 0.22
margin_thresh = 0.08

```

```

valid = (best >= strength_thresh) & (margin >= margin_thresh)

# =====
# STEP 4: BUILD RGB OVERLAY
# =====
base_lo = np.percentile(base_img, 1.0)
base_hi = np.percentile(base_img, 99.5)
if base_hi <= base_lo:
    base_hi = base_lo + 1e-12

base_norm = np.clip((base_img - base_lo) / (base_hi - base_lo), 0,
base_rgb = np.dstack([base_norm, base_norm, base_norm])

overlay_rgb = np.zeros((H, W, 3), dtype=float)
for i in range(n_det):
    overlay_rgb[label_map == i] = colors[i]

overlay_alpha = np.zeros((H, W), dtype=float)
overlay_alpha[valid] = np.clip((best[valid] - strength_thresh) / (1
overlay_alpha *= 0.90

blended = (1 - overlay_alpha)[..., None] * base_rgb + overlay_alpha

# =====
# STEP 5: SHOW A, B, C
# =====
fig, axes = plt.subplots(1, 3, figsize=(18, 6))

# Panel A
ax = axes[0]
ax.imshow(base_img, cmap="gray")
ax.set_title("A")
ax.axis("off")

# Panel B
ax = axes[1]
ax.imshow(blended)
ax.set_title("B")
ax.axis("off")

# Panel C
ax = axes[2]
ax.imshow(dp_img, cmap="gray")

for i, ((qx, qy), color) in enumerate(zip(disk_centers, colors), st
    ax.add_patch(Circle((qx, qy), r_DF, fill=True, alpha=0.75, colo
    ax.text(qx + 4, qy + 4, f"{i}", color="white", fontsize=10, wei

ax.set_title("C")
ax.axis("off")

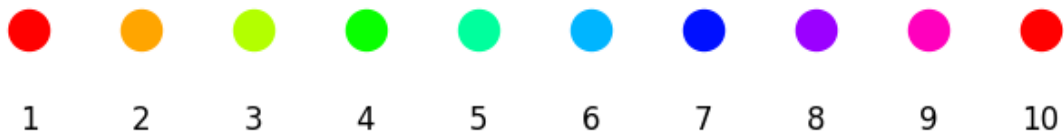
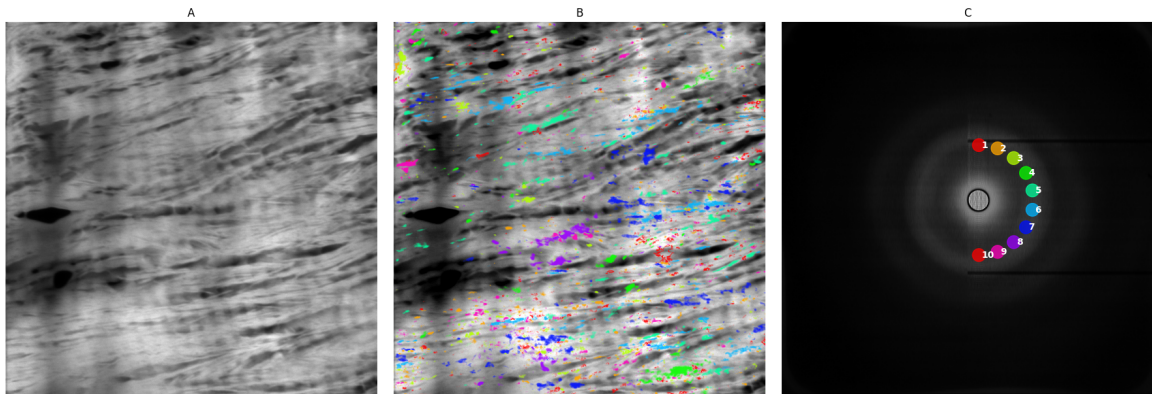
plt.tight_layout()
plt.show()

# =====
# OPTIONAL: COLOR LEGEND

```

```
# =====
fig, ax = plt.subplots(figsize=(8, 1.5))
for i, color in enumerate(colors, start=1):
    ax.scatter(i, 1, s=250, color=color)
    ax.text(i, 0.72, f"{i}", ha="center", va="top", fontsize=12)

ax.set_xlim(0.5, len(colors) + 0.5)
ax.set_ylim(0.5, 1.3)
ax.axis("off")
plt.show()
```



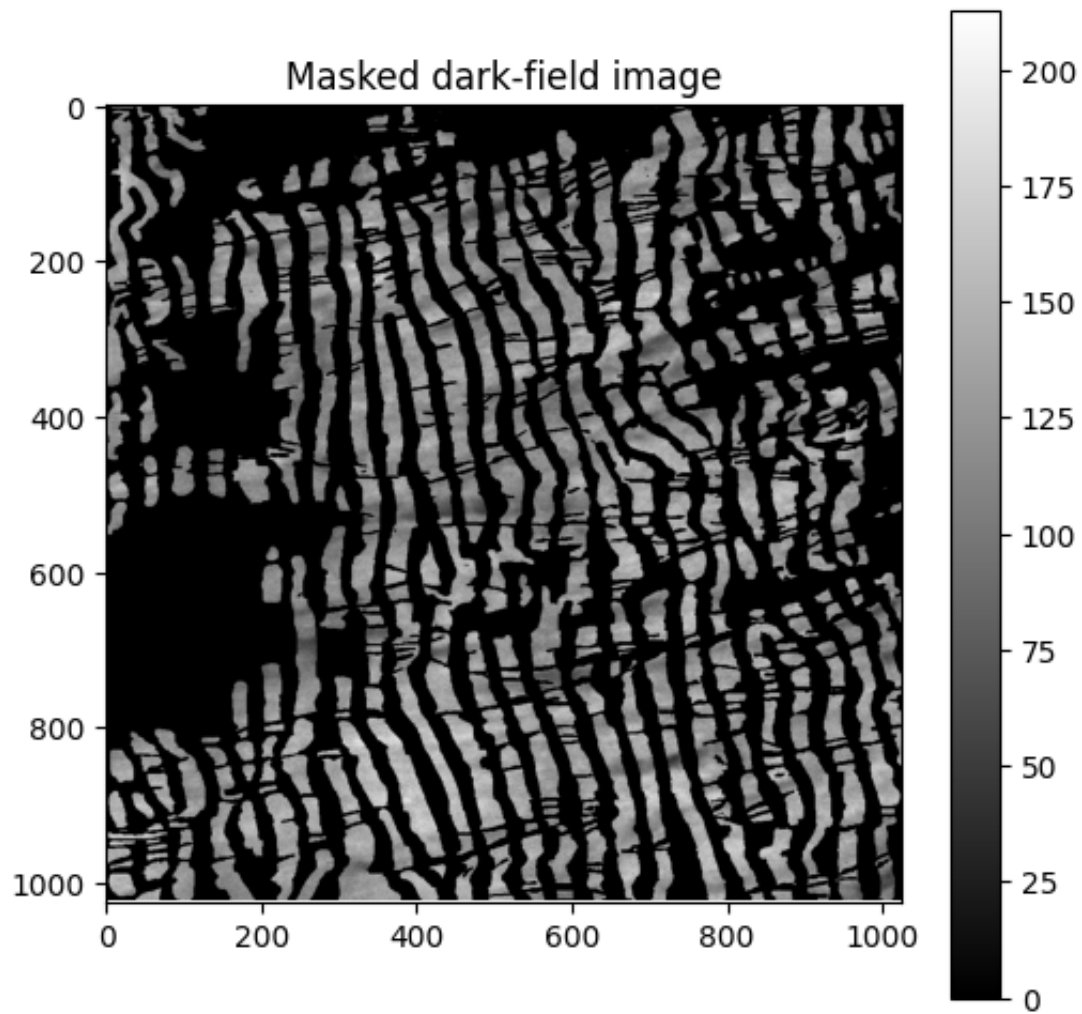
```
In [41]: #Choose the DF image you want to threshold
# Choose the DF image to threshold
# Examples:
# df_source = df_sum
# df_source = df_offaxis.data - if you are using a single off-axis
# df_source = df_moon.data

df_source = df_sum
```

```
In [43]: #Apply the binary real-space mask

df_masked = np.where(mask_roi, df_source, 0)

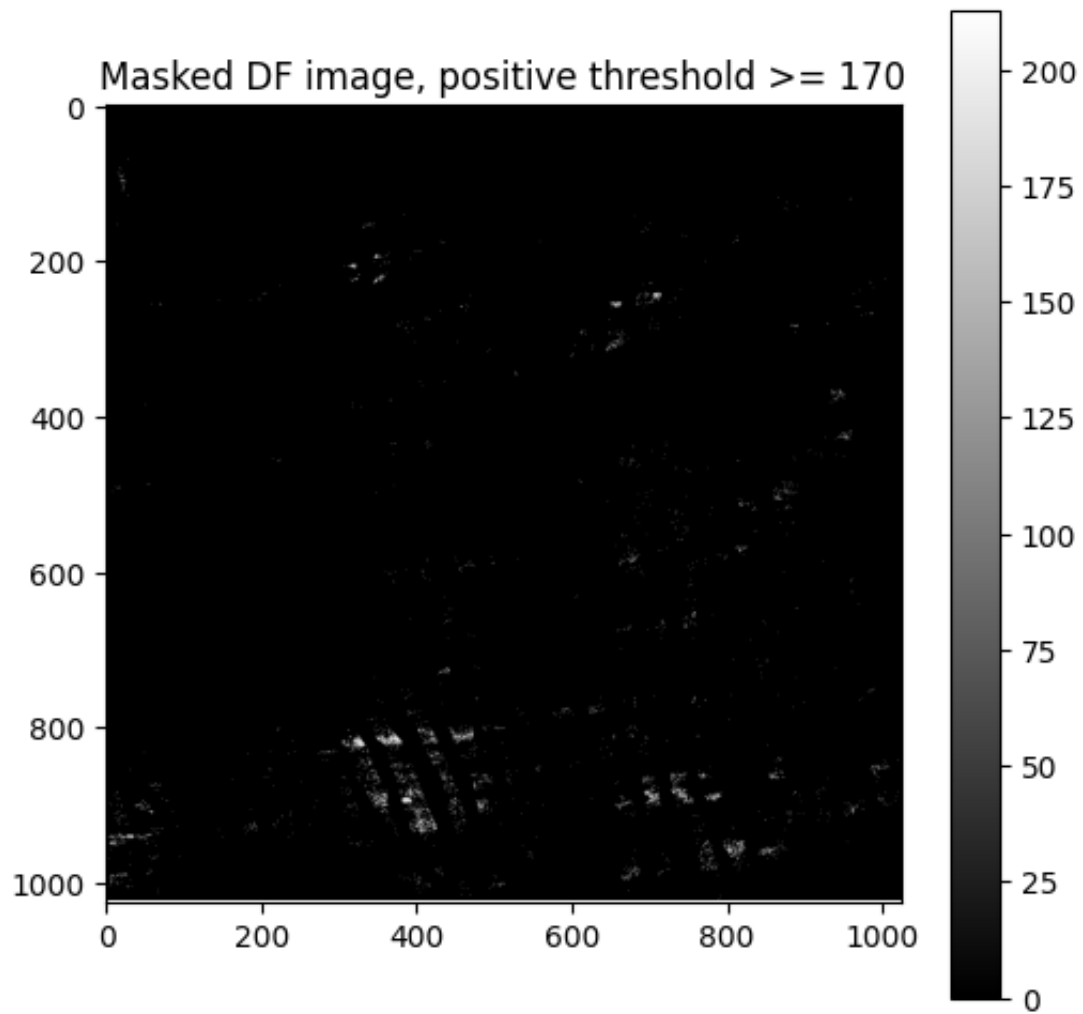
plt.figure(figsize=(6, 6))
plt.imshow(df_masked, cmap="gray")
plt.title("Masked dark-field image")
plt.colorbar()
plt.show()
```

```
In [47]: #Positive thresholded DF image
threshold = 170

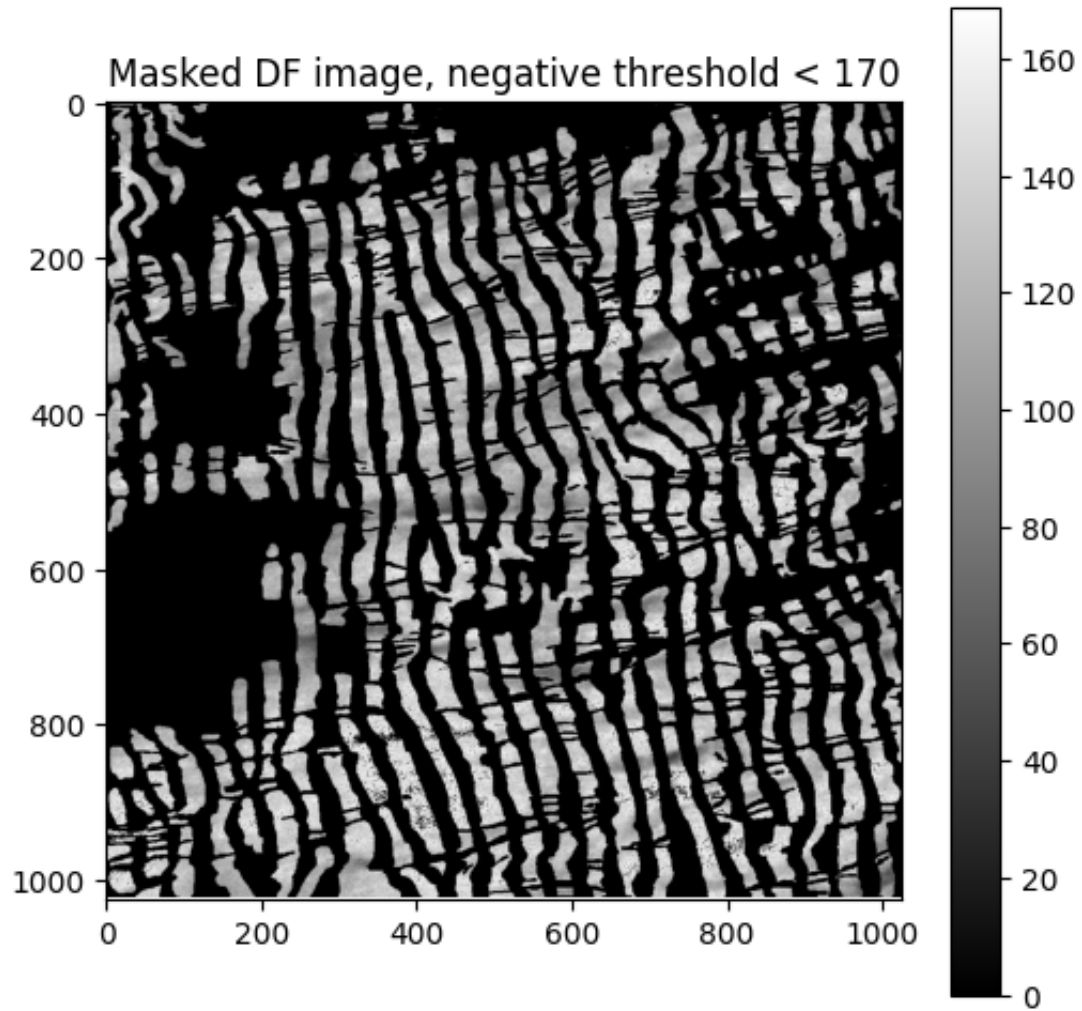
df_masked_thr_pos = np.where(df_masked >= threshold, df_masked, 0)

plt.figure(figsize=(6, 6))
plt.imshow(df_masked_thr_pos, cmap="gray")
plt.title(f"Masked DF image, positive threshold >= {threshold}")
plt.colorbar()
plt.show()
```

```
In [50]: #Negative thresholded DF image
df_masked_thr_neg = np.where((df_masked > 0) & (df_masked < thresho

plt.figure(figsize=(6, 6))
plt.imshow(df_masked_thr_neg, cmap="gray")
plt.title(f"Masked DF image, negative threshold < {threshold}")
plt.colorbar()
plt.show()
```



```
In [53]: #Boolean mask for positive-threshold region
thr_mask_pos = df_masked_thr_pos > 0

print("thr_mask_pos.shape:", thr_mask_pos.shape)
print("dc.Rshape:", dc.Rshape)
print("selected pixels (positive):", thr_mask_pos.sum())

assert thr_mask_pos.shape == dc.Rshape
```

```
thr_mask_pos.shape: (1024, 1024)
dc.Rshape: (1024, 1024)
selected pixels (positive): 4691
```

```
In [54]: #Fraction selected for positive-threshold region
n_selected_pos = thr_mask_pos.sum()
n_total_roi = thr_mask_pos.size
n_total_masked = mask_roi.sum()

print("fraction selected in whole ROI:", n_selected_pos / n_total_r
print("fraction selected within binary mask:", n_selected_pos / n_t
```

```
fraction selected in whole ROI: 0.004473686218261719
fraction selected within binary mask: 0.012508265961304636
```

```
In [55]: #Diffraction pattern from positive-threshold region

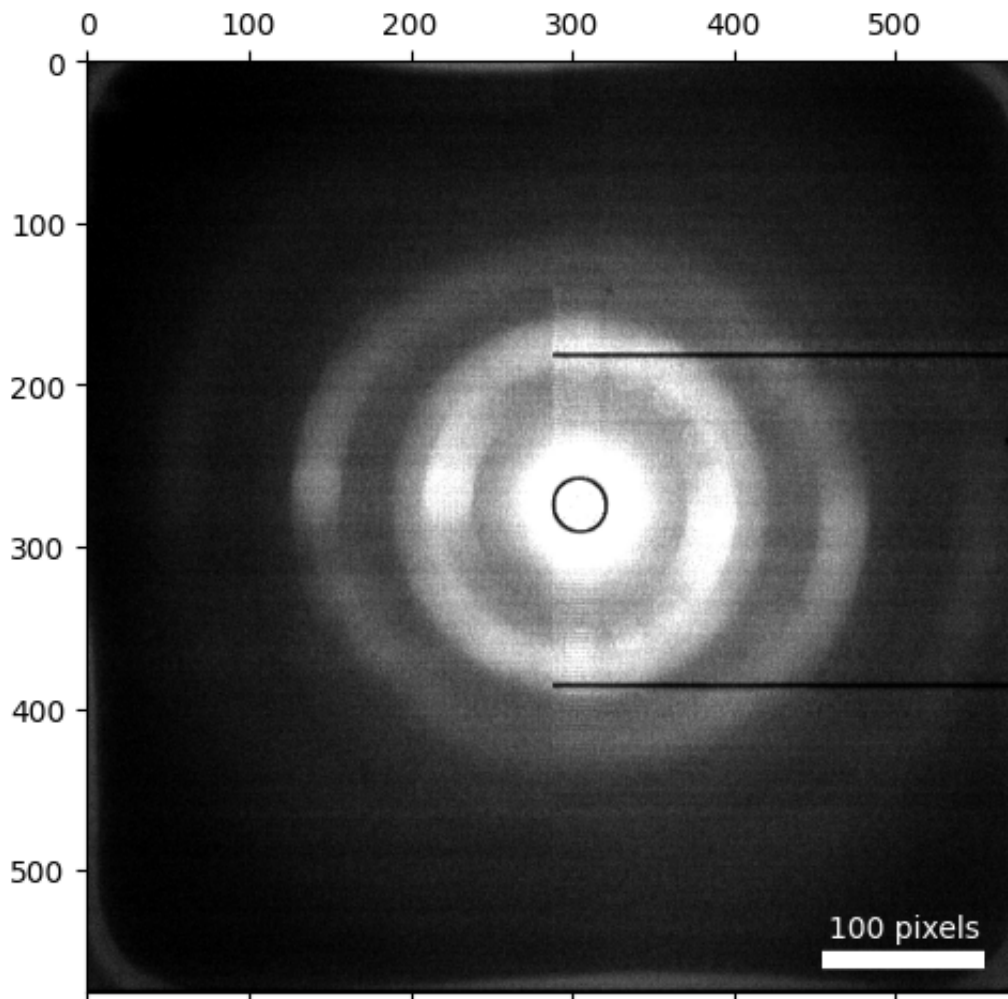
vd_thr_pos = dc.get_virtual_diffraction(
```

```

        method="mean",
        mask=thr_mask_pos,
        name="SS07_R09_Dp01_0FFA02_Df02_DfMBbf02_Th02_Dp02"
    )

py4DSTEM.visualize.show(vd_thr_pos, scaling="none")

```



```

In [58]: #Boolean mask for negative-threshold region
thr_mask_neg = df_masked_thr_neg > 0

print("thr_mask_neg.shape:", thr_mask_neg.shape)
print("dc.Rshape:", dc.Rshape)
print("selected pixels (negative):", thr_mask_neg.sum())

assert thr_mask_neg.shape == dc.Rshape

```

```

thr_mask_neg.shape: (1024, 1024)
dc.Rshape: (1024, 1024)
selected pixels (negative): 370333

```

```

In [59]: #Fraction selected for negative-threshold region
n_selected_neg = thr_mask_neg.sum()
n_total_roi = thr_mask_neg.size
n_total_masked = mask_roi.sum()

print("fraction selected in whole ROI:", n_selected_neg / n_total_r
print("fraction selected within binary mask:", n_selected_neg / n_t

```

fraction selected in whole ROI: 0.3531770706176758

fraction selected within binary mask: 0.9874704025256511

```
In [ ]: #Diffraction pattern from negative-threshold region
vd_thr_neg = dc.get_virtual_diffraction(
    method="mean",
    mask=thr_mask_neg,
    name="SS07_R09_Dp01_0FFA02_Df02_DfMBbf02_Th02_Min02_Dp02"
)

py4DSTEM.visualize.show(vd_thr_neg, scaling="none")
```

DEPARTMENT OF PHYSICS AND ASTRONOMY
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se



CHALMERS
UNIVERSITY OF TECHNOLOGY