doi:10.1017/S1431927621005043
Microsc. Microanal. 27 (Suppl 1), 2021

Using py4DSTEM in GMS: Hybrid Open-Source, Commercial-Freeware Methods for Analyzing 4D STEM Datasets

Benjamin Miller[1], Anahita Pakzad[2], Benjamin Savitzky[3], Colin Ophus[3] and Cory Czarnik[2]

[1]Gatan Inc., California, United States, [2]Gatan Inc., United States, [3]Lawrence Berkeley National Laboratory, California, United States

4D STEM is becoming increasingly prominent as a technique for characterization of materials. From 2005 to 2020 the number of papers found in a search in Scopus for 4D STEM and related terms went from 23 to 932, an increase of over 4000 percent. During the same time, the number of papers found with a search for transmission electron microscopy increased only 205 percent.

Image processing is always important for visualization and analysis of microscopy data, but it is especially important for 4D STEM based techniques, where the dimensionality of the data makes it difficult to directly display or interpret. A wide range of methods for processing, visualizing, and analyzing data exists, and the appropriate algorithm often depends on the way the microscope was configured as well as what kind of information is to be extracted. In addition to the variety of algorithms and methods, there are several broad categories of tools for processing, visualization, and analysis of 4D STEM data. These range from commercial software, to open-source software, and "home-made" scripts/codes.

One of the most commonly used commercial software packages for electron microscopy, and for 4D STEM specifically, is Gatan Microscopy Suite (GMS), sometimes also called Digital Micrograph (DM). The main value of commercial software is that it is designed to be simple to use and robust and consistent across multiple sites. Despite these benefits, it may not be free, and is not easily extensible. Figure 1 depicts just some of the possibilities for working with 4D STEM data in GMS.

Open-Source software is growing in popularity as a tool for processing 4D STEM data. Much of this is written in Python. Examples in this category include py4DSTEM, HyperSpy, LiberTEM, pyXem, and Pycroscopy. Open-source software is always free, in many cases regularly updated, and fully extensible since the code can be directly edited by anyone. However, the quality of open-source software can vary considerably, depending on the people and resources behind it. Figure 2 depicts just some of the possibilities for working with 4D STEM data in py4DSTEM, an open-source software package originating at the Molecular Foundry in Berkeley Lab (Savitzky et al., 2020).

The final type of software, which is no less important, but is by definition less visible, is "home-made." Research groups around the world are constantly working on new ways to collect and analyze data, and often no software exists to perform the exact series of steps which are desired. Writing one's own code using MATLAB, Python, or any other language familiar to the researcher can be the quickest way to get to a novel result out of unique data. Many examples of home-made scripts will undoubtedly be used to generate abstracts submitted to the M&M meeting this year.

Recently, Gatan has introduced the ability to create and run Python scripts directly within GMS. This enables users of GMS to combine the 3 types of software just described into a single workflow. For example, a researcher could collect 4D STEM data using a Gatan camera, align it with the built-in tools in GMS, then import it directly into py4DSTEM which is run from within GMS. This could process the data using one of its ready-made algorithms, exporting the result back to a new image in GMS. That image

could then be analyzed using a custom Python script that directly accesses the image displayed in GMS, outputting a Matplotlib chart. While there is no single approach that will satisfy all requirements, the combination of these approaches allows users to take advantage of each while avoiding their individual limitations.

In this presentation we will show how researchers can use Python in GMS, including how to install and access open-source packages, how to write custom scripts, and how to integrate this into data collection and visualization workflows like the one just described for 4D STEM data. While doing this we will also feature some of the capabilities of the Py4DSTEM package, including classification.

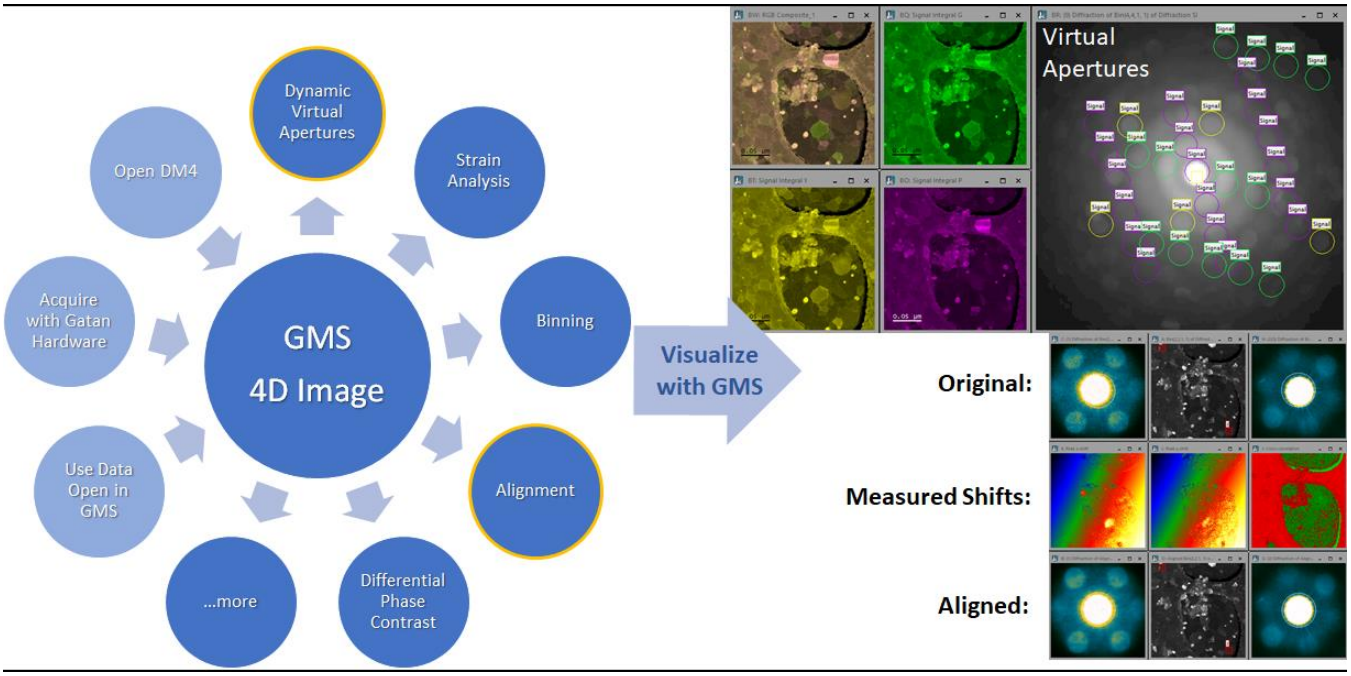

Figure 1. Figure 1 GMS Processing and Visualization. Left: Various capabilities of GMS for working with 4D STEM data. Top-Right: 3 Virtual dark field (DF) images generated from the 4D STEM dataset, plus a composite image combining the 3 DF images. Bottom-Right: Original data showing a small shift of the diffraction pattern center from the top-left of the scan to the bottom-right. Measured shifts are displayed in GMS along with the cross-correlation maximum to enable detailed inspection or analysis of the measurement. Aligned data shows that the pattern center is now stationary across the entire scan area. The dataset was acquired from an Al thin film on lacy carbon, with Ag nanoparticles dispersed on top.

https://doi.org/10.1017/S1431927621005043 Published online by Cambridge University Press

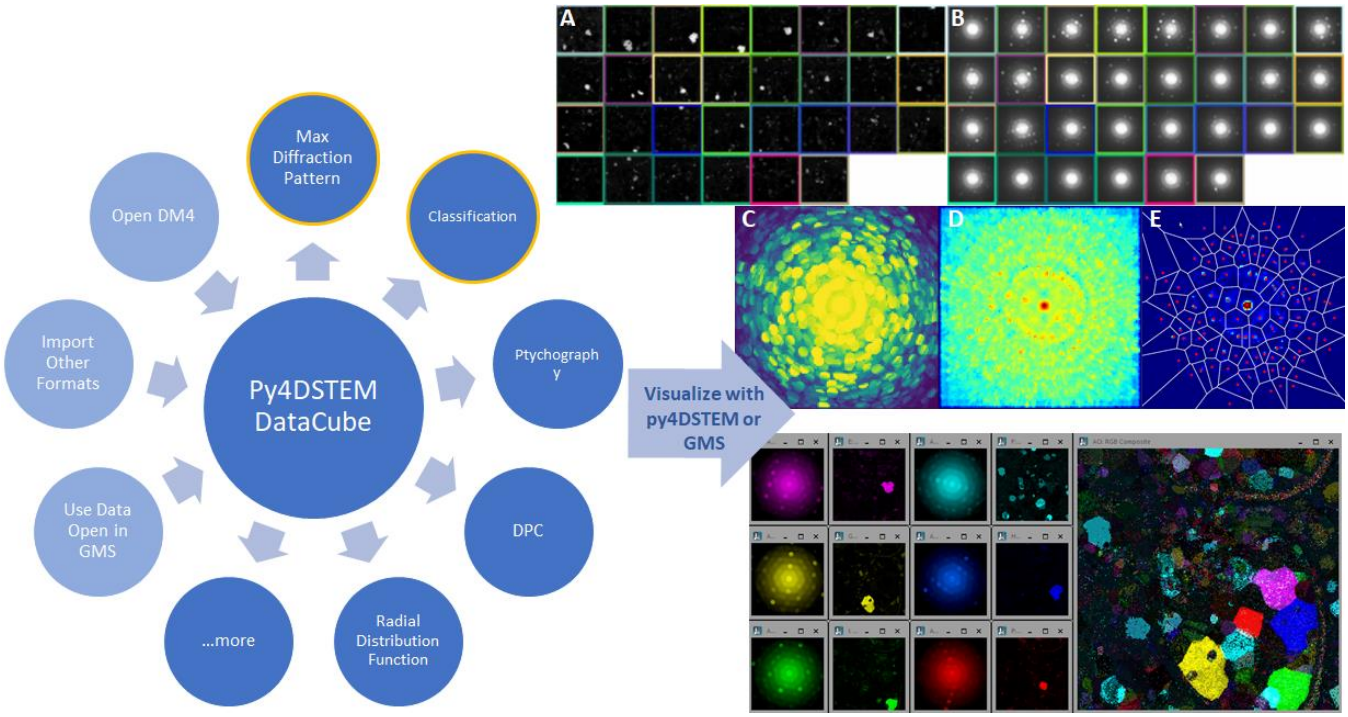

Figure 2. Figure 2 py4DSTEM Processing and Visualization. Left: Various capabilities of py4DSTEM for working with 4D STEM data. Top-Right: Data visualized using py4DSTEM running and operating on data within GMS. A) Class Images. B) Diffraction pattern classes. C) Maximum diffraction pattern. D) Bragg spot vector map. E) Voronoi diagram of Bragg spot vectors. Bottom-Right: A selection of 6 of the diffraction pattern classes and class images generated by py4DSTEM from the same dataset but visualized using the GMS user interface. The 4D STEM dataset is the same one used for Figure 1.

References

SAVITZKY, B. H., HUGHES, L. A., ZELTMANN, S. E., BROWN, H. G., ZHAO, S., PELZ, P. M., BARNARD, E. S., DONOHUE, J., DACOSTA, L. R., PEKIN, T. C., KENNEDY, E., JANISH, M. T., SCHNEIDER, M. M., HERRING, P., GOPAL, C., ANAPOLSKY, A., ERCIUS, P., SCOTT, M., CISTON, J., MINOR, A. M. & OPHUS, C. (2020). py4DSTEM: a software package for multimodal analysis of four-dimensional scanning transmission electron microscopy datasets. *arXiv:2003.09523 [cond-mat, physics:physics]*. http://arxiv.org/abs/2003.09523

https://doi.org/10.1017/S1431927621005043 Published online by Cambridge University Press