doi:10.1017/S1431927619001892
Microsc. Microanal. 25 (Suppl 2), 2019


# Towards Reproducible and Transparent Science of (Big) Electron Microscopy Data Using Version Control

Magnus Nord[1*] and Johan Verbeeck[1]

[1.] EMAT, University of Antwerp, Antwerp, Belgium.
* Corresponding author: Magnus.Nord@uantwerpen.be

The advent of faster, larger and more robust electron detectors has enabled the acquisition of rich multidimensional electron microscopy data. This includes techniques such as electron energy loss spectroscopy and energy-dispersive X-ray spectroscopy, where these improvements have enabled the acquisition of large spectrum maps. A more recent development is the use of fast pixelated detectors for use in scanning transmission electron microscopy (pixelated STEM)[1], which allows for the acquisition of a large fraction of the STEM diffraction pattern for each probe position. These new developments enable new and exciting ways of characterizing materials, but also pose challenges with regards to handling large amounts of data. For example, a medium sized pixelated STEM dataset can easily exceed 20 GB after compression.

A concurrent development has been the increased focus on open science, which aims to make research more transparent and reproducible. An important aspect of this is open data and open source, which means both the raw data itself and the processing should be trackable.

Both these trends necessitate an increased focus on not only data management, but also management of the data processing scripts needed to analyze the data. This becomes tricky with the increased collaborative nature of research, where often several people contribute to the processing of the data. Simply copy-pasting the same data to each researcher can lead to diverging data files and scripts, quickly leading to extensive work in merging the various contributions. Another solution could be to have a central server where are the files are stored and processed. However, the aforementioned pixelated STEM datasets can also easily be several gigabytes making it practical to have local versions for data exploration. For these data types the initial processing steps can often be fairly time consuming (for example radial integration), necessitating the use of intermediate data processing files. Thus, one needs some way of both tracking and synchronizing changes made to both the raw data, intermediate processing files and the scripts. Keeping the original raw data the same, while allowing for updates of the other parts of the processing.

This presentation will focus on the tools and principles of version control, and how the version control software (git)[2,3] can be used to track changes to not only the processing scripts themselves, but also the large intermediate pixelated STEM binary files (git-lfs)[4]. This allows for much better transparent and manageable data processing, as all the changes are easily trackable (Fig. 1). In addition, this enables several people to work on the data at the same time, while simultaneously minimizing the amount of conflicts. Finally, the presentation will include workflows enabled by using this type of version control system, and how it can be tied into continuous integration systems.

References:

[1] D McGrouther et al., Microscopy and Microanalysis (2015), p. 1595.
[2] J Blischak et al., PLOS Computational Biology (2016), p. 1.
[3] git, https://git-scm.com/ (accessed February 21, 2019)

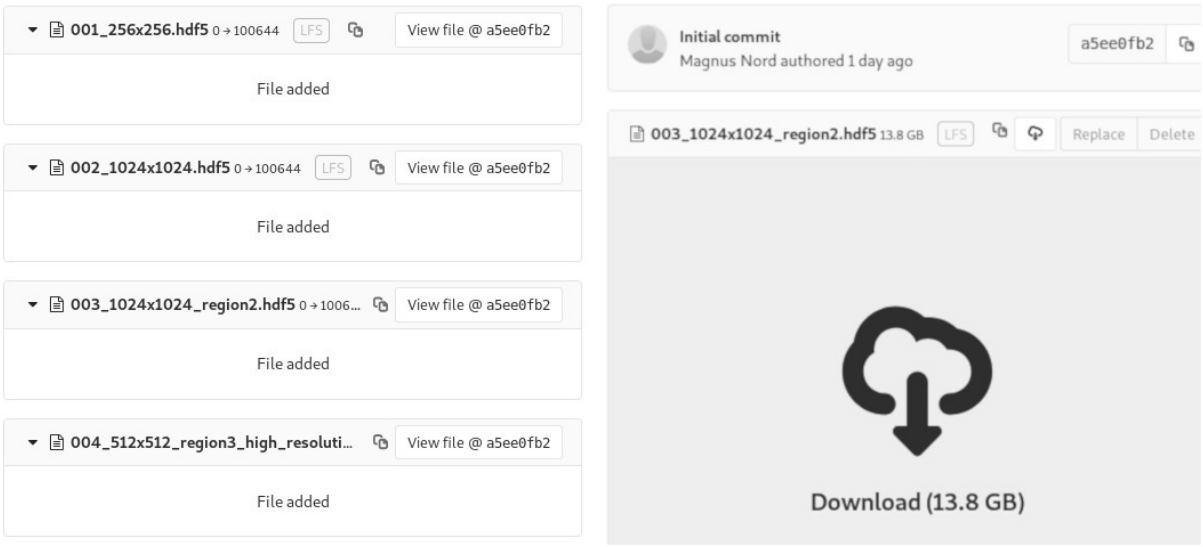


**Figure 1.** Example of version control of large binary files using git-lfs.

https://doi.org/10.1017/S1431927619001892 Published online by Cambridge University Press