doi:10.1017/S143192762100502X
Microsc. Microanal. 27 (Suppl 1), 2021

Compression and Access to Arbitrary Data: The Low-hanging Fruit

Mia Kraft

Department of Earth and Environmental Sciences, University of Minnesota, Minneapolis, Minnesota, United States

Data storage and access is often the most challenging and time-consuming part of electron microscopy. The current extended Work-From-Home situation has created a bottleneck in transfer rates and personal storage, only exaggerating the problems. These problems begin with needing to transfer the data to other devices and extends into what applications, proprietary formats, and vendor software are available for home use, and what number and type (on-site or off-site) of licenses, RAM, and Operating Systems are needed to run all of it.

Many proprietary formats are often straight forward as they are created while data is collected to minimize RAM usage, need to be read quickly and easily, and were created before many improvements in data storage were widely available. Because of this, many proprietary formats in use are simple arrays or hypercubes of data in a structured format with some scattered metadata at the beginning. While this is not always the case, it is a safe starting position for formats that have been in use for a long time. Another option, especially with electron images is to look for proprietary file extensions and change them to various, known image extensions in case of format misdirection.

Exporting all the raw data from these proprietary formats is not always easy and may be time consuming to re-compile into one dataset for further use, and yet more difficult to put into a programing friendly format for automated processing.

I have been working on creating a standardized, open source format for mass storage of simple arrays. While my current implementation is written in JavaScript, it is easily transferable to many modern languages such as Python, Java, Go, and Rust. By creating a structured file system for storage of specific structured data types, it is possible to easily expand a simple array storage with some metadata into storage of x-ray maps, compress the data using hash-based comparison of individual files, and further compress the file system using 7zip (https://7-zip.org/) or another compression algorithm. Basic information such as the type of compression is denoted using the file extension while more advanced structure data is included under specific generic JSON files included inside the zipped file structure.

When collecting analysis points, lines, or maps from a single sample, it is common for the metadata for a specific sub-section of that data to be the same as the larger metadata (A single point in a map should have the same conditions as any other point in that map). However, that metadata can differ and many formats where individual point data is stored in standalone files, the metadata often is copied once for every point. These waste massive amounts of storage as where keeping a single metadata file that all individual analyses when possible is a simple storage saver.

While less common, some analysis programs will export all data from a map as separate files in standalone directories to create a method of data storage reliant on the file system. This has an impactful side effect as it creates a reliance on the speed of the operating system and hardware, causing massive performance degradation on standard long-term storage HDD, or even on a modern SSD. A better solution to this is a flat-file system where files can be accessed more easily and require less operating system interaction. However, this is unrealistic if use is wanted on Windows, Mac, and many Unix operating systems.

Alternatively, all the data can be stored in a single file and paged into to maximize memory, cache, and disk speed. By using 7zip, I can compress a file system into a single file and easily navigate utilizing highly optimized and mature code. 7zip also allows more advanced operations such as higher or lower levels of compression allowing for higher data-capacity with slower performance or significantly faster data access at the cost of data-capacity.

By including various metadata files within the structure of the data storage file, it allows for versioning, customization, and construction of more complex objects. The use of JSON as the metadata format is chosen by the large adoption of this format. Versioning the storage format is important for forward and backwards compatibility as well as for defining when the data should or can be migrated from older to newer formats. Versioning the storage format also allows for custom methods of data storage using the same format, allowing for special types of data to only be accessible from the intended programs. Using generic metadata files can also allow for the creation of complex objects such as an X-ray map consisting of an arbitrary number of analysis points, each with a custom x-y position within the map. This also allows for more advanced methods of data relationships to exist such as a position containing multiple types of data, versioned data, or special conditions.

Often, by de-duplicating metadata it is possible to reduce data size. Using 7zip the compression ratio only improves further with minimal impact on data accessibility performance. Often, formats become slower to access as more data is added to them; However, 7zip does not suffer this problem as accessing the file index is a constant-time operation. Because of this, the largest factor for data access speed is level of compression.

The current version of this storage method and format is available on Github under the name “sxes-compressor” at https://github.com/Bob620/sxes-compressor. While this format was initially created to store data acquired on the JEOL Soft-Xray Emission Spectrometer (SXES) data, it has been expanded to handle not only SXES or wavelength-dispersive spectrometer EPMA data but can include images and arbitrary data. This format is primarily made to provide easy interaction with programs. While this is of direct use in many cases, a further application to view this data would need to be made and is currently under development.

References

Support for this research came from the National Science Foundation: EAR-1849465 (AVDH).

https://doi.org/10.1017/S143192762100502X Published online by Cambridge University Press