APAV: An Open-Source Python Package for Mass Spectrum Analysis in Atom Probe Tomography

Jesse D. Smith [iD] [1][¶] **and Marcus L. Young** [iD] [1]

1 Department of Materials Science and Engineering, University of North Texas, USA ¶ Corresponding author

Summary

As microstructures are increasingly engineered with nanoscale precision, comparably precise metrological tools like Atom Probe Tomography (APT) are crucial for experimental validation. APT has the unique capability of 3D imaging and compositional/isotopic analysis at the sub-nanometer scale (Gault et al., 2021). However, the informative potential of APT is hindered by nebulous, material-dependent evaporation physics, naturally complex data analysis/visualization, and the standards/specifications set by vendors. Transparent and accessible data analysis tools are critical for reproducible analysis of APT experiments. APAV seeks to bridge the gap of openly accessible analysis tools available to scientists leveraging APT in their research.

Statement of need

APAV is a Python package for analysis and visualization of APT experiments, with a focus on mass spectrum analysis and detector multiple events. APAV supplies an object-oriented interface into common positional and ranged file formats, forming the foundation from which APAV or user-adapted analysis can be extended. Commercial applications dominate the field and are limited to IVAS and its successor: AP Suite. Both IVAS and AP Suite are developed by the sole commercial vendor of Local Electrode Atom Probes (LEAP). Generalized lists of open-source contributions may be found online[1], in addition to a collection of software-oriented journal articles (Felfer et al., 2013; Kühbach et al., 2021; Monajem & Felfer, 2022). APAV aims to improve the general coverage of APT tools in the Python space while specifically targeting the analysis of mass spectra.

Python is a popular language in both general and specialized computing domains, partly owing to its comprehensive centralized package management (PyPI and Conda), openness, and reputed ease of use. Such qualities have helped to ingrain Python into the greater academic community, forming our belief that the development of APT tools for Python should be pursued. Though native Python is slow, APAV uses widely accepted scientific packages for numerical acceleration (i.e., Numpy, fast-histogram, etc.), custom C extensions for performance-critical computations, and Qt for graphical interactive tools. It is sustained by a suite of unit tests, continuous integration, git version control, documentation, and is easily acquirable through the Python package index (PyPI).

Input/output

Historically, post-reconstruction APT data is stored in two files: one file tabulating ion positions and the instantaneous experimental state (henceforth referred to as "positional"

---

[1]Such as: https://gitlab.com/jessesds/apav/-/blob/JOSS/docs/APT%20software%20list.md

files or data), and another file mapping user-specified ion compositions to half-open intervals in the mass/charge domain (henceforth referred to as "ranging" files or data). For best interoperability, APAV does not introduce new file formats, instead supporting the major file formats: *.pos (position), *.epos (extended position), *.ato (atom), *.apt (atom probe tomography), *.rng (range), and *.rrng (range). A summary of the positional formats and their content is tabulated in Table 1 with further details in literature (Gault et al., 2012; Larson et al., 2013). APAV does not implement readers or writers for pre-reconstruction data formats (*.rhit, *.rraw, *.hits, *.str, etc.) as these files are proprietary binary formats.

Table 1: Positional file formats with ion positions and state variables. Field inclusion is indicated by × for included, empty for not included, and ∘ for conditionally included.

Format	XYZ	m/n	TOF	DME	DC/Pulse volts	Det pos.
POS	×	×				
ePOS	×	×	×	×	×	×
ATO	×	×	×		×	×
APT	×	×	∘	∘	∘	∘

Positional data is stored in an apav.Roi container and is the primary data structure used for constructing other analysis objects. Readers and writers are accessed through its class methods Roi.from_XXX and Roi.to_XXX or convenience loading functions apav.load_XXX in the top-level namespace. It is worth noting that the *.apt format is a new file introduced by Cameca in AP Suite and is the only dynamic positional format that can contain an arbitrary number of fields. Details of the *.apt format specification can be found in the B.5.4 appendix (version 6.1) of the AP Suite Software User Guide, other formats are well summarized by Larson et al. (2013) in appendix A. The ranging formats are stored in a apav.RangeCollection containers loaded by RangeCollection.from_XXX class methods (or apav.load_XXX convenience functions).

Detector multiple events (DME)

Detector events can be partially characterized by a "multiplicity" defining the number of ions that the position sensitive detector (PSD) was able to resolve during a detection cycle (the time spanning two consecutive pulses). Ideally, all detector events would be "single", indicating a single ion was detected during the cycle. However, with increasingly nonmetallic bonding, the applied field can penetrate multiple atomic layers (due to weak electric-field screening (Tamura et al., 2012)), resulting in both increased molecular evaporation, and the evaporation of multiple ions—a detector multiple event (DME). Analyzing the effect of DMEs is complex, too many ions reaching the PSD in quick succession (dead times are ~3ns) may result in data loss. Similarly, neutral ions produced by molecular dissociation are predominantly undetected.
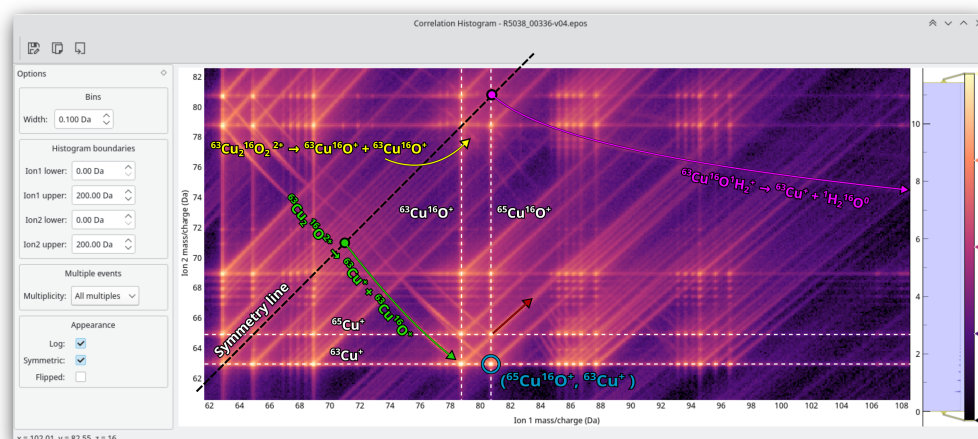
Figure 1: DME correlation histogram of a GdBa2Cu3O7-x superconductor with important physical phenomena annotated: co-evaporation (blue), thermally delayed evaporation (red), molecular dissociation (green), neutral generation (purple), and kinetic energy release (lobe, yellow).

For example, a DME of four ions "ABCD" (multiplicity of four) can be expressed as six unique pairs (AB, AC, AD, BC, BD, CD). Pairing DMEs in this fashion enables analysis through correlation histograms (Saxey, 2011), where the first ion is placed on the abscissa and the second on the ordinate. In APAV DME information can be accessed through `MultipleEventExtractor` that interfaces with a `Roi`—assuming the DME information exists (see Table 1). An example of the interactive correlation histogram tool is shown in Figure 1 for a complex oxide superconductor; this particular correlation histogram exhibits all currently known physical phenomena in a correlation histogram (see Smith et al. (2021) for details).

Mass spectrum quantification

Mass spectrum quantification principally seeks to compute the elemental (or molecular) composition of some region of interest in the APT dataset. Conventionally, this is achieved by accumulating the number of each experimentally observed ion by segmentation of the mass spectrum into explicit bins (which can be optionally background corrected). The composition is typically expressed in terms of the *decomposed* atomic concentration, but the ionic concentration is occasionally used. APAV supports three levels of quantification, differentiated by the level background correction: 1) No correction, 2) Random noise background correction, and 3) Random + local background correction. Conventionally, the background shape is sufficiently described by a power law,

$$I(x) = A(x - x_0)^{-B} \tag{1}$$

which is the default model used in APAV. Quantification in APAV was designed to be as flexible as possible, as the model selection and setup for one experiment may not apply to another. Further, any `Model` from `lmfit` (Newville et al., 2021) can be used (within reason) in addition to custom defined models. The automated background estimation in AP Suite and IVAS are prone to failure for complex spectra with many peaks, with no recourse available. APAV provides a flexible syntax for declaring multiple intervals for fit ranges and ion selections as needed.
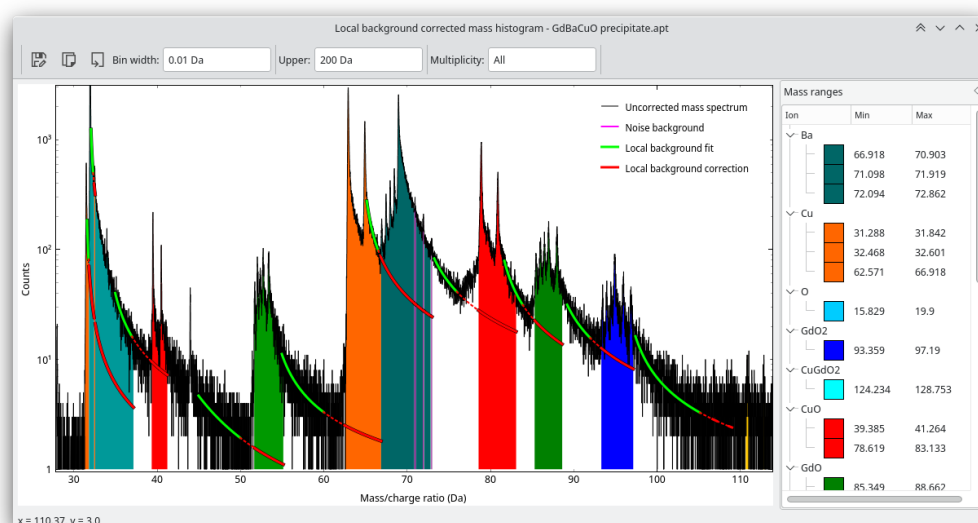
Figure 2: Ranged mass spectrum quantification with random noise and local background correction.

For nonmetals, certain isotopes may be sensitive to DMEs, such that they are not detected in isolation. See Figure 3 for an example of DME-dependent mass spectra, generated through APAV. In these situations, it may be prudent to quantify single events separately from multiple events (DMEs) (Yao et al., 2010). APAV can flexibly accommodate quantification based on DME multiplicity, with most constructors taking an optional `multiplicity` argument for controlling the multiplicity parameterization.
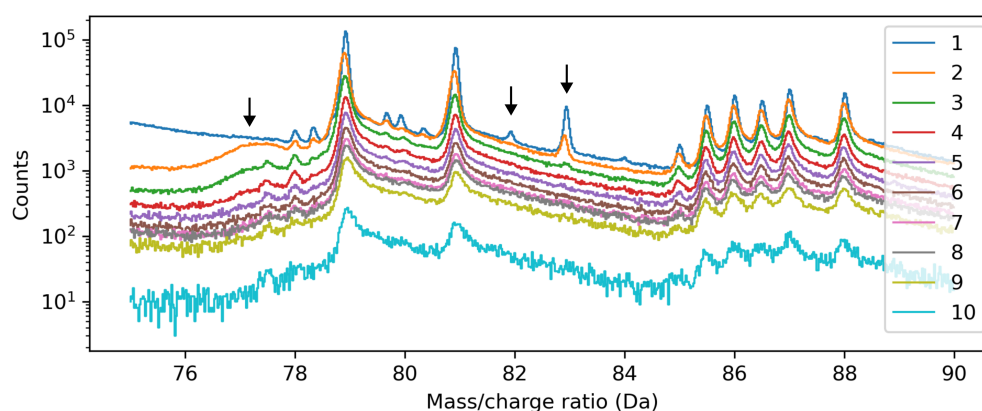

Figure 3: Mass spectra of the first 10 DME multiplicities of a nonmetallic specimen, showing DME-dependent phenomena.

Delocalization and the spatial-compositional domain

APT is often considered to exhibit atomic resolution, however, this truly depends on the specimen, acquisition parameters, and the exposed surface crystallography. Nevertheless, spatial analysis is structured in terms of the point cloud, or a structured compositional grid. Construction of the compositional grid is non-trivial as the initial dataset is discretized as a list of coordinates, and each coordinate does not necessarily represent a singular elemental ion. The compositional grid is initialized by the delocalization of each ion into discrete bins

using a chosen transfer function, which is then smoothed (typically using a Gaussian kernel). This smoothing is applied to mitigate aliasing artifacts possibly introduced by the grid-based sampling process of the ion distribution (and is not related the detection process itself). The amount of delocalization is defined by a specified length for each axis; i.e., d_x is the delocalization distance in the x-axis, which expresses the 3σ (three standard deviations) spatial distance that each ion is "spread" along this axis. The delocalization d_x constrains the standard deviations of both steps, such that they are added in quadrature: $d_x = \sqrt{(3\sigma_{x1})^2 + (3\sigma_{x2})^2}$, where σ_{x1} and σ_{x2} are the standard deviations of the 1st and 2nd pass stages, respectively. The delocalization distances (d_x, d_y, d_z) are user-provided, but default to $3 \times 3 \times 1.5$ nm, meaning the delocalization distance can be different along each axis. The reasoning for the smaller delocalization in the z-axis is due to the better spatial resolution along this axis (the z-coordinate is measured by the time of flight, whereas x/y coordinates are measured by the PSD position).

The first stage (placement into bins) cannot be accelerated by means of fast Fourier transforms (FFT) or other convolution methods. Instead, ion delocalization must be computed directly, per-ion. Computations like this are difficult to express solely with NumPy, this is an example where APAV utilizes multithreaded C extensions for acceleration. For this stage, any distribution could be used to transfer ions to the grid, APAV uses a Gaussian distribution. For example, the decomposed atomic concentration of sodium C_{Na}, at the voxel centered in space at v_i amounts to,

$$C_{\mathrm{Na}}(v_i) = \frac{1}{N_i} \sum_{j \in x} \frac{k_j}{\sigma_1 \sqrt{2\pi}} \exp\left(-\frac{||p_j - v_i||^2}{2\sigma_1^2}\right) \tag{2}$$

where N_i is the total number of (decomposed) ions contained within v_i, p_j is the spatial position of the j-th ion, x is the set of ions containing at least 1 Na atom, k_j is the number of Na atoms constituting the ion at p_j, and σ_1 is the standard deviation of the Gaussian distribution. The standard deviation is chosen such that the 3rd standard deviation is half the bin width, $3\sigma_1 = b/2$; this ensures that an ion in the center of a bin mostly contributes to only that bin (typical bin widths are 0.25-2 nm). For performance reasons, the kernel is truncated after the 1st nearest neighbors (i.e., the central bin and the surrounding 26 bins).

The second stage is a Gaussian smoothing filter on each grid. For small bin sizes there may be low counts in each bin, and smoothing helps to mitigate noise. The standard deviation for this filter, σ_2, is constrained by $d = \sqrt{(3\sigma_1)^2 + (3\sigma_2)^2}$, thus σ_2 can be directly determined by $\sigma_2 = \sqrt{d^2 + (3\sigma_1)^2}/3$. Computation of these grids provides a basis for studying compositional variations in the spatial domain, such as 2D concentration maps, proximity histograms, multivariate histograms, etc.

Acknowledgements

References

Felfer, P., Ceguerra, A., Ringer, S., & Cairney, J. (2013). Applying computational geometry techniques for advanced feature analysis in atom probe data. *Ultramicroscopy*, *132*, 100–106. https://doi.org/10.1016/j.ultramic.2013.03.004

Gault, B., Chiaramonti, A., Cojocaru-Mirédin, O., Stender, P., Dubosq, R., Freysoldt, C., Makineni, S. K., Li, T., Moody, M., & Cairney, J. M. (2021). Atom probe tomography. *Nature Reviews Method Primers*, *1*(51, 1). https://doi.org/10.1038/s43586-021-00047-w

Gault, B., Moody, M. P., Cairney, J. M., & Ringer, S. P. (2012). *Atom probe microscopy* (1st ed.). Springer. ISBN: 146143436X

Kühbach, M., Bajaj, P., Zhao, H., Çelik, M. H., Jägle, E. A., & Gault, B. (2021). On strong-scaling and open-source tools for analyzing atom probe tomography data. *Npj Computational Materials*, *7*(1), 1–10. https://doi.org/10.1038/s41524-020-00486-1

Larson, D., Prosa, T., Ulfig, R., Geiser, B., & Kelly, T. (2013). *Local Electrode Atom Probe Tomography: A User's Guide*. Springer. ISBN: 978-1-4614-8721-0

Monajem, M., & Felfer, P. (2022). APT_PyControl, an open-source python atom probe tomography control software package. *Microscopy and Microanalysis*, *28*(S1), 732–734. https://doi.org/10.1017/S1431927622003397

Newville, M., Otten, R., Nelson, A., Ingargiola, A., Stensitzki, T., Allan, D., Fox, A., Carter, F., Michał, Osborn, R., Pustakhod, D., Lneuhaus, Weigand, S., Glenn, Deil, C., Mark, Hansen, A. L., Pasquevich, G., Foks, L., … Persaud, A. (2021). *Lmfit/lmfit-py: 1.0.3* (Version 1.0.3). Zenodo. https://doi.org/10.5281/zenodo.5570790

Saxey, D. W. (2011). Correlated ion analysis and the interpretation of atom probe mass spectra. *Ultramicroscopy*, *111*(6), 473–479. https://doi.org/10.1016/j.ultramic.2010.11.021

Smith, J. D., Huh, J., Shelton, A., Reidy, R. F., & Young, M. L. (2021). Laser-Assisted Field Evaporation of RBa2Cu3O7-δ (R = Gd, Sm) High-Temperature Superconducting Coated Conductors. *Microscopy and Microanalysis*, 1–18. https://doi.org/10.1017/S1431927621012794

Tamura, H., Tsukada, M., McKenna, K. P., Shluger, A. L., Ohkubo, T., & Hono, K. (2012). Laser-assisted field evaporation from insulators triggered by photoinduced hole accumulation. *Phys. Rev. B*, *86*, 195430. https://doi.org/10.1103/PhysRevB.86.195430

Yao, L., Gault, B., Cairney, J. M., & Ringer, S. P. (2010). On the multiplicity of field evaporation events in atom probe: A new dimension to the analysis of mass spectra. *Philosophical Magazine Letters*, *90*(2), 121–129. https://doi.org/10.1080/09500830903472997