

Building Open Source Geochemical Research Tools in Python

M. J. WILLIAMS¹, L. SCHONEVELD¹, S. BARNES¹ & J. KLUMP¹

¹CSIRO Mineral Resources

While the geochemistry community is transitioning to more open, reproducible and collaborative science practices, effective tools to facilitate this transition remain few in number. Here we describe a number of geochemistry-focused open source Python packages at various stages of development, and highlight some of the software development practises and tools we have used to build them.

Of these, the most advanced is *pyrolite*, a Python package built to facilitate data-driven understanding of geological processes using multivariate geochemical data. The core features of this package include functions for transforming geochemical data and visualisation (including ternary, spider and density diagrams). Other under-development tools include utilities for transformation of image and stage coordinates to allow reproducible positioning between instruments, a simple tool for predicting molecular ion interferences from and a *pyrolite* extension linking compositions to alphaMELTS models. Each of these software projects are co-developed with their documentation and integrated test suites. Online documentation is updated and versioned with the software, and includes a set of examples for key features.

These tools are developed for and by the geochemistry community, and are both freely available and released under an unrestrictive license. Questions and all forms of user contributions are welcomed (e.g. code, documentation, bug reports and feature requests). We hope these tools will contribute to enabling better geochemical research practices, and potentially even establishing common workflows for geochemical data processing and analysis within the user community.

A programmatic approach has distinct advantages over spreadsheet-based tools, and can result in improved transparency and traceability of results. The scientific Python ecosystem provides a solid foundation for prototyping, developing and interlinking tools from data processing through to modelling and machine learning. Tools can be extended with user interfaces and optimised for performance where needed. Where accessibility is kept in mind, developing tools in Python, and designing packages to be re-used and integrated into existing scientific workflows can improve efficiency and reproducibility while enabling geoscientists to develop transferable digital skills.