The Journal of Open Source Software

yadg: yet another datagram

Peter Kraus[1¶], **Nicolas Vetsch**[1], **and Corsin Battaglia**[1]

1 Empa - Swiss Federal Laboratories for Materials Science and Technology, Überlandstrasse 129, 8600 Dübendorf Switzerland ¶ Corresponding author

DOI: 10.21105/joss.04166

Software

- Review
- Repository
- Archive

Editor: Gabriela Alessio Robles

Reviewers:

- @aozorahime
- @1mikegrn
- @pythonpanda2

Submitted: 20 January 2022
Published: 05 April 2022

Summary

The management of scientific data is a crucial aspect of modern data science. Four simple guiding principles combined under the FAIR data moniker define the current "gold standard" in data quality: the data has to be **F**indable by anyone, **A**ccessible without barriers, **I**nteroperable with other programs, and **R**eusable after analysis (Wilkinson et al., 2016). Yet, many scientific data formats do not conform to these principles. This is especially true for proprietary formats, often associated with expensive lab instrumentation. The yadg package helps to resolve this issue by parsing raw data files into a standardised, annotated and timestamped format readable by both humans and machines. Various raw data formats are supported, including chromatograms, electrochemical cycling protocols, reflection coefficient traces, spectroscopic data, and tabulated data. The parsed files include information about data provenance, units of measure, and experimental uncertainties by default. Finally, several common data processing steps, such as applying calibration functions, integration of chromatographic traces, or fitting of reflection coefficients, are available in yadg.

Statement of need

From the point of view of catalytic chemistry, digitalisation is currently a "hot topic". For example, the leading German and Swiss academic bodies in the field of catalysis consider digitalisation an essential task for the current decade. The German Catalysis Society has published a detailed roadmap (Demtröder et al., 2019), and large consortia have been formed (eg. FAIRmat (Draxl, 2020) in Germany or NCCR Catalysis in Switzerland (*NCCR Catalysis*, 2021)) to advance this issue.

However, this change will not happen overnight. One approach for kick-starting this transition is to make the currently existing "small data" available according to the FAIR principles (Mendes et al., 2021), effectively defining a standard for "big data" by superposition of current best practices. Another approach, more specific to catalysis, is that of standardisation of testing and characterisation protocols (Trunschke et al., 2020), which will necessarily lead to a standardised data content. The yadg package presented here has been conceived to combine both approaches, and when combined with a lab automation software and data post-processing tools, forms a robust data-scientific pipeline (Kraus et al., 2021).

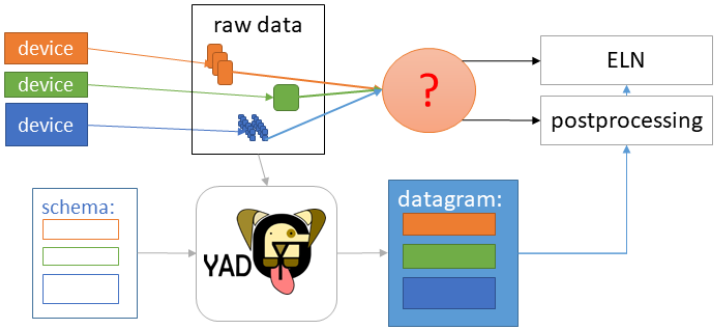

Figure 1: Example workflow for yadg.

An example laboratory workflow is shown in Figure 1. The devices in the laboratory (eg. chromatographs, flow meters, potentiostats, temperature controllers, or spectrometers) create raw data files that are different in shape, size, and number. If one wishes to store this raw data in an electronic lab notebook (ELN) software in a FAIR way or use it in post-processing, the raw data must be parsed. The intermediate parsing step is often not standardised: in the case of chromatography, the data parsing may be specific to the particular data files as opposed to file types, and in the case of flow and temperature data the values are often entered manually and hence prone to human error.

However, it is generally possible to semantically define a set of raw data related to an experiment by placing them in a folder. Then, a hierarchical description (`schema`) of the data in that folder can be created, containing information about the file types, quantities, their locations, etc. The `yadg` package can be used for reproducible processing of such `schema` files into a standardised, timestamped, versioned, annotated, and validated `datagram`, which can be directly uploaded to the ELN or used in post-processing. The file parsers in `yadg` are written in a modular fashion, grouped by scientific application (e.g. chromatography, electrochemistry, etc.), and are fully transparent to the user regardless of the raw file format or device vendor. Additionally, the same `schema` can be used to process different experimental folders using the `preset` functionality of `yadg`.

Features

Here, we present the revised `yadg-4.0`. Compared to the previous version of `yadg` (Kraus et al., 2021), the three main user-facing changes in `yadg-4.0` include units and uncertainties in the raw data, the enhanced chromatogram parser functionality, and the new electrochemistry parser. For a more detailed list of changes, please see the release notes.

Usage

An interactive, Binder-ready Jupyter notebook, illustrating the installation and usage of `yadg-4.0`, is included on Zenodo under DOI: 10.5281/zenodo.6351210. The Binder-ready notebook can be launched by following this link.

Units and uncertainties

In contrast with the previous version of the tool, raw data is now retained in the `datagram` and is annotated by units and measurement uncertainties, making `datagrams` suitable for archiving in an ELN. While units are often present in the raw data files, the uncertainties often have to be determined from instrument specifications. This is a particularly important

feature, as most data scientific packages in the Python ecosystem either support annotating array or tabular data by units (`astropy` (The Astropy Collaboration et al., 2018; The Astropy Collaboration, 2021) or `pint` (Grecco & Chéron, 2021)), or uncertainties (`unumpy` from the `uncertainties` package (Lebigot, 2021)); the combination of both units and uncertainties is quite rare, especially with non-tabular data.

Chromatography

The new version of yadg adds support for several new chromatographic data formats. Several open-source packages for parsing and processing chromatographic data exist, but they are often unmaintained (eg. `PyMS` (O'Callaghan et al., 2015; O'Callaghan et al., 2012), or `Aston` (Bovee et al., 2020)) or they are specialised for a single branch of chromatography, such as `PyMassSpec` (Davis-Foster, 2020). For yadg, the main goal is to parse the often proprietary and binary formats and extract the trace data. The parser now supports several new formats, including Agilent's `.ch` and `.dx` file types, the parsing of which is based on the Matlab routines available in the `chromatography` repository (Dillon, 2016).

The peak integration in yadg has been completely re-written and is now extensively using `numpy.ndarrays` (Harris et al., 2020). Combined with a drop-in calibration file functionality, yadg is a powerful chromatogram processing tool, usable regardless of the type of chromatography or detector used.

Electrochemistry

Another major feature in `yadg-4.0` is the support for electrochemical data. Files containing data from basic electrochemical techniques (chronoamperometry, chronopotentiometry, linear sweep voltammetry), two- or three-electrode experiments, battery cycling protocols, or impedance methods (potentio- and galvano-electrochemical impedance spectroscopy) are supported, among many other techniques. As with other file types, the electrochemistry data is fully annotated with units and uncertainties based on instrument specifications.

The proprietary file formats supported in `yadg-4.0` have been reverse-engineered. This work was inspired by the `galvani` package (Kerr, 2022). The parser in `yadg-4.0` also processes the metadata and settings stored in the binary `.mpr` files, supports parsing of text `.mpt` files, and supports a much wider range of instruments and electrochemical techniques, making yadg the most complete open-source parser for such file types.

Forwards compatibility

File types supported in the previous version of yadg are still supported in the current version. Starting with `yadg-4.0`, an update path for `schema` files is available, meaning old `schema` files containing or referencing calibration data in superseded formats can be seamlessly migrated to the latest version of yadg.

Acknowledgements

The authors acknowledge funding from the Synfuels project of the ETH Board. The authors would like to thank A. Senocrate and F. Bernasconi for discussions and raw data files. P. K. would like to thank E. H. Wolf for helpful discussions during previous iterations of this project.

References

Bovee, R., Yunker, L., & Davis-Foster, D. (2020). Aston: View and interpret UV-visible and mass spectrometry chromatographic data. In *Github repository - v0.7.1*. https://github.com/bovee/Aston.

Davis-Foster, D. (2020). PyMassSpec: Python toolkit for mass spectrometry. In *Github repository - v2.3.0*. https://github.com/PyMassSpec/PyMassSpec.

Demtröder, D., Deutschmann, O., Eck, B., Franke, R., Gläser, R., Goosen, L., Grunwaldt, J. D., Krähnert, R., Kragl, U., Leitner, W., Mestl, G., Reuter, K., Rosowski, F., Schäfer, A., Scheffler, M., Schlögl, R., Schüth, F., Schunk, S. A., Studt, F., … Wolf, D. (2019). *The digitalization of catalysis-related sciences* (p. 28) [Whitepaper]. German Catalysis Society (GeCatS).

Dillon, J. (2016). Chromatography: Functions for chromatography and mass spectrometry data analysis. In *Github repository - v0.1.51*. https://github.com/chemplexity/chromatography.

Draxl, C. (2020). *FAIRmat* [Conference paper]. 2. NFDI Conference, online; Deutsche Forschungsgemeinschaft (DFG).

Grecco, H. E., & Chéron, J. (2021). Pint: Makes units easy. In *Github repository - v0.18*. https://github.com/hgrecco/pint.

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., … Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2

Kerr, C. (2022). Galvani: Read proprietary file formats from electrochemical test stations. In *Github repository - v0.2.1*. https://github.com/echemdata/galvani.

Kraus, P., Wolf, E., Prinz, C., Bellini, G., Trunschke, A., & Schlögl, R. (2021). *Towards automation of operando experiments: A case study in contactless conductivity measurements* [Preprint]. https://doi.org/10.33774/chemrxiv-2021-mh17g

Lebigot, E. O. (2021). Uncertainties: A Python package for calculations with uncertainties. In *Github repository - v3.1.6*. https://github.com/lebigot/uncertainties.

Mendes, P. S. F., Siradze, S., Pirro, L., & Thybaut, J. W. (2021). Open data in catalysis: From today's big picture to the future of small data. *ChemCatChem*, *13*(3), 836–850. https://doi.org/10.1002/cctc.202001132

NCCR Catalysis: Our approach. (2021). https://www.nccr-catalysis.ch/research/approach/.

O'Callaghan, S., De Souza, D. P., Isaac, A., Wang, Q., Hodkinson, L., Olshansky, M., Erwin, T., Appelbe, B., Tull, D. L., Roessner, U., Bacic, A., McConville, M. J., & Likić, V. A. (2012). PyMS: A Python toolkit for processing of gas chromatography-mass spectrometry (GC-MS) data. Application and comparative study of selected tools. *BMC Bioinformatics*, *13*(1), 115. https://doi.org/10.1186/1471-2105-13-115

O'Callaghan, S., Isaac, A., & Likić, V. A. (2015). Pyms: Python toolkit for processing of chromatography-mass spectrometry data. In *Github repository - v0.2*. https://github.com/ma-bio21/pyms.

The Astropy Collaboration. (2021). Astropy: Astronomy and astrophysics core library. In *Github repository - v4.0.6*. https://github.com/astropy/astropy.

The Astropy Collaboration, Price-Whelan, A. M., Sipőcz, B. M., Günther, H. M., Lim, P. L., Crawford, S. M., Conseil, S., Shupe, D. L., Craig, M. W., Dencheva, N., Ginsburg, A., VanderPlas, J. T., Bradley, L. D., Pérez-Suárez, D., de Val-Borro, M., Aldcroft, T. L., Cruz, K. L., Robitaille, T. P., Tollerud, E. J., … Zabalza, V. (2018). The Astropy Project: Building an inclusive, open-science project and status of the v2.0 core package. *The Astronomical Journal*, *156*(3), 123. https://doi.org/10.3847/1538-3881/aabc4f

Trunschke, A., Bellini, G., Boniface, M., Carey, S. J., Dong, J., Erdem, E., Foppa, L., Frandsen, W., Geske, M., Ghiringhelli, L. M., Girgsdies, F., Hanna, R., Hashagen, M.,

Hävecker, M., Huff, G., Knop-Gericke, A., Koch, G., Kraus, P., Kröhnert, J., … Wrabetz, S. (2020). Towards experimental handbooks in catalysis. *Topics in Catalysis*, *63*, 1683–1699. https://doi.org/10.1007/s11244-020-01380-2

Wilkinson, M. D., Dumontier, M., Aalbersberg, Ij. J., Appleton, G., Axton, M., Baak, A., Blomberg, N., Boiten, J.-W., da Silva Santos, L. B., Bourne, P. E., Bouwman, J., Brookes, A. J., Clark, T., Crosas, M., Dillo, I., Dumon, O., Edmunds, S., Evelo, C. T., Finkers, R., … Mons, B. (2016). The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data*, *3*(1), 160018. https://doi.org/10.1038/sdata.2016.18