




AltamISA: a Python API for ISA-Tab files

Mathias Kuhring^{1,2,3}, Mikko Nieminen^{1,3}, Jennifer Kirwan^{2,3}, Dieter Beule^{1,3}, and Manuel Holtgrewe^{1,4}

1 Core Unit Bioinformatics, Berlin Institute of Health (BIH), Berlin, Germany **2** Berlin Institute of Health Metabolomics Platform, Berlin Institute of Health (BIH), Berlin, Germany **3** Max Delbrück Center (MDC) for Molecular Medicine, Berlin, Germany **4** Charité – Universitätsmedizin Berlin, Berlin, Germany

DOI: [10.21105/joss.01610](https://doi.org/10.21105/joss.01610)

Software

- [Review](#) 
- [Repository](#) 
- [Archive](#) 

Submitted: 25 July 2019

Published: 20 August 2019

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC-BY](#)).

Introduction

AltamISA is a Python library for reading, validating, representing and writing the ISA-Tab file format. ISA-Tab is an open, TSV (tab separated values)-based file format for representing the ISA (investigation study assay) tools data model (Sansone et al., 2012). The ISA tools data model allows for representing life science experiments and annotating the modeled objects and steps with arbitrary meta data. The ISA tools data model and TSV format are used by various life science databases, including MetaboLights (Haug et al., 2013).

Shortly, the experimental process from sample extraction from a source (e.g. a donor individual) through processing of the samples to creating read-outs in one or more assays can be represented through DAGs (directed acyclic graphs) consisting of extensively annotatable so-called *material* and *process* nodes. Together, the ISA tools data model and ISA-Tab allow for representing most conceivable experiments in life science and to store them into machine-readable files for exchanging information about such experiments. This greatly facilitates the development of data management applications following the FAIR (findable, accessible, interoperable, and reusable, cf. Wilkinson et al. (2016)) guidelines.

Motivation

The authors are developing an application for managing omics data. In our research regarding existing data schemas and file formats we found the ISA-Tab data model and file format to be highly suitable. Further, it had already seen adoption by databases such as MetaboLights (Haug et al., 2013).

While Sansone et al. (2012) also maintain a Python package `isa-api` for accessing ISA-Tab data our evaluation found several issues. To overcome these issues and to introduce several features important to us we decided to create an independent implementation of a Python library for ISA-Tab access: AltamISA.

Aims and Features

We developed AltamISA with the aim of providing the following features:

- A strictly validating parser that is easy to extend with both validation errors and warnings. The importance of proper validation was underlined when we tried to test our

parser with ISA-Tab sheets from MetaboLights (Haug et al., 2013). Here, a large proportion of ISA-Tab files failed validation (manual checking revealed actual problems in the data sets; data not shown).

- Standard Python exception and warning approaches to allow for user- and application-specific handling of validation issues.
- Well-tested code with good API documentation and proper examples showing the major use cases.
- Support for both reading and writing ISA-Tab files (full *round tripping*). This round trip operation was aimed to be *idempotent*, that is, after the first round trip any further round trip does not change the file content.
- The data structures are *immutable* (based on the `attrs` Python library). In particular in the connection with file access there are several advantages to this approach as one cannot accidentally change the header of the file one is reading, preventing whole classes of errors.
- All public APIs are fully annotated with Python type hints allowing for good IDE support. We found this particularly helpful given the large number of built-in material and process types in the ISA data model.
- The experiment DAG is implemented using a simple, graph theory-based approach with nodes representing ISA sources/samples/materials or processes and arcs explicitly connecting each. In our opinion, using explicit arc objects allows for a more straightforward implementation compared to storing input and output node references as done in `isa-api`. Further, graphs can with ease be subjected to canonical graph algorithms such as breadth-first search or union-find.

Further, we implemented a small number of example applications:

- `isatab2dot` allows converting of ISA-Tab files into the DOT file format for visualization with GraphViz (Gansner, Koutsofios, North, & Vo, 1993). We found this useful for both trouble-shooting AltamISA itself and sample sheets.
- `isatab2isatab` allows to perform the aforementioned round tripping and thus a *normalization* of ISA-Tab files.
- `isatab_validate` allows to read in an ISA-Tab file and run the AltamISA validator suite on the input.

For now, we have excluded the JSON (JavaScript Object Notation)-based file format as well as the emerging RFD (Resource Description Framework)-based file format (linkedISA (Gonzalez-Beltran, Maguire, Sansone, & Rocca-Serra, 2014)) from the scope of this project. These ISA file formats appear to be not widely adapted and require specialized editors, while ISA-Tab can be created and manipulated not only by using the ISAcreeator application (Sansone et al., 2012) but also standard spreadsheet software.

Summary and Conclusion

AltamISA is a practical and modern Python implementation of the ISA-Tab file model. Besides software industry best practices such as automated tests with high test coverage, it features a comprehensive API documentation, a strictly validating parser, and an immutable data model. It is actively maintained in connection to our data management application efforts and has been tested in practice with dozens of ISA-Tab files both prepared using the ISAcreeator and spreadsheet applications. It is our expectation that this library will be useful for other software developers who want to use the ISA model and ISA-Tab file format for file exchange.

License and Availability

AltamISA is distributed under the MIT license and available from GitHub at <https://github.com/bihealth/altamisa>. Each release is also stored to Zenodo. The current version 0.2.2 is available with the DOI 10.5281/zenodo.3369223. Examples and complete, up-to-date API documentation can be found at <https://altamisa.readthedocs.org>. We welcome contributions via GitHub as outlined in the documentation.

References

- Gansner, E. R., Koutsofios, E., North, S. C., & Vo, K.-P. (1993). A technique for drawing directed graphs. *IEEE Transactions on Software Engineering*, 19(3), 214–230. doi:[10.1109/32.221135](https://doi.org/10.1109/32.221135)
- Gonzalez-Beltran, A., Maguire, E., Sansone, S. A., & Rocca-Serra, P. (2014). linkedISA: Semantic representation of ISA-Tab experimental metadata. *BMC Bioinformatics*, 15(S14), S4. doi:[10.1186/1471-2105-15-S14-S4](https://doi.org/10.1186/1471-2105-15-S14-S4)
- Haug, K., Salek, R. M., Conesa, P., Hastings, J., Matos, P. de, Rijnbeek, M., Mahendraker, T., et al. (2013). MetaboLights—an open-access general-purpose repository for metabolomics studies and associated meta-data. *Nucleic Acids Research*, 41(D1), D781–D786. doi:[10.1093/nar/gks1004](https://doi.org/10.1093/nar/gks1004)
- Sansone, S. A., Rocca-Serra, P., Field, D., Maguire, E., Taylor, C., Hofmann, O., Fang, H., et al. (2012). Toward interoperable bioscience data. *Nature Genetics*, 44(2), 121–126. doi:[10.1038/ng.1054](https://doi.org/10.1038/ng.1054)
- Wilkinson, M. D., Dumontier, M., Aalbersberg, I. J., Appleton, G., Axton, M., Baak, A., Blomberg, N., et al. (2016). The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data*, 3, 160018. doi:[10.1038/sdata.2016.18](https://doi.org/10.1038/sdata.2016.18)