Towards FAIRer EPR data – a toolset for fully reproducible data analysis

Mirjam Schröder[1], Jara Popp[2], Till Biskup[2,3]

[1]Leibniz-Institut für Katalyse e.V., Rostock, Germany; [2]Institut für Physikalische Chemie, Albert-Ludwigs-Universität Freiburg, Freiburg, Germany; [3]Current address: Bundesinstitut für Risikobewertung, Berlin, Germany

The progress of science rests on the published work of others. In the digital age, we are confronted with an exponential growth of available data. Hence, appropriate research data management is urgently needed and data need to be FAIR: findable, accessible, interoperable, and reusable[1]. However, a necessary though often underrepresented prerequisite for FAIR data is a gapless record of their provenance. We therefore need tools that automatically record each step from the raw/primary data to their final shareable representation and take care of the respective metadata (*i.e.*, documentation) of each step starting with the data acquisition.

Here, we present both, a general framework for the reproducible analysis of spectroscopic data (ASpecD)[2] as well as two concrete packages based upon it and dedicated to working with continuous-wave[3] (cwepr) and time-resolved[4] (trepr) EPR data.

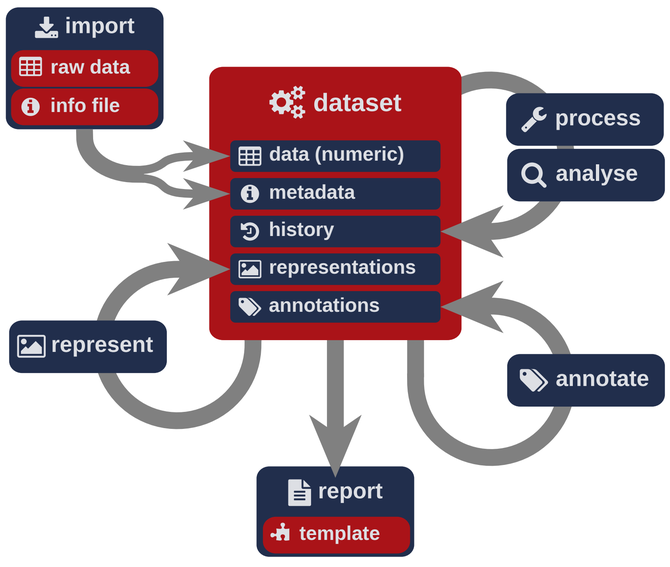


Figure 1: The dataset is one central component of the ASpecD framework and derived packages. It contains both, numerical data and all crucial metadata – a prerequisite for data provenance and hence FAIR(er) data.

Key features

- Fully reproducible data processing and analysis
- Gap-less record of each processing/analysis step, including explicit and implicit parameters.
- Import of EPR data from diverse sources
- Generic representation of EPR data, independent of the original format: dataset (Fig. 1).
- Datasets contain both, numerical data and all crucial metadata – a prerequisite for FAIR data.
- Generic plotting capabilities, easily extendable
- Report generation using pre-defined templates
- Recipe-driven data analysis, allowing tasks to be performed fully unattended in the background
- Open source projects written in Python (>= 3.7)
- Extensive user and API documentation

ASpecD

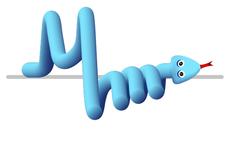
Framework for the *A*nalysis of *Spec*troscopic *D*ata, providing extensive core functionality.

https://docs.aspecd.de/ – Reference: [2]

cwepr

Package for analysing cw-EPR data, supporting many different file formats.

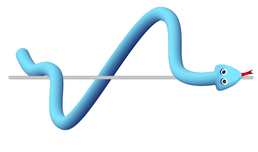
https://docs.cwepr.de/ – Reference: [3]



trepr

Package for analysing tr-EPR data, *e.g.*, transient nutation analysis.

https://docs.trepr.de/ – Reference: [4]



Recipe-driven data analysis by example – intuitive, fully reproducible, no programming skills needed

Key to recipe-driven data analysis is to describe in a structured and intuitive way what shall happen to which datasets. An example recipe is shown in Fig. 2, and it should be fairly obvious even to non-EPR experts what is going to happen.

A recipe is a structured text file (in YAML format) that is both, human-writable and machine-readable. It consists of several blocks. The most important for data analysis are the list of datasets (2) and the list of tasks (3). Each task can operate either on all datasets loaded or only on an explicitly given subset.

1 General settings

Most of the general settings at the beginning of a recipe are optional, but they allow for convenience settings such as the default package to use and the source and output directories.

2 Datasets

Datasets are a key component of the ASpecD framework and derived packages. Each task operates on datasets. Datasets are usually referred to by their file name, but can be given labels for internal reference within a recipe. The file format gets automatically detected and the correct importer chosen.

3 Tasks

Everything that happens to a dataset or a list of datasets is a task, be it processing, analysis, graphical or tabular representation, or even textual reports. Details for some of the different kinds of tasks are given below.

```
1  format:
2    type: ASpecD recipe
3    version: '0.2'
4
5  settings:
6    default_package: cwepr
7    autosave_plots: false
8
9  directories:
10   datasets_source: ../../tests/io/testdata/
11
12 datasets:
13 - BDPA-1DFieldSweep
14
15 tasks:
16 - kind: processing
17   type: BaselineCorrection
18   comment: Zero-order to remove DC offsets
19 - kind: processing
20   type: FrequencyCorrection
21   properties:
22     parameters:
23       frequency: 9.5
24   comment: Always important to be comparable between measurements
25 - kind: singleplot
26   type: SinglePlotter1D
27   properties:
28     parameters:
29       show_legend: true
30       show_zero_lines: true
31     filename: BDPA-FieldSweep.pdf
32     caption:
33       title: A first look at the recorded data.
34       text: >
35         The spectrum has been baseline and frequency-corrected,
36         but is otherwise displayed as recorded.
37   apply_to:
38   - BDPA-1DFieldSweep
39 - kind: report
40   type: LaTeXReporter
41   properties:
42     template: dataset.tex
43     filename: BDPA-FieldSweep-Report.tex
44   apply_to:
45   - BDPA-1DFieldSweep
46   compile: true
```

Figure 2: Example recipe showing essential features of recipe-driven data analysis. Here, the first steps of analysing any cw-EPR dataset is shown, resulting in both, a graphical representation and a well-formatted printed report.

4 Processing and analysis tasks

A processing task always results in an altered dataset, while an analysis task may output anything from a scalar number to a new dataset. The ASpecD framework comes with a large and growing list of both, processing and analysis tasks frequently used when dealing with spectroscopic data. The cwepr and trepr packages provide additional special functionality.

5 Graphical representation: plots

A picture is worth a thousand words, and usually the first step after recording data is to look at a graphical representation. Furthermore, for data to be FAIR we need to be able to track a figure in a publication back to the original raw data and get a detailed account of all the processing and analysis steps including their parameters that led to the final figure.

6 Overview: well-formatted reports

Recipes and particularly the history they generate already provide full reproduciblity. However, particularly if comparing multiple datasets, the impact of well-formatted reports that are automatically generated cannot be overestimated. By using templates, reports can be fully customised and are basically unlimited with respect to the output format.

"Cooking" such a recipe (by issuing a single command in the terminal) results in both, performing all the tasks described therein and writing a history in form of a fully working recipe containing all parameters – explicit and implicit.

Outlook: What's still missing for FAIR (EPR) data?

ASpecD and derived packages provide a basis for FAIR data, focussing on reproducibility, provenance, and documentation. Missing are, *e.g.*, information on the samples, persistent identifiers for datasets and alike, repositories and data catalogues. All this is part of LabInform currently being developed.

LabInform

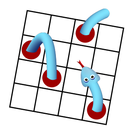
Laboratory Information and Management system, providing, *i.a.*, PIDs and an ELN

https://www.labinform.de/ – Reference: [5]

FitPy

Advanced fitting of models to spectroscopic data, including sampling and global fitting

https://docs.fitpy.de/ – Reference: [6]



[1] M. D. Wilkinson et al., *Sci. Data* 2016, **3**, 160018
[2] J. Popp, T. Biskup, *Chem. Meth.* 2022, **2**, e202100097
[3] M. Schröder, T. Biskup, *J. Magn. Reson.* 2022, **335**, 107140
[4] J. Popp, M. Schröder, T. Biskup, 2022, https://docs.trepr.de/, doi:10.5281/zenodo.4897112
[5] T. Biskup, *ChemRxiv*, 2022, doi: 10.26434/chemrxiv-2022-vz360
[6] T. Biskup, 2021, https://docs.fitpy.de/, doi:10.5281/zenodo.5920380

All software is available via GitHub and PyPI free of charge under a permissive BSD license.

This poster is available on Zenodo: