

PyDPEET: A Python Package for Fast and Easy Battery Data Unification, Processing, and Analysis

M. Otto *, A. Schlösser *, D. Schröder, C. De Simone, A. Hinrichsen, J. Kalisch, and J. Kowal

TU Berlin, Electrical Energy Storage Technology, Einsteinufer 11, D-10587 Berlin

* These authors contributed equally to this work.

Introduction

PyDPEET ("Data Processing for Electrical Energy Storage Technologies") is an open-source Python package developed to facilitate battery data analysis. It targets a common problem: lab and field tests produce large, mixed datasets in different file formats. Manual pre-processing is slow, error-prone, and hard to reproduce. PyDPEET offers a transparent workflow that standardises raw files, processes them with consistent rules, and provides evaluation functionality within a highly integrated code base. The package also offers full user flexibility via custom parameter configuration.

Outlook

Potential Applications

- Fast, easy, and consistent research data analysis
- Large-scale battery data analysis and meta studies
- Feature labelling for machine learning and neural networks

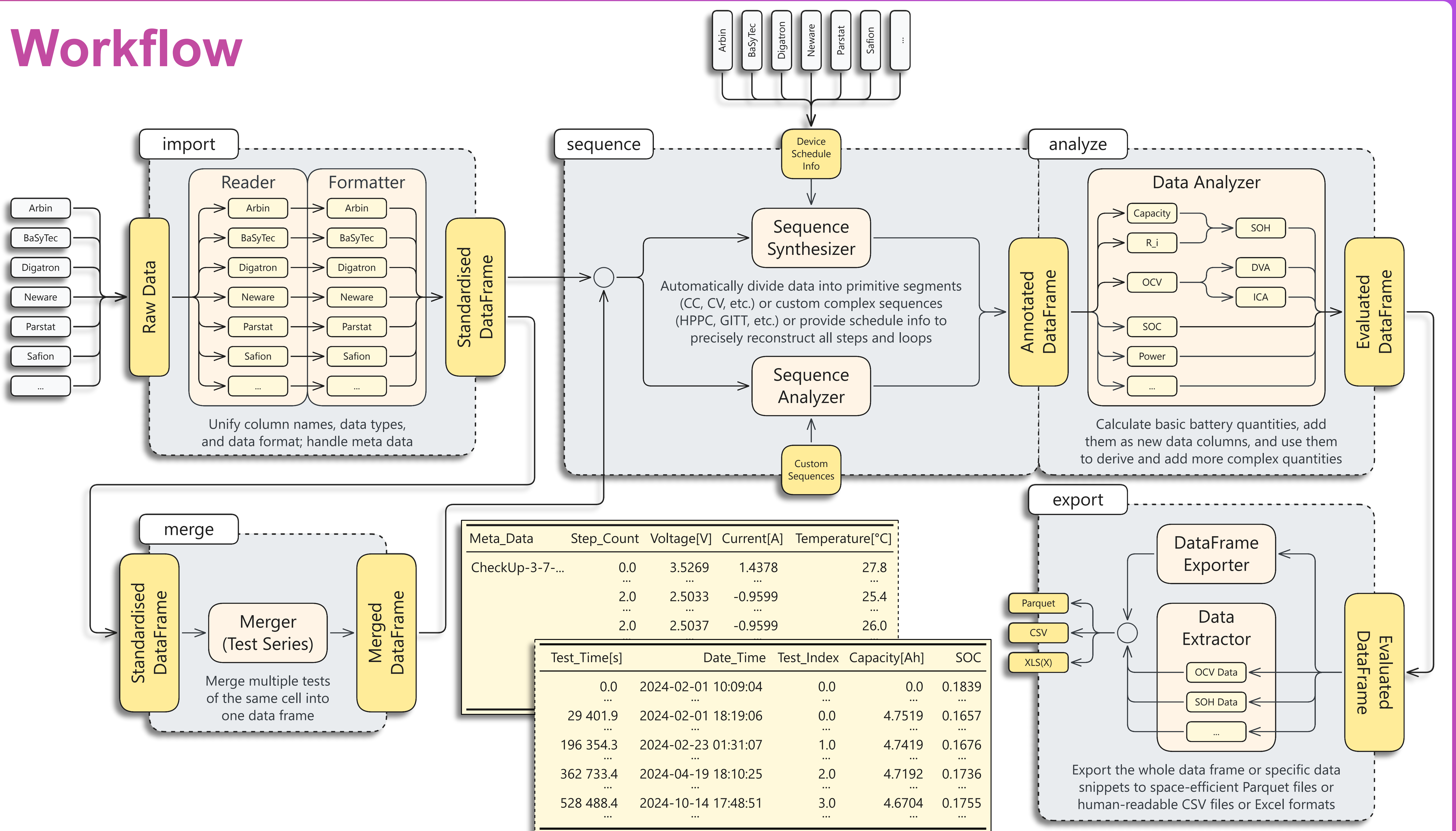
Future Features

- Processing of electrochemical measurement results (EIS, three-electrode experiments, etc.)
- Field data processing incl. data validation and reconstruction
- Integration of modelling and model parameterisation



Star us on GitHub!

Workflow



Usage Example

Experimental Setup

- Calendar ageing test with four checkups
- Storage at elevated temperature in between

Processing

- Read all test files from folder
- Merge individual tests into a single data frame

Analysis

- Add quantities (capacity, SOC, R_i, etc.)
- Extract features (OCV curves, DVA/ICA, etc.)



https://github.com/eet-tu-berlin/pydpeet/blob/main/docs/examples/research/2026_KWB/Usage_Example.ipynb

