




Flexible, Automated Potentiostat/Galvanostat Control Software for Electrochemical Analysis

Alexander J. H. Bell^{1,2}, James A. Dawson², and Michael R. C. Hunt¹

¹ Department of Physics, Durham University, Durham, DH1 3LE, United Kingdom ² Chemistry – School of Natural and Environmental Sciences, Newcastle University, Newcastle upon Tyne, NE1 7RU, United Kingdom

DOI: [10.xxxxxx/draft](https://doi.org/10.xxxxxx/draft)

Software

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

Editor: 

Submitted: 27 November 2025

Published: unpublished

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](https://creativecommons.org/licenses/by/4.0/)).

Summary

We present an open-source graphical user interface (GUI)-based Python program for running and automating electrochemical experiments supporting cyclic voltammetry (CV), linear sweep voltammetry (LSV), Galvanostatic charge/discharge cycling (GCD), chronoamperometry (CA), chronopotentiometry (CP), self-discharge (SD), and rate-testing. Designed to work with low-cost hardware, and specifically an open-source potentiostat/galvanostat design by Dobbelaere et al. (2017), it provides users with access to electrochemical research capabilities previously constrained to far more expensive apparatus. The program was developed with the following core objectives:

- A queuing and automation of experiments.
- Support for complex experiment workflows, e.g., automated open-circuit potential (OCP) equilibration with the use of “OCP” as an input parameter.
- Accessibility to users with little experience performing electrochemical experiments *via* input parameter validation, GUI dialogue boxes, and comprehensive hover-over tooltips.
- Minimal memory usage by clearing data buffers and enabling of widget resizing *via* GUI software options menu to support a range of hardware resolution and memory configurations.

Statement of need

Low-cost, “do-it-yourself” (DIY) potentiostats are increasingly widening in utility, with the CheapStat (Rowe et al., 2011), DStat (Dryden & Wheeler, 2015), and HunStat (Vamos & Kertesz, 2024) providing users the ability to conduct electrochemical experiments at a lower cost than with commercial hardware. However, these are constrained to potentiostatic experiments across limited ranges (± 1 V for the CheapStat and DStat, and ± 1.65 V for the HunStat). This capability was greatly extended by the USB Potentiostat/Galvanostat design presented by Dobbelaere et al. (2017), with a ± 8 V potential range enabling the investigation of battery cells at much higher cell potentials. The addition of a galvanostatic mode, and the ability to accurately measure and apply current from nanoamps to ± 25 mA, has significantly broadened the suite of electrochemical experiments accessible to low-cost DIY devices, with the lowest-cost commercial substitute around 20 times more expensive at \$1900 (Admiral Instruments, 2025). The MYSTAT, a modification of the USB Potentiostat/Galvanostat, pushes this capability even further, achieving maximum potential and current ranges of ± 12 V and ± 200 mA, respectively (Irving et al., 2021).

As is typical for low-cost open source devices, the accompanying GUI software is limited, however, to a narrow range of experiments (CV, GCD, and rate-testing experiments for the USB Potentiostat/Galvanostat and the MYSTAT), single-experiment execution, and basic input

parameters (Caux et al., 2022; Dobbelaere et al., 2017; Dryden & Wheeler, 2015; Irving et al., 2021; Kellner et al., 2015; Rowe et al., 2011; Vámos & Kertész, 2024). Commercial potentiostat/galvanostat instruments, whilst broader in utility, lack open source and customisable software for experiment scheduling, data management, and advanced control.

Our software bridges this gap and provides a flexible, extendable, and user-friendly GUI controller for potentiostat/galvanostat systems, supporting complex experiment workflows and advanced control features. It equips users of the low-cost DIY USB Potentiostat/Galvanostat with functionality often confined to the proprietary software of far more expensive instruments, thereby significantly enhancing the accessibility and experimental capabilities for researchers within the low-cost DIY potentiostat/galvanostat ecosystem.

Features

- Supports CV, LSV, GCD, CA, CP, SD, and rate-testing experiments.
- Experiment queueing using comma-separated variable lists as input parameters.
- Complex experiment workflows by supporting a range of parameter inputs such as potential limits of “1.0, OCP, 2.0, ...” or ramp rates of “5.0, STEP, 10.0, ...”, etc.
- Potential and current limits to prevent unintended damage to the cell under testing, and equilibration parameters to advance experiments when equilibration conditions are satisfied.
- Alternating potential/current sequences for CA/CP experiments, allowing the user to provide alternating equilibration conditions within the experiment.
- Hover-over tooltips to guide users on supported parameters and their experimental implications, and parameter validity checks with detailed user feedback and correction suggestions.
- Automated OCP equilibration and/or fixed-delay between experiments.
- Automated calculation of 1C capacity for rate-testing using GCD analysis.
- Real-time data logging for individual experiments and experiment progress information logging to a summary file for all queued experiments.
- Progress bars displaying the remaining time, cycles, segments, etc., until all experiments are complete.
- Real-time data visualisation options, including comparisons between experiments within the same queue.
- Post-experiment data visualisation.
- Compatible with Linux, Windows, and Mac operating systems and designed to run on light-weight hardware.
- Can be straightforwardly modified for compatibility with the MYSTAT.

Operating principles

The software is designed to be entirely controlled *via* the GUI window built using PySide6 (PyPI, 2025), with the users constructing their experiments using the input fields and checkboxes in the corresponding experiment tab. A CHECK button for each tab executes an input parameter validation, guiding the user to invalid or incompatible inputs using dialogue boxes. A green CHECK button indicates to the user that their experiments are valid and ready to run. An example tab for rate-testing experiments with valid input parameters can be seen in Figure 1. Universal parameters such as the OCP equilibration timescale, tolerance, and timeout threshold, together with more advanced experiment-specific features, are accessible through a software options menu.

The program sequentially executes the queued experiments, handles real-time logging of raw data to individual experiment data files, and writes experiment progress information to a human-readable summary file for all queued experiments. An example GUI seen when running

90 cyclic voltammetry experiments, showing the real-time data visualisation options available to
91 the user, is displayed in Figure 2.

92 All experiments are written in a similar code layout, guided by the PEP 8 standards (Van
93 Rossum et al., 2001) and containing detailed comments, to facilitate users in modifying or
94 adding experiments if required. The typical execution stack that handles the initialisation,
95 running, and stopping of experiments is illustrated in Figure 3.

96 The summary file contains experiment metadata written in a standardised format guided by
97 the FAIR principles (Wilkinson et al., 2016). Users may optionally write experiment notes
98 within the GUI to be included at the top of the summary file to assist with post-experiment
99 data contextualisation. An example summary file generated from the CV experiments running
100 in Figure 2 is displayed in Figure 4.

101 To improve software and experimental accessibility, hover-over tooltips are featured for all input
102 parameter fields and checkboxes, guiding the user on their purpose and accepted parameters.
103 An example tooltip is displayed in Figure 5.

Figure 1: The rate-testing experiment tab showing user input fields and checkboxes, with a green CHECK button indicating that the experiment parameters are valid.

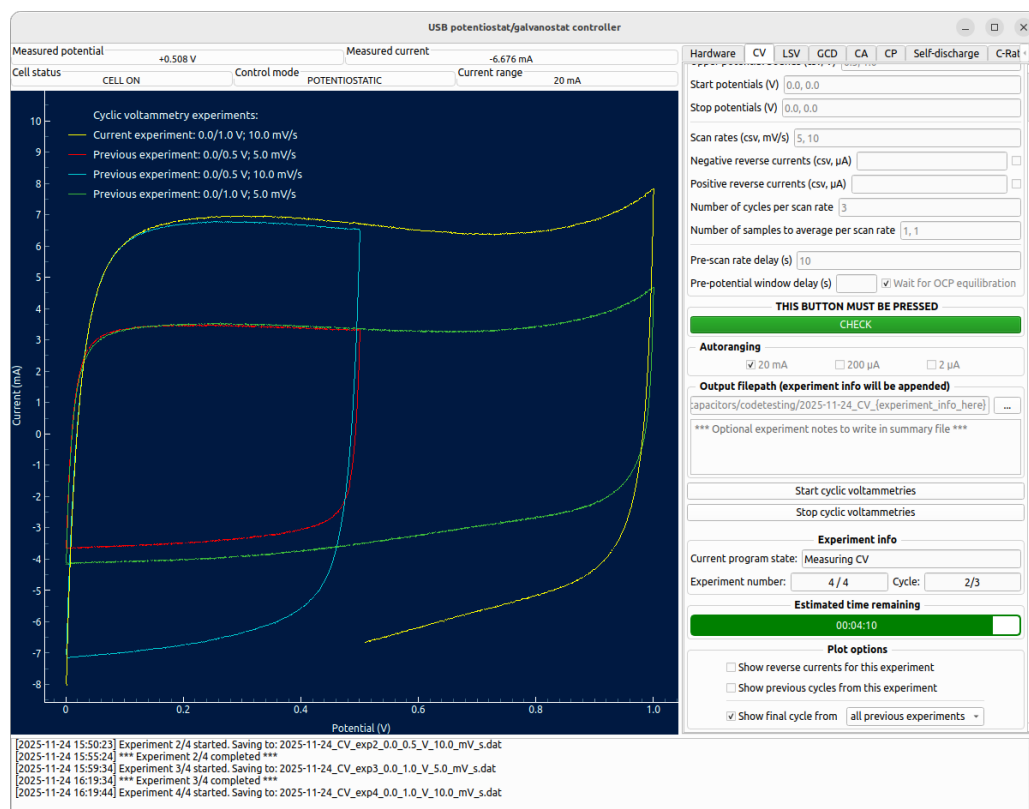


Figure 2: An example GUI during a cyclic voltammetry experiment showing real-time data visualisation options, experiment progress information, and optional experiment notes to be written to the summary file.

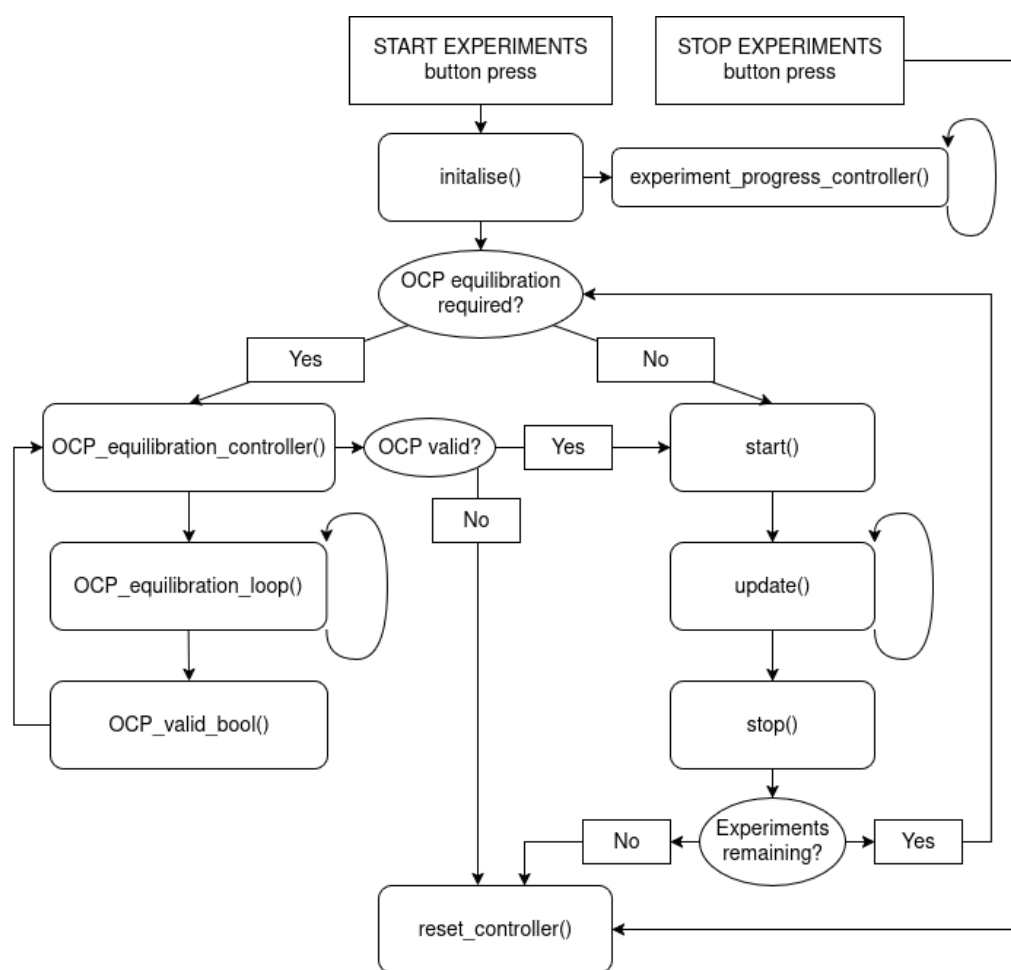
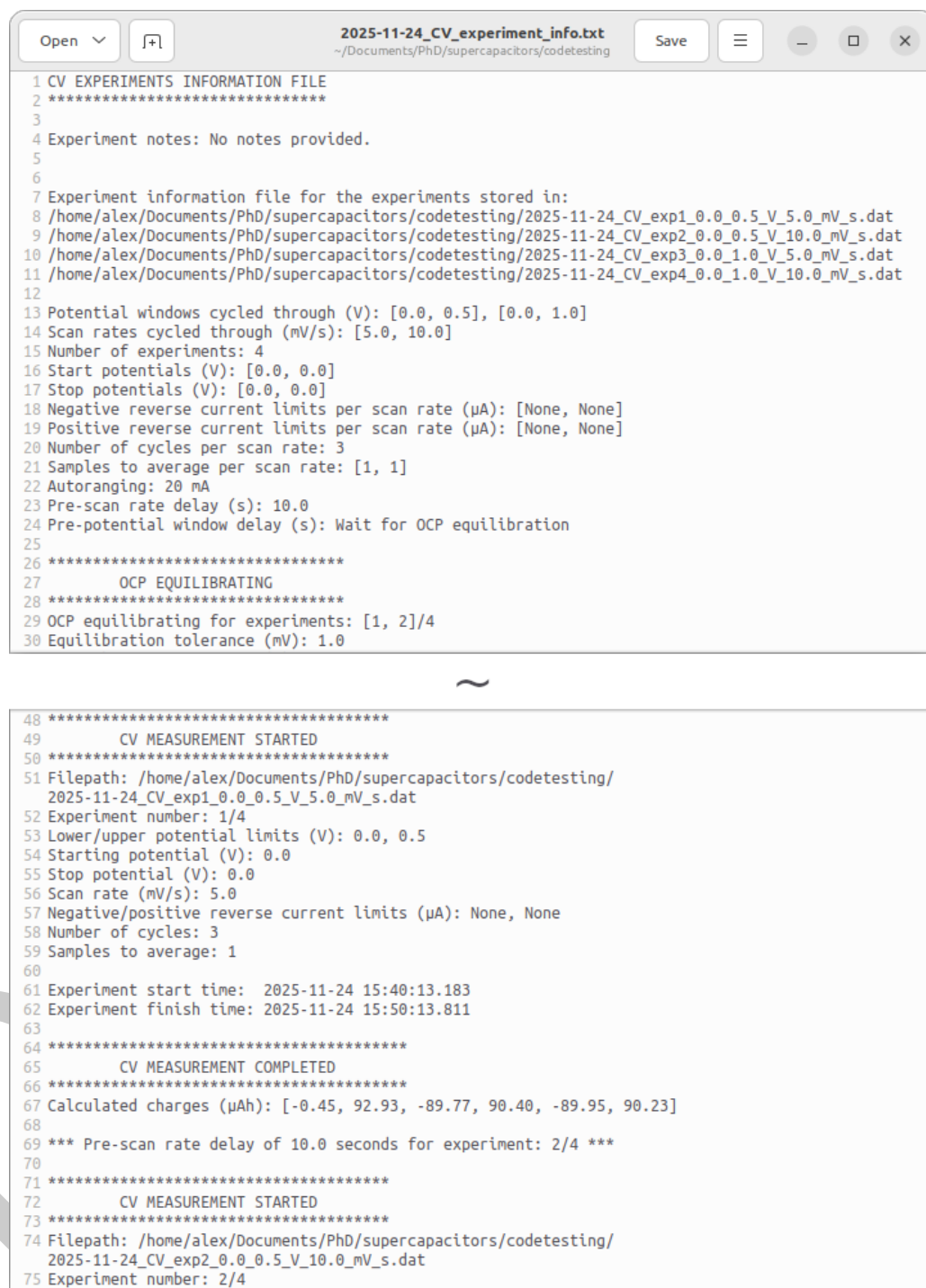


Figure 3: Flowchart of the typical execution stack for a set of experiments.



```

1 CV EXPERIMENTS INFORMATION FILE
2 *****
3
4 Experiment notes: No notes provided.
5
6
7 Experiment information file for the experiments stored in:
8 /home/alex/Documents/PhD/supercapacitors/codetesting/2025-11-24_CV_exp1_0.0_0.5_V_5.0_mV_s.dat
9 /home/alex/Documents/PhD/supercapacitors/codetesting/2025-11-24_CV_exp2_0.0_0.5_V_10.0_mV_s.dat
10 /home/alex/Documents/PhD/supercapacitors/codetesting/2025-11-24_CV_exp3_0.0_1.0_V_5.0_mV_s.dat
11 /home/alex/Documents/PhD/supercapacitors/codetesting/2025-11-24_CV_exp4_0.0_1.0_V_10.0_mV_s.dat
12
13 Potential windows cycled through (V): [0.0, 0.5], [0.0, 1.0]
14 Scan rates cycled through (mV/s): [5.0, 10.0]
15 Number of experiments: 4
16 Start potentials (V): [0.0, 0.0]
17 Stop potentials (V): [0.0, 0.0]
18 Negative reverse current limits per scan rate (µA): [None, None]
19 Positive reverse current limits per scan rate (µA): [None, None]
20 Number of cycles per scan rate: 3
21 Samples to average per scan rate: [1, 1]
22 Autoranging: 20 mA
23 Pre-scan rate delay (s): 10.0
24 Pre-potential window delay (s): Wait for OCP equilibration
25
26 *****
27 OCP EQUILIBRATING
28 *****
29 OCP equilibrating for experiments: [1, 2]/4
30 Equilibration tolerance (mV): 1.0
31
32 ~
33
34 *****
35 CV MEASUREMENT STARTED
36 *****
37 Filepath: /home/alex/Documents/PhD/supercapacitors/codetesting/
38 2025-11-24_CV_exp1_0.0_0.5_V_5.0_mV_s.dat
39 Experiment number: 1/4
40 Lower/upper potential limits (V): 0.0, 0.5
41 Starting potential (V): 0.0
42 Stop potential (V): 0.0
43 Scan rate (mV/s): 5.0
44 Negative/positive reverse current limits (µA): None, None
45 Number of cycles: 3
46 Samples to average: 1
47
48 Experiment start time: 2025-11-24 15:40:13.183
49 Experiment finish time: 2025-11-24 15:50:13.811
50
51 *****
52 CV MEASUREMENT COMPLETED
53 *****
54 Calculated charges (µAh): [-0.45, 92.93, -89.77, 90.40, -89.95, 90.23]
55
56 *** Pre-scan rate delay of 10.0 seconds for experiment: 2/4 ***
57
58 *****
59 CV MEASUREMENT STARTED
60 *****
61 Filepath: /home/alex/Documents/PhD/supercapacitors/codetesting/
62 2025-11-24_CV_exp2_0.0_0.5_V_10.0_mV_s.dat
63 Experiment number: 2/4

```

Figure 4: Sections of the summary file generated for the CV experiments seen running in Figure 2, with a base filename of 2025-11-24.

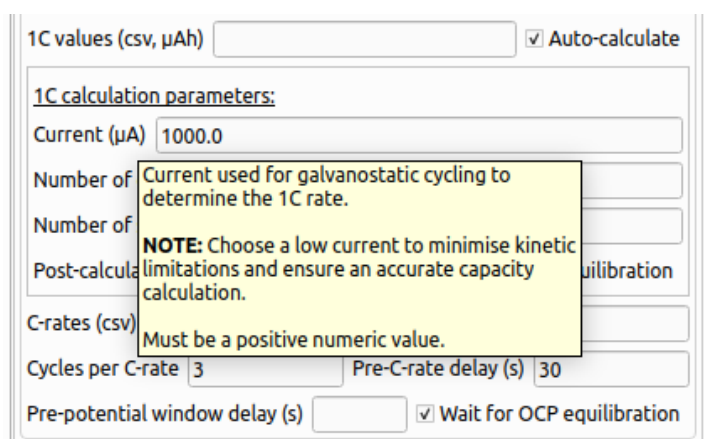


Figure 5: A tooltip for the current input when automatically calculating 1C for rate-testing experiments.

Acknowledgements

This work was supported by the UK Engineering and Physical Sciences Research Council and the Renewable Energy Northeast Universities (ReNU) CDT (EP/S023836/1).

References

- Admiral Instruments. (2025). *Squidstat solo*. <https://www.admiralinstruments.com/product-page/squidstat-solo>.
- Caux, M., Achit, A., Var, K., Boitel-Aullen, G., Rose, D., Aubouy, A., Argentieri, S., Campagnolo, R., & Maisonhaute, E. (2022). PassStat, a simple but fast, precise and versatile open source potentiostat. *HardwareX*, 11, e00290. <https://doi.org/10.1016/j.ohx.2022.e00290>
- Dobbelaere, T., Vereecken, P. M., & Detavernier, C. (2017). A USB-controlled potentiostat/galvanostat for thin-film battery characterization. *HardwareX*, 2, 34–49. <https://doi.org/10.1016/j.ohx.2017.08.001>
- Dryden, M. D., & Wheeler, A. R. (2015). DStat: A versatile, open-source potentiostat for electroanalysis and integration. *PloS One*, 10(10), e0140349. <https://doi.org/10.1371/journal.pone.0140349>
- Irving, P., Cecil, R., & Yates, M. (2021). MYSTAT: A compact potentiostat/galvanostat for general electrochemistry measurements. *HardwareX*, 9, e00163. <https://doi.org/10.1016/j.ohx.2020.e00163>
- Kellner, K., Posniecek, T., Ettenauer, J., Zuser, K., & Brandl, M. (2015). A new, low-cost potentiostat for environmental measurements with an easy-to-use PC interface. *Procedia Engineering*, 120, 956–960. <https://doi.org/10.1016/j.proeng.2015.08.820>
- PyPI. (2025). *PySide6: Python bindings for the Qt cross-platform application and UI framework*. <https://pypi.org/project/PySide6/>.
- Rowe, A. A., Bonham, A. J., White, R. J., Zimmer, M. P., Yadgar, R. J., Hobza, T. M., Honea, J. W., Ben-Yaacov, I., & Plaxco, K. W. (2011). Cheapstat: An open-source, “do-it-yourself” potentiostat for analytical and educational applications. *PloS One*, 6(9), e23783. <https://doi.org/10.1371/journal.pone.0023783>
- Vamos, I., & Kertesz, V. (2024). HunStat—a simple and low-cost potentiostat for analytical and educational purposes. *Analytical Methods*, 16(25), 4198–4204. <https://doi.org/10.1039/D4AY00791C>

- 134 Van Rossum, G., Warsaw, B., & Coghlan, N. (2001). *PEP 8 – style guide for python code*.
135 <https://peps.python.org/pep-0008/>.
- 136 Wilkinson, M. D., Dumontier, M., Aalbersberg, I. J., Appleton, G., Axton, M., Baak, A.,
137 Blomberg, N., Boiten, J.-W., Silva Santos, L. B. da, Bourne, P. E., Bouwman, J., Brookes,
138 A. J., Clark, T., Crosas, M., Dillo, I., Dumon, O., Edmunds, S., Evelo, C. T., Finkers,
139 R., ... Mons, B. (2016). The FAIR Guiding Principles for scientific data management and
140 stewardship. *Scientific Data*, 3(1), 160018. <https://doi.org/10.1038/sdata.2016.18>

DRAFT