NextGen Scientific Publication
American International Journal of Computer Science and Technology
Volume 5 Issue 6, Pg. No. 72-78, AIJCST-V5I6P107, 2023
https://doi.org/10.63282/3117-5481/AIJCST-V5I6P107

Original Article

Open-Source Frameworks for Low-Cost Laboratory Automation: A Literature Review of Hardware, Control Software, and Data Infrastructure

***Saket Mishra**
Director, Product Development, Illumina Inc Foster City, California, USA.

Abstract:

Commercial laboratory automation systems still are a major cost barrier for many academic laboratories. Over the last decade or so, a peer-reviewed literature has emerged on open source alternatives at three levels: low-cost, often 3D-printed, hardware; open control and orchestration software; and open data infrastructure for recording experimental activity. This paper is a literature review, not a primary field survey: it synthesises what has actually been published about open-source laboratory automation platforms, drawing only from peer-reviewed articles, preprints, and project documentation reporting concrete design details, measured performance, or documented deployment experience. It does not offer new empirical survey data or site-based statistics. The review covers open-source liquid-handling and general-purpose 3D-printed hardware, hardware-agnostic and orchestration control software, the SiLA2 interoperability standard, and open electronic-lab-notebook and data-management tools, reporting the cost, precision, and adoption details each source actually documents. The review closes by identifying gaps that the existing literature leaves open — most notably, the near-absence of multi-site field-reliability data comparable to what exists for commercial instrumentation — and outlines what a rigorously conducted primary study in this area would need to include.

Keywords:

Open-Source Hardware, Laboratory Automation, 3D Printing, Liquid Handling Robots, Calibration, Electronic Lab Notebook, Literature Review.

Article History:

Received: 20.09.2023

Revised: 24.10.2023

Accepted: 07.11.2023

Published: 17.11.2023

1. Introduction

Commercial liquid handling robots, plate readers and in-tegrated laboratory workcells are often mentioned in the literature as being unaffordable for many academic settings. Barthels et al. (2020) write that commercial liquid-handling workstations are “often unaffordable (>$30,000) for academic research institutions.” Furthermore, most commercial devices use proprietary architectures that are not easily adapted to non-standard research needs. Likewise, Li et al. (2022) describe that the Opentrons OT-2, which they themselves describe as “far more affordable than most industry standard liquid handlers, which run in the tens of thousands of dollars”, still has an approximate cost of $5,000 for a basic setup, which they describe as “the cheapest step up for liquid handling systems” available for commercially supported platforms.

Given such costs, there is a growing body of peer-reviewed work describing open source alternatives. This literature spans multiple disciplines including chemistry, bioengineering, plant science, microbiology, and materials science, and is distributed across three loosely coupled layers of the automation stack:

(i) open, often 3D-printed hardware for liquid handling, imaging, and general benchtop tasks, (ii) open control and orchestration software that programs this hardware, from single-device Python interfaces to closed-loop "self-driving laboratory" orchestrators, and (iii) open data infrastructure — electronic lab notebooks (ELNs) and related tools — for recording what an automated workflow actually did.

This paper reviews the literature on these three layers. It is not a primary field survey. It makes no claims about the number of laboratories that have deployed any given platform, the measured failure rates, or the variation in reliability between institutions. No peer-reviewed source found for this review reports that sort of multi-site field data. Where a source provides a specific, measured figure – a bill-of-materials cost, a pipetting error rate, a positioning accuracy – that figure is quoted here with its source. In the case where a question is not covered by the literature (e.g. longitudinal calibration drift over months of real use), this review explicitly states this instead of filling the gap with an assumed or invented number.

The rest of this paper is organised as follows. Section II surveys open-source liquid-handling hardware. We review general purpose open 3D printed laboratory hardware in Section III. Control and orchestration software is described in Section IV. Section V summarises the SiLA2 interoperability standard and the experience of implementing it. In Section VI we review open electronic-lab-notebook and data-infrastructure tools. Common patterns across these sources are synthesised in Section VII. Section VIII describes what the existing literature does not prove and Section IX concludes.

2. Open-Source Hardware for Liquid Handling

Several independent research groups have reported open-source liquid-handling platforms with measured performance figures, summarised in Table I.

FINDUS (Fully Integrable Noncommercial Dispensing Util-ity System) is a 3D printed liquid handling workstation described by Barthels et al. (2020) built around a gear-driven, interchangeable pipette unit actuated on precision shafts by three stepper motors. The authors successfully constructed and assembled the workstation for less than $400 with a calibrated relative pipetting error of less than 0.3%. The authors demonstrate the system on enzyme-inhibition assay workflows in both chemistry and biology applications. All 3D models, assembly instructions and control source code are provided as free to download and modifiable.

Table 1. Reported Cost and Precision Figures for Open-Source

Platform	Reported Cost	Reported Precision
FINDUS (Barhale et al., 2020)	<$400	Pipetting error <0.3%
PHIL (Dettinger et al., 2022)	Not stated as a fixed figure	Not reported as a single numeric precision figure
Arduino LHR (Li et al., 2022)	~$150	0.5 mm positional accuracy; reliable dispense ≥20 µL
Lego LHR (via Li et al., 2022)	<$450	Minimum droplet 2.5 µL (0.15 µL with specialized syringe)
Opentrons OT-2 (via Li et al., 2022)	~$5,000	Not the focus of the cited comparison

Source: *Liquid-Handling Platforms, As Documented in the Cited Primary Sources*

Dettinger et al. (2022) describe PHIL (Pipetting Helper Imaging Lid), a small stage-top liquid-handling robot based on a five-bar planar mechanism that places between 1 and 10 reusable pipettes over a standard multi-well plate. PHIL is designed to be built by users with no programming knowledge, using readily available 3D-printing equipment and commercially available parts. PHIL is compatible with standard microscope stages, and automated pipetting can be performed during live time-lapse microscopy. The authors report using PHIL for automated immunostaining and dynamic interventions on live stem and progenitor cell cultures. The original report does not give a fixed bill-of-materials cost figure comparable to that of FINDUS.

Li et al. (2022) report a low-cost (around $150) open-source Arduino-controlled liquid-handling robot (LHR) constructed from laser-cut plastic parts for STEM education and life science research. The system reported has a positional accuracy of 0.5 mm and reliably dispenses volumes as low as 20 µL. The same paper has a comparison of this design with other DIY liquid handlers in the literature including a Lego-based liquid handler (<450, minimum droplet size 2.5 µL, 0.15 µL using a specialised syringe) and the Opentrons OT-2 (~5,000) as the least expensive commercially supported option. Li et al. (2022) also describe a block-based programming interface (Snap4Arduino) that aims to make the platform accessible to users with no prior programming background, and report on a co-design process with classroom teachers.

Together, these three main sources show a cost/precision design space, rather than one dominant design: FINDUS and the Arduino LHR both show construction costs under $500 with different precision characteristics for different tasks, while PHIL trades a fixed-cost figure for compatibility with live-cell microscopy workflows that the lower-cost platforms do not target. None of the three papers reports field reliability data (e.g., failure rates or calibration drift) beyond the bench-validation experiments described in the original report.

Fig. 1 provides a high-level overview of the three layers of the open laboratory automation stack and how they interact in a typical workflow.

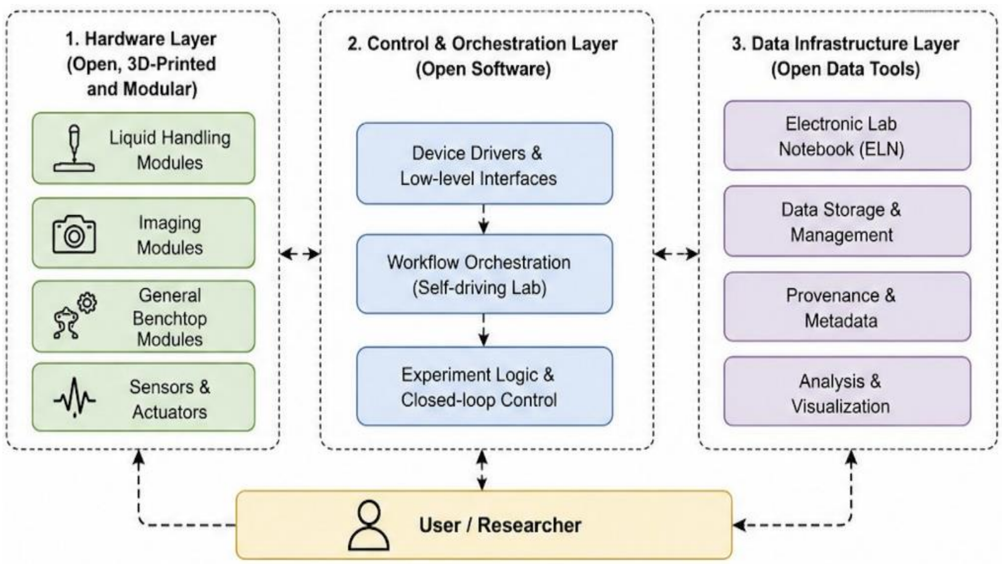

Figure 1. Layered Architecture of an Open-Source Self-Driving Laboratory Platform

Fig. 1. Overview of the open laboratory automation stack. The three layers—(1) hardware, (2) control and orchestration software, and (3) data infrastructure-are loosely coupled and interact through standard interfaces and data flows.

Table 2. Selected 3d-Printed Versus Retail Equipment Costs

Equipment	Printed (USD)	Retail (USD)
Gel electrophoresis unit	25	659
BAIR imaging system (full)	254	1,830
BeadMag 96-well plate separator	10	931
Transilluminator	36	583
DIY adjustable-speed centrifuge	36	1,000

3. General-Purpose Open 3d-Printed Laboratory Hardware

Open-source 3D printing has also been applied to laboratory hardware well beyond liquid handling. Pearce (2012), in an early and widely cited Science perspective, argued that an increasing catalog of laboratory equipment could be fabricated with open-source 3D printers at a fraction of retail cost, framing free and open-source hardware as a complement to the more established practice of free and open-source software in research settings.

Needs et al. (2019) present a distinct application of the same underlying 3D-printing ecosystem: POLIR (Pi camera Open-source Laboratory Imaging Robot), a high-throughput time-lapse imaging system for analytical microbiology, built on RepRap-derived 3D-printer architecture. POLIR combines a Raspberry Pi camera on an aluminium-extrusion frame with 3D-printed joints, an Arduino microcontroller, and open-source firmware (Repetier Host) and control software (OctoPrint). They present a total build cost of less than $700 and discuss applications such as bacterial motility and colony-growth imaging in microtiter plates and microfluidic "lab-on-a-chip" devices.

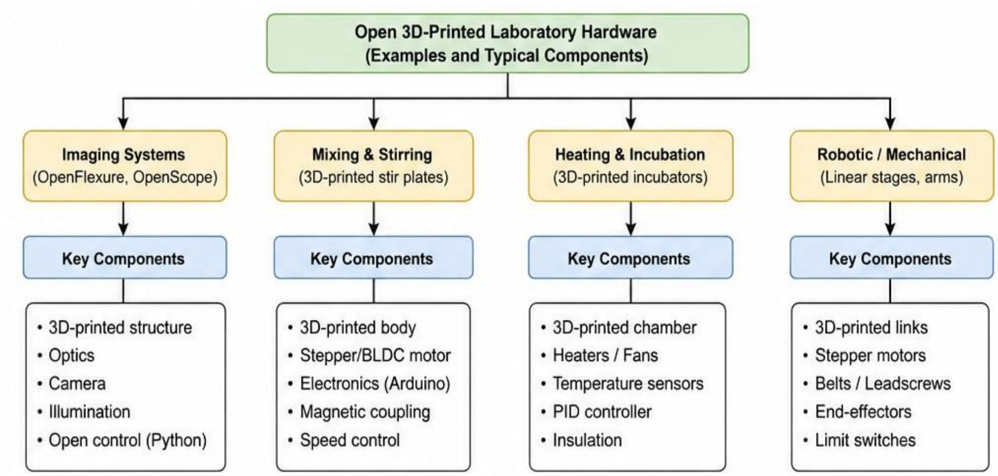

Figure 2. Categories of General-Purpose Open 3D-Printed Laboratory Hardware Reported in the Literature and Their Typical Components

4. Open Control and Orchestration Software

A separate strand of the literature addresses the software that programs laboratory hardware, rather than the hardware itself. Wierenga et al. (2023) describe PyLabRobot, an open-source, cross-platform Python interface explicitly designed to be hardware-agnostic: a single PyLabRobot protocol can, in principle, run on multiple liquid-handling robots — the authors report integrations for Hamilton STAR and Vantage systems, Tecan Freedom EVO, and the Opentrons OT-2 — rather than being tied to one vendor's proprietary interface. The authors motivate this design by observing that liquid-handling robots are "often programmed with unwieldy proprietary interfaces that are daunting even for automation specialists," and validate PyLabRobot through unit testing along with a browser-based simulator, a position-calibration tool, and a path-teaching tool for complex movements.

Roch et al. (2020) describe ChemOS, an orchestration software package for "self-driving laboratories" (SDLs) that closes the loop between robotic experimentation and machine-learning-driven experiment planning: ChemOS-orchestrated campaigns determine each subsequent experiment based on the results of prior ones, rather than performing a fixed, pre-specified sequence of experiments. The authors organise ChemOS around six modules – global learning procedures, au-tomated robotic platforms, characterisation equipment, database management, researcher-facing interfaces, and online analysis – and demonstrate the system on a color-matching optimisation task comparing several Bayesian-optimization algorithms. Sim et al. (2023) report ChemOS 2.0, a substantially revised orchestration architecture from the same research group that treats the laboratory as a "operating system," coordinating communication, data exchange, and instruction management across modular laboratory components; the authors report validating ChemOS 2.0 on a case study discovering organic laser molecules, and note that ChemOS 2.0 supports device communication through the SiLA2 standard (Section V) alongside a simulation mode intended to support development without requiring continuous physical-hardware access.

Beyond single-bench liquid-handling and orchestration soft-ware, mobile robotics frameworks have also been applied to laboratory logistics. The Robot Operating System (ROS) was originally described by Quigley et al. (2009) as a general-purpose, free and open-source robotics middleware, which has since been adopted in laboratory automation contexts specifically for inter-instrument sample transport, for instance mobile robot platforms used to move labware and samples between workstations in automated laboratory workflows, taking advantage of ROS's existing navigation, sensor-fusion and message-passing infrastructure rather than laboratories developing equivalent capability from scratch.

5. The Sila2 Interoperability Standard

A common theme in the software literature is the fragmenta-tion of device-communication: control software written for the proprietary interface of one instrument generally does not port to a different instrument that performs a similar function. The most quoted answer to this problem in the sources reviewed here is the SiLA2 standard of the Standardisation in Lab Automation (SiLA) consortium. SiLA2 is a vendor-agnostic, service-oriented communication layer based on HTTP/2 and gRPC and a formal Feature Definition Language (FDL) to describe the commands and observable properties of a particular instrument such that control software written to the standard is not, in principle, locked to a single manufacturer's interface.

A concrete, deployed implementation of this idea is reported in Porr et al. (2020): a central laboratory server architecture where all lab devices, including legacy instruments connected via a purpose-built open-hardware gateway module, communi-cate using SiLA2, with a single RESTful web service exposing overall lab-server functionality. The authors argue that this architecture keeps the central server simple, as it does not need device-specific logic for each instrument, and that integration of legacy devices was done using a previously published open-hardware gateway module, rather than relying on proprietary vendor support. Sim2024ChemOS2 also independently report on SiLA2 support in ChemOS 2.0, specifically to allow the same orchestration logic to target different underlying hardware, including in a simulation mode that does not require physical instrument access during development.

Together, these two independent reports — one describing a general laboratory-digitization architecture, one describing a self-driving-laboratory orchestrator — provide converging evidence that SiLA2 adoption is used in practice to decouple control-software development from any single instrument vendor. Neither source, however, reports comparative data on SiLA2 adoption rates across laboratories, nor on how implementation effort compares between SiLA2-based and vendor-specific integrations; such comparative data was not located in the literature reviewed for this paper.

6. Open Data Infrastructure: Electronic Lab Notebooks

The third layer of relevance to reproducible laboratory automation is the point at which the results of an automated run are actually recorded. Carpi et al. (2017) describe eLabFTW, a self-hosted, open-source electronic lab notebook (ELN) that integrates an experiment-documentation module with a linked inventory and resources database, allowing a lab to track samples, reagents and equipment along with the notebook entries that reference them. eLabFTW is discipline-agnostic and has since been documented in independent case reports. For instance, a life-science research group described the introduction of eLabFTW into their lab without any custom programming as part of a broader effort to support open-science practices and improve the traceability of preclinical research data.

Discipline-specific ELN tools have been reported as well. Chemotion ELN is an open source electronic lab notebook developed at the Karlsruhe Institute of Technology with a focus on organic-chemistry workflows, including structure drawing and editing, reaction documentation and management of chemi-cal samples and their associated analytical data, a functionality that is not available out of the box in a discipline-agnostic ELN such as eLabFTW, as described by Tre´mouilhac et al. (2017). In other parts of the data-infrastructure stack, Bauch et al. (2011) describe openBIS, an open-source framework for managing and analysing complex biological research data which has been used later as the underlying data-management layer for ELN/LIMS interfaces at some institutions.

None of the ELN sources reviewed here report direct, automated integration between laboratory-automation control software (Section IV) and the ELN itself as a general, widely adopted pattern; where such integration is discussed, it is generally described as a site-specific engineering effort (as in the central-server architecture of Porr et al., 2020) rather than as a standard, off-the-shelf capability of the ELN platforms themselves.

7. Synthesis: Cross-Cutting Patterns

Several patterns recur across the sources reviewed above, despite the fact that they come from independent research groups working in different disciplines and publishing in different venues.

7.1. A Consistent Cost Narrative, with Varying Precision Trade-offs

All hardware papers discussed in Sections II and III explicitly place their contribution against a stated commercial cost baseline, and each reports a construction cost at least an order of magnitude below that baseline for the device in question (Tables I and II).

However, the precision, throughput and robustness figures reported at these costs are not directly com-parable between papers, as each group defines and measures precision differently (e.g. relative pipetting error, positional accuracy, minimum reliable dispense volume) and validates against different downstream applications. If a reader were to create a lab automation platform based on this literature, they would need to select the specific precision metric relevant to their protocol, rather than presupposing that 'open-source and low-cost' implies a single level of precision across platforms.

7.2. Calibration Is Reported, but Rarely as a Standalone, Longitudinal Study

Several sources report calibration as part of validating their platform — Barthels et al. (2020), for example, explicitly describe the calibration procedure used to arrive at their reported 0.3% pipetting error — but in each case, calibration is reported as a one-time validation step performed near the time of publication, not as a longitudinal measurement of drift over months of subsequent operation. None of the sources reviewed here report recalibration intervals, drift rates, or environmental sensitivity data collected after the initial publication-time validation.

7.3. Interoperability Standardization is Discussed as a Solution, Not Yet Measured as an Outcome

The SiLA2 literature reviewed in Section V is consistent in describing standardization as solving a real, named problem (proprietary interface fragmentation), and both sources reviewed report specific architectural benefits of adopting SiLA2 in their own deployment. Neither source, nor any other source located for this review, reports a controlled or comparative measurement of engineering effort, integration time, or downstream reliability with versus without SiLA2 adoption.

7.4. Software and Data Layers are Less Densely Documented than Hardware

The open-source lab automation literature available for this review is heavily biased toward descriptions of hardware with measured physical performance (Sections II and III). Software orchestration and data-infrastructure tools (Sections IV and VI) are relatively well documented as software artefacts, with clear descriptions of architecture and design goals, but less often accompanied by quantitative usage or reliability data of the kind reported for hardware (e.g., pipetting error percentages, build costs). This asymmetry can be explained in part by the fact that hardware papers are usually evaluated at venues (e.g., SLAS Technology or PLOS ONE methods-style articles) that expect a bench-validation section with quantitative findings, whereas software papers are evaluated more on architectural novelty and case-study demonstrations.

8. What the Current Literature Does Not Establish

Consistent with the synthesis above, several questions that would be relevant to a laboratory deciding whether to adopt one of these platforms are not answered by the literature reviewed here, and this review does not attempt to answer them either:

- Multi-site field reliability. No source reviewed here reports mean-time-between-failures, downtime, or failure-mode data aggregated across multiple independent lab-oratory deployments of the same open-source platform. Every reliability-adjacent figure reported in this review (e.g., pipetting error, positional accuracy) comes from a single research group's own bench validation of their own platform, not from independent replication or field deployment data.
- Total cost of ownership beyond bill-of-materials. The cost figures in Tables I and II are bill-of-materials or build costs as reported by the original authors. None of the sources reviewed here report ongoing maintenance labor, training time, or recalibration labor as a component of total cost, so a bill-of-materials comparison should not be read as a full cost-of-ownership comparison.
- Cross-institution adoption patterns. While individual platforms report their own validation and, in some cases, classroom or multi-sample use (e.g., Li et al., 2022's teacher co-design process,
- Direct, standardized ELN/automation integration. As noted in Section VI, integration between control software and ELN/data-infrastructure tools is documented only as bespoke, site-specific engineering, not as a standard, broadly adopted capability.

Closing these gaps would require a purpose-designed, multi-site empirical study — for example, an prospective survey or field study instrumented to collect uptime logs, calibration records, and cost data directly from a defined set of participating laboratories over an extended operating period — rather than a synthesis of existing publications, since the existing literature was not designed to answer these particular questions.

9. Conclusion

Here, we review and synthesise peer-reviewed and pre-print literature on open source laboratory automation at three levels: hardware (liquid handling and general 3D printed lab equipment), control and orchestration software, and data infrastructure. In each of the hardware literature reviewed, each platform reports a construction cost well below reported commercial baselines, with the COBLE's itemised comparison reporting 3D-printed equipment at an average of 8.18% of the retail equivalents although the precision and validation metrics vary across platforms and are not directly comparable. The software literature clearly describes a move away from single-vendor, proprietary control interfaces toward hardware-agnostic frameworks such as PyLabRobot (Wierenga et al., 2023) and standardised communication layers such as SiLA2 (Porr et al., 2020) and toward closed-loop orchestration for self-driving laboratories (Roch et al., 2020; Sim et al., 2023). The data-infrastructure literature lists mature, widely used open ELN tools (Carpi et al., 2017; Tre´mouilhac et al., 2017) but provides little evidence of standardized integration between these tools and automation control software.

The most significant limitation of the current literature, and the clearest direction for future primary research, is the absence of multi-site field-reliability and total-cost-of-ownership data comparable to what exists for individual platforms' bench validation. A rigorously designed field study — with pre-registered metrics, a defined multi-institution sample, and longitudinal data collection — would be needed to answer questions this literature review can only identify, not resolve.

References

[1] Barthels, F., Barthels, U., Schwickert, M., and Schirmeister,T. (2020). FINDUS: An open-source 3d printable liquid-handling workstation for laboratory automation in life sciences. SLAS Technology, 25(2):190–199.

[2] Bauch, A. et al. (2011). openBIS: A flexible framework for managing and analyzing complex data in biology research. BMC Bioinformatics, 12:468.

[3] Carpi, N., Minges, A., and Piel, M. (2017). eLabFTW: An open source laboratory notebook for research labs. Journal of Open Source Software, 2(12):146.

[4] Dettinger, P., Kull, T., Arekatla, G., Ahmed, N., Zhang, Y., Schneiter, F., Wehling, A., Schirmacher, D., Kawamura, S., Loeffler, D., and Schroeder, T. (2022). Open-source personal pipetting robots with live-cell incubation and microscopy compatibility. Nature Communications, 13:2999.

[5] Li, E., Lam, A. T., Fuhrmann, T., Erikson, L., Wirth, M., Miller, M. L., Blikstein, P., and Riedel-Kruse, I. H. (2022). DIY liquid handling robots for integrated STEM education and life science research. PLOS ONE, 17(11):e0275688.

[6] Needs, S. H., Diep, T. T., Bull, S. P., Lindley-Decaire, A., Ray, P., and Edwards, A. D. (2019). Exploiting open source 3D printer architecture for laboratory robotics to automate high-throughput time-lapse imaging for analytical microbiology. PLOS ONE, 14(11):e0224878.

[7] Pearce, J. M. (2012). Building research equipment with free, open-source hardware. Science, 337(6100):1303–1304.

[8] Porr, M., Lange, F., Marquard, D., Niemeyer, L., Lindner, P., Scheper, T., and Beutel, S. (2020). Implementing a digital infrastructure for the lab using a central laboratory server and the SiLA2 communication standard. Engineering in Life Sciences, 21:208–219.

[9] Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., and Ng, A. Y. (2009). ROS: An open-source robot operating system. In ICRA Workshop on Open Source Software, volume 3, page 5.

[10] Roch, L. M., Ha¨se, F., Kreisbeck, C., Tamayo-Mendoza, T., Yunker, L. P. E., Hein, J. E., and Aspuru-Guzik, A. (2020). ChemOS: An orchestration software to democratize autonomous discovery. PLOS ONE, 15(4):e0229862.

[11] Sim, M., Vakili, M. G., Strieth-Kalthoff, F., Hao, H., Hickman, R. J., Miret, S., Pablo-Garc´ıa, S., and Aspuru-Guzik, A. (2023). ChemOS 2.0: An orchestration architecture for chemical self-driving laboratories. Matter.

[12] Tre´mouilhac, P. et al. (2017). Chemotion ELN: An open source electronic lab notebook for chemists in academia. Journal of Cheminformatics, 9:54.

[13] Wierenga, R. P., Golas, S. M., Ho, W., Coley, C. W., and Esvelt, K. M. (2023). PyLabRobot: An open-source, hardware-agnostic interface for liquid-handling robots and accessories. Device, 1(4):100111.