Connecting lab experiments with computer experiments: Making “routine” simulations routine

Kevin Maik Jablonka[1, †], **Michaël Zasso**[2], **Luc Patiny**[3], **Nicola Marzari**[4,5], **Giovanni Pizzi**[4,5], **Berend Smit**[1,5,*], **and Aliaksandr V. Yakutovich**[1,4,5, †]

[1]Laboratory of Molecular Simulation (LSMO), École Polytechnique Fédérale de Lausanne (EPFL), 1951 Sion, Switzerland

[2]Zakodium Sàrl, 1027 Lonay, Switzerland

[3]Institute of Chemical Sciences and Engineering, École Polytechnique Fédérale de Lausanne (EPFL), 1015 Lausanne, Switzerland

[4]Theory and Simulation of Materials (THEOS), École Polytechnique Fédérale de Lausanne, CH-1015 Lausanne, Switzerland

[5]National Centre for Computational Design and Discovery of Novel Materials (MARVEL), École Polytechnique Fédérale de Lausanne (EPFL), CH-1015 Lausanne, Switzerland

[*]corresponding author: Berend Smit (berend.smit@epfl.ch)

[†]these authors contributed equally

ABSTRACT

Nowadays, computer simulations and experiments are closely interlocked. However, the data and analysis workflows are often barred into “silos” of knowledge — even for routine simulations. Here, we show how a typical electronic laboratory notebook (ELN) environment can be seamlessly integrated with a computational modelling infrastructure. We developed a protocol to initiate advanced molecular or atomic simulations directly from an ELN. Such integration ensures that all the relevant sample and experimental data are transferred from the ELN to the modelling infrastructure, and — once the calculations have completed — back to the ELN. The presented protocol works similar to sending out a sample for external characterisation and enables experimentalists to routinely perform “routine” simulations to compare with their experiments while keeping track of the full experiment and simulation provenance. We illustrate our protocol with some examples of geometry optimisation followed by the calculation of adsorption isotherms, but the implementation can be readily generalised to other techniques such as optical absorption or X-ray photoelectron spectroscopy.

Introduction

In chemistry and materials science, the feedback between computer simulations and lab experiments is often crucial. On the one hand, computer simulations are frequently needed to understand, interpret or support some experimental findings. On the other hand, lab experiments are the “ground truth” to validate computational methods. As we will show, although the physics is the same, there are numerous practical barriers for a seamless exchange between simulations and experiments that make a tight integration more difficult than perhaps necessary. This is notable as computer simulations have evolved to the extent that “routine” simulations can provide feedback which might be most valuable if the feedback loop between experiment and simulation is as short as possible. However, a calculation that would be considered “routine” for a computational researcher could very well present a significant challenge for an experimental scientist.

It is hard to make “routine” calculations straightforwardly available in a mixed experimental-simulation environment. Running simulations requires the management of a sometimes highly complex software environment in a heterogeneous hardware landscape, e.g., when test runs are executed on a local laptop or workstation and production runs are performed on a high-performance computing (HPC) cluster with completely different architecture. Furthermore, one needs to transfer input data to the appropriate compute system and prepare them in a way that it can be processed by the employed simulation and analysis tools. Further, to perform correct simulations, one needs to define the right settings and numerical input parameters for the simulations, with at least some minimal understanding of the limitations and assumptions of the employed physical models and algorithms. Manually designing and successfully implementing such a pipeline is not only error-prone, but requires the consultation of a computational expert. If the DFT optimisation is run manually by an experienced computational researcher, the direct link to the experiment will be lost. This is a further disadvantage in addition to the extra time the computational researcher needs to spend on a "routine" simulation.

From a computational perspective, we can mitigate part of this problem by developing robust workflows[1–3]. Such workflows provide the user with a set of suitable default parameters that are well tested to give reliable results for typical systems. One example is setting the initial magnetisation based on guesses for the oxidation state[4].

In this work, we aim to remove the barrier that is imposed by the need to "provide the right input" and make it possible to carry out "routine" calculations in an integrated experimental and computational setting. In this vision, carrying out a simulation should be comparable to sending out an experimental sample to a central facility for routine characterisation. One specifies which calculations need to be done, but the transfer, pre-processing and creation of input data is fully automated.

In the following, we assume that the experimental group uses an electronic lab notebook (ELN) to collect and store all experimental data on a particular sample. We demonstrate how such an ELN can be connected to a simulation platform that gives easy access to routine but state-of-the-art simulations. Such a connection ensures that experimental results can be compared to simulations on a routine basis. We have chosen the cheminfo ELN[5] as the ELN and AiiDAlab[6] as the simulation platform. Both platforms are open-source and allow storing data provenance, meaning that we have a complete record of how simulations and experiments have been performed and influenced each other. The cheminfo ELN makes sure that all required data are sent to AiiDAlab along with the request for a simulation. Once the simulations are done, the results are automatically linked to the sample and interactively visualised in a web browser. The results contain metadata and the unique identifiers of the calculations, enabling users to trace back at a later time the full provenance of their simulations performed by AiiDAlab.

Architecture

Overview

The cheminfo ELN deployment is cloud-based and, therefore, only requires a web browser to run the application. The cheminfo ELN ensures that all experimental data are automatically converted into a FAIR-compliant format (typically, JCAMP-DX[7]). The metadata containing the type of experiments and sample are added in a standardised format (`cheminfo.github.io/data_schema/`). Due to this standardisation and the availability of a representational state transfer (REST) application programming interface (API), the cheminfo ELN is ideally suited for integration with other services.

The AiiDAlab[6] simulation platform (`materialscloud.org/work/aiidalab`) is built on top of the Jupyter infrastructure[8]. AiiDAlab allows packaging scientific workflows and interactive computational environments using Jupyter notebooks. It provides a complete infrastructure for automated workflows and provenance tracking due to its tight integration with the AiiDA[9–11] computational infrastructure.

Typically, an experimental structure is initially uploaded to the ELN, e.g., as the crystal structure derived from a Rietveld refinement or the molecular structure as the educt or product of synthesis (see Figure 1). In the ELN, users have now access to a button that we developed as part of the current integration work, that initiates the transfer

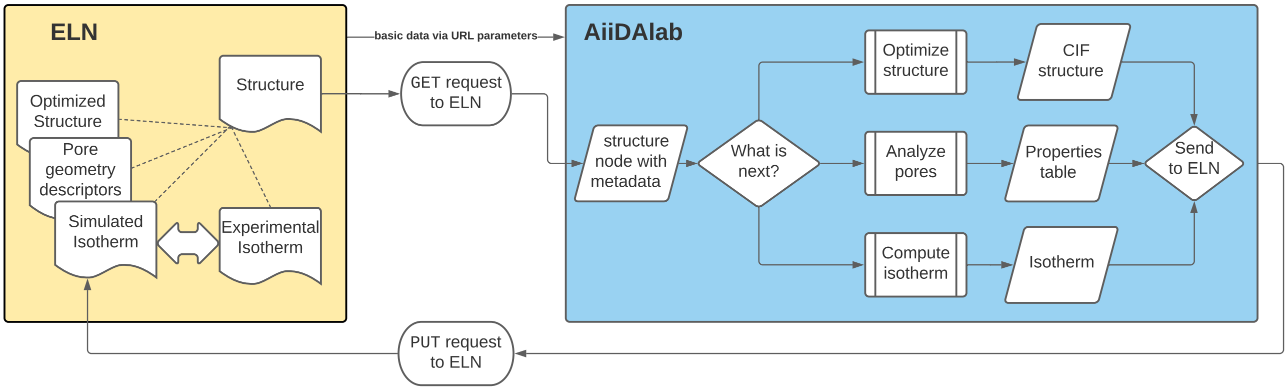


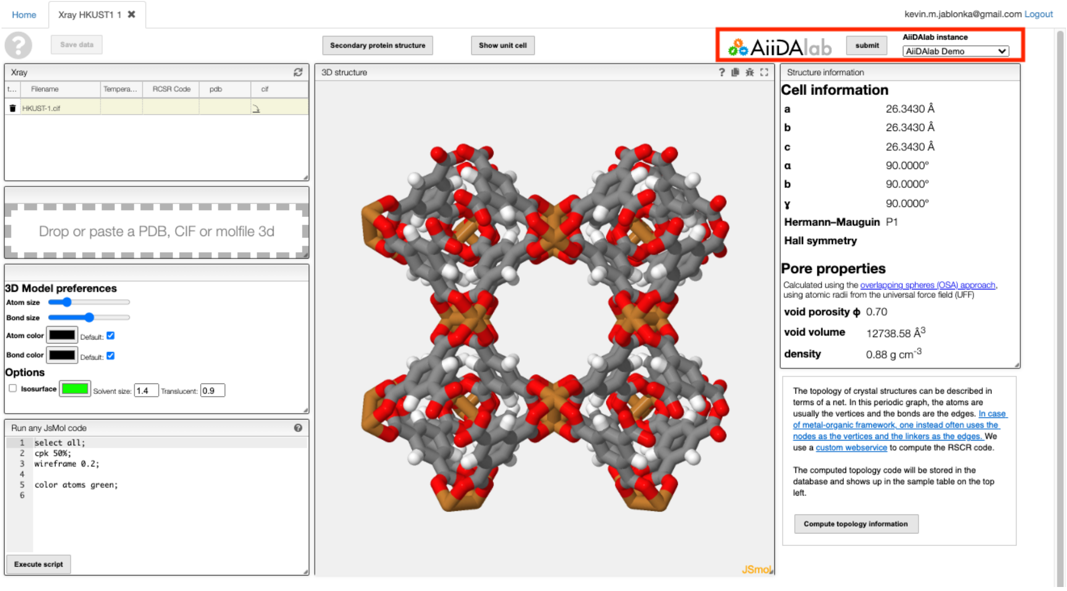
Figure 1. Diagram of the ELN-AiiDAlab interface. AiiDAlab can communicate with the ELN via the ELN REST API to retrieve, for example, a crystal structure. In the AiiDAlab frontend, users can then select among different simulation workflows whose results can, via the REST API of the ELN, be finally uploaded back and visualised in the ELN.

of the structure to their personal AiiDAlab instance using a URL redirect (Figure 2). This request will contain the sample's universally unique identifier (UUID), the database uniform resource identifier (URI), the username, and the structure name as URL query parameters (see Supplementary Figure 6). With this information, AiiDAlab can then employ the REST API of the cheminfo ELN to request the structure file (Figure 1). This structure file can subsequently be used as an input for simulations or other calculations. Once these are completed, the results can be submitted back to the ELN, with reference to the initial structure from which they were launched.

Authentication

A key requirement for our implementation is that the integration is both seamless and secure. For this reason, we use the API token mechanism implemented by the `rest-on-couch` package[12]. For this, AiiDAlab first checks if it already owns a suitable token to connect to the ELN and otherwise prompts the user to set up the connection, if that is not the case. To this aim, the user is asked to provide the ELN instance's address and type (cheminfo in our case) and the access token, which can be obtained via the "Request token" button within the AiiDAlab interface. A click on the button will open a page from the cheminfo ELN in an `iframe`. Before printing the token, the ELN first checks if the user is already authenticated and, if not, it redirects to the authentication page. After the authentication, the user is redirected back to the token-generation page, where the token is shown. This mechanism ensures that the browser contexts remain separated, but we can still use the existing session cookie stored in the browser to authenticate the session in the ELN. The separation of browser contexts also implies that users need to copy and paste the token from the `iframe` into an input field on the same page. At the same time, the reuse of the session cookie implies that users do not have to log in again typing username and password. The token will then be stored in the AiiDAlab file system of the user. We consider the security risk of that as acceptable, since the access to the AiiDAlab file system is already protected by the authentication mechanism of AiiDAlab.

As a note, we also considered passing tokens via URL parameters, but discarded this idea. The URL parameters are secure sockets layer (SSL) encrypted for the transport but might be logged in plain text on the servers and the browser history. We could also not simply provide the token with a RESTful `POST` request from the ELN to the AiiDAlab, as the latter currently implements no REST API, but efforts are ongoing to enable it. We emphasize that, to disable the integration, users can simply delete the token using a custom frontend page provided in the ELN, or delete the token in the ELN setup page of AiiDAlab.



t...	Filename	Tempera...	RCSR Code	pdb	cif
	HKUST-1.cif				

3D Model preferences

Atom size

Bond size

Atom color Default: ☑

Bond color Default: ☑

Options

☐ Isosurface 🟩 Solvent size: 1.4 Translucent: 0.9

```
select all;
cpk 50%;
wireframe 0.2;

color atoms green;

```

AiiDAlab instance: AiiDAlab Demo

Cell information

a	26.3430 Å
b	26.3430 Å
c	26.3430 Å
α	90.0000°
b	90.0000°
γ	90.0000°
Hermann–Mauguin	P1
Hall symmetry	

Pore properties

Calculated using the overlapping spheres (OSA) approach, using atomic radii from the universal force field (UFF)

void porosity ϕ	0.70
void volume	12738.58 Å^3
density	0.88 g cm^{-3}

The topology of crystal structures can be described in terms of a net. In this periodic graph, the atoms are usually the vertices and the bonds are the edges. In case of metal-organic framework, one instead often uses the nodes as the vertices and the linkers as the edges. We use a custom webservice to compute the RSCR code.

The computed topology code will be stored in the database and shows up in the sample table on the top left.

Figure 2. The ELN frontend with the "submit to AiiDAlab button". The cheminfo ELN, with which we implemented this prototype, has different views for different datatypes. In the "crystal structure" view our integration now displays a "submit to AiiDAlab button" (highlighted in the top right) that sends the UUID of the structure, as well as the database URI and user identifiers (encoded as URL parameters) to a specific AiiDAlab URL (the specific instance can be selected with a dropdown menu). The URL points to a custom Jupyter notebook that we developed, that contains code to automatically extract the URL parameters generated by the ELN, initialise the subsequent steps and guide the user through them.

Front-end

ELN

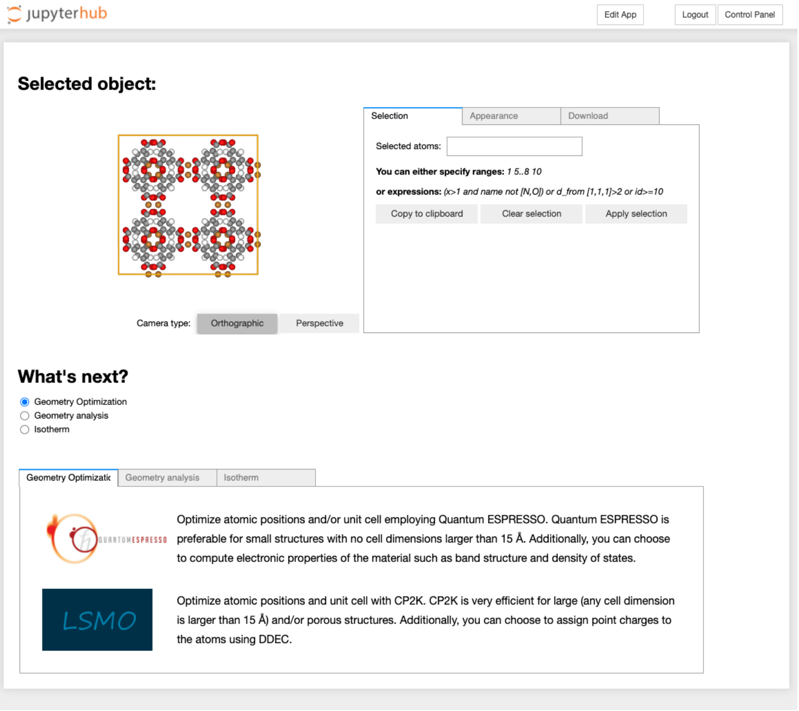
In the ELN, most interfaces are implemented using the visualizer library[13, 14], which comes with powerful plotting tools and type renderers. Computed results, if they are added to the database in a compatible form, will appear in the same way as experimental results — just as if they were created by another analytical instrument. For example, simulated adsorption isotherms can seamlessly be overlapped with experimental ones and processed using the same tools.

AiiDAlab

After clicking the "submit to AiiDAlab button" in the ELN, users will land on a page of the AiiDAlab (see Figure 3) that allows opening the imported structure in an AiiDAlab application. Among others, we provide links to the applications that offer automated simulations via the CP2K[15] and QUANTUM ESPRESSO[16,17] packages that are popular density-functional theory (DFT) codes. Those are used for geometry and cell optimisation, and for the calculation of electronic properties such as band structures. Additionally, AiiDAlab offers applications for force-field based simulations of gas adsorption isotherms and the calculation of geometric properties of nanoporous materials using RASPA[18] and Zeo++[19], respectively.

Pre-simulation sanity checks

Experimental crystal structures, e.g., derived from Rietveld refinements, are typically not computationally ready. Structures often contain atomic overlaps, disorder, floating solvents, or missing hydrogens that could not be resolved in the experiments. A structure file with these characteristics is usually sufficient in an experimental context, but is typically not suitable to serve as direct input for simulations. Since experimental researchers are typically not aware



Selected object:

Selected atoms:

You can either specify ranges: *1 5..8 10*

or expressions: *(x>1 and name not [N,O]) or d_from [1,1,1]>2 or id>=10*

Camera type: Orthographic

What's next?

● Geometry Optimization
○ Geometry analysis
○ Isotherm

Optimize atomic positions and/or unit cell employing Quantum ESPRESSO. Quantum ESPRESSO is preferable for small structures with no cell dimensions larger than 15 Å. Additionally, you can choose to compute electronic properties of the material such as band structure and density of states.

Optimize atomic positions and unit cell with CP2K. CP2K is very efficient for large (any cell dimension is larger than 15 Å) and/or porous structures. Additionally, you can choose to assign point charges to the atoms using DDEC.

Figure 3. The AiiDAlab page on which users land after clicking the "submit to AiiDAlab button" in the ELN. Tabs organise different applications (Geometry/cell optimisation and electronic properties, pore geometry analysis, isotherm calculation) that contain links to different tools (e.g., QUANTUM ESPRESSO and CP2K as DFT engines) using for any given applications.

of such additional constraints, we developed a tool that flags the most commonly encountered issues, focusing on the particular case of metal-organic frameworks, and allows users to directly fix them with automated procedures.

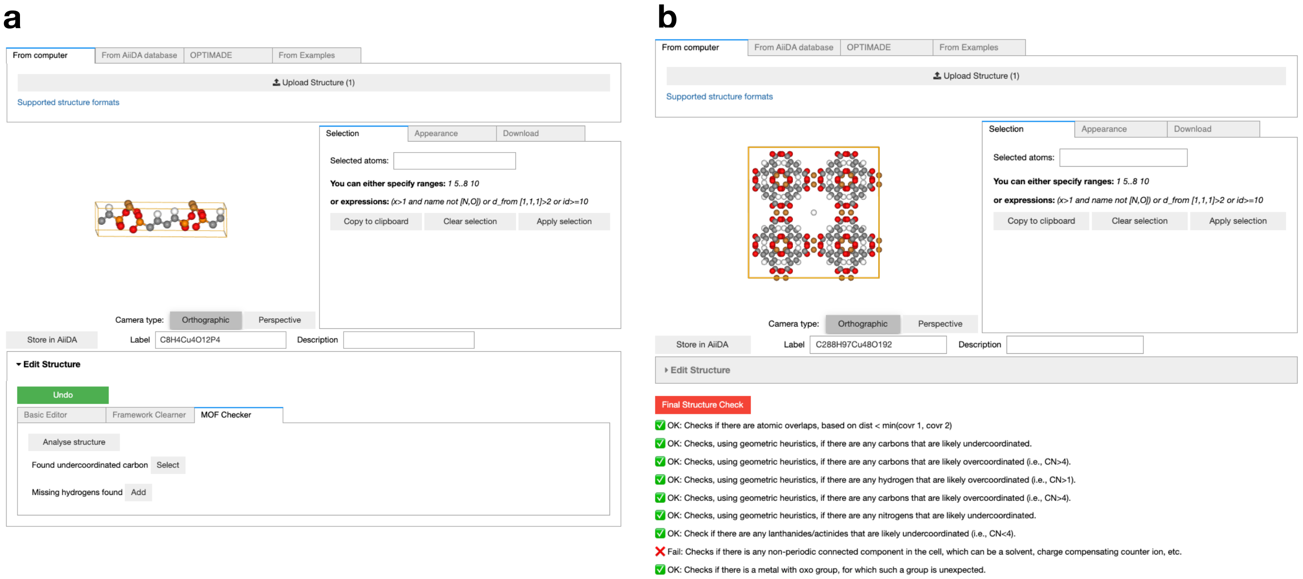
Most of these checks and fixes operate on the structure graph, which we derive using bond heuristics implemented in pymatgen[20,21]. Based on the structure graph, unbound solvent can then be identified as the connected components that do not cross the periodic boundaries. Over or undercoordination can be identified with hard-coded heuristics for common coordination environments. Based on the latter, we can also return coordinates at which hydrogen atoms would be expected. In the frontend, this is implemented at two different levels (see Figure 4). First, a dedicated tab in the structure editor allows to automatically add missing hydrogens and select errors to be fixed (Figure 4**a**). Second, a final check indicates potential issues that should still be fixed (Figure 4**b**).

Provenance

To capture the full provenance of the interaction between the computational and lab experiments, we store additional parameters in both the AiiDA and the ELN database. The AiiDA database is used by the AiiDA workflow engine to orchestrate the workflows, to which AiiDAlab provides the frontend. To this aim, we extended the ELN database schema to include the source type (simulation, experiment, literature), the database URI, and the UUID of the data node from the AiiDA database. Correspondingly, in the AiiDA database, we store the ELN database URI, the type of the ELN, the sample UUID, the attachment filename, and the data type. This information is stored in the so-called "extras" of a node representing the structure in the AiiDA database (of type `CifData` or `StructureData`).

Standardisation

In our opinion, the key difficulty for the integration of different services is that there are still no generally accepted standard schemas for the storage and transfer of scientific data. Most ELNs do not impose a data schema or offer a



Selected atoms:

You can either specify ranges: *1 5..8 10*

or expressions: *(x>1 and name not [N,O]) or d_from [1,1,1]>2 or id>=10*

Label: C8H4Cu4O12P4 Description:

Found undercoordinated carbon

Missing hydrogens found

Selected atoms:

You can either specify ranges: *1 5..8 10*

or expressions: *(x>1 and name not [N,O]) or d_from [1,1,1]>2 or id>=10*

Label: C288H97Cu48O192 Description:

✅ OK: Checks if there are atomic overlaps, based on dist < min(covr 1, covr 2)

✅ OK: Checks, using geometric heuristics, if there are any carbons that are likely undercoordinated.

✅ OK: Checks, using geometric heuristics, if there are any carbons that are likely overcoordinated (i.e., CN>4).

✅ OK: Checks, using geometric heuristics, if there are any hydrogen that are likely overcoordinated (i.e., CN>1).

✅ OK: Checks, using geometric heuristics, if there are any carbons that are likely overcoordinated (i.e., CN>4).

✅ OK: Checks, using geometric heuristics, if there are any nitrogens that are likely undercoordinated.

✅ OK: Check if there are any lanthanides/actinides that are likely undercoordinated (i.e., CN<4).

❌ Fail: Checks if there is any non-periodic connected component in the cell, which can be a solvent, charge compensating counter ion, etc.

✅ OK: Checks if there is a metal with oxo group, for which such a group is unexpected.

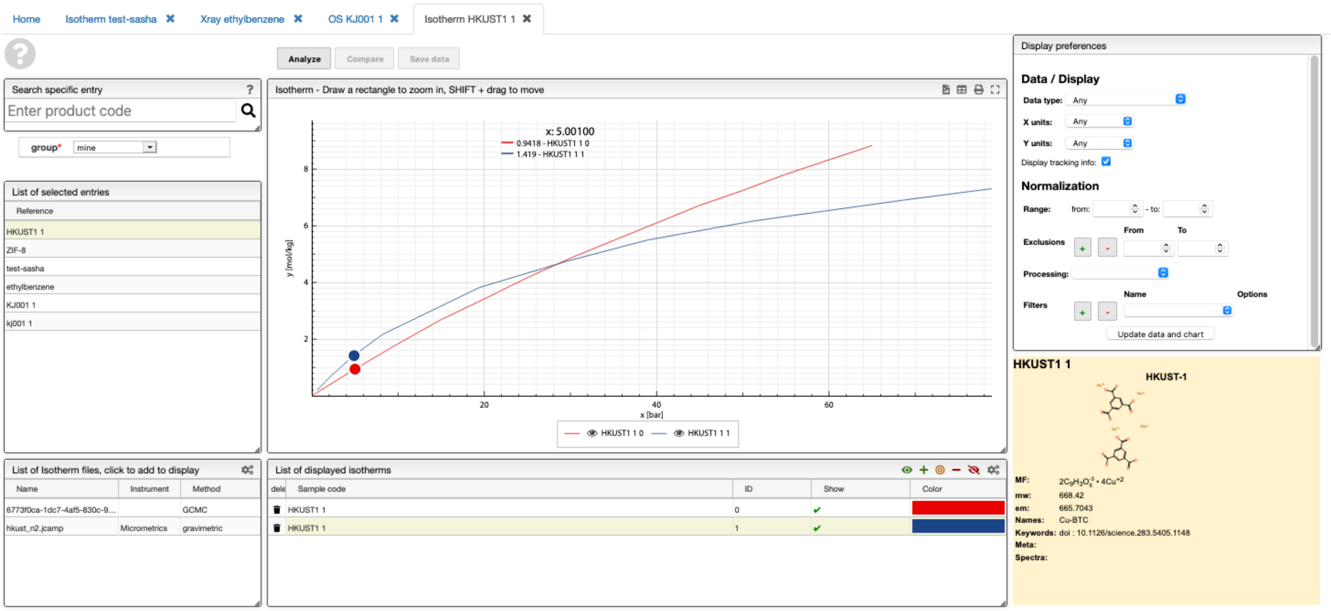
Figure 4. Automated checks help to ensure that structures are ready for simulations. a Many issues can be directly fixed from the structure editor. For example, hydrogen atoms can be automatically added for common coordination geometries. **b** Coloured checkmarks and crosses indicate the checks performed and potential issues. In this particular case, the checks detected a floating atom in the centre of the pore (which can then automatically be removed).

standardised API to access specific characterisation data. Furthermore, those that do define a schema currently use different schemas that are not directly interoperable. This means that, in practice, one needs to manually create mappings between the data schemas of the different platforms that need to interoperate (e.g., here, the AiiDAlab platform and the ELNs). To simplify this process with the cheminfo ELN, we have been developing a Python package (`cheminfopy`) that provides a Python interface for the same abstractions (sample, reaction, and attachments) that are used in the database schema of the cheminfo ELN (see Supplementary Note 3). Using this interface, attributes of samples (e.g., boiling point, molecular mass) in the ELN can be accessed as simple Python object attributes. Since AiiDAlab already implements parsers for the simulation outputs, the only missing step we implemented was to map the AiiDA data objects representing simulation outputs to JCAMP-DX files, the preferred format for spectra in the cheminfo ELN. The conversion has been implemented in the `aiidalab-eln` package. The package is meant to provide a generic interface to connect different ELNs to AiiDAlab (see Supplementary Note 1).

Examples

Geometry optimisation and gas adsorption isotherms

As a concrete example of our integration protocol, we compare the powder X-ray diffractogram (PXRD) of a synthesised metal-organic framework (MOF) with the one obtained from a DFT-optimised structure. Specialised workflows have been developed to optimise the geometry of a MOF using the crystal structure as the starting point[2]. These workflows involve a combination of different optimisation strategies and typically take several hours to converge. Once we have such a DFT-optimised structure, there are several calculations we can do. For example, we can predict the PXRD pattern (see Supplementary Note 5). Large deviations with respect to the experimental PXRD are often a strong indication that the proposed structural model is not stable and might not be the correct one. Importantly, this also clearly indicates that this structural model is not relevant for subsequent simulations of other



group* mine

List of selected entries

Reference
HKUST1 1
ZIF-8
test-sasha
ethylbenzene
KJ001 1
kj001 1

List of Isotherm files, click to add to display

Name	Instrument	Method
6773f0ca-1dc7-4af5-830c-9...		GCMC
hkust_n2.jcamp	Micrometrics	gravimetric

List of displayed isotherms

dele	Sample code	ID	Show	Color
	HKUST1 1	0	✓	🟥
	HKUST1 1	1	✓	🟦

Data type: Any

X units: Any

Y units: Any

Display tracking info: ☑

Range: from: - to:

HKUST1 1

MF: $2C_9H_3O_6^{-3} \cdot 4Cu^{+2}$

mw: 668.42

em: 665.7043

Names: Cu-BTC

Keywords: doi : 10.1126/science.283.5405.1148

Meta:

Spectra:

Figure 5. Overlay of an experimental and a simulated isotherm. In the ELN, the simulated isotherm can be visualized and processed in the same way as an experimental one. The only difference between the simulated the experimental isotherm being the metadata that, in the case of the simulated isotherm, contains the UUID of the output in the AiiDAlab database. Adsorption isotherms that vastly differ in saturation loading can indicate an unsuccessful activation of the material.

spectra. A geometry optimisation might also help in the refinement process, for instance, by providing information about the location of the hydrogen atoms.[22]

Once we have such a DFT optimised crystal structure, AiiDAlab can be used to predict many experimental properties. For example, an important application of MOFs is gas storage and separation[23]. A key observable for the performance in these applications is the gas adsorption isotherm. One also needs to consider the partial charges in the gas adsorption simulations for polar molecules[24]. Since for "routine" simulations one does not want to worry about how charges are computed, the AiiDAlab application automatically derives the partial charges after the geometry optimisation if an isotherm simulation is requested by the user. After the simulation, one can overlay the simulated isotherm with the experimental data in the ELN, as shown in Figure 5. Deviations between the experiment are often due to the inclusion of guest molecules or reduced crystallinity. For this reason, it is important to compare the experimental results with the simulations. Unfortunately, this comparison does not happen routinely at the moment, even though the gas adsorption simulation can be considered as "routine" simulations. We anticipate that the direct integration of atomistic and molecular simulations into an ELN will make such simulations routine.

Future work

To improve the interoperability of the cheminfo ELN with other services and tools we are currently migrating the data schema of the cheminfo ELN to JSON-schema (via TypeScript types, github.com/cheminfo/cheminfo-types), which would allow us to programmatically validate requests to the ELN. In addition, we are also implementing a REST API for AiiDAlab which would also enable ELNs, and other services, to send data to and request data from the AiiDAlab. Clearly, our work highlights the need for a collective effort from the community of ELN developers for the development of a standardised API specification, comparable to OPTIMADE[25], for standardised access to data across simulation and experimental databases.

Conclusions

In this work we have developed a prototype for how "routine" simulations can be made routine, i.e., how experimental data can be integrated with an advanced computational infrastructure. The vision is that of an open science infrastructure integrating an electronic laboratory notebook with a computational platform to seamlessly send a sample for a computational analysis, and then to collect the results in such a way that the full provenance of the calculation, and the correct linkage to the experiment, is guaranteed.

We demonstrated this by integrating the cheminfo ELN and the AiiDAlab platform, but one can envision that such a protocol can be extended to other computational platforms and other types of electronic notebooks. The main advantage of developing a communication protocol rather than fully integrating the AiiDAlab and cheminfo ELN is that such a direct integration would make the infrastructure harder to maintain and, more importantly, imposes unnecessary constraints. For example, AiiDAlab is designed to carry out a limited number of time-consuming simulations remotely on a server using a Jupyter notebook infrastructure, while the ELN is built to run almost completely on the client side, as it is designed for many users carrying out relatively simple operations (comparing data, plotting data, etc.).

Making data infrastructure interoperable usually imposes constraints on the format and the way data are exchanged. However, if M different electronic lab notebooks and N different computational platforms each define and implement their own protocol, one ends up with the unsustainable situation where there are $N \times M$ combined protocols to be developed and maintained. Therefore, we hope that one of the outcomes of this work is that some consensus can be obtained on the future development of such a single standard intercommunication protocol. In our case, we focused on electronic structure calculations and molecular simulations, but the prototype allows for the extension to other types of calculations.

Code availability

The ELN we used for this work is developed in the cheminfo GitHub organisation (`github.com/cheminfo`), for which documentation can be found at `docs.c6h6.org`. The Python interface for the REST API of the ELN is developed on GitHub (`github.com/cheminfo-py/cheminfopy`). The code for the specific implementation on the AiiDAlab is developed on GitHub (`github.com/aiidalab/aiidalab-eln`, `github.com/lsmo-epfl/aiidalab-epfl-lsmo`, `github.com/aiidalab/aiidalab-widgets-base`). The `mofchecker` tool is developed on GitHub (`github.com/kjappelbaum/mofchecker`). Readers can test the integration from the public deployment of the cheminfo ELN at `c6h6.org` and the public AiiDAlab demonstrator available at `aiidalab-demo.materialscloud.org/hub/` using, for example, their Google accounts as authentication technique. All code is licensed under the MIT license.

References

1. Huber, S. P. *et al.* Common workflows for computing material properties using different quantum engines. *arXiv* (2021). 2105.05063.
2. Ongari, D., Yakutovich, A. V., Talirz, L. & Smit, B. Building a Consistent and Reproducible Database for Adsorption Evaluation in Covalent–Organic Frameworks. *ACS Cent. Sci.* **5**, 1663–1675, 10.1021/acscentsci.9b00619 (2019).
3. Gjerding, M. *et al.* Atomic Simulation Recipes – a Python framework and library for automated workflows. *arXiv* (2021). 2104.13431.
4. Jablonka, K. M., Ongari, D., Moosavi, S. M. & Smit, B. Using collective knowledge to assign oxidation states of metal cations in metal–organic frameworks. *Nat. Chem.* 10.1038/s41557-021-00717-y (2021).

5. Patiny, L. *et al.* The c6h6 NMR repository: An integral solution to control the flow of your data from the magnet to the public. *Magn. Reson. Chem.* **56**, 520–528, 10.1002/mrc.4669 (2017).

6. Yakutovich, A. V. *et al.* AiiDAlab – an ecosystem for developing, executing, and sharing scientific workflows. *Comput. Mater. Sci.* **188**, 110165, 10.1016/j.commatsci.2020.110165 (2021).

7. McDonald, R. S. & Wilks, P. A. JCAMP-DX: A standard form for exchange of infrared spectra in computer readable form. *Appl. Spectrosc.* **42**, 151–162, 10.1366/0003702884428734 (1988).

8. Kluyver, T. *et al.* Jupyter notebooks – a publishing format for reproducible computational workflows. In Loizides, F. & Schmidt, B. (eds.) *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, 87 – 90 (IOS Press, 2016).

9. Pizzi, G., Cepellotti, A., Sabatini, R., Marzari, N. & Kozinsky, B. AiiDA: automated interactive infrastructure and database for computational science. *Comput. Mater. Sci.* **111**, 218–230, 10.1016/j.commatsci.2015.09.013 (2016).

10. Huber, S. P. *et al.* AiiDA 1.0, a scalable computational infrastructure for automated reproducible workflows and data provenance. *Sci. Data* **7**, 10.1038/s41597-020-00638-4 (2020).

11. Uhrin, M., Huber, S. P., Yu, J., Marzari, N. & Pizzi, G. Workflows in AiiDA: Engineering a high-throughput, event-based engine for robust and modular computational workflows. *Comput. Mater. Sci.* **187**, 110086, 10.1016/j.commatsci.2020.110086 (2021).

12. Zasso, M. rest-on-couch. https://github.com/cheminfo/rest-on-couch (2021).

13. Pellet, N. jsGraph and jsNMR—advanced scientific charting. *Challenges* **5**, 294–295, 10.3390/challe5020294 (2014).

14. Pellet, N. visualizer. https://github.com/NPellet/visualizer (2020).

15. Kühne, T. D. *et al.* CP2k: An electronic structure and molecular dynamics software package - quickstep: Efficient and accurate electronic structure calculations. *The J. Chem. Phys.* **152**, 194103, 10.1063/5.0007045 (2020).

16. Giannozzi, P. *et al.* QUANTUM ESPRESSO: a modular and open-source software project for quantum simulations of materials. *J. Physics: Condens. Matter* **21**, 395502, 10.1088/0953-8984/21/39/395502 (2009).

17. Giannozzi, P. *et al.* Advanced capabilities for materials modelling with quantum ESPRESSO. *J. Physics: Condens. Matter* **29**, 465901, 10.1088/1361-648x/aa8f79 (2017).

18. Dubbeldam, D., Calero, S., Ellis, D. E. & Snurr, R. Q. RASPA: molecular simulation software for adsorption and diffusion in flexible nanoporous materials. *Mol. Simul.* **42**, 81–101, 10.1080/08927022.2015.1010082 (2015).

19. Willems, T. F., Rycroft, C. H., Kazi, M., Meza, J. C. & Haranczyk, M. Algorithms and tools for high-throughput geometry-based analysis of crystalline porous materials. *Microporous Mesoporous Mater.* **149**, 134–141, 10.1016/j.micromeso.2011.08.020 (2012).

20. Ong, S. P. *et al.* Python Materials Genomics (pymatgen): A robust, open-source python library for materials analysis. *Comput. Mater. Sci.* **68**, 314–319, 10.1016/j.commatsci.2012.10.028 (2013).

21. Pan, H. *et al.* Benchmarking Coordination Number Prediction Algorithms on Inorganic Crystal Structures. *Inorg. Chem.* **60**, 1590–1603, 10.1021/acs.inorgchem.0c02996 (2021).

22. Altomare, A., Ciriaco, F., Cuocci, C., Falcicchio, A. & Fanelli, F. Combined powder X-ray diffraction data and quantum-chemical calculations in EXPO2014. *Powder Diffr.* **32**, S123–S128, 10.1017/s088571561700015x (2017).

23. Zhou, H.-C., Long, J. R. & Yaghi, O. M. Introduction to metal–organic frameworks. *Chem. Rev.* **112**, 673–674, 10.1021/cr300014x (2012).

24. Ongari, D. *et al.* Evaluating charge equilibration methods to generate electrostatic fields in nanoporous materials. *J. Chem. Theory Comput.* **15**, 382–401, 10.1021/acs.jctc.8b00669 (2018).

25. Andersen, C. W. *et al.* OPTIMADE: an API for exchanging materials data. *arXiv* (2021). 2103.02068.

Acknowledgments

K.M.J. thanks Andrew Rosen and Leopold Talirz for bug reports and feedback on the `mofchecker` library. We also thank our experimental colleagues for feedback on the implementation. Moreover, we thank Carl S. Adorf for feedback and support on the implementation and the manuscript. This work was supported by a European Research Council (ERC) Advanced Grant (Grant Agreement No. 666983, MaGic), the Swiss National Science Foundation (SNSF) under Grant 200021_172759 and the National Centre of Competence in Research (NCCR) MARVEL (Grant 51NF40-182892).

Author contributions statement

K.M.J. and A.V.Y. conceived and implemented the integration. L.P. provided guidance for the integration in the ELN frontend. M.Z. is maintainer of the `rest-on-couch` library. B.S. supervised and coordinated the project. N.M. and G.P. contributed to supervision, review and editing of the manuscript. All authors contributed to writing of the manuscript.

Competing interests

The authors declare no competing interests. M.Z. is co-founder and L.P. is chief scientific officer of Zakodium Sàrl, a company dedicated to the development of tools for storing, processing, and analysis of scientific information.