Digital Discovery

Accepted Manuscript

This article can be cited before page numbers have been issued, to do this please use: K. Eimre, M. Evans, B. Macaulay, X. Wang, J. Yu, N. Marzari, G. Rignanese and G. Pizzi, *Digital Discovery*, 2026, DOI: 10.1039/D6DD00125D.

ROYAL SOCIETY OF CHEMISTRY

rsc.li/digitaldiscovery

View Article Online
DOI: 10.1039/D6DD00125D

optimade-maker: Automated generation of interoperable materials APIs from static datasets

Kristjan Eimre [1, 2, *], Matthew L. Evans [3, 4, 5, *, †, ‡], Bud Macaulay [1], Xing Wang [1, 2], Jusong Yu[1, 2], Nicola Marzari [1, 2, 6], Gian-Marco Rignanese [3], and Giovanni Pizzi [1, 2, §]

[1]PSI Center for Scientific Computing, Theory and Data, 5232 Villigen PSI, Switzerland
[2]National Centre for Computational Design and Discovery of Novel Materials (MARVEL), 5232 Villigen PSI, Switzerland
[3]UCLouvain, Institute of Condensed Matter and Nanosciences, Chemin des Ètoiles 8, Louvain-la-Neuve 1348, Belgium
[4]Matgenix SRL, A6K Advanced Engineering Center, Charleroi, Belgium
[5]datalab industries ltd., King's Lynn, United Kingdom
[6]Theory and Simulation of Materials (THEOS), École Polytechnique Fédérale de Lausanne, 1015 Lausanne, Switzerland
*These authors contributed equally.
‡Present address: Yusuf Hamied Department of Chemistry, University of Cambridge, Cambridge, CB2 1EW, United Kingdom
†me388@cam.ac.uk
§giovanni.pizzi@psi.ch

(April 24, 2026)

Abstract

Atomistic structural data are central to materials science, condensed matter physics, and chemistry, and are increasingly digitised across diverse repositories and databases. Interoperable access to these heterogeneous data sources enables reusable clients and tools, and is essential for cross-database analyses and data-driven materials discovery. Toward this aim, the OPTIMADE (Open Databases Integration for Materials Design) specification defines a standard REST API for atomistic structures and related properties. However, deploying and maintaining compliant services remains technically demanding and poses a significant barrier for many data providers. Here, we present `optimade-maker`, a lightweight toolkit for the automated generation of OPTIMADE-compliant APIs directly from raw atomistic structure and property data. The toolkit supports a wide range of raw datasets, enables conversion to a standardised OPTIMADE data representation, and allows for rapid deployment of APIs in both local and production environments. We further demonstrate it through an automated service on the Materials Cloud Archive, which automatically creates and publishes OPTIMADE APIs for contributed datasets, enabling immediate discoverability and interoperability. In addition, we implement data transformation pipelines for the Cambridge Structural Database (CSD) and the Inorganic Crystal Structure Database (ICSD), enabling unified access to these curated resources through the OPTIMADE framework. By lowering the technical barriers to interoperable data publication, `optimade-maker` represents an important step toward a scalable, FAIR materials data ecosystem integrating both community-contributed and curated databases.

1 Introduction

Atomistic structural data of crystalline and molecular systems underpin the most fundamental aspects of materials science, condensed matter physics and chemistry. In the digital era, such data, together with associated properties and derived quantities, are increasingly collected in structured databases and made accessible to the research community through application programming interfaces (APIs) and graphical user interfaces (GUIs) or, most often, as static datasets described by files. These databases may contain experimentally determined structures, such as those in the Crystallography Open Database (COD) [1], the Inorganic Crystal Structure Database (ICSD) [2], the Cambridge Structural Database (CSD) [3], or the Materials Platform for Data Science (MPDS), as well as structures generated through high-throughput computational workflows [4–14].

The primary APIs of atomistic structure databases, where present, are often custom-built and lack any interoperability between different providers. Consequently, client applications aiming to access multiple databases must implement support for a variety of incompatible API standards, significantly increasing complexity for the development of clients and of data-driven pipelines that explore the data. To address this issue, the Open Databases Integration for Materials Design (OPTIMADE) consortium, comprising representatives from many major materials databases worldwide, has developed a common specification for a REST (representational state transfer) API [15] to serve atomistic structure data and related properties. The OPTIMADE specification is designed to accommodate the diverse requirements and constraints of materials databases, enabling uniform access to atomistic structure data across

View Article Online
DOI: 10.1039/D6DD00125D

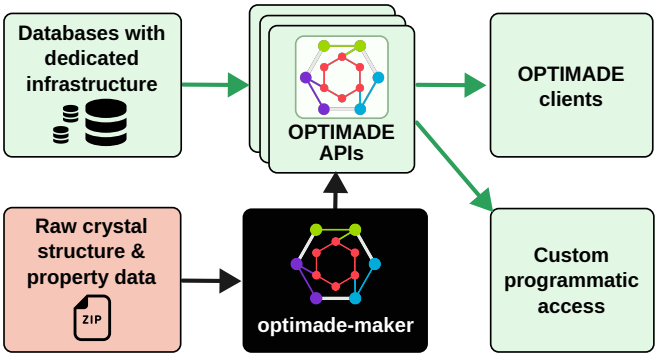

Figure 1: Schematic illustrating the context of the `optimade-maker` toolkit within the OPTIMADE ecosystem. Green boxes indicate already established entities. The red box highlights raw materials data that are not readily integrable into the ecosystem, a gap addressed by `optimade-maker`.

providers. In addition, the specification defines a standard mechanism for dataset discoverability on the web, allowing compliant APIs to align with the FAIR (Findable,Accessible, Interoperable, and Reusable) data principles [16].

As a result, an OPTIMADE ecosystem has emerged. As of March 2026, 20 database providers expose OPTIMADE-compliant APIs, collectively indexing over 25 million materials [17]. A growing number of clients and applications leverage the specification to discover, analyse, and aggregate materials data across multiple sources. OPTIMADE APIs have already been successfully employed in several materials discovery and design projects [18–20].

Despite these advances, deploying and maintaining a materials database together with a fully compliant OPTIMADE API typically requires dedicated infrastructure consisting of hardware, software, and personnel, which is costly and time-consuming to build and maintain. The associated technical overhead and maintenance effort present a substantial barrier for individual researchers or small research groups who wish to disseminate their data in an interoperable manner.

In this work, we present `optimade-maker`, a toolkit that addresses this challenge by enabling the automated generation of OPTIMADE APIs directly from raw materials data files, such as simulation outputs or structural assignments. Built on top of the existing `optimade-python-tools` [21] Python library, `optimade-maker` can be integrated into data pipelines to provide OPTIMADE-compliant services for production web platforms, while also allowing researchers to quickly deploy a local API for using OPTIMADE-compliant clients with their raw data. Figure 1 illustrates the position of `optimade-maker` within the OPTIMADE ecosystem.

We further present representative use cases of `optimade-maker` developed as part of this work. One such use case is the Materials Cloud Archive OPTIMADE service, which enables researchers to automatically obtain an OPTIMADE API for their datasets upon upload to the Materials Cloud Archive [13], an open research data repository, thereby facilitating immediate interoperability and discoverability. The Materials Cloud Archive links these datasets directly to the newly redesigned Materials Cloud OPTIMADE Client, where they can be interactively browsed. This service has already been used to serve several contributed datasets. Furthermore, we used `optimade-maker` to implement the OPTIMADE data transformation pipelines for the Cambridge Structural Database (CSD) and the Inorganic Crystal Structure Database (ICSD).

2 Results

2.1 `optimade-maker` toolkit

The `optimade-maker` toolkit is developed as a Python package which enables the automated creation of OPTIMADE APIs from a range of structural data formats and associated material properties. The toolkit can be used as a Python library or via the `optimake` command line interface (CLI) tool. The primary features provided by the toolkit include: 1) specification of a simple YAML (YAML Ain't Markup Language) configuration file that describes the raw data and makes it parsable for the toolkit; 2) conversion of the raw data into the standard OPTIMADE JSON Lines file format, based on JavaScript Object Notation (JSON), which was developed as part of this work and is now part of the official specification since v1.3.0; 3) serving an OPTIMADE API directly from a raw data archive or from an OPTIMADE JSON Lines file using the reference server implementation from `optimade-python-tools` [21].

In a typical `optimade-maker` workflow, the user provides a collection of raw files describing atomistic structures, possibly in an archive file (e.g., a ZIP file), and optionally, any associated properties for these. `optimade-maker` assigns each structure a unique identifier based on its path and file name (see section 3.2), and the property files can reference either the identifier or the full path. To make this data parsable by `optimade-maker`, an `optimade.yaml` configuration file is provided, describing the locations of files, and defining the relevant property metadata. After this setup, the CLI can be used to convert the data into a standardised OPTIMADE representation and start the API. A step-by-step, beginner-friendly tutorial demonstrating this workflow is available in the project repository, providing a guided introduction for new users.

A schematic overview of a concrete use case is shown in fig. 2. In this example, the structures are packaged in a ZIP archive (`structures.zip`) containing multiple Crystallographic Information Files (CIFs) [22, 23], together with a Comma-Separated Values (CSV) file

View Article Online
DOI: 10.1039/D6DD00125D

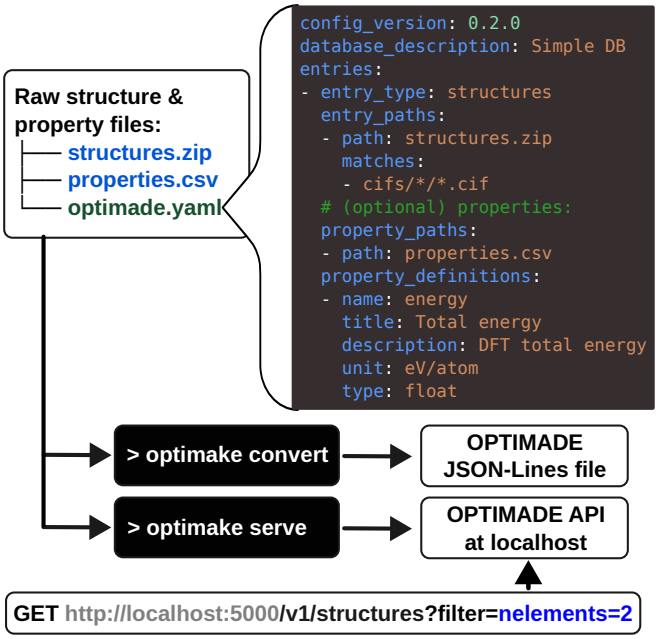

Figure 2: Schematic overview of the main components of the `optimade-maker` CLI. Raw data files are supplemented by the `optimade.yaml` configuration file, describing file locations and property definitions. Black boxes show the two primary CLI commands: `convert` transforms the raw data into the standard JSON Lines format, while `serve` launches an OPTIMADE API server. The server can be queried by any standard OPTIMADE HTTP requests, as shown in the example at the bottom (gray segment denotes the base URL, and blue segment represents a filter selecting binary structures).

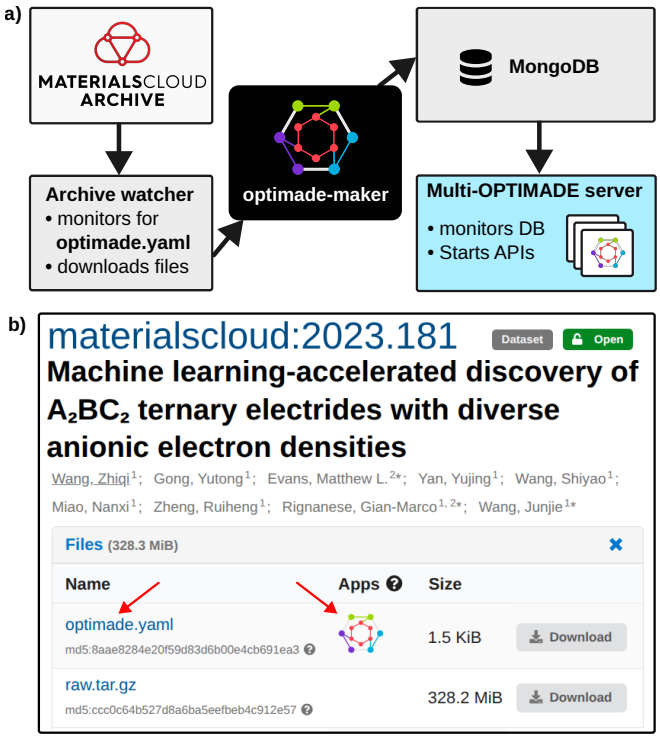

Figure 3: Materials Cloud Archive OPTIMADE service. (a) Data pipeline utilizing the `optimade-maker` toolkit. (b) A webpage for a Materials Cloud Archive entry [29] containing an `optimade.yaml` file, where a link is displayed to directly explore the dataset with the new Materials Cloud OPTIMADE client.

(`properties.csv`) that includes, for each structure, an identifier and a property value – the floating-point total energy per atom. Using this configuration, the `optimake convert` CLI command can convert the raw data into the standard OPTIMADE JSON Lines format, e.g., for archival purposes. The `optimake serve` command directly serves an OPTIMADE API from the raw data. By default, the `serve` command starts the API locally (on `localhost`), where it is immediately available for OPTIMADE-compliant queries by client applications allowing for search across any of the standardised OPTIMADE fields and any extra properties defined in the YAML configuration file.

`optimade-maker` supports raw data in a variety of formats. Atomistic structure files are parsed using the Atomic Simulation Environment (ASE) [24], enabling support for most standard and non-standard formats, including CIF, XYZ, and XSF. Pymatgen [25] JSON files containing structures and related properties are also supported. These files may be compressed or archived using common formats such as `.zip`, `.tar.gz`, and `.tar.bz2`. For property data, CSV and JSON files are supported (see also section 3.2 on how to map properties to structures).

In addition, `optimade-maker` can ingest data from AiiDA [26–28] archive files or directly from an AiiDA profile, as described in detail in section 3.3.

The `optimake serve` command is designed to support both rapid local deployment and more complex data pipeline setups, including production-grade APIs. By default, the command converts the raw data and populates a temporary in-memory MongoDB database using the `MongoMock` Python library, eliminating the need for an external database and enabling immediate access to the data through an OPTIMADE API.

Alternatively, `optimake serve` can be configured using a custom `optimade-python-tools` configuration file. This allows the population of an external MongoDB instance (e.g., for production deployments), customisation of provider metadata, or execution of the data population pipeline without starting the API itself. The latter mode is particularly useful in automated workflows where the API is launched through separate orchestration mechanisms, e.g., as discussed in the next section.

2.2 Materials Cloud Archive automatic OPTIMADE service

The Materials Cloud Archive [13] is an open research data repository for computational materials science. Researchers can upload datasets without restrictions on

View Article Online
DOI: 10.1039/D6DD00125D

file format and make them available to the community for direct download.

Using `optimade-maker`, we developed an automated service on the Materials Cloud platform that deploys OPTIMADE APIs for archive entries compatible with the toolkit, as illustrated in fig. 3. Figure 3a shows a schematic overview of the data pipeline underlying this service. A Python job, referred to as the *Archive watcher*, regularly monitors newly published entries in the Materials Cloud Archive, and checks their compatibility with `optimade-maker` based on the existence of an `optimade.yaml` file in the appropriate format. Compatible entries are processed using the `optimake serve --prepare_only` command, which converts the raw data into the required internal format and generates a corresponding `optimade-python-tools` configuration. The processed data are then injected into a MongoDB instance, which is monitored by the *Multi-OPTIMADE server*, a light wrapper around `optimade-python-tools` that allows the server to efficiently manage multiple OPTIMADE APIs. Upon detecting new data, the *Multi-OPTIMADE server* launches the corresponding APIs and mounts them under distinct subpaths of a single Python REST API.

The resulting OPTIMADE API endpoints are published to the wider OPTIMADE ecosystem under the Materials Cloud Archive provider identifier `mcloudarchive`, and can be accessed by any OPTIMADE-compatible client. Each OPTIMADE API gets its own database identifier based on the Materials Cloud Archive DOI that represents all versions of the entry. If a new version of an Archive entry is published, a new OPTIMADE API is set up with the same identifier that replaces the old version of the API.

The Materials Cloud Archive webpage containing the OPTIMADE configuration (fig. 3b) will contain a direct link to the Materials Cloud OPTIMADE Client, fully redesigned as part of this work (see details in section 3.4). Finally, the created OPTIMADE API endpoint with all relevant metadata and links is also published to the Materials Cloud OPTIMADE overview page at `https://optimade.materialscloud.org`.

2.3 Providing API access to materials design datasets

The Materials Cloud Archive integration described above has been used to disseminate structures and properties for several materials discovery and design projects [18]. For instance, in Ref. [29], Wang *et al.* performed high-throughput first principles calculations and trained machine learning (ML) models to screen $P4/mbm$-A_2BC_2 structural prototypes to design new ternary electride materials. Starting from a library of 214 known A_2BC_2 phases, density-functional theory calculations were performed to assess their electride nature (via the maximum value of the electron localisation function, ELF_{max}) and to create a training set for a series of ML models. The $P4/mbm$-A_2BC_2 prototype structure was then decorated with different elements to form a design space of over 14,000 hypothetical compounds and the trained ML models were used to rapidly predict their ELF_{max} values and thermodynamic stability, identifying high priority materials to investigate further with DFT calculations. Through this approach, 41 stable and 104 metastable A_2BC_2 potential electrides were predicted, with diverse anionic electron densities across the range of electron-deficient, neutral and electron-rich electrides. The three most promising materials were experimentally validated for synthesisability and catalytic activity for ammonia synthesis. The raw data from this work were deposited on the Materials Cloud Archive with a supplementary `optimade.yaml` configuration file that indicated which aspects of the dataset to index as an OPTIMADE API [30]; the authors chose to index the 145 compounds that were computed with DFT and their associated stabilities (formation energies, hull distances versus known phases) and their ELF_{max} values, exposing them for further search.

In two other related projects, Trinquet *et al.* performed a combined ML and DFT active learning screening for materials that exhibit strong optical responses: high refractive indices [19] and strong second-harmonic generation for a given band gap [20]. The initial screening dataset was defined by an OPTIMADE query to curate all hypothetically stable structures that were non-centrosymmetric across contributing databases. Then, seeding the active learning with DFT calculations on the subset of known high performing materials and a random exploratory selection of unknown materials, models were trained to predict optical response. These models were then used to prioritise the next structures on which to perform the high-fidelity DFT calculations, with the overall search campaign comprising several such repeating loops. The resulting datasets from each study, comprising structures, their corresponding DFT-computed refractive indices [31] and their second-harmonic generation coefficients [32], were collated and uploaded to Materials Cloud Archive. Both datasets provide annotations describing the additional computed fields, e.g., `_mcloudarchive_d_kp_conv_neum` with description "The effective Kurtz–Perry powder coefficient from the conventional HSE scissor-corrected SHG tensor" and unit "pm/V", which are reported in the `/info/structures` endpoint of the corresponding OPTIMADE API and are exposed for search.

Each of the three datasets described above return results when OPTIMADE users make cross-provider queries, making use of the OPTIMADE APIs served by Materials Cloud Archive. Custom fields specified by the user are prepended with the `_mcloudarchive` provider prefix and the underlying properties are made queryable via OPTIMADE; for example, the query

View Article Online
DOI: 10.1039/D6DD00125D

"_mcloudarchive_convex_hull_distance < 0.025 AND _mcloudarchive_elf_max > 0.5" will return all low-lying hypothetical structures from Ref. [29].

2.4 Data transformation pipelines for the CSD and ICSD

In addition to providing API access to archived data, the tools developed in `optimade-maker` have been used to provide API access to rolling snapshots of continuously updated databases. Two such live databases are the Cambridge Structural Database (CSD) [3] and the Inorganic Crystal Structure Database (ICSD) [2, 33]. The CSD contains a "complete record of all published organic and metal-organic small molecule crystal-structures" [3], curated by experts and deposited either directly or via journal publication. The ICSD contains "a near exhaustive list of known inorganic crystal structures published since 1913" [2], primarily derived from diffraction data but increasingly including published theoretical structures. As pioneers of data-driven science, both databases have been curated since the 1970s and now contain approximately 1.4 million and 327,000 entries, respectively. However, as commercial databases, programmatic access is limited to license holders and requires bespoke software. OPTIMADE makes no requirement that data conforming to it be open or freely available; this benefits users and database providers as tools written to target commercial (or otherwise closed) datasets that conform to OPTIMADE should also automatically work on open datasets, and vice versa, preventing further fragmentation of the ecosystem.

The UK's Physical Sciences Data Infrastructure (PSDI) [34], following the former Physical Sciences Data Service, provides access to both the CSD and ICSD to UK academics through a combined license. PSDI identified the need for materials API standardisation to enable cross-search of data resources that they collect, aggregate and curate and decided upon OPTIMADE as the enabling technology.

To enable this, data pipelines were developed to map all entries in the CSD and ICSD into the OPTIMADE format, encompassing structural, bibliographic and chemical data pertaining to each entry, via the CSD Python API [35] and the ICSD REST API, respectively. These mapped entries were then written to a combined OPTIMADE JSON Lines file which can be served as an OPTIMADE API using the tools provided by `optimade-maker`.

This approach is significantly simpler than the alternative of mapping database queries and outputs from the existing database-specific formats and returning them in an OPTIMADE compliant way, but comes at the cost of needing to run a secondary database and API that must be periodically updated from the live source. Each database required its own extension fields to the core OPTIMADE structure type; the CSD focusing on chemical identifiers (SMILES, InChI etc.) and molecular crystal properties (Z, Z'), whereas the ICSD made use of several tabulated CIF [23] fields that describe diffraction experiments such as measurement conditions and goodness-of-fit.

Although not the focus of this work, software implementations for these pipelines can be found on GitHub at datalab-industries/csd-optimade and datalab-industries/icsd-optimade. The resulting OPTIMADE APIs are not publicly accessible, as only licence holders are allowed to query and retrieve the underlying data. However, these pipelines are deployed by the PSDI so that UK academic users can receive seamless access to structures from the CSD and ICSD in the PSDI Cross Data Search service [34], alongside many other resources, with unified querying and semantics to access property definitions powered by OPTIMADE.

3 Methods

3.1 Software design

The `optimade-maker` Python package has four components that implement the main functionality: 1) `optimade_maker.config`, 2) `optimade_maker.convert`, 3) `optimade_maker.parsers`, and 4) `optimade_maker.serve`.

`optimade_maker.config` defines the `optimade.yaml` configuration format, making use of Pydantic [36] to provide typed, versioned schema definitions for each field. The configuration can be provided as JSON, YAML, or directly as a Python dictionary. As shown in Figure 2, the configuration consists of a few top-level metadata fields about the database itself, and then sub-configuration objects per entry type that define the files to parse as entries and, optionally, properties. When user-defined properties are provided, they must be accompanied by extra metadata in the OPTIMADE property definition, including the field name, title, OPTIMADE data type (e.g., float or integer), unit, and a human-readable description.

`optimade_maker.convert` implements the pipeline that takes the archived data and applies the scheme defined in the user-supplied `optimade.yaml` configuration file to create a single combined OPTIMADE JSON Lines file for the dataset. The basic process is as follows: first, decompress the archived data (typically provided as a ZIP or tar file), then loop through the OPTIMADE resource types (e.g., structures, references) that have processing rules provided in the configuration file. These rules include a list of patterns that match file paths to attempt to parse as the given entry type, `entry_paths` (e.g., the wildcard `*.cif`), and an optional list of `property_paths` corresponding to auxiliary files that contain property data pertaining to those entries. The properties themselves need to be de-

View Article Online
DOI: 10.1039/D6DD00125D

fined in the `property_definitions` field of the configuration. The conversion process then constructs the appropriate entries for each entry type, decorates them with any provided properties, creates the corresponding `/info/<type>` resource that describes the user-extended entry type in the OPTIMADE format, and finally saves each entry as an individual line in a combined JSON Lines file.

The `optimade_maker.parsers` module is a registry of tools that map files of a given representation (standardised or otherwise) into intermediate objects that can be mapped to OPTIMADE entries. These tools tend to be implemented in other open source libraries, such as ASE [24] or pymatgen [25]. These libraries, combined, provide parsers for many atomistic simulation codes and otherwise standardised structural file formats (e.g., CIF, XYZ), with light wrappers to allow the parsers to fail fast. For structural data, parsers can return either ASE `Atoms` or Pymatgen `Structure` objects, which are then mapped to OPTIMADE structures using the adapters implemented in `optimade-python-tools` [21]. Rather than relying on file extensions, each parser is run in turn on each file until it can be successfully parsed and converted into an OPTIMADE structure. A similar process is followed for parsing property data, although here only CSV and JSON files are supported, which can both be unambiguously read using the `pandas` library or the Python standard library. Extra validation is performed against the user-provided property definitions, ensuring that data types can be appropriately coerced and IDs can be uniquely matched to the created entries (see Section 3.2). These parser components can be easily extended to accommodate new libraries, file formats and entry types.

Finally, `optimade_maker.serve` takes the constructed OPTIMADE JSON Lines file and serves it with an OPTIMADE API using the reference server implementation from `optimade-python-tools` (based on FastAPI and leveraging MongoDB as the database backend), which supports the majority of OPTIMADE 1.3 features and enables search over both standard and user-provided properties. This allows future updates (e.g., new features, performance improvements) to `optimade-python-tools` and the OPTIMADE specification itself to be readily accommodated in `optimade-maker`.

3.2 Structure identifiers

As each structure must have an identifier in OPTIMADE, `optimade-maker` generates, by default, a structure identifier (`id`) based on its path relative to the `optimade.yaml` configuration file. The identifier is constructed using a simple deterministic rule: from the set of all file paths, the longest common prefix and suffix (including file extensions) shared by all paths are removed. For example, given the two structures: `structures.zip/cifs/set1/101.cif` and `structures.zip/cifs/set2/102.cif`, the corresponding identifiers are `set1/101` and `set2/102`.

3.3 AiiDA integration

We also implement the ability for `optimade-maker` to create an OPTIMADE API directly from AiiDA [26–28] databases. The AiiDA workflow management infrastructure allows users to define and execute computational workflows, automatically storing the results and their full provenance in a graph database where each data object and process is represented as a node.

The `optimade-maker` package can access either a live AiiDA database or an exported `.aiida` archive file. The user should specify an AiiDA group containing the subset of the structures to be exposed through the OPTIMADE API. AiiDA UUIDs (universally unique identifiers) are used for the OPTIMADE structure identifiers. Additional properties can be associated with each structure by extracting them either directly from the AiiDA structure node (e.g., from the extras) or from other nodes, according to a query encoded in the property definition.

Figure 4a shows a schematic example of an AiiDA provenance graph representing a density-functional theory (DFT) crystal-structure relaxation followed by an electronic band-structure calculation, using the Quantum ESPRESSO [37, 38] DFT code. The workflow internally performs the two steps, producing data nodes as outputs. The band-structure node does not directly store the band-gap value; instead, this value is computed in an additional postprocessing step, which returns a dictionary (`Dict`) node. To use `optimade-maker` to create an OPTIMADE database and API for the relaxed structures in this AiiDA database, and also associate the calculated band-gap values as properties of each structure, the user can prepare the configuration file shown in fig. 4b. Structures are selected from the AiiDA group `relaxed_structures`. The property definition for the band gap includes an `aiida_query` section, which defines a query for the relevant property node relative to the structure node, using the standard filtering and projection keywords provided by the AiiDA `QueryBuilder`.

Finally, the functionality supporting AiiDA databases is lazily loaded and is not a mandatory dependency, so that `optimade-maker` does not require an AiiDA installation for its basic functionality.

3.4 New Materials Cloud OPTIMADE Client

In addition to enabling researchers to automatically publish their data to the OPTIMADE ecosystem, the Materials Cloud platform provides the OPTIMADE Client, a web application for interactively exploring OPTIMADE-compliant datasets. As part of this work, to

View Article Online
DOI: 10.1039/D6DD00125D

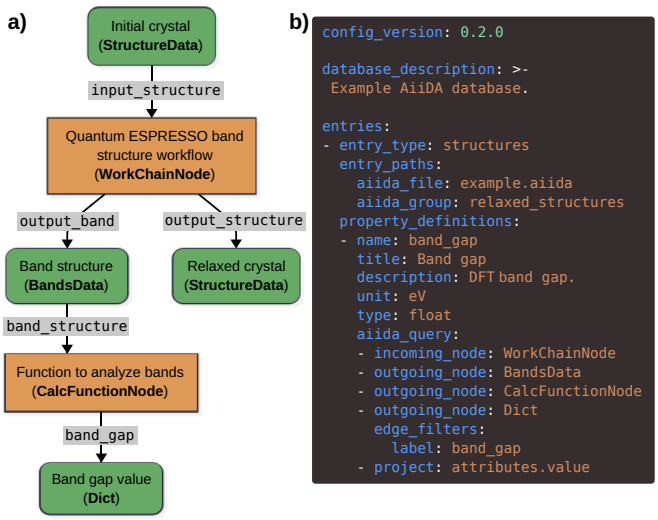

Figure 4: `optimade-maker` integration with AiiDA. (a) A schematic representing an AiiDA provenance graph. Green rounded rectangles and orange rectangles represent data and process nodes, respectively. Each node contains a description, and its AiiDA type (bold). Labels on the arrows represent AiiDA edge labels. (b) The `optimade-maker` configuration file that allows to convert the AiiDA database into the OPTIMADE format, and serve it via the API.

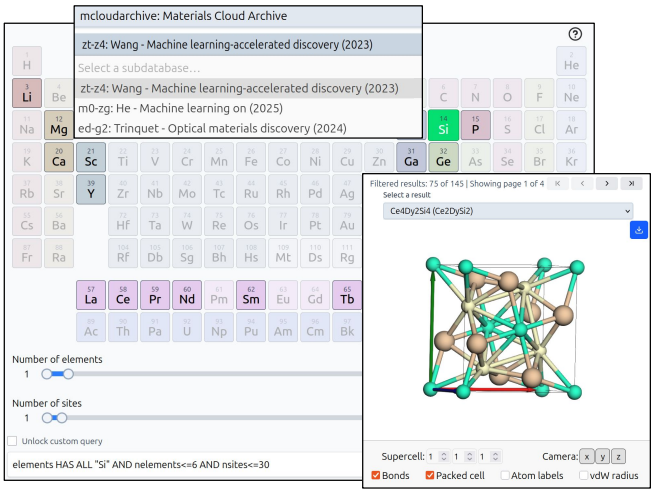

Figure 5: Screenshot of the new Materials Cloud OPTIMADE Client. An OPTIMADE provider and database (here, a contributed dataset from the Materials Cloud Archive) are selected via dropdown menus. Materials are filtered by composition using an interactive periodic table and by structural properties using sliders. The inset shows the results section – a 3D visualisation of a filtered structure.

enhance the automatic Materials Cloud Archive OPTIMADE service presented in section 2.2, the OPTIMADE Client underwent a complete overhaul, rewriting it from Python to JavaScript and resulting in substantial improvements in performance and usability. Figure 5 shows a screenshot of the updated web application. The client automatically displays a list of all public OPTIMADE providers and their databases, including contributed entries from the Materials Cloud Archive, and allows users to select among them. Custom OPTIMADE API URLs are also supported, enabling, for example, the exploration of locally hosted APIs started with `optimade-maker`. To filter materials, the client provides an interactive periodic table for selecting compositions, as well as sliders for constraining structural properties. Any selected filters are reflected in a textbox containing the corresponding raw OPTIMADE query string, which can be reused in other applications or for learning the query language. This textbox can also be edited manually, enabling the construction of more complex OPTIMADE queries. The resulting structures can be browsed and visualised, and any associated properties are displayed. Selected structures can be downloaded in multiple formats or directly imported into the Materials Cloud Quantum ESPRESSO input generator web application. The new OPTIMADE Client is publicly available at `https://optimadeclient.materialscloud.io`.

4 Conclusions and Outlook

In this work, we introduced `optimade-maker`, a toolkit for the automated generation of OPTIMADE-compliant APIs directly from raw materials data. We demonstrated the flexibility and practical impact of the toolkit through its integration into the Materials Cloud ecosystem, in particular by facilitating the automatic deployment of OPTIMADE APIs for datasets contributed by the community to the Materials Cloud Archive [13]. Furthermore, we introduced a new Materials Cloud OPTIMADE client to make any data exposed via OPTIMADE easily accessible through an intuitive web GUI. We also described the mapping and serving of existing live databases through OPTIMADE for the Cambridge Structural Database (CSD) and the Inorganic Crystal Structure Database (ICSD).

`optimade-maker` and the automated Materials Cloud Archive OPTIMADE service mark an important step toward a fully FAIR (Findable, Accessible, Interoperable, and Reusable) ecosystem for materials science. By automatically exposing raw datasets through a standardized API and registering them in the OPTIMADE provider registry, datasets become immediately discoverable and interoperable across the OPTIMADE ecosystem. This enables seamless access through a wide range of clients and tools while placing no additional burden on data contributors, lowering the barrier to FAIR data publication and reuse.

Future developments in the OPTIMADE specification could further enhance this framework. Recent additions, including new entry types such as `files` and `trajectories` and an updated property definition for-

View Article Online
DOI: 10.1039/D6DD00125D

mat with controlled vocabularies and richer array typing [18], could be incorporated into `optimade-maker`. Support for externally defined community properties may also enable richer queries over existing datasets; for example, the `cheminfo` namespace [39] could enable molecular substructure searches via SMILES annotations.

More broadly, the approach could extend beyond structural data. By leveraging the federated registry of machine-actionable parsers developed in the *datatractor* initiative [40], similar API-based access could be provided for experimental datasets while reusing many of the components of `optimade-maker`.

Data availability

The source code of `optimade-maker` is released under the permissive MIT license and is available on GitHub at Materials-Consortia/optimade-maker and archived on Zenodo at DOI: 10.5281/zenodo.18863676. The datasets served by the Materials Cloud Archive OPTIMADE service are available on the Materials Cloud Archive [13] under Creative Commons licenses.

Acknowledgements

K.E., B.M, X.W., J.Y., N.M. and G.P. acknowledge funding by the NCCR MARVEL, a National Centre of Competence in Research, funded by the Swiss National Science Foundation (grant number 205602), and by the Open Research Data Program of the ETH Board (projects "API-03 IntER" and "PREMISE": Open and Reproducible Materials Science Research). M.L.E. thanks the BEWARE scheme of the Wallonia-Brussels Federation for funding under the European Commission's Marie Curie-Skłodowska Action (COFUND 847587) and the Leverhulme Trust for funding via an Early Career Research Fellowship. M.L.E. also thanks the UK's Physical Sciences Data Infrastructure (EP/X032701/1) and the Cambridge Crystallographic Data Centre, in particular Prof Simon Coles, Dr Ian Bruno and Dr Mehmet Giritli, for coordinating and motivating aspects of this work via commercial engagement with datalab industries ltd. We thank Valeria Granata for assistance with the Materials Cloud Archive integration.

Competing Interests

M.L.E. is the founder and director of datalab industries ltd.

Author contributions

K.E.: conceptualisation, data curation, investigation, methodology, resources, software, validation, visualisation, writing - original draft, writing - review and editing. M.L.E.: conceptualisation, data curation, funding acquisition, investigation, methodology, resources, software, validation, writing - original draft, writing - review and editing. B.M.: software, visualisation, writing - review and editing. X.W.: conceptualisation, software, writing - review and editing. J.Y.: conceptualisation, software, writing - review and editing. N.M.: conceptualisation, funding acquisition, resources, writing - review and editing. G-M.R.: conceptualisation, funding acquisition, resources, supervision, writing - review and editing. G.P.: conceptualisation, funding acquisition, resources, supervision, writing - review and editing.

References

[1] S. Gražulis, A. Daškevič, A. Merkys, D. Chateigner, L. Lutterotti, M. Quirós, N. R. Serebryanaya, P. Moeck, R. T. Downs, and A. Le Bail, "Crystallography Open Database (COD): an open-access collection of crystal structures and platform for world-wide collaboration", Nucleic Acids Research **40**, D420–D427 (2012).

[2] D. Zagorac, H. Müller, S. Ruehl, J. Zagorac, and S. Rehme, "Recent developments in the Inorganic Crystal Structure Database: theoretical crystal structure data and related features", Journal of Applied Crystallography **52**, 918–925 (2019).

[3] C. R. Groom, I. J. Bruno, M. P. Lightfoot, and S. C. Ward, "The Cambridge Structural Database", Acta Crystallographica Section B: Structural Science, Crystal Engineering and Materials **72**, 171–179 (2016).

[4] M. K. Horton et al., "Accelerated data-driven materials science with the Materials Project", Nature Materials, 1–11 (2025).

[5] J. Schmidt, H.-C. Wang, T. F. T. Cerqueira, S. Botti, and M. A. L. Marques, "A dataset of 175k stable and metastable materials calculated with the PBEsol and SCAN functionals", Scientific Data **9**, 64 (2022).

[6] R. Armiento, "Database-Driven High-Throughput Calculations and Machine Learning Models for Materials Design", in *Machine Learning Meets Quantum Physics*, edited by K. T. Schütt, S. Chmiela, O. A. von Lilienfeld, A. Tkatchenko, K. Tsuda, and K.-R. Müller (Cham, 2020), pp. 377–395.

[7] C. Draxl and M. Scheffler, "NOMAD: The FAIR concept for big data-driven materials science", MRS Bulletin **43**, 676–682 (2018).

[8] S. Haastrup et al., "The Computational 2D Materials Database: high-throughput modeling and discovery of atomically thin crystals", 2D Materials **5**, 042002 (2018).

View Article Online
DOI: 10.1039/D6DD00125D

[9]J. E. Saal, S. Kirklin, M. Aykol, B. Meredig, and C. Wolverton, "Materials Design and Discovery with High-Throughput Density Functional Theory: The Open Quantum Materials Database (OQMD)", JOM **65**, 1501–1509 (2013).

[10]M. Esters et al., "Aflow.org: A web ecosystem of databases, software and tools", Computational Materials Science **216**, 111808 (2023).

[11]S. P. Huber, M. Minotakis, M. Bercx, T. Reents, K. Eimre, N. Paulish, N. Hörmann, M. Uhrin, N. Marzari, and G. Pizzi, "MC3D: The Materials Cloud computational database of experimentally known stoichiometric inorganics", arXiv:2508.19223 [cond-mat], 10.48550/arXiv.2508.19223 (2025).

[12]N. Mounet et al., "Two-dimensional materials from high-throughput computational exfoliation of experimentally known compounds", Nature Nanotechnology **13**, Number: 3, 246–252 (2018).

[13]L. Talirz et al., "Materials Cloud, a platform for open computational science", Scientific Data **7**, 299 (2020).

[14]M. L. Evans and A. J. Morris, "*Matador*: a Python library for analysing, curating and performing high-throughput density-functional theory calculations", Journal of Open Source Software **5**, 2563 (2020).

[15]C. W. Andersen et al., "OPTIMADE, an API for exchanging materials data", Scientific Data **8**, 217 (2021).

[16]M. D. Wilkinson et al., "The FAIR Guiding Principles for scientific data management and stewardship", Scientific Data **3**, 160018 (2016).

[17]*OPTIMADE providers dashboard*, https://www.optimade.org/providers-dashboard/ (visited on 02/06/2026).

[18]M. L. Evans et al., "Developments and applications of the OPTIMADE API for materials discovery, design, and data exchange", Digital Discovery **3**, 1509–1533 (2024).

[19]V. Trinquet, M. L. Evans, C. J. Hargreaves, P.-P. D. Breuck, and G.-M. Rignanese, "Optical materials discovery and design with federated databases and machine learning", Faraday Discussions **256**, 459–482 (2025).

[20]V. Trinquet, M. L. Evans, and G.-M. Rignanese, "Accelerating the discovery of high-performance nonlinear optical materials using active learning and high-throughput screening", Journal of Materials Chemistry C **13**, 18197–18212 (2025).

[21]M. L. Evans, C. W. Andersen, S. Dwaraknath, M. Scheidgen, Á. Fekete, and D. Winston, "*optimade-python-tools*: a Python library for serving and consuming materials data via OPTIMADE APIs", Journal of Open Source Software **6**, 3458 (2021).

[22]S. R. Hall, F. H. Allen, and I. D. Brown, "The crystallographic information file (CIF): a new standard archive file for crystallography", Acta Crystallographica Section A: Foundations of Crystallography **47**, 655–685 (1991).

[23]H. J. Bernstein, J. C. Bollinger, I. D. Brown, S. Gražulis, J. R. Hester, B. McMahon, N. Spadaccini, J. D. Westbrook, and S. P. Westrip, "Specification of the Crystallographic Information File format, version 2.0", Journal of Applied Crystallography **49**, 277–284 (2016).

[24]A. H. Larsen et al., "The atomic simulation environment—a Python library for working with atoms", Journal of Physics: Condensed Matter **29**, 273002 (2017).

[25]S. P. Ong, W. D. Richards, A. Jain, G. Hautier, M. Kocher, S. Cholia, D. Gunter, V. L. Chevrier, K. A. Persson, and G. Ceder, "Python Materials Genomics (pymatgen): A robust, open-source python library for materials analysis", Computational Materials Science **68**, 314–319 (2013).

[26]G. Pizzi, A. Cepellotti, R. Sabatini, N. Marzari, and B. Kozinsky, "AiiDA: automated interactive infrastructure and database for computational science", Computational Materials Science **111**, 218–230 (2016).

[27]S. P. Huber et al., "AiiDA 1.0, a scalable computational infrastructure for automated reproducible workflows and data provenance", Scientific Data **7**, 300 (2020).

[28]M. Uhrin, S. P. Huber, J. Yu, N. Marzari, and G. Pizzi, "Workflows in AiiDA: Engineering a high-throughput, event-based engine for robust and modular computational workflows", Computational Materials Science **187**, 110086 (2021).

[29]Z. Wang, Y. Gong, M. L. Evans, Y. Yan, S. Wang, N. Miao, R. Zheng, G.-M. Rignanese, and J. Wang, "Machine Learning-Accelerated Discovery of A_2BC_2 Ternary Electrides with Diverse Anionic Electron Densities", Journal of the American Chemical Society **145**, 26412–26424 (2023).

[30]Z. Wang, Y. Gong, M. L. Evans, Y. Yan, S. Wang, N. Miao, R. Zheng, G.-M. Rignanese, and J. Wang, "Machine learning-accelerated discovery of A_2BC_2 ternary electrides with diverse anionic electron densities", Materials Cloud Archive, 10.24435/materialscloud:c8-gy (2023).

[31]V. Trinquet, M. L. Evans, C. Hargreaves, P.-P. De Breuck, and G.-M. Rignanese, "Optical materials discovery and design with federated databases and machine learning", Materials Cloud Archive, 10.24435/materialscloud:5p-vq (2024).

[32]V. Trinquet, M. L. Evans, and G.-M. Rignanese, "Accelerating the discovery of high-performance nonlinear optical materials using active learning and high-throughput screening", Materials Cloud Archive, 10.24435/materialscloud:wk-qm (2025).

View Article Online
DOI: 10.1039/D6DD00125D

[33] G. Bergerhoff, R. Hundt, R. Sievers, and I. D. Brown, "The inorganic crystal structure data base", Journal of Chemical Information and Modeling **23**, 66–69 (1983).

[34] *PSDI Cross Data Search*, https://data-search.psdi.ac.uk/ (visited on 03/10/2026).

[35] R. A. Sykes et al., "What has scripting ever done for us? The CSD Python application programming interface (API)", Journal of Applied Crystallography **57**, 1235–1250 (2024).

[36] *Pydantic GitHub Repository*, https://github.com/pydantic/pydantic (visited on 03/10/2026).

[37] P. Giannozzi et al., "QUANTUM ESPRESSO: a modular and open-source software project for quantum simulations of materials", Journal of Physics: Condensed Matter **21**, 395502 (2009).

[38] P. Giannozzi et al., "Advanced capabilities for materials modelling with Quantum ESPRESSO", Journal of Physics: Condensed Matter **29**, 465901 (2017).

[39] *OPTIMADE Cheminformatics Namespace*, https://github.com/Materials-Consortia/namespace-cheminformatics (visited on 02/06/2026).

[40] M. L. Evans, G.-M. Rignanese, D. Elbert, and P. Kraus, "Datatractor: Metadata, automation, and registries for extractor interoperability in the chemical and materials sciences", MRS Bulletin **50**, 838–845 (2025).

View Article Online
DOI: 10.1039/D6DD00125D

Data Availability Statement

The source code of \texttt{optimade-maker} is released under the permissive MIT license and is available on GitHub at https://github.com/Materials-Consortia/optimade-maker and archived on Zenodo at https://doi.org/10.5281/zenodo.18863676. The datasets served by the Materials Cloud Archive OPTIMADE service are available on the Materials Cloud Archive under Creative Commons licenses.