A Community Contribution Framework for Sharing Materials Data with Materials Project

Patrick Huck, Anubhav Jain, Dan Gunter, Donald Winston, Kristin Persson
Energy Storage & Distributed Resources Division
Lawrence Berkeley National Laboratory
Berkeley, California 94720
{phuck, ajain, dkgunter, dwinston, kapersson}@lbl.gov

***Abstract*—As scientific discovery becomes increasingly data-driven, software platforms are needed to efficiently organize and disseminate data from disparate sources. This is certainly the case in the field of materials science. For example, Materials Project has generated computational data on over 60,000 chemical compounds and has made that data available through a web portal and REST interface. However, such portals must seek to incorporate community submissions to expand the scope of scientific data sharing. In this paper, we describe `MPContribs`, a computing/software infrastructure to integrate and organize contributions of simulated or measured materials data from users. Our solution supports complex submissions and provides interfaces that allow contributors to share analyses and graphs. A RESTful API exposes mechanisms for book-keeping, retrieval and aggregation of submitted entries, as well as persistent URIs or DOIs that can be used to reference the data in publications. Our approach isolates contributed data from a host project's quality-controlled core data and yet enables analyses across the entire dataset, programmatically or through customized web apps. We expect the developed framework to enhance collaborative determination of material properties and to maximize the impact of each contributor's dataset. In the long-term, `MPContribs` seeks to make Materials Project an institutional, and thus community-wide, memory for computational and experimental materials science.**

I. Introduction

As a key player in the U.S. Materials Genome Initiative [1], Materials Project (MP) [2] uses High-Performance Computing (HPC) to determine structural, thermodynamic, electronic, and mechanical properties of over 60,000 inorganic compounds by means of high-throughput ab-initio calculations. The calculation results and analysis tools are disseminated to the public via modern web and application interfaces. These results and tools serve to accelerate the discovery, design and creation of next-generation materials for applications such as batteries, photovoltaics, and semiconductors.

However, the materials science research community has a continually increasing supply of experimental and theoretical material properties that are either not yet calculated by MP or outside the scope of MP's data generation efforts. Therefore, with a growing user base of over 10,000 registered users, it becomes increasingly important for MP and similar scientific platforms to also enable community-driven submissions, which would extend the scope of the possible applications and improve the integrity/quality of the provided datasets, hence enhancing its value to the user community.

The main contribution of this paper is a methodology for integrating and organizing an existing materials dataset with community contributions. We choose to approach this problem statement within the context of the Materials Project for two reasons: first, the solution discussed here represents an abstract solution in response to a specific problem for this initiative, rather than a specific solution in search of abstract problems. Second, we emphasize the "live" nature of our solution, which serves a large and existing user community and maintains an existing database as part of the initiative.

Our solution includes the following novel elements: (i) A syntax, `MPFile`, that allows users to easily describe computed or experimental data; (ii) A mechanism for hierarchically correlating user data with material(s) in the core project database; and (iii), A REST API and UI framework for disseminating results to the community via customizable interactive graphs.

The paper is organized as follows: Section II reviews related work. Sections III through V describe the major components of our framework. Sections VI and VII conclude the paper with a summary and outlook on the future.

II. Related Work

There are a number of community repositories in Materials Science and related disciplines. Traditionally, these repositories have been limited in scope to crystal structures of measured compounds (acting analogously to the Protein Data Bank [3] for biology), and include efforts such as the Inorganic Crystal Structure Database (ICSD) [4] and Pauling File [5]. More recent efforts, such as Materials Project [2], AFLOWlib [6], and others [7]–[11], typically include additional property data (generally computed) but do not currently have a contribution framework. Some repositories, such as NoMaD [12], iochem-bd [13], and the Materials Data Facility (MDF) [14], provide a shared repository of data, with potentially large experimental and structural data items, but many of these are in the early stages of development and without integrated data analysis capabilities over a core dataset. Our approach, by contrast, focuses on relatively small data contributions *that can be joined with the carefully curated core dataset* and extended with an integrated set of data analysis tools to provide a materials design environment.

A number of projects in other sciences, particularly life sciences, share the goals of community-enabling data con-

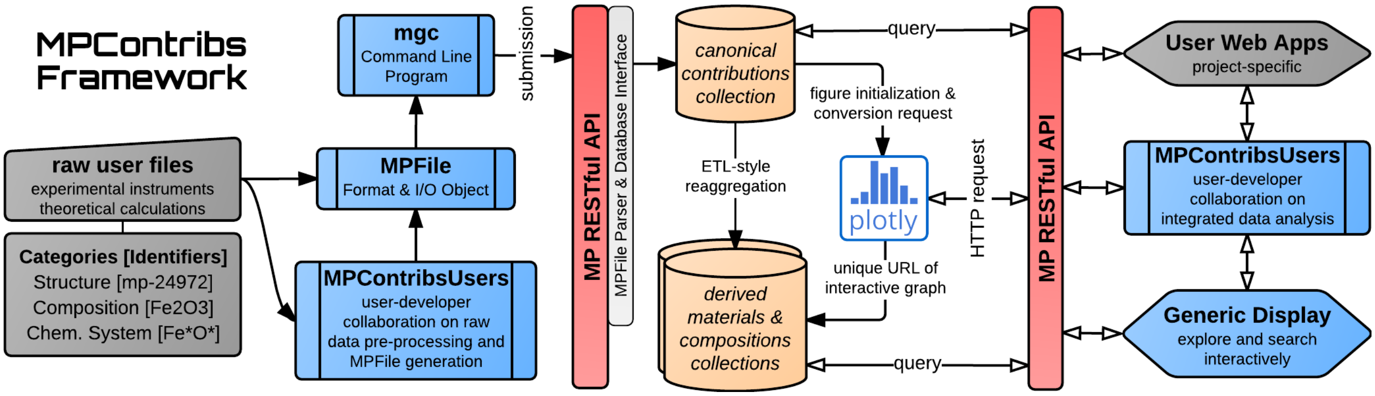


Fig. 1. Processing workflow of `MPContribs` framework. The pre-submission stages, with category identification and `MPFile` generation, are followed by internal parser and builder phases and the dissemination to the interactive portal. Contribution submission as well as queries to the databases (orange) and Plotly requests operate through the MP's REST API (red). Blue boxes and the processing behind the REST API "wall" are provided by the `MPContribs` framework. Grey boxes denote tasks and tools under the user's responsibility. For more details, see text.

tributions and analysis in the context of reference datasets. The iPlant Collaborative [15] combines a "Data Store" with a "Discovery Environment" with the goal of connecting public and private datasets with analysis capabilities; the analysis capabilities include workflows running on remote clusters and private Virtual Machines. The DOE's Systems Biology Knowledgebase (KBase) [16] provides a unified and interactive analysis environment built on the IPython notebook [17] and a back-end store that can store users' analyses together with both their data and standard reference microbial data. The breadth of either of these projects exceeds the near-term goals of the contribution framework described here, but point to the importance of this idea across scientific communities.

Efforts such as ChemSpider [18] emphasize aggregation of data from many individual sources rather than extension of "core" reference data. However, the success of such efforts depends on widely-used standards for identifiers [19] and does not integrate community contributions as added layers on top of a centrally generated and curated dataset. In contrast, our approach enhances the value of both the canonical data (by adding community contributions) and the community contributions (by adding the context of the canonical data). Combining these datasets requires the development of new methods. In the case of materials data/identifiers, the non-trivial problem arises of how to connect the "material" as defined by the user, e.g. through experimental characterization, with a "material" as defined by a computational dataset. Our classification of several types of contribution (see Sec. IV-A) is a novel approach to "divide and conquer" this type of problem.

In conclusion, most repositories in materials science that focus solely on community contributions face difficulties in (i) building a user audience, (ii) developing integrated search across datasets, and (iii) providing processed and interpretable data (e.g., interactive plots) rather than raw data dumps. Our integration of existing and novel datasets is a first attempt at addressing such issues. Thus, our framework occupies a unique and important niche in the materials data sharing space.

III. Framework Overview & Design

An overview of our framework `MPContribs` [20] is shown in Figure 1. From left to right, we see the processing workflow for incorporating a user contribution. First, the raw user output is identified and gathered. Examples of such data include:

- X-ray Absorption Spectroscopy (XAS) at light sources,
- diffusivities computed for various temperatures & solutes,
- band gaps calculated under alternative conditions,
- adsorption properties of nano-porous materials, and
- electronic structure calculations for photovoltaics.

Next, the raw data is converted into a concise summary using the `MPFile` syntax (Sec. IV). The result of this phase is submitted, by the user, via the dedicated command line program `mgc` (Sec. V-A) or manually through the web portal, both of which operate through the RESTful MP API (Sec. V-C). The received data is parsed (Sec. IV-D), and the resulting objects are correlated with existing materials data via suitable identifiers. The results are inserted into the core database. Finally, the data is disseminated through the API—and a graphical UI (Sec. V-B) with interactive graphs, tables and tree structures—for immediate exploration and analysis.

IV. Submission Processing

It is crucial to establish a common language for data interchange across contributors from different research disciplines and to encourage managable contribution sizes. During the pre-submission processing, the user's existing (raw) materials data information is thus converted into a custom but general input file format. This section describes in detail the format and processing steps provided by `MPContribs` for this task.

A. Material Definition & Contribution Scope

A key challenge for the framework is to link together separate datasets that pertain to the same material; this requires a universal comparator for the term *material*. MP defines a material as a group of structures that share the same symmetry group as well as comparable lattice parameters and atom positions, and employs a powerful matching code [21] to categorize materials within sensible tolerances. Each material is assigned a corresponding unique material identifier that maps directly to web resources, can be used in publications, and underpins the entire software/computing infrastructure.

A user's definition of a material might be broader or different than MP's; the framework uses a hierarchy of material categories to allow the user to attribute the contribution as is best suited to the dataset. At the top of the hierarchy is the *chemical space* ($\mathrm{Fe^*O^*}$), which includes many *compositions* (FeO, $\mathrm{Fe_2O_3}$, $\mathrm{Fe_3O_4}$) with the respective elements. Each composition can have multiple *structures*, which in turn correspond to a unique *MP material*.

Generally, a submission can be a contribution in any of the above categories. For example, data received from researchers at LBNL's Advanced Light Source typically refer to chemical spaces or compositions rather than materials. The category for a given contribution is determined by the parser (Sec. IV-D) from the MPFile's root-level section title (Sec. IV-B).

B. I/O File MPFile: Format & Object

In this section, we describe the MPFile format for the creation, editing, organization, and storage of user-contributed data. The format consolidates an experimental or theoretical contribution's tabular data together with its meta-data. MPFiles are plain text files that are designed to be easily read and edited by a person, yet also easily parsed and processed by programs. An MPFile resembles the Investigation Study and Assay Tabular format (ISATab) [22], which is an interchange format for experimental life sciences data with an associated tool suite and REST API. Unlike ISATab, MPFile works as free-form data without templates or ontologies and requires only a valid materials identifier.

Our MPFile format could best be described as CSV sections wrapped and extended by a minimal amount of meta-data necessary to customize the submission for MP. The extensions allow one to link data to existing materials, generate plots, and present hierarchical key-value pairs as well as free-form text. Apart from being human-readable and -editable, the file reflects the structural presentation on the front-end contribution page (Fig. 2). These features, along with native CSV support, is a major reason we chose to deviate from existing generic formats such as XML, JSON, and YAML.

The MPFile format indicates header levels by repetitions of a special character, a strategy borrowed from well-known markup languages such as Markdown and Wikitext. Markdown, for instance, uses the '#' character for this purpose. We chose instead the '>' symbol to avoid conflicts with the way comments are indicated in most CSV files in which the '#' character is reserved for commenting. To avoid collision with the mathematical '>>' sign and to improve visibility/readability, we use a minimum repetition of three ('>>>'). Text following this section delimiter is parsed as section name apart from comments appended on the same line.

Example IV-B.1 shows a sample MPFile for the contribution of data on basic physical properties for caesium and palladium. The MP internal identifiers for caesium and palladium are MP-1 and MP-2, respectively, which are used as section names to associate the contributed data with the corresponding entry in the core dataset. The same form of root-level section headers applies to composition identifiers.

The two line types are section headers (lines 1, 2, 9, etc.) and section contents. The section headers may be a reserved word (described below) or an arbitrary property title. The section contents can be either key/value pairs or a table of comma-separated values. The key/value pairs are used to organize meta-data or annotated numbers in free form using section names that best describe the contained data (lines 3–8, 10–11, 14–17). Possibly multiple tables of data in CSV format, including a header row with column names, are embedded in level-1 sections (lines 19–25, 27–30, 33–39). Note that there is no requirement for MPFile templates that enforce common key or column names. This provides maximum flexibility to the user while still enabling support for the hierarchy of Structure, Composition, and Chemical Space (Sec. IV-A). These restrictions can always be added later in response to a genuine need and independent agreement from within the user community. We also note that the infrastructure is geared towards kilobytes and not megabytes of data per submission. Large raw data can instead be linked to in the meta-data.

Example IV-B.1: Sample MPFile content for two materials. Line numbers and bold section headers were added for presentation purposes. The web page for the MP-1 section is shown in Figure 2.

```
1  >>> MP-1 # caesium
2  >>>> physical properties
3  phase: solid
4  melting point: 301.7 K
5  boiling point: 944 K
6  melting point density: 1.843 g/cm3
7  critical point temperature: 1938 K
8  critical point pressure: 9.4 MPa
9  >>>> references # list of url, bibtex, or doi
10 url-1: “https://en.wikipedia.org/wiki/Caesium”
11 url-2: “http://education.jlab.org/itselemental/ele055.html”
12 >>>> plots
13 >>>>> default data table 2 # overwrite graph properties
14 x: configuration
15 y: ionization energy
16 kind: bar
17 table: table 2
18 >>>> table 1 # can be named freely
19 T, vapor pressure
20 418,1
21 469,10
22 534,100
23 623,1000
24 750,10000
25 940,100000
26 >>>> table 2
27 electron number, ionization energy, configuration
28 1,375.7,6s1/2
29 2,2234.3,5p3/2
30 3,3400,5p1/2
31
32 >>> MP-2 # palladium (bare data section)
33 temperature (K), vapor pressure (Pa)
34 1721,1
35 1897,10
36 2117,100
37 2395,1000
38 2753,10000
39 3234,100000
```

The reserved section titles *general*, *plots*, and *references* indicate special significance and processing:

A *general* section (omitted in the example) is different from normal sections with key/value pairs only in two regards. In the future, MP might require certain unique key names in this section: a mandatory description field, for instance, to provide an informative overview of the contribution on the front-end. More importantly, the *general* section is employed to support an `MPFile` mode in which common meta-data is shared amongst contributions on multiple materials and/or datasets. Section IV-D1 describes the internal implementation of this feature during the `MPFile` parser phase.

By default, a two-dimensional graph is generated for each CSV table with the first column as the abscissae and each remaining column as corresponding ordinates for as many traces. The optional *plots* section can be used to request additional graphs or overwrite properties of the default ones based on the sections containing CSV tables. The plot axes are selected or changed by referring to the header of the table indicated in the *table* sub-key (lines 13–17). The keys chosen for this section (*x*, *y*, *kind*, etc.) are the same as those used by the Pandas Python library (through cufflinks [23]). Details on plot generation using Plotly are given in Section IV-D2.

The *references* section (lines 9–11) allows for the attribution of credit and for proper provenance of the submitted datasets. It only accepts a list of key/value pairs with the keys `url`, `doi`, or `bibtex`. The BibTeX string can either be obtained from the DOI or entered manually, and is subsequently resolved dynamically for prominent display on the front-end.

C. Collaborative Extension: `MPContribsUsers`

The `MPFile` format can be parsed and manipulated by Python functions/classes in the open-source `MPContribs` library. This allows users to construct `MPFiles` manually or through their own custom scripts. To coordinate between users and reduce duplication of work from writing code that deals with similar or identical formats in different domains of materials science, we created `MPContribsUsers` [20], a codebase hosted on Github containing reusable routines for both data generation and data analysis of `MPFiles` (c.f. Fig. 1, left and right of REST API "wall", respectively).

To engage the developer community, quality code practices are important, so for both core and extension repositories we use Github for source code control and issue tracking, and include extensive unit testing and documentation in the code, and in the project using version-controlled Markdown text.

The routines for data generation transform output files from common simulation software or from experimental instruments into the `MPFile` format. For instance, `MPContribsUsers` can assimilate data from the *VASP* software [24] by extending tools built in *pymatgen*. Other routines aid in determining the appropriate materials identifier for a contribution.

The collected code for data analyses performs *joint analyses* of MP core data and `MPFile` contributions. For instance, submitted experimental XAS spectra could be studied with respect to FEFF [25] calculations to relate the measured compositions to possible crystal structures. Similarly, XMCD signals resulting from the XAS spectra can be overlaid with the corresponding theoretical phase diagrams calculated by MP, providing a unified view of experiment and theory. The workflow of this *Unified Theoretical and Experimental x-ray Spectroscopy Application* in addition to `MPContribs`' role in the *Nanoporous Materials Explorer* are discussed in [26]. Note that such dedicated analysis applications within `MPContribsUsers` also drive the feedback loop of derived data back into the core and contribution collections. The collections can subsequently underpin project-specific web applications with capabilities beyond the generic display.

D. Internal Contribution Processing

The two main phases of the internal contribution processing are the parsing of the submitted `MPFile` (Sec. IV-D1), and the (re-)aggregation of single contributions (Sec. IV-D2) to build derived collections and generate interactive Plotly graphs.

1) `MPFile` *Parser:* The purpose of the `MPFile` parser is to (1) handle both the multiple-material and multiple-dataset contribution modes, (2) split the contribution data into "atomic" pieces defined by materials identifier, and (3) store each in a dedicated database with a unique contribution identifier. The underlying algorithm proceeds by recursively unpacking the `MPFile`'s section contents with increasing section level and hence allows for arbitrarily deep section nesting. For the multiple-dataset contribution mode (Sec. IV-B), the first root-level section denotes the main *general* section which is applied to all subsequent root-level sections. This effectively merges the two contribution modes and allows for meta-data to be shared amongst multiple datasets with local *general* subsections taking precedence. All section contents are imported into the internal recursive updating dictionary and data validation/cleaning handled by taking advantage of the tools provided by the Pandas library [27]. Pandas supports the import of data from many sources which makes it a suitable basis for the future implementation of additional features.

2) (Re-)Aggregation and Plot Generation: Derived collections are subsequently built from the canonical contributions collection by (re-)aggregating the single contributions for a specific materials identifier from many contributors. The resulting materials and composition collections serve as direct query access points for the front-end. In addition to ensuring that contributions are connected to the appropriate materials, this *builder* phase creates interactive plots from the user data. The graphs and their options, requested by the user in the `MPFile`'s *plots* subsection, are converted into an interactive graph through a HTTP request to Plotly's REST API [28]. The returned unique URL is saved in the derived collections and used to embed the graph on MP's front-end (c.f. Fig. 2). This allows direct access to the graph online to edit it to the contributor's liking on Plotly's workspace. The framework's integration with Pandas and Plotly during this builder phase is smart enough not to overwrite the customization of existing graphs when contributions are updated.

V. Submission Display and Dissemination

Submissions are accessible to users through a command-line client, an interactive web UI, and a RESTful API. This section describes each of these interfaces in more detail.

A. Command-line Interface

The command-line client interface, called `mgc` (for *Materials Genome Contribution*), uses the MP's REST API (Sec. V-C) to submit user-contributed data to MP for immediate visualization and dissemination to its users. The program accepts a variety of positional arguments to facilitate the contributor's submission management. For instance, the `submit` sub-command enables the submission of a physical `MPFile`. The initial submission embeds the generated contribution identifiers into the `MPFile` to facilitate book-keeping and to automatically trigger data updates upon resubmission.

B. Web UI

Our front-end interface provides access and display of contributed data in the context of the main MP web pages. Given the large number of users of the MP website, it is important that contributors can look at and optimize the final form themselves before public release of the contributed data. Thus, we provide an incubation period during which visibility of a given contribution is restricted to its contributor and invited collaborators. The primary task during this period is for contributors to ensure that the plots presented adhere to their standards for proper visual encoding of the data. Our plans for the system to enable quality control other than that pertaining to formatting are discussed in Sec. VII.

Furthermore, while an incubation period provides a temporary means of access control, we intend data submitted via `MPContribs` to ultimately be openly accessible. Data contributions for activities where the quality of access controls are critical are not well-suited for `MPContribs`; however, a scheme for "sandboxed" data access control has been implemented with success by MP for such situations [29].

We seek to accommodate a variety of implicit data schemas from contributors. Figure 2 depicts a minimal starting point for flexible display of user contributions that consist of arbitrary data in both hierarchical and tabular forms, along with a collection of plots. For the material depicted, a user may toggle among the contributing projects and their associated contributions. This allows a user/institution to quickly browse its own contributions for a material and also directly link to them, ultimately via assigned DOIs.

The interface encourages exploration. A tree of hierarchically structured data expands incrementally as characters are typed into its search field, allowing one to quickly search what may be a formidable hierarchy of key/value pairs. The tabular data display features an incremental-search field as well; the table is both paginated and scrollable. A contributing project's plots each link to the Plotly [28] workspace, so that users can jump directly into editing the graphs.

The built-in extensibility and portability of the plots served with each contribution (through Plotly) help users present explorable explanations of their contributions and retrieve publication-ready static assets. Optional widgets directly customize plot interactivity on the MP site, and plots can be transparently exported to a variety of file formats through simply adding an appropriate file extension to their URL. This portability also ensures that bandwidth-efficient static images of plots are served for small-screened mobile devices.

C. RESTful API

The implementation of `MPContribs`' REST API makes use of MP's existing code base to expose the I/O capabilities provided by the `MPFile` object (Sec. IV-B) and the database interface class. For example, we have implemented a general query function that can search based on `MPFile` section names and attributes. The resulting HTTP endpoints can be used via any language's HTTP library, but for Python dedicated convenience methods have been implemented as part of pymatgen's `MPRester` class.

Once we have a substantial collection of user contributions on which to practice inference of implicit schemas, or alternatively can allow users to explicitly assign known type classes to elements of their data, we can develop tools to enable further analyses of contribution data integrated with MP core data. Such tools, however, need to be developed with care to avoid limiting the framework's flexibility of data ingestion.

Note that the effort to provide a convenient REST API for `MPContribs` accelerates the development progress and user acceptance of the framework. It also allows the framework to be independently developed after it is released.

VI. Summary

As the rate of data generation in the materials science community continues to grow, the need for efficient and collaborative vehicles for data sharing becomes increasingly evident. The internet's history has shown that such continued growth is only possible by directly involving the community in data submission and vetting. In this paper, we introduced the Materials Project contributions framework, a software platform that allows users to share data with over 10,000 users.

A major design challenge is to link data pertaining to the same "material", as the identity of a real material can often neither be fully specified nor even rigorously defined. `MPContribs` utilizes the hierarchical nature of a material specification to intrinsically cover various contribution scopes: from computational data with full structural information to experimental results with only partial material descriptions.

The framework seamlessly integrates data composed of key/value pairs, data tables, nested data structures, and interactive plots. These features provide considerable flexibility in how users can contribute and present their data to the community. The framework implementation merges existing technologies with the custom but generic and human-readable `MPFile` format. Its parser supports two common submission modes and decomposes a contribution into suitable "atomic" MongoDB documents. Post-processing steps intelligently combine contributions from many users such that contributions belonging together can easily be rendered and served instantly to the public by the contribution-agnostic front-end.

Integration of the data handling with an existing collaborative graphing platform puts organization, maintainance, and formatting under users' control. A dynamic programmable REST API not only provides mechanisms for book-keeping and retrieval of submitted entries but also aggregates submissions as needed for integrated analyses that use metrics across contributed and core datasets. Collaboration on raw data processing and integrated analyses with the developer team is enabled by the separate `MPContribsUsers` area which serves as a growing repository of well-maintained user codes.

Our `MPContribs` toolset allows the materials researchers to present curated datasets to the broader community, and to tightly integrate them with those already available on MP. For developers, the presented solution provides a path from prototype to production with managable implementation efforts. Our effort is distinguished from others in that its goal is not to preserve and store large data files in their native format, but rather to provide a curated view of structured and processed data that is suitable for direct analysis and query and is integrated within an existing materials database.

VII. Outlook

We expect future work to be largely guided by user requirements determined through iteration and testing. However, several challenges are already apparent and our strategy to handle them will evolve as the number of contributors grows.

A) Defining universal lexicons for the description of materials and their properties across a broad user base is needed to unify the common elements of the datasets so that they can be queried and analyzed together.

B) Providing users with the ability to validate their data through continuous integration would greatly improve the robustness and reliability of user-submitted datasets. Currently, we closely collaborate with and limit contributions to trusted domain experts to produce curated, high-quality datasets. In this first stage, we aim to provide tools for efficient manual data inspection. For instance, a separate "staging" area for data contributions is already in place [26] and allows for human verification before release on the live web site. Further, our integration of the plotting tool Plotly quickly reveals incorrect data such as outliers. In the next stage, we will move from human to automatic self-verification of the data: we plan to release a data access API along with instructions on how users can employ the API to develop unit testing and continuous integration frameworks themselves. If problems of data quality persist, we will proceed to a third stage of development that allows users to rate each other's dataset quality which enables mechanisms like flagging, removal or "black-listing" of consistently low-quality datasets/users.

C) As the project scales to storing more data in a centralized manner, the issue of preventing data loss and scaling database performance/storage will become increasingly challenging.

However, despite these challenges, and although our framework is still in its early stages, feedback from several institutions has been positive and our project already has a waitlist of several research groups awaiting further developments. Therefore, we are confident of the need for a community-wide framework for sharing materials datasets and are optimistic about the potential of data sharing to accelerate scientific discovery and innovation.

Acknowledgment

This work was intellectually led by the Department of Energy's Basic Energy Sciences program - the Materials Project - under Grant No. EDCBEE and supported by Center for Next Generation Materials by Design, an Energy Frontier Research Center funded by DOE, Office of Science, BES. Work at Lawrence Berkeley National Laboratory was supported by the Office of Science of the U.S. Department of Energy under Contract No. DEAC02-05CH11231. We thank the National Energy Research Scientific Computing Center for providing invaluable computing resources. Finally, we would like to thank all the users of the Materials Project for their support and feedback in improving the project.

References

[1] Materials Genome Initiative. [Online]. Available: www.whitehouse.gov

[2] A. Jain *et al.*, "The Materials Project: A materials genome approach to accelerating materials innovation," *APL Mat. 1*, p. 011002, 2013.

[3] H. M. Berman *et al.*, "The Protein Data Bank," *Nucl. Acids Res.*, vol. 28, pp. 235–242, 2000. [Online]. Available: www.rcsb.org/pdb

[4] A. Belsky *et al.*, "New ICSD Developments: Accessibility in Support of Materials Research and Design," *Acta Cryst. B58*, pp. 364–369, 2002.

[5] Pauling File. [Online]. Available: http://paulingfile.com

[6] AFLOW for Materials Discovery. [Online]. Available: www.aflowlib.org

[7] NIST CCCBDB. [Online]. Available: http://cccbdb.nist.gov

[8] CompES Database. [Online]. Available: http://caldb.nims.go.jp

[9] C. Ortiz *et al.*, "Data mining and accelerated electronic structure theory as a tool in the search for new functional materials," *Comp. Mat. Sci.*, vol. 44, no. 4, pp. 1042–1049, 2009.

[10] Clean Energy DB. [Online]. Available: http://cepdb.molecularspace.org

[11] Open Quantum Materials Database. [Online]. Available: http://oqmd.org

[12] NoMaD Repository. [Online]. Available: http://nomad-repository.eu

[13] M. Alvarez-Moreno *et al.*, "Managing the Computational Chemistry Big Data Problem: The ioChem-BD Platform," *J. Chem. Inf. Model.*, vol. 55, pp. 95–103, 2015. [Online]. Available: http://iochem-bd.org

[14] Mater. Data Facility. [Online]. Available: www.nationaldataservice.org

[15] iPlant Collaborative. [Online]. Available: www.iplantcollaborative.org

[16] Systems Biology Knowledgebase. [Online]. Available: https://kbase.us

[17] F. Pérez and B. E. Granger, "IPython: a System for Interactive Scientific Computing," *Comp. Sci. Eng.*, vol. 9, no. 3, pp. 21–29, 2007.

[18] ChemSpider. [Online]. Available: www.chemspider.com

[19] S. Heller *et al.*, "InChI - the worldwide Chemical Structure Identifier Standard," *J Chem. Inf. Model.*, vol. 5, p. 7, 2013.

[20] P. Huck. Materials Project User Contribution Framework. [Online]. Available: https://github.com/materialsproject/MPContribs{Users}

[21] S. P. Ong *et al.*, "PyMatGen: A robust, open-source Python Library for Materials Analysis," *Comp. Mat. Sci.*, vol. 68, pp. 314–319, 2013.

[22] ISA MetaData Tracking. [Online]. Available: http://www.isa-tools.org

[23] J. Santos. Productivity Tools for Plotly + Pandas. [Online]. Available: https://github.com/santosjorge/cufflinks

[24] VASP. [Online]. Available: http://rcc.its.psu.edu/resources/software

[25] J. J. Rehr *et al.*, "Parameter-free calculations of X-ray spectra with FEFF9," *Phys. Chem. Chem. Phys.*, vol. 12, pp. 5503–5513, 2010.

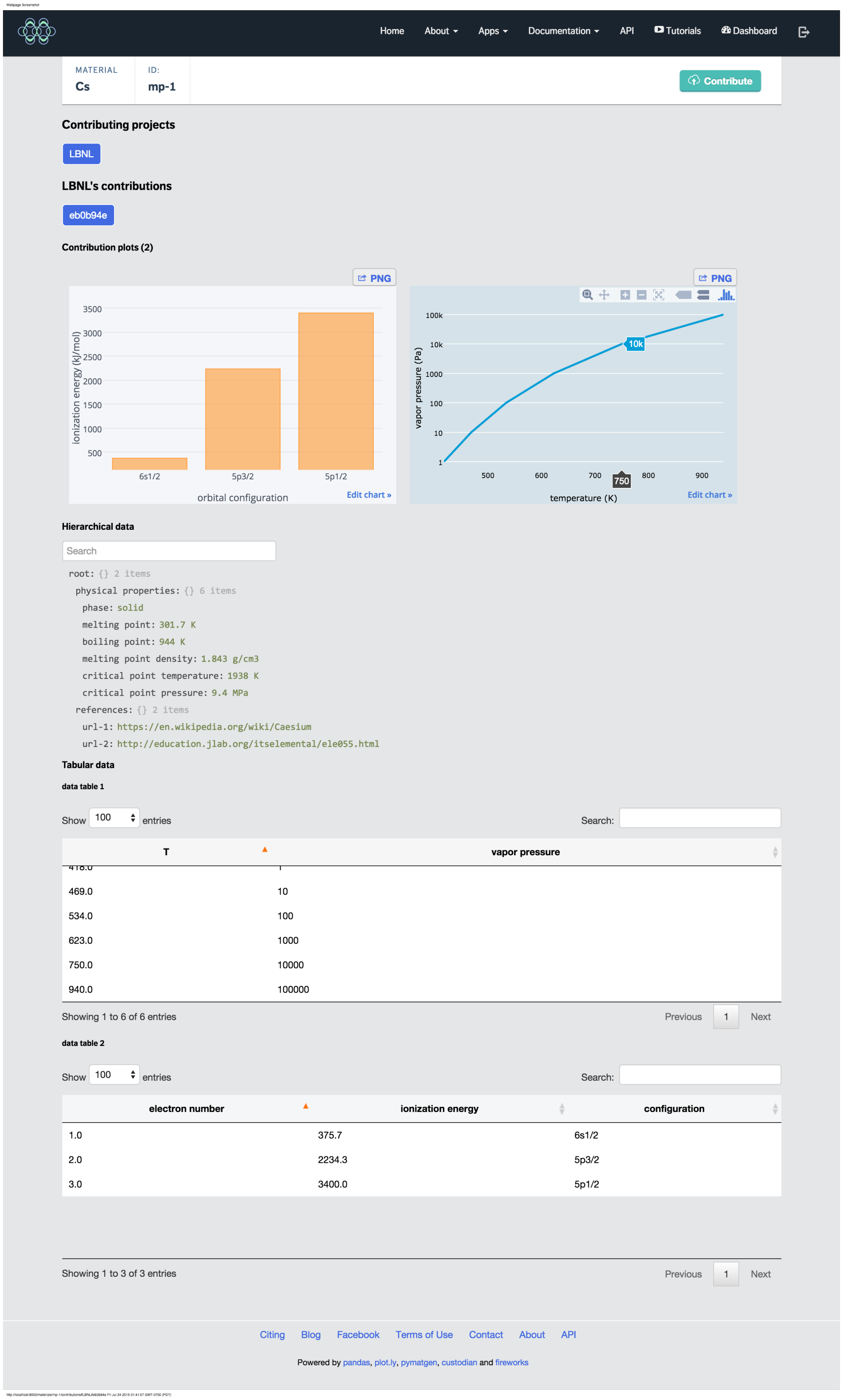
[26] P. Huck *et al.*, "User Applications Driven by the Community Contribution Framework MPContribs in the Materials Project," *Concurrency Computat.: Pract. Exper. (submitted)*, vol. 10th GCE Workshop, 2015.

[27] Pandas Python Library. [Online]. Available: http://pandas.pydata.org

[28] Collab. Graphing Platform Plot.ly. [Online]. Available: https://plot.ly

[29] X. Qu *et al.*, "The Electrolyte Genome project: A big data approach in battery materials discovery," *Comp. Mat. Sci.*, vol. 103, pp. 56–67, 2015.

Webpage Screenshot



MATERIAL
Cs

ID:
mp-1

Contributing projects

LBNL

LBNL's contributions

eb0b94e

Contribution plots (2)

Hierarchical data

Search

```
root: {} 2 items
  physical properties: {} 6 items
    phase: solid
    melting point: 301.7 K
    boiling point: 944 K
    melting point density: 1.843 g/cm3
    critical point temperature: 1938 K
    critical point pressure: 9.4 MPa
  references: {} 2 items
    url-1: https://en.wikipedia.org/wiki/Caesium
    url-2: http://education.jlab.org/itselemental/ele055.html
```

Tabular data

data table 1

Show 100 entries Search:

T	vapor pressure
418.0	1
469.0	10
534.0	100
623.0	1000
750.0	10000
940.0	100000

Showing 1 to 6 of 6 entries Previous 1 Next

data table 2

Show 100 entries Search:

electron number	ionization energy	configuration
1.0	375.7	6s1/2
2.0	2234.3	5p3/2
3.0	3400.0	5p1/2

Showing 1 to 3 of 3 entries Previous 1 Next

Citing Blog Facebook Terms of Use Contact About API

Powered by pandas, plot.ly, pymatgen, custodian and fireworks

http://localhost:8000/materials/mp-1/contributions#LBNL#eb0b94e Fri Jul 24 2015 01:41:07 GMT-0700 (PDT)

Fig. 2. Screenshot of MPContribs' front-end prototype showing the contribution with shortened identifier eb0b94e extracted from the MPFile contents of Example IV-B.1 (first root-level section). The user can toggle different contributing projects and according contributions. For a selected contribution, interactive plots are shown along with its hierarchical and tabular data (top to bottom). The user's cursor is over the plot at right, triggering an interactive mode. Note the search field for hierarchical data that reduces the presented data to sub-dictionaries containing the search string. A real-time demonstration of the workflow enabled by mgc and the MPContribs infrastructure can be found at https://youtu.be/lGwYLk2BlLA.