# Extended Abstract & Demo: Effective and interactive dissemination of diffusion data using MPContribs

Patrick Huck*[1], Henry Wu[2], Tam Mayeshiba[2], Everett Boyer[2], Dan Gunter[1], Dane Morgan[2], and Kristin Persson[1]
*Corresponding author address: Energy Storage and Distributed Resources Division, Lawrence Berkeley National Laboratory, Berkeley, CA, 94709, USA; email: phuck@lbl.gov
[1] Lawrence Berkeley National Laboratory; [2] University of Wisconsin

***Abstract:*** *MPContribs is the Materials Project's (MP) generic framework to enable its users to contribute and curate experimental or theoretical materials data and subsequently disseminate it to the MP community for increased public exposure. One such use case is the goal of the UW authors to share their dilute solute diffusion data with MP and build a targeted application for its exploration. In this presentation, we demonstrate the various MPContribs features by integrating the UW workflow and deploying it as a service on the MP's JupyterHub instance.*

## 1. Introduction

The authors affiliated with University of Wisconsin (UW) have recently made their data on dilute solute diffusion publicly available on FigShare [1]. This includes a spreadsheet shared amongst the collaborators to prepare the data prior to public release. The data contained in the spreadsheet is subsequently manually injected into a SQL database by a software engineer to make it accessible through a PHP web application.

This workflow, however, lacks the possibility of facilitated data updates, feature extensions, flexibility in data format, and most importantly sustainability for a research group composed primarily of non-engineers. Hence, the longer-term goal is to develop an automated and sustainable way for effective data dissemination to the broad materials sciences community without the overhead of manual data ingestion and management requiring technical personnel.

In an effort to achieve these advanced goals, the UW team joined with the staff from Materials Project (MP, [2]) to integrate the dissemination of their diffusion data with the modern MP infrastructure and hence expose the UW data to an existing community of about 25,000 users.

Presented at Gateways 2016, San Diego Supercomputer Center, La Jolla, CA, November 2-3, 2016.
https://gateways2016.figshare.com/

The MP's general contribution framework, *MPContribs* [3,4], is developed with the goal in mind to provide a sustainable solution for well-curated data management, organization and dissemination to MP users. The framework serves the purpose of collectively maintaining contributions to local and MP community databases as annotations to existing MP materials. It subsequently disseminates them through a generic interactive gateway powered by Jupyter notebooks or through custom project web apps enabled by the *webtzite* app kit [5].

*MPContribs* hence encapsulates the MP infrastructure expertise and enables the user to easily employ modern tools for data management and access such as MongoDB, REST APIs, and the Django web framework. With respect to the UW use case, the important features *MPContribs* provides are mechanisms for

1. the injection of custom pre-submission processing to automate local data retrieval and database ingestion, and
2. simplified MP-style web app development without the worry of database maintenance or data and user management.

The LBL team deployed *MPContribs* through the Materials Project JupyterHub. Installation, database setup etc. are hence not necessary for the user to get started and each JupyterHub user will have access to her own *MPContribs* interfaces via separate Docker images [6].

In the following sections, we demonstrate the integration of the UW workflow with *MPContribs* and JupyterHub. See [7] for a quick impression of its general functionality. The video and the demo illustrate how *MPContribs* can be used to contribute, explore and feed data to the generic contribution details pages as well as a project-specific web application.

## 2. Ingesting Data

### 1.1 Load MPFile

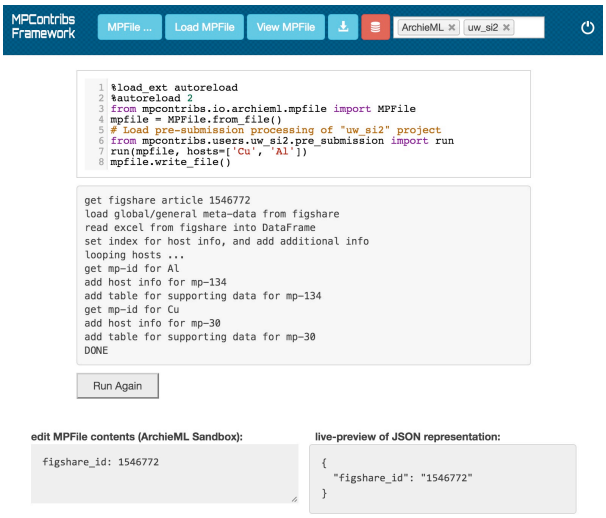


Fig. 1. The MPContribs Ingester is used to pre-process, prepare, and review the data to be submitted to local, on-site or MP databases.

The process of data ingestion is started by providing the *MPContribs Ingester* in Fig. 1 with the FigShare ID. The figshare_id key is used by the UW pre-submission processing code to retrieve the raw data via the FigShare API and prepare it for submission and dissemination through MPContribs.

The uw_si2 project is chosen to inject the respective pre-submission processing code [8] from the *MPContribsUsers* submodule into the Thebe Jupyter cell [9]. The code automatically loaded in the cell is editable to select a set of specific hosts to pull from FigShare, for instance. Executing the cell retrieves the most recent spreadsheet from FigShare, looks up MP identifiers for each host, and saves general as well as host-specific information including data tables into an in-memory MPFile. This human- and machine-readable text file in ArchieML format can be downloaded to disk.

Note that the Jupyter cell functionality does not need to be used to generate an MPFile. Instead, an MPFile can be written manually or with separate code using the *MPContribs* I/O library, and loaded into memory using the "MPFile …" and "Load MPFile" buttons.[1]

[1] An example for this approach is the use of an MPFile as configuration file with more extensive information than just the FigShare ID. This file could contain meta-data entered directly from lab book notes and instructions for the project-specific pre-submission code which routines to use during the extraction of processed data from raw instrumental data. Running the pre-submission code in the Jupyter cell would then simply update the in-memory MPFile based on the instructions in the previously loaded configuration MPFile.

### 1.2 View MPFile

Clicking "View MPFile …" after executing the Jupyter cell runs the in-memory MPFile through the *MPContribs* contribution machinery.

A static view of the hierarchical, tabular, and graphical data for each contribution (i.e. each mp-id or composition) in the MPFile is rendered in the browser. The view is based on Jupyter notebooks that are executed in memory and converted to HTML. Jupyter input cells are included in the view to show how each MPFile component can be accessed and rendered in a notebook using the *MPContribs* I/O library.

Buttons to toggle each component of all contributions at once and to go to a specific contribution directly, facilitate the navigation and review of the data prior to submission into local, on-site, or MP databases. In addition, an overview heatmap can be generated by selecting the numerical values of keys common to the hierarchical data of all contributions as x-, y-, and z-axes. Each data point can be clicked to jump to the underlying contribution directly.

### 1.3 Contribute MPFile

Clicking the red button with the database symbol starts the process of submitting the prepared and reviewed data to the local database or contributing it to MP. The different databases can be selected via a dropdown menu and the according API key for the *MPContribs* REST interface. The API and its documentation to programmatically access the public *MPContribs* database can be found at [10].

## 3. Exploring Data

### 1.4 Generic Contribution Explorer

Once a contribution has been processed and stored in the chosen database, a button directly takes the user to the generic Contribution Detail Page of the *MPContribs* Explorer [11]. The detail page is rendered from the data stored in the respective database and identical to the MPFile pre-view in the Ingester. It also includes Jupyter

input cells to show the user how to retrieve the contribution using the *MPContribsRester*.

The landing page of the *MPContribs* Explorer provides a generic interface to query the contributed data (or data submitted to local/on-site databases) for a list of materials and/or projects, or to show the URLs for a list of contribution identifiers. In its current rudimentary version, the interface simply returns a list of URLs to directly go to the respective Contribution Detail Pages. In the future, the results list will be improved to allow for graph-driven exploring of contributions, and to enable deeper content searches. The *MPContribs* Explorer for data contributed to MP at [11] is also available on localhost to instead explore data submitted to local/on-site databases.

### *1.5 UW Explorer*

As mentioned in the introduction, *MPContribs* facilitates the development of separate project-specific web applications as part of the *MPContribsUsers* submodule. This functionality allows *MPContribs* users to disseminate the contributed data in an interactive way that is tailored to their dataset (possibly in combination with other contributed datasets, and/or theoretical calculations provided by MP). The UW Explorer [12], for instance, has been developed by the UW and LBL teams to enable interested researchers to select host-solute combinations and plot diffusion coefficients as well as activation barriers. The app provides means to filter host/solutes, sort data tables, and look up detail pages for a specific material or contribution.

Repeating the *MPContribs* cycle in Sec. 1.1 with e.g. hosts=['Fe', 'Au'] would automatically add the new data to the database and include it in the UW Explorer (see [6]). The framework hence enables the researchers generating the data to quickly release and update their data without in-depth knowledge of the technical aspects driving the web application.

### *1.6 Jupyter Notebook*

Figure 2 shows how the *MPContribsRester* can be used in a Jupyter Notebook to conveniently retrieve contributed/submitted data through the *MPContribs* RESTful API and render the three MPFile components for interactive exploration, manipulation, and analysis. Every CSV table contained in the MPFile object as a Pandas DataFrame is also rendered such that the user can interactively sort it by columns, paginate through it, or search for specific table cells. In addition, an interactive Plotly graph is generated by default for each table. The graph exposes data values on mouse hover, allows (de-)selection of traces in the legend, and is downloadable as PNG (see [13]).

```
In [26]: from mpcontribs.rest.rester import MPContribsRester

In [27]: SITE = 'http://localhost:5000/test_site' # or http://alpha.materialsproject.org
         API_KEY = 'Gn6tOpaHMlEAsbTr' # copy from SITE dashboard
         ENDPOINT = SITE + '/mpcontribs/rest' # REST API endpoint for SITE
         mpr = MPContribsRester(API_KEY, endpoint=ENDPOINT)

In [28]: # make sure that user is registered on SITE as contributor
         assert(mpr.check_contributor()['is_contrib'])
         contribs = mpr.query_contributions() # get list of user contributions
         cid = contribs[0]['_id'] # contribution identifier (ObjectId)

In [29]: mpfile = mpr.find_contribution(cid) # get MPFile object from MPContribsRester
         mpid = mpfile.ids[0] # MP material identifier (e.g. mp-134)

In [30]: mpfile.hdata[mpid] # show hierarchical data contained in MPFile for mp-134
```

| | |
|---|---|
| doi | *10.6084/m9.figshare.1546772.v4* |
| funding | *NSF award No. 1148011* |
| figshare_url | *https://figshare.com/articles/DFT_dilute_solute_diffusion_in_Al_Cu* |
| Host_melting_temperature_K | *933* |
| title | *DFT dilute solute diffusion in Al, Cu, Ni, Pd, Pt, Mg, Fe, W, Mo, Au,* |

```
In [31]: mpfile.tdata[mpid]['data_supporting'] # show a specific table contained in MPFile
```

Out[31]:

| Hop activation barrier, E_0 [eV] | Hop attempt frequency, v_4 [THz] | Hop attempt frequency, v_2 [THz] |
|---|---|---|
| 0.58143 | 4.22424 | 4.22424 |
| 0.58143 | 4.22632 | 1.3229 |
| 0.58143 | 4.25005 | 0.816836 |
| 0.58143 | 4.36636 | 1.56213 |
| 0.58143 | 4.21968 | 1.79218 |

```
In [32]: mpfile.gdata[mpid]['supporting'] # show graphs for specific table contained in MPFile for mp-134
```

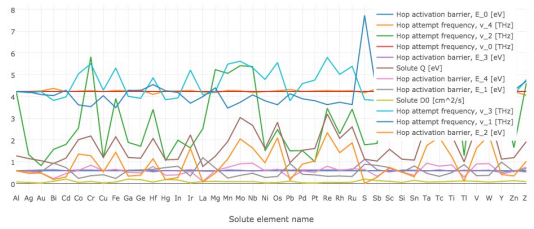


Fig. 2. MPFile Components in Jupyter Notebook.

## 4. Conclusion

We describe how the general approach taken by *MPContribs* solves the very specific challenges faced by the UW researchers in effectively disseminating their data to the public. The presented solution developed in the collaborative effort between UW and LBNL is the first to demonstrate how *MPContribs* can empower research groups through the rapid development and deployment of customized but MP-compatible web applications either using on-site or MP resources.

These efforts directly translate into solutions for the ongoing collaboration with researchers at the Advanced Light Source at LBNL [4] in which we aim to develop a processing pipeline for experimental XAS data from the beamline computer to integrated analysis web apps on MP.

## 5. References


[1] DFT dilute solute diffusion in Al, Cu, Ni, Pd, Pt, Mg, Fe, W, Mo, Au, Ca, Ir, and Pb; https://dx.doi.org/10.6084/m9.figshare.1546772.v4

[2] Materials Project; https://materialsproject.org/

[3] MPContribs GitHub Repository; https://github.com/materialsproject/MPContribs

[4] MPContribs; https://arxiv.org/abs/1510.05024; https://arxiv.org/abs/1510.05727; MRS Spring '16

[5] Webtzite; https://github.com/materialsproject/webtzite

[6] Sandboxed User Envs w/ JupyterHub and Docker. Demo at Gateways2016 by S. Cholia.

[7] MPContribs and UW Video Demo; https://www.youtube.com/watch?v=wbWde5StHnU

[8] UW pre-submission processing; https://github.com/materialsproject/MPContribsUsers/blob/9199df/uw_si2/pre_submission.py

[9] Thebe; https://github.com/oreillymedia/thebe

[10] MPContribs REST API; http://alpha.materialsproject.org/mpcontribs/rest

[11] MPContribs Explorer; http://alpha.materialsproject.org/mpcontribs/explorer

[12] UW Explorer; http://alpha.materialsproject.org/uwsi2/explorer

[13] Plotly; https://plot.ly