

UC Berkeley

UC Berkeley Previously Published Works

Title

A representation-independent electronic charge density database for crystalline materials

Permalink

<https://escholarship.org/uc/item/2mk0g4ww>

Journal

Scientific Data, 9(1)

ISSN

2052-4463

Authors

Shen, Jimmy-Xuan

Munro, Jason M

Horton, Matthew K

et al.

Publication Date

2022

DOI

10.1038/s41597-022-01746-z

Copyright Information

This work is made available under the terms of a Creative Commons Attribution License, available at <https://creativecommons.org/licenses/by/4.0/>

Peer reviewed



OPEN

DATA DESCRIPTOR

A representation-independent electronic charge density database for crystalline materials

Jimmy-Xuan Shen^{1,2}, Jason M. Munro³, Matthew K. Horton^{1,4}, Patrick Huck³, Shyam Dwaraknath⁴ & Kristin A. Persson^{1,4}✉

In addition to being the core quantity in density-functional theory, the charge density can be used in many tertiary analyses in materials sciences from bonding to assigning charge to specific atoms. The charge density is data-rich since it contains information about all the electrons in the system. With the increasing prevalence of machine-learning tools in materials sciences, a data-rich object like the charge density can be utilized in a wide range of applications. The database presented here provides a modern and user-friendly interface for a large and continuously updated collection of charge densities as part of the Materials Project. In addition to the charge density data, we provide the theory and code for changing the representation of the charge density which should enable more advanced machine-learning studies for the broader community.

Background & Summary

The application of Density-Functional Theory (DFT) to many-electron systems has witnessed tremendous growth in the past few decades and has now become the *de facto* simulation tool for physicists, chemists, and materials scientists. The central concept of DFT is that the energy, and in turn all of the physical properties of a quantum system, are completely determined by the electronic charge density of the ground state $\rho(\mathbf{r})$ with \mathbf{r} the position vector¹. The majority of the computational cost in typical DFT calculations is associated with determining ρ via an iterative algorithm to arrive at a self-consistent charge density². For the most commonly used exchange-correlation functionals, like the local-density approximation (LDA)^{2,3} and the semi-local functional by Perdew-Burke-Ernzerhof (PBE)⁴, a converged charge density can be used as the starting point for more expensive calculations such as obtaining a detailed bandstructure⁵ or calculating the optical response of the material⁶.

In addition to its central role in standard DFT calculations, the charge density itself is also useful for the analysis of many materials properties. The critical points of the charge density (i.e. where the gradient is zero) are often used as a boundary between atomic neighborhoods. In turn, this allows for a systematic assignment of charge to specific atoms^{7,8}, as well as the determination of bonding character between neighboring pairs⁹. Within the realm of energy materials, the charge density can be used as an effective potential to study the migration properties of Li in solid-state materials, as low charge density provides a metric of “free” space in a lattice^{10,11}. Consequently, the local minima of the charge density can act as initial guesses for the positions of inserted cations¹².

A single DFT calculation of the primitive unit cell provides one representation of the charge density within that particular basis set. However, depending on the application, alternative representations might be desired. An important example of this is in machine-learning (ML) algorithms, where obtaining a consistent data representation is essential for deep-learning methods. However, the representation of charge density is heavily influenced by the simulation cell and the Bravais lattice of the periodic structure. Hence, a necessary step in using electronic charge densities in machine-learning applications is to obtain alternative representations of the same charge density that represent the same underlying field. While recent work has examined the effectiveness of representations in Fourier space¹³, any ML investigation of local interaction (e.g. adsorption and intercalation of ions) requires flexible representations in real space. Towards that end, our framework will provide code to

¹Department of Materials Science and Engineering, University of California, Berkeley, Berkeley, California, 94720, USA. ²Lawrence Livermore National Laboratory, Livermore, USA. ³Energy Technologies Area, Lawrence Berkeley National Laboratory, Berkeley, 94720, USA. ⁴Energy Sciences Area, Lawrence Berkeley National Laboratory, Berkeley, 94720, USA. ✉e-mail: kapersson@lbl.gov

obtain arbitrary real space representations of charge density for a given material directly from a DFT-computed charge density.

The charge density of any crystalline solid, and indeed any periodic field, is naturally represented in a plane-wave basis set, where the inherent periodicity of the system is embedded in the underlying representation. For a sufficiently converged finite plane-wave basis, the continuous charge density $\rho(\mathbf{r})$ and its Fourier transform $\phi(\mathbf{k})$ can be accurately sampled by a three-dimensional array indexed by i, j , and k with N_1, N_2 , and N_3 evenly spaced grid-points along each lattice vector, and can be converted from one to the other via a discrete Fourier transform

$$\rho(\mathbf{r}) \equiv (\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3, \rho_{i,j,k}) \xleftrightarrow[\mathcal{F}^{-1}]{\mathcal{F}} (\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3, \phi_{i,j,k}) \equiv \phi(\mathbf{k}). \quad (1)$$

where \mathbf{a}_α and \mathbf{b}_α represent the real and reciprocal lattice vectors, and i, j , and k are the indices of regularly-spaced grid points along the lattice vectors. Due to the discrete nature of numerical Fourier transforms, the number of grid points of a real-space representation is always equal to the number of plane waves needed to represent the data in reciprocal space.

A representation of the charge density is uniquely determined by three vectors and a scalar matrix either in real or reciprocal space. Each representation only provides a “view” of the infinite periodic data represented in a specific unit cell and an infinite number of such representations exist for a given charge density. Regardless of the grid size and the periodic cell representation, the DFT-computed charge density represents the same underlying field, yet it is routinely recomputed when any change is needed in the representation, even when the computational parameters are unchanged, often at considerable expense. One common example is the use of the electrostatic potential of a super cell to correct for the periodic image effects of a charged defect¹⁴.

Due to the significant amount of computational resources devoted to computing the electronic charge densities, as well as the growing domains of their application, especially for the training of data-intensive machine learning models, there is a pressing need for a large-scale representation-independent database of charge densities. The Materials Project (<https://materialsproject.org>) - as a rapidly growing (currently more than 215,000 users) materials informatics resource - is a natural platform for the dissemination of such data. The structural and thermodynamics information of solid-state materials are available across other quantum chemical databases such as AFLOW¹⁵, NOMAD¹⁶, and OQMD¹⁵, and a targeted charge density dataset of materials with cubic symmetry has recently been published¹⁷. Our work aims to be the first set of publicly available charge density data for periodic systems without constraints on the structural family. The data and API will be maintained as part of the Materials Project ecosystem. The work presented in this article provides details on how the charge densities in our database are computed and how they can be accessed. In addition, we provide a high-level API for querying and post-processing the charge density data. Among other features, the API will allow users to take an existing atomic structure and query for charge density of the same material, in the representation/view of the user's choosing.

Methods

In this section, we provide details on the scope of the charge densities database and the precise set of computational parameters used to generate the data. Additionally, we will demonstrate features of the API that allow users to obtain arbitrary views of the charge density data, including up-sampling/compressing the data via Fourier analysis and symmetry operations like translations, rotations, and super cell transformations.

Calculation parameters. The charge densities are obtained from DFT calculations performed using the static calculation workflow within the atomate software package¹⁸, and relaxed input structures from the Materials Project (MP) database¹⁹. The projector-augmented wave (PAW) method as implemented in the plane-wave Vienna Ab-initio Simulation Package (VASP) is used in conjunction with the PBE generalized-gradient approximation functional⁴. The default set of MP calculation input parameters was used, which have been demonstrated to produce well-converged results²⁰. Included in these parameters is an energy cutoff of 520 eV, a total energy error threshold of 5×10^{-5} eV/atom, and a reciprocal k -point density of $100/\text{\AA}^{-3}$. The only addition made to the input set is to enable aspherical contributions in the gradient corrections inside the PAW spheres. Hubbard U -corrections are included with materials containing oxygen and fluorine. Elements Co, Cr, Fe, Mn, Mo, Ni, V, and W use values of 3.32, 3.70, 5.30, 3.90, 4.38, 6.20, 3.25, and 6.20 eV, respectively.

Changing the charge density representations. Given one representation $(\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3, \rho_{i,j,k})$ of the charge density ρ , we may transform it to any other representation $(\mathbf{a}'_1, \mathbf{a}'_2, \mathbf{a}'_3, \rho'_{i',j',k'})$ by resampling the data. Due to computation time and data storage constraints, DFT codes will typically use the fewest grid points possible to represent the charge density which limits the effectiveness of local interpolation schemes. However, since our charge densities have periodic boundary conditions and are reasonably smooth (owing to the use of pseudo-potentials), the charge density can be first represented in Fourier space and then interpolated. We can up-sample our data via Fourier interpolations²¹ as shown in Fig. 1. The procedure to perform Fourier interpolation of real space data is as follows:

1. Take the discrete Fourier transform of $\rho_{i,j,k}$.
2. Augment the resulting Fourier data $\phi_{i,j,k}$ with zero-valued higher frequency components.
3. Apply the reverse transformation to obtain the up-sampled data.

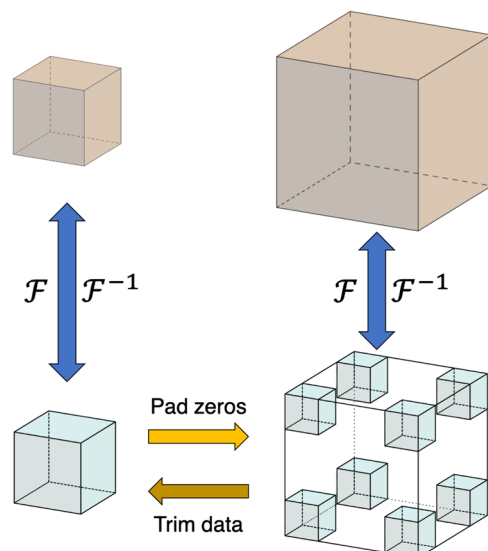


Fig. 1 Schematic of data transfer for Fourier interpolation and compression. The more densely sampled (larger) and the coarsely sampled (smaller) 3D blocks of real-space data can each be transformed to Fourier space, resulting in a Fourier representation of the same size. To up-sample the data, we use the smaller block in Fourier space, augment with zeros while keeping all the data fixed near the origin (at the corners of the cube). To compress, we crop the data in Fourier space and perform an inverse Fourier transform.

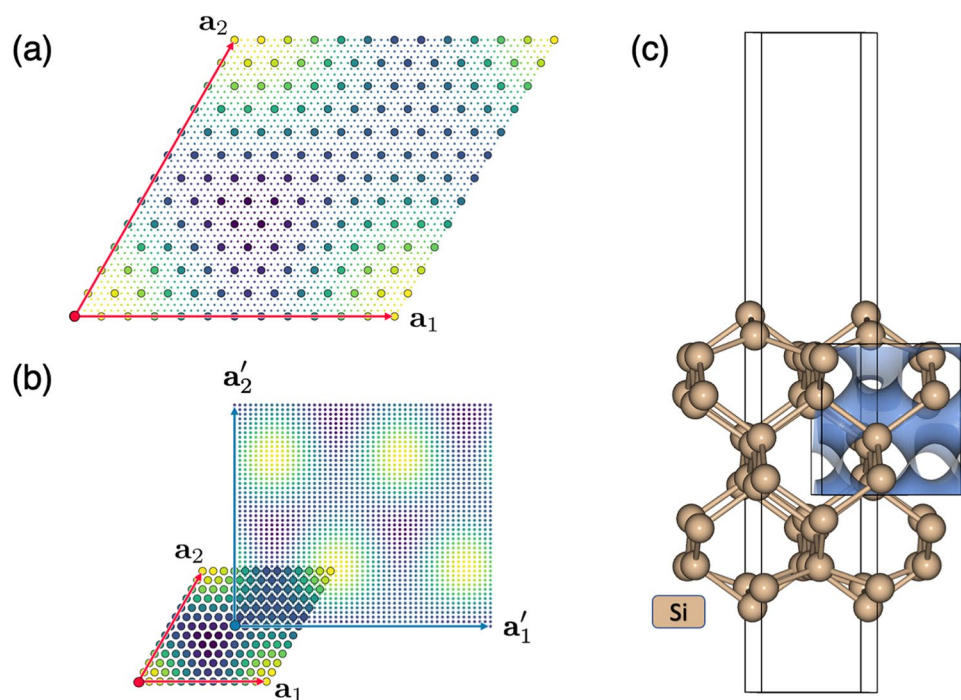


Fig. 2 Periodic re-gridding applied to a 2D slice of the charge density for a two atom Si unit cell where each large circle corresponds to a data point in the original 12×12 grid. The result of Fourier-interpolating the data in the original unit cell onto a 48×48 grid is shown in (a). The transformed periodic representation $\mathbf{a}'_1 = 2\mathbf{a}_1$ and $\mathbf{a}'_2 = 2\mathbf{a}_2 - \mathbf{a}_1$ with a shift of $\mathbf{t} = (\mathbf{a}_1 + \mathbf{a}_2)/2$ and a grid size of 48×48 is shown in (b). A cubic crop of a Si slab calculation is shown in (c).

The augmented Fourier data is mathematically equivalent to the original Fourier data. Thus, the inverse transform of the augmented Fourier data must be equivalent to the original real space data sampled at a higher density. Increasing the grid density using Fourier interpolation enables us to up-sample $\rho_{i,j,k}$ in each direction by a factor of γ_{up} . We may then resample the local grid with a linear interpolation scheme to ensure the fidelity of our data.

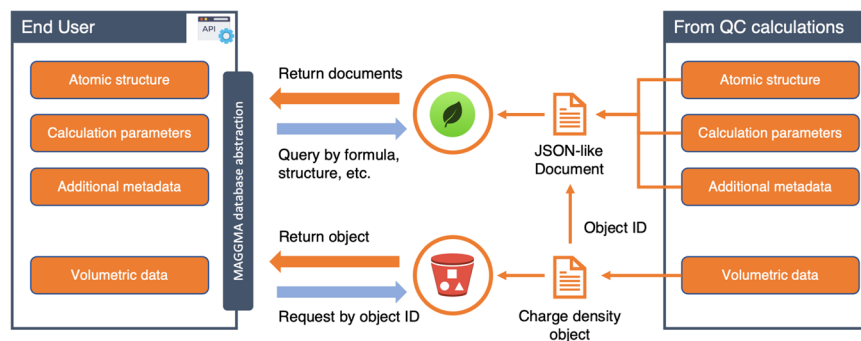


Fig. 3 Data pipeline for the charge density database, illustrating how the output files from quantum chemistry calculations are stored and accessed. Basic data are first converted into a JSON-like format to be stored on a MongoDB server, which allows queries on any of the stored fields. The charge density data is converted into an array-like object with additional meta-data (e.g. ObjectID) and stored in an AWS S3 bucket. Since the ObjectID is stored as a field in the MongoDB database, the API is able to combine the MongoDB data with the according S3 object and reconstruct the original data.

Given a primitive-cell representation of the charge density — $(\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3, \rho_{i,j,k})$, any periodic representation of a scalar field $f(\mathbf{r})$ can be understood as applying an arbitrary translation on the unit cell by a vector \mathbf{t} :

$$\hat{T}_{\mathbf{t}}f(\mathbf{r}) \equiv f(\mathbf{r} - \mathbf{t}), \quad (2)$$

followed by a super cell transformation \hat{P} defined as an integer matrix with $\det(\hat{P}) \geq 1$ which acts on the lattice vectors from the right

$$(\mathbf{a}'_1 \mathbf{a}'_2 \mathbf{a}'_3) = (\mathbf{a}_1 \mathbf{a}_2 \mathbf{a}_3) \hat{P}. \quad (3)$$

Our software is capable of performing the same operations in arbitrary dimensions. As an example, in Fig. 2, we show the results of re-gridding using a 2D slice of the charge density in a two-atom Si unit cell which only cuts across a single atom at the origin, Fig. 2(a) shows the result of Fourier interpolating the field from a 12×12 grid (large circles) onto a 48×48 grid (smaller circles). In Fig. 2(b), the modified representation is obtained by first shifting the origin to the center of the cell at $\mathbf{t} = (\mathbf{a}_1 + \mathbf{a}_2)/2$ followed by a change of basis to $\mathbf{a}'_1 = 2\mathbf{a}_1$ and $\mathbf{a}'_2 = 2\mathbf{a}_2 - \mathbf{a}_1$.

While integer-valued super cell transformations will yield an equivalent periodic underlying charge density, non-integer basis transformations are used to obtain an arbitrary crop of periodic charge density sampled at any density. As an example, we show how a non-periodic cubic sample of the surface charge density can be obtained from the slab calculation in Fig. 2(c). The simulation was performed using a $7.73 \text{ \AA} \times 3.87 \text{ \AA} \times 21.88 \text{ \AA}$ orthorhombic Si slab cell and the charge density is stored on a $120 \times 60 \times 336$ grid. A $5 \text{ \AA} \times 5 \text{ \AA} \times 5 \text{ \AA}$ cropped region of the charge density sampled on a $48 \times 48 \times 48$ grid is indicated by the blue iso-surface in Fig. 2(c). It is important to note that the cropped cell can exceed the boundaries of the original simulation cell in any direction. In the example, the smallest dimension of the simulation cell is 3.87 \AA while the cropped cube has side lengths of 5 \AA . This feature essentially allows us to obtain the charge density in any preferred real-space dimensions, independent of the simulation cell parameters. Additionally, this allows us to freely choose the simulation cell in situations where periodic-image effects are not important.

Database details. We use a hybrid data model to serve the data: Queryable data such as chemical formula, total energy, and calculation parameters are served as JSON-like documents using MongoDB, while much larger and not-queryable charge density data is served using AWS S3 object storage²². When a charge density is parsed from the output file to a serialized object, a unique Object ID is assigned and stored alongside the other data in the MongoDB database. From the client's perspective, two concurrent requests are made: one to obtain calculation inputs and outputs from MongoDB, and another for the charge density data from the S3 bucket. A visual representation of the data flow is provided in Fig. 3.

Data Records

The dataset itself can be viewed through the Materials Project website www.materialsproject.org²³. The raw charge density data output from DFT calculations can be obtained from the corresponding MP API endpoint: https://api.materialsproject.org/charge_density. Each entry can be referenced with a particular DOI through the associated MP material entry. Additionally, the input parameters for the specific calculation used to generate the entry can be obtained from the tasks endpoint at <https://api.materialsproject.org/tasks>. Details for how to interact with the referenced endpoints can be found in the [sec:usage]Usage Notes section.

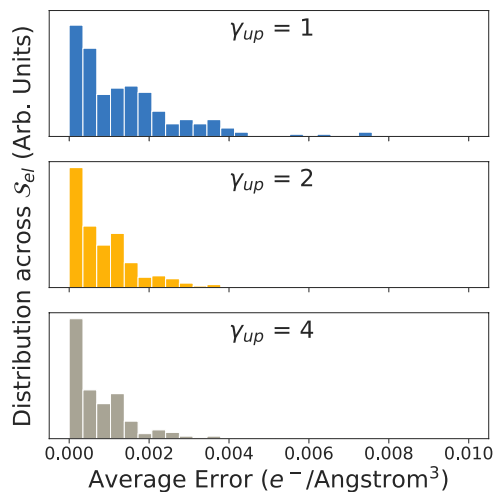


Fig. 4 Distributions of the errors between re-sampled charge density and explicitly calculated charge densities.

```

from mp_api.client import MPRester

# Obtain the CHGCAR associated with a given material
with MPRester(<API_KEY>) as mpr:
    chgcar = mpr.get_charge_density_from_material_id("mp-149") # Si

# Obtain the CHGCAR associated with a given material
# including additional information about the VASP inputs and output
# in dictionary format
with MPRester(<API_KEY>) as mpr:
    chgcar, task_doc = mpr.get_charge_density_from_material_id("mp-149", inc_task_doc=True)
    calc_in = task_doc.calcs_reversed[0].input
    calc_out = task_doc.calcs_reversed[0].output

```

Fig. 5 Example code for querying the charge density database.

Technical Validation

We can elucidate the performance of the re-gridding algorithm using a larger set of elemental polymorphs from the Materials Project. For this test set S_{el} , we selected 117 single-element structures from MP for which the energy above the convex hull was less than $1 \mu\text{eV}$ and the number of atoms in the unit-cell was less than 20. For each structure in S_{el} , we perform VASP static calculations on the primitive unit cell and on two super cells created by:

$$\hat{P}_1 = \begin{pmatrix} 1 & 1 & 0 \\ 1 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad (4)$$

and

$$\hat{P}_2 = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{pmatrix}. \quad (5)$$

For each charge density obtained using an explicit super cell calculation, we obtain the average error compared to a super cell charge density obtained from transforming the unit cell charge density. The results of the comparison are shown in Fig. 4. We observe that using an up-sampling factor of $\gamma_{up} = 4$, the transformed and explicitly calculated pseudo-charge density, whose values can range from 0 in vacuum to $100 \text{e}^-/\text{\AA}^3$ near the atomic cores, only exhibits a difference of $<0.002 \text{e}^-/\text{\AA}^3$. The data and code used in the validation procedure can be accessed as a direct download²⁴.

Usage Notes

To facilitate access to this data, convenience functions have been implemented as part of the Materials Project REST API client. These are contained within the `MPRester` class as part of the `pymatgen` software package¹⁹. More specifically, the member function `get_charge_density_from_material_id` is provided to send requests to the API endpoints. This function takes as input the Materials Project ID associated with a given material in the database and returns a `Chgcar` object. If the `inc_task_doc` flag is set to `True`,

an additional task document containing all of the calculation details will also be returned. The VASP calculation that produced the charge density is always the last calculation that corresponds to the zeroth entry in the `calcs_reversed` list. And the inputs and outputs are stored as dictionary key-value pairs under `input` and `output` attributes. With the `MPRester` class imported, the code workflow in Fig. 5 can be used to query for the charge density and calculation details.

In order to alter the representation of the charge density obtained from the API endpoint, the `pyRho` python package <https://github.com/materialsproject/pyrho> can be used alongside the `pyMatgen Chgcar` object obtained from the API query. Examples of how to re-grid, interpolate, and visualize are included in the repository as a set of Jupyter²⁵ notebooks.

Code availability

Access to the charge density data provided by the Materials Project API (<https://github.com/materialsproject/api>) and grid transforms of the charge density is done using the `pyRho` python package. See the [sec:usage] Usage Notes section for more information. The scripts used to generate the validation data can be access at along with the direct download of the validation dataset²⁴

Received: 22 February 2022; Accepted: 6 October 2022;

Published online: 28 October 2022

References

- Hohenberg, P. & Kohn, W. Inhomogeneous Electron Gas. *Phys. Rev.* **136**, B864–B871, <https://doi.org/10.1103/PhysRev.136.B864> (1964).
- Kohn, W. & Sham, L. J. Self-Consistent Equations Including Exchange and Correlation Effects. *Phys. Rev.* **140**, A1133–A1138, <https://doi.org/10.1103/PhysRev.140.A1133> (1965).
- Ceperley, D. M. & Alder, B. J. Ground State of the Electron Gas by a Stochastic Method. *Phys. Rev. Lett.* **45**, 566–569, <https://doi.org/10.1103/PhysRevLett.45.566> (1980).
- Perdew, J. P., Burke, K. & Ernzerhof, M. Generalized gradient approximation made simple. *Phys. Rev. Lett.* **77**, 3865 (1996).
- Martin, R. M., Martin, R. M. & Press, C. U. *Electronic Structure*. <https://books.google.ca/books?id=dmRTFLpSGNsC&printsec=frontcover&dq=isbn:0521782856&hl=en&sa=X&ved=2ahUKewiO4OfwqJ7sAhWMTt8KHAV7C6kQ6AEwAHoECAAQAQg#v=onepage&q&f=false> (Cambridge University Press, Cambridge, England, UK, 2004).
- Gajdoš, M., Hummer, K., Kresse, G., Furthmüller, J. & Bechstedt, F. Linear optical properties in the projector-augmented wave methodology. *Phys. Rev. B* **73**, 45112, <https://doi.org/10.1103/PhysRevB.73.045112> (2006).
- Bader, R. F. W. *Atoms in Molecules: A Quantum Theory (International Series of Monographs on Chemistry (22))*. <https://www.amazon.com/Atoms-Molecules-International-Monographs-Chemistry/dp/0198558651> (Clarendon Press, 1994).
- Popelier, P. L. A. A fast algorithm to compute atomic charges based on the topology of the electron density. *Theor. Chem. Acc.* **105**, 393–399 (2001).
- Otero-de-la Roza, A., Johnson, E. R. & Luanña, V. Critic2: A program for real-space analysis of quantum chemical interactions in solids. *Comput. Phys. Commun.* **185**, 1007–1018 (2014).
- Rong, Z., Kitchaev, D., Canepa, P., Huang, W. & Ceder, G. An efficient algorithm for finding the minimum energy path for cation migration in ionic materials. *J. Chem. Phys.* **145**, 074112 (2016).
- Kahle, L., Marcolongo, A. & Marzari, N. Modeling lithium-ion solid-state electrolytes with a pinball model. *Phys. Rev. Mater.* **2**, 065405 (2018).
- Shen, J.-X., Horton, M. & Persson, K. A. A charge-density-based general cation insertion algorithm for generating new Li-ion cathode materials. *npj Comput. Mater.* **6**, 1–7 (2020).
- Kajita, S., Ohba, N., Jinnouchi, R. & Asahi, R. A Universal 3D Voxel Descriptor for Solid-State Material Informatics with Deep Convolutional Neural Networks. *Sci. Rep.* **7**, 1–9 (2017).
- Freysoldt, C. *et al.* First-principles calculations for point defects in solids. *Rev. Mod. Phys.* **86**, 253–305, <https://doi.org/10.1103/RevModPhys.86.253> (2014).
- Curtarolo, S. *et al.* AFLOW: An automatic framework for high-throughput materials discovery. *Comput. Mater. Sci.* **58**, 218–226 (2012).
- Draxl, C. & Scheffler, M. The NOMAD laboratory: from data sharing to artificial intelligence. *J. Phys.: Mater.* **2**, 036001 (2019).
- Wang, F. Q., Choudhary, K., Liu, Y., Hu, J. & Hu, M. Large scale dataset of real space electronic charge density of cubic inorganic materials from density functional theory (DFT) calculations. *Sci. Data* **9**, 1–9 (2022).
- Mathew, K. *et al.* Atomate: A high-level interface to generate, execute, and analyze computational materials science workflows. *Comput. Mater. Sci.* **139**, 140–152 (2017).
- Jain, A. *et al.* The Materials Project: A materials genome approach to accelerating materials innovation. *APL Materials* **1**, 011002 <http://link.aip.org/link/AMPADS/v1/i1/p011002/s1&Agg=doi> (2013).
- Jain, A. *et al.* Formation enthalpies by mixing GGA and GGA + *U* calculations. *Phys. Rev. B* **84**, 045115 (2011).
- Russell, F. P., Wilkinson, K. A., Kelly, P. H. J. & Skylaris, C.-K. Optimised three-dimensional Fourier interpolation: An analysis of techniques and application to a linear-scaling density functional theory code. *Comput. Phys. Commun.* **187**, 8–19 (2015).
- Leeper, T. J. AWS S3 Client Package [R package aws.s3 version 0.3.3] <https://cran.microsoft.com/snapshot/2017-06-26/web/packages/aws.s3/index.html> (2017).
- Materials project charge densities data-set. Lawrence Berkeley National Laboratory (LBNL) <https://doi.org/10.17188/1833409> (2021).
- Pyrho Validation - Check re-gridded periodic data, *Figshare* <https://doi.org/10.6084/m9.figshare.19908193> (2022).
- Kluyver, T. *et al.* Jupyter notebooks—a publishing format for reproducible computational workflows. In Loizides, F. & Schmidt, B. (eds.) *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, 87–90 (IOS Press, 2016).

Acknowledgements

This work was supported by the US Department of Energy, Office of Science, Office of Basic Energy Sciences, Materials Sciences and Engineering Division under contract no. DE-AC02-05-CH11231 (Materials Project program KC23MP).

Author contributions

J.X.S. developed the re-gridding analysis software; J.X.S. and S.D. developed the back-end API and J.M.M. front-end API; J.M.M. performed the DFT calculations that produced the charge densities; J.X.S., J.M.M., M.K.H., P.H. and S.D. also participated in aggregating, ingesting and maintaining the data at different stages. K.A.P. was responsible for supervising and advising the project at all stages.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to K.A.P.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2022