Utilizing SciPy and other open source packages to provide a powerful API for materials manipulation in the Schrödinger Materials Suite

Alexandr Fonari‡*, Farshad Fallah‡, Michael Rauch‡

Abstract—The use of several open source scientific packages in the Schrödinger Materials Science Suite will be discussed. A typical workflow for materials discovery will be described, discussing how open source packages have been incorporated at every stage. Some recent implementations of machine learning for materials discovery will be discussed, as well as how open source packages were leveraged to achieve results faster and more efficiently.

Index Terms—materials, active learning, OLED, deposition, evaporation

Introduction

A common materials discovery practice or workflow is to start with reading an experimental structure of a material or generating a structure in silico, computing its properties of interest (e.g. elastic constants, electrical conductivity), tuning the material by modifying its structure (e.g. doping) or adding and removing atoms (deposition, evaporation), and then recomputing the properties of the modified material (Figure 1). Computational materials discovery leverages such workflows to empower researchers to explore vast design spaces and uncover root causes without (or in conjunction with) laboratory experimentation.

Software tools for computational materials discovery can be facilitated by utilizing existing libraries that cover the fundamental mathematics used in the calculations in an optimized fashion. This use of existing libraries allows developers to devote more time to developing new features instead of re-inventing established methods. As a result, such a complementary approach improves the performance of computational materials software and reduces overall maintenance.

The Schrödinger Materials Science Suite [LLC22] is a proprietary computational chemistry/physics platform that streamlines materials discovery workflows into a single graphical user interface (Materials Science Maestro). The interface is a single portal for structure building and enumeration, physics-based modeling and machine learning, visualization and analysis. Tying together the various modules are a wide variety of scientific packages, some of which are proprietary to Schrödinger, Inc., some of which are open-source and many of which blend the two to optimize capabilities and efficiency. For example, the main simulation engine for molecular quantum mechanics is the Jaguar [BHH+13] proprietary code. The proprietary classical molecular dynamics code Desmond (distributed by Schrödinger, Inc.) [SGB+14] is used to obtain physical properties of soft materials, surfaces and polymers. For periodic quantum mechanics, the main simulation engine is the open source code Quantum ESPRESSO (QE) [GAB+17]. One of the co-authors of this proceedings (A. Fonari) contributes to the QE code in order to make integration with the Materials Suite more seamless and less error-prone. As part of this integration, support for using the portable `XML` format for input and output in QE has been implemented in the open source Python package qeschema [BDBF].

Figure 2 gives an overview of some of the various products that compose the Schrödinger Materials Science Suite. The various workflows are implemented mainly in Python (some of them described below), calling on proprietary or open-source code where appropriate, to improve the performance of the software and reduce overall maintenance.

The materials discovery cycle can be run in a high-throughput manner, enumerating different structure modifications in a systematic fashion, such as doping ratio in a semiconductor or depositing different adsorbates. As we will detail herein, there are several open source packages that allow the user to generate a large number of structures, run calculations in high throughput manner and analyze the results. For example, the open source package pymatgen [ORJ+13] facilitates generation and analysis of periodic structures. It can generate inputs for and read outputs of QE, the commercial codes VASP and Gaussian, and several other formats. To run and manage workflow jobs in a high-throughput manner, open source packages such as Custodian [ORJ+13] and AiiDA [HZU+20] can be used.

Materials import and generation

For reading and writing of material structures, several open source packages (e.g. OpenBabel [OBJ+11], RDKit [LTK+22]) have implemented functionality for working with several commonly used formats (e.g. CIF, PDB, mol, xyz). Periodic structures of materials, mainly coming from single crystal X-ray/neutron diffraction experiments, are distributed in CIF (Crystallographic Information File), PDB (Protein Data Bank) and lately mmCIF

* *Corresponding author: sasha.fonari@schrodinger.com*
‡ *Schrödinger Inc., 1540 Broadway, 24th Floor. New York, NY 10036*

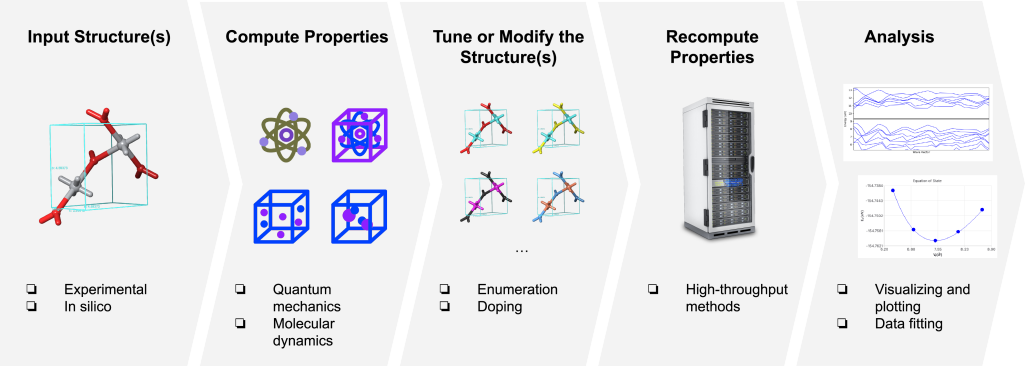

Fig. 1: *Example of a workflow for computational materials discovery.*

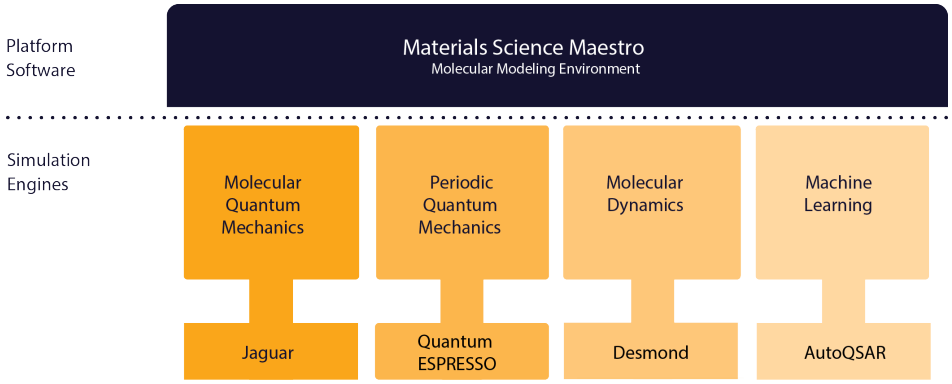

Fig. 2: *Some example products that compose the Schrödinger Materials Science Suite.*

formats [WF05]. Correctly reading experimental structures is of significant importance, since the rest of the materials discovery workflow depends on it. In addition to atom coordinates and periodic cell information, structural data also contains symmetry operations (listed explicitly or by the means of providing a space group) that can be used to decrease the number of computations required for a particular system by accounting for symmetry. This can be important, especially when scaling high-throughput calculations. From file, structure is read in a structure object through which atomic coordinates (as a NumPy array) and chemical information of the material can be accessed and updated. Structure object is similar to the one implemented in open source packages such as pymatgen [ORJ+13] and ASE [LMB+17]. All the structure manipulations during the workflows are done by using structure object interface (see structure deformation example below). Example of Structure object definition in pymatgen:

```
class Structure:

    def __init__(self, lattice, species, coords, ...):
        """Create a periodic structure."""
```

One consideration of note is that PDB, CIF and mmCIF structure formats allow description of the positional disorder (for example, a solvent molecule without a stable position within the cell which can be described by multiple sets of coordinates). Another complication is that experimental data spans an interval of almost a century: one of the oldest crystal structures deposited in the Cambridge Structural Database (CSD) [GBLW16] dates to 1924 [HM24]. These nuances and others present nontrivial technical challenges for developers. Thus, it has been a continuous effort by Schrödinger, Inc. (at least 39 commits and several weeks of work went into this project) and others to correctly read and convert periodic structures in OpenBabel. By version 3.1.1 (the most recent at writing time), the authors are not aware of any structures read incorrectly by OpenBabel. In general, non-periodic molecular formats are simpler to handle because they only contain atom coordinates but no cell or symmetry information. OpenBabel has Python bindings but due to the GPL license limitation, it is called as a subprocess from the Schrödinger Materials Suite.

Another important consideration in structure generation is modeling of substitutional disorder in solid alloys and materials with point defects (intermetallics, semiconductors, oxides and their crystalline surfaces). In such cases, the unit cell and atomic sites of the crystal or surface slab are well defined while the chemical species occupying the site may vary. In order to simulate substitutional disorder, one must generate the ensemble of structures that includes all statistically significant atomic distributions in a given unit cell. This can be achieved by a brute force enumeration of all symmetrically unique atomic structures with a given number of vacancies, impurities or solute atoms. The open source library enumlib [HF08] implements algorithms for such a systematic enumeration of periodic structures. The enumlib package consists of several Fortran binaries and Python scripts that can be run as a subprocess (no Python bindings). This allows the user to generate a large set of symmetrically nonequivalent materials with different compositions (e.g. doping or defect concentration).

Recently, we applied this approach in simultaneous study of the activity and stability of Pt based core-shell type catalysts for the oxygen reduction reaction [MGF+19]. We generated a set of stable doped Pt/transition metal/nitrogen surfaces using periodic enumeration. Using QE to perform periodic density functional

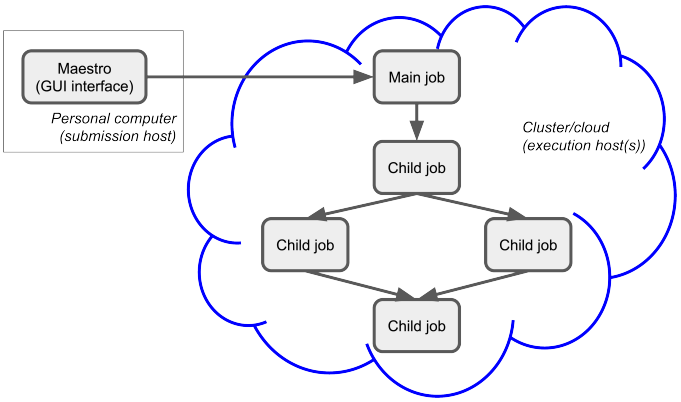

Fig. 3: *Example of the job submission process.*

theory (DFT) calculations, we assessed surface phase diagrams for Pt alloys and identified the avenues for stabilizing the cost effective core-shell systems by a judicious choice of the catalyst core material. Such catalysts may prove critical in electrocatalysis for fuel cell applications.

Workflow capabilities

In the last section, we briefly described a complete workflow from structure generation and enumeration to periodic DFT calculations to analysis. In order to be able to run a massively parallel screening of materials, a highly scalable and stable queuing system (job scheduler) is required. We have implemented a job queuing system on top of the most used queuing systems (LSF, PBS, SGE, SLURM, TORQUE, UGE) and exposed a Python API to submit and monitor jobs. In line with technological advancements, cloud is also supported by means of a virtual cluster configured with SLURM. This allows the user to submit a large number of jobs, limited only by SLURM scheduling capabilities and cloud resources. In order to accommodate job dependencies in workflows, for each job, a parent job (or multiple parent jobs) can be defined forming a directed graph of jobs (Figure 3).

There could be several reasons for a job to fail. Depending on the reason of failure, there are several restart and recovery mechanisms in place. The lowest level is the restart mechanism (in SLURM it is called `requeue`) which is performed by the queuing system itself. This is triggered when a node goes down. On the cloud, preemptible instances (nodes) can go offline at any moment. In addition, workflows implemented in the proprietary Schrödinger Materials Science Suite have built-in methods for handling various types of failure. For example, if the simulation is not converging to a requested energy accuracy, it is wasteful to blindly restart the calculation without changing some input parameters. However, in the case of a failure due to full disk space, it is reasonable to try restart with hopes to get a node with more empty disk space. If a job fails (and cannot be restarted), all its children (if any) will not start, thus saving queuing and computational time.

Having developed robust systems for running calculations, job queuing and troubleshooting (autonomously, when applicable), the developed workflows have allowed us and our customers to perform massive screenings of materials and their properties. For example, we reported a massive screening of 250,000 charge-conducting organic materials, totaling approximately 3,619,000 DFT SCF (self-consistent field) single-molecule calculations using Jaguar that took 457,265 CPU hours (~52 years) [MAS+20]. Another similar case study is the high-throughput molecular dynamics simulations (MD) of thermophysical properties of polymers for various applications [ABG+21]. There, using Desmond we computed the glass transition temperature (T_g) of 315 polymers and compared the results with experimental measurements [Bic02]. This study took advantage of GPU (graphics processing unit) support as implemented in Desmond, as well as the job scheduler API described above.

Other workflows implemented in the Schrödinger Materials Science Suite utilize open source packages as well. For soft materials (polymers, organic small molecules and substrates composed of soft molecules), convex hull and related mathematical methods are important for finding possible accessible solvent voids (during submerging or sorption) and adsorbate sites (during molecular deposition). These methods are conveniently implemented in the open source SciPy [VGO+20] and NumPy [HMvdW+20] packages. Thus, we implemented molecular deposition and evaporation workflows by using the Desmond MD engine as the backend in tandem with the convex hull functionality. This workflow enables simulation of the deposition and evaporation of the small molecules on a substrate. We utilized the aforementioned deposition workflow in the study of organic light-emitting diodes (OLEDs), which are fabricated using a stepwise process, where new layers are deposited on top of previous layers. Both vacuum and solution deposition processes have been used to prepare these films, primarily as amorphous thin film active layers lacking long-range order. Each of these deposition techniques introduces changes to the film structure and consequently, different charge-transfer and luminescent properties [WKB+22].

As can be seen from above, a workflow is usually some sort of structure modification through the structure object with a subsequent call to a backend code and analysis of its output if it succeeds. Input for the next iteration depends on the output of the previous iteration in some workflows. Due to the large chemical and manipulation space of the materials, sometimes it very tricky to keep code for all workflows follow the same code logic. For every workflow and/or functionality in the Materials Science Suite, some sort of peer reviewed material (publication, conference presentation) is created where implemented algorithms are described to facilitate reproducibility.

Data fitting algorithms and use cases

Materials simulation engines for QM, periodic DFT, and classical MD (referred to herein as backends) are frequently written in compiled languages with enabled parallelization for CPU or GPU hardware. These backends are called from Python workflows using the job queuing systems described above. Meanwhile, packages such as SciPy and NumPy provide sophisticated numerical function optimization and fitting capabilities. Here, we describe examples of how the Schrödinger suite can be used to combine materials simulations with popular optimization routines in the SciPy ecosystem.

Recently we implemented convex analysis of the stress strain curve (as described here [PKD18]). `scipy.optimize.minimize` is used for a constrained minimization with boundary conditions of a function related to the stress strain curve. The stress strain curve is obtained from a series of MD simulations on deformed cells (cell deformations are defined by strain type and deformation step). The pressure

tensor of a deformed cell is related to stress. This analysis allowed prediction of elongation at yield for high density polyethylene polymer. Figure 4 shows obtained calculated yield of 10% vs. experimental value within 9-18% range [BAS+20].

The `scipy.optimize` package is used for a least-squares fit of the bulk energies at different cell volumes (compressed and expanded) in order to obtain the bulk modulus and equation of state (EOS) of a material. In the Schrödinger suite this was implemented as a part of an EOS workflow, in which fitting is performed on the results obtained from a series of QE calculations performed on the original as well as compressed and expanded (deformed) cells. An example of deformation applied to a structure in pymatgen:

```
from pymatgen.analysis.elasticity import strain
from pymatgen.core import lattice
from pymatgen.core import structure

deform = strain.Deformation([
    [1.0, 0.02, 0.02],
    [0.0, 1.0, 0.0],
    [0.0, 0.0, 1.0]])

latt = lattice.Lattice([
    [3.84, 0.00, 0.00],
    [1.92, 3.326, 0.00],
    [0.00, -2.22, 3.14],
])

st = structure.Structure(
    latt,
    ["Si", "Si"],
    [[0, 0, 0], [0.75, 0.5, 0.75]])

strained_st = deform.apply_to_structure(st)
```

This is also an example of loosely coupled (embarrassingly parallel) jobs. In particular, calculations of the deformed cells only depend on the bulk calculation and do not depend on each other. Thus, all the deformation jobs can be submitted in parallel, facilitating high-throughput runs.

Structure refinement from powder diffraction experiment is another example where more complex optimization is used. Powder diffraction is a widely used method in drug discovery to assess purity of the material and discover known or unknown crystal polymorphs [KBD+21]. In particular, there is interest in fitting of the experimental powder diffraction intensity peaks to the indexed peaks (Pawley refinement) [JPS92]. Here we employed the open source lmfit package [NSA+16] to perform a minimization of the multivariable Voigt-like function that represents the entire diffraction spectrum. This allows the user to refine (optimize) unit cell parameters coming from the indexing data and as the result, goodness of fit (*R*-factor) between experimental and simulated spectrum is minimized.

Machine learning techniques

Of late, there is great interest in machine learning assisted materials discovery. There are several components required to perform machine learning assisted materials discovery. In order to train a model, benchmark data from simulation and/or experimental data is required. Besides benchmark data, computation of the relevant descriptors is required (see below). Finally, a model based on benchmark data and descriptors is generated that allows prediction of properties for novel materials. There are several techniques to generate the model, such as linear or non-linear fitting to neural networks. Tools include the open source DeepChem [REW+19] and AutoQSAR [DDS+16] from the Schrödinger suite. Depending on the type of materials, benchmark data can be obtained using different codes available in the Schrödinger suite:

- small molecules and finite systems - Jaguar
- periodic systems - Quantum ESPRESSO
- larger polymeric and similar systems - Desmond

Different materials systems require different descriptors for featurization. For example, for crystalline periodic systems, we have implemented several sets of tailored descriptors. Generation of these descriptors again uses a mix of open source and Schrödinger proprietary tools. Specifically:

- elemental features such as atomic weight, number of valence electrons in *s*, *p* and *d*-shells, and electronegativity
- structural features such as density, volume per atom, and packing fraction descriptors implemented in the open source matminer package [WDF+18]
- intercalation descriptors such as cation and anion counts, crystal packing fraction, and average neighbor ionicity [SYC+17] implemented in the Schrödinger suite
- three-dimensional smooth overlap of atomic positions (SOAP) descriptors implemented in the open source DScribe package [HJM+20].

We are currently training models that use these descriptors to predict properties, such as bulk modulus, of a set of Li-containing battery related compounds [Cha]. Several models will be compared, such as kernel regression methods (as implemented in the open source scikit-learn code [PVG+11]) and AutoQSAR.

For isolated small molecules and extended non-periodic systems, RDKit can be used to generate a large number of atomic and molecular descriptors. A lot of effort has been devoted to ensure that RDKit can be used on a wide variety of materials that are supported by the Schrödinger suite. At the time of writing, the 4th most active contributor to RDKit is Ricardo Rodriguez-Schmidt from Schrödinger [RDK].

Recently, active learning (AL) combined with DFT has received much attention to address the challenge of leveraging exhaustive libraries in materials informatics [VPB21], [SPA+19]. On our side, we have implemented a workflow that employs active learning (AL) for intelligent and iterative identification of promising materials candidates within a large dataset. In the framework of AL, the predicted value with associated uncertainty is considered to decide what materials to be added in each iteration, aiming to improve the model performance in the next iteration (Figure 5).

Since it could be important to consider multiple properties simultaneously in material discovery, multiple property optimization (MPO) has also been implemented as a part of the AL workflow [KAG+22]. MPO allows scaling and combining multiple properties into a single score. We employed the AL workflow to determine the top candidates for hole (positively charged carrier) transport layer (HTL) by evaluating 550 molecules in 10 iterations using DFT calculations for a dataset of ~9,000 molecules [AKA+22]. Resulting model was validated by randomly picking a molecule from the dataset, computing properties with DFT and comparing those to the predicted values. According to the semi-classical Marcus equation [Mar93], high rates of hole transfer are inversely proportional to hole reorganization energies. Thus, MPO scores were computed based on minimizing hole reorganization energy and targeting oxidation potential to an appropriate level to ensure a low energy barrier for hole injection from the anode

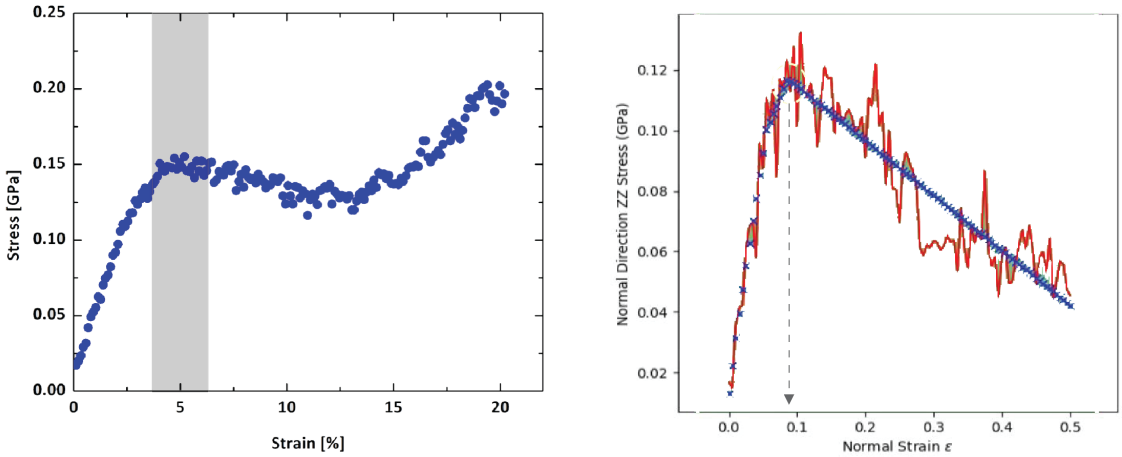

Fig. 4: *Left: The uniaxial stress/strain curve of a polymer calculated using Desmond through the stress strain workflow. The dark grey band indicates an inflection that marks the yield point. Right: Constant strain simulation with convex analysis indicates elongation at yield. The red curve shows simulated stress versus strain. The blue curve shows convex analysis.*

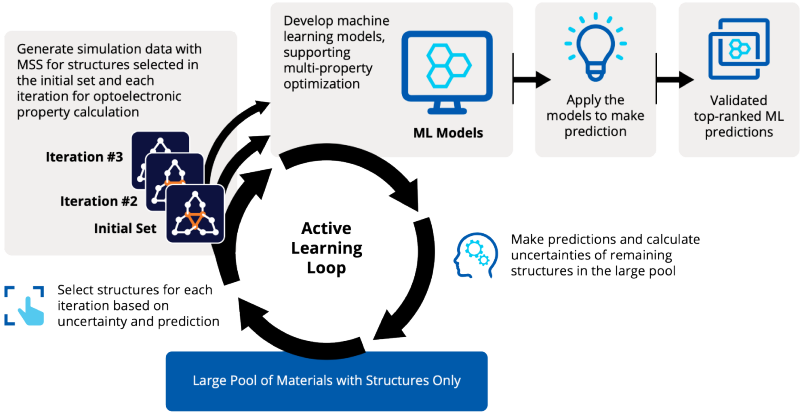

Fig. 5: *Active learning workflow for the design and discovery of novel optoelectronics molecules.*

into the emissive layer. In this workflow, we used RDKit to compute descriptors for the chemical structures. These descriptors generated on the initial subset of structures are given as vectors to an algorithm based on Random Forest Regressor as implemented in scikit-learn. Bayesian optimization is employed to tune the hyperparameters of the model. In each iteration, a trained model is applied for making predictions on the remaining materials in the dataset. Figure 6 (A) displays MPO scores for the HTL dataset estimated by AL as a function of hole reorganization energies that are separately calculated for all the materials. This figure indicates that there are many materials in the dataset with desired low hole reorganization energies but are not suitable for HTL due to their improper oxidation potentials, suggesting that MPO is important to evaluate the optoelectronic performance of the materials. Figure 6 (B) presents MPO scores of the materials used in the training dataset of AL, demonstrating that the feedback loop in the AL workflow efficiently guides the data collection as the size of the training set increases.

To appreciate the computational efficiency of such an approach, it is worth noting that performing DFT calculations for all of the 9,000 molecules in the dataset would increase the computational cost by a factor of 15 versus the AL workflow. It seems that AL approach can be useful in the cases where problem space is broad (like chemical space), but there are many clusters of similar items (similar molecules). In this case, benchmark data is only needed for few representatives of each cluster. We are currently working on applying this approach to train models for predicting physical properties of soft materials (polymers).

Conclusions

We present several examples of how Schrödinger Materials Suite integrates open source software packages. There is a wide range of applications in materials science that can benefit from already existing open source code. Where possible, we report issues to the package authors and submit improvements and bug fixes in the form of the pull requests. We are thankful to all who have contributed to open source libraries, and have made it possible for us to develop a platform for accelerating innovation in materials and drug discovery. We will continue contributing to these projects and we hope to further give back to the scientific community by facilitating research in both academia and industry. We hope that this report will inspire other scientific companies to give back to the open source community in order to improve the computational materials field and make science more reproducible.

Acknowledgments

The authors acknowledge Bradley Dice and Wenduo Zhou for their valuable comments during the review of the manuscript.

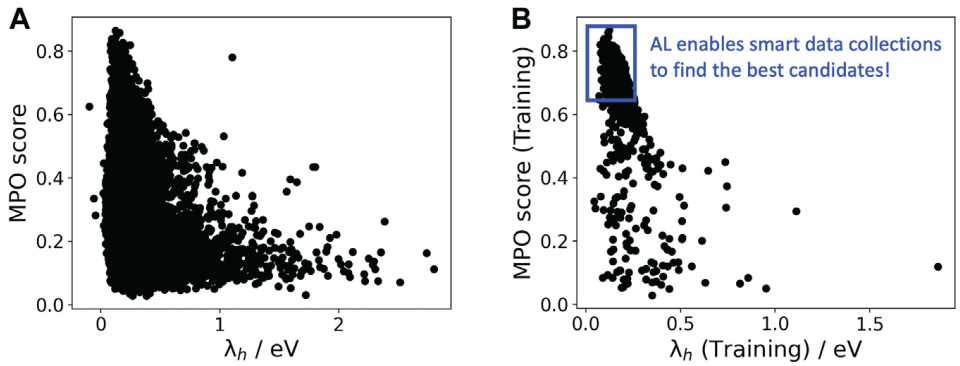

Fig. 6: *A: MPO score of all materials in the HTL dataset. B: Those used in the training set as a function of the hole reorganization energy (λ_h).*

REFERENCES

[ABG+21] Mohammad Atif Faiz Afzal, Andrea R. Browning, Alexander Goldberg, Mathew D. Halls, Jacob L. Gavartin, Tsuguo Morisato, Thomas F. Hughes, David J. Giesen, and Joseph E. Goose. High-throughput molecular dynamics simulations and validation of thermophysical properties of polymers for various applications. *ACS Applied Polymer Materials*, 3, 2021. doi:10.1021/acsapm.0c00524.

[AKA+22] Hadi Abroshan, H. Shaun Kwak, Yuling An, Christopher Brown, Anand Chandrasekaran, Paul Winget, and Mathew D. Halls. Active learning accelerates design and optimization of hole-transporting materials for organic electronics. *Frontiers in Chemistry*, 9, 2022. doi:10.3389/fchem.2021.800371.

[BAS+20] A. R. Browning, M. A. F. Afzal, J. Sanders, A. Goldberg, A. Chandrasekaran, and H. S. Kwak. Polyolefin molecular simulation for critical physical characteristics. *International Polyolefins Conference*, 2020.

[BDBF] Davide Brunato, Pietro Delugas, Giovanni Borghi, and Alexandr Fonari. qeschema. URL: https://github.com/QEF/qeschema.

[BHH+13] Art D. Bochevarov, Edward Harder, Thomas F. Hughes, Jeremy R. Greenwood, Dale A. Braden, Dean M. Philipp, David Rinaldo, Mathew D. Halls, Jing Zhang, and Richard A. Friesner. Jaguar: A high-performance quantum chemistry software program with strengths in life and materials sciences. *International Journal of Quantum Chemistry*, 113, 2013. doi:10.1002/qua.24481.

[Bic02] Jozef Bicerano. *Prediction of polymer properties*. cRc Press, 2002.

[Cha] A. Chandrasekaran. Active learning accelerated design of ionic materials. *in progress*.

[DDS+16] Steven L. Dixon, Jianxin Duan, Ethan Smith, Christopher D. Von Bargen, Woody Sherman, and Matthew P. Repasky. Autoqsar: An automated machine learning tool for best-practice quantitative structure-activity relationship modeling. *Future Medicinal Chemistry*, 8, 2016. doi:10.4155/fmc-2016-0093.

[GAB+17] P. Giannozzi, O. Andreussi, T. Brumme, O. Bunau, M. Buongiorno Nardelli, M. Calandra, R. Car, C. Cavazzoni, D. Ceresoli, M. Cococcioni, N. Colonna, I. Carnimeo, A. Dal Corso, S. De Gironcoli, P. Delugas, R. A. Distasio, A. Ferretti, A. Floris, G. Fratesi, G. Fugallo, R. Gebauer, U. Gerstmann, F. Giustino, T. Gorni, J. Jia, M. Kawamura, H. Y. Ko, A. Kokalj, E. Kücükbenli, M. Lazzeri, M. Marsili, N. Marzari, F. Mauri, N. L. Nguyen, H. V. Nguyen, A. Otero-De-La-Roza, L. Paulatto, S. Poncé, D. Rocca, R. Sabatini, B. Santra, M. Schlipf, A. P. Seitsonen, A. Smogunov, I. Timrov, T. Thonhauser, P. Umari, N. Vast, X. Wu, and S. Baroni. Advanced capabilities for materials modelling with quantum espresso. *Journal of Physics Condensed Matter*, 29, 2017. URL: https://www.quantum-espresso.org/, doi:10.1088/1361-648X/aa8f79.

[GBLW16] Colin R. Groom, Ian J. Bruno, Matthew P. Lightfoot, and Suzanna C. Ward. The cambridge structural database. *Acta Crystallographica Section B: Structural Science, Crystal Engineering and Materials*, 72, 2016. doi:10.1107/S2052520616003954.

[HF08] Gus L.W. Hart and Rodney W. Forcade. Algorithm for generating derivative structures. *Physical Review B - Condensed Matter and Materials Physics*, 77, 2008. URL: https://github.com/msg-byu/enumlib/, doi:10.1103/PhysRevB.77.224115.

[HJM+20] Lauri Himanen, Marc O.J. Jager, Eiaki V. Morooka, Filippo Federici Canova, Yashasvi S. Ranawat, David Z. Gao, Patrick Rinke, and Adam S. Foster. Dscribe: Library of descriptors for machine learning in materials science. *Computer Physics Communications*, 247, 2020. URL: https://singroup.github.io/dscribe/latest/, doi:10.1016/j.cpc.2019.106949.

[HM24] O Hassel and H Mark. The crystal structure of graphite. *Physik. Z*, 25:317–337, 1924.

[HMvdW+20] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with numpy, 2020. URL: https://numpy.org/, doi:10.1038/s41586-020-2649-2.

[HZU+20] Sebastiaan P. Huber, Spyros Zoupanos, Martin Uhrin, Leopold Talirz, Leonid Kahle, Rico Hauselmann, Dominik Gresch, Tiziano Müller, Aliaksandr V. Yakutovich, Casper W. Andersen, Francisco F. Ramirez, Carl S. Adorf, Fernando Gargiulo, Snehal Kumbhar, Elsa Passaro, Conrad Johnston, Andrius Merkys, Andrea Cepellotti, Nicolas Mounet, Nicola Marzari, Boris Kozinsky, and Giovanni Pizzi. Aiida 1.0, a scalable computational infrastructure for automated reproducible workflows and data provenance. *Scientific Data*, 7, 2020. URL: https://www.aiida.net/, doi:10.1038/s41597-020-00638-4.

[JPS92] J. Jansen, R. Peschar, and H. Schenk. Determination of accurate intensities from powder diffraction data. i. whole-pattern fitting with a least-squares procedure. *Journal of Applied Crystallography*, 25, 1992. doi:10.1107/S0021889891012104.

[KAG+22] H. Shaun Kwak, Yuling An, David J. Giesen, Thomas F. Hughes, Christopher T. Brown, Karl Leswing, Hadi Abroshan, and Mathew D. Halls. Design of organic electronic materials with a goal-directed generative model powered by deep neural networks and high-throughput molecular simulations. *Frontiers in Chemistry*, 9, 2022. doi:10.3389/fchem.2021.800370.

[KBD+21] James A Kaduk, Simon J L Billinge, Robert E Dinnebier, Nathan Henderson, Ian Madsen, Radovan Černý, Matteo Leoni, Luca Lutterotti, Seema Thakral, and Daniel Chateigner. Powder diffraction. *Nature Reviews Methods Primers*, 1:77, 2021. URL: https://doi.org/10.1038/s43586-021-00074-7, doi:10.1038/s43586-021-00074-7.

[LLC22] Schrödinger LLC. Schrödinger release 2022-2: Materials science suite, 2022. URL: https://www.schrodinger.com/platform/materials-science.

[LMB+17] Ask Hjorth Larsen, Jens JØrgen Mortensen, Jakob Blomqvist, Ivano E. Castelli, Rune Christensen, Marcin Dułak, Jesper Friis, Michael N. Groves, BjØrk Hammer, Cory Hargus, Eric D. Hermes, Paul C. Jennings, Peter Bjerre Jensen, James Kermode, John R. Kitchin, Esben Leonhard Kolsbjerg, Joseph Kubal, Kristen Kaasbjerg, Steen Lysgaard, Jón Bergmann Maronsson, Tristan Maxson, Thomas Olsen, Lars Pastewka, Andrew Peterson, Carsten Rostgaard, Jakob SchiØtz, Ole Schütt, Mikkel Strange, Kristian S. Thygesen, Tejs Vegge, Lasse Vilhelmsen, Michael Walter, Zhenhua Zeng, and Karsten W. Jacobsen. The atomic simulation environment - a python library for working with atoms, 2017. URL: https://wiki.fysik.dtu.dk/ase/, doi:10.1088/1361-648X/aa680e.

[LTK+22] Greg Landrum, Paolo Tosco, Brian Kelley, Ric, sriniker, gedeck, Riccardo Vianello, NadineSchneider, Eisuke Kawashima, Andrew Dalke, Dan N, David Cosgrove, Gareth Jones, Brian Cole, Matt Swain, Samo Turk, AlexanderSavelyev, Alain Vaucher, Maciej Wójcikowski, Ichiru Take, Daniel Probst, Kazuya Ujihara, Vincent F. Scalfani, guillaume godin, Axel Pahl, Francois Berenger, JLVarjo, strets123, JP, and DoliathGavid. rdkit. 6 2022. URL: https://rdkit.org/, doi:10.5281/ZENODO.6605135.

[Mar93] Rudolph A. Marcus. Electron transfer reactions in chemistry. theory and experiment. *Reviews of Modern Physics*, 65, 1993. doi:10.1103/RevModPhys.65.599.

[MAS+20] Nobuyuki N. Matsuzawa, Hideyuki Arai, Masaru Sasago, Eiji Fujii, Alexander Goldberg, Thomas J. Mustard, H. Shaun Kwak, David J. Giesen, Fabio Ranalli, and Mathew D. Halls. Massive theoretical screen of hole conducting organic materials in the heteroacene family by using a cloud-computing environment. *Journal of Physical Chemistry A*, 124, 2020. doi:10.1021/acs.jpca.9b10998.

[MGF+19] Thomas Mustard, Jacob Gavartin, Alexandr Fonari, Caroline Krauter, Alexander Goldberg, H Kwak, Tsuguo Morisato, Sudharsan Pandiyan, and Mathew Halls. Surface reactivity and stability of core-shell solid catalysts from ab initio combinatorial calculations. volume 258, 2019.

[NSA+16] Matthew Newville, Till Stensitzki, Daniel B Allen, Michal Rawlik, Antonino Ingargiola, and Andrew Nelson. Lmfit: Non-linear least-square minimization and curve-fitting for python. *Astrophysics Source Code Library*, page ascl–1606, 2016. URL: https://lmfit.github.io/lmfit-py/.

[OBJ+11] Noel M. O'Boyle, Michael Banck, Craig A. James, Chris Morley, Tim Vandermeersch, and Geoffrey R. Hutchison. Open babel: An open chemical toolbox. *Journal of Cheminformatics*, 3, 2011. URL: https://openbabel.org/, doi:10.1186/1758-2946-3-33.

[ORJ+13] Shyue Ping Ong, William Davidson Richards, Anubhav Jain, Geoffroy Hautier, Michael Kocher, Shreyas Cholia, Dan Gunter, Vincent L. Chevrier, Kristin A. Persson, and Gerbrand Ceder. Python materials genomics (pymatgen): A robust, open-source python library for materials analysis. *Computational Materials Science*, 68, 2013. URL: https://pymatgen.org/, doi:10.1016/j.commatsci.2012.10.028.

[PKD18] Paul N. Patrone, Anthony J. Kearsley, and Andrew M. Dienstfrey. The role of data analysis in uncertainty quantification: Case studies for materials modeling. volume 0, 2018. doi:10.2514/6.2018-0927.

[PVG+11] Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12, 2011. URL: https://scikit-learn.org/.

[RDK] Rdkit contributors. URL: https://github.com/rdkit/rdkit/graphs/contributors.

[REW+19] Bharath Ramsundar, Peter Eastman, Patrick Walters, Vijay Pande, Karl Leswing, and Zhenqin Wu. *Deep Learning for the Life Sciences*. O'Reilly Media, 2019. https://www.amazon.com/Deep-Learning-Life-Sciences-Microscopy/dp/1492039837.

[SGB+14] David E. Shaw, J. P. Grossman, Joseph A. Bank, Brannon Batson, J. Adam Butts, Jack C. Chao, Martin M. Deneroff, Ron O. Dror, Amos Even, Christopher H. Fenton, Anthony Forte, Joseph Gagliardo, Gennette Gill, Brian Greskamp, C. Richard Ho, Douglas J. Ierardi, Lev Iserovich, Jeffrey S. Kuskin, Richard H. Larson, Timothy Layman, Li Siang Lee, Adam K. Lerer, Chester Li, Daniel Killebrew, Kenneth M. Mackenzie, Shark Yeuk Hai Mok, Mark A. Moraes, Rolf Mueller, Lawrence J. Nociolo, Jon L. Peticolas, Terry Quan, Daniel Ramot, John K. Salmon, Daniele P. Scarpazza, U. Ben Schafer, Naseer Siddique, Christopher W. Snyder, Jochen Spengler, Ping Tak Peter Tang, Michael Theobald, Horia Toma, Brian Towles, Benjamin Vitale, Stanley C. Wang, and Cliff Young. Anton 2: Raising the bar for performance and programmability in a special-purpose molecular dynamics supercomputer. volume 2015-January, 2014. doi:10.1109/SC.2014.9.

[SPA+19] Gabriel R. Schleder, Antonio C.M. Padilha, Carlos Mera Acosta, Marcio Costa, and Adalberto Fazzio. From dft to machine learning: Recent approaches to materials science - a review. *JPhys Materials*, 2, 2019. doi:10.1088/2515-7639/ab084b.

[SYC+17] Austin D Sendek, Qian Yang, Ekin D Cubuk, Karel-Alexander N Duerloo, Yi Cui, and Evan J Reed. Holistic computational structure screening of more than 12000 candidates for solid lithium-ion conductor materials. *Energy and Environmental Science*, 10:306–320, 2017. doi:10.1039/c6ee02697d.

[VGO+20] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R.J. Nelson, Eric Jones, Robert Kern, Eric Larson, C. J. Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, Aditya Vijaykumar, Alessandro Pietro Bardelli, Alex Rothberg, Andreas Hilboll, Andreas Kloeckner, Anthony Scopatz, Antony Lee, Ariel Rokem, C. Nathan Woods, Chad Fulton, Charles Masson, Christian Haggström, Clark Fitzgerald, David A. Nicholson, David R. Hagen, Dmitrii V. Pasechnik, Emanuele Olivetti, Eric Martin, Eric Wieser, Fabrice Silva, Felix Lenders, Florian Wilhelm, G. Young, Gavin A. Price, Gert Ludwig Ingold, Gregory E. Allen, Gregory R. Lee, Hervé Audren, Irvin Probst, Jörg P. Dietrich, Jacob Silterra, James T. Webber, Janko Slavič, Joel Nothman, Johannes Buchner, Johannes Kulick, Johannes L. Schönberger, José Vinícius de Miranda Cardoso, Joscha Reimer, Joseph Harrington, Juan Luis Cano Rodríguez, Juan Nunez-Iglesias, Justin Kuczynski, Kevin Tritz, Martin Thoma, Matthew Newville, Matthias Kümmerer, Maximilian Bolingbroke, Michael Tartre, Mikhail Pak, Nathaniel J. Smith, Nikolai Nowaczyk, Nikolay Shebanov, Oleksandr Pavlyk, Per A. Brodtkorb, Perry Lee, Robert T. McGibbon, Roman Feldbauer, Sam Lewis, Sam Tygier, Scott Sievert, Sebastiano Vigna, Stefan Peterson, Surhud More, Tadeusz Pudlik, Takuya Oshima, Thomas J. Pingel, Thomas P. Robitaille, Thomas Spura, Thouis R. Jones, Tim Cera, Tim Leslie, Tiziano Zito, Tom Krauss, Utkarsh Upadhyay, Yaroslav O. Halchenko, and Yoshiki Vázquez-Baeza. Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature Methods*, 17, 2020. doi:10.1038/s41592-019-0686-2.

[VPB21] Rama Vasudevan, Ghanshyam Pilania, and Prasanna V. Balachandran. Machine learning for materials design and discovery. *Journal of Applied Physics*, 129, 2021. doi:10.1063/5.0043300.

[WDF+18] Logan Ward, Alexander Dunn, Alireza Faghaninia, Nils E.R. Zimmermann, Saurabh Bajaj, Qi Wang, Joseph Montoya, Jiming Chen, Kyle Bystrom, Maxwell Dylla, Kyle Chard, Mark Asta, Kristin A. Persson, G. Jeffrey Snyder, Ian Foster, and Anubhav Jain. Matminer: An open source toolkit for materials data mining. *Computational Materials Science*, 152, 2018. URL: https://hackingmaterials.lbl.gov/matminer/, doi:10.1016/j.commatsci.2018.05.018.

[WF05] John D. Westbrook and Paula M.D. Fitzgerald. The pdb format, mmcif formats, and other data formats, 2005. doi:10.1002/0471721204.ch8.

[WKB+22] Paul Winget, H. Shaun Kwak, Christopher T. Brown, Alexandr Fonari, Kevin Tran, Alexander Goldberg, Andrea R. Browning, and Mathew D. Halls. Organic thin films for oled applications: Influence of molecular structure, deposition method,

and deposition conditions. *International Conference on the Science and Technology of Synthetic Metals*, 2022.