Computational Materials Science 103 (2015) 56–67

Contents lists available at ScienceDirect

Computational Materials Science

journal homepage: www.elsevier.com/locate/commatsci

The Electrolyte Genome project: A big data approach in battery materials discovery

Xiaohui Qu [a], Anubhav Jain [a], Nav Nidhi Rajput [a], Lei Cheng [b], Yong Zhang [c], Shyue Ping Ong [d], Miriam Brafman [a], Edward Maginn [c], Larry A. Curtiss [b], Kristin A. Persson [a,*]

[a] Environmental Energy Technologies Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA
[b] Materials Science Division, Argonne National Laboratory, Argonne, IL 60439, USA
[c] Department of Chemical and Biomolecular Engineering, University of Notre Dame, Notre Dame, IN 46556, USA
[d] Department of NanoEngineering, University of California, San Diego, La Jolla, CA 92093, USA

ARTICLE INFO

Article history:
Received 19 November 2014
Received in revised form 24 February 2015
Accepted 26 February 2015
Available online 1 April 2015

Keywords:
High-throughput
Battery
DFT
Ionization potential
Electron affinity
IP/EA
Dissociation constants
Electrolyte

ABSTRACT

We present a high-throughput infrastructure for the automated calculation of molecular properties with a focus on battery electrolytes. The infrastructure is largely open-source and handles both practical aspects (input file generation, output file parsing, and information management) as well as more complex problems (structure matching, salt complex generation, and failure recovery). Using this infrastructure, we have computed the ionization potential (IP) and electron affinities (EA) of 4830 molecules relevant to battery electrolytes (encompassing almost 55,000 quantum mechanics calculations) at the B3LYP/6-31+G* level. We describe automated workflows for computing redox potential, dissociation constant, and salt-molecule binding complex structure generation. We present routines for automatic recovery from calculation errors, which brings the failure rate from 9.2% to 0.8% for the QChem DFT code. Automated algorithms to check duplication between two arbitrary molecules and structures are described. We present benchmark data on basis sets and functionals on the G2-97 test set; one finding is that a IP/EA calculation method that combines PBE geometry optimization and B3LYP energy evaluation requires less computational cost and yields nearly identical results as compared to a full B3LYP calculation, and could be suitable for the calculation of large molecules. Our data indicates that among the 8 functionals tested, XYGJ-OS and B3LYP are the two best functionals to predict IP/EA with an RMSE of 0.12 and 0.27 eV, respectively. Application of our automated workflow to a large set of quinoxaline derivative molecules shows that functional group effect and substitution position effect can be separated for IP/EA of quinoxaline derivatives, and the most sensitive position is different for IP and EA.

Published by Elsevier B.V.

1. Introduction

The development of high-performance computing and increasingly sophisticated quantum chemistry software is enabling a paradigm shift in material science whereby *ab initio* calculations can be performed on large chemical and structural spaces to aid and guide materials discovery research [1]. However, while the computing resources and available algorithms present a tremendous opportunity, they also bring new challenges. For example, how can one practically generate such large data sets such that the results are easy to query and analyze?

To address this challenge, software infrastructure beyond simple scripting has recently emerged within the computational materials science community. A few efforts in this category are noteworthy in that they use their capabilities to also provide either free data and/or open-source codes to the community as part of their mission and delivery goals. The Material Project (www.materialsproject.org) [2] provides open access to a user-friendly web interface, automated materials analysis codes, and workflow tools operating on a set of more than 50,000 calculated materials and their properties. Other examples include Harvard Clean Energy Project [3,4] which has focused on electrochemical windows for photovoltaic applications, AFLOWlib [5], which specializes in electronic structure, and the OQMD [6] which specializes in alloy stability data.

In this paper, we describe an effort to develop a high-throughput computational workflow and analysis code for multi-component liquid electrolytes within the framework of the Materials Project infrastructure. We initially focus on organic liquid electrolytes for next-generation energy storage solutions, although inorganic liquid

* Corresponding author.

http://dx.doi.org/10.1016/j.commatsci.2015.02.050
0927-0256/Published by Elsevier B.V.

electrolytes have many technological applications. Properties important for future battery electrolytes including wide electrochemical window, high ion conductivity, high solubility, chemical stability towards electrode components, low flammability, environmental friendliness, and low cost.

Many of these important electrolyte properties can now be calculated for targeted electrolyte systems, generally one at a time or in low-throughput. Particularly, electrochemical window calculations for electrolyte components have been pioneered by several other authors [7–13]. Our work is more closely related to two previous high-throughput studies of molecular systems: (1) Aspuru-Guzik and coworkers screened 2.3 million organic photovoltaic candidates using a hierarchical screening procedure, including chemo-informatics descriptors, semi-empirical quantum mechanical calculation and basic highest occupied molecular orbital/lowest unoccupied molecular orbital (HOMO/LUMO) density functional theory (DFT) calculations [3,4,14]. (2) Korth carried out a high-throughput screening of 11,000 organic solvent molecules using a combination of semi-empirical quantum mechanical and DFT calculations [15]. A major difference between our infrastructure and that of Aspuru-Guzik et al. and Korth is its emphasis on high-fidelity calculations of full DFT treatment of IP/EA including the effects of structural relaxation and frequency calculation. This approach emphasizes greater accuracy per calculation, in contrast to the semi-empirical or HOMO/LUMO-based descriptor screening of earlier works that are intended for quick screening of a large number of molecules.

The goal of our project (Electrolyte Genome) [16] is to ultimately address all chemical components present in the electrolyte as well as the interactions between them, including redox active molecules, solvent, salt, impurities and additives. In addition, we aim to eventually couple first-principles calculations with classical molecular dynamics simulations, enabling one to compute more complex properties such as solvation structure, solubility, and chemical and electrochemical stability.

2. Software framework and algorithms

The Materials Project [2] has developed very flexible modules for materials analysis, workflow management and error management. By leveraging these efforts, we add support for molecular properties analysis and the execution of QChem [17,18] to automate the electrolyte properties calculation/screening in the Electrolyte Genome project. This is done by extending 3 Python packages that have been developed in Materials Project and implementing 1 Python package from scratch: (1) *pymatgen* [19] handles input file generation, output file parsing and molecular comparisons – both chemical as well as structural. (2) *FireWorks* [20] handles job/workflow control and storage but is completely agnostic to the specifics of the code used, e.g. QChem [17,18], Gaussian [21] etc. (3) *custodian* monitors and applies recipe-like fixes to common errors within a calculation using "plug-ins" specific to the code used and (4) *rubicon* combines the 3 previous codebases to define the calculation workflows, database operations, property calculations, and job submission application programming interface (API). *pymatgen*, *FireWorks* and *custodian* are developed by the Materials Project, while *rubicon* is developed specifically for the Electrolyte Genome project. All codes are available at https://github.com/materialsproject.

At the backend, the infrastructure uses the QChem [17,18] quantum chemistry software package to perform the *ab initio* calculations. We have also built NWChem [22] and Gaussian [21] adapters into our codes. It is important to note that the calculations are made transparent by encapsulating the input file generation and output file parsing into a QChem I/O module. For example, the following code demonstrates how to submit an IP/EA (ionization potential/electron affinity) calculation for a water molecule defined in the SDF file format:

```
from pymatgen.core.structure import Molecule
from pymatgen.matproj.snl import StructureNL
from rubicon.submission.submission_mongo_eg
  import SubmissionMongoAdapterEG
from rubicon.utils.snl.egsnl_mongo import
  EGSNLMongoAdapter
mol = Molecule.from_file(''mols/h2o.sdf'')
snl = StructureNL(mol, ''Xiaohui Qu
  <xqu@lbl.gov>'', ''Electrolyte Genome'')
sma = SubmissionMongoAdapterEG.auto_load()
sma.submit_snl(snl,
  ''xqu@lbl.gov'',
  parameters={''priority'': 1,
    ''ref_charge'': 0,
    ''nick_name'': ''Water'',
    ''solvent'': ''thf'',
    ''solvent_method'': ''ief-pcm'',
    ''qm_method'': ''B3LYP/6-31+G*//PBE-D3/6-
  31+G*'',
    ''workflow'': ''ipea'',
    ''mission'': ''Test Simple Molecule''})
```

In this example, the coordinates of the molecule are loaded by *pymatgen*; many input formats are supported via the OpenBabel library. Next, metadata on the submission (e.g., author, project) are encapsulated within the StructureNL object representation. Finally, this object is submitted to a database along with higher-level instructions. Subsequently, the infrastructure will use the submitted data to map the molecule to a workflow and execute it over computing resources (see Fig. 1). The user can tune the calculation procedure by specifying certain parameters, for example, the initial charges, the solvent models or functionals, or by programming a custom workflow. Note that many parameters shown above are not mandatory and default values will be used if the corresponding parameter is not specified.

It is worth noting that our infrastructure code is designed for flexibility – both in terms of computing resources and workflow. It is not a simple one-shot script, which must finish all the calculations once launched. In contrast, our infrastructure is able to save intermediate results, execute seamless re-starts, by-pass selected steps and decouple the workflows to several executable sub steps. In addition, more advanced execution modes, such as packing many small molecule jobs over many nodes for supercomputing resources, are provided by using FireWorks [20] as the execution manager.

All inputs, outputs, and workflow related objects are stored in MongoDB [23], which is a document-based schema-less database. Thus, the user typically does not need to refer to flat files (e.g. output files) for analyses and data mining. In contrast to the more common SQL databases, MongoDB stores data as JSON-style [24] documents with a flexible schema. This makes it easy to extend the database to new data types by allowing objects to easily map to a database representation without the need for a separate layer as object-relational mapper. One can design complex data structures as well as store and query them in a simple and straightforward manner.

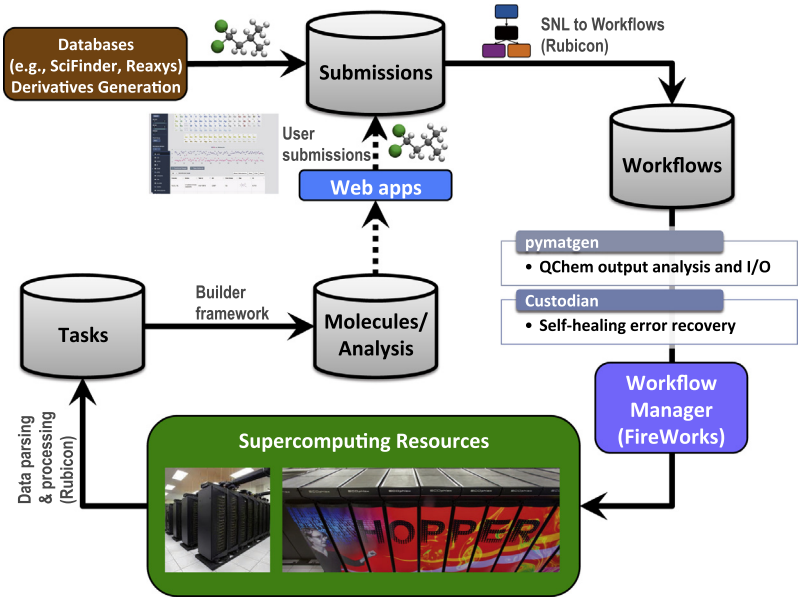


Fig. 1. The Electrolyte Genome project computation infrastructure. Submitted molecules (top-left) are mapped to workflows (top-right), and computed automatically over several supercomputing resources. The results are automatically parsed and put in several MongoDB collections. For more details, see the text.

At the time of writing, 4830 IP/EAs have been successfully calculated using the schemes described above. Simple quantities such as the SCF energy, geometry, and thermodynamic corrections are parsed by the *qchemio* module in the *pymatgen* package, as exemplified below:

```
from pymatgen.io.qchemio import QcOutput
qcout = QcOutput(''filename.qcout'')
final_energy = qcout.final_energy
optimized_mol = qcout.final_structure
```

In this example, first, a multiple jobs QChem output file is loaded to a QcOutput object. The above code retrieves the final energy and geometry in a geometry optimization job, which involves multiple energies and geometries as a function of the structure relaxation iterations.

Furthermore, the infrastructure automatically calculates and stores several derived properties, such as the IP/EA in vacuum/solution phase as a function of different reference electrodes. The database also contains auxiliary information (e.g. INCHI code, SMILES, molecular charge, molecular formula), which facilitates the query, and data mining of molecular structural-chemical property trends. An example of code to query the information in the database is provided below:

```
from pymongo import MongoClient
conn = MongoClient(db_ip_address,
  db_listening_port)
db.authenticate(your_user_name, your_password)
db = conn[db_name]
molecules = db[''molecules'']
quinoxaline_doc = list(molecules.find(
  {''user_tags.molname'': ''quinoxaline''}))
print quinoxaline_doc
```

An example MongoDB document looks like:

```
{
  ''elements'': [''H'', ''C'', ''N''],
  ''user_tags'': {''molname'': ''quinoxaline'',
  ...},
  ''inchi_root'': ''InChI=1S/C8H6N2/c1-2-4-8-
  7(3-1)9-5-6-10-8/h1-6H'',
  ''solvated_properties'': {
    ''water'': {
      ''IP'': 6.859902868147401,
      ''EA'': 2.6703552769304224,
      ''electrode_potentials'': {
        ''oxidation'': {
          ''lithium'': 5.459902868147401,
          ''hydrogen'': 2.4199028681474006,
          ''magnesium'': 4.789902868147401
        },
        ''reduction'': {
          ''lithium'': 1.2703552769304225,
          ''hydrogen'': -1.769644723069578,
          ''magnesium'': 0.6003552769304226
        }
      },
      ...
    },
    ...
    }, ''pointgroup'': ''C2v'',
    ''vacuum_properties'': {
      ''IP'': 8.708162265142164,
      ''EA'': 0.632506682730309,
      ...
    },
    ''charge'': 0,
    ''formula'': ''H6 C8 N2'',
    ...
}
```

Note that for brevity, some fields have been omitted as indicated by ellipses (...).

The goal of our infrastructure is to enable rapid, robust and accurate calculations of critical properties relevant for electrolyte screening and design; and furthermore, providing automatic dissemination of the results through a user-friendly web interface as well as database access. Hence, the data and query structure have been implemented as a separate Molecule Explorer 'App' under the Materials Project, which provides a user-friendly way for internal collaborators to directly access and search the large amount of data. Our intention is to also open this dataset publicly at the end of the Electrolyte Genome project. A snapshot of the web page is shown in Fig. 2.

Before describing specific workflow implementations for electrolyte molecule screening, we describe in the next section general aspects of our workflow software that are of use across several applications.

2.1. Molecule matcher

In many instances, it is useful to have an algorithm to check the equivalence of molecules within a given tolerance. For example, such an algorithm is used by our infrastructure to automatically avoid duplicate calculations. The Kabsch algorithm [25] is able to superimpose two molecules if the order of the atoms is the same in the two molecules. However, the atom order may well be different if the structural information of the molecules is obtained from different sources. To remedy this issue, the INCHI code auxiliary information [26] is employed. Starting from a topological/graph symmetry, the INCHI algorithm – provided by the OpenBabel [27] software package – is able to generate a canonical atom order that is independent of the initially provided atom order.

Fig. 3(a–f) exemplify the steps of the algorithm applied to two identical molecules A and B with different initial atom order. As shown, if we directly superimpose the two molecules via the Kabsch algorithm, without adjusting the atom orders, an erroneous, non-matching result will be obtained. Hence, in step (c) the atoms are labeled in the INCHI canonical order and subsequently matched through the Kabsch algorithm. Finally, the root mean square deviation (RMSD) is calculated as a measure of the difference between the two molecules. The function is available in the *molecule_matcher* module of the *pymatgen* package [19]. By default, the function categorizes the two molecules to be identical if the RMSD is less than 0.01 Å (although this tolerance can be changed). As an example, the equivalence of the two molecules with XYZ files "mol1.xyz" and "mol2.xyz" can be checked using the following code:

```
from pymatgen.core.structure import Molecule
from pymatgen.analysis.molecule_matcher import
  MoleculeMatcher
mm = MoleculeMatcher()
moll = Molecule.from_file(''t3.xyz'')
mol2 = Molecule.from_file(''t4.xyz'')
is_equal = mm.fit(moll, mol2)
```

2.2. Structural change detector

If the structure of a molecule changes fundamentally (e.g. in a geometry optimization job), it is important to register that a major change occurred during calculation. Such calculations may, for example, be responsible for outliers and anomalies in the final results. The INCHI code provides some information in this respect; however, if an equilibrium bond length is slightly longer than the normal bond length, the INCHI code will occasionally give a false negative. Hence, an in-house algorithm that makes use of the human knowledge embedded in the initial structure is utilized. The structural change is divided into two sub problems: new bond formation and bond breaking. Taking the covalent bond length [28] as a reference, a threshold of 30% is used to record all bonds in the initial molecular structure. When checking for new bond formation, we use the same 30% threshold to find any new bonds that

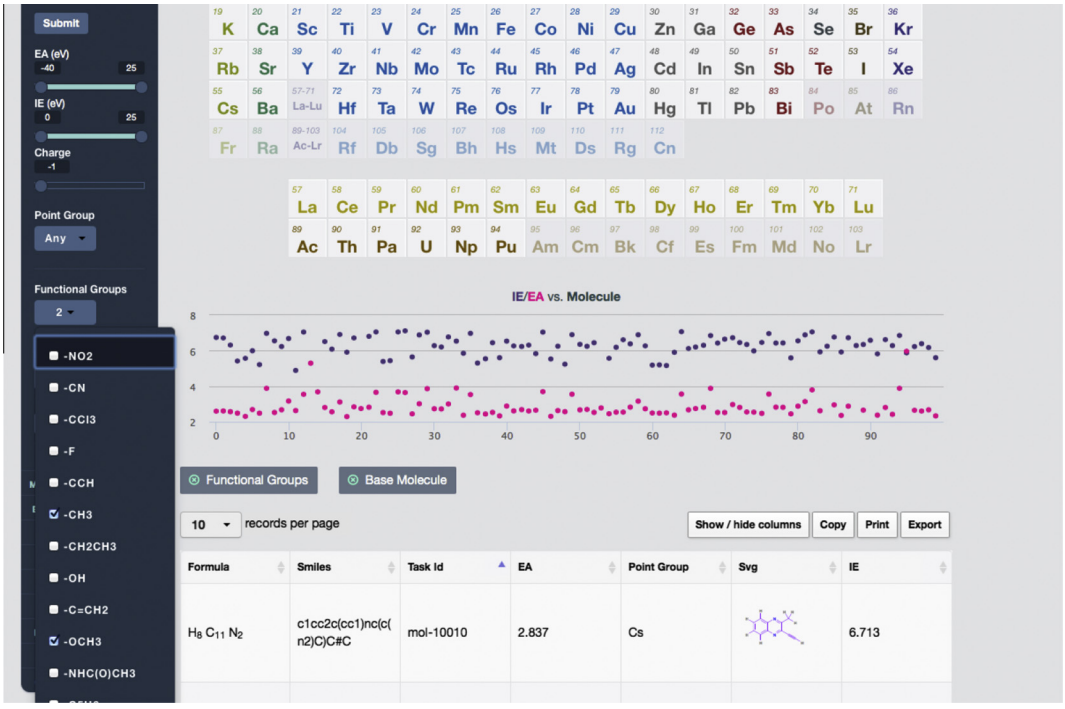


Fig. 2. Snapshot of the Electrolyte Genome Project Web Interface.

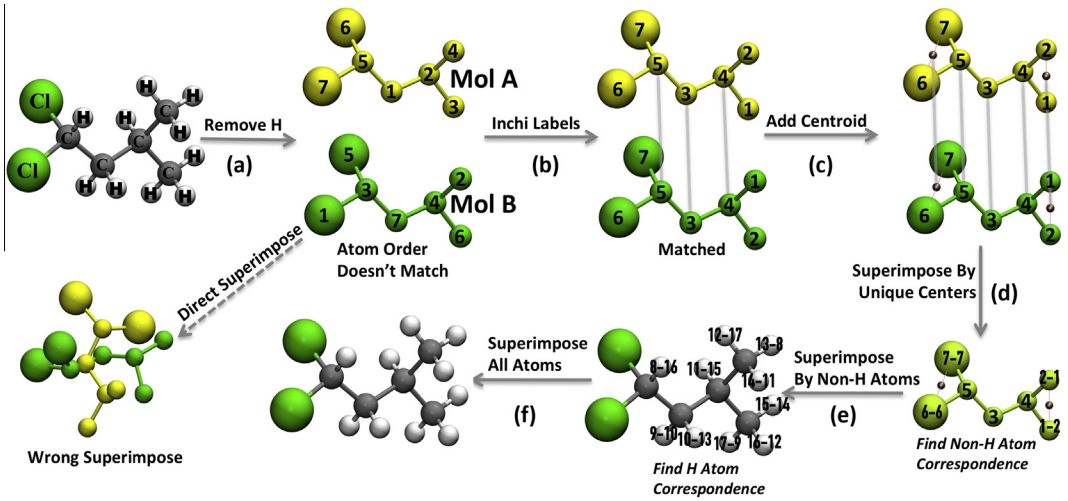


Fig. 3. Identical Molecules Detection. Example of steps in 'Molecule Matching' algorithm: Step (a) the hydrogen atoms are removed to facilitate the matching of the backbone molecule. Step (b) the INCHI auxiliary labels are assigned. The output "AuxInfo = 1/0/N: 3, 4, 1, 2, 5, 6, 7/E:(1, 2)(6, 7)" from OpenBabel [27] indicates that the atoms in canonical order are the 3rd, 4th, 1st, 2nd, 5th, 6th, 7th atoms from the original molecule, for which the 1st and 2nd and 6th and 7th are topologically equivalent, respectively. It is worth noting that the group as a whole is unique, and hence the centroids of these groups are unique. Step (c) the centroid information is added. In Step (d), the best rotation and translation to superimpose the molecules are determined by the unique atoms (3, 4, 5) and the centroids of the equivalent groups (1, 2 and 6, 7). In step (e), pair heavy atoms by closest point and superimpose the molecules by heave atoms. In Step (f), pair the hydrogen atoms and superimpose the molecules by all the atoms.

Table 1
QChem DFT calculation failure fixing methods. The chosen fix depends not only on the error type but also on the calculation status.

Error type		Fixing method
SCF failure	DIIS error $< 10^{-3}$	(a) Increase the SCF max iteration cycles to 200 (b) Set SCF iteration algorithm to DIIS_GDM (c) Set SCF initial guess to GWH (d) Set SCF iteration algorithm to GDM (e) Set SCF iteration algorithm to RCA (f) Set SCF initial guess to Core and SCF iteration algorithm to GDM
	DIIS error $\geqslant 10^{-3}$	(b) Set SCF iteration algorithm to RCA_DIIS (c) Set SCF initial guess to GWH (d) Set SCF iteration algorithm to RCA (e) Set SCF iteration algorithm to GDM (f) Set SCF initial guess to Core and SCF iteration algorithm to RCA
	In geometry optimization	(b) Set initial geometry to the last frame of the optimization; Reset SCF initial guess to default value; Reset SCF iteration algorithm to default value
Geometry optimization failure		(a) Increase maximum iteration cycles (100 for small molecules/300 for large molecules) (b) Use GDIIS geometry optimization iteration algorithm (c) Use Cartesian coordinates in geometry updating
Insufficient memory	OpenMP compatible jobs	Use OpenMP parallelism and increase QChem total memory setting to physical memory limit
	MPI compatible jobs	Reduce number of processes to half of the physical CPU cores and double memory usage per process
Symmetry detecting failure		Disable symmetry
Floating point overflow (NAN values)		Use denser integration grid

are formed during structure optimization. Meanwhile, for detection of a bond breaking, a default large threshold of 80% of the covalent bond length [28] is used to check whether any existing bonds are now broken (substantially elongated). These thresholds can be modified as needed by the user.

2.3. QChem custodian error handler

To computationally screen many molecular properties for a certain application requires automatic error handling and efficient use of computational resources. Of the 54,789 quantum mechanical calculations executed to date, 5045 QChem jobs encountered critical errors, such as self-consistent field (SCF) convergence errors or memory allocation errors. For such large computational investigations, it would be impossible to use human intervention to perform routine tasks such as parsing the error message, applying a series of recommended failure fixes and re-starting the job. We have implemented a QChem plugin to the *custodian* codebase to apply automatic error checking and correction through well-defined rules and intelligent dynamic workflows.

Table 1 summarizes in detail the recommended fixes for common single task level QChem errors. The types of errors addressed are SCF failures, geometry optimization failures, symmetry detection failures, floating point overflows, and insufficient memory. The current ensemble of automatic error handling renders 90.9% (4587 out of the 5045) failed quantum mechanical calculations successful, and leading to an overall success rate of 99.2%. We should emphasize that additional strategies are straightforward to implement by extending the rules defined in the plug-in.

2.4. Dynamic job creation

Even if a calculation is converged, it is still possible that the result is not physical. For example, a stable molecular structure

should have no imaginary frequencies, while a transition state should have exactly one imaginary frequency. The infrastructure code is able to check whether a stationary point has the desired number of imaginary frequencies and fix it automatically through a tested, dynamic change in the workflow. These dynamic changes to the job sequence are a feature of the *FireWorks* [20] workflow software that allows the workflow graph to be automatically modified and appended to during its execution.

To determine the best workflow strategy and parameters for the imaginary frequency elimination, we tested four different methods for 72 molecules with imaginary frequencies. These molecules originated from 576 geometry optimizations and represent common redox-active electrolyte molecule types (e.g., quinoxaline, bipyridine, and DMSO). The four methods evaluated for our workflow were: (1) Re-optimization of the molecular geometry using improved accuracy settings; (2) Direct re-calculation of the vibrational frequency using improved accuracy settings without re-optimizing the geometry; (3) Perturbation of the molecular geometry along the direction of the imaginary frequency vibrational mode, followed by normalization of the vibrational vector by adjusting the maximum atomic displacement to 0.3 Å and re-optimization of the geometry; and finally, (4) Perturbation of the molecular geometry by directly adding the raw vibration vector to the molecular coordinates and then re-optimizing the geometry.

The 72 molecules were clustered into 3 bins (with a bin size of 24 molecules) based on amplitude of the imaginary frequency. The results, shown in Fig. 4, demonstrate that if the frequency is very low (<39.0 cm^{-1}), a high-accuracy frequency calculation can remove up to 42% of the imaginary frequencies. However, high accuracy re-optimization is less helpful for larger imaginary frequencies. In contrast, the two molecular geometry perturbation strategies work extremely well: they remove at least 21 out of the 24 imaginary frequencies for the whole range of imaginary frequencies. The perturbation of 0.3 Å performs slightly better than the perturbation with the raw vibrational vector (the latter can break the structure and is thereby less robust).

Hence, for automated imaginary frequency elimination, we first perturb the molecule by 0.3 Å along the direction of the undesired vibrational mode (see Table 2). Both geometry optimization and frequency calculations are re-performed for the perturbed structure. A direct consequence of the dynamic workflow is that the number of calculations is not necessarily the same for different molecules. As shown in Fig. 5, these dynamically spawned jobs are additional calculations employed to recover stable structures from dynamically unstable ones.

2.5. Symmetry detection

Symmetry plays an important role in the performance of DFT calculations, particularly for properties such as vibrational spectra.

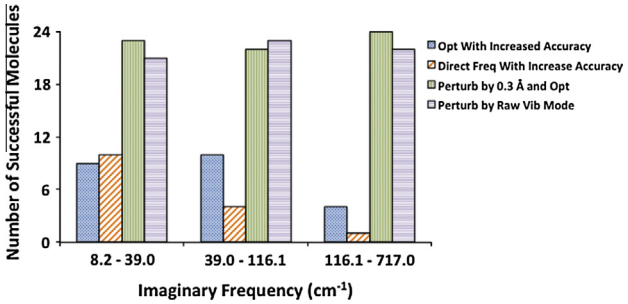


Fig. 4. Number of successfully recovered failures from imaginary frequency errors on 72 total test cases distributed among 3 frequency bins. Methods 3 and 4 are the most effective across all frequencies, and in particular at higher frequencies.

Table 2
Imaginary frequency elimination methods as implemented in the *rubicon* package.

Stationary type	Elimination method
Minimum	(a) Perturb the geometry by 0.3 Å along direction of the vibration mode with the largest imaginary frequency, and use the perturbed geometry as initial geometry to perform a new geometry optimization and vibrational frequency analysis (b) Use a tighter integration threshold (10^{-12}) and a denser grid (Lebedev grid with 128 radial points, 302 angular points), then follow option (a) to re-optimize the geometry. A tighter geometry optimization criterion is imposed by adjusting the threshold to 10% of the default value (c) Similar to (b), but use a different integration grid (Lebedev grid with 90 radial points, 590 angular points)
Transition state	Pick out the second largest imaginary frequency, and use the corresponding vibrational vector to perturb the molecular geometry following the methods for "Minimum"

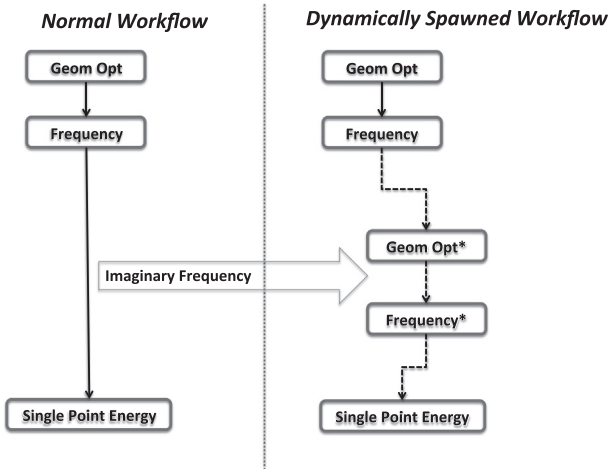


Fig. 5. Dynamic workflow for imaginary frequency elimination. The asterisk denotes jobs created automatically during runtime that are not present when the workflow is first defined.

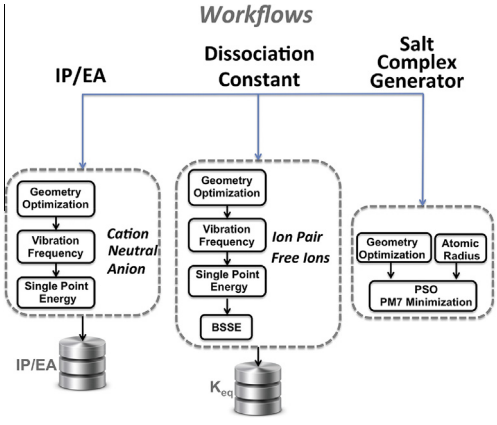


Fig. 6. Fully operational workflows as implemented in the *rubicon* package.

The *pymatgen* [19] package has already implemented a module (*pymatgen.symmetry.analyzer.PointGroupAnalyzer*) that can detect the maximum point group symmetry of the given molecule. This module is used to parse the molecular symmetry in the Electrolyte Genome project.

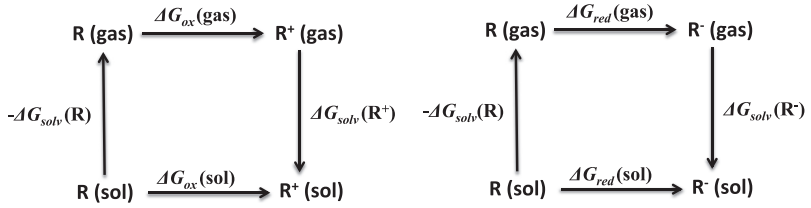


Fig. 7. Free energy cycle for computing the oxidation/reduction potential. *R* denotes the molecule of interest.

3. Workflows for electrolyte screening

We use the term "workflow" to refer to the procedure to calculate a specific property, including all the steps and the corresponding parameters in the calculation. All workflows are developed in a modular way in which elementary tasks are connected into a larger whole, making it very easy to program new workflows. Users can also change pre-programmed behaviors of existing workflows within the molecule submission parameters.

Currently, there exist three fully operational workflows implemented for molecular calculations: IP/EA Calculation, Salt Complex Generator and Ion Pair Dissociation Constant Calculation (see Fig. 6). Other workflows are in development and are discussed in the Future Work section.

3.1. IP/EA

The ionization potential (IP) is the energy required to oxidize a molecule while the electron affinity (EA) is the energy to reduce a molecule. The IP and EA are two of most important properties of an electrolyte component [29]. For a salt or solvent, these properties can determine the electrochemical window, which limits the potential within which the battery can operate [29]. For redox active molecules, it is a proxy for the oxidation/reduction potential, which determines the operating voltage of a redox flow battery [30–32]. The cathodic limit (V_{CL}) is set by reduction of the molecule of interest, whereas the anodic limit (V_{AL}) is set by oxidation of the molecule of interest:

$$V_{CL} = \mathrm{EA} = -\frac{\Delta G_{red}(\mathrm{sol})}{nF} \quad V_{AL} = \mathrm{IP} = -\frac{\Delta G_{ox}(\mathrm{sol})}{nF}$$

where F is the Faraday constant, and ΔG_{ox}(sol) and ΔG_{red}(sol) are the Gibbs free energy change of oxidation and reduction in the solution phase, respectively.

According to the following thermodynamic cycle in Fig. 7, ΔG_{ox}(sol) and ΔG_{red}(sol) can be calculated from the Gibbs free energy change of gas phase:

$$\Delta G_{ox}(\mathrm{sol}) = \Delta G_{ox}(\mathrm{gas}) + \Delta G_{solv}(R^+) - \Delta G_{solv}(R)$$

$$\Delta G_{red}(\mathrm{sol}) = \Delta G_{red}(\mathrm{gas}) + \Delta G_{solv}(R^-) - \Delta G_{solv}(R)$$

ΔG_{solv} is the energy to solvate a molecule/ion from vacuum to solution phase. Several methods to obtain the solvation energy exist with varying degrees of accuracy and computational cost [33–36]. A commonly used approximation is the dielectric continuum model [37], which models the solvent as a dielectric continuum and generates results that match qualitatively with trends obtained from higher accuracy methods [35,38,39]. In the current study, the integral equation formalism polarizable continuum model (IEF-PCM) [40] implicit solvent model is employed to include solvation effects. ΔG_{ox}(gas) and ΔG_{red}(gas) can be calculated from the Gibbs free energy of individual molecule/ion:

$$\Delta G_{ox}(\mathrm{gas}) = G(R^+(\mathrm{gas})) - G(R(\mathrm{gas}))$$

$$\Delta G_{ox}(\mathrm{gas}) = G(R(\mathrm{gas})) - G(R^-(\mathrm{gas}))$$

The individual Gibbs free energies can be computed from the following equation:

$$G = H - T\Delta S = E_{SCF} + E_{ZPVE} + H_{corr} - T\Delta S_{corr} \approx E_{SCF}$$

where E_{SCF}, E_{ZPVE}, H_{corr} and S_{corr} are the calculated DFT energy, zero-point vibrational energy correction, thermal enthalpic correction and entropic correction, respectively. The correction values are typically small, and usually cancel out to a large extent since the geometric configurations are expected to be similar. In the current study, they are not included in the final IP/EA calculation.

Theoretically, the ideal method to calculate IP/EA is the adiabatic IP/EA [41], which optimizes the geometry at different charge states

Table 3
Comparison of adiabatic IP/EA Prediction as compared to CCSD(T) computed with XYGJ-OS and B3LYP based on B3LYP geometries, B3LYP//PBE hybrid approach based on PBE geometries for 15 molecules derived from thiophene.

	IP				EA			
	CCSD(T)	XYGJ-OS	B3LYP	B3LYP//PBE	CCSD(T)	XYGJ-OS	B3LYP	B3LYP//PBE
2-Thiophenamine	7.40	7.43	7.20	7.20	−1.07	−1.07	−0.95	−0.95
2-Thiophenecarboxylic acid	9.08	9.13	8.99	9.01	0.01	0.07	0.34	0.34
2-Thiophenecarbonitrile					−0.05	0.03	0.28	0.28
N,N-Dimethyl-2-thiophenamine	6.93	6.97	6.75	6.75	−1.12	−1.14	−1.00	−1.00
2-Acetamidothiophene					−0.86	−0.84	−0.57	−0.58
2-Ethylthiophene	8.34	8.39	8.13	8.13	−1.09	−1.07	−0.84	−0.84
2-Ethynylthiophene	8.44	8.48	8.16	8.16	−0.44	−0.37	−0.06	−0.06
2-Fluorothiophene	8.72	8.79	8.61	8.61	−0.91	−0.88	−0.58	−0.59
2-Thiopheneol	8.07	8.13	7.91	7.91	−1.05	−1.02	−0.90	−0.91
2-Methoxythiophene	7.76	7.80	7.58	7.58	−1.16	−1.14	−0.90	−0.90
2-Methylthiophene	8.38	8.46	8.21	8.18	−1.21	−1.20	−1.01	−0.98
N-Methyl-2-thiophenamine	7.16	7.20	6.96	6.96	−1.14	−1.14	−0.98	−0.98
2-Nitrothiophene	9.60	9.59	9.53	9.54	0.92	0.93	1.36	1.36
2-(Trichloromethyl)thiophene	8.84	8.96	8.76	8.73	1.05	1.11	1.62	1.64
2-Vinylthiophene	8.11	8.12	7.83	7.82	−0.44	−0.36	−0.15	−0.15

(cation, anion, neutral). This is the default method implemented in our workflow, which emphasizes high-fidelity results. An approximation to the adiabatic IP/EA is the vertical IP/EA, in which all the energy calculations use neutral state geometry. A further approximation would be to rule out the single point energy at the cation and anion state, and use the HOMO/LUMO for the neutral state as an approximation to IP/EA. These approximated methods can also be used by setting the appropriate submission parameters when submitting a molecule, and are related to methods used by the Harvard Clean Energy project [3,4] and Korth's project [15].

3.1.1. Special Treatment for Large Molecules

Although B3LYP works very well for small molecules, its formal computational cost scaling is $O(N^4)$. Even with accelerating algorithms, such as integral screening, the computational cost scales as $O(N^2)$ for numerical integrals with large prefactors and $O(N^3)$ for density matrix diagonalization with small prefactors. In practice, the computational cost scaling of B3LYP is usually close to $O(N^{2.5})$ for medium sized molecules [42], which makes large molecules difficult to handle in high-throughput. Furthermore, the exclusive memory requirements of vibrational frequency analysis and IEF-PCM solvation energy calculation can also cause workflow failures for large molecules. To remedy this issue, we developed a hybrid procedure for molecules with more than 50 atoms: (1) optimize the geometry at the PBE/6-31+G* level [43] with Grimme's dispersion correction [44], which is a pure density functional and more computationally economical since the evaluation of HF exchange is not required. However, the final energy is still evaluated at B3LYP/6-31+G* level; (2) Discard vibrational frequency calculation; (3) Use less grid points in the discretization of the cavity surface of IEF-PCM calculation (194 points per atom, default value is 594 for QChem 3.× and 302 for QChem 4.×); 4) Loosen the convergence threshold of geometry optimization (the threshold of energy change and maximum atomic displacement is increased ten times; the threshold of maximum gradient is kept unchanged). As can be seen from Table 3, the predicted IP/EAs from the B3LYP//PBE hybrid approach and full B3LYP predicted IP/EAs are very close; the maximum deviation is only 0.04 eV. Therefore, the B3LYP//PBE hybrid approach is a suitable alternative to calculate IP/EA for large molecules.

3.2. Salt complex generator

A physically relevant structure is a prerequisite for computational study. However, due to the multiple component nature of the electrolyte, it is challenging to find the lowest energy conformations that likely represent physical systems. The salt complex generator module attempts to determine the lowest energy complex configuration automatically in the presence of multiple salt ions, additives, and solvent molecules [29,31,45].

The salt complex generator is composed of two basic parts: an optimizer and an energy evaluator. For the optimizer we chose artificial intelligence based Particle Swarm Optimization (PSO) [46–50] in preference over Conjugate Gradient (CG) and Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithms because the aim is to search for a global minimum energy structure rather than a local minimum/conformation. Compared to other artificial intelligence algorithms, such as Stratified Sampling (SS) [51] and Genetic Algorithm (GA) [52], which work best for combinatorial searches, PSO performs well for both continuous as well as combinatorial search spaces. As the salt complex generation is a continuous problem, PSO is more suitable. Furthermore, SS require that the solution space be exhaustively partitioned into disjoint subgroups, which is in itself a challenging algorithmic exercise for the salt complex generation space. We developed two variants of energy evaluators to quickly screen conformations: (1) an electrostatics-only model, which is very simple and fast, and (2) a range-separated hybrid model, which is a hybrid of PM7 [53] semi-empirical quantum mechanical energy and empirical gravitational force.

The electrostatic model captures the essence of the interaction of charged particles through the Coulomb potential. The system energy is calculated via Coulomb's Law, while the overlapping of atoms is excluded by a hard sphere based algorithm. As can be seen from Fig. 8, the electrostatic model can dock monovalent salt ions

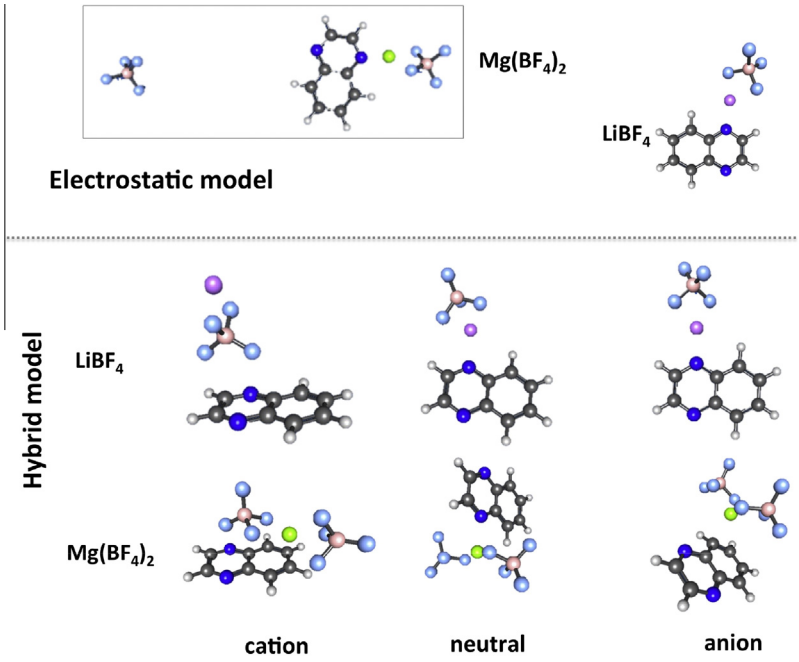


Fig. 8. Result of automated salt complex generator on a quinoxaline base molecule. The electrostatic-only model (top) fails for multivalent $Mg(BF_4)_2$, whereas the hybrid model docks both monovalent and multivalent salts.

to redox active molecules, however, it fails for multivalent salts, e.g. magnesium salts.

The hybrid model calculates the system energy either by empirical gravitational force or by PM7 depending on the distance between ions/molecules. Empirical gravitational force will be used if they are well-separated, which forms a funnel to bring the ions/molecule close to one other. After the ion/molecules are in contact, the more accurate PM7 method is employed. In our codebase, a PM7 based local energy minimization (steepest decent) is performed via the MOPAC package [54,55] in each macro iteration and serves the final energy as the fitness value of PSO. This renders our salt algorithm to be a hybrid optimization algorithm that combines PSO and steepest decent, which results in a more effective lowest energy conformation search toolkit.

The result of the hybrid model is also shown in Fig. 8 for comparison. For the quinoxaline test case, the hybrid model is superior to the electrostatic model: it is not only able to dock the monovalent $LiBF_4$ salt, but is also able to dock the multivalent $MgBF_4$ salt.

3.3. Ion pair dissociation constant

Ion pairing is the phenomena by which the cation and anion are associated together in an electrolyte. In contrast to a fully solvated salt in which each cation/anion has a complete solvent shell and exists as free ions, in an ion pair, the cation is in direct contact with the anion and shares the solvent shell. Ion pairs are observed in many electrolyte systems, particularly at higher salt concentrations [56,57]. To understand ion pair interactions [58] in electrolytes, it is important to calculate the ion-pair formation driving force, which is captured by the ion pair dissociation constant [45]. The dissociation constant is defined by the molar ratio of free ions and ion pair:

$$K = \frac{[\text{cation}][\text{anion}]}{[\text{ion pair}]}$$

and can be deduced from the change in Gibbs free energy between the products and reactants:

$$K = e^{-\Delta G/RT}$$

$$\Delta G = \sum_{product} G_i - \sum_{reactant} G_j$$

G is the total free energy of the individual free ions or ion pair, and can be computed from a similar procedure as discussed in the IP/EA workflow. In our code base, the equilibrium constant calculation is decoupled to single point energy workflows for the cation, anion and ion pair, respectively. By decoupling the workflows, we reuse existing workflows and de-convolute the sub tasks. The counterpoise correction (CP) is applied to eliminate the basis set superposition error (BSSE) [59,60]. To obtain the dissociation constant, the user needs to specify the structure and molar ratios of the cation, anion and ion pair. The equilibrium constants are then automatically calculated and stored in MongoDB.

4. Results

We have calculated thousands of molecules from different sources: (i) systematic structure generation through substitution of functional groups in a base molecule; (ii) molecules from public databases such as SciFinder [61] and Reaxys [62]; (iii) molecules submitted by collaborators (see Fig. 1).

We also tested the proposed method and workflow against available experimental results in the G2-97 [63] test set, and the results of these calculations are described next. All calculations were automatically carried out by the infrastructure and the failed (16 IPs and 13 EAs) molecules were excluded in the final analysis.

4.1. IP/EA benchmarking

First, we evaluated the accuracy of two basis sets, 6-31+G(d) and 6-311++G(2d,2p), in combination with B3LYP functional. The B3LYP/6-31+G(d) and 6-311++G(2d,2p) basis sets both predict IP/EA within a mean absolute error (MAE) of 0.16 eV and root mean squared deviation (RMSD) of 0.28 eV as compared to experimental values in the G2-97 test set (see Table 4). Because the 6-311++G(2d,2p) is significantly more computationally expensive but its accuracy gain is negligible over this test set, we chose to employ the 6-31+G(d) basis set for further IP/EA calculations.

In addition to the basis set, the choice of functional also plays a critical role in the accuracy and computational cost of the IP/EA prediction. Using our infrastructure, we tested a series of functionals on the G2-97 test set, including the XYGJ-OS [64], PW6B95 [65], B3LYP [66], M06 [67], M06-2X [67], PBE0 [68], TPSSh [69], and B97-D [70] functionals. As can be seen from Table 5, the XYGJ-OS functional, which is a double hybrid density functional containing both Hartree–Fock exchange and electron correlation information from an MP2-like term [71], performed significantly better than the B3LYP functional, while computationally scaling at a similar level. For the prediction of both IP and EA, the RMS error of XYGJ-OS (0.1 eV) was less than half of the others (0.25–0.38 eV) and it was the only functional with a maximum error of less than

Table 4
Comparison of prediction errors as compared to experiment for two basis sets (6-31+G(d) and 6-311+G(2d, 2p)) computed with the B3LYP functional over the G2-97 basis set. RMSD: root mean square deviation, MAE: Mean Absolute Deviation, ME: Max Error, R^2: Coefficient of Determination.

	IP (eV)		EA (eV)	
	6-31+G(d)	6-311+G(2d, 2p)	6-31+G(d)	6-311+G(2d, 2p)
RMSD	0.27	0.28	0.25	0.23
MAE	0.16	0.16	0.16	0.13
ME	1.52	1.66	1.04	1.10
R^2	0.985	0.984	0.947	0.956

Table 5
Comparison of IP/EA prediction error as compared to experiment over G2-97 test set computed with different functionals. RMSD: root mean square deviation, MAE: Mean Absolute Deviation, ME: Max Error, R^2: Coefficient of Determination.

		XYGJ-OS	PW6B95	B3LYP	M06-2X	M06	PBE0	TPSSh	B97-D
IP (eV)	RMSE	0.10	0.25	0.27	0.28	0.27	0.29	0.29	0.38
	MAE	0.07	0.13	0.16	0.13	0.18	0.18	0.21	0.26
	Max	0.39	1.54	1.52	1.76	1.31	1.61	1.35	1.57
	R^2	0.998	0.986	0.985	0.984	0.987	0.982	0.982	0.968
EA (eV)	RMSE	0.12	0.23	0.25	0.25	0.32	0.38	0.46	0.82
	MAE	0.10	0.16	0.16	0.16	0.18	0.25	0.26	0.29
	Max	0.52	0.96	1.04	0.96	1.64	1.76	2.59	4.74
	R^2	0.986	0.953	0.947	0.951	0.903	0.868	0.816	0.646

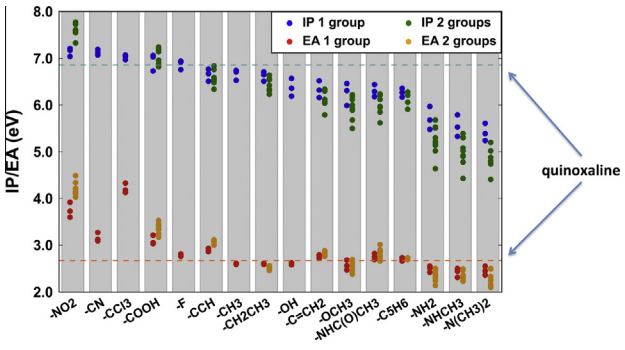


Fig. 9. IP/EA of the single functional group and two identical functional group substituted quinoxaline derivatives. Green and blue dots denote the IP of single and substituted derivatives respectively. Red and orange dots denote the EA of single and double substituted derivatives respectively.

1.0 eV (0.39 eV for IP and 0.52 eV for EA). Furthermore, it was found that XYGJ-OS also performs excellently in the prediction of redox active molecules. A set of 15 thiophene derivatives was used to calculate the IP/EA at the CCSD(T) [72], B3LYP and XYGJ-OS level, respectively. The results are shown in Table 3. The Spearman's rank correlation coefficient [73] was utilized to evaluate the degree of the similarity between two rankings where a value of 1.0 represents a perfect agreement between two rankings, and conversely, 0.0 indicates no agreement. Both XYGJ-OS and B3LYP show excellent rank correlation coefficients (0.995 and 0.989) as compared to CSSD(T) predictions. We note that for the EA prediction, the B3LYP rank correlation coefficient was slightly less, with a value of 0.952.

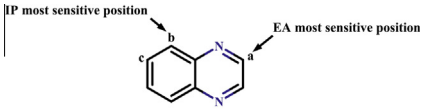
Although XYGJ-OS unambiguously performed the best in the benchmark prediction of IP/EA, while PW6B95 and B3LYP ranked as second best, the decision was made to use B3LYP as our main computational method due to the following pragmatic reasons: (1) B3LYP is widely supported by most quantum chemistry softwares and has been extensively tested [74] and used which makes it is very easy to compare with other results. In contrast, XYGJ-OS is currently only implemented by the QChem [17,18] and FireFly [75] software packages. (2) Several third-party algorithms, such as Truhlar's Minnesota solvation model [33] is known to work with B3LYP, but is not tested with XYGJ-OS. (3) XYGJ-OS lacks a 2nd order analytical derivative, which is preferred for fast thermodynamics correction calculation and makes XYGJ-OS unfavorable for vibrational frequency analysis. (4) B3LYP ranks as the second best in the benchmark data, and its accuracy is still fairly good. In summary, our final choice for the default functional for use in IP/EA calculations is the B3LYP density functional with the 6-31+G(d) basis set.

4.2. Example quinoxaline structure-property trends

To demonstrate a potential use of our infrastructure, we performed 55,000 quantum mechanical calculations, leading to 4830 IP/EAs. 1536 redox active molecules were derived by systematically adding functional groups to the base molecules quinoxaline, anthrachinon, thiane, thiophene and bipyridine [16]. Fig. 9 shows the results for the quinoxaline derivatives, which demonstrate that the addition of an electron-withdrawing functional group increases both IP and EA, while an electron-donating functional group decreases both of them. By using two identical functional groups in different positions, this trend can sometimes be enhanced.

The raw data can also be used to build higher level models. For example, to further study and quantify the effect of functional group and substitution position, we fit our results to the Hammett equation [76], in this case for the IP:

Table 6
Fitted parameters of Hammett equation for quinoxaline derivatives IP/EA prediction. For more information obtained for structure-property trends of redox active molecule, see Ref. [16].



IP		EA	
Position constant (K_p)			
a	0.86		1.07
b	1.21		0.87
c	1.02		0.79
Group constant (K_g)			
$-N(CH_3)_2$	−1.00	$-NH_2$	−0.25
$-NHCH_3$	−1.00	$-NHCH_3$	−0.23
$-NH_4$	−0.83	$-N(CH_3)_2$	−0.23
—OCH	−0.46	$-OCH_3$	−0.10
—OH	−0.42	$-CH_2CH_3$	−0.10
$-NHC(O)CH_3$	−0.41	$-C_5H_6$	−0.02
$-C_5H_6$	−0.38	—OH	0.02
$-C{=}CH_2$	−0.35	$-CH_3$	0.03
$-CH_3$	−0.26	$-NHC(O)CH_3$	0.05
$-CH_2CH_3$	−0.19	$-C{=}CH_2$	0.12
—CCH	−0.13	—CCH	0.19
—F	−0.08	—F	0.30
—CN	−0.05	—COOH	0.36
—COOH	0.12	$-CCl_3$	1.00
$-CCl_3$	0.24	—CN	1.00
$-NO_2$	0.27	$-NO_2$	1.00

$$IP = IP_0 + p * g$$

where IP_0 is the ionization potential of quinoxaline without any substituent, p is a constant that depends only on substituent position, and g is a constant determined solely by the functional group type. The $p * g$ term essentially serves as a correction to the base molecule's ionization potential. The EA can be fitted to a corresponding equation. Table 6 shows the Hammett fit parameters for 300 calculated quinoxaline derivatives yielding a predicted IP and EA with average error of 0.161 and 0.286 eV, respectively. The success of the Hammett algorithm suggests that the chemical and position effect for the IP/EA quinoxaline derivatives can be separated such that each group and each position can be represented by a simple constant factor. This method could also be used to predict promising functional group substitutions for a specific base molecule by calculating a subset of functional group/position combinations to fit the model, and then using the model to predict the remaining values.

5. Future workflows

5.1. Diffusion coefficient and solvation structure

Other key properties for electrolyte performance are the diffusion coefficients of the individual molecular components and the overall viscosity of the solution. Additionally, radial distribution functions (RDFs) and spatial distribution functions (SDFs) serve as intuitive tools to understand preferred dynamic molecule interactions and configurations. Furthermore, the snapshot of the first solvation shell can be used as input to *ab initio* solvation energy calculations [58].

The goal of the *rubicon* package is to automate and expedite not only molecular and electrolyte property calculations obtained through *ab initio* calculations, such as the prediction of the IP/EA, but also those that can be derived through classical molecular dynamics (MD) simulations. The ultimate goal is to seamlessly integrate *ab initio* and through force field generation and

information flow between the two length scales. Currently, the *ab initio* atomic charges are fitted by a Restrained Electrostatic Potential (RESP) [77] procedure and used as input in force field generation. In turn, the structures of the first solvation shell from MD ensembles are used to obtain accurate energetics for the solvation energy. While pieces of this workflow are operational, it is still under testing.

5.2. Artificial intelligence molecular design

The design of novel electrolyte still relies heavily on human intuition. To make use of the information provided by "big data", the Electrolyte Genome project is developing a novel molecule design module based on statistical learning, which combines the prediction power of machine learning models and the solution generation ability of artificial intelligence. The module will be used to search chemical space that is not intuitive to humans, or when the solution space is too large for human search.

5.3. Other workflows

Several workflows are under development and will be successively implemented. A few examples include the *Decomposition pathfinder*, *Solubility Workflow*, and *Quantitative Structure Activity Relationship (QSAR)* which contain different step-wise property calculations, dynamic decisions on calculation sequences, and feedback between *ab initio* and MD length scale domains.

6. Summary

We have developed an open source infrastructure for large scale molecule screening that leverages the resources available, in combination with best practices in information software development and high-performance computing. A variety of techniques and supporting code are established to robustly and efficiently calculate, analyze and organize molecular properties including: (1) redox potential which is helpful for high-voltage battery electrolyte screening, (2) ion pair dissociation constants which is helpful for electrolyte stability studies, (3) salt complex structure which can contribute to the fundamental scientific understanding of electrolytes.

We have successful applied the infrastructure to the calculation of 4830 IP/EAs. The benchmark on the G2-97 test set shows that our calculation procedure is able to give reliable results for both small and large molecules. We have fitted the quinoxaline derivatives IP/EA to the Hammett equation which reveals that the position effect and functional group effect can be separated and that the most sensitive positions for IP and EA are different. The automatic error handling module fixes 90.9% of the calculation failures and improves the success rate from 90.8% to 99.2%. Our molecular matcher is able to identify identical molecules irrespective of atom order in the molecule description. By leveraging artificial intelligence approaches, the salt complex generator is able to find the lowest energy conformation for complicated multi-composition systems.

As demonstrated by the application example, the infrastructure developed in this project is able to process large amounts of molecules. We hope this infrastructure can help to accelerate the process of design and screen for electrolyte with enhanced properties and aid in the fundamental science study of electrolytes.

Acknowledgements

Support for this work came from the U.S. Department of Energy, Basic Energy Science, Joint Center for Energy Storage Research under Contract No. DE-AC02-06CH11357. The calculations were performed using the computational resources of the National Energy Research Scientific Computing Center, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231. The Materials Project (BES DOE Grant No. EDCBEE) is acknowledged for infrastructure and algorithmic support.

References

[1] G. Ceder, K. Persson, Sci. Am. 309 (2013) 36–40, http://dx.doi.org/10.1038/scientificamerican1213-36.
[2] A. Jain, S.P. Ong, G. Hautier, W. Chen, W.D. Richards, S. Dacek, et al., APL Mater. 1 (2013) 011002, http://dx.doi.org/10.1063/1.4812323.
[3] J. Hachmann, R. Olivares-Amaya, A. Jinich, A.L. Appleton, M.a. Blood-Forsythe, L.R. Seress, et al., Energy Environ. Sci. 7 (2014) 698, http://dx.doi.org/10.1039/c3ee42756k.
[4] J. Hachmann, R. Olivares-Amaya, S. Atahan-Evrenk, C. Amador-Bedolla, R.S. Sánchez-Carrera, A. Gold-Parker, et al., J. Phys. Chem. Lett. 2 (2011) 2241–2251, http://dx.doi.org/10.1021/jz200866s.
[5] S. Curtarolo, W. Setyawan, G.L.W. Hart, M. Jahnatek, R.V. Chepulskii, R.H. Taylor, et al., Comput. Mater. Sci. 58 (2012) 218–226, http://dx.doi.org/10.1016/j.commatsci.2012.02.005.
[6] J.E. Saal, S. Kirklin, M. Aykol, B. Meredig, C. Wolverton, JOM 65 (2013) 1501–1509, http://dx.doi.org/10.1007/s11837-013-0755-4.
[7] J.A. Pople, L.A. Curtiss, J. Phys. Chem. 91 (1987) 155–162, http://dx.doi.org/10.1021/j100285a035.
[8] S.F. Nelsen, Isr. J. Chem. 18 (1979) 45–55, http://dx.doi.org/10.1002/ijch.197900005.
[9] S.P. Ong, O. Andreussi, Y. Wu, N. Marzari, G. Ceder, Chem. Mater. 23 (2011) 2979–2986, http://dx.doi.org/10.1021/cm200679y.
[10] N. Shao, X.-G. Sun, S. Dai, D. Jiang, J. Phys. Chem. B 115 (2011) 12120–12125, http://dx.doi.org/10.1021/jp204401t.
[11] Y. Zhang, C. Shi, J.F. Brennecke, E.J. Maginn, J. Phys. Chem. B 118 (2014) 6250–6255, http://dx.doi.org/10.1021/jp5034257.
[12] H. Maeshima, H. Moriwake, a. Kuwabara, C.a.J. Fisher, I. Tanaka, J. Electrochem. Soc. 161 (2014) G7–G14, http://dx.doi.org/10.1149/2.069403jes.
[13] M. Okoshi, Y. Yamada, a. Yamada, H. Nakai, J. Electrochem. Soc. 160 (2013) A2160–A2165, http://dx.doi.org/10.1149/2.074311jes.
[14] J.J.P. Stewart, J. Mol. Model. 13 (2007) 1173–1213, http://dx.doi.org/10.1007/s00894-007-0233-4.
[15] M. Korth, Phys. Chem. Chem. Phys. 16 (2014) 7919–7926, http://dx.doi.org/10.1039/c4cp00547c.
[16] L. Cheng, R.S. Assary, X. Qu, A. Jain, S.P. Ong, N.N. Rajput, et al., J. Phys. Chem. Lett. 6 (2015) 283–291, http://dx.doi.org/10.1021/jz502319n.
[17] A.I. Krylov, P.M.W. Gill, Wiley Interdiscipl. Rev. Comput. Mol. Sci. 3 (2013) 317–326, http://dx.doi.org/10.1002/wcms.1122.
[18] Y. Shao, Z. Gan, E. Epifanovsky, A.T.B. Gilbert, M. Wormit, J. Kussmann, et al., Mol. Phys. (2014), http://dx.doi.org/10.1080/00268976.2014.952696.
[19] S.P. Ong, W.D. Richards, A. Jain, G. Hautier, M. Kocher, S. Cholia, et al., Comput. Mater. Sci. 68 (2013) 314–319, http://dx.doi.org/10.1016/j.commatsci.2012.10.028.
[20] A. Jain, S.P. Ong, D. Gunter, W. Chen, B. Medasani, X. Qu, et al., Fireworks: a dynamic workflow system designed for high-throughput applications, Concurr. Comput. Pract. Exp. (2014) (submitted for publication). https://pythonhosted.org/FireWorks/index.html.
[21] M.J. Frisch, G.W. Trucks, H.B. Schlegel, G.E. Scuseria, M.A. Robb, J.R. Cheeseman, et al., Gaussian 09, Gaussian Inc., Wallingford, CT, 2009.
[22] M. Valiev, E.J. Bylaska, N. Govind, K. Kowalski, T.P. Straatsma, H.J.J. Van Dam, et al., Comput. Phys. Commun. 181 (2010) 1477–1489, http://dx.doi.org/10.1016/j.cpc.2010.04.018.
[23] MongoDB Inc., MongoDB, 2014.
[24] D. Crockford, Google Tech Talks: JavaScript: The Good Parts, 2009. https://www.youtube.com/watch?v=hQVTIJBZook.
[25] W. Kabsch, Acta Crystallogr. Sec. A 32 (1976) 922–923, http://dx.doi.org/10.1107/S0567739476001873.
[26] S. Heller, A. McNaught, S. Stein, D. Tchekhovskoi, I. Pletnev, J. Cheminform. 5 (2013) 7, http://dx.doi.org/10.1186/1758-2946-5-7.
[27] N.M. O'Boyle, M. Banck, C. a James, C. Morley, T. Vandermeersch, G.R. Hutchison, J. Cheminform. 3 (2011) 33, http://dx.doi.org/10.1186/1758-2946-3-33.
[28] B. Cordero, V. Gómez, A.E. Platero-Prats, M. Revés, J. Echeverría, E. Cremades, et al., Dalt. Trans. (2008) 2832–2838, http://dx.doi.org/10.1039/b801115j.
[29] J.M. Vollmer, L.a. Curtiss, D.R. Vissers, K. Amine, J. Electrochem. Soc. 151 (2004) A178, http://dx.doi.org/10.1149/1.1633765.
[30] W. Wang, Q. Luo, B. Li, X. Wei, L. Li, Z. Yang, Adv. Funct. Mater. 23 (2013) 970–986, http://dx.doi.org/10.1002/adfm.201200694.
[31] F.R. Brushett, J.T. Vaughey, A.N. Jansen, Adv. Energy Mater. 2 (2012) 1390–1396, http://dx.doi.org/10.1002/aenm.201200322.
[32] L. Zhang, Z. Zhang, P.C. Redfern, L.a. Curtiss, K. Amine, Energy Environ. Sci. 5 (2012) 8204, http://dx.doi.org/10.1039/c2ee21977h.
[33] A.V. Marenich, C.J. Cramer, D.G. Truhlar, J. Chem. Theory Comput. 9 (2013) 609–620, http://dx.doi.org/10.1021/ct300900e.

[34] D.S. Palmer, A. Llinàs, I. Morao, G.M. Day, J.M. Goodman, R.C. Glen, et al., Mol. Pharm. 5 (2007) 266–279, http://dx.doi.org/10.1021/mp7000878.
[35] A. Klamt, B. Mennucci, J. Tomasi, V. Barone, C. Curutchet, M. Orozco, et al., Acc. Chem. Res. 42 (2009) 489–492, http://dx.doi.org/10.1021/ar800187p.
[36] L.-P. Wang, T. Van Voorhis, J. Chem. Theory Comput. 8 (2012) 610–617, http://dx.doi.org/10.1021/ct200340x.
[37] J. Tomasi, B. Mennucci, R. Cammi, Chem. Rev. 105 (2005) 2999–3093, http://dx.doi.org/10.1021/cr9904009.
[38] F. Zeller, M. Zacharias, J. Phys. Chem. B (2014), http://dx.doi.org/10.1021/jp5015934.
[39] C.J. Cramer, D.G. Truhlar, Chem. Rev. 99 (1999) 2161–2200.
[40] J. Tomasi, B. Mennucci, E. Cancès, J. Mol. Struct. Theochem 464 (1999) 211–226, http://dx.doi.org/10.1016/S0166-1280(98)00553-3.
[41] S.P. Ong, G. Ceder, Electrochim. Acta 55 (2010) 3804–3811, http://dx.doi.org/10.1016/j.electacta.2010.01.091.
[42] Y. Shao, L.F. Molnar, Y. Jung, J. Kussmann, C. Ochsenfeld, S.T. Brown, et al., Phys. Chem. Chem. Phys. 8 (2006) 3172–3191, http://dx.doi.org/10.1039/b517914a.
[43] J.P. Perdew, K. Burke, M. Ernzerhof, Phys. Rev. Lett. 77 (1996) 3865–3868, http://dx.doi.org/10.1103/PhysRevLett. 77.3865.
[44] S. Grimme, J. Antony, S. Ehrlich, H. Krieg, J. Chem. Phys. 132 (2010) 154104, http://dx.doi.org/10.1063/1.3382344.
[45] M.R. Wright, An Introduction to Aqueous Electrolyte Solutions, John Wiley & Sons, England, 2007.
[46] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proc. ICNN'95 – Int. Conf. Neural Networks, IEEE, 1995, pp. 1942–1948. doi:10.1109/ICNN.1995.488968.
[47] V. Namasivayam, R. Günther, Chem. Biol. Drug Des. 70 (2007) 475–484, http://dx.doi.org/10.1111/j.1747-0285.2007.00588.x.
[48] M. Sharafi, T.Y. ELMekkawy, Renew. Energy 68 (2014) 67–79, http://dx.doi.org/10.1016/j.renene.2014.01.011.
[49] R. Poli, J. Kennedy, T. Blackwell, Swarm Intell. 1 (2007) 33–57, http://dx.doi.org/10.1007/s11721-007-0002-0.
[50] F. Wang, P.L.H. Yu, D.W. Cheung, Expert Syst. Appl. 41 (2014) 3016–3026, http://dx.doi.org/10.1016/j.eswa.2013.10.032.
[51] C.-E. Särndal, Stratified Sampling, in: Carl-Erik Särndal (Ed.), Springer, New York, 2003, pp. 100–109.
[52] M. Mitchell, An Introduction to Genetic Algorithms, MIT Press, Cambridge, MA, 1996.
[53] J.J.P. Stewart, J. Mol. Model. 19 (2013) 1–32, http://dx.doi.org/10.1007/s00894-012-1667-x.
[54] James J.P. Stewart, Stewart Computational Chemistry, MOPAC2012, 2014.
[55] J.D.C. Maia, G.A. Urquiza Carvalho, C.P. Mangueira, S.R. Santana, L.A.F. Cabral, G.B. Rocha, J. Chem. Theory Comput. 8 (2012) 3072–3081, http://dx.doi.org/10.1021/ct3004645.
[56] A. Bakker, S. Gejji, J.L. Lindgren, K. Hermansson, M.M. Probst, Contact ion pair formation and ether oxygen coordination in the polymer electrolytes M[N(CF_3SO_2)$_2$]2PEOn for M = Mg, Ca, Sr and Ba, Polymer 36 (1995) 4371–4378.
[57] A. Wahab, S. Mahiuddin, G. Hefter, W. Kunz, B. Minofar, P. Jungwirth, J. Phys. Chem. B 109 (2005) 24108–24120, http://dx.doi.org/10.1021/jp053344q.
[58] S.H. Lapidus, N.N. Rajput, X. Qu, K.W. Chapman, K.A. Persson, P.J. Chupas, Phys. Chem. Chem. Phys. (2014), http://dx.doi.org/10.1039/c4cp03015j.
[59] F.B. van Duijneveldt, J.G.C.M. van Duijneveldt-van de Rijdt, J.H. van Lenthe, Chem. Rev. 94 (1994) 1873–1885, http://dx.doi.org/10.1021/cr00031a007.
[60] R.M. Balabin, J. Chem. Phys. 129 (2008) 164101, http://dx.doi.org/10.1063/1.2997349.
[61] SciFinder, Chemical Abstracts Service, Columbus, OH, 2014.
[62] Reaxys, Reed Elsevier Properties SA, New York, 2014.
[63] L.a. Curtiss, P.C. Redfern, K. Raghavachari, J.a. Pople, J. Chem. Phys. 109 (1998) 42–55, http://dx.doi.org/10.1063/1.476538.
[64] Y. Zhang, X. Xu, W.a. Goddard, Proc. Natl. Acad. Sci. USA 106 (2009) 4963–4968, http://dx.doi.org/10.1073/pnas.0901093106.
[65] Y. Zhao, D.G. Truhlar, J. Phys. Chem. A 109 (2005) 5656–5667, http://dx.doi.org/10.1021/jp050536c.
[66] A.D. Becke, J. Chem. Phys. 98 (1993) 1372–1377, http://dx.doi.org/10.1063/1.464304.
[67] Y. Zhao, D.G. Truhlar, Theor. Chem. Acc. 120 (2007) 215–241, http://dx.doi.org/10.1007/s00214-007-0310-x.
[68] C. Adamo, V. Barone, J. Chem. Phys. 110 (1999) 6158–6170, http://dx.doi.org/10.1063/1.478522.
[69] V.N. Staroverov, G.E. Scuseria, J. Tao, J.P. Perdew, J. Chem. Phys. 119 (2003) 12129–12137, http://dx.doi.org/10.1063/1.1626543.
[70] S. Grimme, T.O. Chemie, O.I.D.U. Münster, J. Comput. Chem. 27 (2006) 1787–1799, http://dx.doi.org/10.1002/jcc.20495.
[71] C. Møller, M.S. Plesset, Phys. Rev. 46 (1934) 618–622, http://dx.doi.org/10.1103/PhysRev. 46.618.
[72] G.D. Purvis, R.J. Bartlett, J. Chem. Phys. 76 (1982) 1910–1918, http://dx.doi.org/10.1063/1.443164.
[73] The Concise Encyclopedia of Statistics, Springer New York, New York, NY, 2008 http://dx.doi.org/10.1007/978-0-387-32833-1.
[74] A.J. Cohen, P. Mori-Sánchez, W. Yang, Chem. Rev. 112 (2012) 289–320, http://dx.doi.org/10.1021/cr200107z.
[75] A.A. Granovsky, Firefly Version 8.0.0, 2013.
[76] L.P. Hammett, J. Am. Chem. Soc. 59 (1937) 96–103, http://dx.doi.org/10.1021/ja01280a022.
[77] C.I. Bayly, P. Cieplak, W.D. Cornell, P.A. Kollman, J. Phys. Chem. 97 (1993) 10269–10280.