Improving molecular models: Generating high-quality quantum mechanics data and quantifying parameter uncertainty

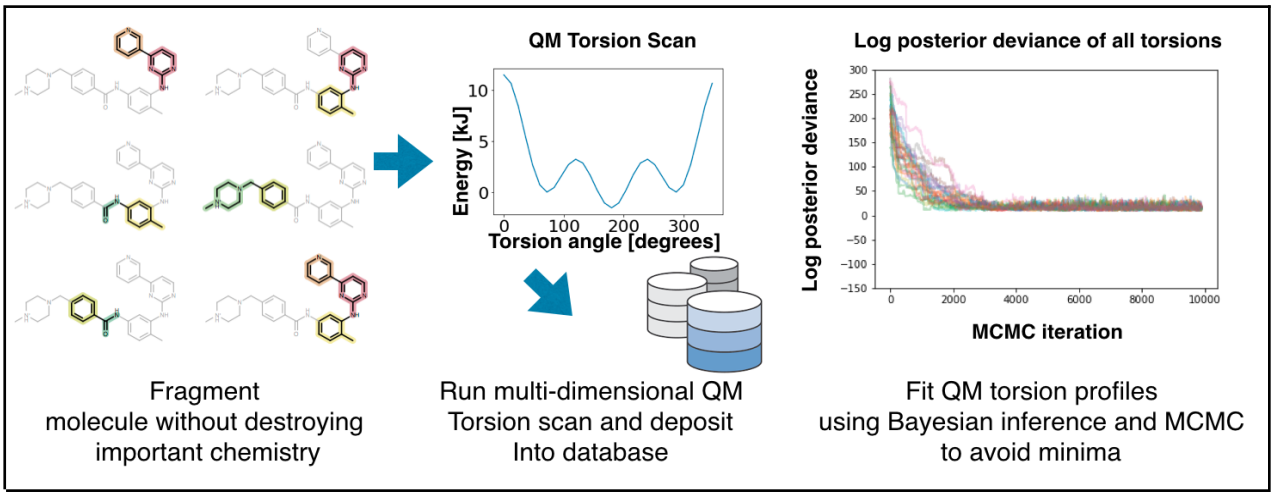
Chaya D. Stern

Background

Molecular mechanics (MM) forcefields are Newtonian physics-inspired atomistic models that are used in a wide variety of fields such as chemistry, biophysics, drug discovery and material science. Accurate small molecule MM forcefields are essential for predicting protein-drug binding and understanding the biophysics of biomolecular systems. Optimizing forcefields should improve the accuracy of our predictions. However, small molecule forcefields and automatic parameterization tools have lagged behind protein forcefield and simulation technologies. Current approaches require significant manual effort and expertise to extend the model coverage of chemical space; many parameters do not transfer well to new chemical structures, and we are unable to quantify the systematic model error in predictions derived from forcefields. My research project addresses all of these issues in the tractable subproblem of torsion parameters, which has traditionally been a difficult aspect of forcefield parameterization[1]. The software being developed for this work is part of the Open Forcefield Inititative[2], is fully open source, and is being designed for the larger molecular simulation community.

Torsion parameters

The forcefield models the potential energy of a system of molecules with additive functionals that describe the intra and intermolecular forces. The torsion energy functional describes the potential energy of a molecule as it rotates around its bonds and is commonly given as a sum of cosines. To model a new molecule, it is usually possible to use the existing parameters of most other bonded functionals based on chemical analogy, but torsion parameters will regularly need manual refitting with various levels of success.[1] I propose this happens for the following reasons: (A) The way quantum mechanics (QM) data are generated may introduce bias; (B) The Fourier series that is usually used to model the torsion energy functional is a highly degenerate model. The work required to address these issues can be roughly divided into two: 1) generating chemically relevant QM data and 2) automate the fitting of parameters and quantify their uncertainty. The figure below illustrates the workflow of generating QM data and continuously updating it using machine learning to find gaps in the data. The detail is described in part 1.



Part 1: Generating high quality QM torsion data

Torsion parameters are usually fit to computationally expensive (QM) torsion "scans" - geometry optimizations of molecules constrained to a grid of torsion angles. To reduce the computational cost and decrease the dimensions of the scan, the molecule is fragmented to smaller entities. However, since many electronic QM effects are non-local, fragmentation can lead to misrepresentation of the chemistry of the parent molecule. Knowing which bonds to fragment while retaining the correct chemistry and electronics currently requires expert knowledge. Having a tool that can encode this chemical perception when fragmenting molecules will reduce the need for an expert to decide how to generate the smallest possible fragments without disrupting the electronics of the bonds. To do that, I am writing **fragmenter**[3] a recursive algorithm that incorporates the Wiberg bond order, a value derived from the molecule's electron density that describes the electronic overlap between atoms in a bond and captures the non-local effects. As a fragment is recursively grown out of the central bond, the Wiberg bond order, together with other general rules can be used as criteria to decide which bonds to fragment.

The resulting fragments may have one rotational bond or many. Wang, L et. al.[4] have shown that a full rotation around a central bond may not return to the starting structure because of the potential relaxation of many

orthogonal degrees of freedom. To overcome this problem, Wang and colleagues are writing **crank**[5] a Python library that runs recursive torsion scans for n-dimensional grids. It will iteratively spawn new constrained geometry optimizations until no lower energy structures are found. I am collaborating with them to test the design and utility of this software.

The number of QM calculations needed to cover the chemical space of FDA approved drugs is around 50 million and to cover all of commercially available chemicals is around 88.5 billion. The results of these calculations need to be accessible via a "chemical identity" for them to be useful to the Open Forcefield initiative, physical simulation, cheminformatics and machine learning applications. A common chemical identity is the SMILES string, (Simplified Molecular-Input Line-Entry System), a line notation to describe the structure of a molecule. Existing tools exist to convert SMILES strings to graph representation of molecules. However, SMILES strings and indices of the nodes of the generated chemical graph are not unique. This poses a problem for the QM calculations and subsequent fitting because the 3-dimensional geometry of the graph is represented as a matrix. Together with Daniel Smith and the Molecular Sciences Software Institute (MolSSI) [6], we are working on extending **QCDB**[7], a quantum chemistry database, to run and deposit these calculations to a central repository and the QM-JSON schema[8] to include a chemical identity that retains the indexing of the nodes in the chemical graph.

Once the data are generated and accessible, the fragmenter and database will greatly reduce the manual and computational effort of generating new torsion parameters. While other semi-automatic parameter generators exist[9,10], our tool will be able to scale to large sets of molecules and will be able to be used internally by industry and academics labs hoping to design new materials or drugs. A new molecule will be automatically fragmented, then the fragments will be queried against the database to find if there are any gaps in the torsion scans already in the database. Then, only the missing torsions will have to be generated. For this to work, I will need to write a querying tool that will work on top of the database. Although the mapping of chemical graphs to SMILES is not unique, the reverse is. I will generate all possible SMILES string that can be associated with the fragment, and use that to query if torsion scans already exist. If we find a similar fragment, we will have to decide how similar is close enough to the queried fragment. This is a non-trivial problem. For the first iteration of this project, we will only use the database for exact chemical matches. In the future, to reduce computational cost, I will explore using different chemical similarity measures (that include Wiberg bond orders to capture non-local effects) to predict the QM torsion drives needed.

Part 2: Fitting torsion parameters

In many force fields, the torsion energy functional is given by Fourier series of the torsion angles. The force constant determines how high the barriers are to rotating around the central bond should be. Current fitting procedures are prone to getting trapped in local minima. Overcoming this is critical if we want to develop generalizable torsions models. In addition, selecting only the best fit out of many that fit the data equally well ignores the fact that predicted properties can be highly sensitive to torsion parameters[11]. Furthermore, we have no direct way to quantify the systematic uncertainty torsion parameters introduce to predicted properties.

To quantify uncertainties that arises from forcefield parameters, to avoid overfitting, and getting trapped in local minima, we cast the parameterization of force fields as a statistical inference problem and adopt a Bayesian probabilistic framework to automate the parameterization of torsion parameters from QM data. Given that the result of Bayesian inference, the posterior, is a probability distribution, we get a distribution of parameters that are consistent with the QM data that can then be used to estimate the uncertainty in computed properties due to parameter error. To propagate the parameter uncertainty, we compute properties, such as hydration or binding free energy with a reference torsion parameter set from the posterior and estimate the property for other sets from the collection by reweighting. The result is a distribution of the computed property where the width represents the error due to parameter uncertainty. I wrote a prototype of this framework using **PyMC**[12], a Python probabilistic programming language that makes it straightforward to define Bayesian models and sample posteriors. Initial results show that for moderately complex molecules, the systematic error introduced by torsion parameters is greater than the experimental error of computed properties. This framework will be scaled up to deal with larger datasets.

Future directions: Torsion parameter transferability

Once all these tools are in place, it will be possible to address torsion parameters' notoriety for their inability to transfer to seemingly similar molecules. Previous work by the Open Forcefield consortium on using chemical perception instead of atom types[13] will allow us to sample over large swaths of chemical space to find a generalizable set of parameters. In addition, we will be able to address the symmetry of the model since we will have the full posterior over a very large set of data. This data set will be highly valuable to artificial intelligence and machine learning applications to chemistry and drug discovery.

Reference:

1. Vanommeslaeghe, K, et. al.; (2010). *JCTC*, *31*, 671–90.
2. http://openforcefield.org/
3. https://github.com/openforcefield/fragmenter
4. Wang, L., et. al. (2017). *JPCB*
5. https://github.com/lpwgroup/cra
6. http://molssi.org/
7. MolSSI QCDB
8. https://github.com/MolSSI/QC_JSON_Schema
9. Mayne, C. G. et. al.(2013). *JCC* , *34*(32), 2757–70.
10. Huang, L., & Roux, B. (2013). *JCTC 9*, 3543–3556.
11. Mobley, D. L. et. al. (2008). *JPCB*, *112*(3), 938–946.
12. Salvatier J, et. al. (2016). *PeerJ Computer Science*
13. Mobley D. et. al. (2018) BioRxiv preprint (doi:10.1101/286542)