The Journal of Open Source Software

BESMARTS: A toolkit for data-driven force field design using binary-encoded SMARTS

Trevor Gokey [1]

1 Department of Chemistry, University of California, Irvine, USA

DOI: 10.21105/joss.06653

Software

- Review
- Repository
- Archive

Editor: Lucy Whalley

Reviewers:

- @AntObi
- @Ruibin-Liu

Submitted: 31 March 2024
Published: 09 April 2025

Summary

A popular method to model the potential energy of molecules is through the use of a force field. Force fields are a collection parameters that typically describe the bonded forces as harmonic spring bond stretching, angle bending, and sinusoidal torsion bending in addition to pairwise electrostatic and van der Waals interactions over non-bonded atoms. Each term requires one or more parameters, e.g. spring stiffness and length, and these must be assigned in some algorithmic way. Popular methods to assign parameters to molecules are generally opaque and difficult to extend. A relatively new method to assign parameters from Open Force Field uses molecular substructure queries called SMARTS to assign force field parameters. Using SMARTS patterns to assign parameters simplifies the task of extending force fields, but determining effective patterns has proven tedious and error prone. The goal of the BESMARTS package described here is to automate the process of searching SMARTS patterns to aid scientists to build better force fields.

Statement of need

BESMARTS is a Python package for manipulating SMARTS patterns using bitwise operations with the goal of distinguishing two or more groups of molecular substructures. This functionality is needed for force field design where each SMARTS substructure maps to force field parameters such as harmonic bond lengths and force constants. The package contains functions to generate patterns that are able split a set of molecules into two or more groups by directly iterating over the information in the SMARTS pattern. Previous effort tackled this problem by using Monte-Carlo sampling of SMARTS patterns until a given partition was satisfied, but was too computationally expensive for general use (Bannan & Mobley, 2019). BESMARTS implements a novel approach to finding SMARTS patterns by iterating through a given SMARTS pattern and generating all possible ways to split a group into two groups using a minimum amount of information in the patterns. This is accomplished by treating the SMARTS as graphs with the SMARTS information, called primitives such as element, connectivity, ring membership, etc, as bit vectors. Each bit represents a particular primitive value in the pattern, e.g. [#6,#7] represents "carbon or nitrogen". This package provides functionality to search for SMARTS patterns with a configurable search exhaustiveness, which controls the computational requirements of the exponential search in chemical space.

BESMARTS has already been used as a research tool in guiding experts in the development in the previously published Sage 2.1 force field (Behara et al., 2023). A more in-depth description of the algorithms and methodology of the BESMARTS package has been previously described (Gokey & Mobley, 2023). This package should be useful to any work requiring a clustering based on SMARTS patterns, or any physical model that needs to assign parameters to particular sets of atoms in a molecular system that can be described with SMARTS patterns. We find the most utility in this package for building and extending force fields, but we have previously shown its utility as a general clustering method to create hierarchies of chemical space (Gokey & Mobley, 2023).

Background

The simplified molecular-input line-entry system (SMILES) is popular in chemistry and related fields due to its human-friendly approach to transcribing molecules. From a SMILES string, SMARTS is a query language that determines whether a particular chemical group exists in the target molecule. These two languages together have enabled a wide range tools and applications.

The SMARTS language has been recently used as a method for parameter assignment for potential energy functions. The collection of parameters is called a force field, and are a staple in many areas of chemistry including drug binding assessment, free-energy calculations, and molecular dynamics of macromolecules such as proteins. A prerequisite to using a force field is determining how the parameters are assigned to the molecules. Traditionally, determining how the parameters are applied has been through the use of specialized, opaque programs that are coupled to their respective force field. For example, the general AMBER force field (GAFF) typically refers to both how parameters are assigned and the parameter values themselves. As mentioned above, a recent advancement is the use of SMARTS patterns to determine where parameters are assigned. Using SMARTS patterns opens up the black-box nature of parameter assignment, and extending a force field simply requires adding or modifying SMARTS patterns. The problem herein this approach is that writing effective SMARTS patterns for force field parameters is difficult, even for experts. Although a cheminformatician may be able to design a clever SMARTS pattern to distinguish an exotic group that is similar in the chemical sense, the cheminformatician must also design the pattern such that the chemistry grouped together is also similar in the physical sense. The Open Force Field (OpenFF) Consortium has defined the SMIRNOFF standard for a force field based on a hierarchy of SMARTS patterns (Mobley et al., 2018). Most of the SMARTS patterns were determined by experts, and extending the current set of SMARTS has proven tedious and error-prone. The difficulty in designing SMARTS patterns is compounded by the fact that the *order* of the patterns is also important in parameter assignment. Because a molecular substructure may match multiple SMARTS patterns in the hierarchy, the SMARTS patterns must be ordered in a way that each molecular substructure is assigned to its intended parameters.

Here, we describe a set of tools based on binary-encoded SMARTS (BESMARTS) that facilitate the search for SMARTS patterns in a data-driven, automatic approach. The core utility of `BESMARTS` is a set of tools to perform bitwise operation between two SMARTS patterns. This is done by treating the patterns as graphs, and the SMARTS information, called primitives, is embedded in the nodes (atoms) and edges (bonds). Additionally, to enable searching SMARTS, the graphs can be iterated in a bitwise fashion over the SMARTS primitives. Doing so generates new SMARTS by turning off one more of the iterated bits. Molecules which no longer match the SMARTS pattern with removed bits therefore is split off from the original SMARTS. This corresponds to creating a new parameter in a force field. Finally, in order to collect a set of SMARTS into a usable hierarchy for force field design, two SMARTS can be compared using set operations, specifically subset and superset relations. This is a necessary operation to define a hierarchy of SMARTS, where we intend to specialize a particular force field parameter by writing a new SMARTS pattern that carves out a subset of chemistry of the original SMARTS. The automated search enabled by the `BESMARTS` package and explicit formation of a SMARTS hierarchy using operations defined here lies at the heart of force field design using SMARTS, and we hope that this work has utility to the broader community of force field designers.

Acknowledgements

We thank David Mobley for advisement on this work. We thank Caitlin Bannan and Christopher Bayly for valuable discussions. We thank the Open Force Field Consortium and Initiative for financial and scientific support, and the Open Molecular Software Foundation (OMSF) for its support of the Open Force Field Initiative.

Funding

We appreciate financial support from the National Institutes of Health (R01GM132386 and R35GM148236) and computing support from the UCI Research Cyberinfrastructure Center, supported in part by NSF Grant CNS-1828779.

References

Bannan, C. C., & Mobley, D. (2019). ChemPer: An open source tool for automatically generating SMIRKS patterns. In *ChemRxiv*. Cambridge Open Engage. https://doi.org/10.26434/chemrxiv.8304578.v1

Behara, P. K., Gokey, Trevor, Cavinder, C., & Horton, J. (2023). Sage-2.1.0. In *GitHub repository*. GitHub. https://github.com/openforcefield/sage-2.1.0

Gokey, T., & Mobley, D. L. (2023). Hierarchical clustering of chemical space using binary-encoded SMARTS for building data-driven chemical perception models. In *ChemRxiv*. Cambridge Open Engage. https://doi.org/10.26434/chemrxiv-2023-v969f-v3

Mobley, D. L., Bannan, C. C., Rizzi, A., Bayly, C. I., Chodera, J. D., Lim, V. T., Lim, N. M., Beauchamp, K. A., Slochower, D. R., Shirts, M. R., & others. (2018). Escaping atom types in force fields using direct chemical perception. *Journal of Chemical Theory and Computation*, *14*(11), 6076–6092. https://doi.org/10.1021/acs.jctc.8b00640