kallisto: A command-line interface to simplify computational modelling and the generation of atomic features

Eike Caldeweyher*1

1 Data Science and Modelling, Pharmaceutical Science, R&D, AstraZeneca, Gothenburg, Sweden

DOI: 10.21105/joss.03050

Software
- Review
- Repository
- Archive

Editor: Richard Gowers
Reviewers:
- @rmeli
- @Sulstice

Submitted: 05 February 2021
Published: 15 April 2021

License
Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

Statement of Need

Machine learning (ML) has recently become very popular within pharmaceutical industry (Roy et al., 2015; Sprous et al., 2010). Tasks as, e.g., building predictive models, performing virtual screening, or predicting compound activities are potential use cases for such ML applications (Li et al., 2021; Simm et al., 2018). Traditionally, ML models often rely on the quantitative structure-activity relationship (QSAR) that has been popularized by medicinal chemists and statisticians to relate bioactivities to specific functional group manipulations (Dudek et al., 2006; Verma et al., 2010). This QSAR approach decreases the dimensionality of the underlying problem and projects the molecular structure into a space spanned by the physicochemical features. While early approaches relied more on linear regression, modern approaches combine such features with non-linear ML algorithms.

Chemoinformatic packages like RDKit (Landrum & others, 2006) enable the fast calculation of atomic/molecular features based on structural information like the molecular graph, while recently an extended Hueckel package has been added as well (Landrum, 2019). However, frequently we want to go beyond a structure-only approach thus incorporating electronic structure effects as obtained, e.g., by a (higher-level) quantum mechanical (QM) treatment. The calculation of QM-based features relies often on well-established quantum chemistry methods like Kohn-Sham density functional theory (DFT) that is currently the workhorse of computational chemistry (Kohn, 1999; Parr, 1980). However, generating the feature space by DFT is computationally demanding and can become the computational bottleneck especially when aiming for high-throughput experiments with several hundred to thousands of molecules.

Since there exists a critical need for an efficient yet accurate featurizer, we developed the kallisto command-line interface that is able to calculate QM-based atomic features for atoms and molecules efficiently (whole periodic table up to Radon). The features are either interpolating high-level references (e.g., static/dynamic polarizabilities with time-dependent DFT data) or are parametrized (Caldeweyher et al., 2019) to reproduce QM references (e.g., DFT Hirshfeld (Hirshfeld, 1977) atomic partial charges). Molecular geometries need to have an xmol or a Turbomole like format to be processed by kallisto. Besides, we implemented several computational modelling helpers to simplify the development of high-throughput procedures. Some of those modelling helpers depend on the open-source xtb tight-binding scheme that has been developed by Stefan Grimme and co-worker (Bannwarth et al., 2020). The kallisto software depends on the scientific libraries Numpy (Harris et al., 2020) and SciPy (Virtanen et al., 2020). The online documentation covers all high-level functionalizations of this software mostly in terms of copy-paste recipes. Furthermore, we cover bits of the underlying theory and compare to experimental data as well as to other modern deep learning models.

*Corresponding author
Atomic and Molecular Features

The following atomic and molecular features are available for all atoms up to Radon

- Coordination numbers (Caldeweyher et al., 2019; Grimme et al., 2010)
- Proximity shells
- Environment-dependent electronegativity equilibration partial charges (Caldeweyher et al., 2019)
- Environment- and charge-dependent dynamic polarizabilities (Caldeweyher et al., 2019; Grimme et al., 2010)
- Environment- and charge-dependent van-der-Waals radii (Fedorov et al., 2018; Mantina et al., 2009; Rahm et al., 2017)
- Sterimol descriptors (L, Bmin, Bmax) (Brethome et al., 2019)

Modelling Helpers

The following modelling helper are implemented

- Breadth first sorting
- Root mean squared deviation (quaternions) (Coutsias et al., 2004)
- Substructure identifier
- Substructure exchanger

Acknowledgements

EC acknowledges contributions from Philipp Pracht (@pprcht) and thanks Kjell Jørner (@kjelljørner) for sharing his Sterimol algorithm.

References


