The Journal of Open Source Software

Machine Learning Validation via Rational Dataset Sampling with astartes

Jackson W. Burns [1,2*¶], **Kevin A. Spiekermann** [2*], **Himaghna Bhattacharjee** [3], **Dionisios G. Vlachos** [3], and **William H. Green** [2]

1 Center for Computational Science and Engineering, Massachusetts Institute of Technology **2** Department of Chemical Engineering, Massachusetts Institute of Technology, United States **3** Department of Chemical and Biomolecular Engineering, University of Delaware, United States ¶ Corresponding author * These authors contributed equally.

DOI: 10.21105/joss.05996

Software

- Review
- Repository
- Archive

Editor: Arfon Smith

Reviewers:

- @du-phan
- @BerylKanali

Submitted: 16 October 2023
Published: 05 November 2023

Summary

Machine Learning (ML) has become an increasingly popular tool to accelerate traditional workflows. Critical to the use of ML is the process of splitting datasets into training, validation, and testing subsets that are used to develop and evaluate models. Common practice in the literature is to assign these subsets randomly. Although this approach is fast and efficient, it only measures a model's capacity to interpolate. Testing errors from random splits may be overly optimistic if given new data that is dissimilar to the scope of the training set; thus, there is a growing need to easily measure performance for extrapolation tasks. To address this issue, we report astartes, an open-source Python package that implements many similarity- and distance-based algorithms to partition data into more challenging splits. Separate from astartes, users can then use these splits to better assess out-of-sample performance with any ML model of choice. This publication focuses on use-cases within cheminformatics. However, astartes operates on arbitrary vector inputs, so its principals and workflow are generalizable to other ML domains as well. astartes is available via the Python package managers pip and conda and is publicly hosted on GitHub (github.com/JacksonBurns/astartes).

Statement of Need

Machine learning has sparked an explosion of progress in chemical kinetics (Komp et al., 2022; Spiekermann et al., 2022a), drug discovery (Bannigan et al., 2021; X. Yang et al., 2019), materials science (Wei et al., 2019), and energy storage (Jha et al., 2023) as researchers use data-driven methods to accelerate steps in traditional workflows within some acceptable error tolerance. To facilitate adoption of these models, researchers must critically think about several topics, such as comparing model performance to relevant baselines, operating on user-friendly inputs, and reporting performance on both interpolative and extrapolative tasks Spiekermann, Stuyver, et al. (2023). astartes aims to make it straightforward for machine learning scientists and researchers to focus on two important points: rigorous hyperparameter optimization and accurate performance evaluation.

First, astartes' key function train_val_test_split returns splits for training, validation, and testing sets using an sklearn-like interface. These splits can then separately be used with any chosen ML model. This partitioning is crucial since best practices in data science dictate that, in order to minimize the risk of hyperparameter overfitting, one must only optimize hyperparameters with a validation set and use a held-out test set to accurately measure performance on unseen data (Géron, 2019; Huyen, 2022; Lakshmanan et al., 2020; Ramsundar et al., 2019; Wang et al., 2020). Unfortunately, many published papers only mention training and testing sets but do not mention validation sets, implying that they optimize the hyperparameters to the test set, which would be blatant data leakage that leads to overly optimistic results. For researchers interested in quickly obtaining preliminary results without using a validation set to optimize hyperparameters, `astartes` also implements an `sklearn`-compatible `train_test_split` function.

Second, it is crucial to evaluate model performance in both interpolation and extrapolation settings so future users are informed of any potential limitations. Although random splits are frequently used in the cheminformatics literature, this simply measures interpolation performance. However, given the vastness of chemical space (Ruddigkeit et al., 2012) and its often unsmooth nature (e.g. activity cliffs), it seems unlikely that users will want to be restricted to exclusively operate in an interpolation regime. Thus, to encourage adoption of these models, it is crucial to measure performance on more challenging splits as well. The general workflow is: 1. Convert each molecule into a vector representation. 2. Cluster the molecules based on similarity. 3. Train the model on some clusters and then evaluate performance on unseen clusters that should be dissimilar to the clusters used for training. Although measuring performance on chemically dissimilar compounds/clusters is not a new concept (Bilodeau et al., 2023; Durdy et al., 2022; Heinen et al., 2021; Jorner et al., 2021; Meredig et al., 2018; Stuyver & Coley, 2022; Terrones et al., 2023; Tricarico et al., 2022), there are a myriad of choices for the first two steps; our software incorporates many popular representations and similarity metrics to give users freedom to easily explore which combination is suitable for their needs.

Example Use-Case in Cheminformatics

To demonstrate the difference in performance between interpolation and extrapolation, `astartes` is used to generate interpolative and extrapolative data splits for two relevant cheminformatics datasets. The impact of these data splits on model performance could be analyzed with any ML model. Here, we train a modified version of Chemprop (K. Yang et al., 2019)–a deep message passing neural network–to predict the regression targets of interest. We use the hyperparameters reported by Spiekermann et al. (2022a) as implemented in the `barrier_prediction` branch, which is publicly available on GitHub (Spiekermann, Pattanaik, et al., 2023). First is property prediction with QM9 (Ramakrishnan et al., 2014), a dataset containing approximately 133,000 small organic molecules, each containing 12 relevant chemical properties calculated at B3LYP/6-31G(2df,p). We train a multi-task model to predict all properties, with the arithmetic mean of all predictions tabulated below. Second is a single-task model to predict a reaction's barrier height using the RDB7 dataset (Spiekermann et al., 2022b, 2022c). This reaction database contains a diverse set of 12,000 organic reactions calculated at CCSD(T)-F12 that is relevant to the field of chemical kinetics.

For each dataset, a typical interpolative split is generated using random sampling. We also create two extrapolative splits for comparison. The first uses the cheminformatics-specific Bemis-Murcko scaffold (Bemis & Murcko, 1996) as calculated by RDKit (Landrum & others, 2006). The second uses the more general-purpose K-means clustering based on the Euclidean distance of Morgan (ECFP4) fingerprints using 2048 bit hashing and radius of 2 (Morgan, 1965; Rogers & Hahn, 2010). The QM9 dataset and RDB7 datasets were organized into 100 and 20 clusters, respectively. For each split, we create 5 different folds (by changing the random seed) and report the mean ± one standard deviation of the mean absolute error (MAE) and root-mean-squared error (RMSE).

Table 1: Average testing errors for predicting the 12 regression targets from QM9 (Ramakrishnan et al., 2014).

Split	MAE	RMSE
Random	2.02 ± 0.06	3.63 ± 0.21
Scaffold	2.20 ± 0.27	3.46 ± 0.49
K-means	2.48 ± 0.33	4.47 ± 0.81

Table 2: Testing errors in kcal/mol for predicting a reaction's barrier height from RDB7 (Spiekermann et al., 2022b).

Split	MAE	RMSE
Random	3.87 ± 0.05	6.81 ± 0.28
Scaffold	6.28 ± 0.43	9.49 ± 0.50
K-means	5.47 ± 1.14	8.77 ± 1.85

Table 1 and Table 2 show the expected trend in which the average testing errors are higher for the extrapolation tasks than they are for the interpolation task. The results from random splitting are informative if the model is primarily used in interpolation settings. However, these errors are likely unrealistically low if the model is intended to make predictions on new molecules that are chemically dissimilar to those in the training set. Performance is worse on the extrapolative data splits, which present a more challenging task, but these errors should be more representative of evaluating a new sample that is out-of-scope. Together, these tables demonstrate the utility of `astartes` in allowing users to better understand the likely performance of their model in different settings.

Several approaches could be taken to further reduce the errors presented here. One could pre-train on additional data or fine-tune with experimental values. Ensembling is another established method to improve model predictions.

Related Software and Code Availability

In the machine learning space, `astartes` functions as a drop-in replacement for the ubiquitous `train_test_split` from scikit-learn (Pedregosa et al., 2011). Transitioning existing code to use this new methodology is as simple as running `pip install astartes`, modifying an `import` statement at the top of the file, and then specifying an additional keyword parameter. `astartes` has been especially designed to allow for maximum interoperability with other packages, using few dependencies, supporting all platforms, and validated support for Python 3.7 through 3.11. Specific tutorials on this transition are provided in the online documentation for `astartes`, which is available on GitHub.

Here is an example workflow using `train_test_split` taken from the `scikit-learn` documentation (Pedregosa et al., 2011):

```
import numpy as np
from sklearn.model_selection import train_test_split

X, y = np.arange(10).reshape((5, 2)), range(5)

X_train, X_test, y_train, y_test = train_test_split(
  X, y, test_size=0.33, random_state=42)
```

To switch to using `astartes`, `from sklearn.model_selection import train_test_split` becomes `from astartes import train_test_split` and the call to split the data is nearly identical and simple in the extensions that it provides:

```python
import numpy as np
from astartes import train_test_split

X, y = np.arange(10).reshape((5, 2)), range(5)

X_train, X_test, y_train, y_test = train_test_split(
  X, y, test_size=0.33, sampler="kmeans", random_state=42)
```

With this small change, an extrapolative sampler based on k-means clustering will be used.

Inside cheminformatics, `astartes` makes use of all molecular featurization options implemented in `AIMSim` (Bhattacharjee et al., 2023), which includes those from virtually all popular descriptor generation tools used in the cheminformatics field.

The codebase itself has a clearly defined contribution guideline and thorough, easily accessible documentation. `astartes` uses GitHub actions for Constant Integration testing including unit tests, functional tests, and regression tests. To emphasize the reliability and reproducibility of `astartes`, the data splits used to generate Table 1 and Table 2 are included in the regression tests. Test coverage currently sits at >99%, and all proposed changes are subjected to a coverage check and merged only if they cover all existing and new lines added as well as satisfy the regression tests.

Acknowledgements

The authors thank all users who participated in beta testing and release candidate testing throughout the development of `astartes`. Authors Kevin Spiekermann and William Green gratefully acknowledge financial support from BASF under award number 88803720. Authors Jackson Burns and William Green gratefully acknowledge financial support from the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Department of Energy Computational Science Graduate Fellowship under Award Number DE-SC0023112. Authors Himaghna Bhattacharjee and Dionisios Vlachos contribution was primarily supported by the National Science Foundation under Grant No. 2134471

Disclaimer

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

References

Bannigan, P., Aldeghi, M., Bao, Z., Häse, F., Aspuru-Guzik, A., & Allen, C. (2021). Machine learning directed drug formulation development. *Advanced Drug Delivery Reviews*, *175*, 113806.

Bemis, G. W., & Murcko, M. A. (1996). The Properties of Known Drugs. 1. Molecular Frameworks. *Journal of Medicinal Chemistry*, *39*(15), 2887–2893. https://doi.org/10.1021/jm9602928

Bhattacharjee, H., Burns, J., & Vlachos, D. G. (2023). AIMSim: An accessible cheminformatics platform for similarity operations on chemicals datasets. *Computer Physics Communications*, *283*, 108579. https://doi.org/10.1016/j.cpc.2022.108579

Bilodeau, C., Kazakov, A., Mukhopadhyay, S., Emerson, J., Kalantar, T., Muzny, C., & Jensen, K. (2023). Machine learning for predicting the viscosity of binary liquid mixtures. *Chem. Eng. J.*, 142454. https://doi.org/10.2139/ssrn.4289793

Durdy, S., Gaultois, M. W., Gusev, V. V., Bollegala, D., & Rosseinsky, M. J. (2022). Random projections and kernelised leave one cluster out cross validation: Universal baselines and evaluation tools for supervised machine learning of material properties. *Digital Discovery*, *1*, 763–778. https://doi.org/10.1039/d2dd00039c

Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media, Inc.

Heinen, S., Rudorff, G. F. von, & Lilienfeld, O. A. von. (2021). Toward the Design of Chemical Reactions: Machine Learning Barriers of Competing Mechanisms in Reactant Space. *J. Chem. Phys.*, *155*(6), 064105. https://doi.org/10.1063/5.0059742

Huyen, C. (2022). *Designing machine learning systems: An iterative process for production-ready applications*. O'Reilly Media, Inc.

Jha, S., Yen, M., Salinas, Y., Palmer, E., Villafuerte, J., & Liang, H. (2023). Learning-assisted materials development and device management in batteries and supercapacitors: Performance comparison and challenges. *Journal of Materials Chemistry A*, *11*, 3904–3936.

Jorner, K., Brinck, T., Norrby, P.-O., & Buttar, D. (2021). Machine Learning Meets Mechanistic Modelling for Accurate Prediction of Experimental Activation Energies. *Chem. Sci.*, *12*(3), 1163–1175. https://doi.org/10.26434/chemrxiv.12758498

Komp, E., Janulaitis, N., & Valleau, S. (2022). Progress Towards Machine Learning Reaction Rate Constants. *Physical Chemistry Chemical Physics*, *24*, 2692–2705. https://doi.org/10.1039/d1cp04422b

Lakshmanan, V., Robinson, S., & Munn, M. (2020). *Machine learning design patterns: Solutions to common challenges in data preparation, model building, and MLOps*. O'Reilly Media, Inc.

Landrum, G., & others. (2006). *RDKit: Open-Source Cheminformatics*. https://www.rdkit.org

Meredig, B., Antono, E., Church, C., Hutchinson, M., Ling, J., Paradiso, S., Blaiszik, B., Foster, I., Gibbons, B., Hattrick-Simpers, J., Mehta, A., & Ward, L. (2018). Can machine learning identify the next high-temperature superconductor? Examining extrapolation performance for materials discovery. *Molecular Systems Design & Engineering*, *3*(5), 819–825. https://doi.org/10.1039/d1cp04422b

Morgan, H. L. (1965). The generation of a unique machine description for chemical structures-a technique developed at chemical abstracts service. *Journal of Chemical Documentation*, *5*(2), 107–113. https://doi.org/10.1021/c160017a018

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Ramakrishnan, R., Dral, P. O., Rupp, M., & Lilienfeld, O. A. von. (2014). Quantum Chemistry Structures and Properties of 134 Kilo Molecules. *Scientific Data*, *1*(1), 1–7. https://doi.org/10.1038/sdata.2014.22

Ramsundar, B., Eastman, P., Walters, P., & Pande, V. (2019). *Deep learning for the life sciences: Applying deep learning to genomics, microscopy, drug discovery, and more*.

O'Reilly Media, Inc.

Rogers, D., & Hahn, M. (2010). Extended-connectivity fingerprints. *Journal of Chemical Information and Modeling*, *50*(5), 742–754. https://doi.org/10.1021/ci100050t

Ruddigkeit, L., Van Deursen, R., Blum, L. C., & Reymond, J.-L. (2012). Enumeration of 166 Billion Organic Small Molecules in the Chemical Universe Database GDB-17. *Journal of Chemical Information and Modeling*, *52*(11), 2864–2875. https://doi.org/10.1021/ci300415d

Spiekermann, K. A., Pattanaik, L., & Green, W. H. (2022a). Fast predictions of reaction barrier heights: Toward coupled-cluster accuracy. *The Journal of Physical Chemistry A*, *126*(25), 3976–3986. https://doi.org/10.1021/acs.jpca.2c02614

Spiekermann, K. A., Pattanaik, L., & Green, W. H. (2022b). High Accuracy Barrier Heights, Enthalpies, and Rate Coefficients for Chemical Reactions. *Scientific Data*, *9*(1), 1–12. https://doi.org/10.1038/s41597-022-01529-6

Spiekermann, K. A., Pattanaik, L., & Green, W. H. (2022c). *High accuracy barrier heights, enthalpies, and rate coefficients for chemical reactions* (Version 1.0.1). Zenodo. https://doi.org/10.5281/zenodo.6618262

Spiekermann, K. A., Pattanaik, L., Green, W. H., Yang, K., Swanson, K., Jin, W., Coley, C., Eiden, P., Gao, H., Guzman-Perez, A., Hopper, T., Kelley, B., Mathea, M., & others. (2023). https://github.com/kspieks/chemprop/tree/barrier_prediction

Spiekermann, K. A., Stuyver, T., Pattanaik, L., & Green, W. H. (2023). Comment on "physics-based representations for machine learning properties of chemical reactions." *Machine Learning: Science & Technology*, *4*(4), 048001.

Stuyver, T., & Coley, C. W. (2022). Quantum Chemistry-Augmented Neural Networks for Reactivity Prediction: Performance, Generalizability, and Explainability. *The Journal of Chemical Physics*, *156*(8), 084104. https://doi.org/10.1063/5.0079574

Terrones, G. G., Duan, C., Nandy, A., & Kulik, H. J. (2023). Low-cost machine learning prediction of excited state properties of iridium-centered phosphors. *Chemical Science*, *14*, 1419–1433. https://doi.org/10.1039/d2sc06150c

Tricarico, G. A., Hofmans, J., Lenselink, E. B., Ramos, M. L., Dréanic, M.-P., & Stouten, P. F. (2022). Construction of balanced, chemically dissimilar training, validation and test sets for machine learning on molecular datasets. *10.26434/Chemrxiv-2022-M8l33*. https://doi.org/10.26434/chemrxiv-2022-m8l33-v2

Wang, A. Y.-T., Murdock, R. J., Kauwe, S. K., Oliynyk, A. O., Gurlo, A., Brgoch, J., Persson, K. A., & Sparks, T. D. (2020). Machine learning for materials scientists: An introductory guide toward best practices. *Chemistry of Materials*, *32*(12), 4954–4965. https://doi.org/10.1021/acs.chemmater.0c01907.s001

Wei, J., Chu, X., Sun, X.-Y., Xu, K., Deng, H.-X., Chen, J., Wei, Z., & Lei, M. (2019). Machine learning in materials science. *InfoMat*, *1*(3), 338–358.

Yang, K., Swanson, K., Jin, W., Coley, C., Eiden, P., Gao, H., Guzman-Perez, A., Hopper, T., Kelley, B., Mathea, M., & others. (2019). Analyzing Learned Molecular Representations for Property Prediction. *Journal of Chemical Information and Modeling*, *59*(8), 3370–3388. https://doi.org/10.1021/acs.jcim.9b00237.s001

Yang, X., Wang, Y., Byrne, R., Schneider, G., & Yang, S. (2019). Concepts of artificial intelligence for computer-assisted drug discovery. *Chemical Reviews*, *119*(18), 10520–10594.