qsarify: High performance machine learning software package for QSAR model development.

Stephen Szwiec Bakhtiyor Rasulev

Department of Coatings and Polymeric Materials, North Dakota State University, Fargo, ND

NDSU NORTH DAKOTA STATE UNIVERSITY

Abstract

qsarify is a new software package for the development, validation, and visualization of machine learning models for Quantitative Structure-Activity Relationship (QSAR) and Quantitative Structure-Property Relationship (QSPR) studies. Written in Python and freely available under the GNU General Public License, this software package provides a focused workflow for the generation of predictive statistical models to better explain and predict the relationship between molecular structure and biological activities or chemical properties. qsarify implements an innovative algorithm to reduce input dimensionality during the feature selection process, utilizing cophenetic clustering followed by a genetic algorithm (GA) for variable selection. Implementing both serial and parallel processing, this algorithm allows for rapid predictive model development and validation of large chemical datasets on low performance computers, while also allowing for complex model development and validation on high performance computers by utilizing multi-processing. Finally, the software also provides for output the of statistical validation metrics and generates plots for model diagnostics, including Williams Plot and Y-scrambling tests.

Background

QSAR and QSPR

- QSAR and QSPR models are used to predict the biological activity or chemical properties of a compound based on its molecular structure, respectively.
- As a data driven approach, QSAR and QSPR models are developed using a combination of molecular descriptors and machine learning (ML) statistical regression models.
- QSAR and QSPR modeling is a powerful tool for the prediction of activity and properties of compounds, and allows for high throughput screening to be performed.
- QSAR and QSPR models are used in a variety of fields, including drug discovery, environmental chemistry, and materials science.

Challenges

- The development of QSAR and QSPR models is a time consuming process.
- Chemical descriptor calculation results in a large number of variables, which can be computationally intensive to process.
- Researcher hardware is often limited, and the development of QSAR and QSPR models can be computationally intensive.
- Current software packages are often proprietary, and are not freely available.

Objectives

- Develop a Free and Open Source Software (FOSS) package for the development, validation, and visualization of QSAR and QSPR models.
- Scale to the hardware available
- Automate workflows
- Rapidly develop and validate models
- Visualize relationships in data

Methods

Dimensionality Reduction

- After processing the data, qsarify performs cophenetic clustering on descriptors [1], reducing data dimensionality.

Cophenetic Clustering

$$c = \frac{\sum_{i<j}^{n}[d(i,j) - \bar{d}]}{[\sum_{i<j} d(i,j)]^2} \quad \text{where} \quad d(i,j) = \frac{\text{cov}(\vec{x}_i, \vec{x}_j)}{\sigma_i \sigma_j} \tag{1}$$

1. Cophenetic correlation of the clustering based on the Euclidean distance d of the Pearson correlation of each descriptor, creating hierarchical clustering.

Methods

Genetic Algorithm

- In the second step, a genetic algorithm (GA) is used to select the most important variables from the clustered descriptors.
- Progressively, the bank of models is refined until the R^2 score is maximized.
- The system can be configured to use either single or multi-processing, using reflection.
- The GA is implemented using a map-filter-reduce paradigm, allowing for the parallelization of the GA.

Model validation

- The model is validated using the test set, and the R^2, Q^2, and $RMSE$ scores are calculated.
- The model is also validated using the Williams Plot and Y-scrambling tests.
- The model can also be further validated using a leave-one-out cross validation (LOOCV) test, and allows for additional QSAR-specific metrics (Q^2_{f3} and CCC) to be provided.

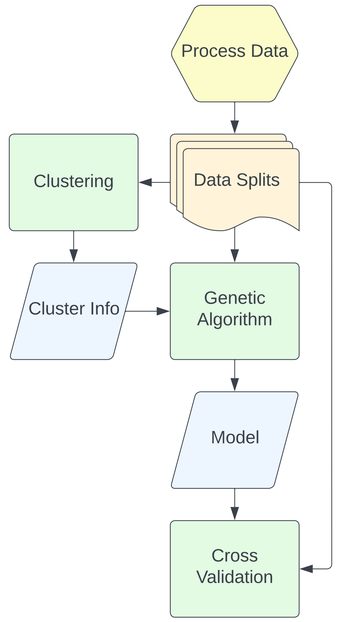

Figure 1. diagram of qsarify workflow

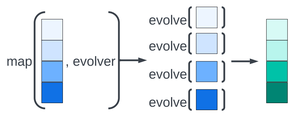

Figure 2. map function applies the evolutionary algorithm to a bank of models

Testing

Nitro-aromatic Toxicity Prediction

- The software was tested using a set of 28 nitro-aromatic compounds taken from literature [2].
- X: 676 descriptors were calculated using the `Dragon 6` software [3].
- Y: published $\log_{10}$ of the LD_{50} values for each compound.
- The qsarify model was trained using a set of 22 compounds, and tested using a set of 6 compounds.

Computational Benchmarking

- During model training, the software was benchmarked for speed and memory usage on:
  - HP EliteDesk 800 G5 SFF Mini Desktop PC
  - Intel Core i7-9700 @ 3.0 GHz
  - 16 GB RAM
  - Linux 6.0.12-artix1-1 x86_64
  - Python 3.10.9 (main, Dec 25 2022) [GCC 11.1.0]

Results

Model	Y = -1.453 * RDF070v - 96.40 * X5Av - 0.076
R^2	0.793
Q^2	0.710
RMSE	0.251

Table 1. Model performance statistics for an example two variable model.

Results

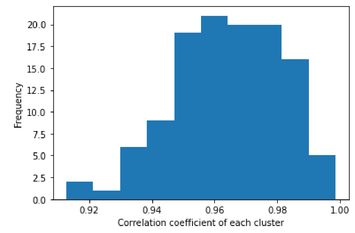

Figure 3. Histogram of autocorrelation coefficients of the clustering.

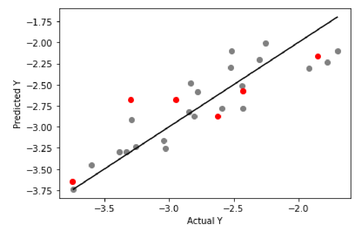

Figure 4. predicted vs actual values for training and testing.

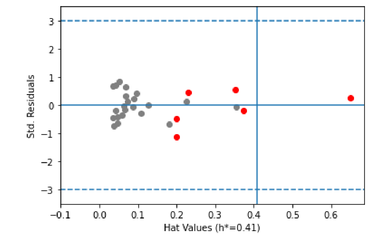

Figure 5. Williams plot for training and testing; figure shows the leverage and residuals of the model, giving an applicability domain.

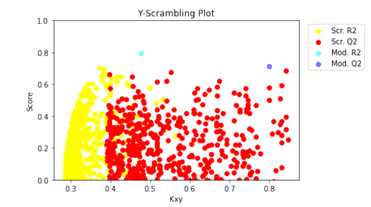

Figure 6. Y-scrambled plot for training and testing.

Computational Benchmarking

- **Single core performance:** 0.20 seconds per learning cycle
- **Multi-core performance:** 0.089 seconds per learning cycle
- **Profiled memory usage:** 4.2 GiB per learning cycle

Conclusions

- Modern machine learning techniques can be delivered in a simple, easy to use package.
- Low performance computers can be used to train models with high predictive power in a short amount of time by combining dimensional reduction with smart algorithms.
- Software freedom is important for the future of science, and the qsarify library is a step in that direction for computational chemistry.

Current Release

The current release of qsarify is available on PyPI and GitHub.

Acknowledgments

Authors acknowledge the support from the National Science Foundation under grant number NSF CHE-1800476. In this work the super-computing resources of the Center for Computationally Assisted Science and Technology (CCAST) at North Dakota State University were used, which was made possible in part by NSF MRI Award No. 2019077 and by the State of North Dakota.

References

(1) Rohlf, F. J.; Fisher, D. R. *Systematic Biology* **1968**, *17*, 407–412.
(2) Isayev, O.; Rasulev, B.; Gorb, L.; Leszczynski, J. *Molecular Diversity* **2006**, *10*, 233–245.
(3) Todeschini, R.; Consonni, V. DRAGON software for the Calculation of Molecular Descriptors, version 6 for windows,