Full Paper

Journal of the Brazilian Chemical Society

https://dx.doi.org/10.21577/0103-5053.20250063

J. Braz. Chem. Soc. **2025**, *36*, 8, e-20250063, 1-9

QSAR-Lit: A No-Code Platform for Predictive QSAR Model Development - From Data Curation to Virtual Screening

Igor H. Sanches,[a,b,c] Francisco L. Feitosa,[a,b,c] Jade M. Lemos,[a,b,c] Sabrina Silva-Mendonça,[a,b,c] Ester Souza,[a,b,c] Victoria F. Cabral,[a,b,c] José T. Moreira-Filho,[a] Henric Gil,[d] Bruno J. Neves,[d] Rodolpho C. Braga,[e] Joyce V. V. B. Borba[a,b,c] and Carolina H. Andrade[*,a,b,c]

[a]*Laboratório de Planejamento de Fármacos e Modelagem Molecular (LabMol), Faculdade de Farmácia, Universidade Federal de Goiás, 74605-170 Goiânia-GO, Brazil*

[b]*Centro para Pesquisas e Avanços em Fragmentos e Alvos Moleculares (CRAFT), Faculdade de Ciências Farmacêuticas de Ribeirão Preto, Universidade de São Paulo, 14040-903 Ribeirão Preto-SP, Brazil*

[c]*Centro de Excelência em Inteligência Artificial (CEIA), Instituto de Informática, Universidade Federal de Goiás, 74605-170 Goiânia-GO, Brazil*

[d]*Laboratório de Quimioinformática (LabChem), Faculdade de Farmácia, Universidade Federal de Goiás, 74605-170 Goiânia-GO, Brazil*

[e]*InsilicAll Ltda, 04363-090 São Paulo-SP, Brazil*

The development of predictive quantitative structure-activity relationship (QSAR) models using machine learning (ML) algorithms has become increasingly feasible due to the growing availability of chemical libraries with experimental data. These models can accelerate the drug discovery process and reduce failure rates by enabling data-driven decision-making. However, existing standalone software often lacks several critical components necessary for effective data preparation and modeling. Here, we introduce QSAR-Lit, an innovative, no-code, and comprehensive workflow designed for curating chemical and biological data, generating QSAR models, and performing virtual screening through an interactive Python-based Streamlit dashboard. The QSAR model development process begins with data curation, collecting and cleaning data on chemical structures and their biological activities. The next step is model building, where the curated data is used to train and optimize QSAR models. Finally, QSAR-Lit provides virtual screening, allowing QSAR models to predict the activity of new chemical structures. This application efficiently screens libraries of chemical compounds, assisting researchers in identifying and prioritizing potential candidates for further investigation.

Keywords: drug discovery, artificial intelligence, data curation, predictive modeling, machine learning, virtual screening

Introduction

The integration of predictive modeling techniques in drug discovery has become imperative due to the escalating complexity of biological systems and the vast array of chemical compounds available for exploration. Among the various methodologies employed, quantitative structure-activity relationship (QSAR) modeling[1,2] has gained significant traction as a robust approach for predicting the biological activity of new compounds based on their chemical structures. By utilizing extensive libraries of chemical data linked to experimental results, QSAR models empower researchers to make informed, data-driven decisions that can streamline the drug development process and mitigate the high failure rates typically associated with new drug candidates.[3]

In recent years, the capacity to gather, analyze, and store diverse types of chemical and biological data has rapidly evolved. Modern techniques, such as combinatorial chemistry and high-throughput/content screening (HTS/HCS),[4] have significantly contributed to the accumulation of

*e-mail: carolina@ufg.br
Editor handled this article: Paula Homem-de-Mello (Executive)

chemical and biological information within public databases. Currently, repositories like ChEMBL[5] and PubChem[6] provide the scientific community with access to extensive datasets that include thousands of chemicals evaluated in various biological assays.

Recent advancements in machine learning (ML) have further enhanced the effectiveness of QSAR modeling,[7,8] allowing for more accurate predictions and deeper insights into the relationships between molecular features and biological activity. For example, the research conducted by Stokes *et al*.[9] has shown that deep learning models, specifically directed message passing neural networks, can effectively capture the chemical complexity of compounds through graph representations. These models are capable of learning non-linear relationships within large datasets, resulting in enhanced virtual screening accuracy. For instance, they achieved a curve-area under the curve (ROC-AUC) score of 0.896 and played a key role in the discovery of halicin, a structurally divergent broad-spectrum antibiotic. However, despite these technological advancements, many existing software solutions in the field lack comprehensive frameworks for data preparation and effective modeling, which can hinder the ability of the researchers to produce reliable results.

Previously, we have developed an automated framework for the curation of chemogenomics data and to develop QSAR models for virtual screening using the open-source KNIME software.[10] Here we introduce QSAR-Lit, a novel and user-friendly workflow designed to facilitate the development of predictive QSAR models from data curation to virtual screening. QSAR-Lit employs an interactive Python-based Streamlit dashboard that streamlines the entire process, making it accessible even to those without extensive programming knowledge. This workflow encompasses three key components: dataset preparation and curation, classification and regression QSAR modeling and virtual screening. Figure 1 illustrates the various modules integrated within QSAR-Lit, providing a clear overview of its functionalities.

Methodology

QSAR-Lit was developed in Python 3.10,[11] employing the Streamlit[12] framework to provide an interactive, web-based interface for visualization and user interaction. The dataset curation module was implemented with ChEMBL[13] standardization library to enforce consistent molecular representation through normalization (correcting valence errors, charges, and atom types), neutralization (removing extraneous charges), mixture removal (discarding compound mixtures), and canonical tautomerization (resolving tautomeric ambiguities). Following curation, data splitting uses the built-in data split functions train_test_split and StratifiedKFold from Scikit-learn,[14] using the stratified method for binary classification only, allocating 20% of the dataset as a held-out test set and subjecting the remaining 80% to a 5-fold cross-validation procedure. The descriptor generation module involves calculating extended connectivity fingerprints[15] (ECFP) with RDKit,[16] providing standardized molecular representations for subsequent modeling. Model training was carried out using light gradient boosting machine (LightGBM),[17] support vector machines (SVM),[18] and random forest,[19] and hyperparameter optimization is conducted via Bayesian search.[20] The explored search spaces included parameters for LightGBM (num_leaves, learning_rate, n_estimators, max_depth, min_child_weight, subsample, colsample_bytree), random forest (max_features, n_estimators, max_depth, min_samples_leaf, min_samples_split), and SVM (C, kernel, gamma, degree, coef0, class_weight), ensuring thorough and data-driven tuning of model configurations to achieve robust and reproducible QSAR models.

Results and Discussion

Dataset preparation and curation module

Most errors in public databases often stem from measurement inaccuracies and insufficient quality control.

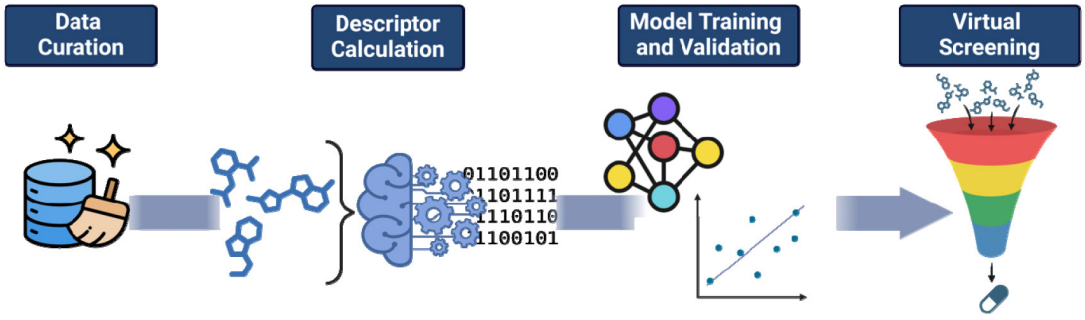

Figure 1. Workflow of the modules integrated in QSAR-Lit.

Proper chemical data curation forms the foundation of the process, enabling the identification and correction of structural inconsistencies.[21,22] Figure 2 shows the curation module from QSAR-Lit.

Input data

The input data must be in comma-separated values (CSV) format, select the column with the SMILES (simplified molecular input line entry system) strings and biological activity for each compound included (Figure 2a).

Curation steps

The "Standardize" button instantly triggers the curation processes, which include data normalization, neutralization, and the removal of mixtures, counter ions, and duplicates (Figure 2b). Alternatively, users can manually select which curation steps will be applied, choosing to execute all of them or only specific ones. After the selected steps are performed, a table is displayed, showing the molecules before and after curation for easy comparison.

Duplicates analysis

A dataset must contain structurally distinct compounds to be ready for modeling. However, a non-curated dataset may contain many instances of the same compound. The predictability of QSAR models will be exaggerated if modelers create datasets with structural duplication in both modeling and external sets. Therefore, before beginning any modeling study, duplicates must be found and eliminated. In this stage, the consistency and quality of the datasets are ensured by looking at the intra and inter-laboratory assay concordance between the duplicate records.[21,22] Duplicates are removed as follows: in the case of binary data, one duplicate entry is preserved in the dataset if the reported results of the duplicates are the same, and they are both eliminated if the reported results are different from each other. In the case of continuous data, (*i*) if the duplicate entries differ by more than 0.2 logarithmic units, both entries are discarded; (*ii*) if the difference in reported potencies is less than 0.2, an average of the values is determined, and one entry is kept in the dataset (Figure 2c).

Output data

The QSAR-Lit preparation and curation module gives as output (*i*) a file of standardized compounds without duplicates (e.g., before duplicate analysis); (*ii*) a report with duplicate analysis showing the number of input compounds, the number of compounds after standardization, the number of duplicated compounds, the number of discordant compounds, and the list of duplicated SMILES; (*iii*) a list of deleted duplicated SMILES; and (*iv*) a file of curated data (standardized without duplicated SMILES) (Figure 2c).

Molecular descriptors module

Molecular descriptors are numerical transformation from chemical structure in a symbolic representation that capture various aspects of the molecule.[23] Figure 3 shows the molecular descriptor module where the ECFP descriptors are calculated.[15] The input data must be the curated dataset in CSV format, with the SMILES and biological activity for each compound included (Figure 3b). The columns must be named, and the column with SMILES must be selected (Figure 3a). The user can adjust the radius (2-6) and the length of the bit vector (1024, 2048) to tailor the calculations to their specific needs. The molecular descriptors can be downloaded as a bit vector spreadsheet (Figure 3c).

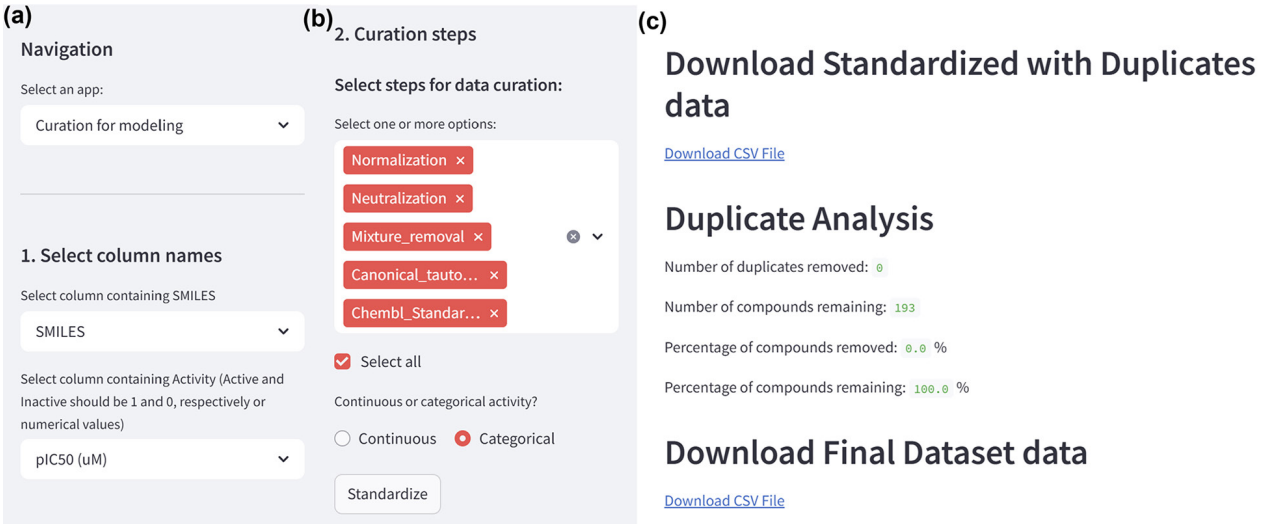

Figure 2. Curation module from QSAR-Lit. The user must select the column names and start performing the visual inspection.

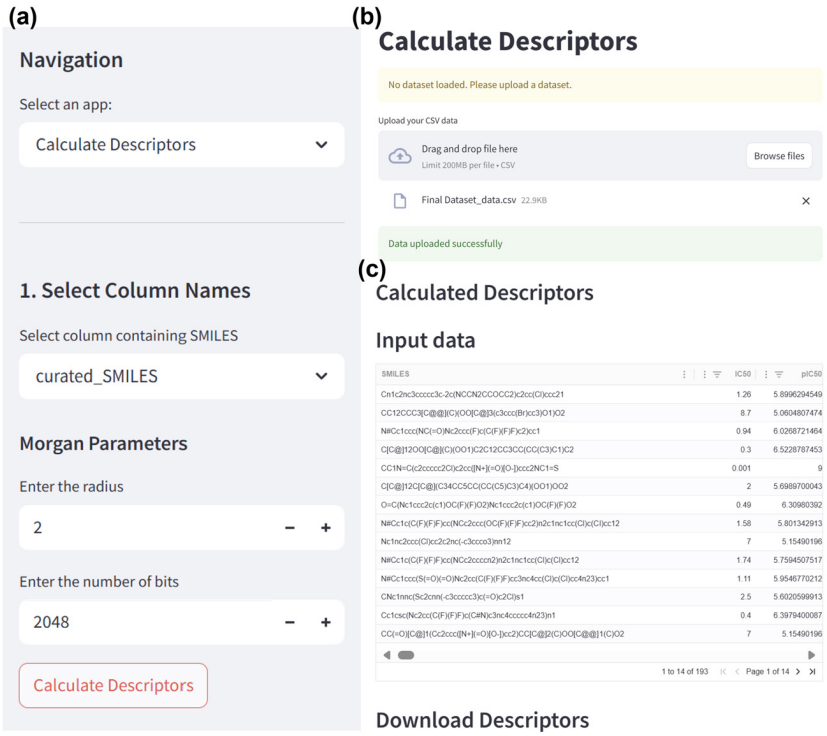

Figure 3. Molecular descriptors module from QSAR-Lit.

Machine learning modeling modules

ML algorithms

Three machine learning algorithms, SVM, RF, and LightGBM, are available for modeling both continuous (regression) and categorical (classification) data as modules in the sidebar of the main menu. Figure 4 shows the machine learning modeling module. The user can specify the number of iterations (n_iter) and the random seed (random_state) prior to training. Once configured, the modeling process can be initiated (Figure 4a).

Input data

After uploading the CSV descriptors file, the user must select specific columns to delete, thereby cleaning the dataset to ensure that the final set consists of only the molecule outcome and the fingerprint bits (Figure 4b). Following this, the user may select in the sidebar which column contains the biological outcome to ensure proper identification and consistency across different datasets. Then, the dataset is split to 5-fold cross-validation for hyperparameter optimization with a 20% external set for performance evaluation.

Performance of ML models

After completing the modeling process, users can review statistical metrics of the models in the external set. QSAR-Lit calculates key metrics for classification models, including positive predictive value (PPV), negative predictive value (NPV), sensitivity (Se), specificity (Sp), accuracy (ACC), Matthews correlation coefficient (MCC), and the area under the ROC curve (AUC). For regression models, it evaluates performance using metrics such as the mean absolute error (MAE), mean squared error (MSE), root mean squared error (RMSE), median absolute error (Median AE), coefficient of determination (R^2), and explained variance (Figure 4c).

Virtual screening module

Virtual screening (VS) is a computational approach widely employed in drug discovery and chemical biology to identify promising drug candidates from extensive molecular libraries. In this study, we utilized QSAR-Lit to implement ligand-based virtual screening (LBVS), applying QSAR models to predict the biological activity of novel, untested compounds.[24-26] This approach leverages known ligand data to prioritize molecules with potential therapeutic relevance, offering advantages such as cost-effectiveness, speed and efficiency, as well as giving insights into SAR for streamlining the identification of potential drug candidates.[27]

Input data

In the VS module, users are required to upload a dataset in CSV format and specify the column names

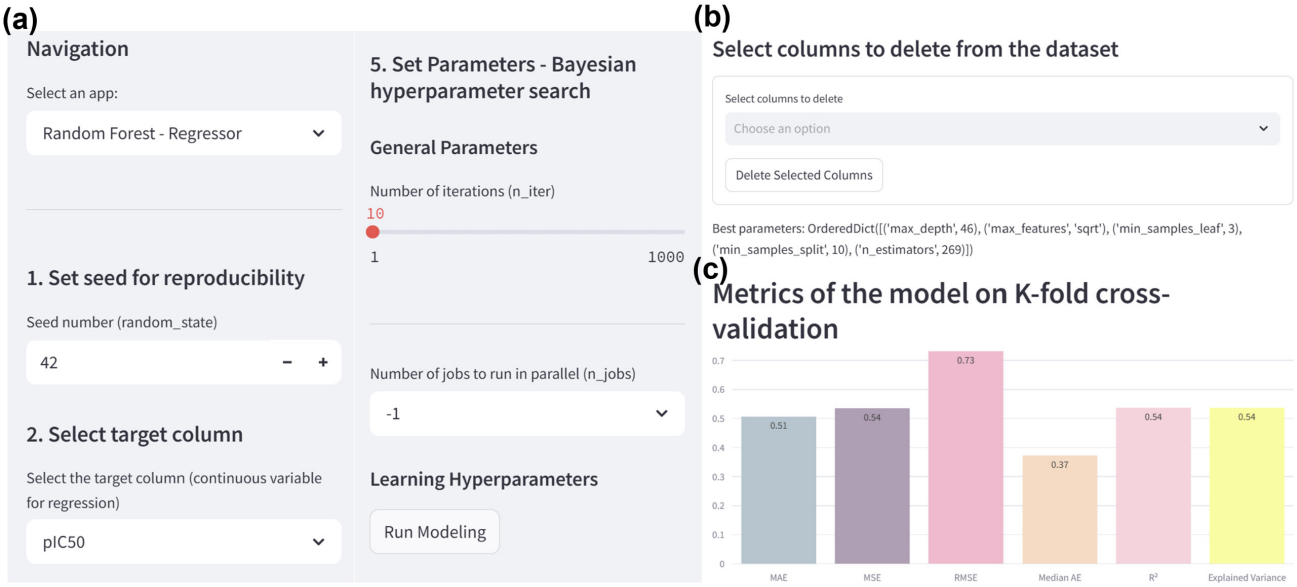

Figure 4. Machine learning modules from QSAR-Lit.

corresponding to the SMILES representations and the biological outcomes. Additionally, users must indicate whether the data is categorical or continuous. A pre-trained model file in PKL format must also be uploaded. Once these inputs are provided, users can initiate the process by pressing the "Run" button.

The server will automatically perform a standardization protocol, remove duplicates, and proceed with model predictions. It is important to note that this module is limited to processing batches of up to 1,000 molecules. For larger datasets, users are advised to install the QSAR-Lit platform locally, as detailed in the GitHub documentation (see section Data Availability Statement).

Output data

After processing the input compounds, the system generates a results table containing the predicted values of each compound and the associated predicted probabilities, ranging from 0.0 to 1.0. Any columns removed during preprocessing are reintegrated into the final output. Additionally, users have the option to download the results for further analysis.

Case study

To test the application, two datasets were selected: one categorical dataset from the Gene Tox database,[28] containing 1,456 compounds with AMES mutagenicity data, and one continuous dataset derived from ChEMBL,[5] containing 1,856 compounds with inhibitory activity data for *Plasmodium falciparum* 3D7 strain. These datasets were preprocessed by standardizing the IC_{50} (half maximal inhibitory concentration) values and converting them to pIC_{50} for the continuous data, as well as converting the IC_{50} values into categorical classes using a threshold of 10 μM. After this preprocessing, the datasets were inputted through the curation module within the application, remaining 820 compounds in categorical data and 1,625 in continuous data.

The curated datasets were used to build both classification and regression models, which were then evaluated using established statistical metrics through a 5-fold external cross-validation (5FECV) procedure. By selecting molecular descriptors and employing available algorithms within the QSAR-Lit web application, it was possible to generate models with acceptable predictive performance in both modalities.

Our models exhibit strong predictive reliability and robust performance, as indicated by their metrics aligning with the acceptable ranges established in the literature. For continuous data (Figure 5a), our top regression model achieved an R^2 value of 0.78, which signifies a good fit, with R^2 values above 0.6 widely recognized as reliable.[29,30] Furthermore, our model presented MSE and RMSE values of 0.48 and 0.69, respectively, fall within the ranges reported in similar studies,[30,31] highlighting superior predictive performance. In the case of categorical data (Figure 5b), our best classification model demonstrated a balanced accuracy (BACC) of 0.77, a sensitivity of 0.90, and a specificity of 0.63, all exceeding the threshold of 0.7, illustrating strong model effectiveness.[30] Additionally, both the Kappa coefficient and Matthews correlation coefficient values were 0.55, indicating moderate agreement and overall good model performance. These robust metrics, coupled with the implementation of a 5FECV

procedure, ensure that our models are not overfitted and are capable of generalize effectively to new data. This further confirms their robustness and reliability within the context of QSAR studies.[29,30]

Comparison with other freely available web tools

Several QSAR/QSPR (quantitative structure-property relationship) web tools, such as the OCHEM[32] (online chemical modeling environment), DPubChem,[33] ChemBench,[34] and DeepScreening,[35] have already been published. Those web tools are used to generate diverse descriptors, create models, and perform high-throughput virtual screening. Table 1 shows some important features of each of those web tools, compared with QSAR-Lit.

Each web tool comes with its own set of advantages and limitations. The OCHEM[32] web platform enables users to store data and develop their own QSAR/QSPR models without the need for a high-performance computer. Since its initial launch, the platform has been enhanced to offer a variety of descriptors, including 1D, 2D, 3D, and feature-based descriptors, as well as models that range from basic machine learning approaches to advanced deep neural networks, such as message-passing neural networks and transformer neural networks. Additionally, OCHEM provides users with options for selecting the type of validation, including stratified and bagging validation

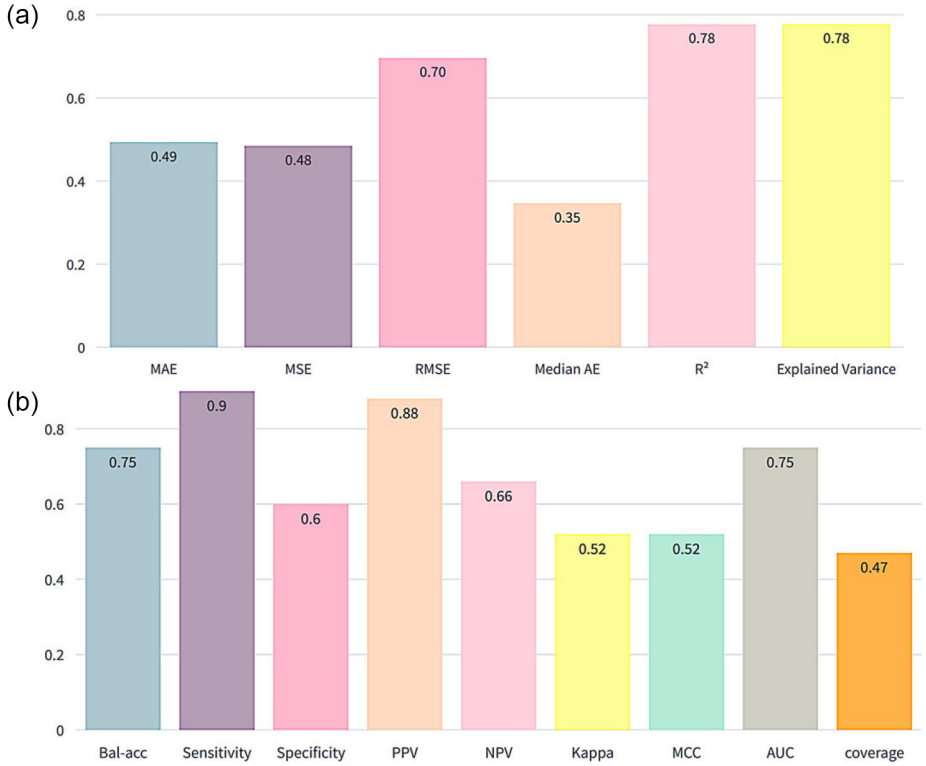

Figure 5. Predictive performance metrics of the best regression and classification models generated by the web application QSAR-Lit using the two selected datasets. (a) Mean absolute error (MAE), mean squared error (MSE), root mean squared error (RMSE), median absolute error (Median AE), the coefficient of determination (R^2), and explained variance. (b) Classification model: balanced accuracy (Bal-acc), sensitivity, specificity, positive predictive value (PPV), negative predictive value (NPV), Kappa, Matthews correlation coefficient (MCC), area under the curve (AUC), and coverage.

Table 1. Comparison of the main characteristics of each web tool for QSAR

Tool	OCHEM[32]	DPubChem[33]	ChemBench[34]	DeepScreening[35]	QSAR-Lit
Data curation	no	no	no	no	yes
SAR analysis	no	no	no	no	no
Feature selection	yes	yes	no	no	no
Modeling	yes	yes	yes	yes	yes
De novo design	no	no	no	yes	no
Virtual screening	no	yes	no	yes	yes

QSAR: quantitative structure-activity relationship.

methods. Users can also upload existing models to facilitate the creation of new models.

DPubChem,[33] which offers the functionality to create QSAR models and conduct high-throughput virtual screening, distinguishes itself from other servers by implementing a "class imbalance solution" that utilizes under sampling and oversampling techniques to address issues related to imbalanced datasets. In contrast, QSAR-Lit employs a calibration method that tackles these challenges without the need to add or remove data, ensuring a true representation of the chemical space of active compounds and their properties. However, DPubChem has a notable limitation: it can only operate with the PubChem bioassay using a single PubChem ID to develop its QSAR models. This constraint is a disadvantage because it restricts the potential to merge multiple PubChem IDs that share the same assay, incubation time, and other parameters, thereby limiting the number of compounds available for modeling a given endpoint.

DeepScreening,[35] launched in 2019, focuses on creating deep learning and *de novo* models for virtual screening. Notable features include the optimization of neural network hyperparameters to enhance manual learning. Additionally, DeepScreening is the only web tool that has ventured into implementing *de novo* library generation. Like other web tools, DeepScreening shares some differences with QSAR-Lit, as previously discussed. However, it also introduces a unique challenge by preparing data using a ten micromolar threshold for classification models. This approach can pose issues for certain endpoints, particularly target endpoints, where a lower micromolar threshold may be more appropriate for accurate classification.

Compared to our previous framework for curation and modeling using KNIME,[10] QSAR-Lit offers a user-friendly, no-code, Python-based platform that greatly simplifies the development of machine learning models. Unlike other applications, Streamlit[12] does not require any additional installations, plugins, or complex configurations, making it highly accessible for users. It can seamlessly operate in both local and cloud environments, offering flexibility for various use cases.

QSAR-Lit enables users to rapidly curate, train, and validate machine learning models without the need for extensive technical expertise. The platform further enhances data interpretation by generating intuitive visualizations, such as bar plots, which effectively convey results to users. Additionally, Streamlit facilitates the convenient export of results into individual CSV files, streamlining data sharing and reporting processes.

Conversely, KNIME necessitates a computer with moderate to robust computational capabilities, which may limit accessibility for users without high-performance hardware. Furthermore, the dependency of KNIME on nodes can lead to potential reliability issues, as these nodes may become deprecated or fail to function properly over time. These challenges position QSAR-Lit as a more efficient and adaptable alternative for machine learning workflows, catering to a wider range of users and ensuring greater reliability in model development.

Limitations and future improvements

While the QSAR-Lit tool offers a convenient interface and functionality for compound analysis, several limitations must be acknowledged to guide its improvement. Firstly, the execution of a high number of compounds on the server is constrained by the cloud hosting limitations of the Streamlit platform. For larger datasets, the tool must be run locally, with the necessary code available on GitHub. Additionally, to maintain the fluidity of the online version, hyperparameter adjustments are not available. Users who require model optimization for specific datasets are encouraged to use the local version for enhanced customization.

Currently, the system incorporates only one type of molecular descriptor calculation and supports three machine learning architectures. This limited variety may not be optimal for diverse datasets and modeling needs. Moreover, deep learning models, while planned for future implementation, are not yet part of the capabilities of the tool.

Another limitation lies in the data curation process, which adopts a general approach to ensure compatibility and accessibility. The curation steps are limited to the removal of duplicates and mixtures and the standardization of SMILES, leaving more specific and sophisticated cleaning procedures unaddressed.

Despite these limitations, the tool remains a valuable resource for users who need streamlined and accessible machine learning applications. However, further development is required to expand its scope, flexibility, and overall robustness.

Conclusions

The use of predictive QSAR models with machine learning algorithms represents a breakthrough in the field of drug discovery. Within this context, QSAR-Lit emerges as an innovative tool that effectively streamlines the entire workflow, from data curation to model building and virtual

screening, while ensuring accessibility through its no-code interface. This streamlined approach not only improves the efficiency of generating accurate QSAR models but also empowers researchers to make informed, data-driven decisions, ultimately expediting the identification of promising drug candidates.

Furthermore, the availability of QSAR-Lit through the LabMol InsightAI web portal and GitHub encourages collaboration and innovation among scientists, creating an environment conducive to exploration and discovery. As a powerful machine learning application, QSAR-Lit plays a pivotal role in drug discovery and toxicity research. By automating critical processes such as data curation and utilizing advanced ML techniques for QSAR modeling, it has the potential to significantly enhance both the efficiency and safety of developing new drug candidates. The workflows provided by this platform are freely accessible to the public, fostering collaboration and innovation within the scientific community and paving the way for future advancements in the field.

Data Availability Statement

The workflows are freely accessible through the LabMol InsightAI web portal (http://insightai.labmol.com.br/) and for download on GitHub repositories of (https://github.com/LabMolUFG/QSARlit).

Acknowledgments

This work has been funded by Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq, grant 440373/2022-0), Fundação de Amparo à Pesquisa do Estado de Goiás (FAPEG, grant 202010267000272) and CNPq BRICS Science, Technology and Innovation (STI) COVID-19 (grant 441038/2020-4). We also thank Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES, for financial support and fellowships, finance code 001). C. H. A. and B. J. N. are CNPq research productivity fellows.

Author Contributions

Carolina H. Andrade was responsible for funding acquisition, project design, and supervision; José T. Moreira-Filho, Henric Gil, and Bruno J. Neves for initiating the project; Igor H. Sanches and Francisco L. Feitosa for developing the framework and debugging the tool; José T. Moreira-Filho, Igor H. Sanches, and Rodolpho C. Braga for implementing the tool on the server; Sabrina Silva-Mendonça, Jade M. Lemos, Ester Souza and Joyce V. V. B. Borba for testing the application, conducting case studies, and performing comparisons with other tools; Victoria F. Cabral for the design and creating the figures. The manuscript was prepared through the contributions of all authors, under the guidance of Carolina H. Andrade. All authors reviewed and approved the final manuscript..

References

1. Tropsha, A.; *Mol. Inf.* **2010**, *29*, 476. [Crossref]
2. Cherkasov, A.; Muratov, E. N.; Fourches, D.; Varnek, A.; Baskin, I. I.; Cronin, M.; Dearden, J.; Gramatica, P.; Martin, Y. C.; Todeschini, R.; Consonni, V.; Kuz'min, V. E.; Cramer, R.; Benigni, R.; Yang, C.; Rathman, J.; Terfloth, L.; Gasteiger, J.; Richard, A.; Tropsha, A.; *J. Med. Chem.* **2013**, *57*, 4977. [Crossref]
3. Tropsha, A.; Isayev, O.; Varnek, A.; Schneider, G.; Cherkasov, A.; *Nat. Rev. Drug Discovery* **2024**, *23*, 141. [Crossref]
4. Raval, K. Y.; Kansagra, J. J.; Ganatra, T. H.; *Curr. Trends Pharm. Pharm. Chem.* **2022**, *4*, 120. [Crossref]
5. Mendez, D.; Gaulton, A.; Bento, A. P.; Chambers, J.; De Veij, M.; Félix, E.; Magariños, M. P.; Mosquera, J. F.; Mutowo, P.; Nowotka, M.; Gordillo-Marañón, M.; Hunter, F.; Junco, L.; Mugumbate, G.; Rodriguez-Lopez, M.; Atkinson, F.; Bosc, N.; Radoux, C. J.; Segura-Cabrera, A.; Hersey, A.; Leach, A. R.; *Nucleic Acids Res.* **2019**, *47*, 930. [Crossref]
6. Kim, S.; Chen, J.; Cheng, T.; Gindulyte, A.; He, J.; He, S.; Li, Q.; Shoemaker, B. A.; Thiessen, P. A.; Yu, B.; Zaslavsky, L.; Zhang, J.; Bolton, E. E.; *Nucleic Acids Res.* **2021**, *49*, 1388. [Crossref]
7. Muratov, E. N.; Bajorath, J.; Sheridan, R. P.; Tetko, I. V.; Filimonov, D.; Poroikov, V.; Oprea, T. I.; Baskin, I. I.; Varnek, A.; Roitberg, A.; Isayev, O.; Curtalolo, S.; Fourches, D.; Cohen, Y.; Aspuru-Guzik, A.; Winkler, D. A.; Agrafiotis, D.; Cherkasov, A.; Tropsha, A.; *Chem. Soc. Rev.* **2020**, *49*, 3525. [Crossref]
8. Soares, T. A.; Nunes-Alves, A.; Mazzolari, A.; Ruggiu, F.; Wei, G. W.; Merz, K.; *J. Chem. Inf. Model.* **2022**, *62*, 5317. [Crossref]
9. Stokes, J. M.; Yang, K.; Swanson, K.; Jin, W.; Cubillos-Ruiz, A.; Donghia, N. M.; MacNair, C. R.; French, S.; Carfrae, L. A.; Bloom-Ackermann, Z.; Tran, V. M.; Chiappino-Pepe, A.; Badran, A. H.; Andrews, I. W.; Chory, E. J.; Church, G. M.; Brown, E. D.; Jaakkola, T. S.; Barzilay, R.; Collins, J. J.; *Cell* **2020**, *180*, 688. [Crossref]
10. Neves, B. J.; Moreira Filho, J. T.; Silva, A. C.; Borba, J. V. V. B.; Mottin, M.; Alves, V. M.; Braga, R. C.; Muratov, E. N.; Andrade, C. H.; *J. Braz. Chem. Soc.* **2021**, *32*, 110. [Crossref]
11. *Python*, version 3.10; Python Software Foundation, Wilmington, USA, 2021.
12. *Streamlit*, version 1.42; Streamlit Inc., San Francisco, CA, USA, 2025.
13. Bento, A. P.; Hersey, A.; Félix, E.; Landrum, G.; Gaulton, A.; Atkinson, F.; Bellis, L. J.; De Veij, M.; Leach, A. R.; *J. Cheminf.* **2020**, *12*, 51. [Crossref]

14. *Scikit-learn*®, version 1.2.2; Scikit-learn Developers, 2023.
15. Rogers, D.; Hahn, M.; *J. Chem. Inf. Model.* **2010**, *50*, 742. [Crossref]
16. *RDKit*®, version 2024.09.5; RDKit Contributors, San Francisco, CA, USA, 2024.
17. *LightGBM*®, version 4.5.0; Microsoft Corporation, Redmond, USA, 2024.
18. Cortes, C.; Vapnik, V.; *Mach. Learn.* **1995**, *20*, 273. [Crossref]
19. Rigatti, S. J.; *J. Insur. Med.* **2017**, *47*, 31. [Crossref]
20. Snoek, J.; Larochelle, H.; Adams, R. P.; *arXiv* **2012**. [Crossref]
21. Fourches, D.; Muratov, E.; Tropsha, A.; *J. Chem. Inf. Model.* **2016**, *56*, 1243. [Crossref]
22. Fourches, D.; Muratov, E.; Tropsha, A.; *J. Chem. Inf. Model.* **2010**, *50*, 1189. [Crossref]
23. Todeschini, R.; Consonni, V.; *Handbook of Molecular Descriptors*, 1st ed.; Wiley: Weinheim, DE, 2000.
24. Alvarez, J.; Shoichet, B.; *Virtual Screening in Drug Discovery*, 1st ed.; CRC Press: Boca Raton, USA, 2005. [Crossref]
25. Stahura, F.; Bajorath, J.; *Curr. Pharm. Des.* **2005**, *11*, 1189. [Crossref]
26. Bhunia, S. S.; Saxena, M.; Saxena, A. K. In *Biophysical and Computational Tools in Drug Discovery*; Saxena, A. K., ed.; Springer: Cham, DE, 2021. [Crossref]
27. Kitchen, D. B.; Decornez, H.; Furr, J. R.; Bajorath, J.; *Nat. Rev. Drug Discovery* **2004**, *3*, 935. [Crossref]
28. Cimino, M. C.; Auletta, A. E.; *Mutagenesis* **1993**, *8*, 163. [Crossref]
29. Shayanfar, S.; Shayanfar, A.; *BMC Chem.* **2022**, *16*, 63. [Crossref]
30. Roy, P. P.; Roy, K.; *QSAR Comb. Sci.* **2008**, *27*, 302. [Crossref]
31. Bosc, N.; Atkinson, F.; Felix, E.; Gaulton, A.; Hersey, A.; Leach, A. R.; *J. Cheminf.* **2019**, *11*, 4. [Crossref]
32. Sushko, I.; Pandey, A.; Novotarskyi, S.; Körner, R.; Rupp, M.; Teetz, W.; Brandmaier, S.; Abdelaziz, A.; Prokopenko, V.; Tanchuk, V.; Todeschini, R.; Varnek, A.; Marcou, G.; Ertl, P.; Potemkin, V.; Grishina, M.; Gasteiger, J.; Baskin, I.; Palyulin, V.; Radchenko, E.; Welsh, W.; Kholodovych, V.; Chekmarev, D.; Cherkasov, A.; Aires-de-Sousa, J.; Zhang, Q. Y.; Bender, A.; Nigsch, F.; Patiny, L.; Williams, A.; Tkachenko, V.; Tetko, I.; *J. Cheminf.* **2011**, *3*, 20. [Crossref]
33. Soufan, O.; Ba-alawi, W.; Magana-Mora, A.; Essack, M.; Bajic, V. B.; *Sci. Rep.* **2018**, *8*, 9110. [Crossref]
34. Walker, T.; Grulke, C. M.; Pozefsky, D.; Tropsha, A.; *Bioinformatics* **2010**, *26*, 3000. [Crossref]
35. Liu, Z.; Du, J.; Fang, J.; Yin, Y.; Xu, G.; Xie, L.; *Database* **2019**, *2019*, 104. [Crossref]

Submitted: December 6, 2024
Published online: April 11, 2025