PyMolSAR: a Python-based toolkit to predict the activity and property of small molecules

Rahul Avadhoot

A thesis

submitted in partial fulfillment of the

requirements for the degree of

Master of Science in Chemical Engineering

(Data Science)

University of Washington

2018

Committee:

David Beck

Elizabeth Nance

Program Authorized to Offer Degree:

Chemical Engineering

University of Washington

Abstract

PyMolSAR: a Python-based toolkit to predict the activity and property of small molecules

Rahul Avadhoot

Chair of the Supervisory Committee:
David Beck
Chemical Engineering

Machine learning is a powerful approach for generating Quantitative structure-activity relationships (QSAR) models to predict the property and biological activity of small molecules. However, building such models in Python is cumbersome for cheminformatics researchers as they must use several Python packages and undertake a sequence of modeling steps. For instance, use Python packages for calculating molecular descriptors and generating models. Therefore, a Python toolkit that integrates these Python packages and modeling steps will immensely benefit cheminformatics researchers.

This work presents a Python toolkit, called PyMolSAR for building predictive structure-activity relationships models for small molecules. The functionality of PyMolSAR includes calculating 759 1D/2D molecular descriptors, data preprocessing, feature selection, training and evaluating predictive models. It is open-source and freely available on GitHub at https://github.com/BeckResearchLab/PyMolSAR.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1: BACKGROUND

Cheminformatics is an interdisciplinary field which applies computer and informational techniques to solve chemistry problems[1,2]. Cheminformatics techniques are widely used for drug discovery and design. The increasing availability of open-source molecular datasets, powerful computing resources, and the development of novel machine learning algorithms have helped cheminformatics research flourish in recent times.[3] One of the applications of cheminformatics is to build models for predicting the properties and activities of molecules given its structure.[4,5]

Quantitative structure–activity relationship (QSAR) models are used in chemical and biological sciences to relate the structural features of a molecule to its biological activity. These models lay out a relationship between molecular features and the activity in a large biological dataset and can predict the activities of new molecules.[6,7] Machine learning is the science that enables computers to automatically learn without being explicitly programmed. Machine learning is a powerful approach for building QSAR models[,8,9,10] and has been used for the prediction of melting points[11], carcinogenicities[12] and toxicities[13]. It has also been used in biomedical research for cancer diagnoses[14], drug design[15] and biodegradability[16].

However, there still exist few barriers to QSAR modeling. Developing predictive models in Python is cumbersome for cheminformatics researchers as they must use several Python packages and undertake a sequence of modeling steps like calculation of molecule descriptors, missing value imputation, feature scaling, feature selection, model training and hyperparameter optimization. To accomplish the above steps, the researcher must select and use different tools. Tools like RDKit[17], Chemopy[18] and OpenBabel[19] are available in Python to compute various

molecular descriptors. For training machine learning and deep learning models, various package like scikit-learn[20] and keras[21] have been implemented in Python. A comprehensive toolkit is now desirable to release such researchers from such tedious efforts.

In view of these limitations, we developed a Python package, called PyMolSAR, as a toolkit for predicting the activity of small molecules. PyMolSAR integrates other Python packages and allows users to complete the entire workflow via a step-by-step process. Currently, PyMolSAR is primarily intended for building predictive QSAR models and has the following functionality: (1) calculation of 759 molecular descriptors, (2) data preprocessing, (3) model building. These modules form an integrated pipeline for modeling, but each step can also be used independently.

Chapter 2: METHODS

2.1. DATA INPUTS

The datasets used in this project was obtained from various publicly available cheminformatics datasets and consists of two columns: the molecular structures of compounds in the form of simplified molecular-input line-entry system (SMILES)[22] along with their target activities. The SMILES strings can be converted back into 2-D and 3-D representations of the molecules. The target activities are continuous variables and are used to build regression models and the target classes are categorical variables and are used to build classification models.

2.2. OVERALL IMPLEMENTATION

In this project, the datasets were processed using Python 3.5, and the repository containing the scripts and instructions for the entire project is located on GitHub[23].

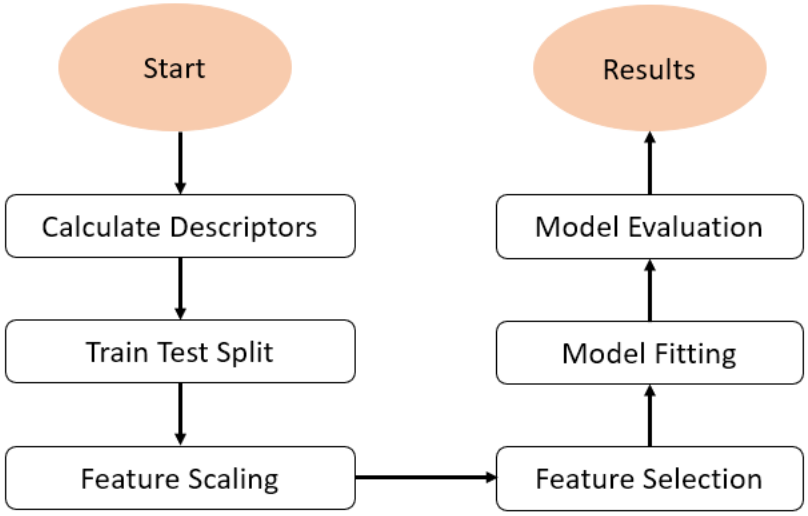

Figure 1: Data Pipeline for Model Training

The datasets are read in using Pandas[24] a Python package providing fast, flexible, and expressive data structures. The activity data frame is then cleaned to remove unnecessary columns and header labels and retain only the SMILES strings and their respective activity scores. Python packages scikit-learn[20] and Keras[21] are used to build and evaluate the machine learning models. Visualization packages like matplotlib,[25] etc. are available for data visualization.

2.3. MOLECULAR DESCRIPTORS

Molecular descriptors are calculated features of a molecule that summarize our understanding of the molecular structure and activity.[26] Molecular descriptors continue to play a crucial role in building QSAR models, virtual high throughput screening, drug ADME/T prediction and the other drug discovery processes.[27]

Feature Group	Features	Number of Descriptors
Autocorrelation	Geary autocorrelation	32
	Moran autocorrelation	32
	Moreau-broto autocorrelation	32
Basak	Basak descriptors	21
Burden	Burden descriptors	64
CATS	CATS descriptors	150
Charge	Charge descriptors	25
Connectivity	Molecular connectivity descriptors	44
Constitution	Molecular constitution descriptors	30
E-state	E-state descriptors	237

Kappa	Kappa descriptors	7
MOE-type	MOE-type descriptors	60
Topology	Topology descriptors	25

Table 1: The list of molecular descriptors computed by PyMolSAR

In this project, we developed the molecular_descriptors sub-module, which allows users to calculate 759 molecular descriptors from 11 feature groups (see Table 2) using RDKit[17] which cover a relatively broad range of molecular properties. These molecular descriptors have been shown to have a reliable performance in characterizing molecular structures.

2.4. DATA PREPROCESSING

The data-preprocessing pipeline is built upon the scikit-learn package and consists of 4 preprocessing steps: imputation of missing values, standardizing the features (zero mean and unit variance) and univariate feature selection.

The data frames are checked for rows containing missing values. These are replaced with the median value of that column. The data is also standardized so that the individual features have zero mean and unit variance. We standardize the data because features having very high variances may make the model less responsive to other features with low variances. Also, it is found that gradient descent converges faster when the data is standardized. The training data is fit and standardized using a standard scaler from scikit-learn's preprocessing module, and this fit is used to transform the test set as well. The process of selection of relevant features from an available set through automated or knowledge-based methods is called feature selection. Features selection helps in removing irrelevant features, simplifies models for easy

interpretability, and improves the speed of model training.[28] The feature_selection sub-module works by selecting k best features based on univariate statistical tests.

2.5. MODELS

In PyMolSAR, we implemented five learning algorithms for classification and regression. Additionally, to get a set of good parameters, a grid search strategy is used by to optimize the hyperparameters. A 10-fold cross validation process was also implemented to assess the model performance for all algorithms. Users can decide which one to choose based on their needs and the dataset available.

Regularized regression models have linear least squares loss functions and a regularization term. Ridge regression performs L2 regularization where the penalty is the square of the magnitude of coefficients. On the other hand, Lasso regression performs L1 regularization where the penalty is the absolute value of the magnitude of coefficients. In both cases, if the coefficients have large values, the optimization function is penalized to reduce these coefficients closer to zero. Ridge regression does not shrink coefficients to zero and uses all features, while lasso does feature selection by shrinking some coefficients to zero.[29] They are used in this project due to the availability of hundreds of molecular descriptors and helps prevent overfitting. The regularization parameter alpha is optimized using grid search.

Neural nets algorithms are inspired by biological neurons for learning complex non-linear relationships. It can approximate the complex relationships between the molecular structure and its activity as a 'black box model', without any explicit knowledge. The weights of the connections between layers are adjusted iteratively, forming accurate and efficient networks.[30] In this work,

a single neural network was designed for all features by using two dense hidden layers. Grid search is employed to optimize parameters like *number of neurons*, the *optimizer* and the *number of epochs*.

Random forest (RF) uses an ensemble of classification trees where each tree is built using a different bootstrap sample of the data. In addition, each node is split using the best feature among a subset of features randomly chosen at that node instead of using the best split among all variables. Due to this randomness, the bias of the forest usually slightly increases, while the variance also decreases for the averaging strategy. These strategies make random forests have excellent performance in classification tasks.[31] To obtain a good random forest model for classification, a set of parameters should be selected. Among these parameters, the *number of trees* and *tree depth* must be selected based on the specific problem at hand. Grid search helps us to optimize these parameters and obtain the best possible random forest model.

Support Vector Machine (SVM) was proposed by Vapnik and his coworkers for classification and regression problems. SVM works by constructing hyperplanes in high-dimensional spaces. SVM has good statistical properties, which has made it a popular choice for applications in various fields.[32] To obtain a good SVM model, a set of parameters should be selected depending on the selection of kernel. Among these parameters, kernel type is the kernel type to be used in the SVM algorithm. It must be one of *linear*, *poly*, *rbf* and *sigmoid*. For this project, we only implemented the linear kernel for support vector classification and regression. The other hyper-parameters of the model are selected by grid search.

Naïve Bayes (NB) are based on Bayes' theorem with the assumption of independence between each pair of features. Despite their assumptions, Naïve Bayes have been used in many studies for drug discovery. In this module, we implement the Naïve Bayes algorithm using a Gaussian distribution as it is suitable for small molecular datasets.

The K-Nearest Neighbors algorithm (k-NN) is a non-parametric method that classifies an unknown sample based on a simple majority vote of its k nearest neighbors, where *k* is a constant specified by the user.[33] The best value of *k* value is highly dependent on the data and is automatically selected by performing grid search.

All classification algorithms, along with their pre-processing steps and parameter grid for grid search, are listed in Table 2.

Algorithm	**Parameter Grid**
Naïve Bayes	scoring = 'f1_weighed'
	cv = 10
K Nearest Neighbors	scoring = 'f1_weighed'
	cv = 10
	n_neighbors = 5, 8, 10
	weights = 'uniform', 'distance'
	p = 1, 2
	algorithm = 'auto', 'ball_tree', 'kd_tree'

Linear Support Vector Machine	scoring = ‘f1_weighed’
	cv = 10
	C = 1, 10, 100
	penalty = ‘l1’, ‘l2’
	loss = ‘hinged’, ‘squared_hinge’
Random Forest	scoring = ‘f1_weighed’
	cv = 10
	n_estimators = 10, 50
	criterion = ‘gini’, ‘entropy’
	max_features = ‘auto’, ‘sqrt’
Artificial Neural Networks	scoring = ‘f1_weighed’
	cv = 10
	epochs = 10, 50
	batch_size = 10,20
	optimizer = ‘SGD’, ‘Adam’, ‘rmsprop’

Table 2: The supported classification algorithms and related parameters

Similarly, all regression algorithms, along with their pre-processing steps and parameter grid for grid search, are listed in Table 3.

Algorithm	Parameter Grid
Ridge	scoring = ‘r2_score’
	cv = 10
Lasso	scoring = ‘r2_score’
	cv = 10
Elastic Net	scoring = ‘r2_score’
	cv = 10
Random Forest	scoring = ‘r2_score’
	cv = 10
	n_estimators = 10, 100
	criterion = ‘mse’, ‘mae’
	max_features = 1, 3, 10, ‘auto’, ‘sqrt’, ‘log2’
	min_samples_split = 2, 3, 10
	min_samples_leaf = 1, 3, 10
	oob_score = True, False
	max_depth = 3, None
Linear Support Vector Machine	scoring = ‘r2_score’
	cv = 10
	C = 1, 5, 10
	epsilon = 0, 0.1

Artificial Neural Networks	scoring = ‘r2_score’
	cv = 10
	epochs = 10, 50
	batch_size = 10,20

Table 3: The supported regression algorithms and related parameters

2.6. MODEL EVALUATION

After training several different models on the same data, the models are evaluated on the test set and various metrics are computed. The selected model can then be used to predict new samples next time. The metrics computed to compare different models are given below:

Mean absolute error (MAE): The average of the absolute differences between the true and predicted values.

Mean squared error (MSE): The average of the squares of the difference between the true and predicted values.

Median absolute error (MedAE): The median of all absolute differences between the true and predicted values.

R^2 score: The coefficient of determination is the proportion of the variance explained by the model.

Accuracy: The accuracy is the fraction of correct predictions among all predictions.

Precision, Recall and F1-Score:

Precision is the ability of the classifier not to label as positive a sample that is negative (i.e. how precise the predictions are), and recall is the ability of the classifier to find all the positive samples.

The F1-score is a harmonic mean of the precision and recall where recall and precision are equally important.

Receiver Operating Characteristic (ROC):

The receiver operating characteristic (ROC), or the ROC curve, is a plot between the true positive rate (sensitivity) and the false positive rate (1 - specificity) at various threshold settings. By computing the area under the roc curve, the curve information is summarized in one number.

Chapter 3: RESULTS AND DISCUSSION

This section looks through the results obtained on various publicly available cheminformatics datasets, analyzes those results and their impact, seeks sources of errors and ways to address them for better work in the future.

3.1 BLOOD BRAIN BARRIER PERMEABILITY

Optimization of pharmacokinetic and pharmacodynamic properties of a drug candidate is a crucial step in the drug design process.[34,35] These properties are needed to achieve sufficient drug concentration at the target site while possibly limiting its distribution elsewhere to reduce potential side effects. One important pharmacokinetic property of a drug is its ability or inability to penetrate the blood-brain barrier (BBB). The prediction of blood-brain barrier permeability is important for drugs targeting central nervous system disorders as they must penetrate the blood−brain barrier (BBB).[36-39]

Previous studies have employed computational methods to study BBB penetration and as potential prescreening tools with the aim to reduce the cost and enhance the speed of BBB penetration analysis.[40] The most widely explored computational methods are regression methods and statistical learning methods. In this work, 415 agents with known BB ratios (the ratio of the steady-state concentrations of a drug in the brain and blood) were selected.[41] The agents were divided into BBB+ and BBB- groups based on whether the BB ratio was >0.1 or <0.1, respectively. Under this definition, there are a total of 276 BBB+ and 139 BBB- agents. 759 molecular descriptors were calculated and used to predict BBB+ and BBB- agents using statistical and machine learning methods.

Model	Accuracy	Area under ROC	Precision	Recall	F1 Score
Naïve Bayes	0.70	0.78	0.55	0.47	0.51
K-Nearest Neighbors	0.75	0.84	0.63	0.63	0.63
Support Vector Machine	0.74	0.84	0.59	0.69	0.64
Random Forest	0.78	0.83	0.72	0.56	0.63
Artificial Neural Networks	0.74	0.79	0.59	0.69	0.64

Table 4: Comparison of the Model Metrics of the BBB-Penetrating (BBB+) and Nonpenetrating (BBB-) Agents by Different Statistical Learning Methods

Of the statistical and machine learning methods studied, Multilayer Perceptron (MLP) gives the highest prediction accuracy of 80.41% and K Nearest Neighbors gives the highest area under the ROC Curve of 0.84. For the other five methods tested in this work, their prediction accuracies and areas under their ROC curves are in the range of $70\sim78\%$ and $0.78\sim0.84$ respectively by using the selected set of descriptors. These results are consistent with the results from an earlier study of BBB+ and BBB- agents and other studies of different chemical and protein systems. This suggests that statistical and machine learning methods are capable of the prediction of BBB+ and BBB- agents at a comparable or perhaps better accuracy with respect to that from other classification methods without requiring either the knowledge of a mechanism or the intrinsic structure-activity relationships.

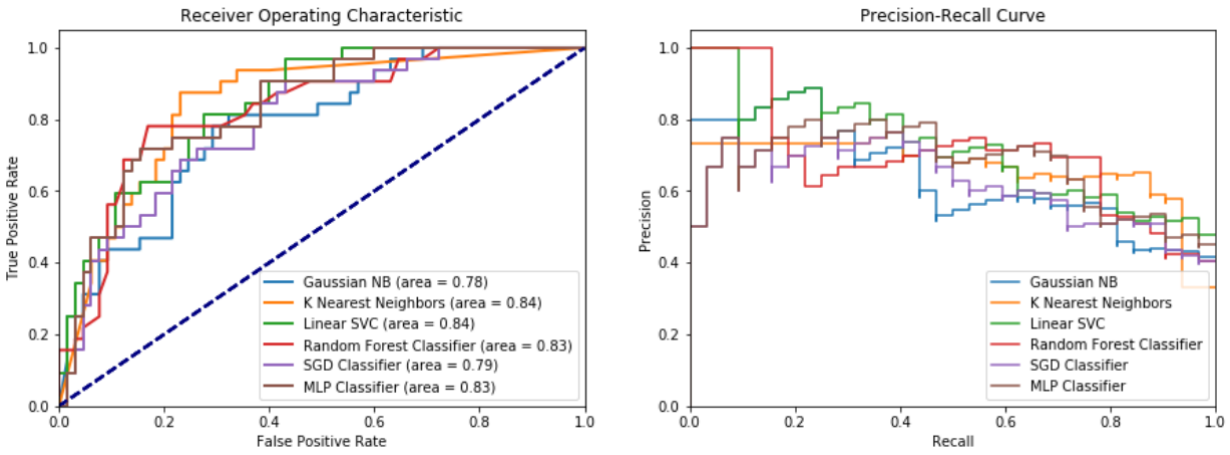

Figure 2: ROC Curves of classification models for Blood Brain Barrier Permeability

3.2 USP1 INHIBITION

USP1 inhibition is a promising approach for the pharmacological treatment of diseases like prostate and colon cancer as it prevents the upregulation of DNA repair pathways in cancer cells.[42,43] In a previous study to find better inhibitors for USP1, binary classification models were built using the structural descriptors for the compounds.[44] They used four machine learning techniques - Naive Bayes, Random Forests, Sequential minimal optimization and J48 - and predicted inhibition in a test set with ~80% accuracy. In another previous study[45], various regression prediction models were built and evaluated on the same dataset with moderate success. Though the mean absolute error (MAE), mean-squared error (MSE) and median absolute error (MedAE) values may be considered moderately accuracy, the models suffered from very low coefficient of determination scores.

The data was obtained from PubChem[46] and consists of the molecular structures of 389,560 compounds and their USP1-inhibition Activity scores, where a score between 40 and 100 was classified as active and a score of 0 was classified as inactive. Also, molecules with an activity

score of 1-39 were marked as inconclusive. Two datasets were created using this data: one dataset consisting of active and inactive compounds and another dataset consisting of active and inconclusive compounds. 759 molecular descriptors were calculated, and classifiers were built to classify molecules as active or inactive and active or inconclusive using statistical and machine learning methods.

Model	Accuracy	Area under ROC	Precision	Recall	F1 Score
Naïve Bayes	0.72	0.73	0.10	0.63	0.17
Support Vector Machine	0.74	0.78	0.12	0.73	0.20
Random Forest	0.96	0.83	0.63	0.16	0.25
Artificial Neural Networks	0.81	0.80	0.15	0.64	0.15

Table 5: Comparison of the Model Metrics of the Active vs Inconclusive Agents by Different Statistical Learning Methods

Of the statistical and machine learning methods studied, Random Forests gives the highest prediction accuracy and area under the ROC Curve of 80.35% and 0.84 respectively. For the other methods tested in this work, their prediction accuracies and areas under their ROC curves are in the range of $72\sim95\%$ and $0.72\sim0.84$ respectively by using the selected set of descriptors. These results are consistent with the results from the study described above.

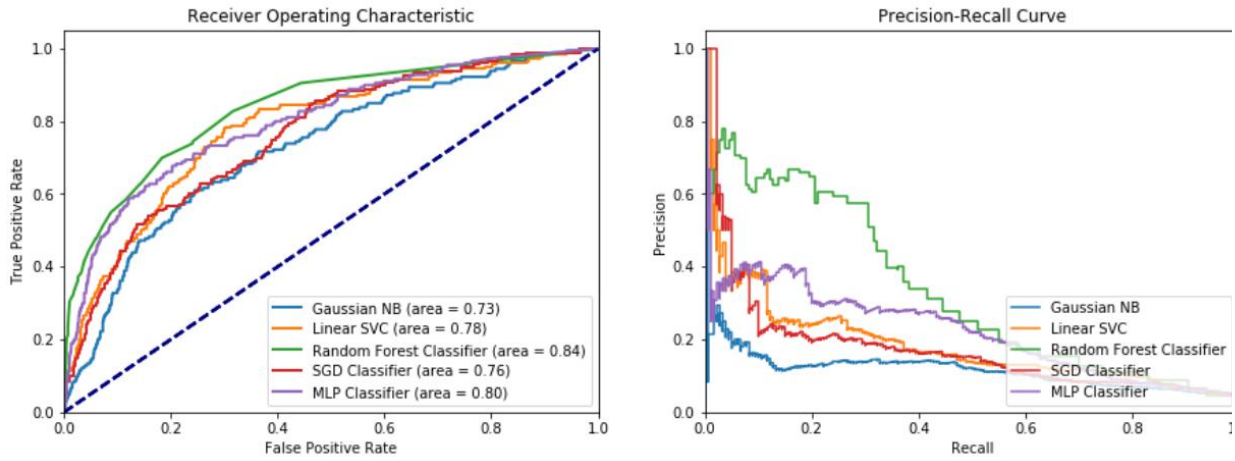

Figure 3: ROC Curves of classification models for USP Inhibition

3.3 SIGMA-2 RECEPTOR SELECTIVE LIGANDS

Sigma-2 (σ2) receptors are overexpressed in tumor cells and ligands that selectively bind to these receptors are currently under investigation for the diagnosis and treatment of cancer.[47] The quantitative prediction of the σ2 receptor pKi can be used for finding novel drugs that can bind with σ2 receptors. A previous study[48] developed QSAR models using the software CORAL[49]. The data was obtained from the Sigma-2 Receptor Selective Ligands Database (S2RSLDB)[50] which has data about σ2 receptor ligands and their pKi (–logKi) values. The data consists of 651 σ2 receptor ligands which was randomly split into a training dataset (520 compounds) for model training and a testing dataset (131 compounds) for model evaluation. 759 molecular descriptors were calculated and used to predict σ2 receptor pKi using statistical methods. The methods tested include Random Forest, Ridge Regression, Lasso Regression, Elastic Net Regressor and Support Vector Machine.

Model	Mean Squared Error	Mean Absolute Error	Median Absolute Error	R^2 Score
Random Forest	0.53	0.55	0.37	0.43
Ridge	0.73	0.69	0.58	0.22
Lasso	0.71	0.69	0.59	0.24
Elastic Net	0.71	0.70	0.59	0.24
SVM	0.78	0.71	0.57	0.17

Table 6: Comparison of the Model Metrics for predicting Sigma-2 Receptor Ligand binding by Different Statistical Learning Methods

Of the statistical and machine learning methods studied, Random Forests gives the highest coefficient of determination score and lowest mean-squared error of 0.43 and 0.53. For the other methods tested in this work, their coefficient of determination and mean-squared error are in the range of $0.17 \sim 0.24$ and $0.71 \sim 0.78$ respectively by using the selected set of descriptors. Other metrics are presented in Table 7.

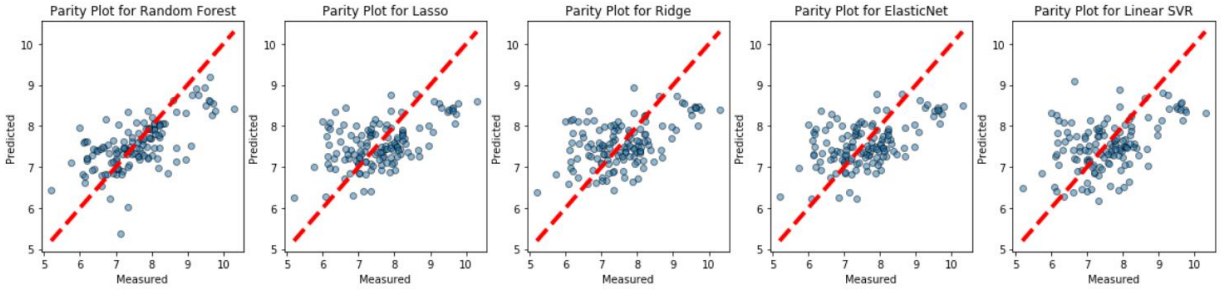

Figure 4: Parity Plots of regression models for Sigma-2 Receptor Selective Ligands

3.4. ANALYSIS OF RESULTS

By using PyMolSAR, one can conveniently build several models on the same data and make a comprehensive comparison to identify the best predictive model. To evaluate the predictive ability of our models, we compared our prediction results with the results obtained from recent research papers. Compared with these published models, our models have almost comparative performances. PyMolSAR can be used to obtain reliable and robust models for the evaluation of Blood-Brain Barrier Permeability, USP1 Inhibition and Sigma-2 Receptor Selective Ligands. Among all the learning algorithms used, the performance of Random Forests and Artificial Neural Networks is noteworthy. Random Forests and Artificial Neural Networks work equally well for both classification and regression QSAR models and we recommend starting with these algorithms on emerging new datasets in cheminformatics.

Data	Literature Metrics	Our Metrics
Blood Brain Barrier	Accuracy: 71~81% MCC: 0.52~0.64	Accuracy: 70~80% MCC: 0.30~0.56
USP1 Inhibition	Accuracy: 79~81% AUC: 0.72~0.87	Accuracy: 72~95% AUC: 0.73~0.84
Sigma-2 Ligands	R^2: 0.43~0.56 MSE: 0.25~0.37	R^2: 0.17~0.43 MSE: 0.53~0.78

Table 7: Comparison of results with literature[40,44,48]

There are a few reasons for the mediocre performance of some predictive models. Structure-activity relationships are highly complex non-linear models which may not be explained

by the calculated molecular descriptors. Computing additional molecular descriptors and molecular fingerprints could potentially improve the training success of the models. During feature selection, independent variables that are useless by themselves may be dropped but could be useful together. Recursive feature selection has also been shown to be successful in selecting the optimum features for small molecules. Additionally, implementing other hyperparameter optimization techniques could help in optimizing model performance. By fixing the issues we faced in descriptor calculation, feature selection and training more machine learning algorithms, the model performances and predictions can be improved.

3.5 COMPARISON WITH OTHER RELATED TOOLS

For further comparison, we studied related publications to collect related tools that build QSAR models. A comparison based on the functionality of these tools is summarized in Table 8. In this table, we compared these tools based on its functionality like type, data preprocessing, feature selection and model selection. The results suggest that PyMolSAR is strongly recommended for multiple advantages of it as shown in the table.

Tool	Type	Structure Processing	Data Processing	Molecular Representation	Feature Selection	Classification Models	Regression Models
PyMolSAR	Python package		✓	✓	✓	✓	✓
ChemSAR[51]	Online	✓	✓	✓	✓	✓	
Chembench[52]	Online	✓	✓	✓	✓	✓	✓
OCHEM[53]	Database	✓	✓	✓		✓	✓

Camb[54]	R package	✓	✓	✓	✓	✓	✓
Rregrs[55]	R package				✓	✓	
QSARINS[56]	Software		✓	✓		✓	

Table 8: The current tools that can be used for QSAR modelling

Chapter 4: CONCLUSION

In this study, we developed the PyMolSAR Python package to build and evaluate quantitative structure-activity relationship (QSAR) classification and regression models. It is open-source and freely accessible on GitHub at https://github.com/BeckResearchLab/PyMolSAR. The main advantage of PyMolSAR is that it provides a comprehensive modeling pipelining by integrating all the model generation steps in a unified workflow enabling cheminformatics researchers to build predictive models easily. In addition, we conducted case studies to illustrate the use of this package in practice, and several models were obtained to predict blood brain barrier permeability, USP1 inhibitory activity and Sigma-2 receptor ligand binding activity.

The current trend for structure-activity relationship models is to making data and models publicly accessible and open-source. PyMolSAR, to some extent, has made a step in this direction. It is expected that PyMolSAR can be applied to many studies where there exists a demand for building models for predicting molecular properties and activities. In the future, we will continue to implement more algorithms and add options for more molecular descriptors, fingerprints and flexible parameter control.

BIBLIOGRAPHY

1. Prakash, N., and D. A. Gareja. "Cheminformatics." *J Proteomics Bioinform* 3 (2010): 249-252.

2. Begam, B. Firdaus, and J. Satheesh Kumar. "A study on cheminformatics and its applications on modern drug discovery." *Procedia engineering* 38 (2012): 1264-1275.

3. Mitchell, John BO. "Machine learning methods in chemoinformatics." *Wiley Interdisciplinary Reviews: Computational Molecular Science* 4.5 (2014): 468-481.

4. Liao, Chenzhong, et al. "Software and resources for computational medicinal chemistry." *Future medicinal chemistry* 3.8 (2011): 1057-1085.

5. Vilar, Santiago, and Stefano Costanzi. "Predicting the biological activities through QSAR analysis and docking-based scoring." *Membrane Protein Structure and Dynamics*. Humana Press, Totowa, NJ, 2012. 271-284.

6. Nantasenamat, Chanin, et al. "A practical overview of quantitative structure-activity relationship." (2009).

7. Bajorath, Jürgen. "Quantitative Structure Activity Relationship." *Encyclopedia of Cancer*. Springer, Berlin, Heidelberg, 2011. 3128-3131.

8. Dixon, Steven L., et al. "AutoQSAR: an automated machine learning tool for best-practice quantitative structure–activity relationship modeling." *Future medicinal chemistry* 8.15 (2016): 1825-1839.

9. Kausar, Samina, and Andre O. Falcao. "An automated framework for QSAR model building." *Journal of cheminformatics* 10.1 (2018): 1.

10. D. Pugazhenthi and S.P. Rajagopalan, 2007. Machine Learning Technique Approaches in Drug Discovery, Design and Development. *Information Technology Journal, 6: 718-724.*`

11. Nigsch, Florian, et al. "Melting point prediction employing k-nearest neighbor algorithms and genetic parameter optimization." *Journal of chemical information and modeling*46.6 (2006): 2412-2422.

12. Tan, N. X., et al. "Prediction of chemical carcinogenicity by machine learning approaches." *SAR and QSAR in Environmental Research* 20.1-2 (2009): 27-75.

13. Pella, Andrea, et al. "Use of machine learning methods for prediction of acute toxicity in organs at risk following prostate radiotherapy." *Medical physics* 38.6Part1 (2011): 2859-2867.

14. Cruz, Joseph A., and David S. Wishart. "Applications of machine learning in cancer prediction and prognosis." *Cancer informatics* 2 (2006): 117693510600200030.

15. Lima, Angélica Nakagawa, et al. "Use of machine learning approaches for novel drug discovery." *Expert opinion on drug discovery* 11.3 (2016): 225-239.

16. Gamberger, Dragan, et al. "Application of artificial intelligence in biodegradation modelling." *Biodegradability Prediction*. Springer, Dordrecht, 1996. 41-50.

17. Landrum, G. "RDKit: open-source cheminformatics http://www. rdkit. org." (2016).

18. Cao, Dong-Sheng, et al. "ChemoPy: freely available python package for computational biology and chemoinformatics." *Bioinformatics* 29.8 (2013): 1092-1094.

19. O'Boyle, Noel M., et al. "Open Babel: An open chemical toolbox." *Journal of cheminformatics* 3.1 (2011): 33.

20. Pedregosa, Fabian, et al. "Scikit-learn: Machine learning in Python." *Journal of machine learning research* 12.Oct (2011): 2825-2830.

21. Chollet, François. "Keras." (2015): 128.

22. SMILES - A Simplified Chemical Language

23. Avadhoot R, A.C. Beck D. PyMolSAR. https://github.com/BeckResearchLab/PyMolSAR

24. McKinney, Wes. "pandas: a foundational Python library for data analysis and statistics." *Python for High Performance and Scientific Computing* (2011): 1-9.

25. Hunter, John D. "Matplotlib: A 2D graphics environment." *Computing in science & engineering* 9.3 (2007): 90-95.

26. Todeschini R, Consonni V. Molecular Descriptors for Chemoinformatics.; 2010. doi:10.1002/9783527628766.

27. Dong J, Cao D, Miao H, Liu S, Deng B, Yun Y, Wang N, Lu A, Zeng W, Chen AF (2015) ChemDes: an integrated web-based platform for molecular descriptor and fingerprint computation. J Cheminform 7(1):60

28. Guyon I, Elisseeff A (2003) An introduction to variable and feature selection. J Mach Learn Res 3(3):1157–1182

29. Ng, Andrew Y. "Feature selection, L 1 vs. L 2 regularization, and rotational invariance." *Proceedings of the twenty-first international conference on Machine learning*. ACM, 2004.

30. van Gerven, Marcel, and Sander Bohte, eds. *Artificial neural networks as models of neural information processing*. Frontiers Media SA, 2018.

31. Breiman L (2001) Random forests. Mach Learn 45(1):5–32

32. Pontil M, Verri A (1998) Properties of support vector machines. Neural Comput 10(4):955–974

33. Altman, Naomi S. "An introduction to kernel and nearest-neighbor nonparametric regression." *The American Statistician* 46.3 (1992): 175-185.

34. Hammarlund-Udenaes, Margareta, et al. "On the rate and extent of drug delivery to the brain." *Pharmaceutical research*25.8 (2008): 1737-1750.

35. Upadhyay, Ravi Kant. "Drug delivery systems, CNS protection, and the blood brain barrier." *BioMed research international*2014 (2014).

36. Banks, William A. "Characteristics of compounds that cross the blood-brain barrier." *BMC neurology*. Vol. 9. No. 1. BioMed Central, 2009.

37. Norinder, Ulf, and Markus Haeberlein. "Computational approaches to the prediction of the blood–brain distribution." *Advanced drug delivery reviews* 54.3 (2002): 291-313.

38. Ajay,*, Guy W. Bemis, and Mark A. Murcko. "Designing libraries with CNS activity." *Journal of medicinal chemistry*42.24 (1999): 4942-4951.

39. Doniger, Scott, Thomas Hofmann, and Joanne Yeh. "Predicting CNS permeability of drug molecules: comparison of neural network and support vector machine algorithms." *Journal of computational biology* 9.6 (2002): 849-864.

40. Li, Hu, et al. "Effect of selection of molecular descriptors on the prediction of blood− brain barrier penetrating and nonpenetrating agents by statistical learning methods." *Journal of Chemical Information and Modeling* 45.5 (2005): 1376-1384.

41. Crivori, Patrizia, et al. "Predicting blood− brain barrier permeation from three-dimensional molecular structure." *Journal of medicinal chemistry* 43.11 (2000): 2204-2216.

42. Priolo, Carmen, et al. "The isopeptidase USP2a protects human prostate cancer from apoptosis." *Cancer research*66.17 (2006): 8625-8632.

43. Popov, Nikita, et al. "The ubiquitin-specific protease USP28 is required for MYC stability." *Nature cell biology* 9.7 (2007): 765.

44. Wahi, Divya, et al. "Cheminformatics models based on machine learning approaches for design of USP1/UAF1 abrogators as anticancer agents." *Systems and synthetic biology* 9.1-2 (2015): 33-43.

45. Philip, Pearl. *Analyzing small molecule inhibition of enzymes: A preliminary machine learning approach towards drug lead generation*. Diss. 2017.

46. NIH Chemical Genomics Center. Inhibitors of USP1/UAF1: Primary Screen. PubChem. https://pubchem.ncbi.nlm.nih.gov/bioassay/743255. Published 2014

47. Mach, Robert H., Chenbo Zeng, and William G. Hawkins. "The σ2 receptor: a novel protein for the imaging and treatment of cancer." *Journal of medicinal chemistry* 56.18 (2013): 7137-7160.

48. Rescifina, Antonio, et al. "Sigma-2 receptor ligands QSAR model dataset." *Data in Brief* 13 (2017): 514-535.

49. Benfenati, Emilio, et al. "CORAL software: QSAR for anticancer agents." *Chemical biology & drug design* 77.6 (2011): 471-476.

50. Nastasi, Giovanni, et al. "S2RSLDB: a comprehensive manually curated, internet-accessible database of the sigma-2 receptor selective ligands." *Journal of Cheminformatics* 9.1 (2017): 3.

51. Dong, Jie, et al. "ChemSAR: an online pipelining platform for molecular SAR modeling." *Journal of cheminformatics* 9.1 (2017): 27.

52. Capuzzi, Stephen J., et al. "Chembench: a publicly accessible, integrated cheminformatics portal." *Journal of chemical information and modeling* 57.2 (2017): 105-108.

53. Sushko, Iurii, et al. "Online chemical modeling environment (OCHEM): web platform for data storage, model development and publishing of chemical information." *Journal of Computer-Aided Molecular Design* 25.6 (2011): 533-554.

54. Murrell, Daniel S., et al. "Chemically Aware Model Builder (camb): an R package for property and bioactivity modelling of small molecules." *Journal of cheminformatics* 7.1 (2015): 45.

55. Tsiliki, Georgia, et al. "RRegrs: an R package for computer-aided model selection with multiple regression models." *Journal of cheminformatics* 7.1 (2015): 46.

56. Gramatica, Paola, et al. "QSARINS: a new software for the development, analysis, and validation of QSAR MLR models." *Journal of Computational Chemistry* 34.24 (2013): 2121-2132.