R software for QSAR analysis in phytopharmacological studies

Sanjoy Singh Ningthoujam[1], Rajat Nath[2], Anupam Das Talukdar[2*], Lutfun Nahar[3*], Satyajit D Sarker[4],

[1] Government Hindi Teachers' Training College, Imphal, Manipur, India

[2] Department of Life Science and Bioinformatics, Assam University, Silchar, Assam, India

[3]Laboratory of Growth Regulators, Institute of Experimental Botany, The Czech Academy of Sciences and Palacký University, Šlechtitelů 27, 78371 Olomouc, Czech Republic

[4] Centre for Natural Products Discovery (CNPD), School of Pharmacy and Biomolecular Sciences, Liverpool John Moores University, Byrom Street, Liverpool L3 3AF, United Kingdom

*Corresponding Authors

Abstract

Introduction: Quantitative Structure-Activity Relationship (QSAR) has emerged as an important tool for drug design and natural product research in recent decades. It helps in identifying and optimizing lead compounds by studying the biological response of phytochemicals. With the availability of bioinformatics and cheminformatics tools, a vast number of descriptors have been generated, making it challenging to select potential independent variables that can accurately relate to the dependent response variable.

Objective: The objective of this study is to demonstrate various descriptor selection procedures, such as Boruta approach, all subset regression, ANOVA approach, AIC method, Stepwise Regression, and Genetic Algorithm, that can be used in QSAR studies. Additionally, we will perform regression diagnostics using R software to test parameters such as normality, linearity, residuals histogram, PP plot, multicollinearity, and homoscedasticity. We aim to provide an easy, extendable, and customizable approach to QSAR studies.

Results: The workflow designed in this study highlights the different descriptor selection procedures and regression diagnostics that can be used in QSAR studies. In-house curated and downloaded data were used to demonstrate the effectiveness of these approaches. The results showed that the Boruta approach and Genetic Algorithm performed better than other methods in selecting potential independent variables. The regression diagnostics parameters tested using R software, such as normality, linearity, residuals histogram, PP plot, multicollinearity, and homoscedasticity, helped in identifying and diagnosing model errors, ensuring the reliability of the QSAR model.

Conclusion: QSAR is a useful tool in drug design and natural product research, and selecting the appropriate descriptors and performing regression diagnostics is critical for developing a reliable QSAR model. The approaches highlighted in this study, such as the Boruta approach, Genetic Algorithm, and regression diagnostics using R software, can provide an easy, extendable, and customizable approach to QSAR studies. This study can serve as a guide for researchers in selecting appropriate descriptors and diagnosing model errors in QSAR studies.

Keywords: MLR; QSAR; R software; descriptor; feature selection; regression diagnostics; regression assumption

1. Introduction

The Quantitative Structure-Activity Relationship (QSAR) is an established computational approach for chemical data analysis that has become an important tool for agrochemistry, pharmaceutical chemistry, toxicology and different facets of chemistry and bioinformatics [1, 2]. The QSAR models describe mathematical equations correlating the chemical activity (response) of compounds with their structural and physiochemical information in the form of numerical quantities, i.e., descriptors. These models are developed using appropriate statistical or machine learning approaches. Many powerful statistical software, both commercial and open source, are available that can efficiently perform QSAR. Though the process is popular and common, there are many issues that are frequently overlooked in its analyses. QSAR analysis is faced with various challenges including lack of basic concepts and flawed interpretation of the generated results. If a data matrix is fed into any statistical software, it is inevitable that some results would be generated. At present one dimensional to multi-dimensional QSAR methods are available that can be used in lead optimization, classification and prediction of pharmacological or biological activity and pharmacokinetic properties [3].

Along with prediction of biological activities, QSAR models help in identifying the parameters responsible for biological response essential for lead compound identification and optimization [4]. As such, QSAR has a significant contribution in rational drug design in the present context. There have been several evolutions of methods and best practices in QSAR, such as prediction of biological activities, ADMET properties, application of QSAR modelling in chemical, pharmaceutical and cosmetic industries [1]. New and interesting areas and applications in process chemistry, synthetic route prediction and optimization have emerged as new domains of QSAR applications. In spite of these advances, QSAR modelling has relevance in drug discovery and identifying pharmacological activities of phytochemicals [4-6].

There are different statistical packages and specific software for QSAR but with different degrees of reliability [7]. These packages may belong to either commercial or open-source products. Though commercial statistical products have many visible features, open source software have the advantages of customizability, freedom, quality, flexibility, interoperability and auditability. Some of the software are dedicated statistical software such as SPSS, Stata, SAR, Weka, Tanagra, RapidMiner and MatLab, R. There are some integrated software such as EasyQSAR, SyByl-X, Codessa and Discovery Studio, which include QSAR functionalities with statistical functions [8]. One popular open source statistical package is R software that can run in a variety of platforms ranging from Windows, MacOS to Linux [9]. Moreover, open source software is a good candidate for implementing statistical analyses in the QSAR studies. The aim of this paper is to review the R software for QSAR analysis that has potential applications in phytopharmacological

studies. Applicability of the R software is focused on three phases of the QSAR steps viz (a) descriptor selection, (b) test for regression assumption and (c) addressing assumption violations.

2. Generalized Steps in QSAR

The general protocol for developing QSAR models particularly for phytochemical analysis consists of several modular steps. The first step is usually 'molecular encoding', where the chemical features and properties are derived from chemical structures or lookup of experimental results. In the next step, feature selection is performed under various techniques to identify the most relevant properties and reduce the dimensionality of the features [10]. In the last, a QSAR model is applied to discover an empirical function that can produce a optimal relationship between the input feature vectors and the biological properties.

Generalized steps in QSAR modelling include -

i. Generate descriptors - it involves converting the molecular structure of the materials into a set of numbers that capture their molecular and physiochemical properties in a relevant way.
ii. Select a subset of descriptors in a context dependent way. Here a small subset of descriptors having the most influence on the biological properties of the compounds is selected.
iii. Deduce the relationship between the descriptors and the biological responses.
iv. Validate the model in terms of its robustness, prediction ability and domain of applicability.

After that the model can be used to estimate the biological properties of new molecules where biological activities are not known. In essence, a QSAR comprises three important parts – (a) the activity data to be modelled and hence predicted (response variable), (b) data with which to model (predictors or descriptors) and (c) a method to formulate the model [11].

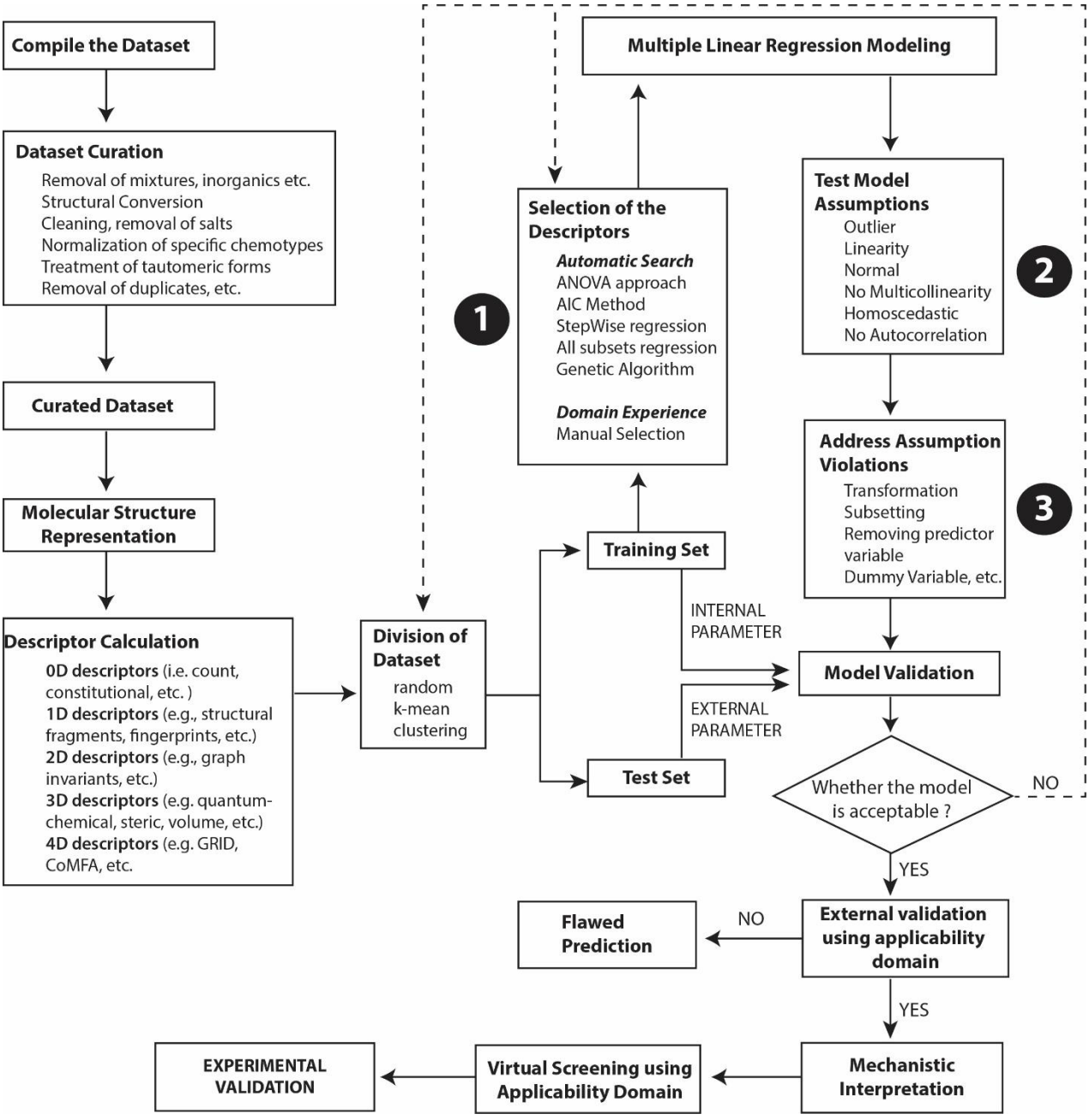

Figure 1: Generalized Workflow of QSAR showing focus of the paper (1) Descriptor Selection and (2) Test for Regression Assumption and (3) Addressing Assumption Violations

3. Methods in QSAR

Methods that can be used in QSAR include various regression and pattern recognition techniques [12]. There have been various methodologies and techniques widely developed ranging from 2-D QSAR to 3-D QSAR. Major differences in these techniques are the selection of structural parameters that are used to characterize molecular identities as well as the mathematical procedure for describing the relationship between

descriptors and biological activity [13]. In the QSAR modelling, various statistical methods are used to derive mathematical correlation involving small to large number of variables. These methods may be (a) regression-based methods, (b) classification-based methods or (c) machine learning methods. QSAR methods may be linear models and non-linear models. Apart from that, QSAR models are also classified as receptor-dependent QSAR and receptor-independent QSAR based on whether molecular receptor binding ways are considered or not.

In the classification-based QSAR models, validation parameters like accuracy, sensitivity, specificity, precision and F-measures are commonly used [14]. Regression based methods are commonly used in many QSAR studies. Multiple Linear Regression (MLR) models are familiar in many QSAR publications because of its simplicity, transparency, reproducibility and easy interpretability [15]. There are various machine learning tools such as artificial neural network, support vector machine, random forests are available. There are new methods such as Gene Expression Programming (GEP), Project Pursuit Regression (PPR), Local Lazy Regression (LLR) also used in QSAR studies [16].

4. Modern Trends in QSAR

There has been unprecedented growth in the area of QSAR thereby changing the dimensions of applications in drug discovery such as lead optimization and drug-receptor interactions and protein-protein interactions. **[17]**. There has been an increase in diversity of datasets used for QSAR with the advancement of robotics and miniaturization. Large volumes of data are also available in public domain databases such as ChEMBL, PubChem, ZINC etc. [1].

4.1. AI and machine learning in QSAR

There is an increasing application of artificial intelligence (AI) and machine learning in QSAR modelling. Various standard methods of ML have been employed in the QSAR that can be categorized into following types.

I. Supervised Learning (e.g. regression analysis, k-nearest neighbor, Bayesian probabilistic learning, SVMs, random forest and neural networks)
II. Unsupervised Learning (e.g. clustering algorithms, dimensionality reduction techniques such as principal components analysis (PCA), independent components analysis (ICA) and several supervised methods that can also support unsupervised learning, such as SVMs, probabilistic graphical models and neural networks)

Use of DNNs is not a new phenomenon, but it began in the 1990s. In the past few decades, algorithmic improvements, advances in hardware and use of GPUshave contributed to the improvement in neural nets. Application of DNNs began after the Merck Molecular Activity Challenge in 2012 [1]. There are other ML

techniques, apart from DNNs, such as kNN (k-nearest neighbors_, partial least squares (PLS), support vector machine (SV), relevance vector machines (RVM), random forest (RF), Gaussian processes (GP) and boosting. Random Forest is another popular method for QSAR modeling as it provides good predictions with few adjustable parameters. At present deep neural nets (DNN) emerge as an important ML method.

4.2. Improvement in Validation Method

There is changing paradigm in the validation methods of QSAR. Common practice is to split the data into external test set and training set. Model developed using the training set is used to predict test set endpoints and for determination of accuracy of prediction. One of the major advancements is the use of time-split test set. In this approach, compounds tested in the later phases are assigned to the test set. Time-split is considered to provide a good estimate of the R2 for true prospective prediction relative to random test set selection and leave class out validation.

4.3. Multi Task Modeling

Generally, one predicted activity is determined at a time in the classical QSAR modelling. During drug development, multiple activities including on and off target are required for study. The technique for prioritizing compounds based on more than one predicted activity simultaneously is multi-parameter optimization or multi task modeling. This approach can be accomplished by an ensemble of single task models or by a single model that can model more than one activity simultaneously. This approach may use either non-neural net or neural net-based techniques including deep learning techniques. The approach of multitask modelling is expected to be significant when data are sparse. This modeling approach utilizes methods such as perturbation theory and machine learning (PTML), inductive learning and multi objective optimization.

4.4. Improvement in Applicability Domain

Applicability domain provides a means for estimating the reliability of QSAR model. It defines the space of molecular features on which the model has been trained and the conditions where the model should be applied. Applicability Domain of the QSAR models can be estimated by (a) leverage approach, (b) the DModX approach, (c) Euclidean distance approach, etc. [14]. As the QSAR models are data-driven models based on patterns or rules of the training samples, these models can be valid within limited applicability domains (AD). Various AD characterization methods and AD metrics are proposed at present with utilization of machine learning algorithms [18].

4.5. Modelability

A significant new trend in QSAR is the concept of modelability that proposes that that predictivity of QSAR models is limited by activity cliffs. Activity cliffs are observed when similar compounds have different

activities. It makes the target property of compounds hard to predict. This limitation can not be easily addressed by changing either the QSAR method or the descriptors used in the modeling. However, using stereochemically aware descriptors can reduce activity cliffs where different stereoisomers exhibit different activities.

4.6. Interpretability

Classical QSAR methods usually deal with molecules that were close analogs. With the development of more sophisticated modeling methods with diverse datasets and esoteric descriptors, the concept of interpretability has become significant. While modeling, only the relevant subset of descriptors is selected. It improves the ability of models to generalize well and make interpretation easier. Models are usually interpreted in two ways - the first is to determine descriptors with capability to derive properties, and the second is to project the most important features from model onto exemplar molecules to highlight structural features associated with favorable activity.

5. Polypharmacological applications of QSAR

Potential applications of QSAR in searching for drugs with polypharmacological implications are also recognized. The classical QSAR models are developed with training sets of compounds having a common bioactivity. Most of these compounds belong to same chemical series. Introduction of diverse assays and different technologies, it is accepted that the drug candidates may interact with the many biological targets. Such polypharmacological properties may generate additive or synergistic effects or create adverse or toxic effects [19]. However, experimental evaluation of millions of drug candidates for thousands of targets in traditional wet lab settings is unrealistic. The situation is aggravated by the fact that there is variability of results for the same ligand-target interaction in different assays. Considering these constraints, *in silico* prediction of multiple bioactivities through QSAR models has emerged as an notable alternative [1].

The concept of biological activity spectrum originates in the basis of multi-target profiling of compounds. The biological activity spectrum is the set of different biological activities resulting from the compound interaction with different biological systems. There are several models for multi target profiling, with pioneering work of PASS (prediction of activity spectra for substances). This tool enables the prediction of various biological activities at molecular, cellular, tissue and organism levels. Perturbation theory machine learning (PTML) methods is another method that is capable of simultaneous prediction of many target properties under different conditions [20].

A major limitation observed in the multi target profiling is about evaluation of the predicted compound sets. There are only a few comparative results about the predictability of different approaches. Out of them, feed-

forward neural networks (FNN) generated the best results while similarity ensemble approach (SEA) provide the lowest predictability.

Another extension of multitarget QSAR is the searching for ligand-target interactions in combined chemical-biological space. This strategy called chemogenomics or proteochemometrics searches for all molecules that are capable of interacting with any biological target [21]. This approach differs from classical QSAR that predicts ligands for a given receptor by predicting protein-ligand interactions at large scales in the protein and chemical spaces. This approach addresses the issues of predicting off-target proteins during drug discovery that causes undesirable side effects during the process [22]. Recently, Monte Carlo based QSAR approach was used to study protease inhibitors for COVID19.

6. Descriptor Free QSAR

Mainstream QSARs are based on linking molecular descriptors (X) to the response variable (Y). However, determining the appropriate descriptors is a complex process because many of them are difficult to explain how they are related to the response activity. Determination of the molecular descriptors may require appropriate software framework and may suffer from human bias. There were attempts to develop QSAR models by completely eliminating molecular descriptors, particularly for large and diverse datasets by using deep learning techniques such as Long-Short-Term memory networks (LSTM) from SMILES code [23].

7. QSAR in Phytochemical Analysis and Drug Discovery

There are various applications of QSAR in predicting the hitherto unknown properties of phytochemical compounds. Rahman et.al.[24] applied QSAR models to predict activity of antiviral phytochemicals against NS3 protease of dengue virus. The pIC(mM) values for three phytochemicals Cyanidin 3-Glucoside, Dithymoquinone and Glabridin were predicted by MLR model with 89.91 goodness of fit to have good potency.

Search for antiviral phytochemicals is newly emerging as a research focus in view of the Covid19 pandemics. Protease of SARS-CoV-2 is one of the important targets for designing and developing antiviral drugs. Islam et.al. developed a MLR model to predict favorable binding energy of antiviral phytochemicals to find out the potent inhibitors against the main protease of SARS-CoV-2 [25].

Spike proteins of the Covid19 contain Receptor Binding Domain (RBD) responsible for virus infection in human cells. These proteins bind to the receptor proteins angiotensin converting enzyme 2 (ACE2) in the Protease Domain of the host cell and multiplies itself. Basu et.al. studied five phytochemicals (hesperidin, anthraquinone, rhein, chrysin and emodin) from Indian and Chinese medicinal plants and studied their antiviral activity through QSAR and molecular docking studies and observed that hesperidin, emodin and chrysin have the potential for using in Covid19 treatment [26].

8. R software in QSAR

The R software is the comprehensive programming language and environment that incorporates all standard statistical tests, models and analysis for practicing statisticians and researchers [27]. As it is an open-source, anybody can access the source code freely, modify it and improve it according to the suitability. Another advantage of R is its extensibility and developers can write their own package for various functions for data manipulation, statistical modeling and graphics. The R software is available as Free Software and can be downloaded from the www.r-project.org [9]. Operations of R can be implemented through various IDE (integrated developing environment) such as R Studio. In addition to R base installations, there are various R packages available in the CRAN website https://cran.r-project.org/web/packages and other resources that can be used for QSAR studies.

Table 1: Some of the R packages that can be used in QSAR analysis

R packages	Purpose	Reference
Boruta	Boruta	[28]
Car: Companion to Applied Regression	Variance Inflation Factor test	[29]
Leaps: Regression Subset Selection	All subsets regression	[30]
Lmtest: Testing Linear Regression Models	Durbin-Watson test	[31]
MASS	Stepwise Regression	[32]
Subselect	Genetic Algorithm	[33]
Ezqsar	Dedicated QSAR package	[34]
Rmol	Transforming SD/Molfile structure information into R objects	[35]
Rregrs	Model selection with multiple regression models	[36]
Camb	Property and bioactivity modeler	[37]

Along with the R base installation, these packages can be used for Feature Selection, Testing Model Assumptions and subsequent solutions as focused on this paper (Fig. 2). Other processes or steps have been left as out of the purview of this paper. Steps mentioned here are not fixed and can be tweaked according to the convenience of the process. In the process, the dataset is split into three mutually exclusive sets - training set, trial set and external evaluation set. External validation sets are different from test set or training set and are used to estimate prediction error to compare models. When sufficient data is available, it is preferable to split into training set and test set. When data available is insufficient to separating validation and test sets, all the data are used in the training set [38].

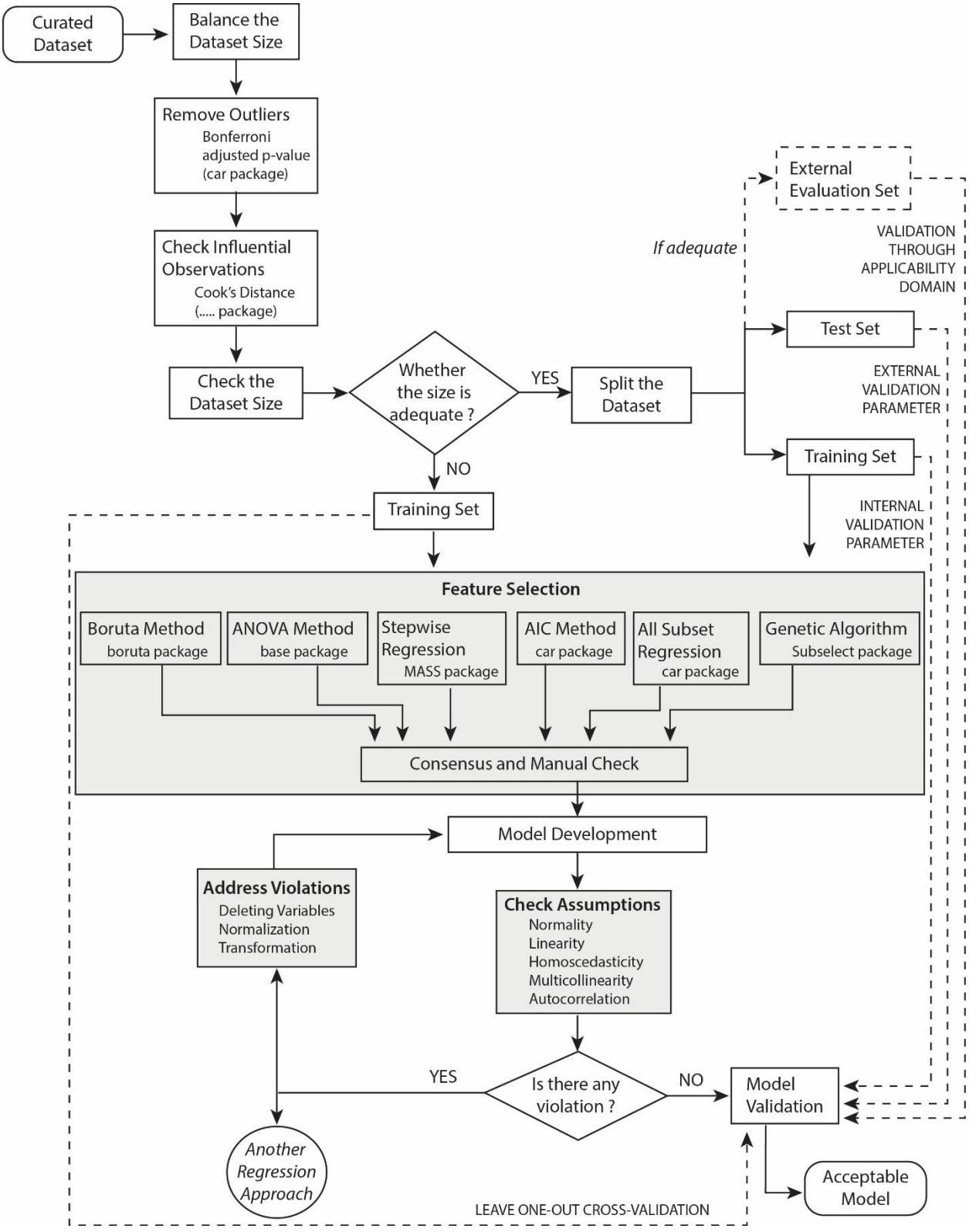

Figure 2: In the QSAR process, focus this review paper is given on feature selection, violations of regression assumptions and their subsequent solutions

9. Feature Selection

In the QSAR studies, it is important to decide whether all the variables should be included under study or drop ones that are considered to be insignificant [39]. Selection of a final regression model is usually a compromise between predictive accuracy (a model that fits the data as well as possible) and parsimony (a simple and replicable model). The purpose of this feature selection is to choose the model with approximate equal predictive accuracy with simpler one.

Feature selection is the technique of extracting a subset of relevant descriptors for model development. In QSAR studies, feature election is significant step for selecting suitable and relevant descriptors for a particular response. Descriptor selection is important because models with fewer variables are easier to interpret, provide improved performance for new samples and decrease the risk of overfitting or overtraining [40]. It is significant for removing noises from the analyses. Many statistical tools are available for feature selection. Descriptor data are pruned or filtered to remove intercorrelated and redundant descriptor data. With the advancement of QSAR, many new techniques and algorithms for descriptor selection have emerged that may belong to either of the three major categories viz. filter, wrapper and embedded/hybrid methods (Fig. 3) [40]. Filter methods do not apply any machine learning process and performed in unsupervised manner. Filter method apply a statistical measure to assign a scoring to each feature and select variables regardless of the model. For instance, they use general features like the correlation with the variable to predict. In the wrapper techniques, a linear or nonlinear classifier (or regressor) is used to select descriptors while hybrid or embedded is combination of the above two techniques.

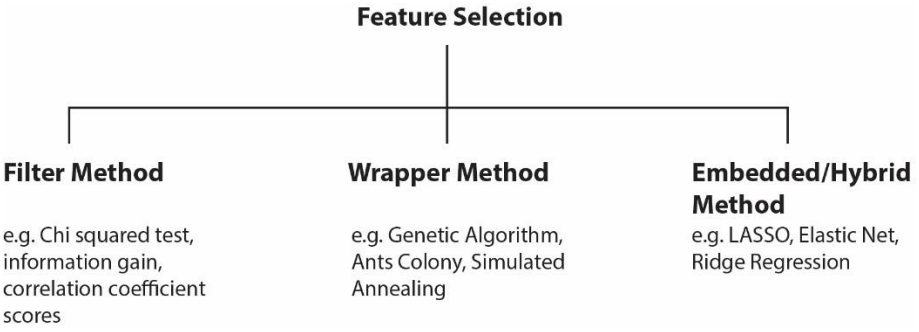

Figure 3: Types of Feature Selection

Traditional methods of feature selection include Human experts who relied on experience and scientific intuition, Correlation analysis of the data set, Application of statistical methods and Intelligent algorithms. Popular methods are Forward Stepping and Backward Elimination, Neural Network Pruning, Simulated Annealing, Genetic Algorithm and Exhaustive Enumerations. In the present approach, Boruta Method,

ANOVA approach, AIC method, Stepwise Regression, all subsets regression and Genetic Algorithm are considered.

9.1. Boruta Method

Boruta is a feature selection algorithm which works as a wrapper algorithm around Random Forest in default mode [28]. An example is shown in the paper. Dataset for this example is based on the Selwood dataset popular in genetic algorithms and QSAR modeling [41]. The syntax of boruta is similar to that of regression lm() function (Listing 1).

```
> library(Boruta)
# to get Boruta object 'fit'
> fit <- Boruta(y~.,data=mydata, doTrace=2)
# to Treat the Tentative Attributes and get finalized object 'finalBoruta'
> finalBoruta <- TentativeRoughFix(fit)
# to get the confirmed attributes
> getSelectedAttributes(finalBoruta, withTentative = F)
```

Listing 1: Syntax of Boruta

While running the function, Boruta gives a clear call on the significance of variables in a dataset. Some attributes are rejected, some are confirmed and rest are designed as tentative. Tentative attributes are close to other attributes that Boruta is not able to make a decision with the desired confidence in default number of random forest runs. So, another process for tentative attributes are required. Then the list of the confirmed attributes can be generated by the R syntax.

In the first run, Boruta produced confirmed attributes with attributes rejected and tentative attributes which can be shown as graphical plot (Fig. 4). Out of these boxplots, blue ones correspond to minimal average and maximum Z score of a shadow attribute. Red, Yellow and Green boxplots represent Z scores of rejected, tentative and confirmed attributes respectively. After taking decision on tentative attributes, they are classified either as confirmed or rejected by comparing the median Z score of the attributes with the median Z score of the best shadow attributes. Then list of the confirmed descriptors can be obtained.

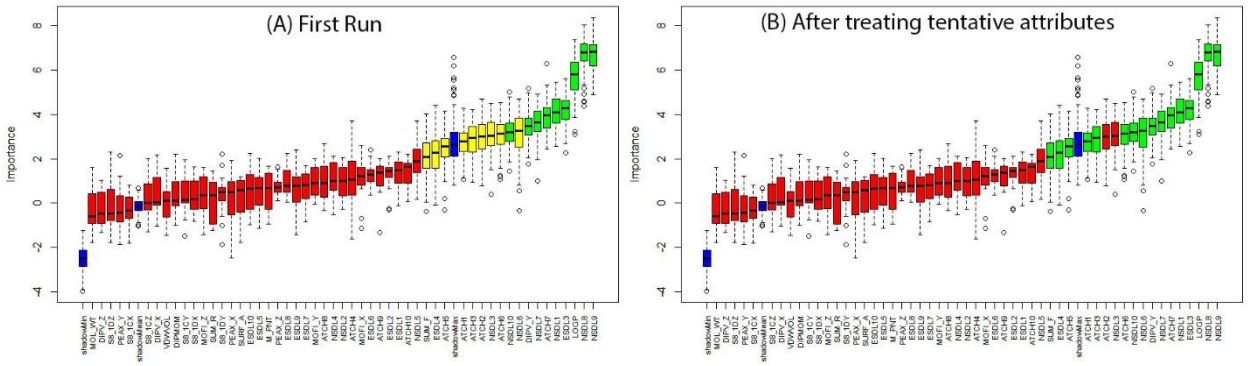

Figure 4: Boxplots showing importance of attributes; (A) after first run; (B) after treating tentative attributes

The package is based on all-relevant feature selection methods and considers all features which are relevant to the outcome variable. Though the Boruta package is based on Random Forest algorithm in default, the package does not handle missing (NA) values. Running this package is time consuming and demands computational resource for large dataset.

9.2. Anova approach

By using anova() function in the R base installation [9], two nested models can be compared. A nested model is one whose terms are completely included in the other model.

For example, in one model, where MIC is the response variable, regression coefficients for MASS and VOLUME are nonsignificant. A model can be tested whether one without these two variables could predict as well as one that includes them. An example syntax of *anova* approach in an inhouse QSAR studies is presented here (Listing 2).

```
> fit1 <- lm(MIC ~ Mass + Volume + SurfaceArea + HydrationEnergy,data=mydata)
> fit2 <- lm(MIC ~ SurfaceArea + HydrationEnergy, data=mydata)
# to compare two models
> anova(fit2, fit1)
```

Listing 2: Syntax of Anova approach

In the example with *anova* approach using inhouse dummy data, two models have been generated. Model 1 is nested within model 2. In this process, Mass and Volume are added to linear prediction. As the test is nonsignificant (p=.994), we can conclude that they don't add to the linear prediction. It means that these two descriptors can be dropped from our model.

```
Analysis of Variance Table
Model 1: MIC ~ SurfaceArea + HydrationEnergy
Model 2: MIC ~ Mass + Volume + SurfaceArea + HydrationEnergy
Res.Df RSS Df Sum of Sq F Pr(>F)
1 47 289.246
2 45 289.167 2 0.079 0.0061 0.994
```

Listing 11: Output from anova approach

9.3. AIC Method

The Akaike Information Criterion (AIC) provides one method for comparing models. Models with smaller AIC values indicating adequate fit with fewer parameters are preferred. R Code for AIC method is implemented in the MASS package [42] with MIC as the response and Mass, Volume, Surface Area and Hydration Energy as the predictors.

```
> library(MASS)
> fit1 <- lm(MIC ~ Mass + Volume + SurfaceArea + HydrationEnergy,data=mydata)
> fit2 <- lm(MIC ~ SurfaceArea + HydrationEnergy, data=mydata)
# to compare two models – fit1 and fit2
> AIC(fit1,fit2)
```

Listing 3: Syntax of AIC Method

The process provides different values. The AIC values with smaller models are usually selected. The AIC method does not require nested approach. Both anova and AIC methods can be implemented in comparing two models but not applicable to multiple models.

9.4. Stepwise Regression

In this approach, variables are added to or deleted from a model one at a time, until some stopping criterion is reached. It can be implemented in either of the any three methods:

(a) In forward stepwise regression we add descriptor to the model one at a time, stopping when the addition of descriptors would no longer improve the model.

(b) In backward stepwise regression, we start with a model that includes all descriptors, then delete them one at a time until removing descriptors would degrade the quality of the model.

(c) In stepwise regression (usually called stepwise), both forward and backward stepwise approaches are combined. Descriptors are entered one at a time. But at each step, the descriptors are re-evaluated and those that do not contribute to the model are deleted.

The stepAIC() function in the MASS package [42] could perform stepwise model selection (forward, backward, stepwise). An example syntax of an inhouse QSAR study is included here (Listing 4).

```
> library(MASS)
> fit1 <- lm(MIC ~ Mass + Volume + SurfaceArea + HydrationEnergy,data=mydata)
# to get stepwise model selection
> stepAIC(fit1, direction="backward")
```

Listing 4: Syntax of Stepwise Regression

Process will continue by generating different models with lesser number of descriptors and lesser number of AIC values. It will stop when removing any descriptor would lead to increase in AIC.

StepWise Regression methods are popular but now becoming quite out of fashion. Stepwise regression is controversial in the sense that it may find a good model, but there's no guarantee that it will find the best model. It is because not every possible model is evaluated. This process may encounter a problem if there are missing values, so it is better to dealt with these missing values before running the process.

9.5. All subsets regression

In this approach, every possible model is inspected. It is performed using the regsubsets() function from the leaps package [30]. We can choose R-squared, Adjusted R-squared, or Mallows Cp statistic as the criterion for reporting "best" models. The results can be plotted with either the plot() function or the subsets() function in the car package [29].

```
> library(leaps)
> leaps <-regsubsets(MIC ~ Mass + Volume + SurfaceArea +
HydrationEnergy,data=mydata, nbest=4)
# plotting with plot function
> plot(leaps, scale="adjr2")
# plotting with subset function in the car package
> library(car)
> subsets(leaps, statistic="cp",main="Cp Plot for All Subsets Regression")
> abline(1,1,lty=2,col="red")
```

Listing 5: Syntax of all subsets regression and plotting

Findings from the regsubsets () for all subset regression in ‘leaps’ package can be observed by plotting the result (Fig. 6). In this example, the first row (from the bottom), indicate a model with the intercept and Mass with an adjusted R-square of 0.33. A model with the Intercept and SurfaceEnergy has 0.1. On the 12th row, a model with the intercept, SurfaceEnergy, Hydration Energy and Volume has a value of 0.54. At the same time, above it, a model with intercept, Surface Area and Hydration Energy has value of 0.55. Model with

fewer descriptors with a larger adjusted R-square is obtained. It suggests that two predictor model (Surface Area and Hydration Energy) is the best.

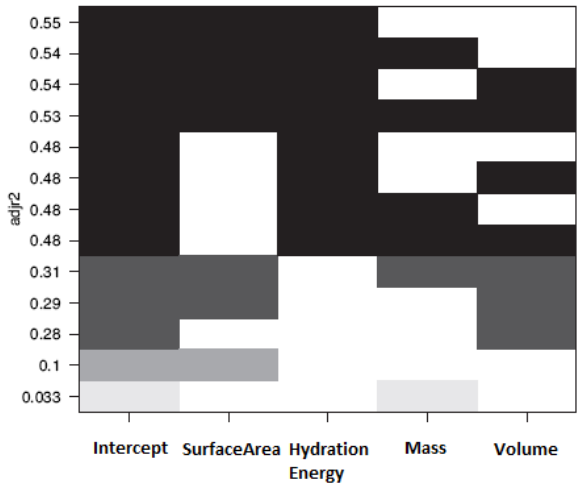

Figure 5: Plot from All Subset Regression

Automatic methods are helpful when the number of descriptors is large. The approach could not provide all possible models. When such conditions exist, it is more efficient to use a search algorithm such as StepWise Regression, Froward Selection etc. to find the best model.

9.6. Genetic Algorithm

Genetic Algorithms (GAs) are method which takes inspiration from Darwin's theory of the evolution. Here, each model competes with the others according to the concept of the 'survival of the fittest'. The genetic function in the subselect package [33] in R can perform genetic algorithm for variable selection [33]. An example syntax of Genetic Algorithm in R is provided here (Listing 6).

```
> library(subselect)
> genalgo <- genetic(cor(data), kmin=10, crit="gcd")
> genalgo$bestsets
```

Listing 6: Syntax for Genetic Algorithm

In the GA approach, a theoretical best model could not be obtained but generated a population of acceptable models. So, this characteristic gives another role of expert knowledge of the researcher. It provides an opportunity to make an evaluation of the relationships with the response from different points of view.

There are thousands of molecular descriptors available for QSAR analysis derived from different calculation methods. It is difficult to determine a clear physical chemical interpretation for many of these

descriptors. Sometimes, clear mechanistic interpretation could not be given as such descriptors are not clearly defined or identified [43]. There is no hard and fast criterion for determining the "best" model. Statistical methods can be used to determine the relative statistically significant descriptors. In the practical sense, it is difficult to determine whether the variables are important or not. For more practical purpose, it is better to depend more on domain knowledge of the researchers in selecting the descriptors that can be filtered through statistically significant variables. The final decision lies on the judgment of the investigator. So, it is better to consider all the highlighted descriptors and reach a consensus by applying the domain knowledge of the investigator.

10. Regression Diagnostic

The R software and add-on packages can be used for regression diagnostic. In the present study, evaluation of the statistical assumptions in the regression analysis is taken up in following manner – (a) Generation of the object through 'lm' function in the R base installation, (b) applying the 'plot()' function to the object returned by the 'lm()' function [9]. The model in the example is based on multiple linear regression. Success of the MLR depend on the degree of the tenability of the assumptions upon which MLR analyses are based. The plot() function to the object returned by the lm() function provides indication of the regression assumption to some extent (Fig. 6).

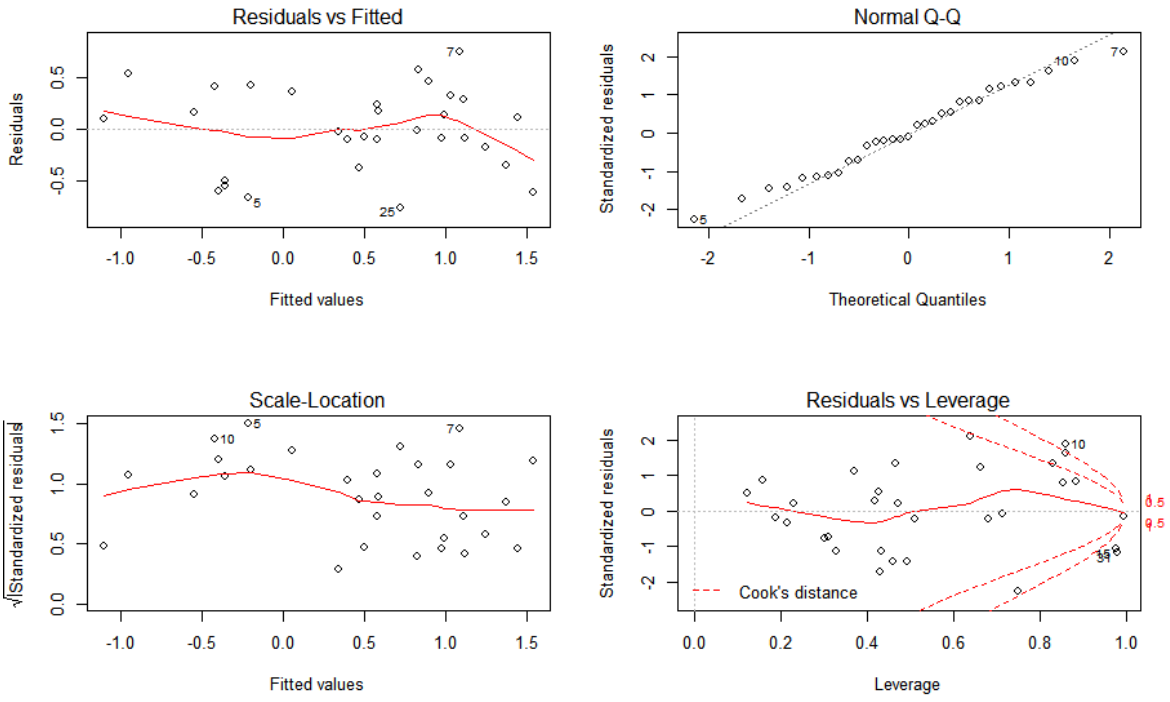

Figure 6: Plots generated by the model (a) Residuals vs Fitted, (b) Scale-Location, (c) Normal Q-Q and (d) Residuals vs Leverage

10.1. Linearity

In regression analysis, it is assumed that independent variables have linear relationship with the response variable. This assumption can be checked through a residual plot. The plot is formed by graphing the standardized residuals on the y-axis and the standardized predicted values on the x-axis. An optional horizontal line can be added to aid in interpreting the output. Syntax for Residual plot in the object 'fit' generated from lm() function is provided (Listing 8).

```
> #to obtain unstandardized predicted and residual values
> unstandardizedPredicted <- predict(fit)
> unstandardizedResiduals <- resid(fit)
> #to get standardized values
> standardizedPredicted <- (unstandardizedPredicted -
mean(unstandardizedPredicted)) / sd(unstandardizedPredicted)
> standardizedResiduals <- (unstandardizedResiduals -
mean(unstandardizedResiduals)) / sd(unstandardizedResiduals)
> #to create standardized residuals plot
> plot(standardizedPredicted, standardizedResiduals, main = "Standardized
Residuals Plot", xlab = "Standardized Predicted Values", ylab = "Standardized
Residuals")
> #to add horizontal line
> abline(0,0)
```

Listing 8: Syntax for plotting standardized Predicted vs standardized Residuals

Assumptions that the variable have linear relationship is analyzed by creating a residual plot in the R software (Fig. 7). Values close to the horizontal line are predicted well. Values high above the horizontal line are underpredicted while those on the lower sides are overpredicted. Linearity assumption is determined when the number of points scattered above and below the line is equal.

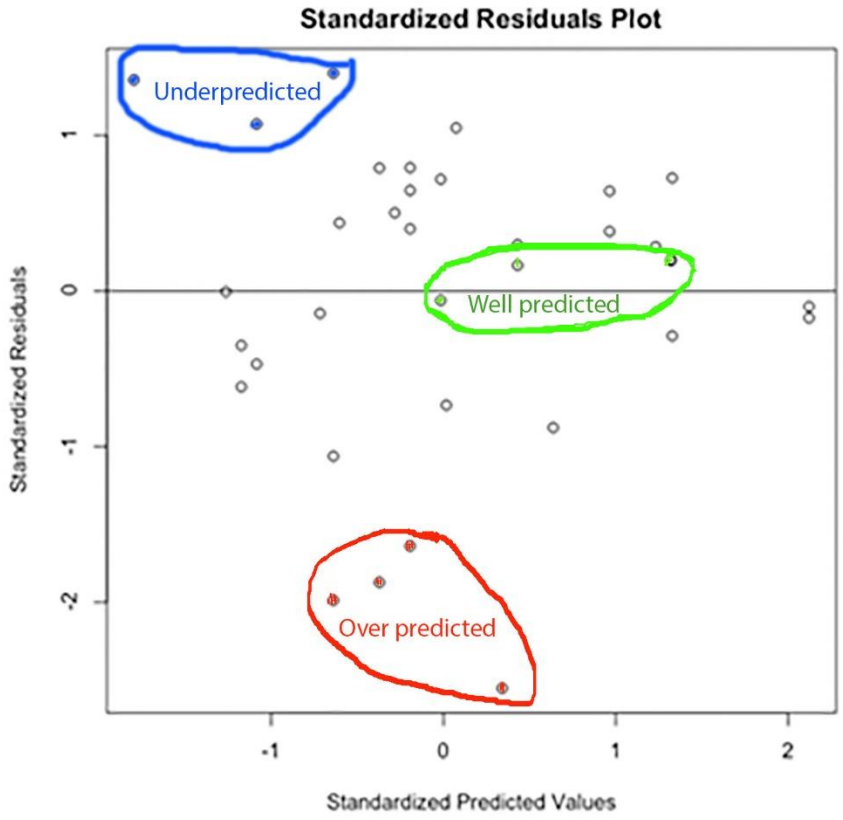

Figure 7: Checking linearity of the model

10.2. Normality

In Regression analysis, it is assumed that the variables have a normal distribution. The error between observed and predicted values (i.e. the residuals of the regression) should be normally distributed. This assumption can be checked by reviewing the Q-Q Plot, goodness of fit (e.g., the Kolmogorov-Smirnov test), PP plot or by developing residual histogram. plotting residual values on a histogram with a fitted normal curve.

Normal Q-Q Plot is a probability plot of the standardized residuals against the values that would be expected under normality. In the residual histogram method, the residual values are plotted on a histogram with a fitted normal curve. Then this plot is used to assess the assumption that the residuals are normally distributed. If there's normality, the points on this graph should fall on the straight 45-degree line (Fig. 6). If there's constant variance assumption, the points in the Scale-Location graph should be a random band around a horizontal line. Apart from these tests, there are other specialized code for testing regression assumptions.

How well the sample can predict a normal distribution can be observed from the matching between histogram and normal distribution. Syntax in R for generating residual histogram from Standardized Residuals (Listing 8) can be implemented in R base installation [9].

```
> #to create residual histogram
> hist(standardizedResiduals, freq = FALSE)
> #to add normal curve
> curve(dnorm, add = TRUE)
```

Listing 8: Syntax for generating residual histogram

10.3. PP Plot

A PP plot can also be used to assess the assumption that the residuals are normally distributed. This plot compares the empirical cumulative distribution function of a data set with a specified theoretical cumulative distribution function. For creating the PP plot in R, probability distribution is fist requirement which is generated using the pnorm(VAR) function in R base installation. VAR is the variable containing the residuals (Listing 9). An abline() function is included in the work to draw a diagonal line across plot for comparison purpose.

```
> #to get probability distribution for residuals
> probDist <- pnorm(standardizedResiduals)
> #to create PP plot
> plot(ppoints(length(standardizedResiduals)), sort(probDist), main = "PP
Plot", xlab = "Observed Probability", ylab = "Expected Probability")
> #to add diagonal line
> abline(0,1)
```

Listing 9: Syntax for creating PP plot

From the PP plot generated by the R script, normality can be detected (Fig. 8). Distribution is considered to be normal to the extent that the plotted points match the diagonal line. The extent to which the plotted points depart away from the diagonal line represents the non-normal feature.

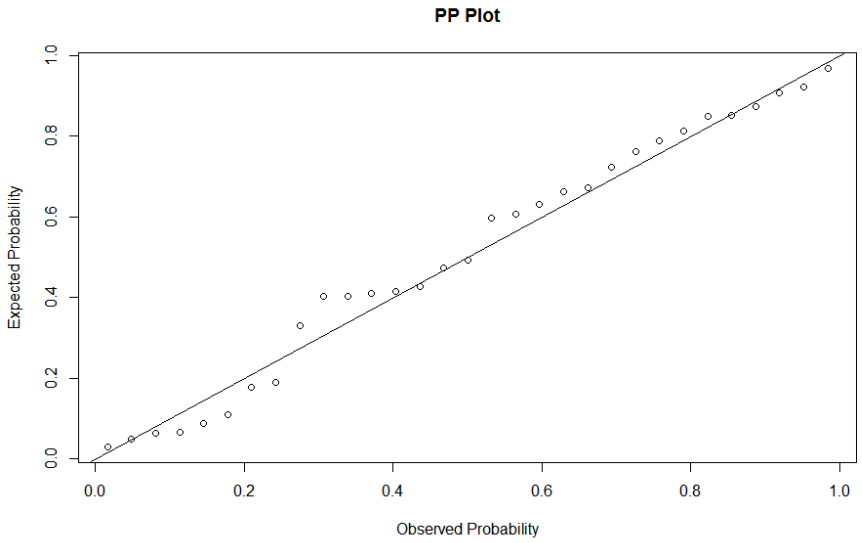

Figure 8: PP Plot

10.4. Residuals Histogram

Residuals are estimates of experimental error determined by subtracting the observed responses from the predicted response. Examining residuals is important in all statistical modeling because careful study of residuals can provide us whether our assumptions are reasonable, and our choice of model is appropriate. Observing the residual histogram (Fig 8), it can be analyzed whether the sample can predict a normal distribution in the population.

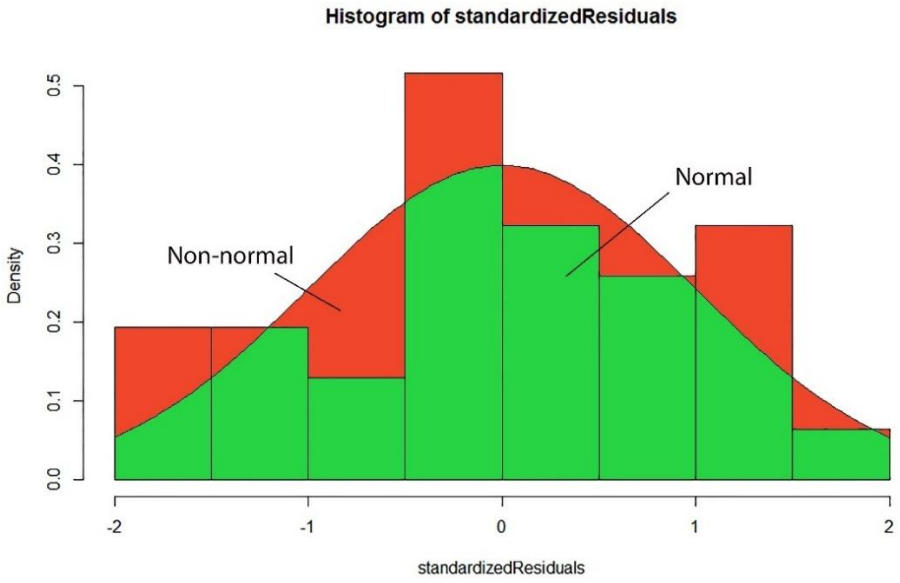

Figure 8: Residual histogram from a sample data

10.5. Homoscedasticity

In linear regression models, the assumption of homoscedasticity is important. It is assumed that the variance error term (i.e. the 'noise' or random disturbance in the relationship between the independent variables and the dependent variable) is the same across all values of the independent variables. Heteroscedasticity is the violation of homoscedasticity when the size of the error term differs across values of an independent variable. Homoscedasticity can be checked by the Breusch-Pagan test and the NCV test. The Breusch-Pagan test can be conducted by the bptest() function in the lmtest package [31]. The NCV Test can be conducted in the car package [29]. Breusch-Pagan test can be performed from the object 'fit' through the following syntax.

```
# run Breusch-Pagan test
> library(lmtest)
> bptest(fit, varformula = NULL, studentize = TRUE, data = list())
# run Non-constant Variance Score Test
> library (car)
> ncvTest(fit)
```

Listing 10: syntax for testing Breusch-Pagan value

In the syntax for lmtest package, 'fit' is the model and varformula describes only the potential explanatory variables. Both the methods have a p-value have less than a significant level of 0.05 when there is loss of homodescasticity.

In the Figure 6, Plot of Residuals vs Fitted and Scale-Location provide the information of homoscedasticity. If there is absolutely no heteroscedastity, there is an expectation of equal distribution of points throughout the range of X axis and a flat line (Fig 10).

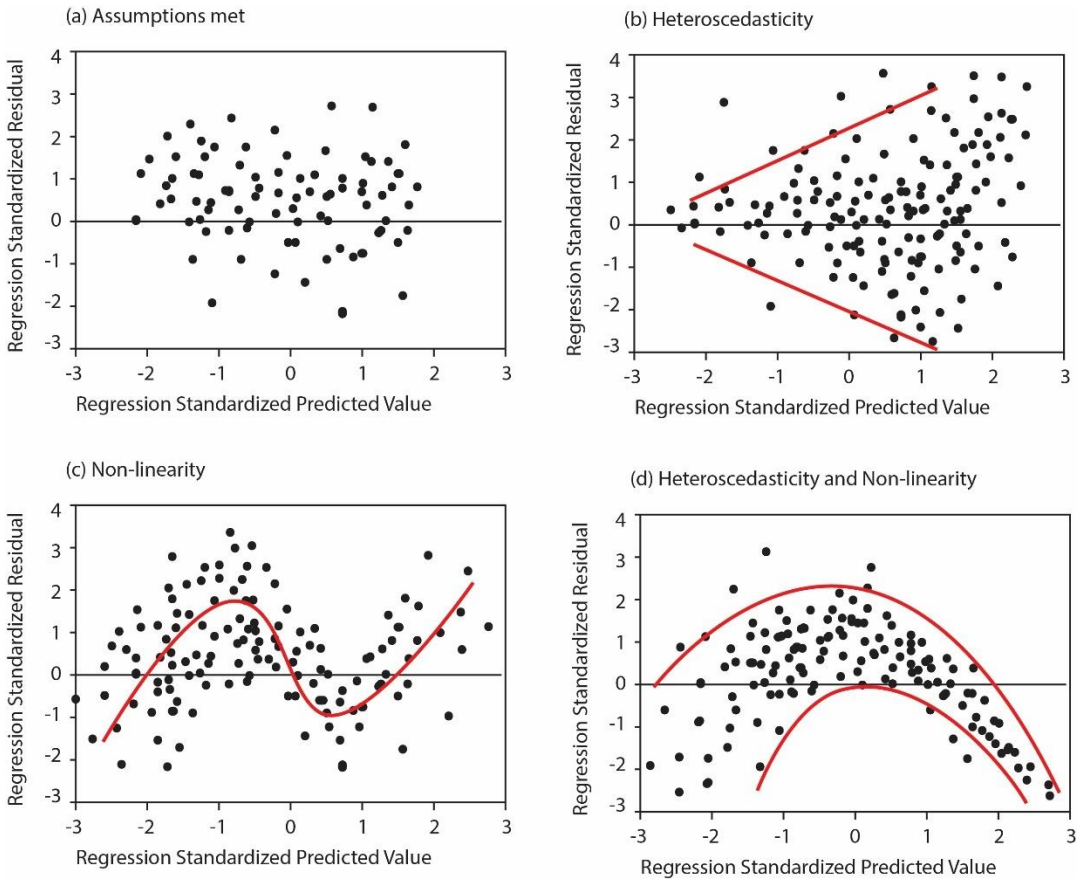

Figure 10: Plots showing assumptions, heteroscedasticity and non-linearity

When the p-values in Studentized Breusch-Pagan test in the 'lmtest' package and Non Constant Variance Score Test in 'car' package provide a significance level of 0.05, then null hypothesis can be accepted. It means that the variance of the residuals is constant and infer that heteroscedasticity is indeed absent.

10.6. No Multicollinearity

In MLR, there are assumptions that independent are not highly correlated with each other, i.e., there is no multicollinearity among them. Multicollinearity (also collinearity) is a phenomenon in which two or more predictor variables in a multiple regression model are highly correlated. It occurs when the independent variables are not independent from each other. It is a common practice to eliminate highly correlated descriptors in MLR equations for QSAR/QPSR model development. Multicollinearity is checked against four main criteria - correlation matrix, tolerance, variance inflation factor (VIF) and condition index.

In our approach, correlation matrix and VIF criteria are used. While detecting multicollinearity with variance inflation factor (VIF), the general rule, $\sqrt{vif} > 2$ indicates a multicollinearity problem. In R software, another approach is to calculate VIF by using 'vif()' function in the 'car' package [29]. Multicollinearity can be checked by using another handy tool called the ENMTool, which is developed for

environmental niche modelling [44]. This tool is a Perl script that is implemented through Tk+ package which is a graphical user interface toolkit for Tcl programming language.

In the first approach, checking multicollinearity is implemented by estimating cross-correlation using Pearson Correlation Coefficient in the R software [9] itself or ENMTools [44]. In the first approach, correlation matrix is generated by the cor() function in the R base installations. There are many alternative approaches for generating the correlation matrix, such as Hmisc package, corrplot etc.

From the matrix of Pearson's Bivariate Correlation among all independent variables, the correlation coefficients need to be smaller than .08.

Variables	Desc01	Desc02	Desc03	Desc04	Desc05	Desc06	Desc07	Desc08	Desc09	Desc10	Desc11
Desc01											
Desc02	-0.292										
Desc03	0.598	-0.158									
Desc04	-0.728	0.380	**-0.849**								
Desc05	**0.832**	0.005	0.195	-0.250							
Desc06	0.962	-0.422	0.717	-0.871	0.666						
Desc07	-0.703	0.558	-0.810	**0.975**	-0.201	-0.865					
Desc08	0.666	-0.279	0.332	-0.395	0.546	0.587	-0.403				
Desc09	0.858	-0.170	0.535	-0.650	0.749	0.849	-0.611	0.259			
Desc10	0.896	-0.153	0.268	-0.351	0.980	0.755	-0.332	0.637	0.765		
Desc11	0.967	-0.345	0.730	-0.877	0.676	0.995	-0.851	0.602	0.846	0.757	
Desc12	0.481	-0.654	0.595	-0.661	0.086	0.602	-0.732	0.435	0.307	0.224	0.577

In the matrix generated by either of the methods, r values generated by each pair of variables are checked. When r values are greater than or less than a particular threshold value (say +0.8 or -0.8), only one of the variable was selected for the model. Out of the pair of variables that possess multi-collinearity, which variable is to be selected is based on the potential relevance to the overall properties of the QSAR study.

According to the Topliss and Costello Rule, the ratio of training set chemicals to descriptors should be at least 5:1 in the linear regression to minimize the risk of chance correlations [45]. Restricting number of features having multicollinearity in MLR may hold true in common statistical analysis, but in QSAR, there are some exceptions. Multicollinearity has no effect on the overall fitness of a model whose purpose is strictly predictive in nature as opposed to explanatory [46]. While pruning highly correlated descriptors as default setting in commercial MLR software packages, one may overlook meaningful correlations. Sometimes, descriptors may act synergistically and may provide models that performed better than the sum of the individual components. In a study involving Density Functional Theory based QSAR, the model developed with hardness, E_{HOMO}, MR_{A-4} and MR_{B-4} could predict the activity of the set of chalcone

molecules against Mycobacterium tuberculosis [47]. In this study, descriptors hardness and E_{HOMO} are found to be related with the following formula :

$$\eta = \frac{E_{LUMO} - E_{HOMO}}{2}$$

Where η is the hardness and E_{HOMO} is the Energy of the Highest Occupied Molecular Orbital. It is up to the researchers to analyse the mechanisms of action or where such association can improve model's acceptance.

In other statistical analysis, multicollinearity is a genuine concern and can be addressed by applying any statistical approaches. However, in QSAR analysis, multicollinearity problem has to be addressed carefully. Perfect multicollinearity occurs when one of the independent variables in a regression model is perfectly correlated with another descriptor or a linear combination of other descriptors. Such issues can be easily handled. In lesser degrees of multicollinearity which are common, their diagnosis and assessment need important attention. There are instances where highly correlated, poorly performing, single descriptors may provide important information to the model [46]. In this case, domain knowledge of the researchers become more important.

10.7. Autocorrelaton

In the MLR, there is another assumption that there is no autocorrelation in the dataset. Autocorrelation is the situation where the residuals are not independent from each other. For instance, the value of y(x+1) is not independent from the value of y(x). Presence of autocorrelation can be checked by scatterplot or Durbin-Watson test. The Durbin-Watson tests the hypothesis that the residuals are not linearly correlated. The value of d ranges between 0 and 4 - with values around 2 indicate no autocorrelation. Rule of thumb is 1.5<d<2.5 shows that there is no auto correlation in the MLR dataset. The Durbin-Watson test is performed by using 'dwtest()' function in the package 'lmtest' [31] of the R software.

11. Outliers

Outliers are data point that aren't predicted well by the model and numerically distant from the rest of the data [48]. They are either unusually large positive or negative residuals. Positive residuals indicate that the model is underestimating the response value, while negative residuals indicate an overestimation. Points in the Q-Q plot that lie outside the confidence band are considered outliers. A rule of thumb is that standardized residuals that are larger than 2 or less than -2 need attention [49]. In our approach, the 'car' package that provides a statistical test for outliers is used. The outlierTest() function in this package reports the Bonferroni adjusted p-value for the largest absolute studentized residual [29].

Success of QSAR analysis depends on the identification and removal of the outliers [50] as MLR is very sensitive to outliers effects. Outliers are usually indicative either of measurement error or of population,

sometimes may occur also by chance. Particularly in QSAR, outlies in the data may not be due to statistical fluctuations or measurement errors but due to the presence of activity cliffs. These activity cliffs can be defined as the ratio of the difference in activity of two compounds to their 'distance' of separation in a given chemical space [51]. Presence of 'cliffs' in the descriptor space can lead to change in the bioactive properties dramatically on addition or removal of one or small group of chemicals. If an outlier is present, it is better to rerun to test after deleting the outlier data. In the following example, observation no. 5 is an outlier.

```
No Studentized residuals with Bonferonni p < 0.05
Largest |rstudent|:
   rstudent unadjusted p-value Bonferonni p
5 -2.732763           0.017091      0.52983
```

Listing 12 : Output from Bonferonni test

12. Influential Observations

Influential observations are observations that have a disproportionate impact on the values of the model parameters. For instance, the model may change dramatically with the removal of a single observation. There are two methods for identifying influential observations - Cook's distance, or D statistic [29]) and added variable plots [49]. Cook's D values greater than 4/(n-k-1), where n is the sample size and k is the number of predictor variables, indicate influential observations. Added variable plot and Cook's D plot can be drawn in the 'car' package [29].

13. Corrective Measures

When there are violations of regression assumptions, there are many approaches to deal with these issues. Some of the commonly occurring corrective measures against violation of assumptions.

13.1. Deleting observations

In this approach, identified outliers are deleted to improve a dataset's fit to the normality assumption. Influential observations are often deleted because they have an inordinate impact on the results. After largest outlier or influential observation is deleted, the model is refit. If there are still outliers or influential observations, the process is repeated until an acceptable fit is obtained. There should be some caution in deleting observations. If an observation is identified as an outlier because of data errors, inaccurate protocol etc., deleting the offending observations is reasonable. In certain cases, the unusual observation may be the most interesting thing about the data. Checking why this observation differs from the rest can provide great insight to uncovering the response variable. So here is the chance of serendipity for great discovery.

13.2. Transforming variables

When models don't meet the criteria of normality, linearity or homodescasticity assumptions, transformation of variables is an important option to correct the situation. Transformations involve replacing a variable Y with Y^{λ}.

(i) When the model violates the normality assumption, we may attempt a transformation of the response variable.

(ii) When the assumption of linearity is violated, a transformation of the predictor variables can help.

(iii) In the issues involving heteroscedasticity (nonconstant error variance), transformations of the response variables may help.

13.3. Normalization

Normally distributed data is more preferable for QSAR analysis. When data is not normally distributed, the cause of non-normality can be determined for taking up appropriate remedial actions. Data transformations can be done by various methods.

Some of the common examples of data transformations in our daily life are currency exchange (US $ to INR) and conversion of degree Celsius into degree Fahrenheit. These two transformations are called linear transformations as the original data is simply multiplied or divided by a specific constant or coefficient, or subtracted or added. However, such type of transformations does not change the shape of the data distribution and did not contribute much in normalization.

One of the popular methods is Box-Cox power transformation which is a procedure developed by statisticians George Box and David Cox. The process identified an appropriate exponent (lambda) to use to transform data into a 'normal shape'. The Lambda value indicates the power to which all data should be raised. In the process, the Box-Cox power transformation searches from Lambda = -5 to Lambda = +5 until the best value is found.

The Box-Cox power transformation is not a guarantee for normality in many cases. The method checks for the smallest standard deviation. It is necessary to always check the transformed data for normality using a probability plot. One limitation of Box-Cox Power transformation is that it only works if all the data is positive and greater than 0. To achieve this, a constant (c) is sometimes added to all data such that it all becomes positive before it is transformed. The transformation equation then becomes

$$Y^1 = (Y + C)^1$$

The **Box-Cox** procedure chooses an optimal transformation to remediate deviations from the assumptions of the linear regression model. For the linear model fit, we give the R commands (Listing 12) to receive a plot of the "log likelihood" of the parameter λ (lambda) against values of λ from -2 to 2.

```
# to get boxcox
> library(MASS)
> boxcox(mydata)
# refining the boxcox plot
> boxcox(mydata, lambda = seq(1, 2, 0.1))
```

Listing 14: Syntax for boxcox analysis

The dotted vertical line indicates that the ideal value of λ is about 1.5 (Fig. 9 A). To refine our estimate, we can change the range of λ to, say, from 1 to 2 by steps of 0.1.

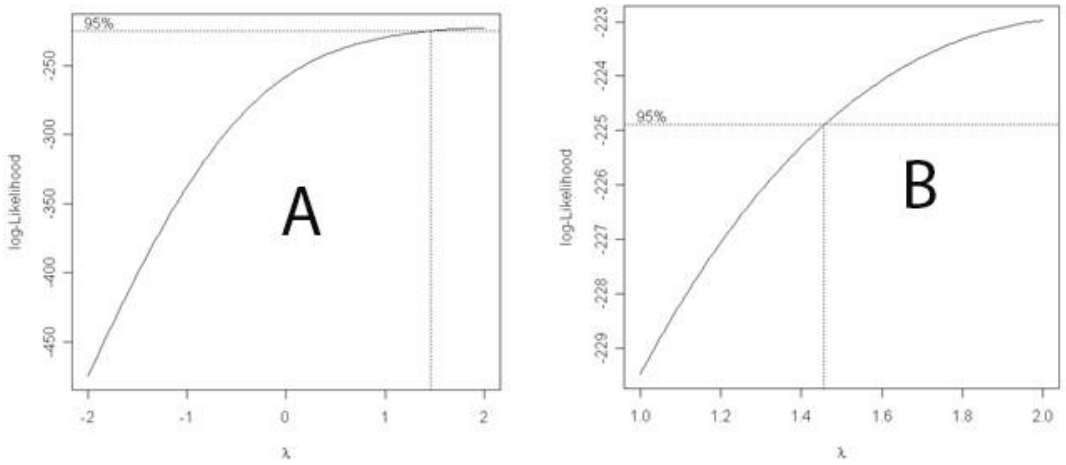

Figure 11 : Box Cox Plot

In the new plot (Fig 9 B), the estimates are refined to indicate the best value of lambda. The plot indicates that the best value of λ is about 1.45. We then transform the response variable accordingly, add it to the original data set, and run a new linear model.

Table 2: Common Box Cox Transformation

Lambda	**Y' (transformed value of Y)**
-2	$Y^{-2} = \frac{1}{Y^2}$
-1	$Y^{-1} = \frac{1}{Y}$
-0.5	$Y^{-0.5} = \frac{1}{\sqrt{Y}}$
0	log(y)
0.5	$Y^{0.5} = \sqrt{Y}$
1	$Y^{1} = Y$

2	Y^2

13.4. Adding or deleting descriptors

Changing the variables in the model will impact the fit. Deleting descriptor is important approach for dealing with multicollinearity. If the sole purpose of QSAR is to make predictions, then multicollinearity isn't a problem [46]. If we want to extend the QSAR up to interpretations about individual descriptor, we have to deal with multicollinearity issues. Most common approach is to delete one of the descriptors involved in the multicollinearity with the square root VIF greater than 2. Other alternatives include use of ridge regression and genetic algorithm for selecting the best set of descriptors.

13.5. Different approach

There are different statistical methods that can be attempted if MLR approaches could not provide satisfactory results. As explained above, one approach is to use ridge regression for addressing multicollinearity issues. If there are outliers or influential observations, robust regression model prefers more. If there's violation of normality assumption, nonparametric regression is more suited. When there's significant nonlinearity, application of nonlinear regression model is more appropriate.

14. Example of QSAR with specific phytochemicals

The QSAR becomes efficient methods for finding probable phytochemical candidate for determination of effective and less toxic drugs [4]. Prior knowledge of the biological system, various factors concerning physiological process and pathological conditions, molecular dynamics and their properties, modulating factors of drugs highly contribute to the development of an effective QSAR models [14]. Many QSAR studies have contributed to the determination of the biological properties of various phytochemicals. Some of these phytochemicals are highlighted.

In a study conducted by the Agarwal et al., 2022 [52], employed a random forest-based binary QSAR model to screen a library of natural compounds for their anticancer activity against EGFR double mutant, and identified a few leads with potential for overcoming drug resistance in cancer.

Another study carried out by Shukla et al., 2020 [53]; QSAR model used in this study predicted the biological activities of GA derivatives against triple-negative breast cancer cell line MDA-MB-231. The model identified specific structural features that significantly contributed to the cytotoxic activity of the compounds. The addition of an acetyl group at C-3 increased the lipophilicity of GA and improved its cytotoxicity, while substitutions at C-30 with propylamide, butyl amide, and amino ethyl amide decreased the cytotoxicity potential. The study also confirmed that the C-30 carboxylic group is crucial for GA-based

cytotoxic activity. The results of this study provide valuable insights into early lead discovery against metastatic breast cancers.

A study by Yadav et al., on 3D-QSAR and docking analysis of ursolic acid derivatives for anticancer activity against bladder cancer cell line T24 showed that structural modifications of the compounds could significantly affect their biological activity. The 3D-QSAR models developed in this study helped to identify the important structural features required for enhancing the activity of the derivatives. The study revealed that the compounds with a bulky and electron-withdrawing substituent at position C-3 of the triterpenoid skeleton showed higher anticancer activity by inhibiting the NF-kB pathway. The docking analysis further supported the biological activity of the derivatives by showing their favorable binding interactions with the active site of the target protein. The findings of this study could be useful for the design and development of potent anticancer agents targeting the NF-kB pathway. Overall, the study highlights the potential of 3D-QSAR and docking analysis in predicting the biological activity of compounds and guiding the development of new drugs [54].

The field-based 3D-QSAR model proposed in the study [55] provided a molecular-level understanding and a clear structure-activity relationship for triterpene maslinic acid and its analogs. The model was successfully applied to virtually screen a large number of compounds for potential anticancer activity against breast cancer cell line MCF7. The model's acceptable regression and cross-validation coefficients demonstrated its accuracy and reliability for identifying active compounds. The activity-atlas models provided a global view of the training set and helped in the identification of key features responsible for SAR. The virtual screening process, which involved applying filters for oral bioavailability, drug-like features, chemical synthesis, and cellular target docking, resulted in the identification of P-902 as the top hit. Overall, the results of this study demonstrate the usefulness of QSAR models in early anticancer drug discovery and lead optimization from natural active scaffolds. QSAR modeling can be a valuable tool in identifying potential drug candidates and in reducing the time and cost associated with traditional drug discovery methods.

Some more QSAR based studies with phytochemicals are listed in Table3.

Table 3: Some of the QSAR studies on the phytochemicals using different software

Natural Product Source	**Disease/Properties**	**Reference**
Withanolide present in roots and leaves of Withania somnifera	Human breast cancer cell lines	[54]
Pulvinic acid and coumarine derivatives	Antioxidant properties against radiation sources of Fenton, gamma, and UV	[56]
Bioactive compounds of Gracilaria corticata	Lipinski rule of five and ADMET prediction	[57]

Natural Product Source	Disease/Properties	Reference
705 phytochemical compounds	SARS-CoV and MERS-CoV	[58]
Leaf fractionated compounds from Gongronema latifolium	type 2 diabetes mellitus	[59]
40 antiviral phytochemicals	NS3 protease of dengue virus	[24]
Plant-derived essentials oils	fumigant and topical activities on Musca domestica	[60]

Acknowledgements:

The authors are grateful DelCON's e-Journal Access Facility. Lutfun Nahar gratefully acknowledges the financial support of the European Regional Development Fund - Project ENOCH (No. CZ.02.1.01/0.0/0.0/16_019/0000868) and the Czech Agency Grant - Project 23-05474S

References

1. Muratov, E.N., et al., QSAR without borders. *Chemical Society Reviews*, 2020; **49**(11): 3525-3564.
2. Selassie, C. and R.P. Verma, History of quantitative structure-activity relationships. *Burger's medicinal chemistry and drug discovery*, 2003; **1**: 1-48.
3. Veerasamy, R., QSAR—An Important In-Silico Tool in Drug Design and Discovery, in *Advances in Computational Modeling and Simulation*. 2022, Springer. p. 191-208.
4. Das, A.P. and S.M. Agarwal, Recent advances in the area of plant-based anti-cancer drug discovery using computational approaches. *Molecular Diversity*, 2023: 1-25.
5. Ojo, O.A., et al., Deciphering the interactions of bioactive compounds in selected traditional medicinal plants against Alzheimer's diseases via pharmacophore modeling, auto-QSAR, and molecular docking approaches. *Molecules*, 2021; **26**(7): 1996.
6. Omoboyowa, D.A., Exploring molecular docking with E-pharmacophore and QSAR models to predict potent inhibitors of 14-α-demethylase protease from Moringa spp. *Pharmacological Research-Modern Chinese Medicine*, 2022; **4**: 100147.
7. McCullough, B.D., Assessing the reliability of statistical software: Part I. *The American Statistician*, 1998; **52**(4): 358-366.
8. Emmert-Streib, F., *Statistical modelling of molecular descriptors in QSAR/QSPR*. 2012: John Wiley & Sons.
9. Team, R.C., R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. *http://www. R-project. org/*, 2016.
10. Lo, Y.-C., et al., Machine learning in chemoinformatics and drug discovery. *Drug discovery today*, 2018; **23**(8): 1538-1546.

11. Puzyn, T., Recent advances in QSAR studies methods and applications. 2022.

12. McNaught, A.D. and A. Wilkinson, Compendium of chemical terminology. IUPAC recommendations. 1997.

13. Golbraikh, A. and A. Tropsha, QSAR modeling using chirality descriptors derived from molecular topology. *Journal of chemical information and computer sciences*, 2003; **43**(1): 144-154.

14. Kar, S. and K. Roy, QSAR of phytochemicals for the design of better drugs. *Expert opinion on drug discovery*, 2012; **7**(10): 877-902.

15. Roy, K., et al., Statistical methods in QSAR/QSPR. *A Primer on QSAR/QSPR Modeling: Fundamental Concepts*, 2015: 37-59.

16. Liu, P. and W. Long, Current mathematical methods used in QSAR/QSPR studies. *International Journal of Molecular Sciences*, 2009; **10**(5): 1978-1998.

17. Sharma, S. and V. Bhatia, Recent trends in QSAR in modelling of drug-protein and protein-protein interactions. *Combinatorial Chemistry & High Throughput Screening*, 2021; **24**(7): 1031-1041.

18. Wang, Z. and J. Chen, Applicability Domain Characterization for Machine Learning QSAR Models, in *Machine Learning and Deep Learning in Computational Toxicology*. 2023, Springer. p. 323-353.

19. Roth, B.L., D.J. Sheffler, and W.K. Kroeze, Magic shotguns versus magic bullets: selectively non-selective drugs for mood disorders and schizophrenia. *Nature reviews Drug discovery*, 2004; **3**(4): 353-359.

20. Luan, F., et al., Computer-aided nanotoxicology: assessing cytotoxicity of nanoparticles under diverse experimental conditions by using a novel QSTR-perturbation approach. *Nanoscale*, 2014; **6**(18): 10623-10630.

21. Kubinyi, H., Chemogenomics in drug discovery. *Ernst Schering Res Found Workshop*, 2006(58): 1-19.

22. Playe, B. and V. Stoven, Evaluation of deep and shallow learning methods in chemogenomics for the prediction of drugs specificity. *Journal of cheminformatics*, 2020; **12**(1): 11.

23. Chakravarti, S.K. and S.R.M. Alla, Descriptor free QSAR modeling using deep learning with long short-term memory neural networks. *Frontiers in artificial intelligence*, 2019; **2**: 17.

24. Rahman, M.M., et al., Antiviral phytochemicals as potent inhibitors against NS3 protease of dengue virus. *Computers in Biology and Medicine*, 2021; **134**: 104492.

25. Islam, R., et al., A molecular modeling approach to identify effective antiviral phytochemicals against the main protease of SARS-CoV-2. *Journal of Biomolecular Structure and Dynamics*, 2021; **39**(9): 3213-3224.

26. Basu, A., A. Sarkar, and U. Maulik, Molecular docking study of potential phytochemicals and their effects on the complex of SARS-CoV2 spike protein and human ACE2. *Scientific reports*, 2020; **10**(1): 17699.

27. Williams, G., *Data mining with Rattle and R: The art of excavating data for knowledge discovery*. 2011: Springer Science & Business Media.

28. Kursa, M. and W. Rudnicki, Boruta: wrapper algorithm for all relevant feature selection, Version 5.2. 0. 2017.

29. Fox, J. and S. Weisberg, *An R companion to applied regression*. 2011: Sage publications.

30. Lumley, T., Leaps: Regression subset selection. R package version 3.0. Based on Fortran code by Alan Miller. 2017.

31. Hothorn, T., et al., lmtest: Testing Linear Regression Models. R package version 0.9-34. 2015.

32. Jiratchayut, K. and C. Bumrungsup, Penalized linear regression methods where the predictors have grouping effect. *Thailand Statistician*, 2019; **17**(2): 212-222.

33. Cerdeira, J.O., et al., Subselect : Selecting Variable Subsets Version 0.13. 2017.

34. Shamsara, J., Ezqsar: an R package for developing QSAR models directly from structures. *The Open Medicinal Chemistry Journal*, 2017; **11**: 212.

35. Grabner, M., K. Varmuza, and M. Dehmer, RMol: a toolset for transforming SD/Molfile structure information into R objects. *Source Code for Biology and Medicine*, 2012; **7**(1): 1-4.

36. Tsiliki, G., et al., RRegrs: an R package for computer-aided model selection with multiple regression models. *Journal of cheminformatics*, 2015; **7**(1): 1-16.

37. Murrell, D.S., et al., Chemically Aware Model Builder (camb): an R package for property and bioactivity modelling of small molecules. *Journal of cheminformatics*, 2015; **7**: 1-10.

38. Alexander, D.L., A. Tropsha, and D.A. Winkler, Beware of R 2: simple, unambiguous assessment of the prediction accuracy of QSAR and QSPR models. *Journal of chemical information and modeling*, 2015; **55**(7): 1316-1322.

39. Varmuza, K., P. Filzmoser, and M. Dehmer, Multivariate linear QSPR/QSAR models: Rigorous evaluation of variable selection for PLS. *Computational and structural biotechnology journal*, 2013; **5**(6): e201302007.

40. Goodarzi, M., B. Dejaegher, and Y.V. Heyden, Feature selection methods in QSAR studies. *Journal of AOAC International*, 2012; **95**(3): 636-651.

41. Rogers, D. and A.J. Hopfinger, Application of genetic function approximation to quantitative structure-activity relationships and quantitative structure-property relationships. *Journal of chemical information and computer sciences*, 1994; **34**(4): 854-866.

42. Ripley, B., et al., MASS: Support functions and datasets for Venables and Ripley's MASS. R package version. 2017; **7**: 3 - 47.

43. Cherkasov, A., et al., QSAR modeling: where have you been? Where are you going to? *Journal of medicinal chemistry*, 2014; **57**(12): 4977-5010.

44. Warren, D.L., R.E. Glor, and M. Turelli, ENMTools: a toolbox for comparative studies of environmental niche models. *Ecography*, 2010; **33**(3): 607-611.

45. Topliss, J.G. and R.J. Costello, Chance correlations in structure-activity studies using multiple regression analysis. *Journal of medicinal chemistry*, 1972; **15**(10): 1066-1068.

46. Peterangelo, S.C. and P.G. Seybold, Synergistic interactions among QSAR descriptors. *International journal of quantum chemistry*, 2004; **96**(1): 1-9.

47. Barua, N., et al., DFT-based QSAR Models to Predict the Antimycobacterial Activity of Chalcones. *Chemical Biology & Drug Design*, 2012; **79**(4): 553-559.

48. Hawkins, D.M. and D. Hawkins, Multivariate outlier detection. *Identification of outliers*, 1980: 104-114.

49. Kabacoff, R., R in Action. . *Manning Publications Co.*, 2011.

50. Tropsha, A., Best practices for QSAR model development, validation, and exploitation. *Molecular informatics*, 2010; **29**(6-7): 476-488.

51. Maggiora, G.M., On outliers and activity cliffs why QSAR often disappoints. 2006, ACS Publications. 1535-1535.

52. Agarwal, S.M., P. Nandekar, and R. Saini, Computational identification of natural product inhibitors against EGFR double mutant (T790M/L858R) by integrating ADMET, machine learning, molecular docking and a dynamics approach. *RSC advances*, 2022; **12**(26): 16779-16789.

53. Shukla, A., et al., 2D-and 3D-QSAR modelling, molecular docking and in vitro evaluation studies on 18β-glycyrrhetinic acid derivatives against triple-negative breast cancer cell line. *Journal of Biomolecular Structure and Dynamics*, 2020; **38**(1): 168-185.

54. Yadav, D.K., et al., Molecular docking, QSAR and ADMET studies of withanolide analogs against breast cancer. *Drug Design, Development and Therapy*, 2017: 1859-1870.

55. Alam, S. and F. Khan, 3D-QSAR studies on Maslinic acid analogs for Anticancer activity against Breast Cancer cell line MCF-7. *Scientific reports*, 2017; **7**(1): 6019.

56. Ahmadi, S., et al., Predictive QSAR modeling for the antioxidant activity of natural compounds derivatives based on Monte Carlo method. *Molecular Diversity*, 2021; **25**: 87-97.

57. Biswal, A., et al., 2D QSAR, Admet prediction and multiple receptor molecular docking strategy in bioactive compounds of Gracilaria corticata against Plasmodium falciparum (contractile Protein). *Informatics in Medicine Unlocked*, 2019; **17**: 100258.

58. Bhargav, A., et al., Phytovid19: a compilation of phytochemicals research in coronavirus. *Structural Chemistry*, 2022; **33**(6): 2169-2177.

59. Ajiboye, B.O., et al., Screening of potential antidiabetic phytochemicals from Gongronema latifolium leaf against therapeutic targets of type 2 diabetes mellitus: multi-targets drug design. *SN Applied Sciences*, 2022; **4**(1): 14.

60. Duchowicz, P.R., et al., QSAR models for insecticidal properties of plant essential oils on the housefly (Musca domestica L.). *SAR and QSAR in Environmental Research*, 2021; **32**(5): 395-410.