ChemSAR: an online pipelining platform for molecular SAR modeling

Jie Dong¹, Zhi-Jiang Yao¹,², Min-Feng Zhu¹,², Ning-Ning Wang¹, Ben Lu², Alex F. Chen¹,², Ai-Ping Lu³, Hongyu Miao⁴, Wen-Bin Zeng¹ and Dong-Sheng Cao¹,3*

Abstract

Background: In recent years, predictive models based on machine learning techniques have proven to be feasible and effective in drug discovery. However, to develop such a model, researchers usually have to combine multiple tools and undergo several different steps (e.g., RDKit or ChemoPy package for molecular descriptor calculation, ChemAxon Standardizer for structure preprocessing, scikit-learn package for model building, and ggplot2 package for statistical analysis and visualization, etc.). In addition, it may require strong programming skills to accomplish these jobs, which poses severe challenges for users without advanced training in computer programming. Therefore, an online pipelining platform that integrates a number of selected tools is a valuable and efficient solution that can meet the needs of related researchers.

Results: This work presents a web-based pipelining platform, called ChemSAR, for generating SAR classification models of small molecules. The capabilities of ChemSAR include the validation and standardization of chemical structure representation, the computation of 783 1D/2D molecular descriptors and ten types of widely-used fingerprints for small molecules, the filtering methods for feature selection, the generation of predictive models via a step-by-step job submission process, model interpretation in terms of feature importance and tree visualization, as well as a helpful report generation system. The results can be visualized as high-quality plots and downloaded as local files.

Conclusion: ChemSAR provides an integrated web-based platform for generating SAR classification models that will benefit cheminformatics and other biomedical users. It is freely available at: http://chemsar.scbdd.com.

Keywords: Online modeling, Molecular descriptors, Machine learning, QSAR/SAR, Cheminformatics

Background

The increasing availability of data on characteristics and functions of biomolecules and small chemical compounds enables researchers to better understand various chemical, physical and biological processes or activities with the use of machine learning methods [1–7]. Particularly in the drug discovery field, machine learning methods are frequently applied to build in silico predictive models in studies of structure–activity relationships (SAR) and structure–property relationships (SPR) to assess or predict various drug activities [8, 9], and ADME/T properties [10–16]. Nowadays, with the development of various public data sources (e.g., ChEMBL [17], PubChem [18], and DrugBank [19]), more and more scientific studies are utilizing predictive SAR/SPR models to perform virtual screening [20], study drug side effects [21–24], predict drug–drug interactions [25] or drug–target interactions [26–28], and investigate drug repositioning [29, 30]. Undoubtedly, robust and predictive SAR models built upon machine learning techniques provide a powerful and effective way for pharmaceutical scientists to tackle the aforementioned problems; however, there still exist two major barriers to overcome.

First, owing to the fusion of different scientific disciplines, a higher level of background knowledge and professional skills is required to solve many existing biological problems. For example, to reliably predict drug
ADME/T properties, a researcher must be familiar with both pharmacokinetics and a modern programming language [14]; however, in many cases, researchers from the pharmaceutical or biomedical fields may lack formal training in computing skills. It may thus become necessary to save these investigators from tedious programming or deployment work such that they can focus on solving scientific problems.

Second, even if a researcher acquired the related background knowledge and computing skills, it is very time-consuming to build a predictive model as a number of steps are needed, including molecule representation, feature filtering, selection of a suitable machine learning method, prediction of new molecules, and relevant statistical analysis. In particular, the researcher needs to select and combine different tools to accomplish these steps; for example, using RDKit [31] to calculate molecular descriptors, using libSVM [32] or scikit-learn [33] to establish a model, using ggplot2 [34] to plot or visualize the results. However, the selection and integration of such tools involve lots of programming issues and efforts.

For molecular representation, tools like RDKit, CDK [35], Chemopy [36], OpenBabel [38], Cinfony [39], PyDPI [40], Rcpi [41], have been developed to provide thousands of molecular descriptors. For building SAR/SPR models based on machine learning algorithms, a series of package have been implemented, including scikit-learn in Python, and pls [42], earth, caret [43], randomForest [44], kernlab [45], and RRreg [46] in R. Visualization packages like matplotlib [47], ggplot2 and seaborn [48] are also freely available to produce high-quality statistical graphics. In addition, several online web services such as ChemDes [49], BioTriangle [50], E-DRAGON [51], QSAR4U [52] and OpenTox [53] are also available for drug discovery purpose. However, these tools are developed independently using different programming languages and APIs such that a unified and comprehensive platform is desirable to release biomedical investigators from such tedious and repeated efforts.

Toward this goal, a few previous studies made attempts to integrate one or two of the steps into a single package. For instance, the VCCLAB group [51, 54] developed E-DRAGON, an online platform for DRAGON software, to calculate various molecular descriptors, and also provided several online machine learning tools like PLSR and ASNN for model building. However, these tools are independent of each other and cannot be further integrated together to accomplish the entire modeling process. The OChem platform [55] was developed to provide a practical online chemical database. Also, this platform provides a modelling environment that enables users to standardize molecules, calculate molecular descriptors and build QSAR models. However, the OChem database is not specialized for SAR modelling and thus still lacks some essential functionalities like feature selection and advanced statistical analysis. Its modeling function is solely based on small molecules and thus cannot be used to analyze other independent biomedical dataset. More recently, Murrell et al. [56] developed a relatively well-integrated R package, named ccmB, for QSAR modeling. However, it does require users to have sufficient programming skills in R. In 2010, Chembench [57, 58] was developed and make progress in the simplicity of the use of QSAR modelling for analyzing experimental chemical structure–activity data. Other software applications that should be mentioned include eTOXlab [59], AZOrange [60], QSARINS [61], OECD QSAR Toolbox [62], Build-QSAR [63], Molecular Operating Environment (MOE) [64] and Discovery Studio (DS) [65]; however, these software are either commercial or difficult for users to deploy by themselves.

In view of these limitations, we implemented a web-based platform, called ChemSAR, as an online pipeline for SAR model development. ChemSAR integrates a set of carefully selected tools and provides a user-friendly web interface and allows users to complete the entire workflow via a step-by-step submission process without involving any programming effort. Currently, ChemSAR is mainly designed for molecular SAR analysis and is capable of accomplishing seven modeling steps: (1) structure preprocessing, (2) molecular descriptor calculation, (3) data preprocessing, (4) feature selection, (5) model building and prediction, (6) Model interpretation and (7) statistical analysis. These seven steps together with several ancillary tools are implemented in six modules: (1) User space, (2) Structure preprocessing, (3) Data preprocessing, (4) Feature selection, (5) Model interpretation, (6) Tools. The six modules form an integrated pipeline for modeling, but each of these modules can also be used as a standalone tool. The whole workflow is shown in Fig. 1.

Methods
Implementation
The whole project runs on an elastic compute service (ECS) server of Aliyun. The number of CPU cores and memory are automatically allocated to the running instances on demand, which ensures the elastically stretchable computing capability. Python/Django and MySQL are used for server-side programming, and HTML, CSS, JavaScript are employed for front side web interfaces. The realization of functionality should go through three main components (MCV short for Model-Control-View model). To illustrate the implementation of the ChemSAR architecture, we consider the functionality of “Feature Calculation” as an example and the corresponding diagram is shown in Fig. 2 (see also Additional
Fig. 1 The pipelining of ChemSAR. It contains six main modules: (1) User space, (2) Structure preprocessing, (3) Data preprocessing, (4) Modelling process, (5) Model interpretation (6) Tools. Each of them not only could be utilized as one part of the whole pipelining but also could be used as an independent tool.
file 1). This module consists of four *.py files and a template folder: models.py acts as “M” to access the database; views.py acts as “C” to realize the functionality; the functions.py acts as a library to store the key calculation procedures which could be called into views.py; the forms.py stores input forms that can be used in templates; The HTML pages in the template folder act as “V” to visualize the results. Firstly, users go to the index page of “Feature Calculation” and submit the request. Then, the function: fingerprint_list (from views.py) executes as the back-end calculation program. In this function, (1) the input data from users is handled; (2) the input data is saved into database; (3) the key calculation function is called and executed; (4) the calculation result is stored into database and provided as file for download; (5) all the related variables are rendered into content for view. Among them, UserData and FeatureData are called from models.py to store users’ data and result data; handle_uploaded_file and calcChemopyFingerprints_list are imported from functions.py to store the uploaded file and calculate specified fingerprints. At last, the calculated fingerprints values will be rendered to static contents displayed in fingerprint_result3.html. As an easy-to-use web service, ChemSAR supports commonly-used file formats for data exchange between the server-side and the client-side. Specifically, simplified molecular input line entry specification (SMILES) and Structure Data Format (SDF) are acceptable molecular file formats (or users can convert their files into these two formats using OpenBable [37] or ChemCONV [49]). The modeling results will be presented as HTML web pages, but users can download the results in SDF, CSV, PNG or PDF format (see Table 1 for details).

User interface
To accomplish the complex modelling steps by using a web-based tool, a user-friendly interface is very necessary. In ChemSAR, database and session technologies are utilized to help develop a complete job submission and user space system. The AJAX technology is used in those processes that usually take a long time to finish, which makes it possible to check the status of different jobs at a convenient time. Besides, a logging system is developed to make sure that every step or wrong operation will trigger user-friendly tips or messages. The user interface consists of three main parts: the “Model” section, “Manual” section and “Help” section. The “Model” section is the main entrance for structure preprocessing, data preprocessing, modelling process and tools. The “Manual” section describes theories and requirements.

Fig. 2 The process of calculating molecular fingerprints—an example to explain the development methods of ChemSAR
for each module. The “Help” section gives detailed explanations of each module and a standard example of each step of building an SAR classification model.

## Module functionality

### User space

In general, to build a model, there is a need of storage space for user’s data and computing results. In this project, the “User space” module is developed to enable users to view, download and reuse all related files or models at any time.

### Structure preprocessing

It is very difficult for SAR practitioners to collect and integrate chemical structures from multiple sources due to the use of different structural representations, diverse file formats, distinct drawing conventions and the existence of labeling mistakes in such sources [66]. Therefore, preprocessing and standardizing these structures are very important tasks that will ensure the correctness of graphical representation and the consistence of molecular property calculation. Hence, we developed the “Structure preprocessing” module for molecule structure standardization based on the RDKit package. This module consists of three sub-modules, called Validation of molecules, Standardization of molecules, and Custom preprocessing, respectively. The “Validation of molecules” sub-module can check and visualize molecular structures. A warning message will be triggered if any anomaly is detected (e.g., a molecule has zero atoms, or has multiple fragments, or is not an overall neutral system, or contains isotopes), and both the molecular structure and the validation result will be displayed in an interactive table. The “Standardization of molecules” sub-module consists of the following steps: removing any hydrogen from

### Table 1 The valid file formats and requirements for each module

| Module name                  | Input          | Output          | Description                                                                 |
|------------------------------|----------------|-----------------|------------------------------------------------------------------------------|
| Feature calculation          | *.smi; *.sdf   | *.csv          | The standard version of molfile in SDF must be V2000; 2D or 3D information are both valid; the first row of *.csv file is the descriptor names; the first column is the SMILES of molecules |
| Model selection              | *.csv          | Data table in the page | The first column of X_train file can be molecular identifiers like molecular names or IDs; the first row of X_train file must be descriptor names; the first column of y_train file must be the same with X_train file; the second column must be experimental values of the sample (different presentation styles of classes must be converted into 0 or 1) |
| Validation of molecules      | *.sdf          | *.csv          | The standard version of molfile in SDF must be V2000                           |
| Standardization of molecules | *.sdf          | *.sdf          | The same with “Validation of molecules”                                        |
| Custom preprocessing         | *.sdf          | *.sdf          | The same with “Validation of molecules”                                        |
| Imputation of missing values | *.csv          | *.csv          | The first row of input file must be header like descriptor names; each column including the first one must be feature values like descriptor values |
| Removing low variance features | *.csv        | *.csv          | The same with “Imputation of missing values”                                   |
| Removing high correlation features | *.csv   | *.csv          | The same with “Imputation of missing values”                                   |
| Univariate feature selection | *.csv          | *.csv          | The first row of input file must be header like descriptor names; each column from the first one to the penultimate one must be feature values like descriptor values; The last column must be experimental values of the sample (different presentation styles of classes must be converted into 0 or 1) |
| Tree-based feature selection | *.csv          | Data table in the page; *.csv | The same with “Univariate feature selection”                                   |
| RFE feature selection        | *.csv          | Data table in the page; *.csv; *.png | The four columns of input file must be in order: molecular identifier, predict label, predict probability, experimental value; the label name can be defined by users |
| Statistical analysis         | *.csv          | Data table in the page; *.png   | The same with “Univariate feature selection”                                   |
| Random training set split    | *.csv          | *.csv          | The same with “Model selection”                                               |
| Diverse training set split   | *.sdf          | *.sdf          | The de facto standard version of molfile in SDF must be V2000; 2D or 3D information are both valid |
| Feature importance           | *.csv          | *.csv          | The same with “Univariate feature selection”                                   |
the molecule, sanitizing the molecule, breaking certain covalent bonds between metals and organic atoms, correcting functional groups and recombining charges, reionizing a molecule such that the strongest acids ionize firstly, discarding tautomeric information and retaining a canonical tautomer. Different from the one-click process implemented in “Standardization of molecules”, the “Custom preprocessing” sub-module provides flexible options for users to construct customized standardization process according to their own preferences of operations and execution orders.

Data preprocessing
Feature selection is one of the focuses of many SAR-based researches, for which datasets with tens (or hundreds) of thousands of variables need to be analyzed and interpreted. The variables from the descriptor calculation step usually need to be selected for the following reasons: removing unneeded, irrelevant or redundant features, simplifying models for easiness of interpretation, and shortening training time [67]. This module is built upon the scikit-learn package and consists of six sub-modules, including imputation of missing values (imputer), removal of low variance features (rm_var), removal of highly correlated features (rm_corr), univariate feature selection (select_univariate), tree-based feature selection (select_tree_based) and recursive feature elimination (select_RFE). The imputer module can impute missing values (e.g., nan) in the data. The rm_var and rm_corr modules remove features by a predefined threshold of variance or correlation coefficient without incurring a significant loss of information. The select_univariate module works by selecting \( k \) best features based on univariate statistical tests (e.g., Chi square or F tests). The select_tree_based module discards trivial features according to the importance computed using an estimator of randomized decision trees. The select_RFE module performs recursive feature elimination in a cross-validation loop to find the optimal number of features. Here, an estimator of support vector classification with a linear kernel is invoked to compute a cross-validated score for each recursive calculation. After the calculation, a figure is displayed on the result page to show the relationship between the number of features and the cross-validation scores. A table that contains the optimal number of features, the feature ranking and the cross-validation scores is also presented there.

Modeling process
The core steps of building an SAR model are implemented in the “Modeling process” module. It contains four sub-modules: feature calculation, model selection, model building, and prediction.

Feature calculation
In this project, we developed the feature calculation sub-module as an online tool [36], which allows users to calculate 783 molecular descriptors from 12 feature groups (see Table 2). These features cover a relatively broad range of molecular properties and are carefully selected based on our experience. In recent years, molecular fingerprints are widely used in drug discovery area, especially for similarity search, virtual screening and QSAR/SAR analysis due to their computational efficiency when handling and comparing chemical structures. In this sub-module, ten types of molecular fingerprint algorithms are implemented (see Table 2). These molecular fingerprints have been shown to have a good performance in characterizing molecular structures.

Table 2 The list of molecular descriptors computed by ChemSAR

| Feature group   | Features                                    | Number of descriptors |
|-----------------|---------------------------------------------|-----------------------|
| Constitution    | Molecular constitutional descriptors        | 30                    |
| Topology        | Topological descriptors                     | 35                    |
| Connectivity    | Molecular connectivity indices              | 44                    |
| E-state         | E-state descriptors                         | 245                   |
| Kappa           | Kappa shape descriptors                     | 7                     |
| Basak           | Basak descriptors                           | 21                    |
| Burden          | Burden descriptors                          | 64                    |
| Autocorrelation | Moreau-Broto autocorrelation                | 32                    |
| Charge          | Charge descriptors                          | 25                    |
| Property        | Molecular property                         | 6                     |
| MOE-type        | MOE-type descriptors                        | 60                    |
| CATS            | CATS descriptors                            | 150                   |
| Fingerprints    | Topological-Torsion fingerprints           | 1024                  |
|                 | MACCS keys                                 | 167                   |
|                 | FP4 fingerprints                            | 307                   |
|                 | FP2 fingerprints                            | 1024                  |
|                 | FP3 fingerprints                            | 210                   |
|                 | E-state fingerprints                        | 79                    |
|                 | Daylight-type fingerprints                 | 1024                  |
|                 | ECFP2 fingerprints                          | 1024                  |
|                 | ECFP4 fingerprints                          | 1024                  |
|                 | ECFP6 fingerprints                          | 1024                  |
algorithms, along with their parameters and the recommended defaults, are listed in Table 3. The detailed description of each algorithm and how to choose proper algorithms for different datasets are described in the “Manual” section of the website. After the calculation, a table is created to display the classifier, parameter, best parameter, number of positive samples, number of negative samples and score_means (the mean score of cross-validation for each parameter combination).

Model building
In the previous sub-module, a proper learning algorithm and the corresponding parameter set have been obtained. In this sub-module, a predictive classification model based on the selected method and parameters can be established. Here the same training dataset as that in the model selection sub-module should be chosen (or uploaded manually). The result table generated from this sub-module will give detailed information about the model, including classifier, parameter, number of positive samples, number of negative samples, AUC score, accuracy, MCC (Matthews correlation coefficient), F1 score, sensitivity, specificity and ROC curve. The built model will be stored in my model and can be employed to predict new samples next time.

Prediction
This sub-module is used to predict the samples from the test set or new samples from virtual chemical libraries, which is the ultimate goal of building an SAR model. In the submitting page, users can check the detailed information about the model including the classifier, parameter, accuracy, cross-validation fold, features in X (i.e., selected features used in training set), X_train, y_train. When the file of the test set matrix is uploaded, the model will calculate the predict_prob and predict_label for submitted samples. In the result page, an interactive table containing prediction results will be displayed and can also be downloaded.

Model interpretation and application domain
It is very necessary to have a reasonable interpretation of machine learning models and to define its application domain [66, 73, 74]. Here, we developed two related sub-modules to help researchers to interpret their models. The feature importance module enables researchers to interpret models in terms of feature importance. The forests of trees are used to evaluate the importance of features. By using this module, researchers can obtain a figure displaying the feature importances of the forest, along with their inter-trees variability. Another module is tree visualization which enables one to observe how the features classify the samples step by step in the decision tree model. By using this, the model will be displayed as a clear tree along with class names and explicit variables.

Moreover, we define an S index in the prediction module to help users to estimate which ones are considered reliable. It only works for chemical datasets. The S indices represent the mean similarity between each molecule from external samples and all molecules from training set using Tanimoto similarity metrics based on MACCS fingerprints. The higher the S index for a new molecule, the closer the molecule is to the main body of the training set, and thus we could conclude that a more reliable prediction for this molecule should be obtained by our constructed predictive model.

| Algorithms     | Parameters                                      | Recommended parameters                                      |
|----------------|-------------------------------------------------|-----------------------------------------------------------|
| RandomForest   | _n_estimators_: The number of trees in the forest; _max_features_: The number of features to consider when looking for the best split; (start_feature, end_feature and step make up the attempts of max_features) _cv_: cross-validation fold | _n_estimators_: 500; _max_features_: sqrt(N); N stands for number of features; _cv_: 5 |
| SVM            | _kernel_type_: rbf, sigmoid, poly, linear; _C_: penalty parameter C of the error term; _gamma_: kernel coefficient for ‘rbf’, ‘poly’ and ‘sigmoid’; _degree_: degree of the polynomial kernel function; _cv_: cross validation fold | _C_: 2^5, 2^15, 2^2 (format: start, end, step) _gamma_: 2^(-15), 2^3, 2^2; _degree_: 1, 7, 2 _cv_: 5 |
| Naive Bayes    | _Bayes_classifier_type_; _cv_: cross validation fold | BernoulliNB for binary-valued variable; GaussianNB for continuous variable; _cv_: 5 |
| K Neighbors    | _n_neighbors_: number of neighbors to use; _cv_: cross validation fold | _n_neighbors_: 1–10; _cv_: 5 |
| DecisionTree   | _Algorithm_: algorithm used to compute the nearest neighbors {‘ball_tree’, ‘kd_tree’, ‘brute’}; _cv_: cross validation fold | Automatic decision; _cv_: 5 |
Report system
One of the striking features of ChemSAR is that it provides a complete report generation system. It retrieves the results of each calculation step and re-arrange them into an organized HTML page and a PDF file for users. After finishing the whole modelling pipeline, the user can go to the “My Report” module to obtain the report. At the index page of this module, all job IDs that the user has created will be listed there. A “Get a PDF” button allows the user to generate a PDF file for off-line usage. A “Query” button is available to query the information about models created in other jobs. This is very helpful when a user attempts to construct multiple models using different machine learning methods and computing parameters, or when the user wants to build more than one model by the same client at the same time.

Useful tools
In addition to main modules mentioned above, ChemSAR offers three useful and convenient auxiliary tools. The first tool, statistical analysis, can be used to analyze the model performance. This tool is separate from the prediction module because the test set may have no real response values. The “attach to current job or not” option allows the user to predict different test sets and get a complete report each time. After the calculation, commonly used statistical indicators related to classification are displayed, including number of positive samples, number of negative samples, AUC score, accuracy, MCC, F1 score, sensitivity, specificity and ROC curve. The second tool, random training set split, can be used to split training set and test set by picking a subset of molecules randomly. The third tool, diverse training set split, can be used to split training set and test set by picking a subset of diverse molecules [75]. First, the similarity of ECFP4 fingerprints based on Dice similarity metric [31] is employed to calculate distances between molecular objects and then the MinMax algorithm is applied to select a subset of diverse molecules based on the aforementioned distances. This is usually a good strategy to avoid the unsuitable training/test data split problem.

Results and discussion
The most important strategy of pharmaceutical industry to overcome its productivity crisis in drug discovery is to focus on the molecular properties of absorption, distribution, metabolism and excretion (ADME). Nowadays, machine learning based approaches have been becoming a very popular choice to predict ADME properties of drug molecules. Here, in order to demonstrate the practicability and reliability of ChemSAR, we studied the Caco-2 Cell permeability using dataset from our previous publication [12]. All the compounds were divided into two classes according to the Caco-2 permeability cutoff value [12]. Then, we obtain a dataset of 1561 molecules containing 528 positive samples and 1033 negative samples. A detailed workflow of building the permeability models is shown in Fig. 3.

Firstly, through the structure preprocessing step (Standardization of molecules) using the default parameters, 1561 molecules were left. The random training set split tool (set test size for the data: 0.2) was then used to split the training set and test set. After this, a training set of 1249 samples (423 positive samples and 826 negative samples) and a test set of 312 samples (105 positive samples and 207 negative samples) were obtained.

In the feature calculation step, 203 descriptor were calculated including 30 constitution descriptors, 44 connectivity indices, 7 kappa indices, 32 Moran auto-correction descriptors, 5 molecular properties, 25 charge descriptors and 60 MOE-Type descriptors. Four filtering steps in data preprocessing were then performed: (1) missing values were imputed with the default parameters, and (2) descriptors with zero variance or near zero variance were removed with a cut-off value of 0.05, and (3) one of two highly correlated descriptors were randomly removed with a cut-off value of 0.95, and (4) perform the tree-based feature selection (n_estimators: 500, max_features: auto, threshold: mean). After these steps, 43 features were selected to build the model.

To test every module of ChemSAR and to build a model with high prediction performance, we employed five methods (RF, SVM, k-NN, NB, DT) to build classification models. In model selection, the parameters for each learning method were optimized using grid search strategy (The best parameters set for RF: {cv: 5, ‘max_features’: 9, ‘n_estimators’: 500}; SVM: {‘kernel’: ‘rbf’, ‘C’: 2, ‘gamma’: 0.125}; k-NN: {n_neighbors: 5}; NB: {‘classifier’: ‘GaussianNB’}). Then, a robust model was established with a 5-fold cross validation again. Each of the modelling processes will be repeated 10 times and then the statistical results will be reported as “mean ± variance”. Additionally, to test the fingerprints module and to make a further comparison, we also calculated five kinds of fingerprints and then built corresponding models. The model performance was displayed in Additional file 2: Fig. S1 and Additional file 2: Table S1. From the results, we can find that RF using 2D descriptors performs best: {Accuracy: 87.3±0.3, Sensitivity: 80±0.6, Specificity: 91.1 ± 0.3, MCC: 71.5 ± 0.6, AUC: 92.9 ± 0.3} for training set and {Accuracy: 85.3, Sensitivity: 77.1, Specificity: 89.4, MCC: 66.8, AUC: 89.9} for test set. Consequently, the RF method is more suitable for building a classification model for this dataset and the 2D descriptors can characterize molecules of this dataset more adequately. Clearly, by using the modelling module in ChemSAR, one could conveniently construct different
Fig. 3 The workflow of building Caco-2 Cell permeability models
Table 4 The current tools that can be used for SAR modelling

| Tool name          | Type   | Structure preprocessing | Data preprocessing | Molecular representation | Feature selection | Model selection | Algorithm type | Fees/register   | Coupling |
|--------------------|--------|-------------------------|--------------------|--------------------------|------------------|----------------|----------------|----------------|----------|
| ChemSAR            | Online | ✓                       | ✓                  | ✓                        | ✓                | ✓              | I              | Free           | Low      |
| OCHEM              | Database | ✓                    | ✓                  | ✓                        | ✓                | ✓              | I, II          | Restricted     | High     |
| Chembench          | Online | ✓                       | ✓                  | ✓                        | ✓                | ✓              | I, II          | Need register  | High     |
| Vcclab             | Online | ✓                       | ✓                  | ✓                        | ✓                | ✓              | I              | Free           | Low      |
| OpenTox            | Online | ✓                       | ✓                  | ✓                        | ✓                | ✓              | I              | Free           | High     |
| QSAR4U             | Online | ✓                       | ✓                  | ✓                        | ✓                | ✓              | I, II          | Free           | High     |
| Camb               | R package | ✓               | ✓                  | ✓                        | ✓                | ✓              | I, II          | Free           | Low      |
| AZOrange           | Software | ✓                  | ✓                  | ✓                        | ✓                | ✓              | Disabled for invalid dependences | Free | Low |
| RRegrts            | R package | ✓              | ✓                  | ✓                        | ✓                | ✓              | II             | Free           | Low      |
| eTOXlab            | Software | ✓                      | ✓                  | ✓                        | ✓                | ✓              | Models for production environments | Free | High |
| QSARINS            | Software | ✓                      | ✓                  | ✓                        | ✓                | ✓              | II             | Restricted     | High     |
| OECD QSAR Toolbox  | Software | ✓                      | ✓                  | ✓                        | ✓                | ✓              | Models for data gap filling | Restricted | High |
| MOLGEN QSPR       | Software | ✓                      | ✓                  | ✓                        | ✓                | ✓              | II             | Free           | High     |
| BuildQSAR          | Software | ✓                      | ✓                  | ✓                        | ✓                | ✓              | II             | Free           | High     |
| McQSAR             | Software | ✓                      | ✓                  | ✓                        | ✓                | ✓              | II             | Free           | High     |
| StarDrop           | Software | ✓                      | ✓                  | ✓                        | ✓                | ✓              | I, II          | Commercial     | –        |
| MOE                | Software | ✓                      | ✓                  | ✓                        | ✓                | ✓              | II             | Commercial     | Low      |
| DS                 | Software | ✓                      | ✓                  | ✓                        | ✓                | ✓              | II             | Commercial     | Low      |

The I and II represent the classification algorithms and regression algorithms; The “restricted” means that some modules of the tool are limited to the public or need the permits of the developers; The “low” coupling means that the main modules of the tool can be called in the modelling pipelining and can also be used as an independent tool, while the “high” coupling means that they must work together to build a model.
algorithm models for one dataset and then makes a comprehensive comparison and further analysis to identify the best prediction model for the current problem. To further evaluate the prediction ability of our models, we compared our prediction results with the published models in recent papers. The latest report was in 2013 [76], the authors built a model using DT method with 1289 compounds which could accurately predict 78.4/76.1/79.1% of H/M/L compounds on the training set and 78.6/71.1/77.6% on the test set. In 2011 [77], Pham et al. built a model using linear discriminant analysis (LDA) method with 674 molecules which reported results: $\text{MCC} = 0.62$, $\text{Accuracy} = 81.56\%$ (training set), $\text{Accuracy} = 83.94\%$ (test set). Compared with the two models above, our model has an almost comparative or better performance. Obviously, ChemSAR has the capacity to obtain the reliable and robust classification model for the evaluation of Caco-2 Cell permeability.

Comparison with other related tool sources
For the purpose of further comparison, we have studied related publications as much as we can, and searched on Google to collect related tools that possess SAR modeling functionality. Then, a comparison based on application scenarios and functionality was performed, which was summarized in Table 4. In this table, we compared and then marked several aspects of each tool, including “type”, “structure preprocessing”, “data preprocessing”, “molecular representation”, “feature selection”, “model selection”, “algorithm type”, “charge or free” and “coupling”. The results suggest that ChemSAR is strongly recommended for multiple advantages of it as shown in the table. Note that we cannot absolutely guarantee the accuracy of the description for each tool because we get all the available information mainly from the corresponding publications or its documentations but some tools are not accessible or commercial. Also, the features of each tool evaluated here are from this tool’s main framework, not the plugins provided by its user community.

Conclusion
In this study, we developed the ChemSAR platform as an online pipelining of building SAR classification models. It is freely accessible to the public and is platform-independent so users can access this platform via almost all different types of operation systems (Linux, Microsoft windows, Mac OS, Android) and clients (PC clients, mobile clients). The main advantages of the proposed platform are summarized as follows: (1) ChemSAR implements a complete online model-building process, which enables biomedical investigators to construct predictive models easily without suffering from tedious programming and deployment work. (2) ChemSAR provides a comprehensive modelling pipelining by integrating six model generation steps into a unified workflow. (3) The modular design of the framework enables six sub-modules to run independently to accomplish specific functionalities. (4) The job submission strategy allows users to query the calculation results at spare time. This provides an essential basis for the report system to generate a clear modeling report. (5) The modular design of the framework allows researchers to deal with not only the analysis of small molecules but also modelling problems in the biomedical field. For example, building a classification model based on the biochemical indicators of patients helps to study the disease classification or stage. In addition, we conducted a case study to illustrate the use of this platform in practice, and several models were obtained to evaluate the Caco-2 Cell permeability at the same time.

A major goal of the cheminformatics development is to make its techniques to be applied into the study of practical problems. The trend for future development of SAR models is towards making models publicly accessible on-line, interactive, and usable [78]. ChemSAR, to some extent, has made a step in this direction. It is expected that ChemSAR can be applied to a wide variety of studies when there exists a significant demand of using SAR models. In the future, we will continue to implement more classification algorithms and add the options for a more flexible parameter control. Also, we will add the regression algorithms if needed.

Additional files

**Additional file 1:** The code snippets to show the implementation of calculating molecular fingerprints.

**Additional file 2:** Table S1. Classification results of different models in the evaluation of Caco-2 Cell permeability. Fig. S1. The ROC curves for different models in the evaluation of Caco-2 Cell permeability.

Authors’ contributions
JD and DSC designed and implemented the platform. JD, HM and DSC wrote and revised the manuscript. ZJY, MFZ and NNW helped in preparing figures and tables, testing and validating the results. BL, AFC, APL and WBZ helped in giving suggestions to improve the platform. All authors read and approved the final manuscript.

Author details
1 Xiangya School of Pharmaceutical Sciences, Central South University, No. 172, Tongzipo Road, Yuelu District, Changsha, People’s Republic of China. 2 The Third Xiangya Hospital, Central South University, Changsha, People’s Republic of China. 3 Institute for Advancing Translational Medicine in Bone and Joint Diseases, School of Chinese Medicine, Hong Kong Baptist University, Kowloon Tong, Hong Kong SAR, People’s Republic of China. 4 Department of Biostatistics, School of Public Health, University of Texas Health Science Center, Houston, TX 77030, USA.

Competing interests
The authors declare that they have no competing interests.
Availability and requirements
Project name: ChemSAR
Project home page: http://chemsarcbd.com
Operating system(s): Platform independent.
Programming language: Python, JavaScript, HTML5, CSS.
Other requirements: Modern internet browser supporting HTML5 and JavaScript. The recommended browsers: Safari, Firefox, Chrome, IE (Ver. >8).
License: http://creativecommons.org/licenses/by-nc-sa/4.0/.
Any restrictions to use by non-academics: License needed.

Funding
This work is financially supported by the National Key Basic Research Program (2015CB910700), the National Natural Science Foundation of China (Grants No. 81402853), the Hunan Provincial Innovation Foundation for Postgraduate (CX2016B058), the Project of Innovation-driven Plan in Central South University, and the Postdoctoral Science Foundation of Central South University, the Chinese Postdoctoral Science Foundation (2014T70794, 2014M562142). The studies meet with the approval of the university’s review board.

Publisher’s Note
Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 3 December 2016 Accepted: 24 April 2017
Published online: 04 May 2017

References
1. Hopkins AL (2009) Drug discovery: predicting promiscuity. Nature 462(7270):167–168
2. Murphy RF (2011) An active role for machine learning in drug development. Nat Chem Biol 7(6):327–330
3. Gertrudes JC, Maltarollo VG, Silva RA, Oliveira PR, Honorio KM, Da Silva ABF (2012) Machine learning techniques and drug design. Curr Med Chem 19(23):4289–4297
4. Ding H, Takigawa I, Mamitsuka H, Zhu S (2014) Similarity-based machine learning methods for predicting drug–target interactions: a brief review. Brief Bioinform 15(5):734–747
5. Cortes-Ciriano I, van Westen GPJ, Lenselink EB, Murrell DS, Bender A, Maltavirgin M, et al. (2014) PyChemometric modeling in a Bayesian framework. J Cheminform 6(1):35
6. Cheng J, Trege AN, Baldi P (2008) Machine learning methods for predicting protein structure properties. IEEE Rev Biomed Eng 1:41–49
7. Agarwal S, Dugar D, Sengupta S (2010) Ranking chemical structures for drug discovery: a new machine learning approach. J Chem Inf Model 50(5):716–731
8. Speck-Planche A, Kleanandria W (2012) QSAR and molecular docking for the discovery of potent monoamine oxidase B inhibitors: computer-aided generation of new rasagiline bioisosteres. Curr Top Med Chem 12(18):1734–1747
9. van der Linde LA, Baskin I (2012) Machine learning methods for property prediction in chemoinformatics: quo vadis? J Chem Inf Model 52(6):1413–1437
10. Roncaglioni A, Toropov AA, Toropova AP, Benfenati E (2013) In silico methods to predict drug toxicity. Curr Opin Pharmacol 13(5):802–806
11. Wang J, Dong J, Deng Y, Zhu M, Yun X, Yao N, Xiao N, Liang Y (2015) In silico evaluation of logD7.4 and comparison with other prediction methods. J Chemom 29(7):389–398
12. Wang J, Cao D, Zeng B, Zeng W, Xu Q, Liang Y, Lu A, Chen AF (2015) In silico toxicity prediction of chemicals from EPA toxicity database by kernel fusion-based support vector machines. Chemom Intell Lab Syst 146:494–502
13. Wang J, Cao D, Zhu M, Yun X, Yao N, Xiao N, Liang Y (2015) In silico evaluation of logD7.4 and comparison with other prediction methods. J Chemom 29(7):389–398
14. Gaulton A, Bellis LJ, Bento AP, Chambers J, Davies M, Hersey A, Light Y, McGlinnery S, Michalovich D, Al-Lazikani B, Overington JP (2012) CHEMBL, a large-scale bioactivity database for drug discovery. Nucleic Acids Res 40(D1):D1100–D1107
15. Wang J, Xiao J, Suzeq TQ, Zhang J, Wang J, Bryant SH (2009) Pubchem: a public information system for analyzing bioactivities of small molecules. Nucleic Acids Res 37(W633–W633
16. Knox C, Law V, Jewison T, Liu P, J L, S, Frolik A, P A, Bano K, Mac K, Neveu V, Djoumbe P, Essner R, Guo AC, Wishart DS (2011) DrugBank 3.0: a comprehensive resource for ’omics’ research on drugs. Nucleic Acids Res 39(1):D135–D141
17. Gaulton A, Swindell R, Yamanishi Y (2011) Predicting drug side-effect profiles: a chemical fragment-based approach. BMC Bioinf 12(1):169
18. Perez-Nuerno M, Souchet M, Karaboga AS, Ritchie DW (2015) GESEE: predicting drug side effects from drug–target relationships. J Chem Inf Model 55(9):1804–1823
19. Yamanishi Y, Pauwels E, Kotera M (2012) Drug side-effect prediction based on the integration of chemical and biological spaces. J Chem Inf Model 52(12):3284–3292
20. Zhang L, Zhang YM, Zhao P, Huang S (2009) Predicting drug–drug interactions: an FDA perspective. AAPS J 11(2):300–307
21. Cao D, Xiao N, Li Y, Zeng W, Liang Y, Lu A, Xu Q, Chen AF (2015) Integrating multiple evidence sources to predict adverse drug reactions based on a systems pharmacology model. CPT Pharmacometrics Syst Pharmacol 4(9):498–506
22. Pauwels E, Stoven V, Yamanishi Y (2011) Predicting drug side-effect profiles: a chemical fragment-based approach. BMC Bioinf 12(1):169
23. Perez-Nuerno M, Souchet M, Karaboga AS, Ritchie DW (2015) GESEE: predicting drug side effects from drug–target relationships. J Chem Inf Model 55(9):1804–1823
24. Yamanishi Y, Pauwels E, Kotera M (2012) Drug side-effect prediction based on the integration of chemical and biological spaces. J Chem Inf Model 52(12):3284–3292
25. Zhang L, Zhang YM, Zhao P, Huang S (2009) Predicting drug–drug interactions: an FDA perspective. AAPS J 11(2):300–307
26. Geppert H, Vogt M, Bajorath J (2010) Current trends in ligand-based virtual screening: molecular representations, data mining methods, new application areas, and performance evaluation. J Chem Inf Model 50(2):205–216
27. Wang J, Cao D, Zhu M, Yun X, Yao N, Xiao N, Liang Y, Lu A, Cao D (2016) TargetNet: a web service for predicting potential drug–target interaction profiling via multi-target SAR models. J Comput Aided Mol Des 30(5):413–424
28. Cao D, Zhao Q, Liu S, Zhang L, Xu Q, He M, Liang Y (2013) Large-scale prediction of drug–target interactions using protein sequences and drug topological structures. Anal Chem Acta 752:1–10
29. Yao Z, Dong J, Che Y, Zhu M, Wen W, Wang N, Wang S, Lu A, Cao D (2016) Targhitest: a web service for predicting potential drug–target interaction profiling via multi-target SAR models. J Comput Aided Mol Des 30(5):413–424
30. Cao D, Zhou Q, Liu S, Zhang L, Xu Q, He M, Liang Y (2013) Large-scale prediction of human kinase-inhibitor interactions using protein sequences and molecular topological structures. Anal Chem Acta 792:10–18
31. Kinnings SL, Liu N, Tonge PJ, Jackson RM, Xie L, Bourne PE (2011) A machine learning-based method to improve docking scoring functions and its application to drug repurposing. J Chem Inf Model 51(2):408–419
32. Napolitano F, Zhao Y, Moreira VM, Tagliaferri R, Kere J, D’Amato M, Greco D (2013) Drug repositioning: a machine-learning approach through data integration. J Cheminf 5(1):30
33. RDKit: Open-source cheminformatics. http://www.rdkit.org. Accessed 28 Nov 2016
34. Chang C, Lin C (2011) LIBSVM: a library for support vector machines. ACM Trans Intell Syst Technol 2(3):27
35. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Bengfort S, et al. (2011) Scikit-learn: machine learning in python. J Mach Learn Res 12:2825–2830
36. Ginestet C (2011) ggplot2: elegant graphics for data analysis. J R Stat Soc A Stat 174(1):245
37. Steinbeck C, Han YQ, Kuhn S, Horlerach O, Luttmann E, Willighagen E (2003) The chemistry development kit (CDK): an open-source Java library for chemo- and bioinformatics. J Chem Inf Comput Sci 43(2):493–500
38. Cao D, Xu Q, Hu Q, Liang Y (2013) ChemoPy: freely available python package for computational biology and chemoinformatics. Bioinformatics 29(8):1092–1094
39. O’Boyle NM, Banck M, James CA, Morley C, Vandermeersch T, Hutchinson GR (2011) Open babel: an open chemical toolbox. J Cheminform 3(1):33
38. Yap CW (2011) PaDEL-descriptor: an open source software to calculate molecular descriptors and fingerprints. J Comput Chem 32(7):1466–1474
39. O’Boyle NM, Hutchison GR (2008) Cifinfony—combining Open Source cheminformatics toolkits behind a common interface. Chem Eng J 21(1):24
40. Cao D, Liang Y, Yan J, Tan G, Xu Q, Liu S (2013) PyDPI: freely available python package for chemoinformatics, bioinformatics, and chemogenomics studies. J Chem Inf Model 53(11):3086–3096
41. Cao D, Xiao N, Xu Q, Chen AF (2015) Rcpi/R/bioconductor package to generate various descriptors of proteins, compounds and their interactions. Bioinformatics 31(2):279–281
42. Mevik B, Wehrens R (2007) The pls package: principal component and partial least squares regression in R. J Stat Softw 28(5):1–26
43. Kuhn M (2008) Building predictive models in R using the caret package. J Stat Softw 28(5):1–26
44. Liaw A, Wiener M (2002) Classification and regression by randomForest. R News 2(3):18–22
45. Zeileis A, Hornik K, Smola A, Karatzoglou A (2004) Kernlab-an S4 package for kernel methods in R. J Stat Softw 11(9):1–20
46. Tsiliki G, Munteanu CR, Seoane JA, Fernandez-Lozano C, Sarimveis H, Willighagen EL (2015) Rreg: an R package for computer-aided model selection with multiple regression models. J Cheminform 7(1):46
47. Hunter JD (2007) Matplotlib: a 2D graphics environment. Comput Sci Eng 9(3):90–95
48. Seaborn: statistical data visualization. https://web.stanford.edu/~mwaskom/software/seaborn/index.html. Accessed 28 Nov 2016
49. 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
50. Dong J, Yao Z, Wen M, Zhu M, Wang N, Miao H, Lu A, Zeng W, Cao D (2016) BioTriangle: a web-accessible platform for generating various molecular representations for chemicals, DNAs/RNAs and their interactions. J Cheminform 8(1):1–13
51. Tetko IV, Gasteiger J, Todeschini R, Mauri A, Livingstone D, Erli P, Palyulin VA, Radchenko EV, Zefirov NS, Makarenko AS, Tanchuk VY, Prokopenko VV (2005) Virtual computational chemistry laboratory—design and description. J Cheminform 7(1):1–29
52. QSAR4U. http://qsar4u.com/pages/pred_online.php. Accessed 28 Nov 2016
53. Hai PT, Gonzalez-Alvarez I, Bermejo M, Garrigues T, Angel Cabrera-Perez M (2011) In silico prediction of Caco-2 permeability by a classification QSAR approach. Mol Inform 30(4):376–385
54. Walker T, Grulke CM, Pozefsky D, Tropsa A (2010) Chembench: a cheminformatics workbench. Bioinformatics 26(23):3000–3001
55. Capuzzo SJ, Kim IS, Lam W, Thornton TE, Muratov EN, Pozefsky D, Tropsa A (2017) Chembench: a publicly accessible, integrated cheminformatics portal. J Chem Inf Model 57(2):105–108
56. Carpio P, Lopez O, Sanz F, Pastor M (2015) eTOXlab: an open source modeling framework for implementing predictive models in production environments. J Cheminform 7(1):1–9
57. Stirling JC, Carlsson LA, Almeida P, Boyer S (2011) AZOrange—high performance open source machine learning for QSAR modeling in a graphical programming environment. J Cheminform 3(1):28
58. Gramatica P, Chirico N, Papa E, Cassani S, Kovarich S (2013) QSARINS: a new software for the development, analysis, and validation of QSAR MLR models. J Comput Chem 34(24):2121–2132
59. OECD QSAR Toolbox. http://www.oecd.org/chemicalsafety/risk-assessment/theoecdsqartoolbox.htm. Accessed 28 Nov 2016
60. de Oliveira DB, Gaudio AC (2001) BuildQsar: a new computer program for QSAR analysis. Quant Struct Act Relatsh 19(6):599–601
61. Molecular Operating Environment. http://www.chemcomp.com/MOE-Molecular_Operating_Environment.htm. Accessed 28 Nov 2016
62. Discovery Studio. http://accelrys.com/products/collaborative-science/biovia-discovery-studio/. Accessed 28 Nov 2016
63. Tropsa A (2010) Best practices for QSAR model development, validation, and exploitation. Mol Inform 29(6–7):476–488
64. Guyon I, Elisseeff A (2003) An introduction to variable and feature selection. J Mach Learn Res 3(3):1157–1182
65. Breiman L (2001) Random forests. Mach Learn 45(1):5–32
66. Tropsha A (2010) Best practices for QSAR model development, validation, and exploitation. Mol Inform 29(6–7):476–488
67. Quinlan JR (1999) Simplifying decision trees. Int J Hum Comput Stud 51(2):497–510
68. Weaver S, Gleeson NP (2008) The importance of the domain of applicability for QSAR analysis. Quant Struct Act Relatsh 19(6):599–601
69. Gramatica P, Gramatica P, Gombar VK (2003) The importance of being earnest: validation is the absolute essential for successful application and interpretation of QSAR models. QSAR Comb Sci 22(1):69–77
70. Walker T, Grulke CM, Pozefsky D, Tropsa A (2010) Chembench: a cheminformatics workbench. Bioinformatics 26(23):3000–3001
71. Capuzzo SJ, Kim IS, Lam W, Thornton TE, Muratov EN, Pozefsky D, Tropsa A (2017) Chembench: a publicly accessible, integrated cheminformatics portal. J Chem Inf Model 57(2):105–108
72. Carpio P, Lopez O, Sanz F, Pastor M (2015) eTOXlab: an open source modeling framework for implementing predictive models in production environments. J Cheminform 7(1):1–9
73. Stirling JC, Carlsson LA, Almeida P, Boyer S (2011) AZOrange—high performance open source machine learning for QSAR modeling in a graphical programming environment. J Cheminform 3(1):28
74. Gramatica P, Chirico N, Papa E, Cassani S, Kovarich S (2013) QSARINS: a new software for the development, analysis, and validation of QSAR MLR models. J Comput Chem 34(24):2121–2132
75. OECD QSAR Toolbox. http://www.oecd.org/chemicalsafety/risk-assessment/theoecdsqartoolbox.htm. Accessed 28 Nov 2016
76. de Oliveira DB, Gaudio AC (2001) BuildQsar: a new computer program for QSAR analysis. Quant Struct Act Relatsh 19(6):599–601
77. Molecular Operating Environment. http://www.chemcomp.com/MOE-Molecular_Operating_Environment.htm. Accessed 28 Nov 2016
78. Discovery Studio. http://accelrys.com/products/collaborative-science/biovia-discovery-studio/. Accessed 28 Nov 2016
79. Tropsa A (2010) Best practices for QSAR model development, validation, and exploitation. Mol Inform 29(6–7):476–488
80. Guyon I, Elisseeff A (2003) An introduction to variable and feature selection. J Mach Learn Res 3(3):1157–1182
81. Breiman L (2001) Random forests. Mach Learn 45(1):5–32
82. Pontil M, Verri A (1998) Properties of support vector machines. Neural Comput 10(4):955–974
83. k-nearest neighbors algorithm. https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm. Accessed 28 Feb 2017
84. Naive Bayes classifier. https://en.wikipedia.org/wiki/Naive_Bayes_classifier. Accessed 28 Feb 2017
85. Quinlan JR (1999) Simplifying decision trees. Int J Hum Comput Stud 51(2):497–510
86. Tropsa A, Gramatica P, Gombar VK (2003) The importance of being earnest: validation is the absolute essential for successful application and interpretation of QSAR models. QSAR Comb Sci 22(1):69–77
87. Weaver S, Gleeson NP (2008) The importance of the domain of applicability for QSAR modeling. J Mol Graph Model 26(8):1315–1326
88. Ashton M, Barnard J, Casset F, Charlton M, Downs G, Gorse D, Holliday J, Lahana R, Willett P (2002) Identification of diverse database subsets using property-based and fragment-based molecular descriptions. Quant Struct Act Relatsh 21(6):598–604
89. Hai P, Gonzalez-Alvarez I, Bermejo M, Garrigues T, Huang L, Angel Cabrera-Perez M (2013) The use of rule-based and QSAR approaches in ADMET profiling: a case study on Caco-2 permeability. Mol Inform 32(5–6):459–479
90. Hai PT, Gonzalez-Alvarez I, Bermejo M, Mangas Sanjuan V, Centelles J, Garrigues TM, Angel Cabrera-Perez M (2011) In silico prediction of Caco-2 Cell permeability by a classification QSAR approach. Mol Inform 30(4):376–385
91. Tetko IV, Maran U, Tropsa A (2016) Public (Q) SAR services, integrated modeling environments, and model repositories on the web: state of the art and perspectives for future development. Mol Inform 36(3):1–14