J Comput Aided Mol Des (2011) 25:1121–1133
DOI 10.1007/s10822-011-9496-z

DemQSAR: predicting human volume of distribution and clearance of drugs

Ozgur Demir-Kavuk · Jörg Bentzien · Ingo Muegge · Ernst-Walter Knapp

Received: 7 July 2011 / Accepted: 13 November 2011 / Published online: 20 November 2011

Abstract *In silico* methods characterizing molecular compounds with respect to pharmacologically relevant properties can accelerate the identification of new drugs and reduce their development costs. Quantitative structure–activity/-property relationship (QSAR/QSPR) correlate structure and physicochemical properties of molecular compounds with a specific functional activity/property under study. Typically a large number of molecular features are generated for the compounds. In many cases the number of generated features exceeds the number of molecular compounds with known property values that are available for learning. Machine learning methods tend to overfit the training data in such situations, i.e. the method adjusts to very specific features of the training data, which are not characteristic for the considered property. This problem can be alleviated by diminishing the influence of unimportant, redundant or even misleading features. A better strategy is to eliminate such features completely. Ideally, a molecular property can be described by a small number of features that are chemically interpretable. The purpose of the present contribution is to provide a predictive modeling approach, which combines feature generation, feature selection, model building and control of overtraining into a single application called DemQSAR. DemQSAR is used to predict human volume of distribution (VD_{ss}) and human clearance (CL). To control overtraining, quadratic and linear regularization terms were employed. A recursive feature selection approach is used to reduce the number of descriptors. The prediction performance is as good as the best predictions reported in the recent literature. The example presented here demonstrates that DemQSAR can generate a model that uses very few features while maintaining high predictive power. A standalone DemQSAR Java application for model building of any user defined property as well as a web interface for the prediction of human VD_{ss} and CL is available on the webpage of DemPRED: http://agknapp.chemie.fu-berlin.de/dempred/.

Keywords Volume of distribution · Clearance · Prediction · DemQSAR · Feature selection · QSAR

Abbreviations

VD_{SS}	Volume of distribution at steady state
CL	Clearance
QSA/PR	Quantitative structure–activity/-property relationship
GMFE	Geometric mean fold-error
RFE	Recursive feature elimination
CDK	Chemistry development kit
PST	Performance size trade-off

Background

Discovering new drugs is a time consuming and resource intensive endeavor. A promising compound, which shows a

Electronic supplementary material The online version of this article (doi:10.1007/s10822-011-9496-z) contains supplementary material, which is available to authorized users.

O. Demir-Kavuk · E.-W. Knapp (✉)
Institute of Chemistry and Biochemistry, Freie Universität Berlin, Fabeckstrasse 36A, 14195 Berlin, Germany
e-mail: knapp@zedat.fu-berlin.de

O. Demir-Kavuk
e-mail: o.demir@fu-berlin.de

J. Bentzien · I. Muegge
Department of Medicinal Chemistry, Boehringer Ingelheim Pharmaceuticals, Inc., Ridgefield, CT 06877, USA
e-mail: joerg.bentzien@boehringer-ingelheim.com

I. Muegge
e-mail: ingo.mugge@boehringer-ingelheim.com

desired activity in vitro, may fail in vivo due to toxicity or unacceptable pharmacokinetic characteristics. In fact many compounds are filtered out during late stages of drug discovery as they show adverse effects or have other undesired properties. The pharmaceutical industry has a need to detect these compounds at early stages of drug discovery in order to save money, resources, and time [1, 2]. Therefore, in vitro screening methods have been developed over the past years to predict ADME-Tox (absorption, distribution, metabolism, excretion, toxicity) parameters of new compounds and prioritize drug candidates [3–7]. Nevertheless, even in vitro methods are still resource intensive and time consuming. To overcome these problems various *in silico* screening methods have been developed [8–13]. The advantages are significant as nowadays computers are cheap and predictions are easy and fast to compute. The working principle of *in silico* prediction is to correlate descriptors (features) of a large number of putative drug candidates with their activity or other relevant functional properties and uses these correlations in a predictive manner. These approaches are known as quantitative structure–activity/-property relationship (QSAR/QSPR) [14–17].

Many QSAR software products have been presented in the past [18–20]. However, most of the available software packages for QSAR analysis are not free of charge and thus not readily accessible to everyone, which hampers usage, validation, and comparison of performances of different approaches. Even if the developer of a QSAR model has the necessary licenses, the generated models cannot be made publicly available through web services as the end user may not have valid licenses. There are many open source machine learning packages available in various programming languages [21–25] and open source software packages that generate QSAR descriptors [26–28]. These packages can be combined to build QSAR models, though some basic programming skills are still required. Hence, there is a need for QSAR software that is easy to use, powerful, widely applicable, and free of charge.

To address this need we introduce here the DemQSAR software and demonstrate its usefulness for predicting human volume of distribution (VD_{ss}) and clearance of drugs (CL). DemQSAR combines feature generation, feature selection, model building, and control of overtraining into a single platform-independent Java application. DemQSAR makes use of the open source CDK library [29] to calculate various QSAR descriptors and topological fingerprints, which can be complemented by additional descriptors. The feature generation step is followed by an optional feature selection step. The core of the model building process is a linear scoring function, which allows combining various loss functions and kernels. The generated models are saved into a single file and thus can be shared easily.

The performance of DemQSAR has been tested by building models for the prediction of human VD_{ss} and CL. Volume of distribution is a measure of how a drug is distributed between plasma and tissues. Clearance is a measure of the rate at which a drug is removed from the body. Both values together determine the half-life of a drug and thus the appropriate dose and frequency of drug application. A drug should be administered such that the free plasma concentration is large enough to obtain an effect throughout the dosing interval, while lessening the maximal concentration over time by clearance and thereby reducing the potential for side effects. The generated DemQSAR models are able to predict human VD_{ss} and CL of new compounds within a geometric mean fold error (GMFE) of 2.0 and 2.4 respectively. Both models are accessible via an easy to use web interface, which does not require any additional knowledge of QSAR analysis. Furthermore, a standalone Java application for model building of any user defined target property is available for download free of charge.

Methods

Datasets of compounds

To build models for the prediction of human VD_{ss} and CL, training datasets of compounds with experimentally measured VD_{ss} and CL values are needed. There are two publicly available datasets, which were compiled by large scale literature searches. Obach and coworkers [30] published a dataset of 670 compounds together with their VD_{ss}, CL, mean residence time and terminal half-life. These molecular properties were obtained or derived from original references. Only studies based on intravenous application were considered. Earlier Lombardo and coworkers [31] published a dataset of 384 compounds, which also contains the human VD_{ss}. A comparison of both datasets (from Lombardo and Obach) by compound name, CAS number, and structure revealed that only nine compounds of the Lombardo dataset were not present in the Obach dataset. Hence, for consistency reasons, only compounds from the Obach dataset were used throughout this study. To ensure that all descriptors could be calculated for all compounds we removed compounds containing phosphorous, boron, and metal atoms, all macrocycles, and some fragment-like compounds, e.g. Metformin, from the dataset. This resulted in a final dataset of 584 compounds referred to as *DataWhole*.

To test the performance of our models on an independent test set, *DataWhole* was divided into a training and test set. All 338 compounds that were already present in the older dataset of Lombardo and coworkers [31] were used to

Table 1 Overview of used datasets

Name	Size	Description
DataWhole	584	[30]
DataTrain	338	[31]
DataTest	246	*DataWhole* without *DataTrain*
Data10Fold	58/59	*DataWhole* randomly splitted into 10 parts.
DataExternal	29	[11]

train the prediction models. All remaining compounds that were added by Obach and coworkers [30] were used as an independent test set. This resulted in a training set of 338 compounds and a test set of 246 compounds from now on referred to as *DataTrain* and *DataTest* respectively. This is a historically based split of the data, albeit only 60% of the data are used for learning. We believe it may provide a more realistic testing scheme than a random split. To decrease the influence of the division into train and test set and to monitor how the result behave in case the training set is increased, we additionally applied a 10-fold cross-validation on *DataWhole* to measure the performance of our models. Finally the performance has been tested on an external independent test set of 29 compounds provided by Berellini and coworkers [11], referred to as *DataExternal.* Table 1 gives an overview of the data sets of compounds used in this study. All data sets used in this study are available as supplementary material.

Molecular descriptors

We used several software packages to compute molecular descriptors and topological fingerprints, which characterize the structures and physicochemical features of the molecules. Table 2 provides an overview of the applied software packages and the number of calculated descriptors. A total of 404 molecular descriptors were computed for each compound using various software packages (ACD/Labs, ClogP, Volsurf, MolConn-Z, Scitegic, MOE, Pipeline Pilot). A complete list of computed molecular descriptors can be found in the supplementary material. In addition four different kinds of fingerprints were generated: Daylight (1,024 features), Isis (166 features), ECFP4 (1,024 features), FCFP4 (1,024 features). Concatenation of these features resulted in a total of 3,642 features, referred to as commercial descriptors. Most of the software packages which were used to calculate these descriptors are not free of charge. To make the final model publicly available and also give the user the possibility to build new predictor models we additionally used the open-source Chemistry Development Kit (CDK) to calculate descriptors [29]. The CDK is a free open-source Java library for structural chemo- and bioinformatics and is developed by more than 30 contributors world-wide. The CDK library is tightly integrated into DemQSAR such that no additional setup is needed by the user. CDK descriptors and fingerprints are automatically calculated during model building if desired.

Table 2 Overview of computed and used molecular features

Commercial	#	Description	Reference
ACD/labs	4	Fraction anionic/cationic and pKa values	[45]
ClogP	1	Calculated logP	[46]
Volsurf	104	Molecular descriptors from 3D interaction fields	[47]
MolConn-Z	55	Molecular connectivity, shape, and information indices	[48]
Scitegic	56	Various molecular descriptors	[49]
MOE 2D	184	Various molecular descriptors	[50]
Daylight	1,024	Path based structure fingerprints	[49]
Isis Keys	166	Fragments to describe overall molecular structure	[49]
ECFP4	1,024	Atom type based, extended connectivity fingerprint	[49]
FCFP4	1,024	Functional class based, extended connectivity fingerprint	[49]
Sum	3,642		
Public		Description	Reference
QSAR	219	Various molecular and physico chemical properties	[29, 51]
Standard	1,024	Daylight style hashed fingerprints	
Extended	1,024	Extended by ring structures	
Graph Only	1,024	Does not take bond orders into account	
Estate	79	Fingerprints using the E-state fragments	
MACCS	166	166 predefined MDL keys	
Substructure	307	Predefined CDK substructure occurrence	
Sum	3,843		

A detailed description can be found in supplementary material

The CDK library provides 219 molecular descriptors and six different types of fingerprints. Concatenation of molecular descriptors and fingerprints yields a total of 3,843 descriptors referred to as CDK descriptors. A detailed description of all CDK descriptors and fingerprints can be found in the supplementary material.

Scoring function

Describing molecules by real valued descriptor vectors $\vec{x}_i \in \mathbb{R}^d$ allows their representation as points in a d-dimensional feature space. In a general two-class classification approach, a hypersurface is defined that separates the data points belonging to different classes. In the simplest case, the hypersurface is planar separating the multidimensional feature space into two half-spaces such that all positive data points (property value $+1$) are located on one side of the hyperplane whereas all negative data points (property value -1) are located on the other side. New data points are then classified according to the half-space they belong to.

In a regression task a hyperplane is constructed where the distances of the data points to the hyperplane are proportional to their property values. The property value of a new molecule can then be predicted by computing its proportional distance to the hyperplane. To determine the parameter vector $\vec{w}$ normal to the hyperplane and the offset b of the hyperplane from the origin we set up an objective function L that is minimal for the optimal hyperplane:

$$L(\vec{w}, b) = \underbrace{\frac{1-(\lambda_1+\lambda_2)}{N}\sum_{i=1}^{N}\{\mu_i\, g(f(\vec{x}_i;\vec{w},b), m_i)\}}_{\text{model terms}} + \underbrace{\lambda_2||\vec{w}||_2^2 + \lambda_1||\vec{w}||_1^1}_{\text{regularization terms}}, \quad (1)$$

where m_i is the property value and $g(f, m_i)$ is a loss function that determines stiffness of the correlation between the property values m_i and the scoring function $f(\vec{x}_i)$. The additional parameters μ_i in the objective function (1) can be used to weigh more reliable data points higher than others. Once a hyperplane is defined new, compounds can be predicted using a linear scoring function:

$$f(\vec{x};\vec{w},b) = \vec{w}^t \cdot \vec{x} + b. \quad (2)$$

The objective function L, Eq. 1, consists of two parts that compete with each other. The first part involves the so called ‘model terms’, which optimize the prediction performance on the training set (recall). Whenever the considered model returns a poor prediction on the training set the loss-function $g(f_i, m_i)$ invokes a penalty that depends on the error margin. Hence, during learning the hyperplane parameters will be chosen such that predictions on the training set are as close as possible to their experimentally measured property values. Figure 1 gives an overview of the loss functions available in DemQSAR. For regression analysis, e.g. squared error loss functions are used. They punish deviations from the property value in both directions evenly. However, due to the quadratic form of this loss function, outliers have a strong influence on the resulting model. Hence, in cases where outliers are expected, the Lorentzian loss function may be a better choice. Both squared error and Lorentzian loss function can also be used for classification tasks. However, for classification, better results may be achieved with loss functions, which only punish deviations in one direction. As a consequence positive data points classified as strongly positive and negative data points classified as strongly negative are not punished. For that purpose loss functions such as the Smooth Hinge, Sigmodial, Binomial Log Likelihood (BNLL) and Weighted Biased Regression (WBR) loss functions may be used, which mostly differ in the way they treat outliers (see Fig. 1).

Regularization

Determining the model parameters such that the training data are recalled optimally can lead to overfitting, i.e. the model parameters are particularly adjusted to recall the property values of the training set, but may show low performance in predicting property values of new compounds. Therefore, the objective function is augmented by two regularization terms: a quadratic Tikhonov, $\lambda_2\ ||\vec{w}||_2^2$, [32] and a linear Lasso regularization, $\lambda_1\ ||\vec{w}||_1^1$, [33] $(0 < \lambda_2 + \lambda_1 < 1)$. Both regularization terms adopt their minimum value when all components of the parameter vector $\vec{w}$ vanish. Normally this is in conflict with the first term of Eq. 1, which requires specific non-vanishing model parameters. The trade-off is that the model parameters governing the less important features are set to small (Tikhonov) or even vanishing (Lasso) values. Model parameters referring to features that exhibit strong correlations with the property values are kept. Hence, increasing the strength of one of the two regularization terms by increasing the corresponding λ value leads to a simplified model description. As a consequence the recall performance decreases, while simultaneously the prediction performance can increase, which demonstrates that overfitting is reduced. However, if the regularization terms become too large not only recall but also prediction performance will decrease, as important features may now be suppressed as well.

The aim is to optimize the regularization parameters (λ_1, λ_2) by increasing their values up to the point just before the prediction performance starts to diminish. DemQSAR

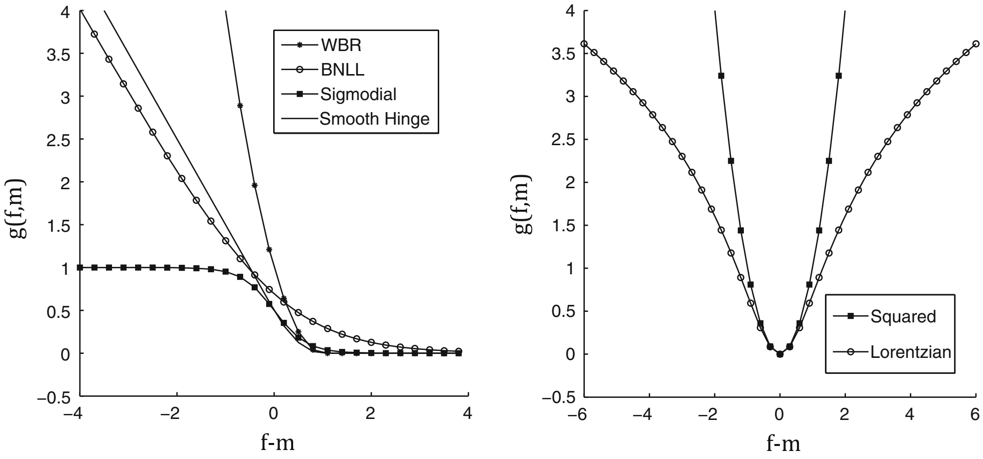

Fig. 1 Different loss functions *g(f,m)* implemented in DemQSAR as a function of the difference $f - m$ between the value of the scoring function *f* estimating the property value and the corresponding true property value *m*. *Left side* one-sided loss functions, which can be used for two-class classification tasks. *Right side* symmetric loss functions, which can be used for classification and regression tasks. Algebraic expressions of the different loss functions can be found in the supplementary material

does this automatically by evaluating the prediction performance observing the error in *k*-times *n*-fold cross-validation. For that purpose we define a candidate set of λ values which cover a range to be tested. For each λ value of this set the training set is randomly divided into *n* subsets. One subset is retained as a validation set while the other $n - 1$ subsets are used to train the model. This process is repeated *n* times such that each subset is used exactly once as a validation set. Repeating the whole process *k*-times ensures that the initial division has no influence on the estimated error. The average over these *k* times *n* validation errors is the estimated performance. The λ values that reveal the smallest validation error are then used to train the classifier on the whole dataset.

Feature selection

Recent software developments allow one to calculate many descriptors for molecular compounds. Selecting the features most relevant for prediction and reducing hidden dependencies among the features are key issues in deriving meaningful models. DemQSAR offers two approaches to automatically select relevant features.

The first one is to use a linear Lasso regularization. Due to its linear form the Lasso regularization leads to sparse models where the weights of many features are set to zero rigorously. According to the structure of the scoring function, Eq. 2, features whose weights are set to zero are ignored.

The second approach is to use a quadratic Tikhonov regularization. Using a Tikhonov regularization, none of the feature weights will be set exactly to zero. Nevertheless, the quadratic Tikhonov regularization can also be used for explicit feature selection. Assuming unimportant features will get smaller parameter weights than important ones, a backward selection can be performed by removing features with small weights after each training round. This approach is called RFE (recursive feature elimination) [34]. In spite of its simplicity, the RFE algorithm yields excellent results and has been successfully used in many classification and regression tasks [34–37].

To determine for both feature selection methods, RFE and Lasso, when to stop, the feature subset performances can be measured in two ways. The first method is to leave out a test set. Prediction performances on this test set are then used to estimate the prediction performance of the actual feature subset. However, it can be difficult to extract a representative test set from the whole data set, especially for small datasets one may want to use all available data for model building. Therefore, feature subset performances are estimated using k-times n-fold cross-validation, where in the present application $k = 5$ and $n = 10$. But, selecting a feature subset based on the smallest estimated error may not be the best choice. If a prediction model with optimal performance is desired, a large number of features may be necessary. Alternatively, a prediction model that uses only a small number of features is easier to interpret. These are two conflicting objectives: good performance is most often achieved with more features whereas easy interpretation requires small feature sets. To solve this conflict with variable emphasis we provide a feature selection criterion that can compromise between high performance and a small size of employed features. Let *Err*(*S*) be the error of feature subset *S* estimated via cross validation and let *Num*(*S*) be the number of features in subset *S*. We furthermore define S_{min} as the smallest feature subset during feature selection (one single feature) and S_{opt} as the feature subset with smallest estimated error. The finally selected subset S_{fin} fulfills the following conditions

$$\begin{aligned} &\min \quad Num(S_{fin}) \\ &\text{s.t.} \quad Err(S_{fin}) \leq Err(S_{opt}) + PST\big(Err(S_{min}) - Err(S_{opt})\big)\,{}^{,}. \\ &\text{for given } PST \text{ with } 0 \leq PST \leq 1. \end{aligned} \tag{3}$$

The quantity PST (performance-size-tradeoff) is a user defined parameter, which controls the finally selected feature subset S_{fin}. If PST is set to 0 the feature subset with the smallest estimated error will be selected, i.e. $S_{fin} = S_{opt}$. For this choice $Num(S_{fin})$ may be large. However, as mentioned above the user may be interested in smaller feature subsets. Larger PST values will yield feature subsets of smaller sizes. If PST is set to 1, the smallest possible feature set is taken, which consists of just a single feature, i.e. $S_{fin} = S_{min}$.

Nonlinear models

For most QSAR applications there are many more descriptors available than there are compounds. Hence, a linear model will most often be the best choice. However, there may be problems where a linear model is not sufficiently flexible to describe the studied data. In such a case, a non-linear transformation of the original model data into a generalized feature space may render the dataset more suitable for a linear separation. This corresponds to a non-linear separation in the original feature space using a more general hypersurface instead of a hyperplane. Nevertheless, an explicit transformation of the dataset of compounds may computationally be too expensive or even intractable. Instead of transforming the compound data explicitly, DemQSAR employs the so called kernel trick [38] to transform the data implicitly. For that purpose, the objective function L, Eq. 1, is rewritten such that the parameter vector $\vec{w}$ can be expressed as a weighted sum of the training feature vectors $\vec{x}_i$

$$\vec{w} = \sum_{i=1}^{N} \alpha_i \vec{x}_i, \tag{4}$$

with

$$\alpha_i = -\frac{(1-\lambda_2)\mu_i}{N2\lambda_2} \frac{\partial g(f(\vec{w}, b; \vec{x}_i), m_i)}{\partial f}. \tag{5}$$

Hence, we can rewrite the linear scoring function as

$$f(\vec{w}, b; \vec{x}) = \sum_{i=1}^{N} (\alpha_i \vec{x}^t \cdot \vec{x}_i + b) = \sum_{i=1}^{N} (\alpha_i K(\vec{x}^t, \vec{x}_i) + b). \tag{6}$$

Now, instead of finding the hyperplane normal $\vec{w}$, we determine the scalar multipliers α_ι Eq. 5. Note that finding the multipliers as well as predicting new data points can be done solely using the values of the dot products $\vec{x}^t \cdot \vec{x}_i$. Replacing all dot products by a kernel function $K(\vec{x}^t, \vec{x}_i)$ transforms the linear model into a non-linear model [39, 40]. DemQSAR comes with three widely used kernel functions which can be used for non-linear classification and regression: the polynomial kernel, radial basis function (RBF), and sigmodial kernel. Algebraic expressions of these kernel functions can be found as supplementary material.

Classification of unbalanced data

For a two-class (+ and −class) classification task problems can arise if the sizes of the two classes available for learning are very different. To avoid false positives for the majority class, it can be advantageous to split the N^+ positive $\vec{x}_i^+$ from the N^- negative $\vec{x}_j^-$ data ($N = N^+ + N^-$) leading to the balanced objective function:

$$L_{balanced} = (1-\lambda_2) \sum_{s=+-} \frac{w_s}{N^s} \sum_{i=1}^{N^s} \mu_i g\big(f(\vec{x}_i^s), m_i^s\big) \; + \lambda_2 \vec{w}^t \cdot \vec{w}, \tag{7}$$

where N is the size of the data set and w^+ and w^- ($w^+ + w^- = 1$) are the weights for the positive and negative data, respectively.

Normalization and transformation

As each feature represents a different molecular property whose numerical value depends on the units used, the values of different features can span very different scales. Therefore, it is important to normalize the features of the training dataset before the computations are performed to ensure a fair comparison between different features. For this purpose, each feature of the training set is subject to a linear transformation such that it has a zero mean and a standard deviation of unity. Of course, the same linear transformation must also be applied for the test set. Features that have a standard deviation of zero in the training set do not contain any information for model building and are therefore removed from training and test sets before further processing. Since both properties VD_{ss} and CL cover a broad range of values and are not normally distributed, they have been transformed to $\log_{10}$ scale prior to model building, yielding $\widehat{m}_i = \log_{10}(m_i)$. Figure 2 shows a binned histogram of the log transformed VD_{ss} and CL values for the *DataWhole*.

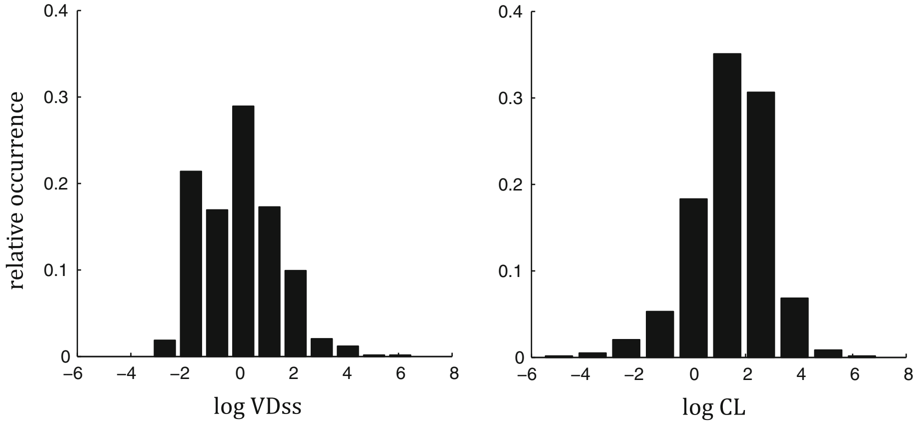

Fig. 2 Binned histogram of the log transformed VD_{ss} and CL values for the *DataWhole*

Quality assessment

Two different quality measurements were used to assess prediction performance. These are the predictive coefficient of determination (R^2),

$$R^2 = 1 - \frac{\sum_m \left(\widehat{m}_{exp} - \widehat{m}_{pred}\right)^2}{\sum_m \left(\widehat{m}_{exp} - \text{average}\left(\widehat{m}_{exp}\right)\right)^2} \tag{8}$$

and the geometric mean fold error (GMFE),

$$\text{GMFE} = exp\left|\frac{1}{N}\sum_m \ln\left(\frac{m_{pred}}{m_{exp}}\right)\right|. \tag{9}$$

Where N denotes the size of the predicted data set; and m_{exp} and m_{pred} represent the experimental and predicted target values respectively (here, $m = VD_{ss}$ or CL). $\widehat{m}_{exp}$, $\widehat{m}_{pred}$ denote the $\log_{10}$ transformed experimental and predicted target values respectively. Note that many papers compute R^2 by just squaring the linear correlation coefficient, sometimes also called Pearson's product-moment coefficient. This may result in different values compared to using the correct definition, Eq. 8 [41]. The GMFE is not defined for negative target values. Hence, target values were transformed back from $\log_{10}$ scale to normal scale before computing the GMFE. The coefficient of determination, Eq. 8, measures the quadratic error and thus may be strongly affected by outliers. This is not the case for the GMFE, Eq. 9. A perfect model yields a GMFE value of 1. A model with a GMFE of 2 would make predictions on average twofold off (100% above or 50% below). Models with a GMFE $\leq$ 2 are usually considered to be successful [7].

Results

The squared error and the Lorentzian loss functions were tested and the former was found to perform better. From the considered kernels (linear, polynomial, RBF, sigmodial) the linear showed the best results (data not shown). Hence, all results shown in the present study were obtained with the linear squared error model. We had no additional information which data points are more reliable than others. Hence, all data points were weighted equally by setting the μ_i parameter to unity.

Using Lasso regularization for feature selection is much faster than the RFE method, since feature selection and model building are combined into a single training round. For RFE selection a new model has to be built after each feature elimination step. Nevertheless, a recursive feature selection yields more detailed insight into the importance of individual features, as only one is deleted after each training round. Thus, for our studies λ_1 has been set to zero, and feature selection has been performed using RFE. Suitable values of the regularization parameter λ_2 were determined as described in the Methods section. For our experiments the candidate set of λ_2 values to be tested during cross-validation has been set to [10^{-4}, 10^{-3}, 10^{-2}, 0.1, 0.2, 0.3…0.9, 0.93]. The optimized values were between 0.7 and 0.9 for all models. Smaller values resulted in a much lower prediction performance indicating that overfitting occurred (results not shown). This can be explained by the fact that the number of features is much larger than the number of data points.

DataTrain/*DataTest*

The *DataTrain* set has been used for initial model building with different *PST* values, Eq. 3. Table 3 shows prediction results of these models for the *DataTest* set. The *DataTest* set has been neither used for model building nor for feature selection and, hence, represents an independent test set.

For human VD_{ss}, using all 3,642 commercial features resulted in a GMFE of 2.03 and R^2 of 0.52. Using all open source CDK features yielded comparable results with a GMFE and R^2 of 2.11 and 0.45, respectively. Using feature

Table 3 Prediction results for *DataTest*. *DataTrain* has been used to build the model

Descriptors	Volume of distribution: VD_{ss}			Clearance: CL		
	# feat.	GMFE	R^2	# feat.	GMFE	R^2
Commercial						
All features	3,642	2.03	0.52	3,642	2.47	0.19
PST: 0.0	524	2.04	0.52	631	2.5	0.19
PST: 0.4	22	2.21	0.42	46	2.52	0.18
CDK						
All features	3,843	2.11	0.45	3,843	2.48	0.21
PST: 0.0	380	2.08	0.5	660	2.5	0.2
PST: 0.4	18	2.31	0.36	54	2.62	0.13

The regularization parameter λ_2 has been optimized by applying 5-times a 10-fold cross-validation

selection the initial number of features could be strongly reduced without deteriorating the prediction performance. With $PST = 0.0$, the RFE feature selection utilized 524 out of the 3,642 commercial features and performed almost similarly to using all features. A *PST* of 0.4 resulted in only 22 commercial features, predicting human VD_{ss} with a GMFE of 2.21. For open source CDK features, feature selection yielded similar results. Here, with $PST = 0.0$ the initial feature number was reduced from 3,843 to 380, resulting in a model that performed slightly better than using all features (GMFE: 2.08). $PST = 0.4$ resulted in just 18 features, predicting human VD_{ss} with a GMFE of 2.31.

Commercial and open source CDK features yield clearance prediction performances with GMFEs of about 2.5. Feature selection did not improve the results. Nevertheless, the number of features can be reduced considerably without decreasing the predictive power significantly.

Figure 3 shows the prediction performance for VD_{ss} on *DataTest* using CDK and commercial descriptors depending on the number of features. It can be seen clearly that the number of descriptors can be reduced remarkably while maintaining the predictive power. The commercial and CDK descriptors reach their minimum GMFE at 406 and 267 descriptors, respectively.

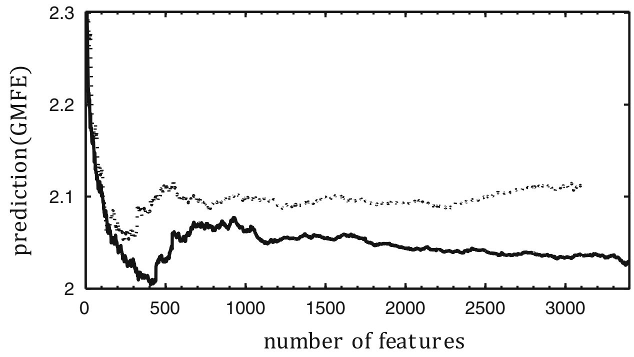

Fig. 3 Prediction performance for VD_{ss} on *DataTest* during feature selection with RFE. *Dotted line* CDK features, *Solid line* commercial features. Features with a standard deviation of zero were removed previous to model building

DataWhole

Table 4 shows the results of 10-fold cross-validation using *DataWhole*. During cross validation it is crucial not to use the left out subset for any stage of model building, including feature selection. Otherwise the test set information is implicitly used. This will lead to overly optimistic prediction results and does not reflect the real predictive power of the model [41]. In this study the left out subset has not been used at any stage of model building including feature selection. Thus the left out subset represents an independent, external test set which was randomly selected out of the training set. Repeating the prediction for all left out subsets all data points in *DataWhole* are predicted once as part of an independent, external test set. Therefore, the results in Table 4 represent an unbiased prediction of *DataWhole* and are representative for the predictive power of the model.

For human VD_{ss} predictions the results are consistent with those found for the initial single split of the datasets from Lombardo and Obach. A small improvement in prediction performance can be explained by the fact that a larger fraction of compounds is used for training (90%) than for the initial single split (60%). Using all commercial features without feature selection, human VD_{ss} can be predicted with GMFE of 2.01 and R^2 of 0.56. Applying feature selection with $PST = 0.0$ yields a subset of 758 features, which perform as good as using all features. Further reduction of the number of features deteriorates the results slightly. Nevertheless, when using only 27 features human VD_{ss} can still be predicted with GMFE of 2.24.

CL predictions were slightly better for random cross-validation than for the artificial single split between training and test set. But, prediction performance of CL remains worse than for VD_{ss} [42]. Again, the best results were obtained when using all commercial or CDK features resulting in a GMFE of about 2.4. Feature selection resulted in only slightly worse predictions, but was able to reduce the number of features remarkably.

Table 4 Prediction results for applying 10-fold cross-validation on *DataWhole*

Descriptors	Volume of distribution: VD_{ss}			Clearance: CL		
	# feat.[a]	GMFE[b]	R^{2b}	# feat.[a]	GMFE[b]	R^{2b}
Commercial						
All features	3,642	2.01	0.56	3,642	2.37	0.27
PST: 0.0	758	2.01	0.56	778.1	2.39	0.27
PST: 0.4	27	2.24	0.44	60.4	2.53	0.2
CDK						
All features	3,843	2.09	0.52	3,843	2.39	0.27
PST: 0.0	641.9	2.09	0.52	549.9	2.42	0.26
PST: 0.4	26.3	2.35	0.4	51.6	2.58	0.19

The regularization parameter λ_2 has been optimized applying 5-times 10-fold cross-validation. Whole model building procedure including feature selection has been performed for every cross-validation step, i.e. each data point has been predicted once as part of an external, independent test set. Thus, these results represent an unbiased prediction of *DataWhole*

[a] Number of selected features averaged over 10-fold cross-validation

[b] Results computed for the complete dataset (*DataWhole*), not averaged over 10-folds

DataExternal

Finally all available compounds of *DataWhole* have been used for model building with various *PST* parameter values. Table 5 shows prediction results of these models on *DataExternal*. For human VD_{ss} all feature subsets show excellent performance. Using only 27 commercial features the external test set could be predicted with GMFE of 1.81 and R^2 of 0.53. Similar results were obtained using open source CDK features. Using a subset of just 23 CDK features, a GMFE of 1.84 could be achieved. The supplemental material contains names and structures of the best four and worst four VD_{ss} predictions. However, it is not obvious which structural parts of a compound influence its prediction accuracy. One has to keep in mind that all measured values originate from in vivo human experiments. Sample sizes are therefore very limited and sometimes highly biased. Hence, predictions may seem quite poor for individual compounds even if they are not. This fact hampers the determination of chemical groups which influence the prediction accuracy. Predictions for clearance were quite poor compared to VD_{ss} predictions. The best results were obtained using CDK features for the reduced set of 671 features ($PST = 0$), where GMFE and R^2 are 2.65 and 0.26, respectively.

Using very small feature subsets human VD_{ss} predictions were still of good quality allowing direct interpretation of the selected features. Setting $PST = 0.6$ resulted in eight commercial or eight CDK features that were able to predict the external test set with GMFE = 1.89 or GFME = 2.04, respectively. Figure 4 illustrates these selected features ordered by their absolute parameter value. Since all features were normalized before model building, relative parameter values reflect the importance of a particular feature. That is, the larger a parameter compared to others the more important its influence on human VD_{ss}, irrespective of its absolute magnitude. Fingerprints are hashed and therefore hard to interpret but all other physicochemical features have a clear

Table 5 Prediction results for *DataExternal*

Descriptors	Volume of distribution: VD_{ss}			Clearance: CL		
	# feat.	GMFE	R^2	# feat.	GMFE	R^2
Commercial						
All features	3,642	1.79	0.47	3,642	2.64	0.18
PST: 0.0	836	1.79	0.48	788	2.67	0.15
PST: 0.4	27	1.81	0.53	65	2.69	0.17
PST: 0.6	8	1.89	0.49	25	2.76	0.10
CDK						
All features	3,843	1.93	0.37	3,843	2.65	0.26
PST: 0.0	578	1.88	0.45	671	2.62	0.20
PST: 0.4	23	1.84	0.45	49	2.77	0.09
PST: 0.6	8	2.04	0.36	19	3	0.04

Prediction models were generated using *DataWhole*

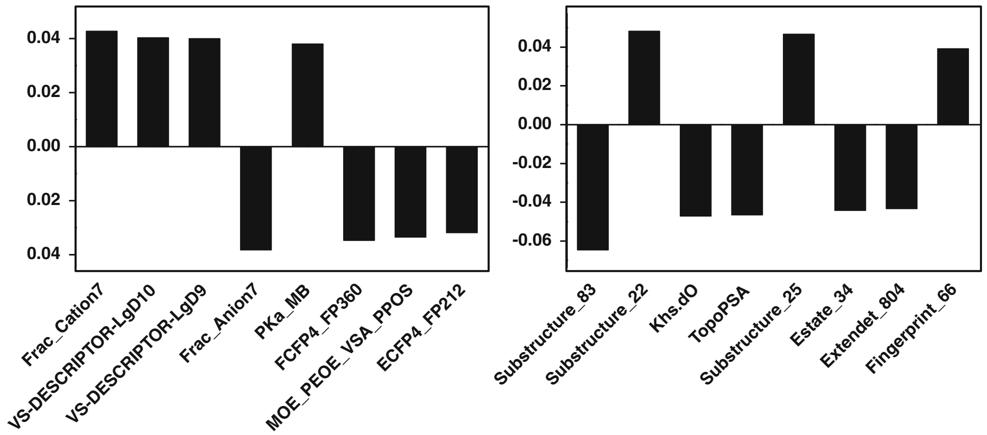

Fig. 4 Parameter weights for the human VD_{ss} prediction model using all data *(DataWhole)* with $PST = 0.6$ where eight features remain. As larger the absolute values of the weights are as more important are the corresponding features. *Left side* commercial features. *Right side* CDK features

meaning. Among the commercial features strong positive correlations with human VD_{ss} were observed for: “fraction cationic at pH 7”, “pKa MB (mostly basic)” and “VS-descriptor LgD 10/9/8 (logarithm of the partition coefficient between 1-octanol and water)”. Strong negative correlations were observed for the following interpretable feature: “fraction anionic at pH 7”. Hence, the human VD_{ss} of a compound could be enhanced by increasing its cationic fraction and decreasing its anionic fraction at pH 7 or by increasing its octanol–water distribution coefficient, logD (hydrophobic drugs). For CDK features strong positive correlations with human VD_{ss} were observed for various substructures: 83 (carboxylic ester), 22 (primary aliph amine) and 25 (quaternary aliph ammonium). Smart patterns for these substructures are provided as supplementary material. Furthermore negative correlations were observed for TopoPSA and Khs.do. TopoPSA is a calculation of topological polar surface area based on fragment contributions. Khs.do counts the number of double bonded oxygen atoms ($O = *$).

Y-randomization

Another tool for model validation in QSAR/QSPR analysis is y-randomization [43]. The performance of the original model is compared to that of models built for randomly shuffled property values, based on the original features. For this purpose, the VD_{ss} and CL values of *DataWhole* were randomly shuffled and used for model building including automatic feature selection. The generated VD_{ss} and CL models have been used to predict the VD_{ss} and CL values of *DataExternal* respectively. This procedure has been repeated 25 times. As seen in Table 6 the y-randomized models show no predictive power, having negative R^2 and large GMFE. These results indicate that the CL models, despite their overall low performance, still have some statistical significance.

Discussion

In the present application we used DemQSAR to build *in silico* models for human volume of distribution at steady state (VD_{ss}) and human clearance (CL). The prediction performances were measured on a left out test set (*DataTest*) as well as by applying a 10-fold cross-validation for all data (*Data10Fold*). The DemQSAR application was able to select small feature sets that yielded GMFE prediction performance of about 2.0 for human VD_{ss}. Human CL prediction was performing considerably worse. This

Table 6 Y-randomization results

Descriptors	Volume of distribution: VD_{ss}			Clearance: CL		
	# feat.	GMFE	R^2	# feat.	GMFE	R^2
All features	3,642	2.7	−0.14	3,642	3.2	−0.1
PST: 0.0	692	2.66	−0.1	692	3.18	−0.08
PST: 0.4	70.6	2.62	−0.05	73.5	3.17	−0.05

The VD_{ss} and CL values of *DataWhole* were randomly shuffled and used for model building. The generated models have been used to predict *DataExternal*. Given values represent averages over 25 separate runs can be explained by the fact that clearance depends strongly on drug metabolism, which itself is highly variable (numerous diverse enzymes, multiple sites per molecule) [42]. Nevertheless, y-randomization experiments showed that predictions for clearance are still statistically significant with GMFE of about 2.35.

Prediction results were verified on an external independent test set of 29 compounds (*DataExternal*). Here our models achieved GMFE = 1.8 for human VD_{ss} using just 27 selected descriptors. Berellini et al. [11] used the same original dataset which consists of 669 compounds to perform *in silico* predictions of human VD_{ss}. The models have been assessed using the same external test set of 29 compounds used in this study (*DataExternal*). Their best model yielded also a GMFE of 1.8.

Yu [10] combined the original dataset of 669 compounds and the additional 29 compounds of the *DataExternal* to build models for the prediction of human CL. His model did not perform well on neutral compounds, which were therefore removed previous to model building. On the remaining 462 compounds Yu's model yielded GMFE = 2.17 in a fivefold cross-validation and GMFE = 2.25 in a threefold cross-validation. However, when predicting all compounds including the neutral ones Yu's model just achieved GMFE = 2.46 in a threefold cross-validation. Our models yielded a GMFE of about 2.4 using a 10-fold cross-validation on all data (*DataWhole*) including neutral compounds which is comparable to Yu's results.

Prediction performances using the open source CDK feature sets were most often as good as those using their commercial counterparts reaching GMFEs of about 2.0 for human VD_{ss} predictions and around 2.4 for human CL predictions in 10-fold cross-validation. Hence, CDK features seem to be a good alternative to commercial features for scientific research [44].

The regularization parameters λ_2 had a large influence on the predictive power of the models (results not shown). If the regularization parameter was set too small overfitting occurred resulting in low predictive power. This can be explained by the fact that the number of initial descriptors is much larger than the number of data points. Hence, it is very important to set the regularization parameter to an appropriate value. DemQSAR can do this in an automated way as proposed in the Methods section.

A simple linear model without any kernels yielded best results. This can be explained by the fact that the usage of kernels adds a nonlinearity and thus additional information to the input descriptors. Since the number of parameters is

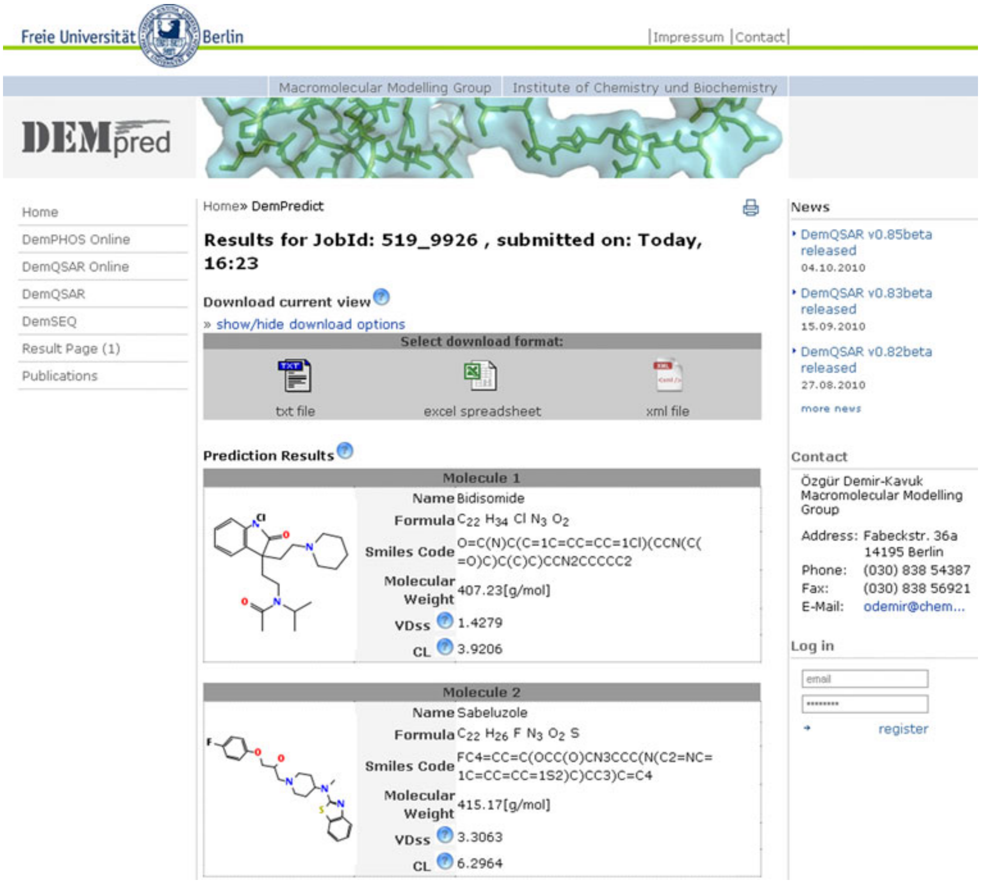

Fig. 5 Human VD_{ss} and CL values are predicted. Furthermore 2-dimensional structures, smiles codes, molecular structure formula and molecular weights are computed for a better overview. All results can be downloaded as preformatted Excel, text and XML files

Results for JobId: 519_9926 , submitted on: Today, 16:23

Download current view

» show/hide download options

Select download format:

Prediction Results

Molecule 1

Name	Bidisomide
Formula	$C_{22} H_{34} Cl N_3 O_2$
Smiles Code	O=C(N)C(C=1C=CC=CC=1Cl)(CCN(C(=O)C)C(C)C)CCN2CCCCC2
Molecular Weight	407.23[g/mol]
VDss	1.4279
CL	3.9206

Molecule 2

Name	Sabeluzole
Formula	$C_{22} H_{26} F N_3 O_2 S$
Smiles Code	FC4=CC=C(OCC(O)CN3CCC(N(C2=NC=1C=CC=CC=1S2)C)CC3)C=C4
Molecular Weight	415.17[g/mol]
VDss	3.3063
CL	6.2964

Özgür Demir-Kavuk
Macromolecular Modelling Group
Address: Fabeckstr. 36a
14195 Berlin
Phone: (030) 838 54387
Fax: (030) 838 56921
E-Mail: odemir@chem...

much larger than the number of training data, adding additional information from a kernel may hamper the model building process.

Many studies in this field perform a cross validation or leave one out prediction after model optimization or feature selection. However, in doing so, the test set information is used implicitly leading to biased prediction results, which are too optimistic [41]. In our computations these biased models were able to predict VD_{ss} with GMFE of 1.7 and CL with GMFE of 1.9, which is much better than the unbiased prediction performance (VD_{ss} GMFE 2.0, CL GMFE 2.4). Nevertheless, when done properly, cross validation is a powerful tool for estimating the predictive power of a model. Our results show that the cross validation results are consistent with those found for the initial split (*DataTrain*, *DataTest*). Therefore, during cross validation it is crucial not to use the left out subset for any stage of model optimization including feature selection.

Conclusions

We presented a publicly available and platform independent application for QSAR analysis called DemQSAR. DemQSAR integrates the open source CDK library to compute various molecular descriptors and fingerprints. QSAR models can thus be built without any additional software. Only two dimensional structures of the molecules have to be provided. CDK features can be easily complemented by external feature sets computed by other software. DemQSAR incorporates two state of the art feature selection strategies: embedded Lasso and RFE. Depending on whether a classification or regression analysis is performed the appropriate quality measures are computed automatically. A flowchart of the standalone application is provided as supplementary material.

The final models for the prediction of human VD_{ss} and human CL are accessible through an easy to use web interface (http://agknapp.chemie.fu-berlin.de/dempred/demqsar). In addition to the predicted VD_{ss} and CL values, 2-dimensional images, smiles codes, molecular formula and molecular weights are computed for the uploaded compounds. All results can be exported in pre-formatted Excel, text and XML files. Figure 5 shows a screenshot of the DemQSAR web interface where compounds of the external test set were predicted. The provided web server is to our knowledge the only publicly available resource to predict human VD_{ss} and CL.

The focus of this paper was not just to build good models for VD_{ss} and CL predictions, but also to demonstrate examples of the web application DemQSAR. Nevertheless, the predictive power of the generated models for VD_{ss} and CL are comparable to the best results of previous publications. All models were generated with a publicly available, fully automated approach. No additional programming was needed by the user and features were not selected by hand as done in other works. Hence, its public availability, fully automated approach and good predictive power make DemQSAR an interesting tool for many other QSA/PR tasks.

Acknowledgments The authors are grateful to the developers of the CDK library, which made it possible to make this application publicly available. We also like to acknowledge useful comments from an anonymous referee. This work was supported by the International Research Training Group (IRTG) on “Genomics and Systems Biology of Molecular Networks” (GRK1360 and GRK1772, German Research Foundation (DFG)).

References

1. Paul SM et al (2010) How to improve R&D productivity: the pharmaceutical industry’s grand challenge. Nat Rev Drug Discov 9(3):203–214
2. Kola I, Landis J (2004) Can the pharmaceutical industry reduce attrition rates? Nat Rev Drug Discov 3(8):711–715
3. Lau YY et al (2002) Development of a novel in vitro model to predict hepatic clearance using fresh, cryopreserved, and sandwich-cultured hepatocytes. Drug Metab Dispos 30(12):1446–1454
4. Obach RS (1999) Prediction of human clearance of twenty-nine drugs from hepatic microsomal intrinsic clearance data: an examination of in vitro half-life approach and nonspecific binding to microsomes. Drug Metab Dispos 27(11):1350–1359
5. Stringer RA, Strain-Damerell C, Nicklin P, Houston JB (2009) Evaluation of recombinant cytochrome P450 enzymes as an in vitro system for metabolic clearance predictions. Drug Metab Dispos 37(5):1025–1034
6. Rotroff DM et al (2010) Incorporating human dosimetry and exposure into high-throughput in vitro toxicity screening. Toxicol Sci 117(2):348–358
7. Obach RS et al (1997) The prediction of human pharmacokinetic parameters from preclinical and in vitro metabolism data. J Pharmacol Exp Ther 283(1):46–58
8. Vilar S, Chakrabarti M, Costanzi S (2010) Prediction of passive blood-brain partitioning: straightforward and effective classification models based on in silico derived physicochemical descriptors. J Mol Graph Model 28(8):899–903
9. Hutter MC (2009) In silico prediction of drug properties. Curr Med Chem 16(2):189–202
10. Yu MJ (2010) Predicting total clearance in humans from chemical structure. J Chem Inf Model 50(7):1284–1295
11. Berellini G, Springer C, Waters NJ, Lombardo F (2009) In silico prediction of volume of distribution in human using linear and nonlinear models on a 669 compound data set. J Med Chem 52(14):4488–4495
12. Olah M, Bologa C, Oprea TI (2004) An automated PLS search for biologically relevant QSAR descriptors. J Comput Aided Mol Des 18(7–9):437–449
13. Gleeson MP, Hersey A, Montanari D, Overington J (2011) Probing the links between in vitro potency, ADMET and physicochemical parameters. Nat Rev Drug Discov 10(3):197–208
14. Tropsha A (2010) Best practices for QSAR model development, validation, and exploitation. Mol Inform 29(6–7):476–488
15. Wolff ME (1995) Burger’s medicinal chemistry and drug discovery. J Chem Educ 72(8):A170

16. Hoekman D (1996) Exploring QSAR fundamentals and applications in chemistry and biology. J Am Chem Soc 118(43):10678
17. Dearden JC, Cronin MT, Kaiser KL (2009) How not to develop a quantitative structure-activity or structure-property relationship (QSAR/QSPR). SAR QSAR Environ Res 20(3–4):241–266
18. Cruciani G, Pastor M, Guba W (2000) VolSurf: a new tool for the pharmacokinetic optimization of lead compounds. Eur J Pharm Sci 11(Supplement 2):S29–S39
19. Stevenson JM, Mulready PD (2003) Pipeline Pilot 2.1 By Scitegic, 9665 Chesapeake Drive, Suite 401, San Diego, CA 92123-1365. www.scitegic.com. J Am Chem Soc 125(5):1437–1438
20. Vilar S, Cozza G, Moro S (2008) Medicinal chemistry and the molecular operating environment (MOE): application of QSAR and molecular docking to drug discovery. Curr Top Med Chem 8(18):1555–1572
21. Joachims T (1999) Making large-scale support vector machine learning practical. Advances in kernel methods: support vector learning. MIT Press, Cambridge, MA, pp 169–184
22. Igel C, Glasmachers T, Heidrich-Meisner V (2008) Shark. J Mach Learn Res 9:993–996
23. Abeel T, Van de Peer Y, Saeys Y (2009) Java-ML: a machine learning library. J Mach Learn Res 10:931–934
24. Hall M et al (2009) The WEKA data mining software: an update. SIGKDD Explor Newsl 11(1):10–18
25. Schaul T et al (2010) PyBrain. J Mach Learn Res 11:743–746
26. Li ZR et al (2007) MODEL-molecular descriptor lab: a web-based server for computing structural and physicochemical features of compounds. Biotechnol Bioeng 97(2):389–396
27. Melville JL, Hirst JD, TMACC (2007) Interpretable correlation descriptors for quantitative structure—activity relationships. J Chem Inf Model 47(2):626–634
28. Cited; Available from: http://www.ra.cs.uni-tuebingen.de/software/joelib/
29. Steinbeck C et al (2006) Recent developments of the chemistry development kit (CDK)—an open-source java library for chemo- and bioinformatics. Curr Pharm Des 12(17):2111–2120
30. Obach RS, Lombardo F, Waters NJ (2008) Trend analysis of a database of intravenous pharmacokinetic parameters in humans for 670 drug compounds. Drug Metab Dispos 36(7):1385–1405
31. Lombardo F et al (2006) A hybrid mixture discriminant analysis-random forest computational model for the prediction of volume of distribution of drugs in human. J Med Chem 49(7):2262–2267
32. Tychonoff AN (1943) On the stability of inverse problems. Dokl Akad Nauk SSSR 39(5):195–198
33. Tibshirani R (1996) Regression shrinkage and selection via the lasso. R Stat Soc Series B Stat Methodol 58(1):267–288
34. Guyon I, Weston J, Barnhill S, Vapnik V (2002) Gene selection for cancer classification using support vector machines. Mach Learn 46(1–3):389–422
35. Yu H, Yang J, Wang W, Han J (2003) Discovering compact and highly discriminative features or feature combinations of drug activities using support vector machines. Proc IEEE Comput Soc Bioinform Conf 2:220–228
36. Li H et al (2005) Effect of selection of molecular descriptors on the prediction of blood-brain barrier penetrating and nonpenetrating agents by statistical learning methods. J Chem Inf Model 45(5):1376–1384
37. Zhu J, Hastie T (2004) Classification of gene microarrays by penalized logistic regression. Biostatistics 5(3):427–443
38. Aizerman A, Braverman EM, Rozoner LI (1964) Theoretical foundations of the potential function method in pattern recognition learning. Autom Remote Control 25:821–837
39. Genton MG (2001) Classes of kernels for machine learning: a statistics perspective. J Mach Learn Res 2:299–312
40. Shawe-Taylor J, Cristianini N (2004) Kernel methods for pattern analysis. Cambridge University Press, Cambridge
41. Golbraikh A, Tropsha A (2002) Beware of q2! J Mol Graph Model 20((4):):269–276
42. Gleeson MP, Hersey A, Hannongbua S (2011) In-silico ADME models: a general assessment of their utility in drug discovery applications. Curr Top Med Chem 11(4):358–381
43. Rucker C, Rucker G, Meringer M (2007) y-Randomization and its variants in QSPR/QSAR. J Chem Inf Model 47(6):2345–2357
44. Gupta RR et al (2010) Using open source computational tools for predicting human metabolic stability and additional absorption, distribution, metabolism, excretion, and toxicity properties. Drug Metab Dispos 38(11):2083–2090
45. Adv. Chemistry Development Inc, Toronto, Canada. Cited; Available from: http://www.acdlabs.com/home/
46. clogP v4.3, BioByte Corp., Claremont, USA. Cited; Available from: http://www.biobyte.com/index.html
47. VolSurf, Molecular Discovery Ltd, Middlesex, UK. Cited; Available from: http://www.moldiscovery.com/index.php
48. Molconn-Z, EduSoft LC, Richmond USA. Cited; Available from: http://www.edusoft-lc.com/
49. Pipeline Pilot v7.5.2, Accelrys, San Diego USA. Cited; Available from: http://accelrys.com/
50. MOE, Chemical Computing Group Inc, Montréal, Canada. Cited; Available from: http://www.chemcomp.com/
51. Steinbeck C et al (2003) The Chemistry Development Kit (CDK): an open-source Java library for chemo- and bioinformatics. J Chem Inf Comput Sci 43(2):493–500