J Comput Aided Mol Des (2008) 22:431–440
DOI 10.1007/s10822-008-9193-8

Automatic QSAR modeling of ADME properties: blood–brain barrier penetration and aqueous solubility

Olga Obrezanova · Joelle M. R. Gola · Edmund J. Champness · Matthew D. Segall

Received: 19 October 2007 / Accepted: 30 January 2008 / Published online: 14 February 2008

Abstract In this article, we present an automatic model generation process for building QSAR models using Gaussian Processes, a powerful machine learning modeling method. We describe the stages of the process that ensure models are built and validated within a rigorous framework: descriptor calculation, splitting data into training, validation and test sets, descriptor filtering, application of modeling techniques and selection of the best model. We apply this automatic process to data sets of blood–brain barrier penetration and aqueous solubility and compare the resulting automatically generated models with 'manually' built models using external test sets. The results demonstrate the effectiveness of the automatic model generation process for two types of data sets commonly encountered in building ADME QSAR models, a small set of in vivo data and a large set of physico-chemical data.

Keywords Automatic model generation process · QSAR modeling · ADME properties · Blood–brain barrier penetration · Aqueous solubility · Gaussian Processes · Drug discovery

Introduction

The demand for fast model (re)building whenever data becomes available has given rise to a trend to develop computational algorithms for automatic model generation [1, 2]. Automatic modeling processes allow computational scientists to explore large numbers of modeling approaches very efficiently and make QSAR/QSPR model building accessible to non-experts. Automatic model generation requires unsupervised computational techniques which do not require any input from a user, are able to deal with a large number of descriptors and are not prone to overtraining. In recent years, a variety of such modern modeling techniques have been applied to QSAR modeling; some examples include Bayesian Neural Networks [3], Associative Neural Networks [4] and Gaussian Processes [5–7].

In our previous work [6] we developed new techniques for implementing the Gaussian Processes method and compared their performance with other modeling techniques. The Gaussian Processes method is highly appropriate for automatic model generation: it does not require subjective determination of model parameters, is able to handle a large pool of descriptors and select the important ones, it is inherently resistant to overtraining and offers a way of estimating uncertainty in predictions. It is a powerful, robust method capable of modeling non-linear relationships.

In this article, we present an automatic model generation process at the core of which is the Gaussian Processes modeling technique. The process manages each of the steps necessary to ensure that models are built and evaluated consistently within a rigorous framework. In a truly automatic process, the user is only required to provide the minimum necessary initial information; a set of molecules and a set of associated property values to be modeled. From this starting point, the process applies a number of different mathematical techniques to generate a series of models before selecting the best one.

To demonstrate the applicability of the automatic model generation process described, we present results comparing two models built 'automatically' with two equivalent

O. Obrezanova (✉) · J. M. R. Gola · E. J. Champness · M. D. Segall
BioFocus DPI Ltd., Darwin Building, Chesterford Research Park, Saffron Walden CB10 1XL, UK
e-mail: olga.obrezanova@glpg.com

models developed 'manually' by computational chemists. The first model is of blood–brain barrier penetration and the second is of aqueous solubility. We would like to note that these data sets are different from the ones modeled in our previous work [6]. The 'manual' and 'automatic' models are then tested on new external data. We selected these properties as they represent two commonly encountered types of data sets when building ADME QSAR models. Namely, a small set of in vivo data and a large set of physico-chemical data.

We will present the stages of the automatic model generation process in section "Automatic model generation process". A description of the original 'manual' models, the data sets used and the external data sets is given in section "'Manually' built models". In section "Application of the automatic model generation process" we will present results of the application of the process to model building and compare the resulting models with the 'manual' models. We will also demonstrate how a model for blood–brain barrier penetration can be rebuilt by including new data points.

Automatic model generation process

In this section we describe an automatic model generation process which can be subdivided into following stages:

- Descriptor calculation
- Data set splitting
- Descriptor pre-filtering
- Application of modeling techniques
- Selection of the best model

Descriptor calculation

Within our automatic model generation process we use 330 in-house molecular descriptors. These comprise a total of 321 two-dimensional SMARTS [8] based descriptors and nine whole molecule properties including log*P*, molecular weight, topological polar surface area [9] and the McGowan's Volume [10]. The SMARTS based descriptors are counts of atom types (e.g. fluorine atoms) and counts of functional groups (e.g. ketone).

We have chosen SMARTS based descriptors because they have worked well when modeling a number of QSAR and ADME data sets.

Data set splitting

In order to rigorously select the best model and assess its predictive power, the initial data set is split into three subsets, known hereafter as the training, validation and test sets. The training set is used to fit several models to the observed data and the validation set is used to select the best model from among all of the models built. The test set is completely independent and is used to assess the predictive power of the chosen model. Within our process we keep 70% of the compounds for the training set. The validation and test sets each contain 15% of initial data set.

The use of two independent sets to supervise model performance might seem to be wasteful when data are scarce, but in this way we can be confident that the test set is truly independent of both the model building and selection processes.

To split the initial data set into three subsets, we have used an approach based on cluster analysis. Compounds are clustered using an unsupervised non-hierarchical clustering algorithm developed by Butina [11]. The cluster analysis of the chemical structures is based on 2D path-based chemical fingerprints and the Tanimoto similarity index. The algorithm identifies dense clusters, where similarity within each cluster is controlled by a user-defined Tanimoto value. Compounds that do not belong to any cluster are referred to as singletons. Once the clusters are formed, the cluster centroids and singletons are put into the training set. Then the remaining compounds in each cluster are sorted by Y values (the modeled property) and divided into bins. Compounds from each bin are divided randomly between the training, validation and test sets to obtain the required proportions. The Tanimoto value used during clustering is 0.7.

A random split into three subsets would provide a more rigorous estimate of the predictive ability of a model. However, although it is likely to work well for large data sets, a random split is less likely to be suitable for modeling small data sets. In a random split, the ranges of property values in the three subsets could be very different and some chemistry present in the original set might not be represented well in the training set. Our choice of technique for set splitting was defined by an attempt to make the training set chemically representative of the original set of compounds and to make the ranges of property values approximately equal for the three subsets.

Descriptor pre-filtering

The calculated descriptors are subjected to a commonly used descriptor selection step that removes descriptors with low variance and low occurrence [12]. Specifically, descriptors with a standard deviation less than 0.0005 and descriptors represented by less than 4% of the compounds in the training set are excluded from the three sets. Also, highly correlated descriptors are excluded (when the

pairwise correlation exceeds 0.95 in the training set), such that just one of the pair remains.

Application of modeling techniques

We use the following techniques to build QSAR models within our automatic model generation process:

- Partial Least Squares (PLS)
  This is a well-known robust technique for generation of linear models based on multiple descriptors [13]. To determine the optimal number of PLS components we use a cross-validation technique.
- Gaussian Processes (GP)
  This machine learning technique produces six models, each of which uses a variation of the non-linear modeling technique. Three of them are built on a full set of descriptors (GP-Fixed, GP-2D, GP-Nest) and the other three variations have a built-in descriptor selection algorithm (GP-FVS, GP-RFVS, GP-Opt). Details of these variations are provided below.

Gaussian Processes is a powerful machine learning method based on a Bayesian probabilistic approach. This technique has many advantages. First, it does not require any a-priori determination of model parameters, which makes it very suitable for an automatic process. It also works well for a large pool of correlated descriptors and is robust to overtraining. Furthermore, Gaussian Processes provides an estimate of uncertainty together with each prediction. Despite these strengths there have been only a small number of published examples of its application to QSAR and ADME modeling [5–7, 14–16].

We have described the concept of the Gaussian Processes method for regression problems and provided details of our implementation in a previous publication [6]. Additionally, an in depth description of Gaussian Processes theory can be found in books by MacKay [17] and by Rasmussen and Williams [18]. However, below we summarize the basic concepts underlying this technique.

Gaussian Processes

The Gaussian Processes method proceeds by Bayesian inference directly in the space of functions and can be best illustrated graphically.

Initially, assume a prior probability distribution over functions controlled by 'hyperparameters' of the Gaussian Process. Figure 1a shows a number of sample functions drawn at random from the prior distribution specified by a particular Gaussian Process. Next we take into account available experimental data, observed property values. We consider only those functions from the prior which pass close to or exactly through the training points. The combination of the prior and the data leads to a posterior distribution over functions. Figure 1b shows a number of samples (as dashed lines) from the posterior distribution, i.e. functions that agree with the experimental observations. We can average over all functions in the posterior distribution and take the mean value as the prediction, shown by solid line in Fig. 1b. The regions between dotted lines denote twice the standard deviation at each input value and provide an estimate of the uncertainty for each prediction.

Choosing the prior distribution is important because it defines properties of functions considered for inference. These properties are controlled via the covariance function of the Gaussian Process, which ensures that the prediction function is smooth and matches the observed data as well as possible. We choose the Automatic Relevance Determination covariance function, the most common choice, because it allows us to identify the most relevant descriptors for describing the property via 'length scale' hyperparameters (parameters of the covariance function are

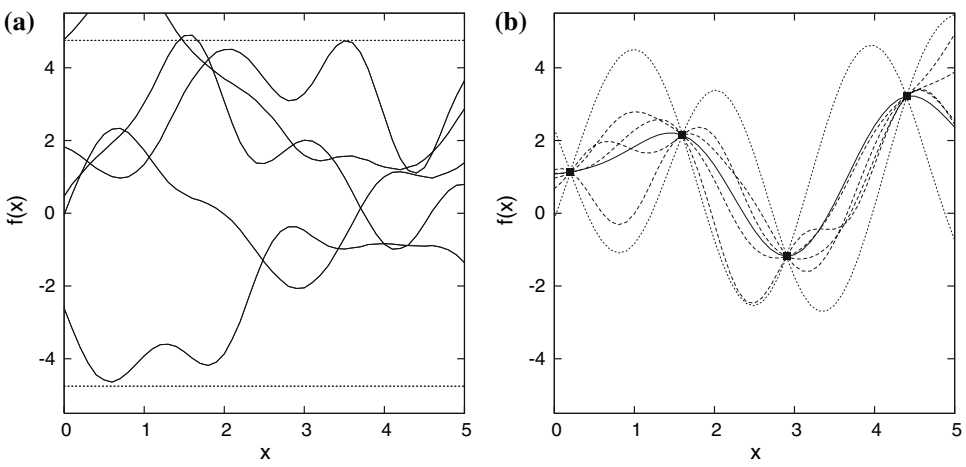

Fig. 1 Graphical illustration to Gaussian Processes inference. Graph (**a**) shows five functions drawn from the prior distribution. Graph (**b**) shows four samples drawn from the posterior distribution (dashed lines) and a mean of posterior distribution (solid line). In both plots the regions between dotted lines denote twice the standard deviation at each input value. In this example, it is assumed that there is no uncertainty in the observed data

called hyperparameters in the Bayesian approach, for details, see [6]).

The problem of *learning* the Gaussian Process model is the problem of finding the hyperparameters. They define the properties of the prediction function and control a tradeoff between smoothness and fitting the data. The hyperparameters are learned from the data and finding the best values for hyperparameters corresponds to optimizing the marginal log-likelihood [6, 18].

Optimization of the hyperparameters in multidimensional space is commonly performed by conjugate gradient methods. However this has two associated problems; first, the method is computationally expensive and, second, it might converge on a local maximum of the marginal log-likelihood as opposed to the global maximum. We have developed new techniques for finding hyperparameters which overcome some of these problems. A detailed description of the techniques is given in [6] and here we list them in order of the increasing computational time they demand:

- Fixed hyperparameters (GP-Fixed)
  All hyperparameters are set to fixed values which are proportional to the standard deviations of the observed property vector and descriptor columns. This is a very fast method and works well for large data sets.
- Two-dimensional search for parameters (GP-2D)
  Most of the hyperparameters are set to fixed values, and a search is performed for two parameters. One of those is a hyperparameter controlling the noise level in the model.
- Forward variable selection procedure and rescaled procedure (GP-FVS and GP-RFVS)
  By setting the length scales to fixed values, as in the previous two approaches, we lose the ability to identify the most relevant descriptors. This technique allows feature selection by employing a forward variable selection procedure starting from the hyperparameter values found by the GP-2D technique. The rescaled forward selection procedure involves updating the length scales depending on the number of descriptors used in the model.
- Optimization by conjugate gradients (GP-Opt)
  This technique performs optimization of the marginal log-likelihood in the space of the length scale hyperparameters only, using values found by the GP-2D technique as an initial point. The optimized length scale hyperparameters are used to select the most relevant descriptors and the final model is built using this selection.
- Nested Sampling (GP-Nest)
  This is the only technique that performs a search in the full hyperparameter space, exploring a wide prior space of hyperparameters. It does not get 'trapped' in local maxima of marginal log-likelihood. Nested sampling does not require any input from a user or any initial starting point; it offers complete automation of the learning process. If computational resources permit, this would be our technique of choice for finding hyperparameters. Although with this approach the models are built on a full set of descriptors, the length scale hyperparameters obtained can be used to identify and select the most important descriptors for describing the property.

Selection of the best model

To evaluate model performance we use two statistical measures—the root mean square error (RMSE) and the coefficient of determination R^2:

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(Y_i^{\text{obs}} - Y_i^{\text{pred}})^2}{\sum_{i=1}^{N}(Y_i^{\text{obs}} - \overline{Y_i^{\text{obs}}})^2},$$

where N is the set size, Y_i^{obs} is the observed value for compound i and Y_i^{pred} is the predicted value for compound i.

To choose the best out of all the available models we look at their performance on the validation set. Then the performance of the best model is assessed on the test set as a final test of the predictive ability of the model.

After the best model is selected we use the RMSE of the combined validation and test sets to obtain the RMSE of prediction.

In this paper we will also use the squared correlation coefficient as a statistic to evaluate performance of a model, although it is not used in the automatic model generation process. The squared correlation coefficient has often been used in QSAR modeling and it is defined as follows

$$r^2_{\text{corr}} = \frac{\left(\sum_{i=1}^{N}(Y_i^{\text{obs}} - \overline{Y_i^{\text{obs}}})(Y_i^{\text{pred}} - \overline{Y_i^{\text{pred}}})\right)^2}{\sum_{i=1}^{N}(Y_i^{\text{obs}} - \overline{Y_i^{\text{obs}}})^2 \sum_{i=1}^{N}(Y_i^{\text{pred}} - \overline{Y_i^{\text{pred}}})^2}.$$

Software implementation

The automatic model generation process described above is implemented in BioFocus DPI's software package for compound design, in silico modeling and optimization and is referred to as the Auto-Modeler[1] (http://www.biofocusdpi.com/In_silico_optimization). The process can

[1] Auto-Modeler™ is a trademark of Galapagos NV and/or its affiliates.

work at two levels; it requires only minimal input from a user, which is suitable for non-experts, but alternatively expert users can influence each stage of the process, e.g. import new descriptors (as SMARTS or imported data), use a predefined data set split or change the parameters of the automated data set split, descriptor pre-filtering or modeling techniques.

'Manually' built models

Two models that were constructed manually are used herein for comparison with the automatic model generation process. These are for blood–brain barrier penetration and aqueous solubility, expressed in terms of the logarithm of the concentration ratio between brain tissue and blood/plasma (logBB) and the logarithm of the intrinsic aqueous solubility (log*S* with *S* in μM), correspondingly.

In both cases, the process for building the models, from gathering model data to deploying the final model was lengthy due to the diversity of available tools and the variety of data formats and requirements necessary for using each of them. At each stage in the process, techniques for descriptor calculation, descriptor selection, data set splitting and finally model building were investigated. The methods used varied from third party modeling tools for PLS (Simca from Umetrics), rule-based regression models (Cubist from Rulequest) or unsupervised forward selection (UFS from the University of Portsmouth) to shell scripts developed in-house to calculate descriptors and run model building algorithms such as Radial Basis Functions [19]. Each tool had to be investigated and assessed to determine its value towards assisting in building the final model produced; as such we describe these models as manually built.

Here we present a summary of the processes followed to achieve both predictive models, followed by a description of additional external sets, recently obtained from the literature, that we have used to compare these models with those generated automatically.

Blood–brain barrier penetration model

After rigorous quality control of the quantitative data available from the literature, 151 compounds with logBB values were gathered. The set was split randomly into training (108 structures) and internal evaluation (43 structures) sets.

A set of 157 SMARTS-based 2D descriptors was calculated using a proprietary script. The number of descriptors was then reduced using the unsupervised forward selection program (UFS) [20] in order to minimise the possibility that chance correlations could give rise to a model. UFS removed descriptors from the training set with low variance as well as those with squared multiple correlation coefficients greater than 0.9, giving a reduced set of 64 descriptors from the initial 157 including calculated log*P*, descriptors relating to molecular size and shape plus those related to hydrogen bonding. All are physico-chemically relevant and are implicated in recently reviewed blood–brain barrier models [21]. The training set was investigated using various techniques including PLS, multiple linear regression (MLR), Cubist and RBF [19]. The latter, RBF, achieved the best statistical results on the test set with a model based on log*P* and six 2D descriptors including a flexibility index, molecular charge count and hydrogen bonding terms. The predictions for the internal evaluation set have an R^2 value of 0.73 with an RMSE of 0.36 log units. Note that the evaluation set was used to select the descriptors to use in the RBF model, so it cannot be considered as a truly independent test set.

Aqueous solubility model

Experimental aqueous solubility values for 20–30 °C, S, in μM, were obtained from the Syracuse database (see [22]) which provided a set of 3,313 organic compounds.

These compounds were randomly separated into training and test sets. Two thousand six hundred and fifty compounds, representing almost 80% of the data set, were taken as the training set. The remaining 663 compounds were used as a test set.

The initial set of 157 descriptors was reduced to 108 using the UFS program. Descriptors with standard deviations less than 0.05 and pairwise correlations exceeding 0.90 were excluded from the training and test set.

PLS, MLR, Cubist and RBF model building techniques were investigated. The RBF technique led to the best results on the test set of 663 compounds. The predictions for the test set have an R^2 value of 0.82 with an RMSE of 0.79 log units.

Additional external sets

Abraham et al. [23] reported literature values for the in vivo distribution of drugs from blood, plasma, or serum to rat brain for 207 compounds. A further 95 in vitro data points on volatile and inorganic compounds were also listed. Out of the total 302 compounds, 143 were unique to the Abraham set, i.e. not used when building the manual model, and could therefore be used to compare the manually and automatically built blood–brain barrier penetration models. We will refer to these 143 compounds

as the 'Abraham 143' set. 92 of these compounds are non volatile and 51 are volatile. It is interesting to note that Abraham et al. stressed the difference between the two in vivo and in vitro sets and recommended not to mix the data for modeling purposes [23].

Similarly, 564 compounds from the Huuskonen solubility set [24] are not present in the manually built solubility model. The experimental aqueous solubility values for these compounds were measured at 20–25 °C. The compounds are mainly hydrocarbons. We will refer to these 564 compounds as the 'Huuskonen 564' set and use this set to compare the manually and automatically built aqueous solubility models.

Application of the automatic model generation process

In this section we describe the application of the automatic model generation process to the blood–brain barrier penetration and aqueous solubility data sets described in sections "'Manually' built models—Blood–brain barrier penetration model" and "Aqueous solubility model" (we will refer to these as the 'original' data sets) and compare the resulting automatic models with the manual models on the additional external data sets described in section "Additional external sets".

We will also illustrate rebuilding the blood–brain barrier penetration model to include new compounds taken from the Abraham 143 set (see section "Additional external sets").

Blood–brain barrier penetration model

The automatic model generation process was applied to the original logBB set described in section "'Manually' built models—Blood–brain barrier penetration model". The full data set of 151 compounds was used as the input to the process. The results are summarized in Table 1.

The Gaussian Processes technique with nested sampling (GP-Nest) produced the best model for this data set. On the validation set, it achieved an $R^2 = 0.721$ and an $r^2_{corr} = 0.77$; on the test set $R^2 = 0.66$ and $r^2_{corr} = 0.73$; on the combined validation and test sets RMSE = 0.438. The RMSE of prediction is higher than that of the manual model (RMSE = 0.36 log units). However, a direct comparison is not possible because the manual model did not have an independent test set. Recently published models [25] have an RMSE of prediction of approximately 0.3–0.4 log units and the estimated experimental error in logBB measurements is about 0.3 log units. Figure 2 shows the graph of predicted logBB values versus observed for this model.

Table 1 Modeling blood–brain barrier penetration

Method	Desc.	TRN		VAL		TEST[a]	
		R^2	RMSE	R^2	RMSE	R^2	RMSE
PLS	162(1)[b]	0.50	0.48	0.55	0.48	0.53	0.58
GP-Fixed	162	0.76	0.33	0.71	0.38	0.65	0.50
GP-2D	162	0.77	0.33	0.72	0.38	0.66	0.50
GP-FVS	53	0.73	0.36	0.69	0.40	0.64	0.51
GP-RFVS	20	0.79	0.31	0.70	0.39	0.66	0.50
GP-Opt	16	0.81	0.30	0.63	0.43	0.62	0.52
GP-Nest	**162**	**0.79**	**0.32**	**0.72**	**0.38**	**0.66**	**0.49**

[a] Training set 106 compounds, validation set 23 compounds, test set 22 compounds

[b] Number of PLS components in brackets

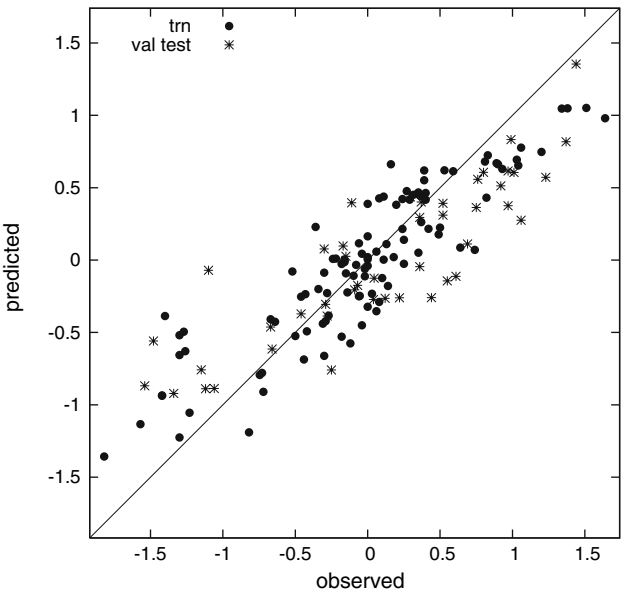

Fig. 2 Model of blood–brain barrier penetration obtained by the GP-Nest technique. Predicted logBB values versus observed for compounds in the training set (shown by circles) and the compounds in the validation and test sets (shown by stars). The line of unity ($y = x$) is shown for comparison

The results of applying the automatic blood–brain barrier penetration model and the manually built model to the additional external Abraham 143 set are shown in Table 2. Examining the R^2 statistic, the performance of both models appears quite disappointing. However, this statistic is not highly relevant due to the small range of values represented by the external set (the standard deviation in Y is 0.57 log units). This is comparable with the RMSE in prediction for the models and hence a high R^2 is not expected. An RMSE = 0.44 log units on this set using the manual model is much higher than the RMSE of prediction for that model, which equals 0.36 log units; while the RMSE achieved on this set by the GP-Nest model is much closer to that estimated during the model building process. Therefore, the

Table 2 Predictive ability of two blood–brain barrier penetration models based on the Abraham 143 set

Model	RMSE pred[a]	% Compounds predicted within		R^2	r^2_{corr}	RMSE[b]
		±0.4 log unit	±0.8 log unit			
Manual	0.36	62.9	93.0	0.39	0.44	0.44
GP-Nest	0.44	63.6	90.9	0.27	0.36	0.49

[a] RMSE of prediction for the model obtained on an original test (and validation) set

[b] RMSE for the Abraham 143 set

estimation of the predictive power of the automatic model is more realistic. The percentages of compounds predicted within 0.4 log units are very similar for both models (around 64%). Figure 3 shows the graph of predicted log-BB values versus observed for the Abraham 143 set.

We have analyzed the compounds mispredicted by the GP-Nest model by means of a structure similarity analysis between the original logBB set and the Abraham 143 set. We have defined mispredicted compounds as those with an absolute error in prediction greater than 0.6 log units. There are 24 compounds with this level of error. None of these compounds had a Tanimoto similarity greater than 0.7 to any compound in the training set, and hence, it is not surprising that these compounds were mispredicted. Nevertheless, there is no correlation between the error of prediction of the test set compounds and their Tanimoto similarity to the training set (which we defined as the Tanimoto level between the test set compound in consideration and its most similar compound from the training set).

Gaussian Processes methods provide us with the standard deviation in prediction, which can be used as an indicator of the domain of applicability of the model. If a compound lies far from any training set compound in the descriptor space of the model, this will be reflected by a large standard deviation reported for the prediction. We have considered the standard deviations of predictions by the GP-Nest model for the external Abraham 143 set, shown in Fig. 4. Unfortunately we did not see a particular correlation between the reported standard deviation of prediction and the actual error of prediction (difference between the predicted and the observed). The most mispredicted compound had a relatively large error bar, however, many compounds were either mispredicted but had small error bars or were predicted well but had large error bars. Possible explanations for this fact are the variability of the experimental data which were collected in different labs and by different methods (in vitro/in vivo), a poor population of compounds with low blood–brain ratio

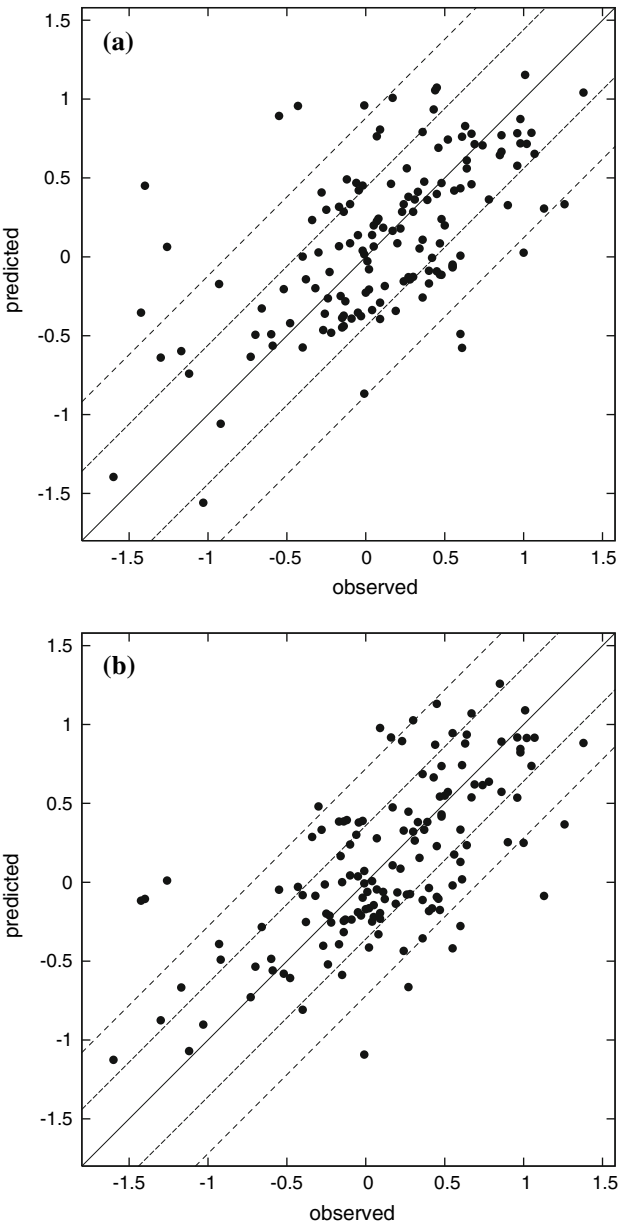

Fig. 3 Predicted logBB values versus observed for the Abraham 143 set. Wide dashed lines represent the 1-RMSE interval for each model, dashed lines represent the 2-RMSE interval. The line of unity ($y = x$) is shown as a solid line for comparison. (**a**) Prediction by the GP-Nest model build by the automatic model generation process (RMSE = 0.44). (**b**) Prediction by the manual model (RMSE = 0.36)

in the available data and possible sources of variation not captured by the descriptors used in the model.

Blood–brain barrier penetration model on new data

To illustrate the effect of rebuilding a model, we combined the original logBB set (see section "'Manually' built models—Blood–brain barrier penetration model") with the

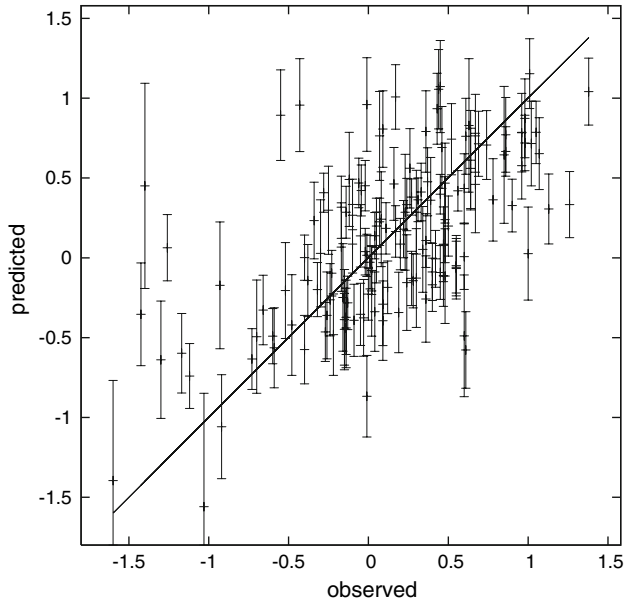

Fig. 4 Predicted logBB values versus observed for the Abraham 143 set with error bars. Predictions are made by the GP-Nest model. The error bars do not include the contribution of the noise term, which is constant. The line of unity ($y = x$) is shown for comparison

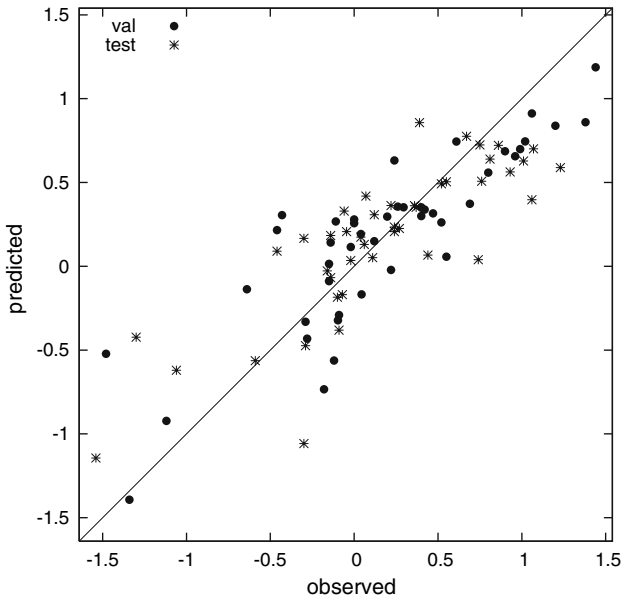

Fig. 5 Blood–brain barrier penetration model built on 292-compound data set by the GP-2D technique. Predicted logBB values versus observed for validation (shown by circles) and test sets (shown by stars). The line of unity ($y = x$) is shown for comparison

Abraham 143 set (see section "Additional external sets") and obtained a 292-compound set which we have used to build a new logBB model. The automatic model generation process was applied and the results are given in Table 3.

The GP-2D model is the best model, achieving on the validation set an $R^2 = 0.73$ and an $r^2_{\text{corr}} = 0.74$ and $R^2 = 0.67$, $r^2_{\text{corr}} = 0.67$ on the test set. On the combined validation and test sets $RMSE = 0.34$ which compares well with other published logBB models. The graph of predicted logBB versus observed for both the validation and test sets is given in Fig. 5. The GP-2D model built on 167 descriptors and 292 compounds covers a wider chemical space than the manual model described in section "'Manually' built models—Blood–brain barrier penetration model".

We have separated the 30 compounds from the validation and test sets of the GP-2D model which originated from the Abraham 143 set, then we looked at performance of the GP-2D model and the GP-Nest model (built on the original logBB set of 151 compounds, for details see section "Application of the automatic model generation process—Blood–brain barrier penetration model") on these 30 compounds. The original 'automatic' GP-Nest model achieved RMSE = 0.44, $R^2 = -0.1$ and $r^2_{\text{corr}} = 0.23$ on these 30 compounds. However the 'rebuilt' GP-2D model achieved RMSE = 0.27, $R^2 = 0.59$ and $r^2_{\text{corr}} = 0.61$, therefore the rebuilding of the model incorporating new data gave a significant improvement in prediction on these data.

Table 3 Modeling logBB on combined original data and the Abraham 143 set, 292-compound set

Method	Desc.	TRN		VAL		TEST[a]	
		R^2	RMSE	R^2	RMSE	R^2	RMSE
PLS	167(2)[b]	0.57	0.43	0.57	0.42	0.55	0.40
GP-Fixed	167	0.77	0.31	0.72	0.34	0.66	0.35
GP-2D	**167**	**0.80**	**0.29**	**0.73**	**0.33**	**0.67**	**0.35**
GP-FVS	47	0.74	0.33	0.68	0.36	0.62	0.37
GP-RFVS	42	0.80	0.29	0.73	0.34	0.63	0.37
GP-Opt	29	0.81	0.28	0.68	0.37	0.65	0.36
GP-Nest	167	0.79	0.30	0.73	0.33	0.64	0.36

[a] Training set 205 compounds, validation set 44 compounds, test set 43 compounds

[b] Number of PLS components in brackets

Aqueous solubility

The automatic model generation process was applied to the aqueous solubility (log$S$) data set described in section "Aqueous solubility model". Due to the large size of this set we have applied only the less computationally demanding modeling techniques. The results are given in Table 4.

The Gaussian Processes technique GP-2D produced the best model for this data set. On the validation set, it achieved $R^2 = 0.85$ and $r^2_{\text{corr}} = 0.85$, on the test set $R^2 = 0.84$ and $r^2_{\text{corr}} = 0.84$; on the combined validation and

Table 4 Modeling aqueous solubility

Method	Desc.	TRN		VAL		TEST[a]	
		R^2	RMSE	R^2	RMSE	R^2	RMSE
PLS	166(7)[b]	0.76	0.92	0.77	0.85	0.79	0.80
GP-Fixed	166	0.88	0.66	0.85	0.68	0.84	0.69
GP-2D	**166**	**0.87**	**0.66**	**0.85**	**0.68**	**0.84**	**0.70**

[a] Training set 2651 compounds, validation set 325 compounds, test set 337 compounds

[b] Number of PLS components in brackets

test sets RMSE = 0.69. Its performance compares well with the manual aqueous solubility model which has achieved $R^2 = 0.82$ and RMSE = 0.79 on the original log*S* 663-compounds test set.

We have tested both of these models on an additional external set of compounds, the Huuskonen 564 set. Table 5 summarizes the performance of the two aqueous solubility models on this set. The automatically built model GP-2D gives much better predictions than the manual model for the Huuskonen 564 set, with $R^2 = 0.82$ and RMSE = 0.96.

Figure 6 shows the graph of predicted log*S* values versus observed for the Huuskonen 564 set. It is apparent from Fig. 6b that the descriptors needed to explain compounds with low log*S* (log*S* < 0) are missing in the manual model, which was built on fewer descriptors (108 descriptors). The compounds with low log*S* are mainly hydrocarbons and were absent in the original aqueous solubility data set (see section "Aqueous solubility model").

The automatic model requires 162 descriptors including calculated log*P*, atom based fragments (e.g. sp3 nitrogen), functionalities (e.g. carboxylic acids) and intramolecular hydrogen bonding as well as more general descriptors such as hydrogen bond donor and acceptor counts and partial charges. The majority of the descriptors are 2D descriptors that if considered on their own would not have the ability to define the molecular charge as a function of p*K*a. However, a combination of these 2D descriptors, as used in the model, is satisfactory to explain the aqueous solubility of potentially charged compounds.

The model was found to be applicable to a variety of acidic, basic, zwitterionic and neutral compounds. Not surprisingly, the majority of the compounds in the sets are neutral (more than 66% in training, validation and test sets and 80% in the Huuskonen set). The remaining compounds represent the acidic, basic and zwitterionic molecules. The performance of the model across the range of the potentially charged compounds was investigated as seen in Table 6. The model gives a reasonable RMSE throughout the classes in all sets.

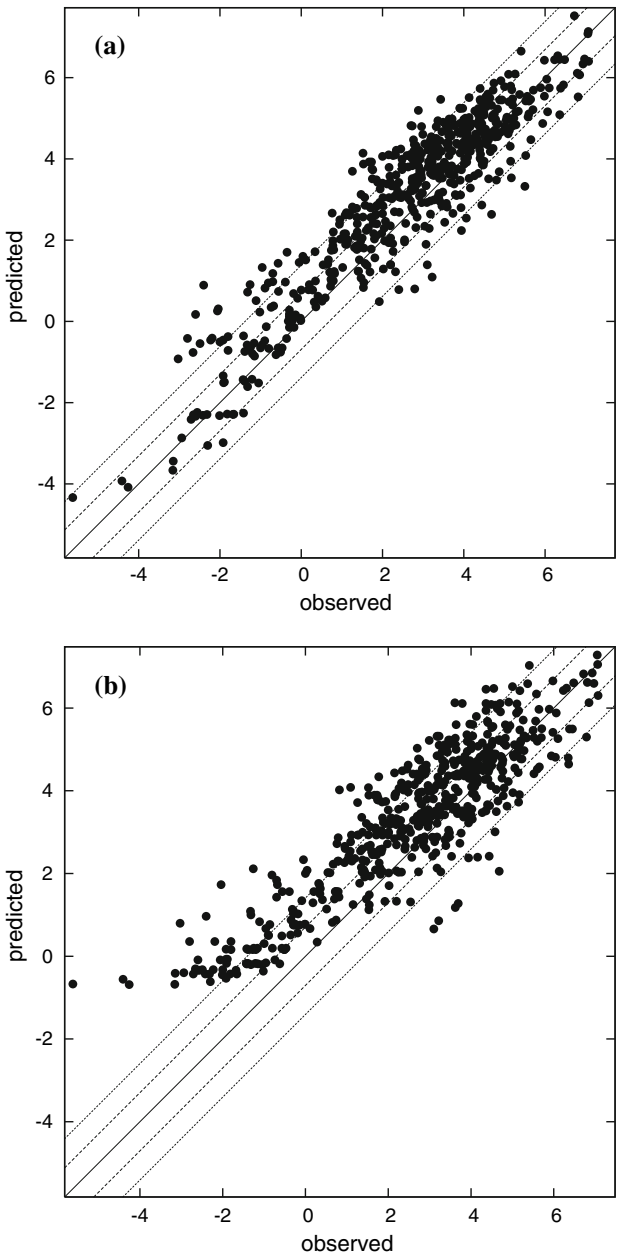

Fig. 6 Predicted log*S* (*S* in μM) values versus observed for the Huuskonen 564 set. Dashed lines represent 0.7 log units interval, dotted lines represent 1.4 log units interval. The line of unity ($y = x$) is shown as a solid line for comparison. (**a**) Prediction by the GP-2D model build by the automatic model generation process. (**b**) Prediction by the manual log*S* model

Table 5 Predictive ability of two aqueous solubility models tested on the Huuskonen 564 set

Model	% Compounds predicted within		R^2	r^2_{corr}	RMSE
	±0.7 log unit	±1.4 log unit			
Manual	39.9	70.9	0.68	0.80	1.28
GP-2D	54.1	85.9	0.82	0.86	0.96

Table 6 Performance of the automatic aqueous solubility model on groups of neutral and potentially charged compounds

Compound type	Training set		Validation set		Test set		Huuskonen set	
	%	RMSE	%	RMSE	%	RMSE	%	RMSE
Overall	–	0.66	–	0.67	–	0.7	–	0.96
Neutral	66	0.63	66	0.66	68	0.62	80	1.01
Acidic	15	0.71	17	0.76	16	0.90	7	0.74
Basic	15	0.72	13	0.63	13	0.83	10	0.63
Zwitterionic	4	0.75	4	0.44	3	0.56	3	0.9

The acidic compound subset of the test set has a high RMSE value of 0.90 log units. A compound in this set, cinnamic acid, was greatly mispredicted with an experimental log$S$ value of -0.48 and a predicted log$S$ value of 3.81. It was found after a similarity search that analogs of this compound have log$S$ values greater than 3 log units. An incorrect experimental value for cinnamic acid could be the reason for the misprediction of this compound and indeed affect the overall RMSE value for this subset.

Conclusions

We have described an automatic model generation process for QSAR modeling and applied it to build blood–brain barrier penetration and aqueous solubility models. These represent two types of data set commonly encountered when building ADME QSAR models, a relatively small set of in vivo data and a large set of physico-chemical data. We have demonstrated that the automatic model generation process can effectively produce models which compare well with models built manually by computational chemists by testing them on external data sets. In the first model, of blood–brain barrier penetration, it can be seen that the automatically built model reports a slightly higher but comparable RMSE to the original manual model. In the second example, modeling aqueous solubility, the automatically built model reports a lower RMSE than the manual model.

We have also used the automatic model generation process to build a new blood–brain barrier penetration model, on combined original logBB data and the Abraham 143 set. It has achieved $R^2 = 0.73$ on the validation set and $R^2 = 0.67$ on the test set and RMSE = 0.34 on combined validation and test set which compares well with existing logBB models. Furthermore this demonstrated that rebuilding a model using a sub-set of a new data set provided a significant improvement in prediction for previously poorly predicted compounds.

We have demonstrated that the performance of the automatic model generation process is robust and comparable to manual model building. Additionally, it is much quicker than manual modeling and can be applied by non-experts.

References

1. Cartmell J, Enoch S, Krstajic D, Leahy DE (2005) J Comput Aid Mol Des 19:821
2. Zhang S, Golbraikh A, Oloff S, Kohn H, Tropsha A (2006) J Chem Inf Model 46:1984
3. Winkler DA, Burden FR (2004) J Mol Graph Model 22:499
4. Tetko IV (2002) J Chem Inf Comput Sci 42:717
5. Burden FR (2001) J Chem Inf Comput Sci 41:830
6. Obrezanova O, Csányi G, Gola JMR, Segall MD (2007) J Chem Inf Model 47:1847
7. Schwaighofer A, Schroeter T, Mika S, Laub J, Laak AT, Sulzle D, Ganzer U, Heinrich N, Muller KR (2007) J Chem Inf Model 47:407
8. Daylight Chemical Information Systems, Inc., SMARTS Tutorial. Retrieved from http://www.daylight.com/dayhtml_tutorials/languages/smarts/index.html 16/10/2007
9. Ertl P, Rhodes B, Selzer P (2000) J Med Chem 43:3714
10. Abraham MH, McGowan JC (1987) Chromatographia 23:243
11. Butina D (1999) J Chem Inf Comput Sci 39:747
12. Livingstone D (1995) Data analysis for chemists. Oxford University Press, Oxford, UK
13. Wold S, Sjöström M, Eriksson L (1998) In: Schleyer PvR, Allinger NL, Clark T, Gasteiger J, Kollman P, Schaefer HF III, Schreiner PR (eds) The encyclopedia of computational chemistry, vol 3. Wiley, Chichester UK, pp 2006–2022
14. Enot D, Gautier R, Le Marouille J (2001) SAR QSAR Environ Res 12:461
15. Tino P, Nabney IT, Williams BS, Losel J, Sun Y (2004) J Chem Inf Comput Sci 44:1647
16. Schroeter T, Schwaighofer A, Mika S, Laak AT, Sulzle D, Ganzer U, Heinrich N, Muller KR (2007) J Comput Aided Mol Des 21:485
17. MacKay DJC (2003) Information theory, inference, and learning algorithms. Cambridge University Press, Cambridge, UK
18. Rasmussen CE, Williams CKI (2006) Gaussian Processes for machine learning. The MIT Press, Cambridge, MA
19. Buhman MD (2003) Radial basis functions: theory and implementations. Cambridge University Press, Cambridge, UK
20. Whitley DC, Ford MG, Livingstone DJ (2000) J Chem Inf Comput Sci 40:1160
21. Clark DE (2005) In: Doherty AM (ed) Annual reports in medicinal chemistry, vol 40. Elsevier Academic Press, San Diego, CA, pp 403–415
22. Butina D, Gola JRM (2003) J Chem Inf Comput Sci 43:837
23. Abraham MH, Ibrahim A, Zhao Y, Acree WE Jr (2006) J Pharm Sci 95:2091
24. Huuskonen J (2000) J Chem Inf Comput Sci 40:773
25. Rose K, Hall LH, Kier LB (2002) J Chem Inf Comput Sci 42:651