smLogP: A free and easily integrated ensemble based logP prediction tool for small molecules.

Daniel Murrell[1] , Ian P Stott[2], Andreas Bender[1]and Robert C Glen*[1]

[1]Unilever Centre for Molecular Science Informatics, Department of Chemistry, University of Cambridge, Lensfield Road, Cambridge CB2 1EW, UK
[2]Unilever Research, Bebington, UK

Email: Daniel Murrell - dsm38@cam.ca.uk; Ian P Stott - Ian.Stott@unilever.com; Andreas Bender - ab454@cam.ac.uk; Robert C Glen - rcg28@cam.ac.uk;

*Corresponding author

Abstract

Multiple predictive models for the octanol/water partition coefficient have been constructed using descriptors computed from open source software (PaDEL-Descriptor) combined with measured logP values from the BioByte Star Set. Two ensemble techniques are used to combine these models and the best ensemble of models (smLogP) uses a weighted combination of support vector regression and generalized boosted regression. When tested on two combined independent external validation sets, this ensemble outperforms the best logP prediction methods available. A novel prediction accuracy estimation technique which reports standard deviations on new predictions using a closest distance to training set approach is presented. A freely available R package (smPredict) combines select molecular descriptors generated using PaDEL-Descriptor, Indigo based standardisation, and smLogP to allow predictions on new molecules. Pipeline-Pilot and KNIME integrations are made available. smPredict can be used standalone, enabling the estimation of logP in secure environments when compounds are subject to IP restrictions.

Introduction

Absorption, Distribution, Metabolism and Elimination (ADME) are of enormous interest to the medicinal chemistry community. Physicochemical properties, measured or predicted, are vital components for the prediction of ADME. This work investigates the prediction of arguably the most important physicochemical property relating to ADME, lipophilicity. The partition coefficient (P), a surrogate measure for both permeability and solubility, is the ratio of the concentrations of a molecule in its neutral state in two immiscible solvents. Due to the large range of values P can take, it is usually referred to and most easily interpreted by using the $\log_{10}$ form,

$$\log P = \log C_{organic} - \log C_{water} \tag{1}$$

where C is the concentration of molecules in the organic or water solvents. 1-octanol is most commonly chosen as the organic solvent as it is believed to relate to cellular membrane like systems and also due to it's chemical stability, commercial availability, non-volatility, low toxicity and very low UV light absorption [1].

A Coarse Review of Computational Approaches to Predict LogP

Hansch and Fujita's π-System

1964 saw the development of the first logP prediction method by the group of Corwin Hansch [2]. They developed a method that was based on fragment substitution where fragments were assigned additive contributions (π values) to logP and, as such, relied on a congeneric series of molecules, *i.e.* molecules that are related to each other by a common scaffold with a fixed set of variable substituents.

Hansch chose the octanol—water system as a standard for their calculations, finding that the π values of the substituents varied depending on the local atomic environment they were placed in.

The main weakness of this approach is that it relies on having a parent structure with a measured logP value. If a prediction is required for a molecule constructed by making multiple substitutions to a precursor molecule with an experimentally measured logP, then errors can accumulate on each substitution and the prediction accuracy can suffer greatly.

Fragment based methods

Fragment based methods divide the molecular graph structure into fragments of connected atoms and use multiple linear regression techniques to find fragment contributions that best fit the experimentally measured data.

Assuming general additivity of fragment contributions to overall molecule lipophilicity, fragments can be assigned values that may be summed as contributions to logP. To enhance prediction, specific intramolecular interactions are taken into account by defining correction factors which also contribute additively to the final logP prediction. LogP calculation using a fragment based method can be summarised using the following equation, where the first term represents the fragment contributions and the second, correction factor contributions:

$$LogP = \sum_{i=1}^{n} a_i f_i + \sum_{j=1}^{m} b_j F_j \tag{2}$$

Here, n is the number of fragments defined for the model, f_i is the contribution of the i_{th} fragment, a_i counts the occurrences of f_i within the molecule, m is the number of correction factors defined for the model, F_j is the j_{th} correction factor and b_j its frequency of occurrence.

The main advantage of using a fragment based approach is that intramolecular interactions can be considered within fragments. Disadvantages include ambiguity in defining fragments and rare fragments in molecules requiring prediction, that did not occur in the training data.

There have been many fragment based methods published, including: Rekker's $\sum F$ System [3], CLogP [4–6], KLogP [7], TLogP [8], AB/LogP [9], KowWIN [10], MolLogP [11], ACD/logP [12] and MiLogP [13].

Atom-based methods

Atom-based methods can be regarded as an extension of fragment-based methods. These methods fragment molecules into single typed atoms and also assume additivity in the contribution of atom types to lipophilicity but, unlike fragment based methods, most do not require correction factors. This is due to the large number of atom types defined according to their structural environment [14]. In atom based schemes, logP can be calculated as follows:

$$logP = \sum_{i=1}^{m} n_i a_i \tag{3}$$

where, m is the total number of atom types, n_i is the number of atoms of type i and a_i is the contribution of atoms of type i to logP. Since no correction factors are needed the complexity of the model is reduced. The major disadvantage is that the additivity assumption is not always valid due to long range interactions [15]. Atom based methods include ALogP [16], ALogP98 [17], XLogP [18] and XLogP3 [19].

Whole molecule approches

Many other new methods for logP prediction have been developed that take aspects of the molecule as a whole into account [15]. These included advanced calculation schemes based on quantum mechanical (QM) and semi-empirical QM calculations, continuum solvation models, molecular dynamic calculations, empirical approaches, and topological descriptor based methods [14].

Of these methods, the use of topological descriptors proves more computationally efficient. Topological descriptors are whole molecule properties calculated from the molecular graph structure, and are thus preserved over the conformational range of the molecule. Methods that use topological descriptors are many orders of magnitude faster than methods that rely on quantum or molecular dynamic calculations.

Recent comparison of existing methods

Mannhold et al. [14] provide an excellent review of existing logP prediction methods. They benchmark methods on a public dataset taken from Avdeef [20] which is composed of 223 molecules that intersect with the BioByte's Star Set (SS), and 43 molecules that were not in the Star Set (NSS). Table 1 shows the published results of their benchmark on some of the methods mentioned here (with our method, smLogP, included for comparison). Mannhold et al. [14] make it evident that the performance ranking of models between the two datasets is not the same and that models that perform well on SS do not necessarily carry that performance over to NSS relative to other models. A good example of this is AB/LogP (not included here), which scores an RMSE of 0.41 on SS (better than all other models), but then has a relatively large RMSE of 1.00 on NSS. Since CLOGP and ALOGP are such popular methods, we decided to compare smLogP against all models that score an RMSE below or equal to 0.92 on NSS (those listed in Table 1). It is worth noting that although the *Consensus* model listed here scores highly, it is a combination of many models including some that make use of QM calculations. Consequently it is difficult to reproduce here given the non-availability of these methods and so is not considered in further comparisons.

Table 1: Benchmark of existing methods from [14], ordered by performance on the set of molecules that do not include Star Set molecules. Root mean squared error (RMSE) is used to compare models.

	RMSE SS (N=223)	RMSE NSS (N=43)
smLogP	0.35	0.78
Consensus logP	0.50	0.80
ALOGPS	0.53	0.82
MiLogP	0.57	0.86
S+logP	0.45	0.87
XLOGP3	0.62	0.89
CLOGP	0.52	0.91
ALOGP	0.69	0.92

Approach Adopted Here

In this work, several regression approaches are investigated for their ability to predict logP using a set of well-curated partitioning measurements: BioByte's logP Star Set [4]. A diverse set of 2D descriptors are calculated using the open source PaDEL-Descriptor software [21] which combines routines from the Chemistry Development Kit [22] together with some additional descriptor calculations including atom type, electro-topological state descriptors, McGowan volume, molecular linear free energy relation descriptors, ring counts, counts of chemical substructures identified by Laggner [23] and those identified by Klekota and Roth [24]. The optimised regression approaches are subjected to two ensemble generation techniques to produce a final model: smLogP. Also, a distance based error variance estimation technique is constructed that allows the end user to reasonably quantify model accuracy on external data.

This work aims to demonstrate the utility of descriptors generated using PaDEL-Descriptor as a viable source of input to the learning approaches for the prediction of logP, as well as provide the community with a reliable and open logP prediction tool that is easy to incorporate in analysis pipelines.

Methods

Training set preparation

Data Source

77,726 LogP measurements along with their molecular SMILES descriptions were obtained from BioByte's THOR database. The *Star Set* [4] of 12,570 molecules with the highest qualitative confidence labels were selected for further standardisation. The full distribution of measured logP values can be seen in figure 2.

Standardisation and filtering

The C API of the Indigo framework [25] was used to convert molecules into a standard representation. The framework was used to perform the following operations (seen in figure 1):

- Molecules were read in either SMILES or SDF format.
- Hydrogens were made implicit.
- Molecules were excluded if they did not pass the Indigo checks for correct valence (*'Bad Valence'*) or ambiguous Hydrogens ('*Ambiguous H*').
- Atomic isotopes were converted to their common forms.
- Molecules were excluded if they contained any atoms not in {H, C, N, O, P, S, F, Cl, Br, I}.

- Indigo's de-aromatise algorithm was invoked, converting molecules stored using aromatic bonds to a kekulised form.

- De-aromatised molecules were converted to their Inchi representation [26] using Indigo's Inchi plugin.

- All Inchi layers below and including the tautomeric layer were removed.

- SMILES were constructed using this shortened Inchi, thus converting multiple tautomeric forms to a single form.

- Molecules were excluded if they had a molecular weight less than 20 Da or greater than 900 Da.

- Molecules were excluded if they contained more than 3 of any one halogen. In particular, the training set contained 239 molecules containing more than three Chlorine atoms, so this operation made the training set more representative of small organic molecules that were likely to be used as drugs, for example.

- Negative charges on oxygens bonded to any of Carbon, Sulphur or Phosphorus were removed (using SMARTS pattern [O-$(*[#6,#14,#15])]), neutralising these negatively charged acidic groups.

- Positive charges on NH_2 groups were removed (using SMARTS pattern [#7+$(*[H])]), neutralising these positively charged nitrogen bases.

- Salts were removed by isolating connected components and removing all but the largest component.

- Duplicates were removed if they had matching canonical SMILES.

- Molecules were written out in SDF format.

Descriptor generation

PaDEL-Descriptor (version 2.16) was used to generate 729 2D molecular descriptors. A configuration file was used with the following options set to true: Compute2D, DetectAromaticity, StandardizeNitro. All other options were set to false. Descriptors based on other logP prediction methods {'ALogp', 'ALogp2', 'XLogP', 'MLogP', 'LipoaffinityIndex', 'CrippenLogP'} were removed so as not to bias cross-validation estimates. Hastie's R imputation package (impute.knn [27]) with default parameters was used to impute descriptors that PaDEL-Descriptor failed to calculate. Missing values of descriptors were replaced by the average value of non-missing descriptors in the 10 nearest neighbours found using Euclidean distance.

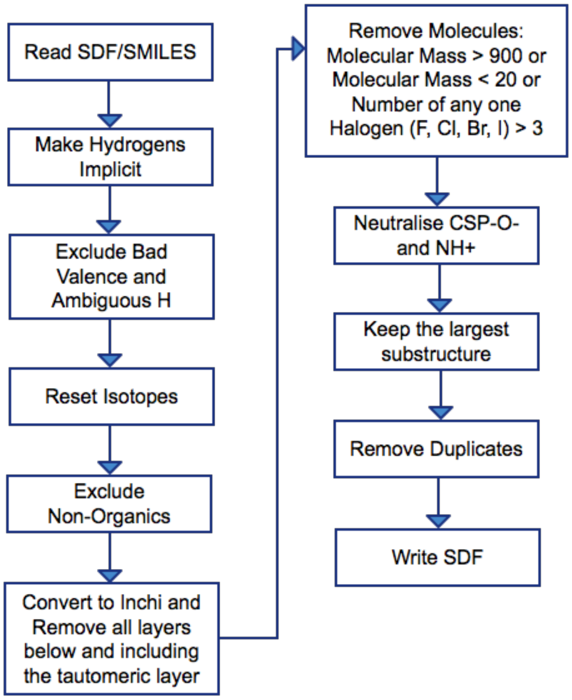

Read SDF/SMILES
Make Hydrogens Implicit
Exclude Bad Valence and Ambiguous H
Reset Isotopes
Exclude Non-Organics
Convert to Inchi and Remove all layers below and including the tautomeric layer
Remove Molecules: Molecular Mass > 900 or Molecular Mass < 20 or Number of any one Halogen (F, Cl, Br, I) > 3
Neutralise CSP-O- and NH+
Keep the largest substructure
Remove Duplicates
Write SDF

Figure 1: Sequence of standardisation and filtering steps undertaken to ensure that molecules presented to the training algorithms were represented similarly.

Model Building

The *caret* package [28] was used to build models in R [29]. *caret* provides a common interface to multiple regression techniques from various other R packages, allowing hyper-parameter tuning through a standard cross validation framework. *caret* also includes methods for pre-processing training data and calculating variable importance.

Data partitioning

Simple random sampling was used to split the data into a training and holdout test set consisting of 4/5 and 1/5 of the data respectively. The holdout set was excluded from training and was used to estimate the model's performance on new data with a similar chemical space distribution to the training set.

Variable selection and data normalisation

Two simple univariate filter based variable selection steps were performed.

Descriptors with entropy close to zero were removed by considering the ratio of the frequency of the most commonly occurring value to the frequency of the second most commonly occurring value. If this ratio was greater than 30:1, the descriptor was removed. Highly correlated descriptors were removed by considering pairs of variables with a Pearson's correlation greater than 0.95 and removing the descriptor with the highest mean correlation to all other descriptors. The remaining descriptors were autoscaled: centred around zero and scaled to have unit variance.

More complex variable selection methods were attempted, but for the non-linear learning methods investigated, there were no notable observed gains in performance. This is believed to be due to the large amount of training data. Forward variable selection was performed by ranking descriptors by their random forest importance score or by different association measures (MIC [30], distance correlation [31], A [32], Pearson's correlation) to the target variable (logP). Models were trained using the most highly ranked descriptors. The number of ranked descriptors used was then increased from a minimum number of 10 until an optimum model was found. Models with a reduced descriptor count showed reduced performance in cross-validation and on the holdout set. Results of these more complex variable selection attempts are not shown here.

Model training

The descriptors that remained after the application of the above two simple univariate filtering steps were used to train a number of learning algorithms ranging from simple linear regression models to more complex non-linear methods. These algorithms include: generalized linear models with a gaussian noise assumption (GLM, included in the R *stats* package), generalized linear models with elasticnet regularisation (GMLNET, R package *glmnet* [33]), generalized boosted regression models a with gaussian noise assumption (GBM, R package *gbm* [34]), K-nearest neighbours (KNN, R package *caret* [28]), partial least squares regression (PLS, R package *pls* [35]), random forests (RF, R package randomForest [36]), support vector regression (SVR, R package *kernlab* [37]), relevance vector regression (RVR, *kernlab*), and gaussian process regression (GPR, *kernlab*). Optimised hyper-parameters for the models were found by minimising the 5 fold cross-validation root mean squared error (RMSE) over a selection of parameters. The folds used in the cross-validation were held constant over all model trainings as this was important in generating ensembles of models later. In the case of algorithms requiring the optimisation of only a single hyper-parameter, the search for an optimum was computationally straightforward requiring the exploration of only a single sequence of parameters. For SVR, the only method requiring a two dimensional parameter search, a manual grid search was performed first across a wide hyper-parameter space and then another grid search was performed within the most promising narrow hyper-parameter space.

The optimal hyper-parameters found for each model in cross-validation were used to train models that used all 5 folds of the training data and these models were used to make predictions on the holdout set and two external test sets. Using the RMSE metric, SVR was found to perform better than all other methods both in cross-validation and on the holdout set.

Ensemble generation

Two ensemble models were created using techniques available through the *caretEnsemble* R package [38]. Both take as input the output of the models created above after hyper-parameter optimisation using all 5 folds of the training data. The first used a greedy algorithm, based on the work of Caruana et al. [39], which tries to optimise the RMSE on the fold out-of-sample predictions by using a weighted ensemble of models. The weights are obtained by starting with a zeroed weight vector, and repeatedly incrementing (by 1) the weight corresponding to a model if the subsequent normalised weight vector allowed a closer match between this weighted combination of model predictions and the measured values. This repetition is carried out 100 times and a the resulting weight vector is normalised to obtain a final weighting. The second is a stacking approach which makes use of a meta model that uses the predictions of the input models as feature inputs. A GLM was built here as the meta model. The stacking approach gave the best results over cross-validation and on the holdout set but the greedy approach performed better on the two external data sets. For this reason, the greedy approach was selected for comparison with existing methods and retrained on all the data (original training set + holdout set). The greedy approach emerged as a linear combination of SVR and GBM algorithms which are described briefly here.

Support vector regression

Support vector regression is an extension of support vector machines which were developed by Vapnik and Cortes [40] to perform non-linear binary classification. SVMs employ a kernel function (ϕ) which casts the original data into a higher dimensional space where the data is separated 'as best as possible' by a hyperplane that maximises the distance from the closest input vectors (support vectors). The most popular and well performing kernel function is the radial basis function which is usually of the form

$K(x, y) = e^{-\sigma||x-y||^2}$ where sigma is a hyper-parameter that defines the width of the function.

Support Vector Regression, instead of using a hyperplane to separate the data in the higher dimensional space, uses ordinary least squares or partial least squares regression in that space to find a linear relationship between the new feature vectors and the target. The cost function that is minimised to obtain the best regression model is made up of two components: a loss function and a smooth regularisation term and the goal is to minimise both the size of the coefficients (smoothness) and the prediction errors (accuracy). Another hyper-parameter requiring adjustment is the error weight C which controls the importance of smoothness vs accuracy. If C is high, accuracy is favoured and smoothness is neglected resulting in a model that will over-fit the training data. To make sure that over-fitting is not occurring, SVR relies upon re-sampling techniques to find the optimal hyper-parameters σ and C.

Vapnik uses an ϵ-insensitive loss functiosn where only data points with predictions which deviate more than ϵ are taken into account. In the SVR context, only these data points are called support vectors and only these data points determine the final training of the SVR model.

Advantages of SVMs and SVR are that, once trained, they are fast at predicting new data points since they only rely on their support vectors. For a more detailed theoretical background on SVMs and SVR, the reader is referred to [41].

Generalized boosted regression

Write up a section about how generalized boosted regression works.

Model Validation

Models and ensembles of models were compared using the RMSE performance metric on the training set, holdout set and in cross-validation. Additional to this, two external test sets were obtained from the literature. The first from the Pharmaco$K$inetics Knowledge Base (PKKB) database [42] consisting of 1012 molecules with logP measurements. The second from a study done by Martel et al. [43], which obtained logP measurements for the specific purpose of benchmarking studies. Martel et al. selected 1000 diverse molecules from the 4.5 million ZINC [44] compounds and from the 759 that were purchasable, 707 yielded validated logP measurements. Molecules from these two datasets were standardised using an identical procedure to that applied in training and those that were identical to any molecules in the training set (including tautomeric matches) were removed before external validation. This ensures that the comparison of models is not biased towards our model. It could be the case however, that molecules in these sets used for comparison were present in the training sets of existing methods which would favour them in the comparison. It is not possible to ascertain if this does occur or with what magnitude so this must be noted into the comparison.

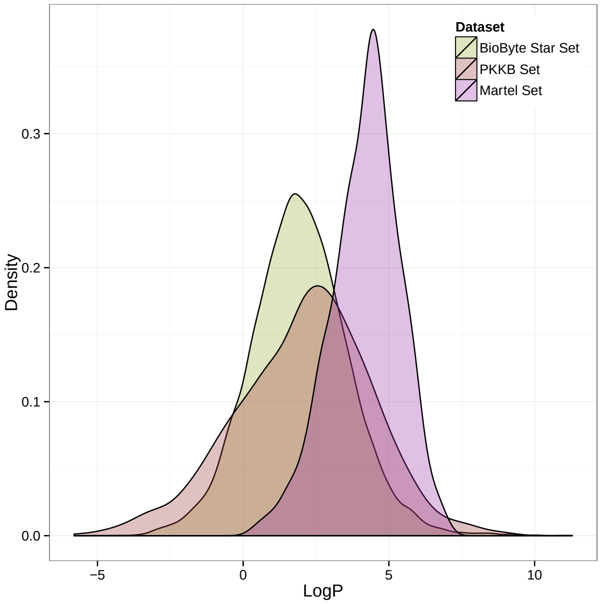


Figure 2: Comparison of the distribution of measuered logP values in the training and external test sets.

The performance of the best model (greedy ensemble: smLogP) was compared to that of the existing prediction tools that proved most accurate ($RMSE < 0.92$) when applied to the NSS molecules used by Mannhold et al. [14]. This includes ALOGPS [45] (version 2.1, uploaded SMILES via the Virtual Computational Chemistry Laboratory at www.vcclab.org/lab/alogps/), MiLogP [13] (SMILES sent to the Molinspiration RESTful Web Service at www.molinspiration.com/services/restfulapi.html), S+logP (ADMET Predictor, v 2.0, Simulations Plus Inc), XLOGP3 [19] (v 3.2.2), CLOGP (v 5.4, BioByte Corp 2013) [4–6], ALOGP and MLOGP [16] (as implemented in Pipeline Pilot v8.0). These are the logP prediction methods that are most accurate on unseen external data. When methods failed to make predictions for molecules, the mean measured logP value of the external set under consideration was used as the prediction. Comparisons were also made to existing logP predictions that emerge as calculated descriptors of PaDEL_Descriptor. These include ALogP, XLogP, MLogP and CrippenLogP.

Estimating Prediction Accuracy

Estimating prediction accuracy in ensembles formed by multiple regression models is an unsolved problem. Nearest neighbour approaches from the literature include (TBD list approaches). In this work a weighted distance nearest neighbours approach is adopted to obtain estimates of the variance in the errors of predictions on new external data points. The approach operates on the errors generated by predictions of smLogP on the PKKB external test set and assumes a Gaussian distribution for the errors with a fixed mean of zero. Only the PKKB set is used here due to a bias that, from this analysis, seems to exist in the Martel set that causes all methods to under predict logP values, invalidating a zero mean assumption. The approach tries to find a relationship between the distance of a molecule in the external test set to the training set and the variance of the errors, σ^2, obtained considering molecules within a moving *distance to training set* window.

For each molecule, e, in the external test set, a weighted Euclidean distance to each molecule, t, in the training set is calculated as:

$$D_{et}^{P} = \sqrt{\sum_{v=1}^{V} A_v^P |e_v - t_v|^2} \tag{4}$$

where e_v is the normalised value of the v_{th} descriptor of e, V is the number of descriptors used by the model, A_v is a measure of association between the v_{th} descriptor and the target value using a sample (size 2000) of molecules from the training set, and P is a scaling power which will allow selection of an optimal association weighting strategy by determining how much the association, A_v, weights the contribution of descriptor v to the distance calculation. The measure of association used here comes from the R package *matie* [32], which improves on existing methods of generalised association such as the recently developed maximal information coefficient (MIC) [30] or distance correlation [31].

For each molecule, e, in the external set, the distance to the training set is defined as:

$$D_e^{NP} = \frac{1}{N} \sum_{t \in \mathcal{N}(e)} D_{et}^{P} \tag{5}$$

the average distance to the closest N neighbours of e, $\mathcal{N}(e)$, in the training set. The flexible parameters in this approach are the number of neighbours, N, and the scaling power, P. An example of the distances generated for $P = 0$ and $N = 1$ is given on the left of figure 3. In this example a Gaussian variance is estimated for 5 windows of width 4 with centers 8, 12, 16, 20 and 24 in the *distance to training set* dimension. The window width was set to 4 based on manual experimentation of different sized windows. If

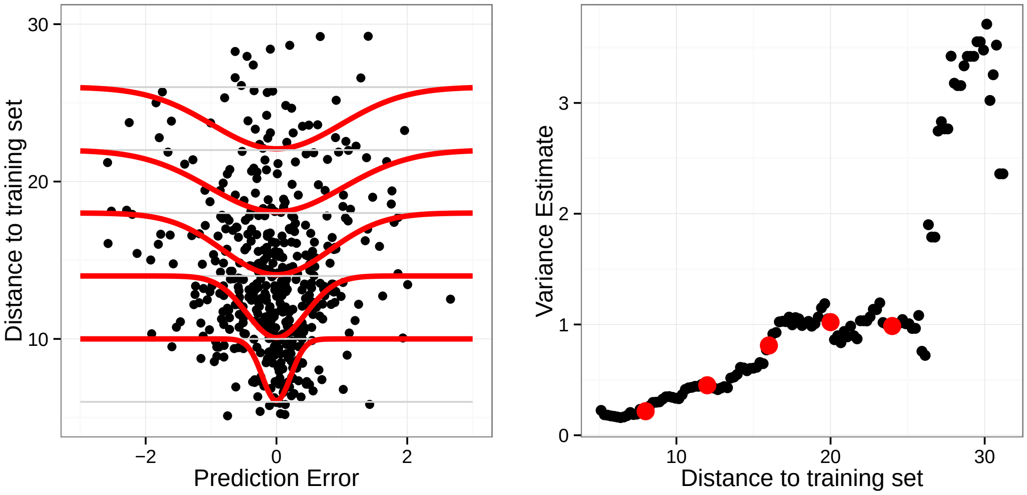


Figure 3: A visual example of how the variances are estimated from the error in predictions for molecules within a given fixed window in the distance dimension. In this example $P = 0$ and $N = 1$. Left: Variances are estimated for windows of width 4 with centers 8, 12, 16, 20 and 24. Right: Black points are variances generated considering windows centered at each molecule's distance from the training set. Red points are show where the Gaussian distributions fitted on the left appear when plotting the variance as a function of distance to the training set.

less than 10 points were found within a window, this window was ignored in the procedure. The variance is estimated by minimising the negative log likelihood of a Gaussian with an assumed zero mean, which conveniently simplifies to:

$$\sigma^2_{ML} = \frac{1}{M} \sum_{m=1}^{M} error^2_m \tag{6}$$

where M is the number of molecules in the distance window under consideration and $error_m$ is the prediction error of the m_{th} molecule.

The red points on the right of figure 3 place the variance estimations of the illustrative windows (mentioned above) indicated between the light grey horizontal lines. The complete relationship between the estimated variance and the *distance to the training set* (black points on the right of figure 3) is found by using 200 equally spaced windows who's centers span D_e^{NP}'s range. This range can vary given different parameters P and N, so, for each choice of these parameters, the distances are scaled to constrain their mean to an arbitrary constant of 15. This value of 15 was selected as the closest integer value to the mean of D_e^{NP} over all molecules in the external test set for $P = 0$ and $N = 1$ after setting the window width to 4.

A desirable property of the relationship between D_e^{NP} and σ^2, is that the variance should increase monotonically with increasing distance from the training set. To find optimal values, P and N were varied and the relationship between D_e^{NP} and σ^2 was scored using Spearman's rank correlation, which assesses how well the relationship between two variables can be described using a monotonic function. The relationship between D_e^{NP} and σ^2 that scored the highest rank correlation was fitted by a spline so that the estimated variance in the prediction, σ^2, can be read off given D_e^{NP} for any new external data point.

Results and Discussion

Data preparation

Out of the initial 12,570 molecules used for training, the filtering procedure removed 124 inorganic molecules. 2 with molecular weight below 20 Da, 13 with molecular weight above 900 Da. 239 with more than 3 Chlorines, 8 with more than 3 Bromines, 71 with more than 3 Fluorines and 15 duplicates leaving 12098 molecule-target pairs for training. PaDEL-Descriptor took 6 minutes and 16 seconds to calculate 729 descriptors for each of the 12098 remaining molecules when run over 14 Intel Xeon E5520 2.27GHz cores. Simple random sampling was performed, dividing the initial set into 9678 molecules for the training set and 2420 molecules for the holdout set.

6 of the logP related descriptors were removed, 362 descriptors were removed due to low variance and 127 descriptors were removed because of their high correlation to other descriptors. The remaining 234 descriptors were used to train the models. These descriptors are listed in SI table 1.

Descriptor relevance

To learn about the relationship between the descriptors and their relative contribution towards the prediction of logP, a descriptor importance ranking was obtained using the RF's permutation importance method. For each descriptor, this method measures the decrease in mean squared error (MSE) averaged over all trees on their out-of-bag sample after permuting the values for that descriptor. These differences are normalised by the standard error. Importance rankings are given in table 2.

Learning curves

The effect of training set size on model performance was investigated by randomly selecting training data from the training set (figure 4). This was done in steps of 1000 molecules. At each step the random subset was split into a training and holdout set consisting of 4/5 and 1/5 of the data respectively. Gaussian process regression was trained at each step as this requires only a single hyper-parameter to be tuned. 5

Table 2: The 10 most significant descriptors in order of importance.

Descriptor	Description	Importance
MLFER_S	Combined dipolarity/polarizability	41.24
SHBd	Sum of E-States for (strong) Hydrogen Bond donors	39.89
MLFER_BH	Overall or summation solute hydrogen bond basicity	34.24
minsCH3	Minimum atom-type E-State: -CH3	30.89
SsCH3	Sum of atom-type E-State: -CH3	28.01
BCUTw-1h	nlow highest atom weighted BCUTS	27.76
ATSc3	ATS autocorrelation descriptor, weighted by charges	27.22
Kier3	Third kappa (κ) shape index	26.73
ATSm1	ATS autocorrelation descriptor, weighted by scaled atomic mass	25.81

fold cross validation was performed on the training set and the model with the lowest cross-validated RMSE was used to make predictions on the holdout set. There was a clear trend of increasing model performance with increasing training set size. Compared with the performance trend using cross-validation, the test set performance exhibits more random fluctuation since it is not averaged over the 5 cross validation folds.

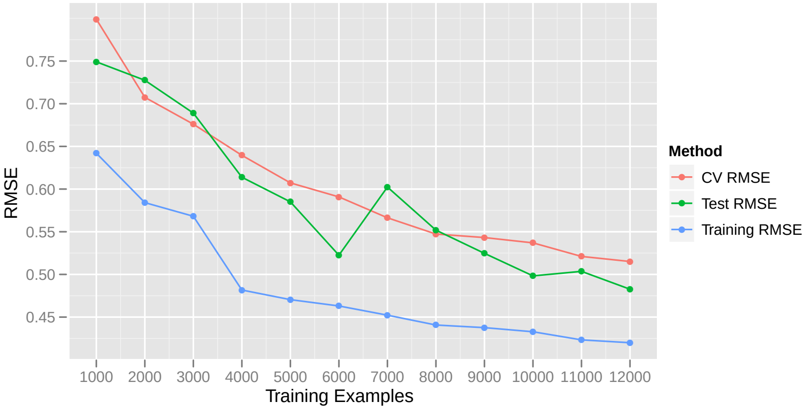


Figure 4: Gaussian process regression accuracy increases as the size of random subsets increase. RMSE is used as the performance metric to compare the performance between the model on the training set (4/5 of the random subset), holdout set (1/5 of the random subset) and in 5 fold cross-validation (CV) over the training set.

Comparison of regression and ensemble methods

GLM, GLMNET, GBM, KNN, PLS, RF, SVR, RVR, GPR and the greedy and stacking ensemble approaches were compared by evaluating their performance in cross-validation, on the training set, on the holdout set, and on the two external test sets. The results of this comparison can be seen in table 3.

RMSE was used as a performance metric as it more heavily punishes predictions with larger errors and also allows the best model to be loosely (considering sample size) compared with the performance of other models given in table 1 on the NSS molecules, none of which appeared in the training set used here.

The final weighted combination of models used to build the greedy ensemble was found using SVR, GBM and KNN with respective weights of 0.711, 0.288 and 0.001. Considering the minute contribution of the KNN's contribution and it's relatively long prediction times, the greedy ensemble's formulation procedure was rerun excluding the KNN to arrive at a final weighted contribution that uses only two models, SVR and GBM with respective weights of 0.712 and 0.288.

As a single method, SVR was found to outperform all other learning algorithms in cross-validation and on the holdout set. GBM comes in third after RVR but still contributes to the greedy ensemble since the predictions it makes are, on average, more different from the SVR prediction than the RVR predictions are. SI Figure 2 shows the final narrow hyper-parameter search space for SVR training. When averaged over the 5 held out test folds, SVR reaches an optimum cross-validation RMSE when σ is 0.0015 and Cost is 30. At this resolution the RMSE basin is quite flat and increasing grid resolution would not achieve worthwhile gains in RMSE when compared with model training time. Finding the optimum SVR hyper-parameters required a 2D grid search over a large log scaled hyper-parameter space (not shown) and then a subsequent 2D grid search over the most promising subspace (shown in SI figure 2).

Comparing ensemble methods, the stacking method outperformed the greedy method in cross validation as well as on the holdout test set but the greedy method was more accurate on both the external data sets. Since the stacking method uses all models as input, this makes it prohibitively large and also time consuming in prediction. smLogP was constructed to make use of the greedy ensemble and is a linear combination of only SVR and GBM prediction contributions.

Table 3: RMSE of trained learning algorithms.

Model	Training	Cross Validation	Holdout Test	PKKB External	Martel External
GLM	0.557	0.617	0.584	1.075	1.212
GLMNET	0.602	0.644	0.614	0.856	1.243
GBM	0.162	0.433	0.427	0.844	1.278
KNN	0.537	0.710	0.680	1.258	1.548
PLS	0.588	0.605	0.601	0.875	1.247
RF	0.211	0.549	0.524	0.943	1.321
SVR	0.187	0.389	0.383	0.921	1.242
RVR	0.287	0.421	0.414	0.931	1.334
GPR	0.405	0.513	0.476	1.129	1.364
Greedy	0.164	0.374	0.372	0.851	1.217
Stacking	0.143	0.369	0.368	0.866	1.229

Model validation

The performance of the SVR and GBM models on the holdout set in relation to their performance on the training set prove that these models are not over-fitting the training data. Chance correlations are highly unlikely to occur due to the large number of training points (9678 molecules) and the comparatively small number of descriptors used in model building (234 descriptors). smLogP, which is a linear combination of these two model's outputs will should also not over-fit the training data.

Further validation of the model was attained by invoking the use of two external datasets for comparison of smLogP to the existing models with top performance on the NSS.

PKKB Dataset

Of the 1685 molecules acquired from the PKKB database, 1666 were compatible with Indigo's SDF import module. 654 were missing a measured logP value and 12 duplicates existed, leaving 1000 unique molecules with measured logP values. Using the same standardisation procedure applied in training, a further 4 molecules were removed as tautomeric duplicates of other molecules in the PKKB set and a further 482 molecules were removed as they were duplicates of molecules in the training set. 9 molecules were removed because their molecular weight was greater than 900 Da and so were considered to be out of the scope of the model (not small molecules). No constraints on the number of halogens were used. Visual inspection of the molecules with large errors (figure 5) revealed 13 molecules containing tetravalent nitrogens. Since it is not possible to measure the logP of these molecules in their neutral state, they can be considered as outside the scope of the model and were thus removed from the comparison. Two of these molecules with large prediction errors (Diltiazem and Verapamil) were found to be represented by an incorrect structure in the PKKB database. When their structures were corrected they were found to be duplicates of molecules in the training set with smLogP predictions almost exactly match the measured value given in the PKKB database. Being duplicates, these two molecules were also removed from the comparison set leaving 490 molecules. smLogP made predictions with an RMSE of 0.859 on this set.

Martel Dataset

706 molecules with measured logP values were read in SMILES format by Indigo's import module. None of the molecules in this set were found as duplicates of molecules in the training set or of other molecules in the same set and none had a molecular weight above 900 Da. Thus, all 706 were used as an external test set. Visual inspection of the molecules with large errors revealed nothing obvious about the structure of

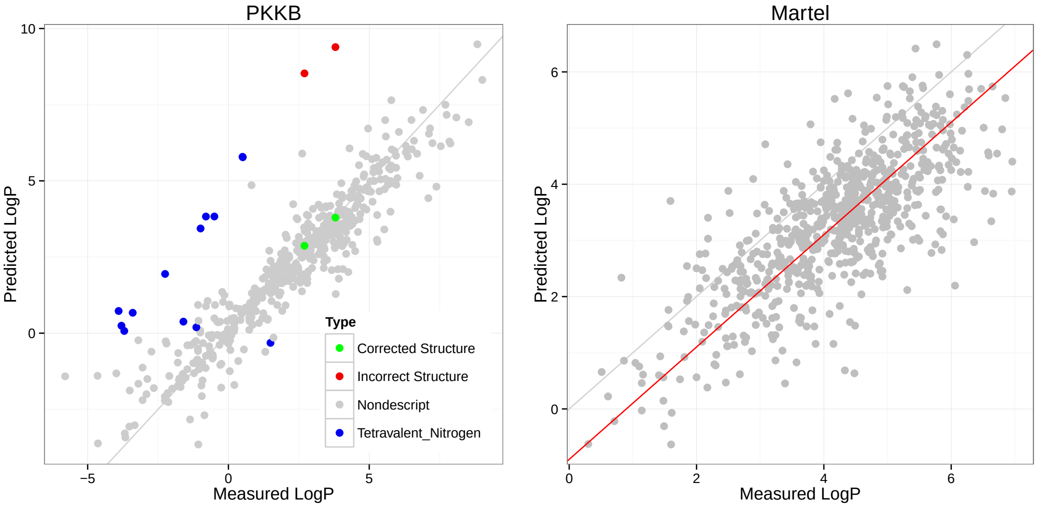


Figure 5: Correlation plots showing the error in predictions on the PKKB and Martel external test sets. In the PKKB set: the logP of the 13 molecules containing tetravalent nitrogens (blue) are, with one exception, overpredicted. The prediction of two molecules in the dataset with incorrect structure (red) is corrected when the correct structures are used (green). Predictions in the lower extremities of logP (below -2.5) are more prone to error. In the Martel set: smLogP underpredicts the set as a whole. The red line is constrained to have slope 1 and is fitted to the data resulting in a y-intercept of -0.897.

the molecules that could cause bad predictions. As can be seen in figure 5, smLogP noticeably under predict logP in the Martel set. All the prediction methods, in fact, under predict logP in the Martel set: an outcome that might be expected if they were trained on a similar set of training molecules as those used here or if the distribution of logP values in their training sets are also much lower that that in the Martel set. In the case of smLogP, this could be due to the mean of the logP values being around 4.5 log units in the Martel set which is much higher than the mean value of the measured logP values used in training. Predictions made could be biased by the mean logP value in the training data which can be seen in figure 2. To test this hypothesis, a greedy ensemble was trained with only those molecules from the original training set with logP values higher than the mean. This cuts the training set in half so one might expect worse performance but the RMSE is reduced from 1.222 log units to 1.033 log units showing how the mean of the training set can indeed influence predictions in an external set with a narrow distribution of logP values and a much higher mean.

Comparison with existing methods

Existing logP prediction methods were compared against smLogP by assessing their accuracy on the two external datasets. This comparison can be seen in figure 6. Methods prepended by 'p.' are implementations of the PaDEL-Descriptor package. The *am* entry is a baseline model that gives a prediction equal to the mean measured logP value in a particular set for all molecules.

RMSEs are shown for all methods on both external data sets. smLogP achieves the lowest RMSE (0.859 log units) on the PKKB set and has an RMSE (1.222 log units) only 0.05 log units higher than the most accurate method on the Martel set. Since there are an unequal number of molecules in the two external sets, a fair comparison between methods is obtained by ordering the average of the RMSE over both external sets weighting the sets equally in the comparison. smLogP is ranked best under the average RMSE.

Due to the narrow range of logP values in the Martel external set, the *am* baseline does surprisingly well when predicting the Martel set with an RMSE of only 0.015 greater than the best method on that set (better than smLogP and most other models).

ALogP as calculated by PaDEL-Descriptor is a surprisingly bad predictor of LogP and far worse than ALogP as calculated by Pipeline Pilot (v8.0). From this, it can be seen that the same method, implemented by different software can differ wildy and care should be taken when selecting software not only on which methodology to use. It is possible that lessons from this observation might extend into the areas of other molecular properties.

As mentioned in the methods section, it is not possible to know if or how many molecules from either of these external sets were used in the training sets of the models that are being compared here. Any inclusion of external set molecules in the training sets of other methods should greatly improve their perceived performance in this comparison. This stems from the fact that models built here always score a far lower RMSE when assessed on molecules from the training set compared to molecules in external test sets.

Estimating Prediction Accuracy

The association power P and the number of neighbours, N, in D_e^{NP} were varied through the values {0, 0.25, 0.50, 0.75, 1, 1.25, 1.5, 1.75, 2, 2.25, 2.5, 2.75, 3} and {1, 2, 3, 4, 5} respectively. All Spearman rank correlations obtained are high, since, given any D_e^{NP}, the variance in the error is generally monotonically increasing with increased distance to the training set. However, there are combinations of P and N which result in elevated rank correlations. Specifically, there is a dense region of elevated rank correlations around a maximum at $P = 1.5$ and $N = 5$. For this combination, smLogP makes use of a spline fitted to the relationship between D_e^{NP} and σ^2 to estimate the error for a given logP prediction of an unseen molecule. It is interesting to note that if descriptor association with the target value is not used ($P = 0$), then the

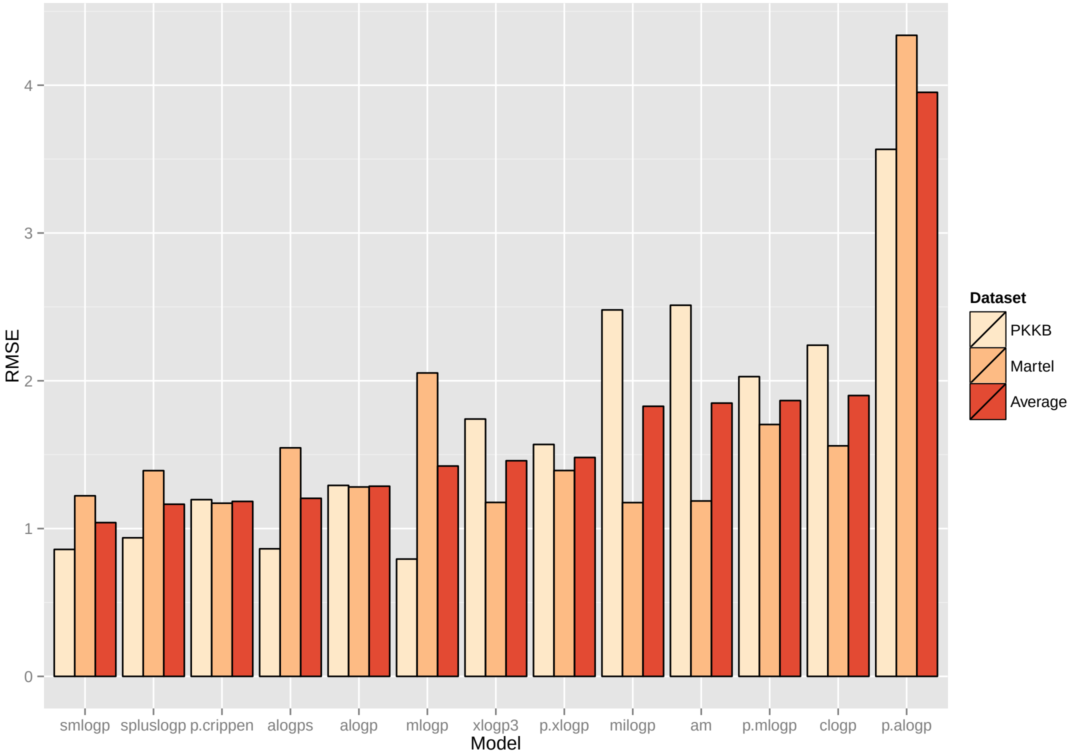


Figure 6: Methods are compared by their RMSE on the two external test sets. The methods are placed in ascending order from left to right by the average RMSE over both external test sets. This weights the external test sets equally in the comparison as they contain an unequal number of molecules.

monotonic relationship between D_e^{NP} and σ^2 is weak. This lends strength to the concept of weighting descriptors when using distance to training set based formulations for error variance estimation. If the association is weighted too strongly ($P = 3$), Spearman's correlation is also low, indicating that it is important that not only the most associated descriptors to logP are used in weighting the distance. The rank correlation also depends on the number of neighbours used to average the distance over. Loosely, if less than three or more than six neighbours are used in the average, rank correlations are low. For this data there seems to be a sweet spot around N=4 or N=5 for the number of neighbours to use. Other applicability domain techniques that use distance to the training set have trouble deciding how many neighbours to average over [?]. This approach provides a principled way to decide the correct number of nearest neighbours to average over when finding a distance to the training set and could be applied to other data sets.

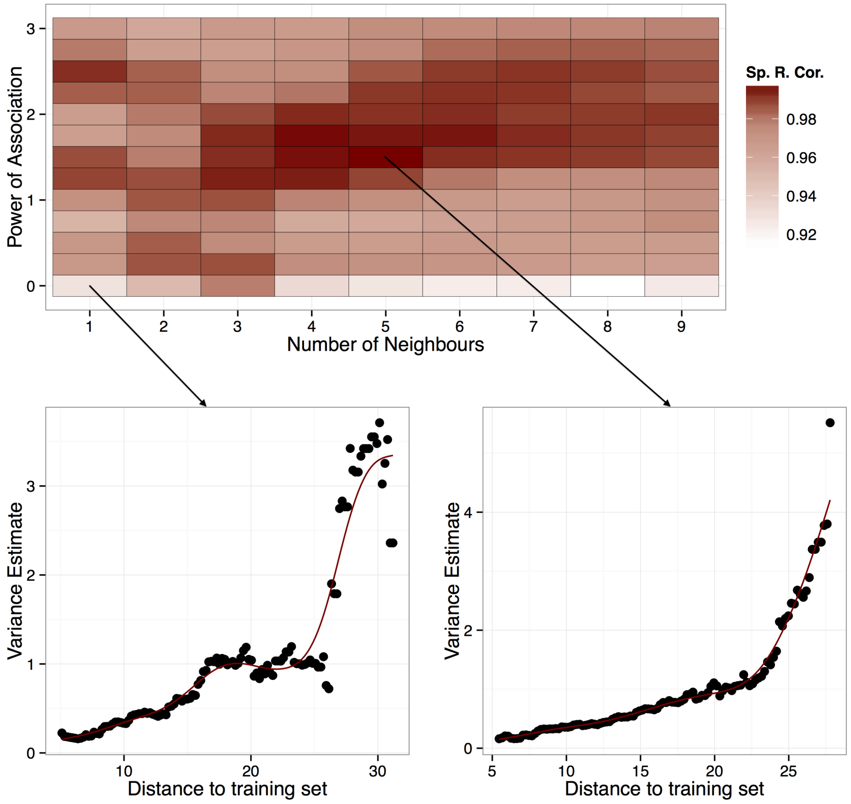


Figure 7: Top: Heatmap of Spearman rank correlations of D_e^{NP} vs σ^2. Bottom left: D_e^{NP} vs σ^2 when $P = 0$ and $N = 1$. This is what might be naively used when finding a distance to the training set. The spline does not fit this relationship well. Bottom right: D_e^{NP} vs σ^2 when $P = 1.5$ and $N = 5$. This combination of parameters is the maximum rank correlation and falls within a dense region of high rank correlations. The spline more accurately fits this relationship.

Package and Use

An R package, *smPredict* (Small Molecule Predict), that predicts the properties of small molecules is available on CRAN (http://cran.....). Currently, *smPredict* contains only smLogP. Other small molecule property prediction methods will soon be added to the package.

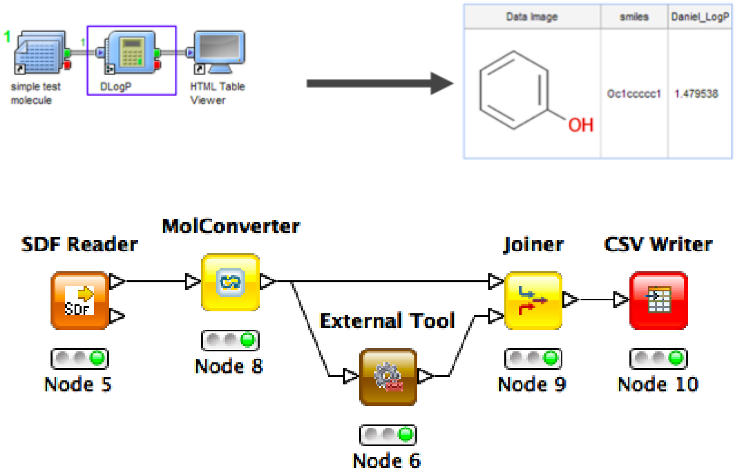
The package relies on running PaDEL-Descriptor which makes it dependant on the rJava package [46] which in turn requires that a Java runtime environment is available. Additional R package requirements include: *caret*, *impute*, *kernlab* and *gbm*.

The main function of the package, *PredictLogP*, takes as input a SMILES or SDF file of molecules and, as default, returns an R data.frame with two columns: molecule ID and logP prediction. A measure of the confidence on each prediction can be requested, in which case the function will add a column containing the estimated variance in error assuming a Gaussian distribution as well as a 95% confidence interval of the prediction. Other % confidence intervals can be requested.

Auxiliary functions include options for calculating and saving the descriptors of molecules to file as well as options to apply the standardisation technique that was used in training to a set of molecules.

100 molecules were subset from the PKKB dataset as a performance test. 43.4 seconds elapsed while predicting logP values using a single core on a 1.7 GHz Intel Core i5 Macbook Air.

The webpage *smlogp.smpredict.com* is available for the upload of either SMILES or SDF files with logP predictions returned via email. Pipeline Pilot and KNIME nodes are also available from this location for standalone use.



Data Image	smiles	Daniel_LogP
OH	Oc1ccccc1	1.479538

Figure 8: Top: Pipeline pilot workflow. Prediction can be incorporated into a single new node. Bottom: Prediction in KNIME is done through a pre-existing external tool node.

Conclusions

The ensemble model, smLogP, that is a linear combination of the outputs of a support vector regression model and a generalized boosted regression model was validated against existing logP prediction methods on two external data sets. smLogP outperformed all existing methods when considering the average RMSE over the two external sets. smLogP was outperformed by MLOGP from Pipeline Pilot over the PKKB set but MLOGP performs very poorly over the Martel set indicating that the performance of MLOGP on the PKKB set *could be* due to inclusion of PKKB molecules in the training set of MLOGP. smLogP was outperformed by CrippenLogP and XLOGP3 over the Martel set but both these models perform substantially worse than smLogP on the PKKB set.

The authors express some concerns about the Martel set as a benchmark dataset for models because it has a high mean logP and narrow logP distribution and, considering the performance of existing models on this set, the existing measured values in the public domain are distributed over a logP range that is far lower than that of the Martel set, leading to the problem of underestimation as seen in this work.

The novel prediction accuracy estimation approach used here suggests that, at least for this data, the number of neighbours used when calculating the distance of external molecules to the training set should be greater than one. In this case, an optimal number of 5 nearest neighbours was found to ensure that large distances between new molecules and the training set maximally correspond to large variance in prediction error. Also, weighting the descriptors used in the distance calculation by association to the target value was found to be important for the monotonicity the relationship between prediction error variance and distance to training set. The extent of this weighting, as modified by the power of the association, also plays a role, and an optimal power of association of 1.5 was found in this work.

An R package was created which allows free and easy access to the ensemble that predicts logP and the estimated variance in error. The model is also accessible as an online service and Pipeline Pilot or KNIME nodes can be downloaded for insertion into pipelining frameworks from smlogp.smpredict.com.

Author's contributions

DM, IPS, AB, RCG: Conceived of the research.

DM, AB, RCG: Wrote the manuscript.

DM: Trained models and created the R package *smPredict*.

IPS: Created the Pipeline Pilot component.

Acknowledgements

The authors would like to thank Unilever for funding, Leo Van Buuren of the Unilever Research Development team at Vlaardingen for useful discussions on the development of this work, Mikhail Rybalkin and Mikhail Kvyatkovskiy for their help with integrating the Indigo framework, and Yap Chun Wei for many conversations related to the PaDEL-Descriptor software.

References

1. Smith RN, Hansch C, Ames MM: **Selection of a reference partitioning system for drug design work**. *Journal of Pharmaceutical Sciences* 1975, **64**(4):599–606.
2. Fujita T, Iwasa J, Hansch C: **A New Substituent Constant, \pi, Derived from Partition Coefficients**. *Journal of the American Chemical Society* 1964, **86**:5175–5180.
3. Rekker RF, Mannhold R: *Calculation of Drug Lipophilicity: The Hydrophobic Fragmental Constant Approach.* Wiley-Blackwell, 1 edition 1992.
4. Leo A, Hansch C, Elkins D: **Partition coefficients and their uses**. *Chemical Reviews* 1971, **71**(6):525–616, [http://dx.doi.org/10.1021/cr60274a001].
5. Hansch C, Leo A: *Substituent constants for correlation analysis in chemistry and biology.* New York: Wiley 1979.
6. Leo AJ: **Calculating log Poct from structures**. *Chemical Reviews* 1993, **93**(4):1281–1306, [http://dx.doi.org/10.1021/cr00020a001].
7. Klopman G, Li JY, Wang S, Dimayuga M: **Computer Automated log P Calculations Based on an Extended Group Contribution Approach**. *Journal of Chemical Information and Computer Sciences* 1994, **34**(4):752–781.
8. Junghans M, Pretsch E: **Estimation of partition coefficients of organic compounds: local database modeling with uniform-length structure descriptors**. *Fresenius' Journal of Analytical Chemistry* 1997, **359**:88–92.
9. Japertas P, Didziapetris R, Petrauskas A: **Fragmental Methods in the Design of New Compounds. Applications of The Advanced Algorithm Builder**. *Quantitative Structure-Activity Relationships* 2002, **21**:23–37, [http://onlinelibrary.wiley.com/doi/10.1002/1521-3838(200205)21:1⟨23::AID-QSAR23⟩3.0.CO;2-E/abstract].
10. Meylan WM, Howard PH: **Estimating log P with atom/fragments and water solubility with log P**. *Perspectives in Drug Discovery and Design* 2000, **19**:67–84.
11. **Molsoft L.L.C.: Drug-Likeness and molecular property prediction.** 2013, [http://molsoft.com/mprop/].
12. Petrauskas AA, Kolovanov EA: **ACD/Log P method description**. *Perspectives in Drug Discovery and Design* 2000, **19**:99–116, [http://link.springer.com/article/10.1023/A%3A1008719622770].
13. **Molinspiratoin, Interactive logP calculator.** 2013, [http://www.molinspiration.com/services/logp.html].
14. Mannhold R, Poda GI, Ostermann C, Tetko IV: **Calculation of molecular lipophilicity: State-of-the-art and comparison of log P methods on more than 96,000 compounds**. *Journal of Pharmaceutical Sciences* 2009, **98**(3):861–893, [http://onlinelibrary.wiley.com/doi/10.1002/jps.21494/abstract].
15. Buchwald P, Bodor N: **Octanol-water partition: searching for predictive models**. *Current medicinal chemistry* 1998, **5**(5):353–380. [PMID: 9756979].
16. Viswanadhan VN, Ghose AK, Revankar GR, Robins RK: **Atomic physicochemical parameters for three dimensional structure directed quantitative structure-activity relationships. 4. Additional parameters for hydrophobic and dispersive interactions and their application for an automated superposition of certain naturally occurring nucleoside antibiotics**. *Journal of Chemical Information and Computer Sciences* 1989, **29**(3):163–172, [http://dx.doi.org/10.1021/ci00063a006].
17. Wildman SA, Crippen GM: **Prediction of Physicochemical Parameters by Atomic Contributions**. *Journal of Chemical Information and Computer Sciences* 1999, **39**(5):868–873, [http://dx.doi.org/10.1021/ci990307l].
18. Wang R, Fu Y, Lai L: **A New Atom-Additive Method for Calculating Partition Coefficients**. *Journal of Chemical Information and Computer Sciences* 1997, **37**(3):615–621, [http://dx.doi.org/10.1021/ci960169p].
19. Cheng T, Zhao Y, Li X, Lin F, Xu Y, Zhang X, Li Y, Wang R, Lai L: **Computation of OctanolWater Partition Coefficients by Guiding an Additive Model with Knowledge**. *Journal of Chemical Information and Modeling* 2007, **47**(6):2140–2148, [http://dx.doi.org/10.1021/ci700257y].

20. Avdeef A: *Absorption and Drug Development: Solubility, Permeability, and Charge State*. John Wiley & Sons 2012.

21. Yap CW: **PaDEL-descriptor: An open source software to calculate molecular descriptors and fingerprints**. *Journal of Computational Chemistry* 2011, **32**(7):1466–1474, [http://onlinelibrary.wiley.com/doi/10.1002/jcc.21707/abstract].

22. Steinbeck C, Han Y, Kuhn S, Horlacher O, Luttmann E, Willighagen E: **The Chemistry Development Kit (CDK): An Open-Source Java Library for Chemo- and Bioinformatics**. *Journal of Chemical Information and Computer Sciences* 2003, **43**(2):493–500, [http://dx.doi.org/10.1021/ci025584y].

23. Laggner C: *SMARTS Patterns for Functional Group Classification*. [Available at http://code.google.com/p/semanticchemistry/ source/browse/wiki/InteLigand.wiki?spec5svn41r541.].

24. Klekota J, Roth FP: **Chemical substructures that enrich for biological activity**. *Bioinformatics* 2008, **24**(21):2518–2525, [http://bioinformatics.oxfordjournals.org/content/24/21/2518]. [PMID: 18784118].

25. **GGA Software Services – Indigo Toolkit**[http://ggasoftware.com/opensource/indigo].

26. **IUPAC - International Union of Pure and Applied Chemistry: The IUPAC International Chemical Identifier (InChI)**[http://www.iupac.org/home/publications/e-resources/inchi.html].

27. **Imputing Missing Data for Gene Expression Arrays, Stanford University Statistics Department Technical report - Google Scholar**[http://scholar.google.co.uk/scholar?q=Imputing+Missing+Data+for+Gene+Expression+Arrays%2C+Stanford+University+Statistics+Department+Technical+report&btnG=&hl=en&as_sdt=0%2C5].

28. Kuhn M: **Building Predictive Models in R Using the caret Package**. *Journal of Statistical Software* 2008, **28**(5):1–26, [http://www.jstatsoft.org/v28/i05].

29. R Development Core Team: *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria 2012, [http://www.R-project.org/]. [ISBN 3-900051-07-0].

30. Reshef DN, Reshef YA, Finucane HK, Grossman SR, McVean G, Turnbaugh PJ, Lander ES, Mitzenmacher M, Sabeti PC: **Detecting Novel Associations in Large Data Sets**. *Science* 2011, **334**(6062):1518–1524, [http://www.sciencemag.org/content/334/6062/1518]. [PMID: 22174245].

31. Székely GJ, Rizzo ML, Bakirov NK: **Measuring and testing dependence by correlation of distances**. *The Annals of Statistics* 2007, **35**(6):2769–2794, [http://projecteuclid.org/euclid.aos/1201012979]. [Mathematical Reviews number (MathSciNet): MR2382665; Zentralblatt MATH identifier: 1129.62059].

32. Murrell B, Murrell D, Murrell H: **Discovering general multidimensional associations** 2013. [ArXiv:1303.1828 [stat.AP]].

33. Friedman J, Hastie T, Tibshirani R: **Regularization Paths for Generalized Linear Models via Coordinate Descent**. *Journal of statistical software* 2010, **33**:1–22, [http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2929880/]. [PMID: 20808728 PMCID: PMC2929880].

34. Ridgeway G: *gbm: Generalized Boosted Regression Models* 2013, [http://CRAN.R-project.org/package=gbm]. [R package version 2.0-8].

35. Mevik BH, Wehrens R, Liland KH: *pls: Partial Least Squares and Principal Component regression* 2011, [http://CRAN.R-project.org/package=pls]. [R package version 2.3-0].

36. Liaw A, Wiener M: **Classification and Regression by randomForest**. *R News* 2002, **2**(3):18–22, [http://CRAN.R-project.org/doc/Rnews/].

37. Karatzoglou A, Smola A, Hornik K, Zeileis A: **kernlab – An S4 Package for Kernel Methods in R**. *Journal of Statistical Software* 2004, **11**(9):1–20, [http://www.jstatsoft.org/v11/i09/].

38. Mayer Z: *caretEnsemble: Framework for combining caret models into ensembles* 2013. [R package version 1.0].

39. Caruana R, Niculescu-Mizil A, Crew G, Ksikes A: **Ensemble selection from libraries of models**. In *Proceedings of the twenty-first international conference on Machine learning*, ICML '04, New York, NY, USA: ACM 2004:18–, [http://doi.acm.org/10.1145/1015330.1015432].

40. Cortes C, Vapnik V: **Support-vector networks**. *Machine Learning* 1995, **20**(3):273–297, [http://link.springer.com/article/10.1007/BF00994018].

41. Cristianini N, Shawe-Taylor J: *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press 2000.

42. Cao D, Wang J, Zhou R, Li Y, Yu H, Hou T: **ADMET Evaluation in Drug Discovery. 11. PharmacoKinetics Knowledge Base (PKKB): A Comprehensive Database of Pharmacokinetic and Toxic Properties for Drugs**. *Journal of Chemical Information and Modeling* 2012, **52**(5):1132–1137, [http://dx.doi.org/10.1021/ci300112j].

43. Martel S, Gillerat F, Carosati E, Maiarelli D, Tetko IV, Mannhold R, Carrupt PA: **Large, chemically diverse dataset of logP measurements for benchmarking studies**. *European Journal of Pharmaceutical Sciences* 2013, **48**(1–2):21–29, [http://www.sciencedirect.com/science/article/pii/S0928098712004198].

44. Irwin JJ, Shoichet BK: **ZINC - A Free Database of Commercially Available Compounds for Virtual Screening**. *Journal of chemical information and modeling* 2005, **45**:177–182, [http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1360656/]. [PMID: 15667143 PMCID: PMC1360656].

45. Tetko IV, Bruneau P: **Application of ALOGPS to predict 1-octanol/water distribution coefficients, logP, and logD, of AstraZeneca in-house database**. *Journal of Pharmaceutical Sciences* 2004, **93**(12):3103–3110, [http://onlinelibrary.wiley.com/doi/10.1002/jps.20217/abstract].

46. Urbanek S: *rJava: Low-level R to Java interfaces* 2013, [http://CRAN.R-project.org/package=rJava]. [R package version 0.9-4].