articles

molecular pharmaceutics

Machine Learning Models for Lipophilicity and Their Domain of Applicability

Timon Schroeter,*[,†,‡] Anton Schwaighofer,[‡] Sebastian Mika,[§] Antonius Ter Laak,[||] Detlev Suelzle,[||] Ursula Ganzer,[||] Nikolaus Heinrich,[||] and Klaus-Robert Müller[†,‡]

Fraunhofer FIRST, Kekuléstrasse 7, 12489 Berlin, Germany, Department of Computer Science, Technische Universität Berlin, Franklinstrasse 28/29, 10587 Berlin, Germany, idalab GmbH, Sophienstrasse 24, 10178 Berlin, Germany, and Research Laboratories of Bayer Schering Pharma AG, Müllerstrasse 178, 13342 Berlin, Germany

Received April 20, 2007; Revised Manuscript Received June 16, 2007; Accepted June 20, 2007

Abstract: Unfavorable lipophilicity and water solubility cause many drug failures; therefore these properties have to be taken into account early on in lead discovery. Commercial tools for predicting lipophilicity usually have been trained on small and neutral molecules, and are thus often unable to accurately predict in-house data. Using a modern Bayesian machine learning algorithm—a Gaussian process model—this study constructs a log D_7 model based on 14556 drug discovery compounds of Bayer Schering Pharma. Performance is compared with support vector machines, decision trees, ridge regression, and four commercial tools. In a blind test on 7013 new measurements from the last months (including compounds from new projects) 81% were predicted correctly within 1 log unit, compared to only 44% achieved by commercial software. Additional evaluations using public data are presented. We consider error bars for each method (model based error bars, ensemble based, and distance based approaches), and investigate how well they quantify the domain of applicability of each model.

Keywords: Drug discovery; modeling; domain of applicability; machine learning; Bayesian; Gaussian process; error bar; error estimation; random forest; ensemble; decision tree; support vector machine; support vector regression; distance

1. Introduction

Lipophilicity of drugs is a major factor in both pharmacokinetics and pharmacodynamics. Since a large fraction of drug failures (∼50%)[1] results from an unfavorable PC-ADME/T profile (absorption, distribution, metabolism, excretion, toxicity), the octanol water partition coefficients log *P* and log *D* are nowadays considered early on in lead discovery.

Due to the confidentiality of in-house data, makers of predictive tools are usually not able to incorporate such data from pharmaceutical companies. Commercial predictive tools are therefore typically constructed using publicly available measurements of relatively small and mostly neutral molecules. Often, their accuracy on the in-house compounds of pharmaceutical companies is relatively low.[2]

In our work, we follow a different route to derive models for lipophilicity that are tailored to in-house data. We use a modern machine learning tool, a so-called Gaussian process

* Author to whom correspondence should be addressed. Mailing address: Fraunhofer FIRST, Intelligent Data Analysis, Kekuléstrasse 7, 12489 Berlin, Germany. Tel: +49 (0)30 6392 1882. Fax: +49 (0)30 6392 1805. E-mail: timon.schroeter@first.fraunhofer.de.

† Technische Universität Berlin.

‡ Fraunhofer FIRST.

§ idalab GmbH.

|| Research Laboratories of Bayer Schering Pharma AG.

(1) Hou, T. J.; Xu, X. J. ADME evaluation in drug discovery. 3. Modeling blood-brain barrier partitioning using simple molecular descriptors. *J. Chem. Inf. Comput. Sci.* **2003**, *43* (6), 2137−2152.

(2) Bruneau, P.; McElroy, N. R. Generalized fragment-substructure based property prediction method. *J. Chem. Inf. Model.* **2006**, *46*, 1379−1387.

10.1021/mp0700413 CCC: $37.00
Published on Web 07/19/2007

model[3] (GP), to obtain a nonlinear mapping from descriptors to lipophilicity. A specific advantage of the tool is its Bayesian framework for model selection, that provides theoretically well founded criteria to automatically choose the "right amount of nonlinearity" for modeling. We can avoid extensive grid search in cross-validation or expert intervention to choose optimal parameter settings. Thus, the process can be fully automated. For the first data analysis phase, structures do not need to be disclosed, since all modeling is descriptor based.

Apart from the high performance, the chosen modeling approach shows another virtue that makes it an excellent tool for applications in chemistry: GP models have their roots in Bayesian statistics, and thus can supply the user with an error bar for each individual prediction. This quantification of the prediction uncertainty allows reduction of the error rate, by discarding predictions with large error bars, or by reconfirming the prediction with a laboratory experiment. In our work, we also compare these error bars with error bar heuristics that can be applied to other commonly used modeling approaches.

Performance is compared with models constructed using three established machine learning algorithms: support vector machines, random forests, and linear ridge regression. We show that the different log P and log D_7 models exhibit convincing prediction performance, both on benchmark data and on in-house data of drug molecules. We compare our results with several commercial tools, and show a large improvement of performance, in particular on the in-house classes of compounds.

Using machine learning algorithms, one can construct models of biological and chemical properties of molecules from a limited set of measurements.[4−8] This so-called training set is used to infer the underlying statistical properties and select a prediction model. Tuning of (hyper)parameters is usually performed using cross-validation or resampling methods. To evaluate the performance of the model, one should use a set of data that was not used in model building in any form. In the best case, the model is evaluated in a *blind test*, where the modelers do not have access to the held out data. Instead, the final model is applied to the blind test data by an independent evaluating team. In normal benchmark evaluations, retuning models on held-out data is possible and typically results in too optimistic results. In contrast, the blind-test strategy is nearly unbiased, because "cheating", i.e., using results on the held-out data for retuning the model, becomes infeasible. Note however that the blind test set of data needs to be of somewhat reasonable size, and should represent the typical application scenario of the model that is to be evaluated.

GP models have been previously applied in computational chemistry (our own recent results of modeling aqueous solubility are presented in refs 5 and 6), but rather small data sets were used, and typically no blind test was conducted:

(a) Burden[9] learned the toxicity of compounds and their activity on muscarinic and benzodiazepine receptors using up to 277 compounds.

(b) Enot et al.[10] predicted log P using 44 compounds from a *1,2-dithiole-3-one* series.

(c) Tino et al.[11] built GP models for log P from a public data set of 6912 compounds. Here, a blind test was conducted, but the validation set (provided by Pfizer) contained only 266 compounds.

This study goes beyond the prior work: Our models were trained on large sets of public and in-house data (up to 14556 compounds). A blind test was performed by an independent evaluating team at Bayer Schering Pharma using a set of 7013 drug discovery molecules from recent projects, that have not been available to the modeling team Fraunhofer FIRST and idalab. To facilitate reproduction of our results by other researchers, the complete list of compounds in the public data set is included in the supporting information to our initial communication.[4]

2. Estimating the Domain of Applicability of Models

A typical challenge for statistical models in the chemical space is to adequately determine the domain of applicability, i.e., the part of the chemical space where the model's predictions are reliable. To this end several "classical" approaches exists: *Range based methods* are based on

(3) Rasmussen, C. E; Williams, C. K. I. *Gaussian Processes for Machine Learning*; MIT Press: Cambridge, MA, 2005.

(4) Schroeter, T.; Schwaighofer, A.; Mika, S.; Ter Laak, A.; Suelzle, D.; Ganzer, U.; Heinrich, N.; Müller, K.-R. Predicting lipophilicity of drug discovery molecules using gaussian process models. *ChemMedChem.* URL: http://dx.doi.org/10.1002/cmdc.200700041.

(5) Schroeter, T.; Schwaighofer, A.; Mika, S.; Ter Laak, A.; Suelzle, D.; Ganzer, U.; Heinrich, N.; Müller, K.-R. Estimating the domain of applicability for machine learning qsar rmodels: A study on aqueous solubility of drug discovery molecules. *J. Comput.-Aided Mol. Des.*, accepted for publication. URL: http://dx.doi.org/10.1007/s10822-007-9125-z.

(6) Schwaighofer, A.; Schroeter, T.; Mika, S.; Laub, J.; ter Laak, A.; Sülzle, D.; Ganzer, U.; Heinrich, N.; Müller, K.-R. Accurate solubility prediction with error bars for electrolytes: A machine learning approach. *J. Chem. Inf. Model.* **2007**, *47* (2), 407−424. URL http://dx.doi.org/10.1021/ci600205g.

(7) Müller, K.-R.; Rätsch, G.; Sonnenburg, S.; Mika, S.; Grimm, M.; Heinrich, N. Classifying 'drug-likeness' with kernel-based learning methods. *J. Chem. Inf. Model.* **2005**, *45*, 249−253.

(8) Müller, K.-R.; Mika, S.; Rätsch, G.; Tsuda, K.; Schölkopf, B. An introduction to kernel-based learning algorithms. *IEEE Trans. Neural Networks* **2001**, *12* (2), 181−201.

(9) Burden, F. R. Quantitative structure-activity relationship studies using Gaussian processes. *J. Chem. Inf. Comput. Sci.* **2000**, *41* (3), 830−835.

(10) Enot, D. P.; Gautier, R.; Le Marouille, J. Y. Gaussian process: an efficient technique to solve quantitative structure-property relationship problems. *SAR QSAR Environ. Res.* **2001**, *12* (5), 461−469.

(11) Tino, P.; Nabney, I.; Williams, B. S.; Lösel, J.; Sun, Y. Nonlinear prediction of quantitative structure-activity relationships. *J. Chem. Inf. Comput. Sci.* **2004**, *44* (5), 1647−1653.

checking whether descriptors of test set compounds exceed the range of the respective descriptor covered in training.[12,13] A warning message is raised when this occurs. Also, *geometric methods* that estimate the convex hull of the training data can be used to further detail such estimates.[14] Mind that both these methods are not able to detect "holes" in the training data, that is, regions that are only scarcely populated with data. In principle, this can be achieved using geometric methods in a suitable feature space. To the best of our knowledge, there exists no published study about this kind of approach.

If experimental data for some new compounds are available, error estimates based on the *library approach*[15] can be used. By considering the closest neighbors in the library of new compounds with known measurements, it is possible to get a rough estimate of the bias for the respective test compound.

Probability density distribution based methods could, theoretically, be used to estimate the model reliability.[14] Still, high dimensional density estimation is recognized as an extremely difficult task, in particular since the behavior of densities in high dimensions may be completely counterintuitive.[16]

Distance based methods and *extrapolation measures*[2,12,14,17,18] consider one of a number of distance measures (Mahalanobis, Euclidean, etc.) to calculate the distance of a test compound to its closest neighbor(s) or the whole training set, respectively. Another way of using distance measures is to define a threshold and count the number of training compounds closer than the threshold. Hotelling's test or the leverage relies on the assumption that the data follows a Gaussian distribution in descriptor space and computes the Mahalanobis distance to the whole training set. Tetko correctly argues in ref 18 that descriptors have different relevance for predicting a specific property and concludes that property specific distances (respectively similarities) should be used. There is an interesting parallel to Gaussian process models: When GP models are allowed to assign weights to each descriptor that enters the model as input, they implicitly construct a property specific distance measure and use it both for making predictions and for estimating prediction errors.

When estimating the domain of applicability with *ensemble methods*, a number of models are trained on different sets of data. Typically the sets are generated by (re)sampling from a larger set of available training data. Therefore the models will tend to agree in regions of the descriptor space where a lot of training compounds are available and will disagree in sparsely populated regions. Alternatively, the training sets for the individual models may be generated by adding noise to the descriptors, such that each model operates on a slightly modified version of the whole set of descriptors. In this case the models will agree in regions where the predictions are not very sensitive to small changes in the descriptors and they will disagree in descriptor space regions where the sensitivity with respect to small descriptor changes is large. This methodology can be used with any type of model, but ensembles of ANNs[2,17−20] and ensembles of decision trees[13,17] ("random forests", Breiman[21]) are most commonly used.

The idea behind *Bayesian methods* is to treat all quantities involved in modeling as random variables. By means of Bayesian inference, the *a priori* assumptions about parameters are combined with the experimental data, to obtain the *a posteriori* knowledge. Hence, such models naturally output a probability distribution, instead of the "point prediction" in conventional learning methods. Regions of high predictive variance indicate not only compounds outside the domain of applicability but also regions of contradictory or scarce measurements. The most simple and also most widely used method is the naive Bayes classifier.[22,23] Gaussian process

(12) Tropsha, A. Variable selection qsar modeling, model validation, and virtual screening. In *Annual Reports in Computational Chemistry*; Spellmeyer, D. C., Ed.; Elsevier: Amsterdam, 2006; Vol. 2, Chapter 7, pp 113−126.

(13) Tong, W.; Xie, Q.; Hong, H.; Shi, L.; Fang, H.; Perkins, R. Assessment of prediction confidence and domain extrapolation of two structure-activity relationship models for predicting estrogen receptor binding activity. *Environ. Health Perspect.* **2004**, *112* (12), 1249−1254.

(14) Netzeva, T. I.; Worth, A. P; Aldenberg, T.; Benigni, R.; Cronin, M. T. D.; Gramatica, P.; Jaworska, J. S.; Kahn, S.; Klopman, G.; Marchant, C. A.; Myatt, G.; Nikolova-Jeliazkova, N.; Patlewicz, G. Y.; Perkins, R.; Roberts, D. W.; Schultz, T. W.; Stanton, D. T.; van de Sandt, J. J. M.; Tong, W.; Veith, G.; Yang, C. Current status of methods for defining the applicability domain of (quantitative) structure-activity relationships. *Altern. Lab. Anim.* **2005**, *33* (2), 1−19.

(15) Kühne, R.; Ebert, R.-U.; Schüürmann, G. Model selection based on structural similarity-method description and application to water solubility prediction. *J. Chem. Inf. Model*. **2006**, *46*, 636−641.

(16) Silverman, B. W. *Density Estimation for Statistics and Data Analysis*; Number 26 in Monographs on Statistics and Applied Probability; Chapman & Hall: London, 1986.

(17) Bruneau, P.; McElroy, N. R. Generalized fragment-substructure based property prediction method. *J. Chem. Inf. Model.* **2004**, *44*, 1912−1928.

(18) Tetko, I. V.; Bruneau, P.; Mewes, H.-W.; Rohrer, D. C.; Poda, G. I. Can we estimate the accuracy of ADME-tox predictions? *Drug Discovery Today* **2006**, *11* (15/16), 700−707.

(19) Göller, A. H.; Hennemann, M.; Keldenich, J.; Clark, T. In silico prediction of buffer solubility based on quantum-mechanical and hqsar- and topology-based descriptors. *J. Chem. Inf. Model.* **2006**, *46* (2), 648−658.

(20) Manallack, D. T.; Tehan, B. G.; Gancia, E.; Hudson, B. D.; Ford, M. G.; Livingstone, D. J.; Whitley, D. C.; Pitt, W. R. A consensus neural network-based technique for discriminating soluble and poorly soluble compounds. *J. Chem. Inf. Model.* **2003**, *43*, 674−679.

(21) Breiman, L. Random forests. *Machine Learning* **2001**, *45*, 5−32. URL: http://dx.doi.org/10.1023/A:1010933404324.

(22) Bender, A.; Mussa, H. Y.; Glen, R. C. Screening for dihydrofolate reductase inhibitors using molprint 2d, a fast fragment-based method employing the nave bayesian classifier: Limitations of the descriptor and the importance of balanced chemistry in training and test sets. *J. Biomol. Screening* **2005**, *10* (7), 658−666. URL: http://jbx.sagepub.com/cgi/content/abstract/10/7/658.

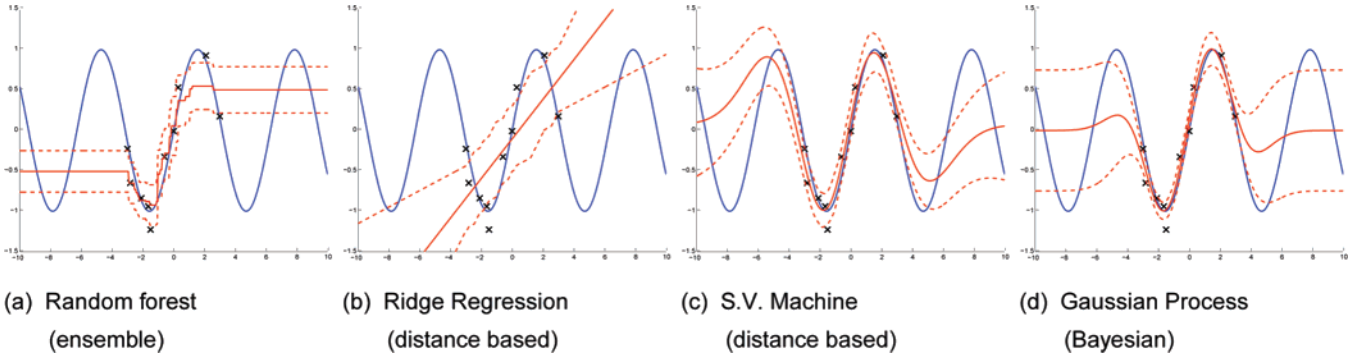


Figure 1. The four different regression models employed in this study are trained on a small number of noisy measurements (black crosses) of the sine function (blue line). Predictions from each model are drawn as solid red lines, while dashed red lines indicate errors estimated by the respective model (in case of the Gaussian process and random forest) or a distance based approach (in case of the support vector machine and ridge regression model).

regression and classification are more sophisticated Bayesian methods, see section 3.5.4.

In the present study, we use the Bayesian Gaussian process models, ensembles, and distance based methods. All of these can handle empty regions in descriptor space and quantify their confidence, rather than just marking some predictions as possibly unreliable. Confidence estimates will be presented in a form that is intuitively understandable to chemists and other scientists.

2.1. One-Dimensional Examples. Figure 1 shows a simple one-dimensional example of the four different methods of error estimation we use in this study. The sine function (shown as a blue line in each subplot) is to be learned. The available training data are 10 points marked by black crosses. These are generated by randomly choosing *x*-values and evaluating the sine function at these points. We simulate measurement noise by adding Gaussian distributed random numbers with standard deviation 0.2 to the *y*-values.

The random forest, Figure 1a, does provide a reasonable fit to the training points (yet the prediction is not smooth, due to the space dividing property of the decision trees). Predicted errors are acceptable in the vicinity of the training points, but overconfident when predictions far from the training points are sought. It should be noted that the behavior of error bars in regions outside of the training data depends solely on the ensemble members on the boundary of the training data. If the ensemble members, by chance, agree in their prediction, an error bar of zero would be the result.

The linear model, Figure 1b, clearly cannot fit the points from the nonlinear function. Therefore, the distance based error estimations are misleading: Low errors are predicted in regions close to the training points, but the actual error is quite large due to the poorly fitting model. This shows that the process of error estimation should not be decoupled from the actual model fitting: The error estimate should also indicate regions of poor fit.

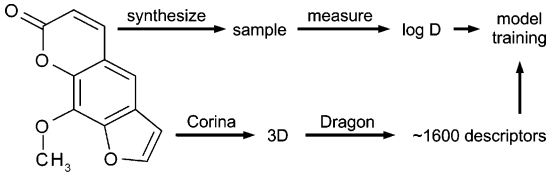


Figure 2. The process of model building.

The support vector machine, Figure 1c, adapts to the nonlinearity in the input data and extrapolates well. The error estimation (the same distance based procedure as described for the real data, section 4.4) produces slightly conservative (large) error bars in the region close the training points, and too small errors when extrapolating.

The Gaussian process, Figure 1d, also captures the nonlinearity in the input data and is able to extrapolate. Predicted errors are small in the region close to the training points and increase strongly enough in the extrapolation region.

3. Methods and Data

3.1. Methodology Overview. The training procedure is outlined in Figure 2. We use Corina[24] to generate a 3D structure for each molecule. Molecular descriptors are calculated using the Dragon[25] software. Finally, a number of machine learning algorithms are used to "train" models, i.e., to infer the relationship between the descriptors and the experimental values for log *P* and log D_7.

To make predictions for new compounds, structures are again converted to 3D and descriptors are calculated. From the descriptors of each molecule, the model generates a prediction of log *P* and/or log D_7, and in case of the Gaussian process and random forest also a confidence estimate (error bar).

3.2. Data Preparation. 3.2.1. Multiple Measurements. If multiple measurements exist for the same compound, we

(23) Sun, H. An accurate and interpretable bayesian classification model for prediction of herg liability. *ChemMedChem* **2006**, *1* (3), 315−322.

(24) Sadowski, J.; Schwab, C.; Gasteiger, J. *Corina v3.1*; Erlangen, Germany.

(25) Todeschini, R.; Consonni, V.; Mauri, A.; Pavan, M. *DRAGON v1.2*; Milano, Italy.

Table 1. Summary of the Different Setups That Are Used for Performance Evaluation[a]

Setup	Prediction	Data
In-house	log D	Training: in−house (14556); Validation
In-house validation	log D	Training: in−house (14556); Validation: in−house validation (7013)
Public	log P	Training: Physprop/Beilstein (7926); Validation

[a] See section 3.3 for a description and section 3.2 for details on the individual data sets.

merge them as described in the following to obtain a consensus value for model building. For each compound we generate the histogram of experimental values. Characteristic properties of histograms are the spread of values (*y*-spread) and the spread of the bin heights (*z*-spread). If all measured values are similar (small *y*-spread), the median value is taken as consensus value. If a group of similar measurements and smaller number of far apart measurements exists, both *y*-spread and *z*-spread are large. In this case we treat the far apart measurements as outliers, i.e., we remove them and then use the median of the agreeing measurements as consensus value. If an equal number of measurements supports one of two (or more) far apart values (high *y*-spread and zero *z*-spread), we discard the compound. Initial experiments suggested that 0.5 (on the measurements log-scale) is a suitable value for the threshold between small and large *y*-spreads.

3.2.2. Dataset 1: In-House. Dataset 1 consists of 14556 drug discovery compounds of Bayer Schering Pharma (Table 1). log *D* was measured following the experimental procedure described in section A.

For the majority of compounds, log *D* was measured at pH = 7.0. For about 600 compounds log *D* was measured at pH = 7.4. Although for particular compounds with pK_a values close to pH = 7 one can expect deviations in log *D* of up to 0.4 (extreme case), first experiments showed that building separate models is not necessary. No negative impact on the model accuracy was observed when the measurements performed at pH = 7.4 are included in the larger set.

3.2.3. Dataset 2: In-House Validation. Dataset 2 is a set of 7013 new measurements of drug discovery molecules of Bayer Schering Pharma that were collected in the months after dataset 1 had been measured, and thus also includes compounds from new projects. log *D* was measured following the same experimental procedure as was used for dataset 1, see section A.

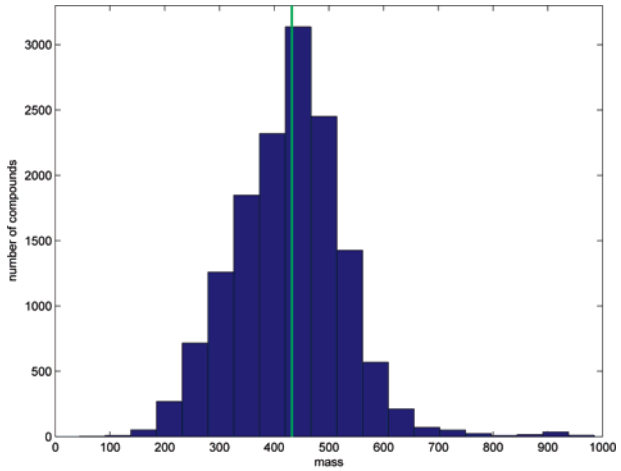
3.2.4. Dataset 3: Public. This set contains measurements of log *P* for 7926 unique compounds extracted from the Physprop[26] and Beilstein[27] databases. log *D* measurements performed at various pH values are often reported as log *P* in the literature, despite the fact that log *P* applies, by definition, only to a molecule in its neutral form (i.e., the pH of the solution has to be adjusted so that the molecule is neutral). To avoid these wrongly reported log *P* values, the set was restricted to compounds predicted to be completely neutral at pH 2 to 11 by ACDLabs v9, since, for these compounds, log *D* values in the given pH ranges coincide with the correct log *P* values.

3.2.5. Differences between In-House and Public Data. Histograms of the molecular weight for each dataset are given in Figure 3. The median of the molecular weight is 227 g/mol for the public dataset, 432 g/mol for the in-house set, and 405 g/mol for the in-house validation set (marked by vertical green lines in the plots). As we can see from the histogram, more than 90% of the compounds in the public set have a molecular mass lower than 400 g/mol, that is well below the median of the molecular mass for the two in-house sets of data. In this study, we separately evaluate models on the public and in-house sets of data. In principle, data from internal and external sources can be combined. However, care has to be taken when evaluating models on mixed sets, since such models typically perform well on compounds with low molecular weight (see section 4.2) but are less accurate for the larger compounds relevant to drug discovery (see section 4.3).

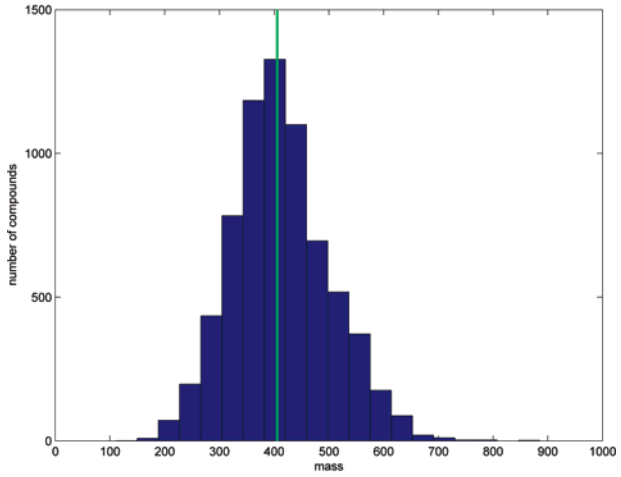
3.3. Training and Validation Setups. 3.3.1. Cross-Validation. On the in-house and public set of data, models are evaluated in leave 50% out cross-validation, i.e., the data is randomly split into two halves. A model is trained on the first half and evaluated on the other half. This is repeated with the two halves of the validation set exchanged, so that predictions for all compounds in the set are generated. The overall procedure is then repeated 10 times with a different

(26) *Physical/Chemical Property Database (PHYSPROP)*; Syracuse, NY.

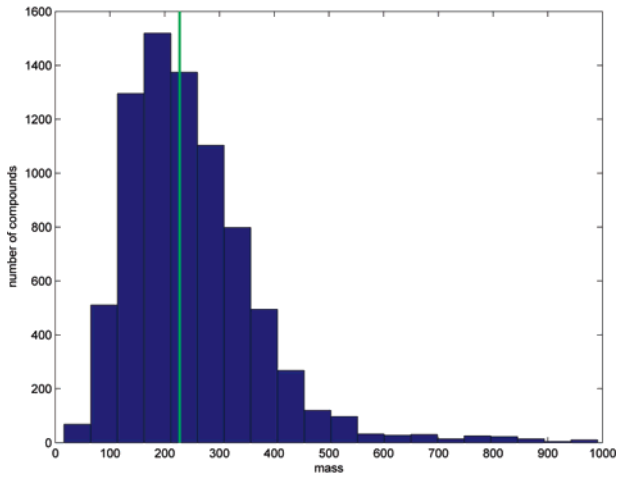
(27) *Beilstein CrossFire Database*; San Ramon, CA.



(a) Data Set 1: in-house



(b) Data Set 2: in-house validation



(c) Data Set 3: public

Figure 3. Histograms of molecular weight. Vertical green lines mark the median of the molecular weight of the respective data set.

random split. Each prediction is an out-of-sample prediction, made by a model that has not seen the particular compound in its training data.

3.3.2. Blind Test. Gaussian process models built by the modelers at Fraunhofer FIRST and idalab on the in-house set of data were evaluated by researchers at Bayer Schering Pharma on the *in-house validation* set of data. At this point in time, the modelers had no knowledge of the nature or log *D* values of the validation set. Later, the validation data was revealed to the modelers and used as an external validation set to assess the performance of other types of models.

3.4. Molecular Descriptors. We use the Dragon descriptors by Todeschini et al.[28] They are organized in 20 blocks and include, among others, constitutional descriptors, topological descriptors, walk and path counts, eigenvalue-based indices, functional group counts. and atom-centered fragments. A full list including references can be found online.[29]

As one of their most pronounced features, Gaussian process models allow the assignment of weights to each descriptor that enters the model as input. The similarity for two compounds as computed by the GP model takes into account that the *i*th descriptor contributes to the similarity with weight w_i (see 3.5.4). These weights are chosen automatically during model fitting and can then be inspected in order to get an impression of the relevance of individual descriptors.

We found that using a small (<50) set of descriptors results in only slightly decreased accuracy when comparing to models built on the full set of 1664 descriptors. The error predictions, however, turn out to be too optimistic in this case. Including whole blocks containing important descriptors leads to both accurate predictions and accurate error estimations (see section 4.1). In this study, we used the full Dragon blocks 1, 2, 6, 9, 12, 15, 16, 17, 18, and 20. A discussion of the importance of individual descriptors can be found in section 4.1.

3.5. Machine Learning Methods. 3.5.1. Introductory Remarks. Since the application of Gaussian process regression is still relatively new in the field of chemoinformatics, we chose to explain and illustrate the modeling idea. Support vector machines are seen as established, but still deserve some discussion due to interesting parallels and differences with the Bayesian GP approach.

Linear ridge regression, decision trees, and ensembles of trees (random forests) are considered established methods—here we mainly note how the employed implementation differs from the original algorithm, for which the reader is referred to the literature.

3.5.2. Linear Ridge Regression. Ridge regression combines a linear model with a regularization term that effectively shrinks coefficients of the model toward zero. This is particularly important for our application since a standard

(28) Todeschini, R.; Consonni, V. *Handbook of Molecular Descriptors*; John Wiley & Sons, Ltd.: Chichester, 2000.

(29) Todeschini, R.; Consonni, V.; Mauri, A.; Pavan, M. Dragon for windows and linux 2006. URL: http://www.talete.mi.it/help/dragon_help/ (accessed 14 May 2006).

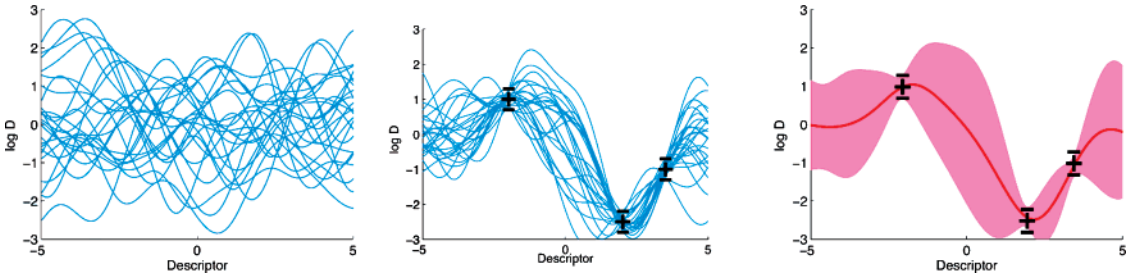


Figure 4. Bayesian modeling with Gaussian processes.

linear model runs into problems when descriptors are correlated. We choose the complexity parameter λ that controls the amount of shrinkage by grid search in nested cross-validation.

3.5.3. Random Forest. A modified version of the random forests method of Breiman[21] is employed. Trees are constructed without bagging or bootstrapping, and pruning of individual trees is done using a CART-style error-size tradeoff.

The predictive variance is calculated by averaging the variance of the predictions from the different trees in the forest and the average estimated variance from training points found at each tree leaf.

3.5.4. Gaussian Process Regression. Gaussian process (GP) models have their origin[30] in the field of Bayesian statistics. A description of the methodology, including mathematical derivations, can be found in Schwaighofer et al.[6] For in-depth coverage we refer the reader to a recent book by Rasmussen.[3]

Figure 4 illustrates the principles behind GP models: Before having measured log *D* values, any relationship between the descriptor (in this 2-dimensional example, only one descriptor is used and plotted on the *x*-axis) and log *D* (*y*-axis) is equally likely. This is represented by an infinitely large family of functions that map from descriptor space to log *D* space. The family is described by a *Gaussian process prior*, and 25 examples are shown in Figure 4 (left).

When training the model with log *D* values for a number of molecules (symbolized by black crosses in Figure 4 (middle)), we discard (or put lower weight on) all functions that do not pass near by these known data points.

To predict log *D* values for new molecules, we just average over the functions remaining in the pool (the red line in Figure 4 (right)) and read off the value corresponding to the new molecules' descriptors. To predict error bars, we calculate the standard deviation of the functions remaining in the pool at the position given by each new molecule's descriptors. The 2σ environment for all descriptor values on the *x*-axis is marked by the red region in Figure 4 (right). Close to known points, the uncertainty is small, but not zero: Measurements are assumed to be noisy. The uncertainty increases far from known points and in regions where measurements disagree.

Effectively, all the steps described above are not implemented by sampling, but via integral operations.[6] The Bayesian concept of a weighed average of functions with a certain mean (log *D* prediction) and standard deviation (error bar) is, however, preserved.

In order to derive the GP model prediction, let f be a function that depends on a vector $\mathbf{x}$ of d molecular descriptors and outputs log *D*, i.e., $f(\mathbf{x}) \approx \log D(\mathbf{x})$. We assume that each possible function f is a realization of a Gaussian stochastic process, and thus can be fully described by considering pairs of compounds $\mathbf{x}$ and $\mathbf{x}'$. By the properties of the Gaussian process, functional values $f(\mathbf{x}_1)$, ..., $f(\mathbf{x}_n)$ for any finite set of n points form a Gaussian distribution. The covariance for each pair is then given by the covariance function,

$$\mathrm{cov}(f(\mathbf{x}),f(\mathbf{x}')) = k(\mathbf{x},\mathbf{x}') \qquad (1)$$

which has a role similar to the kernel function in support vector machines[8,31] and other kernel based learning methods. Any previous knowledge of the phenomenon to be predicted is expressed in the covariance function k.

For n compounds the actual data consist of n log *D* measurements, y_1, ..., y_n and n descriptor vectors, x_1, ..., x_n (each of length d). Assuming that measurements are noisy, we relate the n measured values to the true log *D* by

$$y_i = f(\mathbf{x}_i) + \epsilon \qquad (2)$$

where ϵ is Gaussian noise with standard deviation σ. σ can be a scalar, meaning that all measurements are equally noisy. σ can also be a vector, allowing, in principle, the use of a different noise level for each individual compound. In practice we found it useful to assume equal measurement noise for groups of compounds that, e.g., have been measured in the same laboratory. In this way, model performance can be improved and we can learn the noise level resulting from different (or uniform) experimental procedures directly from the data.[6]

Applying a number of transformations and steps of statistical inference[6] we find that the predicted log *D* for a new compound $\mathbf{x}_*$ follows a Gaussian distribution with mean $\bar{f}(\mathbf{x}_*)$ and standard deviation std $f(\mathbf{x}_*)$, with

(30) O'Hagan, A. Curve fitting and optimal design for prediction. *J. R. Stat. Soc., Ser. B: Methodological* **1978**, *40* (1), 1−42.

(31) Schölkopf, B.; Smola, A. J. *Learning with Kernels*; MIT Press: Cambridge, MA, 2002.

$$\bar{f}(\mathbf{x}_*) = \sum_{i=1}^{n} \alpha_i k(\mathbf{x}_*, \mathbf{x}_i) \quad (3)$$

$$\text{std } f(\mathbf{x}_*) = \sqrt{k(\mathbf{x}_*, \mathbf{x}_*) - \sum_{i=1}^{n} \sum_{j=1}^{n} k(\mathbf{x}_*, \mathbf{x}_i)\, k(\mathbf{x}_*, \mathbf{x}_j) L_{ij}} \quad (4)$$

Coefficients α_i are found by solving a system of linear equations, $y = (K + \sigma^2 I)\alpha$, with $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$. For the standard deviation, L_{ij} are the elements of the matrix $L = (K + \sigma^2 I)^{-1}$.

Details on inferring the parameters of the covariance function k and the measurement noise σ can be found in Schwaighofer et al.[6]

In direct implementations, memory demand increases quadratically with the number of data points. Recent developments of approximation and sampling techniques[32] allow the training of Gaussian process models on thousands of data points. For the data sets used in this study, we precede the actual GP training by a k-means clustering, such that each cluster contains up to 5000 compounds and train one GP per cluster. When applying the model, predictions from the individual GP models are generated and the prediction with the highest confidence (smallest error bar) is chosen.

3.5.5. Support Vector Regression. Support vector machines for regression and classification are based on the principle of structural risk minimization. Out of a certain class of functions we want to find the function that minimizes some notion of error, measured by the so-called loss function. Using a very large class of functions (i.e., a very complex model) one can perfectly fit to the training data, but the resulting function will not generalize to new, unseen data (over-fitting). On the contrary, using a small class of functions (simple, e.g., linear models) one may not be able to fit the data reasonably, again resulting in inaccurate predictions.

Choosing a function class with functions of the right complexity can be achieved by regularization: We combine the empirical loss on the training data with a penalty term for the complexity and then minimize the sum (objective function). Under certain assumptions (for example, that the training and test data are sampled from the same distribution), it can be proven that this way of choosing the function class leads to an optimal model.[33−35]

In the following we will first describe the idea behind linear SVR and then generalize to the nonlinear case.

Given a vector $\mathbf{x}$ of descriptors for a compound, the quantity of interest y (in our case log D) will be predicted as $y = f(\mathbf{x})$. Linear SVM finds a predictor $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b$, such that the empirical error and the norm of the weight vector $\mathbf{w}$ are minimal. We employ an ϵ-insensitive loss function which does not penalize deviations from the measured value that are smaller than ϵ. Model training is done by solving the convex quadratic optimization problem:

$$\min_{\mathbf{w},b,\xi} \frac{1}{2} ||\mathbf{w}||^2 + C \sum_{i=1}^{n} \xi_i$$

subject to

$$|f(\mathbf{x}_i) - y_i| \le \epsilon + \xi_i, \quad \xi_i \ge 0, \quad i = 1, ..., n$$

The threshold ϵ from the loss function manifests in the constraints. "Slack variables" ξ are introduced and penalized in the objective function such that deviation by more than ϵ increases the objective function only linearly. This reduces the influence of outliers in the data. The constants ϵ and C are chosen by cross-validation. In principle, it is also possible to use more sophisticated approaches[36] that compute SVR solutions for multiple parameter values in an efficient manner.

Employing the so-called kernel trick[8,35] one can generalize to nonlinear models. Functions f of the form $f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i k(\mathbf{x}_i, \mathbf{x}) + b$ can be generated by rewriting the liner SVM equations such that the descriptors $\mathbf{x}$ only appear inside scalar products ($\mathbf{x}_i^T\mathbf{x}_j$). These scalar products can then be replaced by a kernel function $k(\mathbf{x}_i, \mathbf{x}_j)$, that implicitly maps the descriptors into a high-dimensional feature-space and computes the scalar product there. There are many interesting connections between SVM and GP methods. One of them is that the valid kernel functions for support vector algorithms are also valid covariance functions for a GP model and vice versa. In this study, we use support vector regression with an RBF kernel function.[6,8]

4. Results and Discussion

4.1. Choice of Descriptors. Gaussian process models can assign weights to each descriptor that enters the model as input (see section 3.4 for details). The 30 interpretable descriptors with highest weight are clearly connected with log P and log D_7. They include the sum of geometrical distances between pairs of oxygen atoms, counts of various functional groups [donor atoms for H-bonds (N and O); H attached to heteroatom; hydroxyl groups; hydroxyl groups in phenol, enol, carboxyl; ether groups; oxygen atoms; benzene-like rings; carbon atoms; quaternary nitrogen; tertiary amines; secondary amines], and a number of continuous quantities [topological polar surface area using N, O polar contributions; topological polar surface area using N, O, S, P polar contributions; mean atomic van der Waals volume (scaled on carbon atom); harmonic oscillator model of aromaticity index total; molar refractivity; hydrophilic factor;[37] molecular weight and 11 other measures of size, e.g., sum of conventional bond orders, sum of atomic van der Waals volumes, and size indices].

(32) Quionero-Candela, J.; Rasmussen, C. E. A unifying view of sparse approximate Gaussian process regression. *J. Machine Learn. Res.* **2005**, *6* (December), 1939−1959. URL: http://www.jmlr.org/papers/volume6/quinonero-candela05a/quinonero-candela05a.pdf.

(33) Vapnik, V. N. *Statistical Learning Theory*; Wiley: New York, 1998.

(34) Cristianini, N.; Shawe-Taylor, J. *An Introduction to Support Vector Machines*; Cambridge University Press: Cambridge, U.K., 2000.

(35) This reference was deleted on revision.

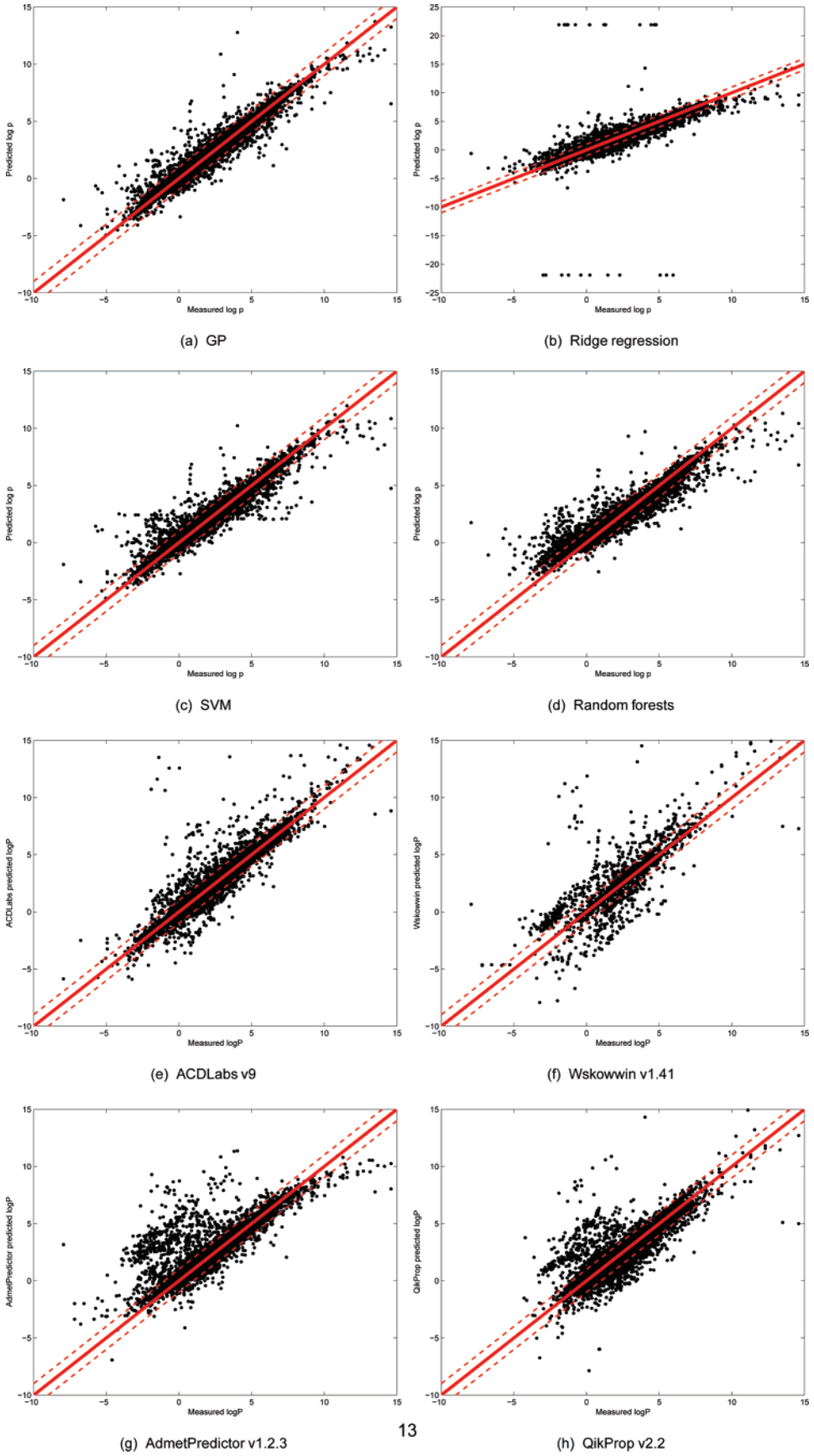


Figure 5. Scatter plots for GP, SVM, ridge regression, and random forests (one arbitrarily chosen cross-validation run each) and all four commercial tools on the public data set (Physprop/Beilstein).

Table 2. Accuracy Achieved on the Public Data Sets Physprop/Beilstein Using Different Machine Learning Methods Compared with the Performance of Commercial Tools[a]

public data Physprop/Beilstein	MAE	RMSE	% ± 1
Gaussian process	0.38	0.66	92.6
linear ridge regression	0.59	0.89	84.4
support vector machine	0.40	0.71	91.8
random forest	0.52	0.82	87.6
ACDLabs v9	0.43	0.90	89.2
Wskowwin v1.41	0.25	0.90	91.6
AdmetPredictor v1.2.3	0.65	1.32	86.9
QikProp v2.2	0.76	1.23	79.6
baseline: predict mean log *P*	1.68	2.24	40.7

[a] MAE, RMSE, and % ± 1 denote the mean absolute error, the root mean squared error, and the percentage of compounds predicted with less than 1 log unit error.

Table 3. Accuracy Achieved Using Gaussian Process Models, Support Vector Machines, Linear Ridge Regression, and Random Forests for the In-House Datasets, Compared with the Performance of ACDLabs v9[a]

	MAE	RMSE	% ± 1
In-House Cross-Validation			
Gaussian process	0.41	0.66	90.7
linear ridge regression	0.53	0.96	88.3
support vector machine	0.44	0.70	89.8
random forest	0.55	0.80	84.4
ACDLabs v9	1.41	1.90	46.6
baseline: predict mean log D_7	1.13	1.47	53.4
In-House Blind Test			
Gaussian process	0.60	0.82	81.2
linear ridge regression	0.60	0.83	82.2
support vector machine	0.58	0.81	81.6
random forest	0.74	1.00	74.8
ACDLabs v9	1.40	1.79	44.2
baseline: predict mean log D_7	1.17	1.51	51.7

[a] MAE, RMSE, and % ± 1 denote the mean absolute error, the root mean squared error, and the percentage of compounds predicted with an error less than 1.

We found that using a small set of descriptors results in only slightly decreased accuracy when comparing to models built on the full set of 1664 descriptors. The error predictions, however, turn out to be too optimistic. In other words: The log D_7 is predicted accurately for most compounds, but the model cannot correctly detect whether the test compound has, for example, additional functional groups. These functional groups might not have occurred in the training data, and were thus not included by the feature selection step. In the test case, the information about these additional functional groups is important since it helps to detect that these compounds are different from those the model has been trained on, i.e., the error bar should increase. Including whole blocks containing important descriptors leads to both accurate predictions and accurate error estimations. For, e.g., a GP model these *surplus* descriptors will get only a small weight during training—but the weight will not be zero. In consequence the model has more information than it needs for predicting log D_7 and will respond to new properties (functional groups etc.) of molecules by estimating a larger prediction error.

In this study, we used the full Dragon blocks 1, 2, 6, 9, 12, 15, 16, 17, 18, and 20, thereby including constitutional descriptors, topological descriptors, 2D autocorrelations, topological charge indices, geometrical descriptors, WHIM descriptors, GETAWAY descriptors, functional group counts, atom-centered fragments, and molecular properties. With this set of 904 descriptors, the model's accuracy is only slightly smaller than the accuracy of models built on all 1664 descriptors, but the computational cost and memory requirements are significantly reduced, and predicted error bars display close to ideal statistical properties (see section 4.4 and section 4.5).

4.2. Overall Accuracy: Public Data. The accuracy achieved on the public data set using different machine learning methods is compared with the performance of ACDLabs v9, Wskowwin v1.41, AdmetPredictor v1.2.3, and QikProp v2.2 in Table 2. The row labeled "baseline" lists the performance achieved when constantly predicting the average log *P* of the dataset. Scatter plots for all methods (one arbitrarily chosen cross-validation run each) and all four commercial tools are given in Figure 5.

The support vector machine and random forest models exhibit similarly high performance (91.6% respectively 87.6% correct within 1 log unit) as the three best performing commercial tools ACDLabs v9, Wskowwin v1.41, and AdmetPredictor v1.2.3 (86.9% to 91.6% correct ±1). The Gaussian process model performs slightly better (92.6% ±1) than the best performing commercial tool (91.6% ±1). The linear ridge regression model predicted a number of log *P* values as high as 10^{16}. For all plots and statistical evaluations, predictions from the linear ridge regression model were postprocessed, setting 1.5 times the highest/lowest log *P* values in the training data as upper/lower limits. Thus, error measures like mean absolute error can be used in a more meaningful way. 84.4% of all predictions were correct within 1 log unit. In general, we found that the nonlinear methods are more accurate and, in particular, produce fewer "far off" predictions, as can be seen in Figure 5a,c,d.

Examining Figure 5e−h, we find that all four commercial tools produce a number of outliers. ACDLabs v9 and Wskowwin v1.41 generate fewer than 10 very "far off"

(36) Wang, G.; Yeung, D.-Y.; Lochovsky, F. H. Two-dimensional solution path for support vector regression. In *Proceedings of ICML06*; De Raedt, L., Wrobel, S., Eds.; ACM Press: New York, NY, 2006; pp 993−1000. URL: http://www.icml2006.org/icml_documents/camera-ready/125_Two_Dimensional_Solu.pdf.

(37) Todeschini, R.; Gramatica, P. Linear and nonlinear functions on modeling of aqueous solubility of organic compounds by two structure representation methods. *Quant. Struct. Act. Relat.* **1997**, *16*, 116−125.

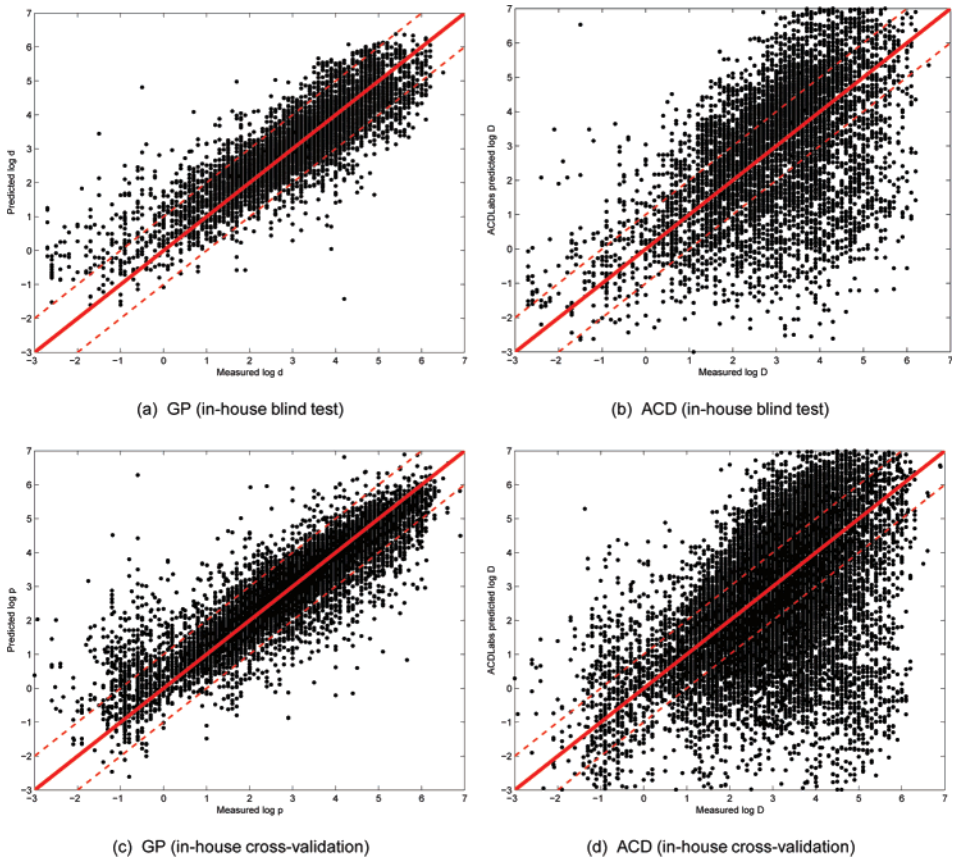


Figure 6. Scatter plots for Gaussian process and ACDLabs v9 on in-house validation data in blind test (subplots a, b) and on in-house data in cross-validation (subplots c, d).

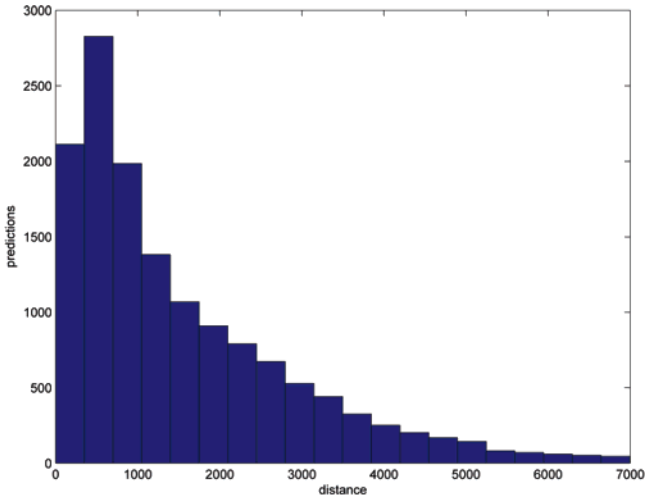
predictions, but their log *P* is overestimated by more than 10 orders of magnitude. For ∼50 compounds the predicted values are too high by 2 or 3 log units. Still, the overall performance of both ACDLabs v9 and Wskowwin v1.41 is good, which is also reflected in the low MAE and RMSE, see Table 2. Neither QikProp v2.2 nor AdmetPredictor v1.2.3 produces very "far off" predictions (>10 orders of magnitude). For several hundreds of compounds, log *P* is predicted too high by 2 or 3 orders of magnitude, reducing the overall performance (see Table 2).

All four commercial tools have been trained using a number of compounds that are also included in the Beilstein and Physprop databases. In these cases the correct value is reproduced, rather than predicted. This effect can be seen most clearly in the results for Wskowwin, where many of the model predictions for the public data are right on the optimal prediction line. Thus, the presented evaluation is, most likely, biased in favor of the commercial tools.

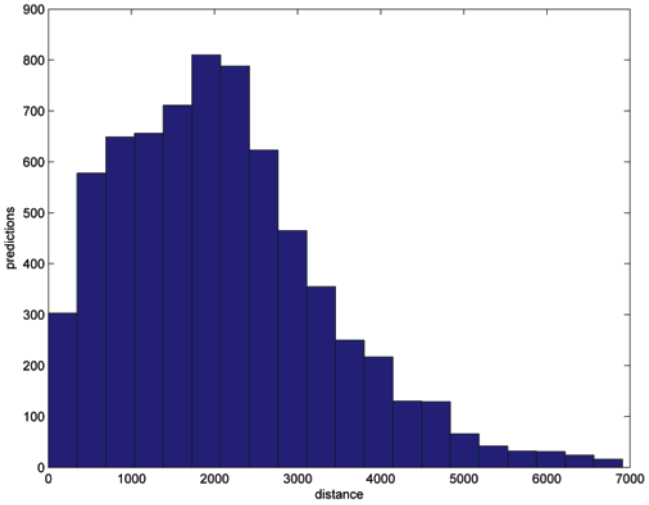
Our own results were obtained in 2-fold cross-validation (train on half of the data, evaluate on the other half), repeated 10 times with different random splits of the data. Therefore, test and training data tend to have a similar distribution across different compound classes. This is not the case in the typical application scenario of such models: In new projects, new compound classes will be investigated, resulting in less accurate predictions. To get a realistic estimate of the performance on unseen data, a "blind test" evaluation on data including different compound classes is important. For models built on the Bayer Schering Pharma in-house data, we present such an evaluation in the subsequent section.

4.3. Overall Accuracy: In-House Data. The results for predicting log D_7 on Bayer Schering Pharma in-house data are listed in Table 3. The corresponding scatter plots are given in Figure 6. When evaluated in 2-fold cross-validation on the in-house data (see Table 3, top), the Gaussian process model, the support vector machine, and the linear ridge regression yielded good results (88.3% to 90.7% correct within 1 log unit), with the Gaussian process model performing best (90.7% ±1). This model was then validated in blind evaluation at Bayer Schering Pharma on a set of 7013 new measurements from the last months. Later, the data was made available to the modeling team at Fraunhofer and idalab and other methods were evaluated, treating the former blind test data as an external validation set. These results are given in Table 3 (bottom). Among the commercial tools that were available to us, only ACDLabs is able to calculate log D_7, and can thus be used as a benchmark.

With ACDLabs v9, only 44.2% of the compounds are predicted correctly within 1 log unit. Mind that ACD has been trained on shake-flask measurements, while the in-house measurements used in this study were performed with the HPLC methodology described in section A. With our tailored models, we achieved 81.2% to 82.2% correct predictions. These are very good results, considering that the structures



(a) In-house cross-validation



(b) In-house blind test

Figure 7. Histograms of Mahalanobis distances from each compound to the closest compound in the respective training set. Distances for the cross-validated in-house setup (a) were calculated for the training/validation-split of one arbitrarily chosen cross-validation run.

were at no point in time available to the modeling team at FIRST/idalab. Furthermore, the blind test data stems from new drug discovery projects, and thus represents different structural classes than those present in the training data.

The fact that performance decreases when comparing the results achieved in cross-validation with the blind test could be taken as a hint that the nonlinear models did overfit to their training data. However, typical symptoms of overfitting, like a too large number of support vectors in SVM models, were not present. A large fraction of all compounds in the validation set is, however, very dissimilar to the training data. Histograms of Mahalanobis distances from each compound in the validation to the closest training compound are presented in Figure 7. We used the same set of descriptors for both model building and distance calculation.

In a typical cross-validation run on the in-house data set, 50% of the compounds have a nearest neighbor closer than 1100 units, see Figure 7, top. In the blind test set, less than 25% of the compounds have neighbors closer that 1100 units, see Figure 7, bottom.

This supports our hypothesis that the difference in performance between the cross-validation results and the blind test is caused by a large number of compounds being dissimilar to the training set compounds. Therefore it should be possible to achieve higher performance by focusing on compounds that are clearly inside the domain of applicability of the respective model. We investigate this question in section 4.5.

4.4. Individual Error Estimation for Interactive Use. Researchers establishing error estimations based on the distance of compounds to the training data typically present plots or tables where prediction errors are binned by distance, i.e., averaging over a large number of predictions, because the correlation between distances and errors is typically not too strong when considering individual compounds. When binning by the distance, one can clearly see how the error increases as the distance increases.[14,17] One can fit a function to this relationship and use it to generate an error prediction for each prediction the model makes. But how does the user know what an error prediction of, e.g., 0.6 log unit really means? In how many cases does the user expect the error to be larger than the predicted error? How much larger can errors turn out?

The most commonly used description of uncertainty (such as measurement errors, prediction errors, etc.) in chemistry, physics, and other fields is the error bar. Its definition is based on the assumption that errors follow a Gaussian distribution. When using a probabilistic model that predicts a Gaussian (i.e., a mean $\bar{f}$ and a standard deviation σ), it follows that the true value has to be in the interval $\bar{f} \pm \sigma$ with 68% confidence, and in the interval $\bar{f} \pm 2\sigma$ with 95% confidence, etc. To evaluate the quality of the predicted error bars, one can therefore compare with the true experimental values, and count how many of them are actually within the σ, 2σ, etc. intervals. (We found this procedure to be more reliable than using numeric criteria, such as the log probability of the predictive distribution.)

The Gaussian process model can directly predict error bars. In the implementation of random forests used in this study, the predictive variance is calculated by averaging the variance of the predictions from the different trees in the forest and the average estimated variance from training points found at each tree leaf.

For the linear ridge regression models and the support vector machines, error bars were estimated by fitting exponential and linear functions to the errors observed when evaluating the models in cross-validation and the Mahalanobis distances to the closest neighbors in the training set of the respective split. Since both linear and exponential functions worked equally well, we chose the simple linear functions to estimate error bars from the distances.

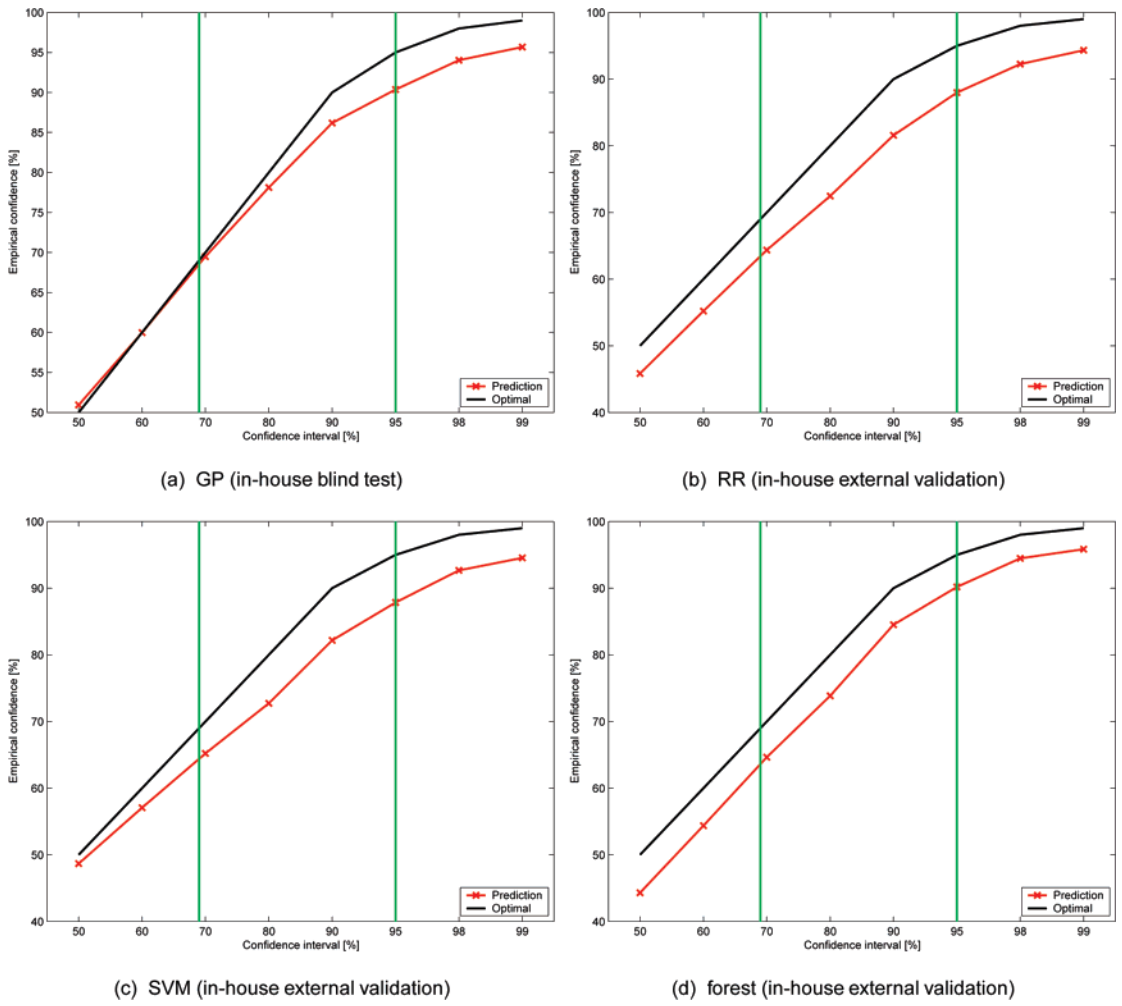


Figure 8. Predicted error bars can be evaluated by counting how many predictions are actually within a σ, 2σ, etc. environment (red line) and comparing with the optimal percentage (black line). The vertical green lines indicate the σ and 2σ environments, and the corresponding numbers can be found in Table 4.

Table 4. Predicted Error Bars Can Be Evaluated by Counting How Many Predictions Are Actually within a σ, 2σ, etc. Environment and Comparing with the Optimal Percentage[a]

environment	pred $\pm$ σ	pred $\pm$ 2σ
optimal pred $\pm$ σ	68.7	95.0
GP	67.5	90.4
RR	62.6	88.0
SVM	63.7	87.9
forest	62.5	90.2

[a] A graphical presentation of these results including fractions of σ can be found in Figure 8.

Plots of the empirical confidence versus the confidence interval are presented in Figure 8 (red line). The optimal curve is marked by the black line. The σ and 2σ environments are marked by green lines, with the corresponding percentages of predictions within each environment being listed in Table 4. Predicted error bars of all four models exhibit the correct statistical properties, with the GP log D error predictions being closest to the ideal distribution. The results presented for the GP model stem from a "blind test" of the final model delivered to Bayer Schering Pharma.[4−8] The remaining algorithms have been evaluated *a posteriori*, after the experimental values for the validation set had been revealed.

In conclusion, using Bayesian models, ensemble models, or distance based approaches one can not only identify compounds outside of the models domain of applicability but also quantify the reliability of a prediction in a way that is intuitively understandable for the user.

4.5. Increasing Accuracy by Focusing on the Domain of Applicability. In section 4.3 we presented statistics obtained by applying our models to all compounds in the respective test sets, without considering the models' domain of applicability. In section 4.4 we have evaluated methods for quantifying the confidence in predictions, and found that this can be achieved in a reliable way. Therefore it should be possible to increase model performance by focusing on more confident predictions or, in other words, on compounds clearly inside the domain of applicability.

In Figure 9 we present a histogram-like bar plot obtained in the following way: We assign compounds to bins based

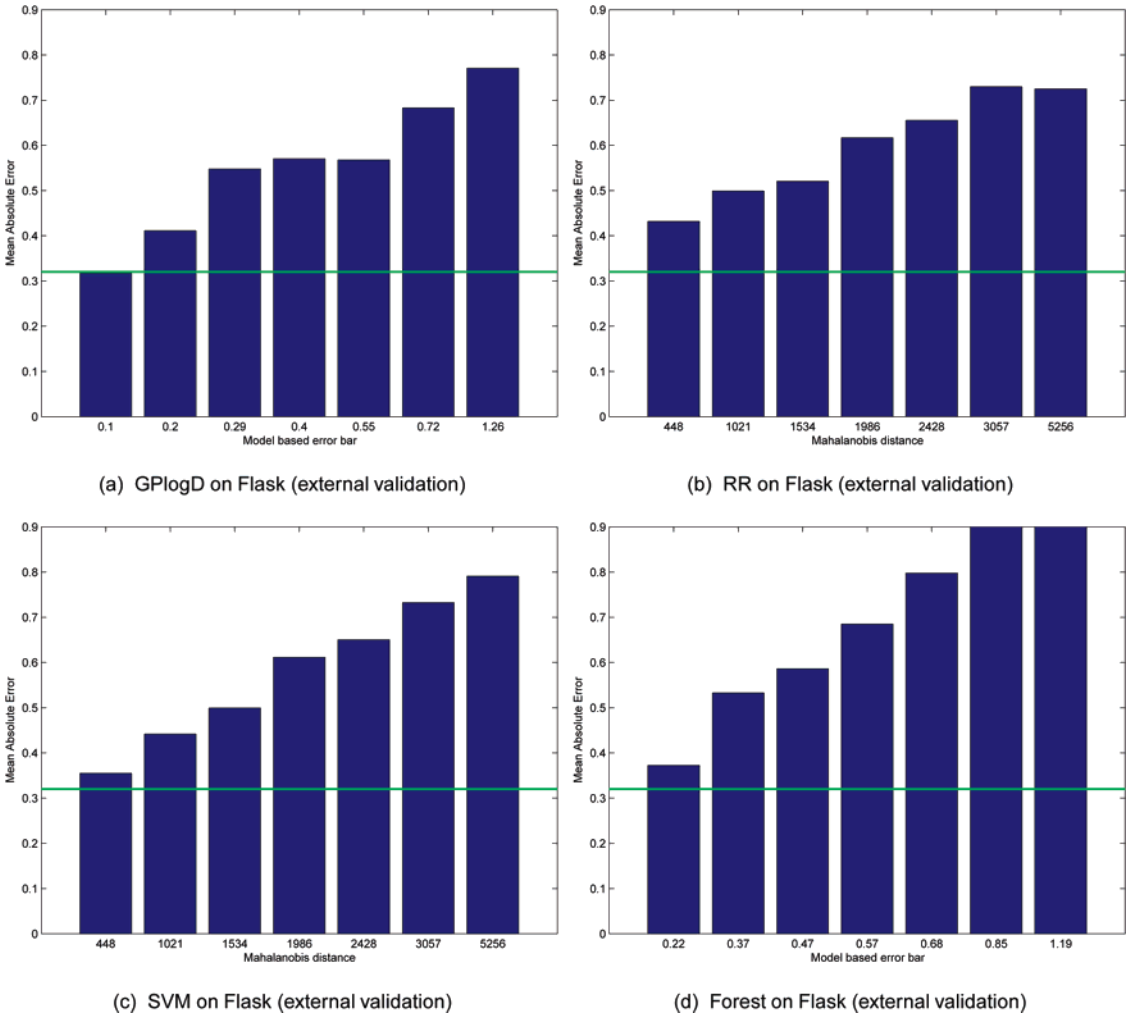


Figure 9. Mean absolute error achieved when binning by the model based error bar (in the case of the GP and the random forest) respectively the Mahalanobis distance to the closest point in the training set (linear ridge regression and support vector machines do not provide error bars). Each bin contains one-seventh (1000 compounds) of the in-house validation set. Corresponding numbers can be found in Table 5.

on the confidence in the prediction, i.e., the model based error bar (GP and random forest) or distance to training points (for ridge regression and SVM), such that each of the seven bins contains 1000 compounds (one-seventh of the in-house validation data). Within each bin (representing a different degree of confidence in the predictions), we compute the mean absolute error. For each algorithm tested, the mean absolute error decreases, as compounds in bins with higher confidence are considered. In the case of the GP model, the mean absolute error decreases from 0.55 to 0.42, when focusing on the 3000 compounds with the lowest predicted error bars. When focusing on the 1000 compounds with lowest predicted error bars, the mean absolute error can even be reduced to only 0.32 log unit.

Table 5. Mean Absolute Error Achieved When Binning by the Model Based Error Bar (for GP and Random Forest) Respectively the Mahalanobis Distance to the Closest Point in the Training Set (Linear Ridge Regression and SVM, Since These Methods Do Not Provide Model Based Error Bars)[a]

error bar (av in bin)	0.10	0.20	0.29	0.40	0.55	0.72	1.26
MAE GP	0.32	0.41	0.55	0.57	0.57	0.68	0.77
error bar (av in bin)	0.22	0.37	0.47	0.57	0.68	0.85	1.19
MAE (forest)	0.37	0.53	0.59	0.69	0.80	0.95	1.24
distance (av in bin)	448	1021	1534	1986	2428	3057	5256
MAE (RR)	0.43	0.50	0.52	0.62	0.66	0.73	0.73
MAE (SVM)	0.35	0.44	0.50	0.61	0.65	0.73	0.79

[a] Bins were chosen such that each contains one-seventh (around 1000 compounds) of the in-house validation set. A graphical representation of this information can be found in Figure 9.

In conclusion, focusing on confident predictions, i.e., compounds within the domain of applicability, allows us to achieve more accurate predictions than we found when validating models on the whole in-house validation set (Table 3). The previously observed decrease in performance relative to the cross-validation on the training data can therefore be avoided.

5. Summary

We presented results of modeling lipophilicity using the Gaussian process methodology on public and in-house data. The statistical evaluations show that the prediction quality of our GP models compares favorably with four commercial tools and three other machine learning algorithms that were applied to the same sets of data. The positive results achieved with the model on in-house drug discovery compounds are reconfirmed by a blind evaluation on a large set of measurements from new drug discovery projects at Bayer Schering Pharma.

It should be noted that GP models not only are capable of making accurate predictions but also can provide fully automatic adaptable tools: Using a Bayesian model selection criterion allows for retraining without user intervention whenever new data becomes available. As a further advantage for everyday use in drug discovery applications, GP models quantify their domain of applicability in a statistically well founded manner. The confidence of each prediction is quantified by error bars, an intuitively understood quantity. This allows both (1) increasing the average accuracy of predictions by focusing on predictions that are inside the domain of applicability of the model and (2) judging the reliability of individual predictions in interactive use.

A. Appendix: Measuring log D_7 using HPLC

High performance liquid chromatography (HPLC) is performed on analytical columns packed with a commercially available solid phase containing long hydrocarbon chains chemically bound onto silica. Chemicals injected onto such a column move along it by partitioning between the mobile solvent phase and the hydrocarbon stationary phase. The chemicals are retained in proportion to their hydrocarbon−water partition coefficient, with water-soluble chemicals eluted first and oil-soluble chemicals last. This enables the relationship between the retention time on a reverse-phase column and the *n*-octanol/water partition coefficient to be established. The partition coefficient is deduced from the capacity factor $k = (t_r - t_0)/t_0$, where t_r is the retention time of the test substance and t_0 is the dead time, i.e., the average time a solvent molecule needs to pass the column. In order to correlate the measured capacity factor k of a compound with its log D_7, a calibration graph is established. The partition coefficients of the test compounds are obtained by interpolation of the calculated capacity factors on the calibration graph using a proprietary software tool "POW Determination".

A.1. Apparatus and Materials. Experiments are carried out following the *OECD Guideline for Testing of Chemicals No. 117*. A set of 9 reference compounds with known log D_7 values selected from this guideline is used.

HPLC:	Waters Alliance HT 2790 with DAD- and MS-detection
column:	Spherisorb ODS 3 μm, 4.6 $\times$ 60mm
mobile phase:	methanol/0.01 m ammonium acetate buffer (pH 7) 75:25
dead time compound:	formamide, stock solution in MeOH
test compounds:	10 mM DMSO stock
reference compounds (stock solutions in MeOH):	acetanilide, 4-methylbenzyl alcohol, methyl benzoate, ethyl benzoate, naphthalene, 1,2,4-trichlorobenzene, 2,6-diphenylpyridine, triphenylamine, DDT

Acknowledgment. The authors gratefully acknowledge DFG grant MU 987/4-1 and partial support from the PASCAL Network of Excellence (EU #506778). We thank Vincent Schütz and Carsten Jahn for maintaining the PCADMET database and Gilles Blanchard for implementing the random forest method as part of our machine learning toolbox.

MP0700413