A Probabilistic Approach to Classifying Metabolic Stability

The original publication is available at pubs.acs.org
http://dx.doi.org/10.1021/ci700142c

Anton Schwaighofer[1†], Timon Schroeter[‡ †], Sebastian Mika[¶], Katja Hansen[‡ †], Antonius ter Laak[‖], Philip Lienau[‖], Andreas Reichel[‖], Nikolaus Heinrich[‖], Klaus-Robert Müller[‡ †]

[†] Fraunhofer FIRST, Kekuléstraße 7, 12489 Berlin, Germany
[‡] Technische Universität Berlin, Department of Computer Science, Franklinstraße 28/29, 10587 Berlin, Germany
[¶] idalab GmbH, Sophienstraße 24, 10178 Berlin, Germany
[‖] Research Laboratories of Bayer Schering Pharma, Müllerstraße 178, 13342 Berlin, Germany

Abstract

Metabolic stability is an important property of drug molecules that should—optimally—be taken into account early on in the drug design process. Along with numerous medium or high throughput assays being implemented in early drug discovery, a prediction tool for this property could be of high value. However, metabolic stability is inherently difficult to predict, and no commercial tools are available for this purpose. In this work we present a machine learning approach to predicting metabolic stability, that is tailored to compounds from the drug development process at Bayer Schering Pharma. For four different *in vitro* assays, we develop Bayesian classification models to predict the probability of a compound being metabolically stable. The chosen approach implicitly takes the "domain of applicability" into account. The developed models were validated on recent project data at Bayer Schering Pharma, showing that the predictions are highly accurate and the domain of applicability is estimated correctly. Furthermore, we evaluate the modelling method on a set of publicly available data.

[1]Corresponding author e-mail: anton@first.fraunhofer.de

1 Introduction

In the drug development process, 50% of the failures[1] in late development stages are due to an unfavorable ADMET profile (Absorption, Distribution, Metabolism, Excretion & Toxicity). A lot of research effort has been invested in obtaining *in silico* predictions for properties that are closely related to the ADMET profile, like aqueous solubility[2,3] or lipophilicity.[4,5] Commercial tools are available for a number of properties relevant to the drug development process. Along with the optimized high and medium throughput methods in early pharmacokinetics, predictive tools for metabolic stability are called for. For this property, however, building general-purpose models that are accurate over a large number of structural classes is virtually impossible, since a plethora of not fully understood mechanisms is involved in metabolizing a chemical compound for example in the human liver.

Furthermore, experimental protocols and assays can vary widely, such that tool predictions and actual experimental outcome may exhibit large differences. Only when the classes of compounds are limited, one can hope to establish Quantitative Structure Property/Activity Relationship (QSPR) models that reliably predict a property like metabolic stability. To date, there is only little published work about such approaches,[6,7] despite development efforts by various pharmaceutical companies.

In this work we investigate the use of different regression and classification methods to develop assay-specific models for metabolic stability. The approach we finally chose was a Bayesian method, namely nonlinear classification with Gaussian Process priors. Each of the models (for human, male mouse, female mouse, and male rat) predicts the probability of a compound being metabolically stable in the *in vitro* assay. Models are based on experimental data collected in the drug development process at Bayer Schering Pharma, where the percentage of compound remaining after incubation with liver microsomes for 30 min is measured. During model fitting, the statistical fine structure of the molecular descriptor space is learned, allowing the model to predict the stability for unseen compounds.

A particular strength of our approach is to provide an implicit check for the "domain of applicability". The model is fully probabilistic, and outputs the probability (between 0 and 1) for the compound to be stable. If the model is queried outside its range of expertise, or in areas of conflicting data, a model output close to 0.5 indicates that it is equally likely for the compound to be stable or unstable. A blind test of the final models confirmed this behavior for compounds from new projects and shows that performance clearly increases when focussing on the

compounds that can be predicted with high confidence by the model. Also, the developed models are, in a statistical sense, well calibrated:[8] the predicted probability does correlate with the empirical probability for a compound being stable. The absolute value of the model output thus carries an intuitive meaning.

1.1 Background: Machine Learning

Machine learning subsumes a family of algorithmic techniques with a solid statistical foundation that aim to find reliable predictions by inferring from a limited set of experimental data. In computational chemistry, this could be measurements from which we seek to derive, e.g., a predictor for the property "metabolic stability", or for the water solubility of a compound.[2,3] A large variety of techniques has been developed in the machine learning and statistics communities to account for different prediction tasks and application areas.[9,10]

The use of machine learning techniques for computational chemistry is of course not new. Neural Networks,[11,12] for example, have a long history in computational chemistry.[13,14] Recently, the successful application of Support Vector Machines[10,15,16] in many domains has also initiated their use for predicting properties of chemical compounds.[17–19]

In virtually all application scenarios for such QSAR models, it is a key requirement to provide confidence estimates.[20,21] Users of QSAR models need to be able to assess whether they can trust the predictions made by the model. With that in mind, SVMs are not ideal for applications in computational chemistry, since they can not provide theoretically well founded confidence estimates (only heuristics such as "Platt scaling"[22] are available).

We find that Bayesian modeling approaches are more suitable for computational chemistry. In a Bayesian approach, one strives to treat all quantities involved in model building as uncertain, and describe them via probability distributions. In such a framework, the model output is also a probability distribution, which includes the required confidence estimates. Bayesian approaches can be applied to different forms of models. In this work, we use a specific Bayesian nonlinear classification model, a Gaussian Process (GP) model.[23,24]

The authors have demonstrated in recent work how Gaussian Process regression models can be used to accurately predict the water solubility[2,3] and the lipophilicity[4,5] of drug discovery molecules. The main advantages of GP models in this context are error bars for each individual prediction, and a fully automatic procedure for model selection that allows for simple re-training of the model whenever new data becomes available. In this paper, we show how a re-

lated approach can be used to predict the metabolic stability of drug candidates, using a Bayesian classification method. Again, our focus is not only on accurate predictions, but also on *meaningful probabilistic outcomes*.

1.2 Background: Metabolic Stability

Measuring the metabolic stability is part of the first *in vitro* studies on drug discovery molecules in the pharmaceutical industry, aiming at predicting the *in vivo* pharmacokinetics.[25] This, in turn, determines important factors like how much and often the drug will need to be given. Metabolic stability is considered one of the properties of a compound that are critical for its market potential.[25]

Metabolic stability provides information about the extent of metabolic clearance of a compound. When administered orally, the elimination during the first passage through the liver can be calculated (first pass effect), which—under certain assumptions—leads to an estimate of the oral bioavailability of the compound. In general, the process of metabolizing a compound can be divided into phase I and phase II metabolic enzymes.[26] Investigations from this work were perfomed using liver microsomal preparations, hence cover only oxoreductive phase I metabolism, involving enzymes like cytochrome P450, flavine monooxidases, esterases and epoxide hydrolases. A overview of the involved enzymes and processes depending on the biological matrix can e.g. be found in the work of Cashman *et al*.[27] Details on the correlations between *in vitro* and *in vivo* clearance are given by Masimirembwa *et al*.[25]

The metabolism of a compound depends on a large number of variables related to both the chemical itself and the biological system. Even when the metabolism can be attributed to a specific enzyme (for example, one in the cytochrome P450 family), modeling can be difficult due to the promiscuous nature of the enzyme.[26]

A large variety of approaches has been developed in order to address the issue of metabolic stability by in silico methods. Two recent reviews[26,28] give an overview. Following the nomenclature of Gombar *et al*.,[28] the approaches can be grouped as follows:

- Rule-based systems apply a large number of programmed bio-transformations to the molecule, in order to directly predict metabolites.

- Oxidation by a CYP enzyme is one of the most common early processes in metabolism. One can therefore estimate the likelihood of applying a one-electron oxidation to each site in the molecule, and thus identify the metabolically labile “hot spots”.

- By predicting substrate binding, it is possible to estimate whether a molecule can dock into the active site of the (CYP) enzyme.

- Prediction of metabolism inhibition and enzyme induction can help to estimate potential drug-drug interactions, by estimating whether a compound can change the pharmacokinetics of co-administered drugs.

- The last class of approaches aims at directly predicting the overall metabolic stability, via a descriptor-based statistical modeling of experiments that measure stability in *in vitro* assays.

The work presented in this paper falls into the last category.

To date, there is only little published work on models to directly predict the metabolic stability. Bursi *et al*.[29] present results on a small data set of 32 steroidal androgens, modeled using a decision tree approach. As the authors note, the generalization ability seemed to be rather poor (around 50% for classifying stable versus non-stable). A set of 130 calcitriol analogs is used by Jensen et al.[7] to build partial least squares (PLS) models, combined with methods for feature selection. On a set of 20 validation compounds, an accuracy of 85% was achieved. Shen et al.[6] use modified nearest neighbor approach, including a distance-based heuristic to estimate the domain of applicability[20] on a set of 631 compounds. The most recent reference describes the process of building metabolic stability models at Eli Lilly.[28] Also, the application of the developed models in new drug discovery projects is described.

In our work, we aim at improving upon the previous work in the following aspects:

- The modeling approach should correctly take the domain of applicability (DOA) into account, and thus detect when the model is queried outside its range of data. Many of the currently used measures for the DOA are quite difficult to interpret.[3] Thus, we wish to achieve a measure for the DOA that conveys an intuitive meaning to the user.

- Ideally, the process of model building should be fully automatic and not require user intervention for choosing parameters. Thus, whenever new experimental data becomes available, an improved model can be constructed easily.

2 Methods and Data

2.1 Methodology overview

For each molecule, the 3D structure of one conformation is predicted using the program Corina.[30] From the 2D structure and the predicted 3D structure, a set of Dragon[31] descriptors is generated. Based on the descriptor and measurements of the percentage of each compound remaining after incubation with liver microsomes for 30 min, a Gaussian Process classification model is fitted. When applying this model to a previously unseen compound, descriptors are calculated as described above and passed on to the trained model. The model in turn predicts the probability that the compound in question is metabolically stable, i.e., after incubation for 30 min, more than 50% of the compound remains.

2.2 Experimental Protocol

The experimental protocol used to measure in-vitro metabolic stability is as follows: Liver microsomes were adjusted to a cytochrome P450 concentration of 0.2 μM. Sodium phosphate buffer was used at 100 mM at pH 7.4. The cofactors were glucose-6-phosphate (8 mM), MgCl (4 mM), NADP (0.5 mM), and G-6-P dehydrogenase (1 IU/ml).

Compounds were tested at 3 μM. Two samples were incubated at 37 °C and constant shaking for 30 min and were stopped by addition of icecold methanol (1+1). 0 min samples were stopped by icecold methanol before adding the test compound. All samples were stored in the freezer (-20 °C) over night and thawed during centrifugation at 2000 g before taking an aliquot for HPLC-UV/Vis analysis.

Experimental outcome is the per-cent recovery at 30 min, given as peak area of the parent compound in relation to the 0 min value. Testosterone was used as metabolic reference compound at 100 μM. All incubations were perfomed in duplicate.

2.3 Experimental Data

The Bayer Schering Pharma in-house data used for model building includes measurements of metabolic stability using microsomes of human, female mouse, male mouse and male rat liver. The number of measurements for each assay is listed in Table 1. After model selection and building, a set of compounds from recent drug

Assay	# experimental data	# data for model building
Human	2196	1931 (1172 stable, 759 unstable)
Mouse female	1268	1134 (560 stable, 574 unstable)
Mouse male	1022	904 (408 stable, 496 unstable)
Rat male	1647	1459 (758 stable, 701 unstable)

Table 1: Number of available experimental data for each assay. The middle column lists the number of raw data per assay, the right column lists the number of data after merging multiple measurements and removing outliers

Assay	# experimental data	# data for blind test
Human	700	631 (361 stable, 270 unstable)
Mouse female	358	326 (139 stable, 187 unstable)
Mouse male	194	183 (98 stable, 85 unstable)
Rat male	290	264 (148 stable, 116 unstable)

Table 2: Number of blind test data for each assay. The middle column lists the number of raw data per assay, the right column lists the number of data after merging multiple measurements and removing outliers. Experimental values for these compounds were only available to FIRST/idalab after model building had been completed

discovery projects were used as blind test data for the final models. These data are summarized in Table 2.

2.4 Multiple Measurements

For a number of compounds, several experimental data are available for a specific assay (for example, if a compound has been measured several times in mouse liver microsomes). Thus, it is necessary to fuse multiple measurements into a consensus value.

The set of measurement values is noisy and contains large outliers. Compounds where the spread of experimental values is larger than 30% were removed completely. To merge multiple measurements into a consensus value, we proceed as follows: We consider the histogram of measured values. Such a histogram can be characterized by two quantities, the spread of experimental values (y-spread) and the spread of the bin heights (z-spread).

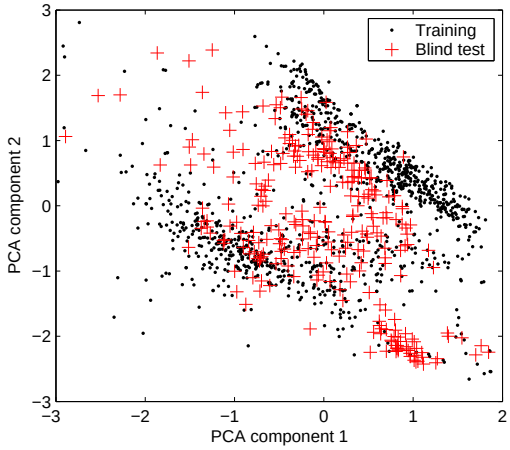
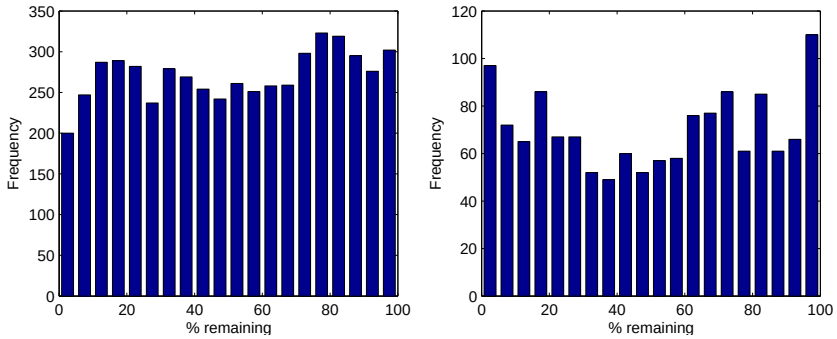


Figure 1: Visualization of training and blind test data for the assay “mouse female” by principal component analysis (PCA). The blind test data covers recent projects, and thus follows a distribution that is different from the training data



(a) Histogram of experimental values for training data

(b) Histogram of experimental values for blindtest data

Figure 2: Histograms of raw experimental values for training and blindtest data, with all assays pooled

Several cases arise regularly: For small *y*-spreads (all measured values are similar), taking the median value is the most sensible choice. On the other hand, large *y*-spread with large *z*-spread hints at outliers. In such a case, we use the median of the values in the higher of the two bins as the consensus value. The worst case is given by two far apart bins of equal height (high *y*-spread and zero *z*-spread). In this case we omit the compound altogether, since we have equally strong evidence for the conflicting measurements. Our analysis suggests that 25% is a suitable threshold between small and large spreads.

2.5 Training and Validation Setups

To build machine learning models from the data sets described in Sec. 2.2, we used the following protocol:

Training: In order to choose the right descriptors, model structure, and also to estimate model performance, we used 2fold cross-validation on the training data for each assay. The training data is split in two halves. A GP classification model is built on the first half, and evaluated on compounds in the second half. This is repeated with the roles of the two halves exchanged. The overall procedure is then repeated 5 times with different random splits. Thus, in each of the 5 runs, model predictions for the full training set are generated, where each prediction is an out-of-sample prediction, made by a model that has not seen the particular compound in its training data. Based on the cross-validation performance, optimal model settings were chosen, and used to build final models from all training data.

Blind test: The final models were used to make predictions for a set of blind test data compiled at Bayer Schering Pharma. Initially, the experimental data for the blind test data were not available to the modelling team. They were revealed after the model performance had turned out to be sufficient.

2.6 Molecular descriptors

Initially, we used the full set of 1664 Dragon descriptors. These include, among others, constitutional descriptors, topological descriptors, walk and path counts, eigenvalue-based indices, functional group counts and atom-centered fragments. A full list of these descriptors including references can be found online.[31]

After a first modeling stage using all descriptors, it turned out that a large number of descriptors can be omitted without significantly impacting the models performance. In particular, it was possible to omit the computationally most expensive blocks, Dragon blocks 5 and 13. The models described hereafter are based on all or subsets of the descriptors from Dragon blocks 1, 2, 6, 9, 12, 15, 16, 17, 18, 20. (Most of these Dragon descriptors only depend on the 2D structure of the molecule, while some actually take 3D information into account.) In Sec. B we describe the influence of different strategies for selecting a set of relevant descriptors (feature selection) on ranking quality and on quality of confidence estimates.

2.7 Choice of Models

Based on the available experimental data (per-cent recovery after 30 min) it is possible to build either regression models that predict the per-cent recovery, or to build classification models that predict whether a compound is stable (recovery $> 50\%$). We investigated both strategies, a quick summary of the results is given in Sec. B.

With the actual application scenario at Bayer Schering Pharma in mind, we decided to choose a Gaussian Process classification (GPC) model. Classification is appropriate here as the model is typically used in early development stages, where a distinction between compounds with moderate and high stability does not (yet) need to be made. Another aspect is that the output of a GPC model can readily be used for compound ranking. Since GPC is a Bayesian method, the output incorporates already a measure for the prediction uncertainty, or domain of applicability (DOA). Thus, compounds that are stable and in the DOA are ranked before those that are outside the DOA. If we were to use a regression model for ranking, we would need to fuse the model output (per-cent recovery) and a measure for the DOA into a single number that determines the compound ranking. Another point in favor of GPC models is the possibility for fully automatic model building (see the list of criteria at the end of Sec. 1.2).

2.8 Gaussian Process Models

We start here with a short overview of the key ideas of Gaussian Process classification. For an in-depth treatment we refer to a recent book.[23]

Building a GP classification model follows, in principle, methods such as logistic (linear) regression:

- We introduce a "latent" (unobserved) function that could potentially model the dependence of metabolic stability from the descriptor. In the case of logistic regression, this latent function depends linearly on the descriptors. In the case of a GP classifier, a nonlinear function is used that can be described by a Gaussian stochastic process. Mind that these functions are never actually observed, and will later be "removed" by an integral operation.

- The latent function is then transformed nonlinearly by a function that maps from the real numbers to a probability in the range between 0 and 1. This transformation plays a role similar to the transfer functions in neural networks, or the link function in logistic regression.

These ideas and the process of inference are summarized in Figure 3. It is important to note here that we use random variables as latent functions, and thus also obtain a random quantity after the (deterministic) transformation. Only from such a model, we can expect to obtain a meaningful quantification of the classification uncertainty.

2.8.1 Modeling

We consider data for N compounds, each described by a vector of descriptors $\mathbf{x}_1, \ldots, \mathbf{x}_N$. Each compound is assigned to either of two classes, which we label $+1$ and -1. The class assignment is denoted by $y_1, \ldots, y_N$, with $y_i \in \{+1, -1\}$. For classification, our goal is to model the probability distribution of the class label y for a given data point, $p(y \,|\, \mathbf{x})$.

For the sake of model building, we introduce a latent (unobserved) function. As our *a priori* information about this function, we assume that it follows a Gaussian stochastic process. The latent function is then mapped ("squashed") through a transfer function Φ, that gives an output in the range of $[0,1]$. As a transfer function, we choose the Gaussian cumulative distribution function, $\Phi(z) = \int_{-\infty}^{z} \mathcal{N}(x;0,1)dx$. Learning with such a model essentially amounts to inferring the behavior of the latent function f, or solving a "hidden" regression problem.

This gives as the basic classification model

$$p(y = +1 \,|\, \mathbf{x}) = \Phi(f(\mathbf{x})), \tag{1}$$

equivalently this likelihood term can be written as $p(y \,|\, f(\mathbf{x})) = \Phi(y f(\mathbf{x}))$. Furthermore, we assume that $f(\mathbf{x})$ follows a Gaussian Process, described by a mean function (which we assume to be zero) and covariance function k. By assuming a

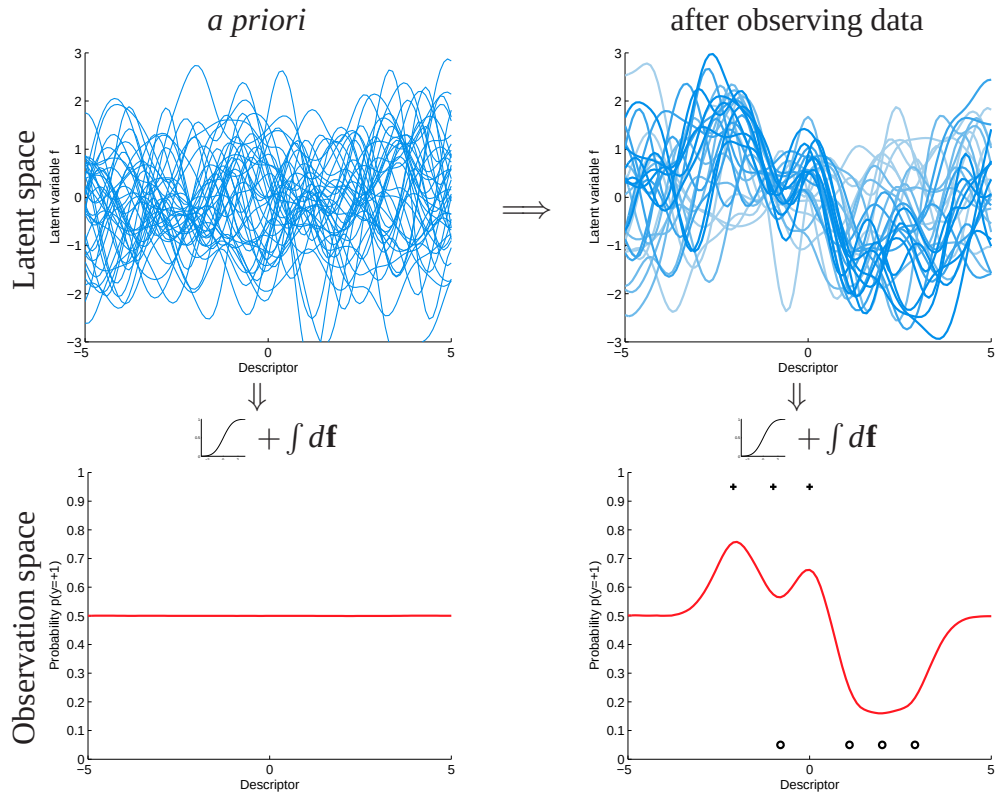


Figure 3: Classification with Gaussian Process priors for the latent f. *Left:* 40 samples from a Gaussian Process prior over functions, each plotted as $y = f(x)$. For illustration, we only consider functions for one-dimensional input x. The probability of membership in class $+1$ is obtained by squashing each of these functions through the transfer function and averaging by an integral operation. Without data, the probability is 0.5 throughout. *Right:* We observe seven data points, marked by $+$ for class $+1$, and $\circ$ for class -1. We weight each function (in latent space) according to the degree to which they can explain the data (in observation space). Well matching functions are shown in dark shading, poorly matching functions in light shading. After transforming and integrating, we obtain the learned class membership information.

GP, we can reduce the burden of dealing with a probability distribution over functions to a (Gaussian) probability distribution on function values at the points of interest, that is, the points in descriptor space corresponding to the experimental data and the test compound on which the model is evaluated. For each of the N observations at hand, $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, we have one latent function value. Denote the latent function values at the N training data points as $\mathbf{f} = (f(\mathbf{x}_1), \ldots, f(\mathbf{x}_N))$. These follow a joint multivariate Gaussian distribution,

$$p(\mathbf{f}|X) = \mathcal{N}(0, K), \tag{2}$$

where the $N \times N$ covariance matrix K can be computed by pairwise evaluations of the covariance function k, with $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$.

With these prerequisites, we can use Bayesian inference to infer the distribution of the latent function on a test point, denoted by $\mathbf{x}_*$. This inference step is summarized in the appendix Sec. A. In the inference step, we estimate the latent function value f_* on the test point, given all the observed (training) data. This *a posteriori* belief about f_* is described by the probability distribution $p(f_*|X, \mathbf{y}, \mathbf{x}_*)$. To obtain the class membership probability, we average (integrate) over the transformed function, weighted by the degree of belief:

$$p(y = +1|\mathbf{x}_*) = \int \Phi(f_*) p(f_*|X, \mathbf{y}, \mathbf{x}_*) df_*. \tag{3}$$

With our particular choice of transfer function Φ and a Gaussian distribution for $p(f_*|X, \mathbf{y}, \mathbf{x}_*)$, this integral can be solved analytically.

2.8.2 Predictions

In the inference step, we obtain the distribution of latent function values, dependent on the location of the test point $\mathbf{x}_*$. Effectively, this describes what the upper right graphics in Figure 3 looks like. It turns out that the *a posteriori* distribution of latent function values on the test point $\mathbf{x}_*$ follows a Gaussian distribution with mean $\bar{f}_*$ and variance var f_*,

$$\bar{f}(\mathbf{x}_*) = \sum_{i=1}^{n} \alpha_i k(\mathbf{x}_*, \mathbf{x}_i) \tag{4}$$

$$\text{var}\, f(\mathbf{x}_*) = k(\mathbf{x}_*, \mathbf{x}_*) - \sum_{i=1}^{n} \sum_{j=1}^{n} k(\mathbf{x}_*, \mathbf{x}_i) k(\mathbf{x}_*, \mathbf{x}_j) L_{ij} \tag{5}$$

From the inference step (see Sec. A) we obtain a vector $\mathbf{m}$ and a matrix S, which in turn allows us to compute the coefficients α_i by the matrix expression $\alpha = (K+S)^{-1}\mathbf{m}$. L_{ij} denote the elements of matrix $L = (K+S)^{-1}$.

From that, the final output of the GP classification model (the probability that a test compound falls into class $+1$) is given by

$$p(y_* = +1) = \Phi(\bar{f}(\mathbf{x}_*)/\sqrt{1+\mathrm{var}\, f(\mathbf{x}_*)}) \tag{6}$$

2.8.3 Adapting Parameters

In our model to predict metabolic stability, we use a covariance function of the form

$$k(\mathbf{x},\mathbf{x}') = \left(1 + \sum_{i=1}^{d} w_i (x_i - x_i')^2\right)^{-\nu} \tag{7}$$

(the "rational quadratic" covariance function[23]). $k(\mathbf{x},\mathbf{x}')$ describes the "similarity" (covariance) in the behavior of two compounds, given by their descriptor vectors $\mathbf{x}$ and $\mathbf{x}'$. The contribution of each descriptor to the overall similarity is weighted by a factor $w_i > 0$ that effectively describes the importance of the ith descriptor for the modeling task.

In order to set the weights w_i and the parameter ν, we consider a Bayesian criterion called the evidence (marginal likelihood), that is computed by averaging over all possible values for the latent function on the training data. It can be seen as a measure of how well the data can be explained by the current class of latent functions, irrespective of the (unknown) actual values for the latent function. This allows for a fully automatic optimization of the criterion with respect to the evidence, thus all of model parameters can be chosen without user intervention. A brief summary of this procedure is given in Sec. A.1.

3 Results

We evaluate the resulting modes with respect to the following criteria:

Quality of ranking: The main application area for the developed models at Bayer Schering Pharma is compound ranking. We will use receiver-operating-characteristics (ROC) curves to measure the quality of the ranking, results will be presented in Sec. 3.2.

Probabilistic output: We aim at intuitively understandable classifier outputs, and thus want to achieve a well calibrated[8] classifier. For a well calibrated system, a prediction "probability of 0.9 for being stable" means that 9 out of 10 such compounds should indeed be stable. Calibration is described in more detail in Sec. 3.3, along with the calibration curves for the final model.

When evaluating the ranking performance, it is important to take "don't know" predictions and experimental values into account in a meaningful way. Consider a compound with an experimental value of 48% recovery, and a classifier prediction "stable with probability 0.52". Shall we count that as a mistake? After all, we must expect that when repeating the lab experiment, the experimental value might as well be 40% or 60%.

Thus, when evaluating only the confident parts of a compound ranking, we exclude both compounds with an unclear experimental outcome (values around the stable/unstable threshold in the interval $[50-q, 50+q]$) and compounds with an unsure prediction (values around the "don't know" prediction in the interval $[0.5-r, 0.5+r]$). A graphical summary of the procedure is given in Figure 4. From the remaining compounds, ROC curves to assess the ranking performance can be computed as usual.

Subsequently, we will evaluate the ranking performance when considering all data ($q = r = 0$), when focussing on the moderately confident predictions ($q = 15$, $r = 0.15$) and when focussing on confident predictions ($q = 30$, $r = 0.3$).

3.1 Model Selection

We investigated a large number of modeling approaches, based on regression and classification, built with different parameter settings. The criteria to select between the models were ranking quality (measured by area under the ROC curve) and, when applicable, shape of the calibration curve. Both measures were evaluated in 2-fold cross-validation on the training data.

Model selection details are given in the appendix, Sec. B. It turned out that optimal results could be achieved by

- using all available descriptors without feature selection of any kind.
- including compounds that have an experimental value around 50% (the chosen threshold for stable compounds).

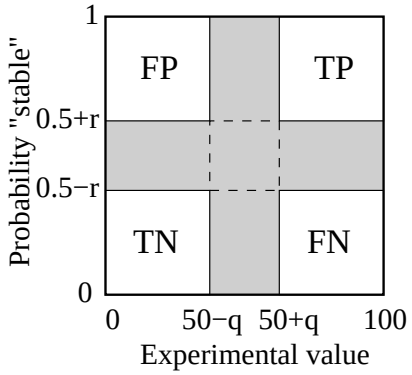


Figure 4: Evaluating a classifier rates when assuming uncertainty for both labels and predictions. Only compounds where the experimental value is far from 50%, and where the predicted probability is far from 0.5, actually contribute to the computation of TP (true positive), TN (true negative), FP (false positives), and FN (false negative) rates

The result on feature selection seems to contradict conventional wisdom, yet is in agreement with other work[32,33] on feature selection for QSAR modelling. As described in Sec. B, we found that, for most methods, the difference in ranking quality between filter-based feature selection and using all features is non-significant. This holds in particular for the kernel-based methods, where the number of descriptors is independent of the number of model parameters (except for GP methods with a covariance function that has a width parameter w_i for each descriptor in Eq. (7)). When also considering the confidence estimates, it turned out that stronger feature selection tends to make models over-confident. Since we wish to achieve both high ranking quality and reliable confidence estimates, we chose to not perform feature selection.

In the subsequent sections, we only list the performance achieved by the "final" model (GP classification, no feature selection) that is now implemented at Bayer Schering Pharma.

3.2 Ranking Performance

Figure 5 shows receiver-operating-characteristics (ROC) curves for the final models for each of the four assays, both when evaluated on the training data (2-fold cross-validation) and on the blind test data (predictions of the final model). A summary of the performance in terms of area under the ROC curve (AUC) is listed in

Assay	All data	Moderately confident		Confident	
	AUC	% of data	AUC	% of data	AUC
Human	85.0	58.2%	94.4	29.1%	99.4
Mouse female	83.2	51.7%	93.9	19.0%	98.2
Mouse male	82.7	50.6%	93.0	18.0%	98.5
Rat male	85.0	54.3%	94.4	24.1%	98.5

Table 3: Evaluating the ranking performance: Area (AUC) under the ROC curves shown in Figure 5 for **2-fold cross-validation on the training data**. In column "All data" we evaluate the performance on the full list of compounds, "Moderately confident" is the performance on the subset of data with $q = 15, r = 0.15$ (see Sec. 3), and "Confident" evaluates the subset of data with confident outcomes ($q = 30, r = 0.30$)

Assay	All data	Moderately confident		Confident	
	AUC	% of data	AUC	% of data	AUC
Human	71.8	50.7%	80.7	17.0%	92.8
Mouse female	69.0	59.5%	80.6	18.7%	95.0
Mouse male	83.5	55.2%	93.7	21.3%	100.0
Rat male	76.4	37.5%	93.8	15.9%	92.7

Table 4: Evaluating the ranking performance: Area (AUC) under the ROC curves shown in Figure 5 for the **blind test data**. The subsets of data with moderately confident and confident outcomes are defined as in Table 3

Table 3 and Table 4. In each case, we investigate the performance on all data, on the subset of data with moderate confidence for experimental outcome and prediction ($q = 15, r = 0.15$ in Figure 4), and on the subset of data with confident experimental value and prediction ($q = 30, r = 0.30$).

When comparing the results on the training data and on the blind test data (536 drug candidates from recent projects at Bayer Schering Pharma), we can observe a small drop in performance. Still, the performance remains promisingly high. Note that the blind test data stems from new projects, and thus follows a different distribution than the training data. Thus, the fraction of compounds inside the model's domain of applicability is smaller, leading to smaller number of compounds that can be predicted with high confidence.

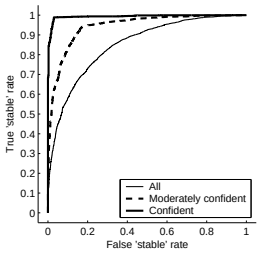
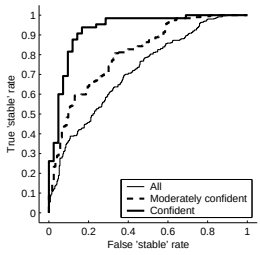
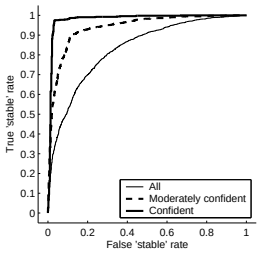
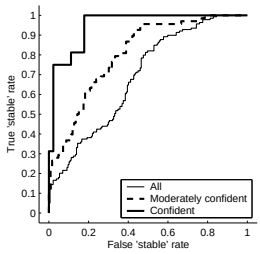
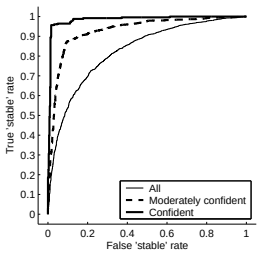
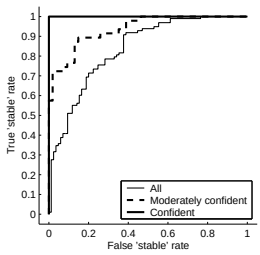
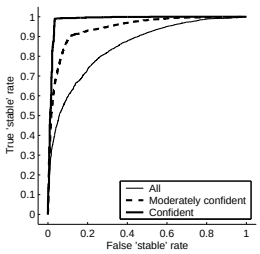
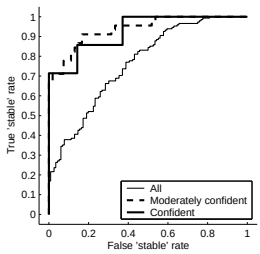
Assay	Training	Blind test data
Human m/f		
Mouse female		
Mouse male		
Rat male		

Figure 5: Evaluating the ranking performance: ROC curves for metabolic stability predictions in 2-fold cross-validation on the training set (left column) and on the blind test data (right column). We plot ROC curves for all data, for the subset of data with moderate confidence ($q = 15, r = 0.15$, see Sec. 3 and Figure 4) and for the subset of data with confident outcomes ($q = 30, r = 0.30$). A summary of the performance in terms of AUC (area under the ROC curve) is given in Table 3 and Table 4

3.3 Probabilistic Outputs

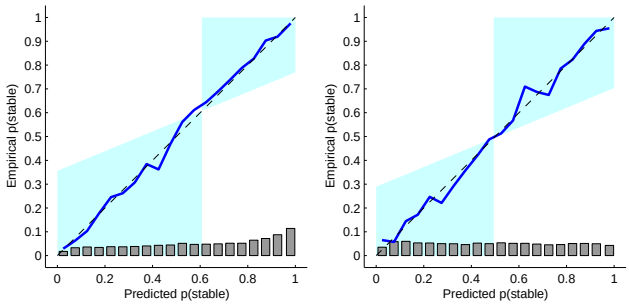
Our choice of Gaussian Process models was (amongst other criteria, see Sec. 2.7) guided by their beneficial properties when it comes to understanding the classification results: The model output is the probability that a particular compound belongs to the class of stable compounds. The closer the probability is to 0 or 1, the more certain the model is about its prediction. In addition to the actual classification performance, we thus also need to evaluate whether the predicted probability reflects the confidence of the result. Our goal is to achieve a well calibrated classifier:[8] For a well calibrated system, a prediction "probability of 0.9 for being stable" means that 9 out of 10 such compounds should indeed be stable. When using the model output for compound ranking, calibration has an intuitive counterpart: Among the compounds with high probability of being stable, there should be a larger fraction of compounds that actually are stable, than among the compounds with low probability of being stable.

We evaluate the calibration property by means of a calibration curve. Here, we consider groups of compounds for which the predicted probability p_{pred} is in bins centered around $[0.1, 0.2, \ldots, 0.9]$. For each group, we compute the within-bin fraction of stable compounds, p_{emp}. Ideally, the within-bin fraction of stable compounds should be 1 out of 10 (for the bin at $p_{\text{pred}} = 0.1$) up to 9 out of 10 (for the bin at $p_{\text{pred}} = 0.9$). In the calibration curve, we plot predicted probability p_{pred} on the x-axis versus empirical probability p_{emp} on the y-axis. Ideally, the result should be a diagonal line.

It should also be noted here that classifier performance and calibration are antagonist quantities: An error free classifier results in a poor calibration curve (a horizontal line at $p_{\text{emp}} = 0$ for all unstable compounds, then a horizontal line at $p_{\text{emp}} = 1$ for all stable compounds).

The calibration curves for the training data, evaluated in 2-fold cross-validation, are listed in Figure 6. All the curves show very good agreement between predicted and empirical probabilities. The according plots for the blind test data are shown in Figure 7. The curves for the models "Human" and "Rat male" show acceptable agreement between predicted and empirical probabilities, with the model for "Rat male" being slightly over-confident for some compounds that are correctly predicted to be unstable. The curve for "Mouse female" shows non-optimal behavior in the regions around $p_{\text{pred}} = 0.35$ and $p_{\text{pred}} = 0.7$. This is mainly due to a cluster of highly similar compounds that are all falsely predicted to be stable.

As a last remark, note that pseudo-probabilities can also be computed from methods such as SVMs, by fitting a sigmoid function to the SVM output.[22] How-



(a) Human m/f (b) Mouse female

(c) Mouse male (d) Rat male

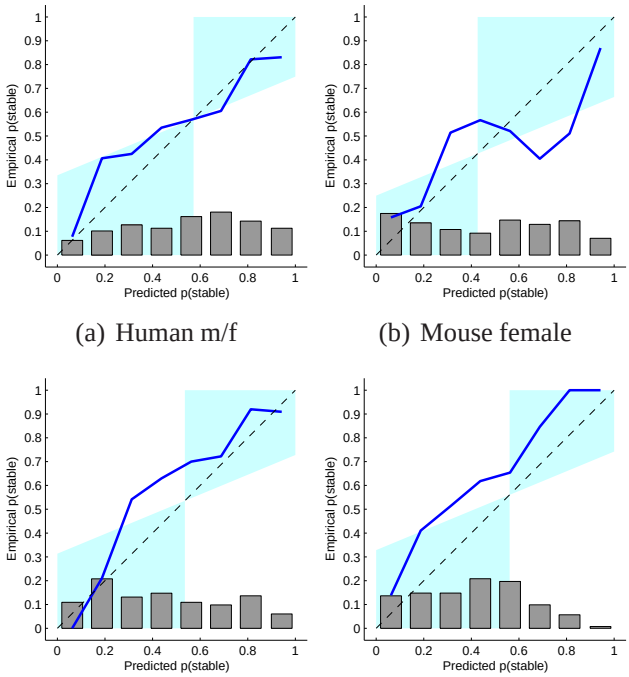
Figure 6: Calibration curves (see Sec. 3.3) of metabolic stability predictions on the training set, evaluated in 2fold cross-validation. All curves show excellent agreement between predicted and empirical probabilities. The small histogram bars show the relative frequency of compounds that attain a classifier output in the respective bin

ever, this comes at the price of having to sacrifice a part of the data only for fitting the parameters of the sigmoid functions.

4 Comparison With Existing Work

To our knowledge, only two publications[6,7] deal with directly modeling metabolic stability. Shen *et al.*[6] used in-house data from GSK, whereas Jensen *et al.*[7] used a set of data that is publicly available. In the following section, we test our modelling method on the data provided by Jensen *et al.* They structured their work as follows:

1. A set of 130 compounds was randomly split into a training set (87 com-



(a) Human m/f (b) Mouse female

(c) Mouse male (d) Rat male

Figure 7: Calibration curves of metabolic stability predictions on the blind test data. See Sec. 3.3 for further discussion

pounds) and a validation set (42 compounds). Four outlying compounds were removed from the training set, and one compound was removed from the validation set by visual inspection. Initial experiments were done in cross-validation on the training set, followed by a performance evaluation on the validation set. Feature selection was done in 5 different ways, each time followed by building a model using partial least squares (PLS) regression.

2. Using the five feature selection techniques, models were built using all 125 compounds from step one, and used to predict the metabolic stability of 240 compounds for which no measurements existed.

3. From this set of 240 new compounds, 20 compounds were selected where the agreement of the five models built in step two was largest. The metabolic stability of these compounds was measured and compared to the predictions of the "consensus model" built from the five regression models.

To facilitate a fair comparison with the method used by Jensen *et al*., we followed the above steps as closely as possible. Jensen and co-workers evaluate their models in terms of the root mean square error RMSE or as a 3-class ordinal regression task (low, medium, and high metabolic stability). Both metrics are not suitable for use with the GP classification model we had used on the in-house data provided by BSP, we thus chose a GP regression model. Also, we used the same descriptor set that Jensen *et al*. had used.

Jensen *et al*. investigated only one single random split of their initial 125 (130 compounds minus 5 outliers) compounds into training and validation set. Their models yield an RMSE between 21 and 16 on the validation set. On the same single split a GP model yields an RMSE of 20.3. To find out whether the split used by Jensen *et al*. is particularly "easy", we generated 100 new random splits, built models for each split, and found an average RMSE of 21.0, with a standard deviation of 1.12. We conclude that the performance on the single split chosen by Jensen *et al*. is better that one would expect on average, but still within one standard deviation as calculated from 100 random splits.

Finally, we trained a model on the whole set of 125 compounds and applied it to the 20 compound external validation set used by Jensen *et al*. A scatterplot of predicted stability vs. measured stability can be found in Figure 8. Overall, we achieve an RMSE of 12.4.

Jensen and co-workers report that their model makes three mis-predictions out of the 20 compound test set (with a slightly optimistic interpretation of mea-

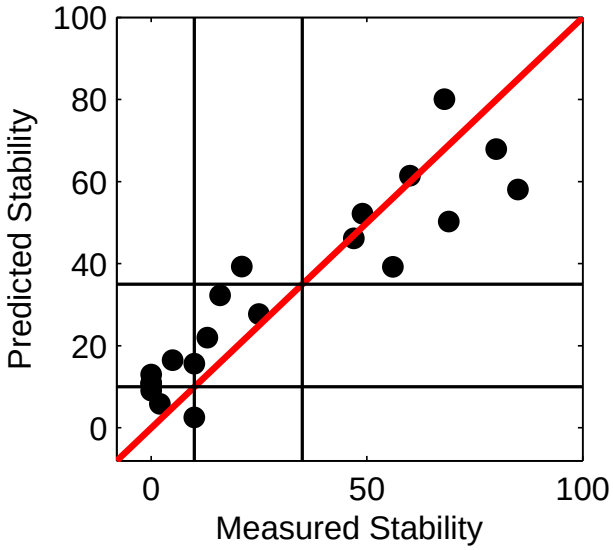


Figure 8: Scatterplot for predicted stability vs. measured stability on Jensen's[7] 20 compound external validation set. The vertical and horizontal lines correspond to the regions of low, medium, and high metabolic stability that were used in the original work

surement uncertainty, that is, the model prediction is assumed correct when it falls within experimental value plus/minus experimental standard deviation). This evaluation was done with a 3-class ordinal regression model that predicts low, medium, or high stability. To allow a comparison with that, we can partition the predictions of the GP regression model into 3 regions, using the same thresholds that Jensen *et al*. had used. When now counting errors the same way as Jensen *et al*. did, we see that the Gaussian Process model mis-predicts only one out of the 20 validation set compounds (compound M, experimental S9 of 5, but predicted as 17). The mis-predictions for other compounds are still within the intervals given by the experimental measurement uncertainty (Jensen *et al*. assume 5% standard deviation).

Summing up, the performance of our model on the 125 compound dataset is similar to the performance of Jensen's models. Applying a model trained on all 125 compounds to Jensen's 20 compound external validation set, we find that our model generalizes very well. However, it should be noted here that Jensen and co-workers chose the validation set as compounds where their five models agreed on, thus it probably contains compounds that are relatively easy to predict.

5 Summary

The availability of computer based models to predict properties of chemical compounds has shown a tremendous impetus on many areas of chemical research. Models to predict physico-chemical properties, such as log P or water solubility, have been developed to high standards already. Still, in drug design, there is a large need also for models that predict ADME properties. Due to the complexity of these endpoints, only few off-the-shelf tools are available.

In this work, we presented a novel machine learning method for ranking compounds with respect to their metabolic stability in different *in vitro* assays. Data stem from drug design projects at Bayer Schering Pharma. Thus, the developed models are tailored to the classes of compounds that Bayer Schering Pharma typically considers. Our evaluations showed that the developed models provide a highly accurate compound ranking, both when checked with cross-validation and on a validation set that was not known at the time of model building. Results werde confirmed using a set of publicly avaible data. Comparing with the work of Jensen *et al.*,[7] we find that the performance of our Gaussian Process models is competitive.

One of the main features of the developed models is an accurate and intuitive notion for the "domain of applicability". The model is fully probabilistic, and outputs the probability for a compound to be metabolically stable. Outside the range spanned by training data, and in regions of conflicting measurements, the probability gets closer to 0.5, indicating that the prediction is most likely not accurate. The model implicitly conflates its prediction and a measure for the domain of applicability into a single quantity that can be directly used for compound ranking. Furthermore, we showed that the model output is calibrated, allowing for an intuitive understanding of the model output.

The final GPmet model has been implemented as a batch predictor and is fully integrated into the working environment at Bayer Schering Pharma. The model can produce around 50 predictions per second on a single 2 GHz Pentium CPU. Along with the GPmet model, a fully automatic re-training tool has been developed, that allows for a model extension whenever new data becomes available, and thus constantly enlarges the models' domain of applicability.

Acknowledgments

The authors gratefully acknowledge partial support from the PASCAL Network of Excellence (EU #506778) and DFG grant MU 987/4-1. We would like to thank

Vincent Schütz and Carsten Jahn for maintaining the PCADMET database, Lars Norgaard for providing the data used in ref,[7] and the anonymous reviewers for their detailed comments on the first version of this paper.

References

1. Hou, T.; Xu, X. ADME Evaluation in Drug Discovery. 3. Modeling Blood-Brain Barrier Partitioning Using Simple Molecular Descriptors. *J. Chem. Inf. Comput. Sci.* **2003,** *43,* 2137–2152.

2. 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. *Journal of Chemical Information and Modelling* **2007,** *47,* 407–424.

3. Schroeter, T.; Schwaighofer, A.; Mika, S.; Laak, A. T.; 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.* **2007,** online.

4. Schroeter, T.; Schwaighofer, A.; Mika, S.; ter Laak, A.; Sülzle, D.; Ganzer, U.; Heinrich, N.; Müller, K.-R. Predicting Lipophilicity of Drug Discovery Molecules using Gaussian Process Models. *ChemMedChem* **2007,** online.

5. Schroeter, T.; Schwaighofer, A.; Mika, S.; Laak, A. T.; Suelzle, D.; Ganzer, U.; Heinrich, N.; Müller, K.-R. Machine Learning Models for Lipophilicity and their Domain of Applicability. *Mol. Pharm.* **2007,** *4,* 524–538.

6. Shen, M.; Xiao, Y.; Golbraikh, A.; Gombar, V.; Tropsha, A. An in Silico Screen for Human S9 Metabolic Turnover Using k-Nearest Neighbor QSPR Method. *J. Med. Chem.* **2003,** *46,* 3013-3020.

7. Jensen, B. F.; Sorensen, M. D.; Anne-Marie, K.; Björkling, F.; Sonne, K.; Engelsen, S. B.; Norgaard, L. Prediction of in vitro metabolic stability of calcitriol analogs by QSAR. *J. Comput.-Aided Mol. Des.* **2003,** *17,* 849–859.

8. Murphy, A. H.; Winkler, R. L. Diagnostic Verification of Probability Forecasts. *International Journal of Forecasting* **1992,** *7,* 435–455.

9. Hastie, T.; Tibshirani, R.; Friedman, J. *The Elements of Statistical Learning;* Springer Verlag, 2001.

10. Schölkopf, B.; Smola, A. J. *Learning with Kernels;* MIT Press, 2002.

11. Bishop, C. M. *Neural Networks for Pattern Recognition;* Oxford University Press, 1995.

12. Orr, G.; Müller, K.-R., Eds.; *Neural Networks: Tricks of the Trade;* volume 1524 Springer LNCS, 1998.

13. Gasteiger, J.; Engel, T., Eds.; *Chemoinformatics: A Textbook;* Wiley-VCH, 2003.

14. Yan, A.; Gasteiger, J.; Krug, M.; Anzali, S. Linear and nonlinear functions on modeling of aqueous solubility of organic compounds by two structure representation methods. *J. Comput.-Aided Mol. Des.* **2004,** *18,* 75-87.

15. Vapnik, V. N. *The Nature of Statistical Learning Theory;* Springer Verlag, 1995.

16. Müller, K.-R.; Mika, S.; Rätsch, G.; Tsuda, K.; Schölkopf, B. An Introduction to Kernel-based Learning Algorithms. *IEEE Neural Networks* **2001,** *12,* 181–201.

17. Merkwirth, C.; Mauser, H.; Schulz-Gasch, T.; Roche, O.; Stahl, M.; Lengauer, T. Ensemble Methods for Classification in Cheminformatics. *J. Chem. Inf. Comput. Sci.* **2004,** *44,* 1971-1978.

18. Byvatov, E.; Fechner, U.; Sadowski, J.; Schneider, G. Comparison of Support Vector Machine and Artificial Neural Network Systems for Drug/Nondrug Classification. *J. Chem. Inf. Comput. Sci.* **2003,** *43,* 1882–1889.

19. 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.

20. Netzeva, T. I. *et al.* Current Status of Methods for Defining the Applicability Domain of (Quantitative) Structure-Activity Relationships. *Alternatives to Laboratory Animals* **2005,** *33,* 1-19.

21. 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,* 700–707.

22. Platt, J. Probabilistic outputs for Support Vector Machines and comparisons to regularized likelihood methods. In *Advances in Large Margin Classifiers*; Smola, A. J.; Bartlett, P.; Schölkopf, B.; Schuurmans, D., Eds.; MIT Press, 1999; pp 61–74.

23. Rasmussen, C. E.; Williams, C. K. *Gaussian Processes for Machine Learning;* MIT Press, 2005.

24. Neal, R. M. Regression and Classification Using Gaussian Process Priors. In *Bayesian Statistics 6*; Bernardo, J. M.; Berger, J.; Dawid, A.; Smith, A., Eds.; Oxford University Press, 1998; Vol. 6, pp 475–501.

25. Masimirembwa, C. M.; Bredberg, U.; Andersson, T. B. Metabolic Stability for Drug Discovery and Development: Pharmacokinetic and Biochemical Challenges. *Clinical Pharmacokinetics* **2003,** *42,* 515-528.

26. Fox, T.; Kriegl, J. M. Machine Learning Techniques for In Silico Modeling of Drug Metabolism. *Curr. Top. Med. Chem.* **2006,** *6,* 1579–1591.

27. Cashman, J. R. Drug Discovery and Drug Metabolism. *Drug Discovery Today* **1996,** *1,* 209–216.

28. Gombar, V. K.; Alberts, J. J.; Cassidy, K. C.; Mattioni, B. E.; Mohutsky, M. A. In Silico Metabolism Studies in Drug Discovery: Prediction of Metabolic Stability. *J. Comput.-Aided Drug Des.* **2006,** *2,* 177–188.

29. Bursi, R.; de Gooyer, M. E.; Grootenhuis, A.; Jacobs, P. L.; van der Louw, J.; Leysen, D. (Q)SAR Study on the Metabolic Stability of Steroidal Androgens. *J. Mol. Graphics Modell.* **2001,** *19,* 552–556.

30. Sadowski, J.; Schwab, C.; Gasteiger, J. *Corina v3.1;* Molecular Networks GmbH Computerchemie: Erlangen, Germany,.

31. Todeschini, R.; Consonni, V.; Mauri, A.; Pavan, M. Dragon For Windows and Linux 2006, `http://www.talete.mi.it/help/Dragon_help/` (accessed 20 Aug 2007).

32. Liu, Y. A Comparative Study on Feature Selection Methods for Drug Discovery. *J. Chem. Inf. Comput. Sci.* **2004,** *44,* 1823–1828.

33. Wegner, J. K.; Fröhlich, H.; Zell, A. Feature Selection for Descriptor Based Classification Models. 2. Human Intestinal Absorption (HIA). *J. Chem. Inf. Comput. Sci.* **2004,** *44,* 931-939.

34. Kuss, M.; Rasmussen, C. E. Assessing Approximate Inference for Binary Gaussian Process Classification. *Journal of Machine Learning Research* **2005,** *6,* 1679–1704.

35. Zhu, C.; Byrd, R. H.; Nocedal, J. L-BFGS-B: Algorithm 778: L-BFGS-B, FORTRAN routines for large scale bound constrained optimization. *ACM Transactions on Mathematical Software* **1997,** *23,* 550–560.

A Inference for Gaussian Process Classification

The predictions in a GP classification model are essentially described by the latent function values f_* on the test point $\mathbf{x}_*$, given all training data X with their labels $\mathbf{y}$. To compute this distribution $p(f_*|X,\mathbf{y},\mathbf{x}_*)$, we first consider the joint distribution of latent function values on test *and* training data. By an integral operation, we can "remove" the unobserved function values on the training data. The joint distribution, in turn, can be factorized into a term relating f_* to $\mathbf{f}$, and a term relating $\mathbf{f}$ to the experimental data:

$$p(f_*|X,\mathbf{y},\mathbf{x}_*) = \int p(f_*,\mathbf{f}|X,\mathbf{y},\mathbf{x}_*)d\mathbf{f} = \int p(f_*,|\mathbf{f},\mathbf{x}_*)p(\mathbf{f}|X,\mathbf{y})d\mathbf{f} \qquad (8)$$

The probability distribution of the latent function values $\mathbf{f}$ on the training data are obtained directly by Bayes' rule as

$$p(\mathbf{f}|X,\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|X)}{\int p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|X)d\mathbf{f}} \qquad (9)$$

For GP classification models, the major problem is computing the term in the denominator of Eq. (9). With the chosen likelihood, $p(y_i|f(\mathbf{x}_i)) = \Phi(y_i f(\mathbf{x}_i))$, the integral can not be solved analytically. Different approximations have been proposed in the literature, in our implementation we used the method of "expectation propagation", EP,[34] to obtain a local Gaussian approximation for each of the likelihood terms,

$$p(y_i|f(\mathbf{x}_i)) \approx Z_i\mathcal{N}(f(\mathbf{x}_i)|m_i,s_i) \qquad (10)$$

The parameters of this approximation are found in an iterative procedure, by matching mean and variance of the exact and the approximate likelihood. The outcome of this approximation procedure is summarized by a vector $\mathbf{m} = (m_1, \ldots, m_N)$ and a diagonal matrix S with parameters s_i along the diagonal.

A.1 Learning the Family of Latent Functions

The most important decision in modelling with Gaussian Process classification is of course the choice of the family of latent functions. Commonly, this is referred to as setting the "hyper parameters",[23] since the latent functions are solely described by the covariance function, which in turn has some parameters. To facilitate choosing these parameters, we consider the marginal likelihood:

$$\mathcal{L} = p(\mathbf{y}\,|\,X,\theta) = \int p(\mathbf{y}\,|\,\mathbf{f})\,p(\mathbf{f}\,|\,X,\theta)\,d\mathbf{f} \tag{11}$$

We use $p(\mathbf{f}\,|\,X,\theta)$ to explicitly denote that the distribution of latent function values depends on a set of parameters θ of the covariance function (in the case of Eq. (7), $\theta = \{\nu, w_1, \ldots, w_d\}$ for a total of d descriptors). A gradient ascent method, such as the Broyden-Fletcher-Goldfarb-Shanno method,[35] can now be used to maximize $\mathcal{L}$ with respect to covariance function parameters θ.

B Choice of Models

A large number of experiments was run to evaluate the performance of different types of models, and with different setups. The most important questions that had to be addressed were:

- Can a regression model provide a better compound ranking than a classification model? After all, the classification model is only trained on the (coarse) stable/unstable information. We investigate two classification models (Support Vector Machines SVM[10] and Gaussian Process classification GPC, see Sec. 2.8) and three regression models (linear ridge regression RR,[9] Gaussian Process regression GPR[23] and Support Vector regression SVR[10]). For the non-Bayesian models, parameters were chosen in nested cross-validation on the respective training sets with depth search in good parameter regions. For the Bayesian models, we performed a maximization of marginal likelihood.

- Which type of feature selection shall be used? We investigated three different choices:
  - No feature selection at all, using all descriptors in Dragon blocks $\{1, 2, 6, 9, 12, 15, 16, 17, 18, 20\}$
  - Filter-based feature selection using a non-parametric correlation test, discarding those features that are uncorrelated with the end-point at high p-values
  - Model-based feature selection via the automatic relevance determination procedure for GPC and GPR models,[23] where each descriptor is assigned a weight w_i in Eq. (7) that is subsequently found by maximizing marginal likelihood
- Furthermore, we investigated the influence of conformation dependent descriptors. In our work, we used Corina[30] to predict one conformation of the molecule, which in turn is used in Dragon[31] to compute conformation dependent 3D descriptors. The 3D descriptors are thus only approximations, and might mislead the model.
- Shall the models be built on all data, or only on those that have a clear experimental outcome? For classification approaches, we assign compounds with an experimental value $\geq 50\%$ to class "stable", the others to class unstable. Due to the high measurement noise, some compounds might thus be mis-labelled and bring wrong information into the classifier. Thus, it might be helpful to exclude data with an experimental value around 50%.

Table 5 summarizes the full results for each pair of feature selection method and modelling approach. Note that, for most methods, there is only a minor (non-significant) difference in performance between no feature selection and filter-based feature selection. Furthermore, the regression methods that are trained on a more fine-grained data can only achieve a small gain in performance over the classification methods. Amongst the classification methods, GP classification usually outperforms the SVM.

The result on feature selection seems to contradict the experience of most people working in QSAR modelling, but also in machine learning, where most text books advocate the use of descriptor (feature) selection mechanisms. However, methods based on kernel functions (such as Support Vector Machines) or, equivalently, covariance functions (such as Gaussian Process models) are known to be

Human	RR	SVR	SVM	GP	GPC
None	82.3 ± 0.2	85.5 ± 0.3	83.5 ± 0.4	86.2 ± 0.1	85.0 ± 0.3
Filter based	82.8 ± 0.2	85.7 ± 0.3	83.8 ± 0.4	86.3 ± 0.1	85.2 ± 0.2
Model based	n/a	n/a	n/a	87.2 ± 0.2	85.7 ± 0.3
Mouse female	RR	SVR	SVM	GP	GPC
None	79.6 ± 0.4	84.0 ± 0.4	82.1 ± 0.3	84.1 ± 0.3	83.2 ± 0.5
Filter based	82.8 ± 0.2	84.0 ± 0.3	82.2 ± 0.5	84.6 ± 0.4	83.5 ± 0.5
Model based	n/a	n/a	n/a	84.6 ± 0.4	83.0 ± 0.6
Mouse male	RR	SVR	SVM	GP	GPC
None	81.0 ± 0.6	84.2 ± 0.4	81.0 ± 0.4	84.2 ± 0.4	82.7 ± 0.4
Filter based	82.2 ± 0.4	83.7 ± 0.5	80.6 ± 0.2	84.6 ± 0.4	83.4 ± 0.3
Model based	n/a	n/a	n/a	83.7 ± 0.4	82.0 ± 0.4
Rat male	RR	SVR	SVM	GP	GPC
None	80.5 ± 0.4	85.4 ± 0.1	83.5 ± 0.2	86.1 ± 0.1	85.0 ± 0.3
Filter based	82.0 ± 0.3	85.1 ± 0.3	83.8 ± 0.1	86.5 ± 0.2	85.5 ± 0.3
Model based	n/a	n/a	n/a	86.1 ± 0.2	84.4 ± 0.6

Table 5: Ranking quality, evaluated in terms of area under the ROC curve for 2-fold cross-validation on the training data, for different types of models and feature selection methods. The table lists the average performance over 5 runs of cross-validation $\pm$ standard deviation

quite robust with respect to the presence of a large number of descriptors, even if these do not carry information. This may be attributed to the fact that extra descriptors do not increase the dimensionality of the model's parameter space: In a linear model, each extra descriptor requires an extra parameter to be estimated, whereas usually only one shared "width" parameter has to be chosen in an SVM kernel function (irrespective of whether there are 10 or 10,000 descriptors). In a Gaussian Process model, a width parameter can be introduced for each descriptor dimension, but here the Bayesian framework provides a quite reliable framework to choose each width parameter without running risk of over-fitting.

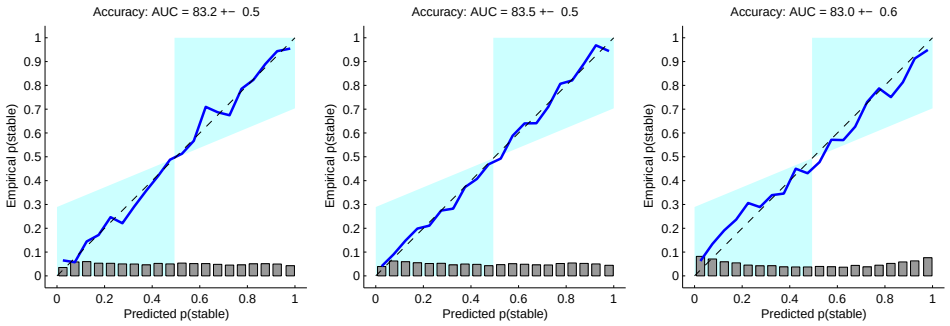
We find our results in line with previous work[32,33] on descriptor selection in QSAR modelling with SVMs. Excellent results can be achieved with an SVM with 2934 descriptors[33] or even more than $100,000$ descriptors.[32] By using different methods of feature selection (genetic algorithms[33]) or filter based methods[32]) it is possible to reduce the number of features drastically. However, at least for SVMs, this reduction usually has a negative impact on classification performance.[32] Reported performance gains[33] have large standard errors, hence the gains are not statistically significant.

So far, the focus was only on the impact of feature selection on the ranking quality. Figure 9 plots the calibration curves for a GPC model with different methods of feature selection. We have chosen the assay "Mouse female" here, since the corresponding results for ranking quality (shown in Table 5, top) are essentially identical. However, the calibration curves show that model-based feature selection leads to slightly over-optimistic predictions. As shown by the histogram bars, more compounds are (wrongly) predicted to be stable or unstable with high certainty. A numerical comparison of the three methods for feature selection can be made by considering the cross-entropy loss function H, where actual class $y_i \in \{+1,-1\}$ and predicted probability p_i are compared via

$$H(y_i,p_i) = -\frac{y_i+1}{2}\log p_i - \frac{1-y_i}{2}\log(1-p_i). \tag{12}$$

Smaller loss indicates a better model fit. Values for H are listed in Figure 9 along with the calibration curves. Just as the calibration curves, the values of H show that feature selection does have an adverse effect on the quality of the probabilistic predictions.

As suggested by one of the reviewers, we also investigated the impact of 3D (and therefore conformation dependent) descriptors. To this end, we built models using both Support Vector Machines and Gaussian Processes for all four species, once with and once without 3D descriptors. Results are presented in Table 6. To



(a) No feature selection: $H = 2.07$

(b) Filter-based feature selection: $H = 2.11$

(c) Model-based feature selection: $H = 2.44$

Figure 9: Calibration curves of a GPC model with different feature selection methods. The small histogram bars show the relative frequency of compounds that attain a classifier output in the respective bin. We also list the average cross-entropy loss H, as defined in Eq. (12). Low values for H indicate good model fit. The plots shown here are for a GPC model for the assay "Mouse female", evaluated in 2-fold cross-validation on the training data. Results for other assays showed similar behavior

allow a meaningful comparison, no feature selection was used. When comparing the performance with and without 3D descriptors, only non-significant differences can be observed (the ± 1 standard deviation intervals overlap in all eight cases). We conclude that the impact of conformation dependent Dragon 3D descriptors on the prediction performance of our models is negligible.

In Table 7, we list the ranking quality of methods that either include or exclude data where the experimental value is around 50%. We focus only on two exemplary methods, SVR (regression, non-Bayesian) and GPC (classification, Bayesian). Clearly, including the data with experimental values around 50% is beneficial, in particular for the regression method SVR.

Assay	SVR: incl. 3D	SVR: excl. 3D	GP: incl. 3D	GP: excl. 3D
Human	85.4 ± 0.3	85.9 ± 0.3	86.2 ± 0.3	86.7 ± 0.2
Mouse female	83.7 ± 0.6	82.9 ± 0.5	84.1 ± 0.8	83.7 ± 0.5
Mouse male	83.7 ± 1.0	84.1 ± 0.5	84.2 ± 0.9	84.4 ± 0.7
Rat male	85.5 ± 0.4	85.1 ± 0.5	86.2 ± 0.3	86.1 ± 0.3

Table 6: Does the performance change when including or excluding 3D descriptors? Experiments were conducted without feature selection, and on all data including those with an experimental value of 50% stability. The table lists the average performance (area under the ROC curve, AUC) over 5 runs of cross-validation $\pm$ standard deviation.

Assay	SVR: all data	SVR: omit [35 65]	GPC: all data	GPC: omit [35 65]
Human	85.7 ± 0.3	84.4 ± 0.2	85.2 ± 0.2	84.9 ± 0.2
Mouse female	84.0 ± 0.3	82.4 ± 0.2	83.5 ± 0.5	83.7 ± 0.4
Mouse male	83.7 ± 0.5	82.5 ± 0.4	83.4 ± 0.3	83.0 ± 0.3
Rat male	85.1 ± 0.3	84.2 ± 0.2	85.5 ± 0.3	84.7 ± 0.3

Table 7: Model selection: Shall the models be built on all data, or only on those that have a clear experimental outcome? The table lists the average performance (area under the ROC curve, AUC) over 5 runs of cross-validation $\pm$ standard deviation