

## A Probabilistic Approach to Classifying Metabolic Stability

Anton Schwaighofer,<sup>\*,†</sup> Timon Schroeter,<sup>†,‡</sup> Sebastian Mika,<sup>§</sup> Katja Hansen,<sup>†,‡</sup> Antonius ter Laak,<sup>||</sup> Philip Lienau,<sup>||</sup> Andreas Reichel,<sup>||</sup> Nikolaus Heinrich,<sup>||</sup> and Klaus-Robert Müller<sup>†,‡</sup>

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, and Research Laboratories of Bayer Schering Pharma, Müllerstraße 178, 13342 Berlin, Germany

Received April 23, 2007

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 modeling method on a set of publicly available data.

### 1. INTRODUCTION

In the drug development process, 50% of the failures<sup>1</sup> in late development stages are due to an unfavorable absorption, distribution, metabolism, excretion, and toxicity (ADMET) profile. 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<sup>2,3</sup> or lipophilicity.<sup>4,5</sup> 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 can one hope to establish quantitative structure–property/activity relationship (QSPR/QSAR) models that reliably predict a property like metabolic stability. To date, there is only little published work about such approaches,<sup>6,7</sup> 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 (GP) priors. Each of the models (for the assays 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 the ability to provide an implicit check for the “domain of applicability” (DOA). 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 focusing on the compounds that can be predicted with high confidence by the model. Also, the developed models are, in a statistical sense, well-calibrated:<sup>8</sup> 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, for example, a predictor for the

\* Corresponding author e-mail: anton@first.fraunhofer.de.

<sup>†</sup> Fraunhofer FIRST.

<sup>‡</sup> Technische Universität Berlin.

<sup>§</sup> idalab GmbH.

<sup>||</sup> Research Laboratories of Bayer Schering Pharma.

property “metabolic stability” or for the water solubility of a compound.<sup>2,3</sup> A large variety of techniques has been developed in the machine learning and statistics communities to account for different prediction tasks and application areas.<sup>9,10</sup>

The use of machine learning techniques for computational chemistry is of course not new. Neural networks,<sup>11,12</sup> for example, have a long history in computational chemistry.<sup>13,14</sup> Recently, the successful application of support vector machines (SVMs)<sup>10,15,16</sup> in many domains has also initiated their use for predicting properties of chemical compounds.<sup>17–19</sup>

In virtually all application scenarios for such QSAR models, it is a key requirement to provide confidence estimates.<sup>20,21</sup> 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 cannot provide theoretically well-founded confidence estimates (only heuristics such as “Platt scaling”<sup>22</sup> 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 a probability distribution as well, 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 model.<sup>23,24</sup>

The authors have demonstrated in recent work how Gaussian process regression models can be used to accurately predict the water solubility<sup>2,3</sup> and the lipophilicity<sup>4,5</sup> 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 retraining of the model whenever new data become available. In this paper, we show how a related 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.** The measurement of metabolic stability was part of the first in vitro studies on drug discovery molecules in the pharmaceutical industry, aiming at predicting the in vivo pharmacokinetics.<sup>25</sup> 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 is critical for its market potential.<sup>25</sup>

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.<sup>26</sup> Investigations from this work were performed using liver microsomal preparations and, hence, cover only oxoreductive phase I metabolism, involving enzymes like cytochrome P450, flavine mono-oxidases, esterases, and epoxide hydrolases. An overview of the involved enzymes and processes depending on the biological matrix can, for example, be found in the work of Cashman.<sup>27</sup>

Details on the correlations between in vitro and in vivo clearance are given by Masimirembwa et al.<sup>25</sup>

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.<sup>26</sup>

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

- Rule-based systems apply a large number of programmed biotransformations 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 coadministered 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.<sup>29</sup> 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 unstable). A set of 130 calcitriol analogs was used by Jensen et al.<sup>7</sup> 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.<sup>6</sup> used a modified nearest neighbor approach, including a distance-based heuristic to estimate the domain of applicability<sup>20</sup> on a set of 631 compounds. The most recent reference describes the process of building metabolic stability models at Eli Lilly.<sup>28</sup> 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 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.<sup>3</sup> 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 become available, an improved model can be constructed easily.

**Table 1.** Number of Available Experimental Data for Each Assay<sup>a</sup>

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)

<sup>a</sup> 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.

## 2. METHODS AND DATA

**2.1. Methodology Overview.** For each molecule, the 3D structure of one conformation is predicted using the program Corina.<sup>30</sup> From the 2D structure and the predicted 3D structure, a set of Dragon<sup>31</sup> descriptors is generated. On the basis of the descriptors and measurements of the percentage of each compound remaining after incubation with liver microsomes for 30 min, a Gaussian process classification (GPC) 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; that is, 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  $\mu$ M. A sodium phosphate buffer was used at 100 mM and pH 7.4. The cofactors were glucose-6-phosphate (8 mM), MgCl (4 mM), NADP (0.5 nM), and G-6-P dehydrogenase (1 IU/mL).

Compounds were tested at 3  $\mu$ M. Two samples were incubated at 37 °C and subjected to constant shaking for 30 min and were stopped by the addition of ice-cold methanol (1 + 1). Zero minute samples were stopped by ice-cold methanol before adding the test compound. All samples were stored in the freezer (−20 °C) overnight and thawed during centrifugation at 2000g before taking an aliquot for HPLC-UV/vis analysis.

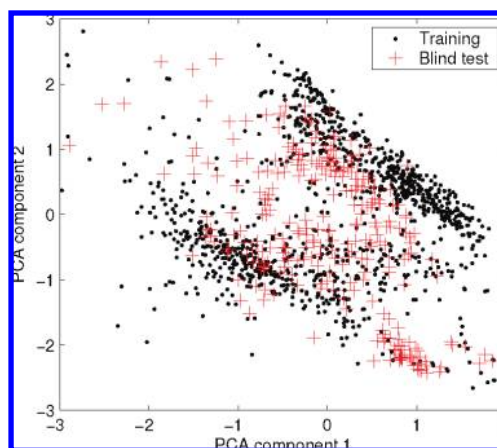
Experimental outcome is the percent recovery at 30 min, given as the peak area of the parent compound in relation to the 0 min value. Testosterone was used as a metabolic reference compound at 100  $\mu$ M. All incubations were performed in duplicate.

**2.3. Experimental Data.** The Bayer Schering Pharma in-house data used for model building include 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 discovery projects was used as blind test data for the final models. These data are summarized in Table 2. Figure 1 illustrates the differences between training and blind test data via a principal component analysis (PCA) plot. PCA was computed on the descriptors for the training data, afterwards the blind test data were projected into the PCA coordinate system. The plot shows that training and blind test data contain compounds that are structurally different. Figure 2 highlights the differences in experimental values between the two data sets, by looking at the histograms (pooled over all assays).

**Table 2.** Number of Blind Test Data for Each Assay<sup>a</sup>

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)

<sup>a</sup> 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.



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

**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.

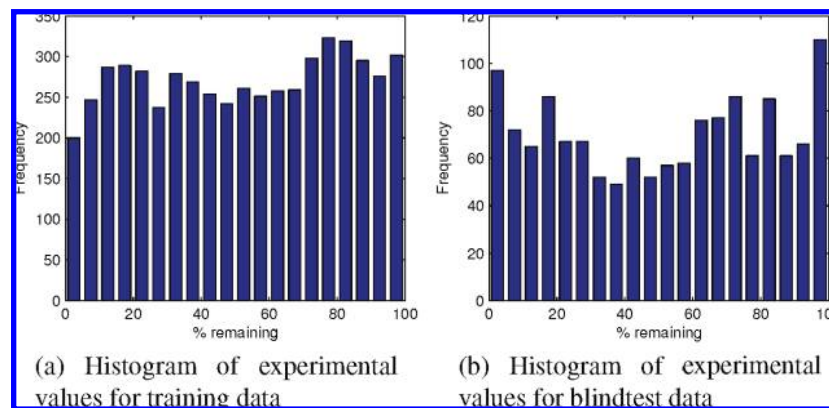
The set of measurement values is noisy and contains large outliers. Compounds where the spread of experimental values is larger than 30% were removed completely. To merge multiple measurements into a consensus value, we proceed as follows: We consider the histogram of measured values. Such a histogram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

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, a large *y* spread with a 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 section 2.2, we used the following protocol:

**Training:** In order to choose the right descriptors and model structure, and also to estimate model performance,





**Figure 2.** Histograms of raw experimental values for training and blind test data, with all assays pooled.

we used 2-fold cross-validation on the training data for each assay. The training data are split into two halves. A GP classification model is built on the first half and evaluated on compounds in the second half. Note that the GP model can choose its parameters automatically via a Bayesian model selection criterion, and thus does not require an extra loop of cross-validation to choose model parameters. This is repeated with the roles of the two halves exchanged. The overall procedure is then repeated five times with different random splits. Thus, in each of the five 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. On the basis of the cross-validation performance, suitable pre-processing steps were chosen (see Appendix B) 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 modeling 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.<sup>31</sup>

After a first modeling stage using all descriptors, it turned out that a large number of descriptors can be omitted without significantly impacting the model's 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, and 20. (Most of these Dragon descriptors only depend on the 2D structure of the molecule, while some actually take 3D information into account.) In Appendix B, we describe the influence of different strategies for selecting a set of relevant descriptors (feature selection) on the ranking quality and on the quality of confidence estimates.

**2.7. Choice of Models.** On the basis of the available experimental data (percent recovery after 30 min), it is possible to build either regression models that predict the percent 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 Appendix B.

With the actual application scenario at Bayer Schering Pharma in mind, we decided to choose a 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 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 (percent 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 section 1.2).

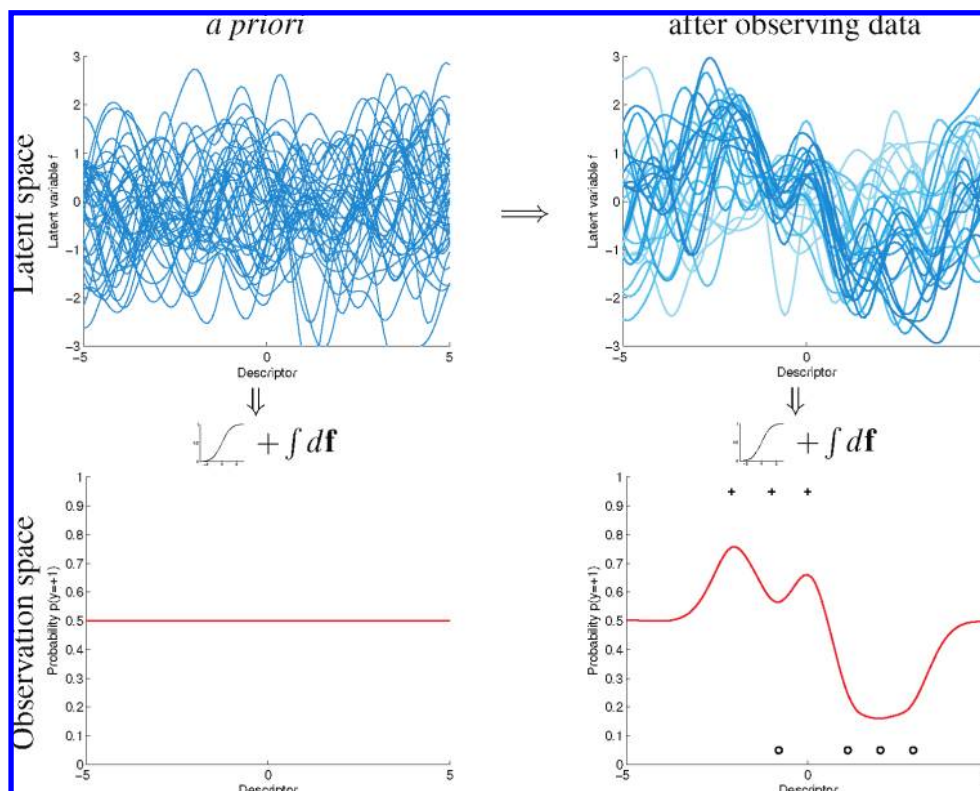
**2.8. Gaussian Process Models.** We start here with a short overview of the key ideas of Gaussian process classification. For an in-depth treatment, we refer to a recent book.<sup>23</sup>

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 can we 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  $\mathbf{X}_1, \dots, \mathbf{X}_N$ . Each



**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 it can explain the data (in observation space). Well-matching functions are shown in dark shading and poorly matching functions in light shading. After transforming and integrating, we obtain the learned class membership information.

compound is assigned to either of two classes, which we label +1 and -1. The class assignment is denoted by  $y_1, \dots, y_N$  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,  $\Phi$ , that gives an output in the range of [0,1]. As a transfer function, we choose the Gaussian cumulative distribution function,  $\Phi(z) = \int_{-\infty}^z N(x; 0,1) dx$ . 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

$$p(y = 1|\mathbf{x}) = \Phi(f(\mathbf{x})) \quad (1)$$

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 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,  $\mathbf{X} = \{\mathbf{X}_1, \dots, \mathbf{X}_N\}$ , we have one latent function value. Denote the latent function values at the  $N$  training data points as  $\mathbf{f} = (f(\mathbf{X}_1), \dots, f(\mathbf{X}_N))$ .

These follow a joint multivariate Gaussian distribution:

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

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

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 Appendix 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  $p(f_*|\mathbf{X}, \mathbf{y}, \mathbf{x}_*)$ . To obtain the class membership probability, we average (integrate) over the transformed function, weighted by the degree of belief:

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

With our particular choice of transfer function  $\Phi$  and a Gaussian distribution for  $p(f_*|\mathbf{X}, \mathbf{y}, \mathbf{x}_*)$ , 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 look 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  $f_*$ ,

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

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

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

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

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

**2.8.3. Adapting Parameters.** In our model to predict metabolic stability, we use a covariance function of the form

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

(the “rational quadratic” covariance function<sup>23</sup>).  $k(\mathbf{x}, \mathbf{x}')$  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  $w_i \geq 0$  that effectively describes the importance of the  $i$ th descriptor for the modeling task.

In order to set the weights  $w_i$  and the parameter  $\nu$ , 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 Appendix A.

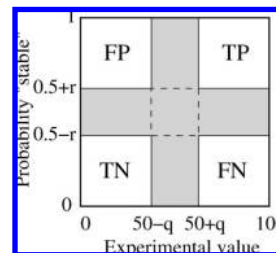
### 3. RESULTS

We evaluate the resulting models 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 section 3.2.

**Probabilistic Output:** We aim at intuitively understandable classifier outputs and, thus, want to achieve a well-calibrated<sup>8</sup> 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 section 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



**Figure 4.** Evaluating 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.

as a mistake? After all, we must expect that when repeating the laboratory 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 focusing on the moderately confident predictions ( $q = 15, r = 0.15$ ), and when focusing on confident predictions ( $q = 30, r = 0.3$ ).

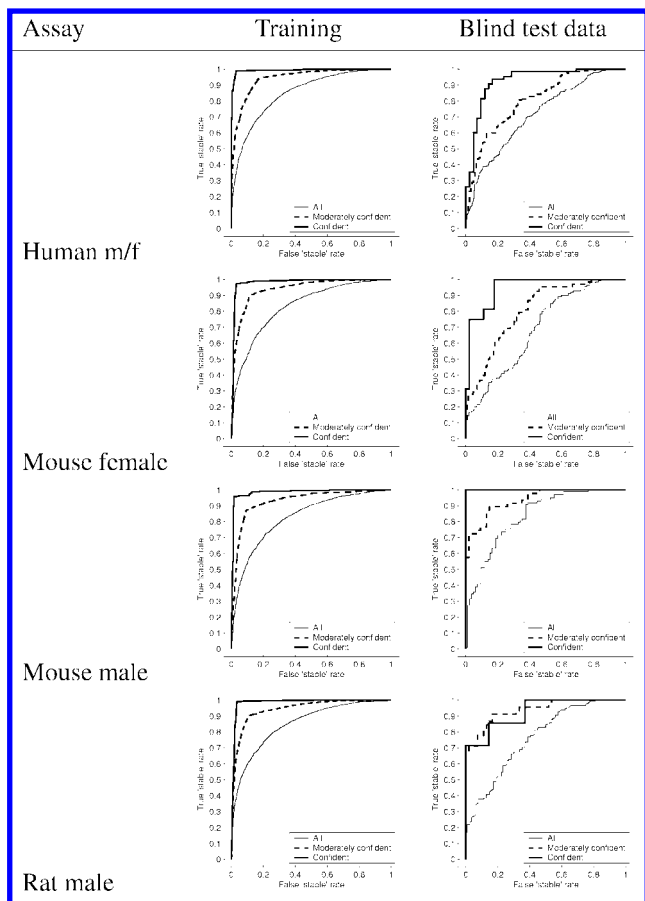
**3.1. Model Selection.** We investigated a large number of modeling approaches, based on regression and classification, built with different parameter settings. The criteria to select between the models were ranking quality (measured by area under the ROC curve) and, when applicable, shape of the calibration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

Model selection details are given in Appendix B. It turned out that optimal results could be achieved by using all available descriptors without feature selection, and including compounds that have an experimental value around 50% (the chosen threshold for stable compounds).

The result on feature selection seems to contradict conventional wisdom, yet is in agreement with other work<sup>32,33</sup> on feature selection for QSAR modeling. As described in Appendix B, we found that, for most methods, the difference in ranking quality between filter-based feature selection and using all features is nonsignificant. 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 overconfident. 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.





**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 Results 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 Tables 3 and 4.

**Table 3.** Evaluating the Ranking Performance: AUC Shown in Figure 5 for 2-Fold Cross-Validation on the Training Data<sup>a</sup>

assay	all data		moderately confident		confident	
	AUC	% of data	AUC	% of data	AUC	% of data
human	85.0	58.2%	94.4	29.1%	99.4	19.0%
mouse female	83.2	51.7%	93.9	18.0%	98.5	14.3%
mouse male	82.7	50.6%	93.0	24.1%	98.5	14.3%
rat male	85.0	54.3%	94.4	24.1%	98.5	14.3%

<sup>a</sup> 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  $q = 15$ ,  $q = 0.15$  (see Results), and “confident” evaluates the subset of data with confident outcomes ( $q = 30$ ,  $q = 0.30$ ).

**3.2. Ranking Performance.** Figure 5 shows ROC curves for the final models for each of the four assays, both when evaluated on the training data (2-fold cross-validation) and when evaluated 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 Tables 3 and 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

**Table 4.** Evaluating the Ranking Performance: AUC Shown in Figure 5 for the Blind Test Data<sup>a</sup>

assay	all data		moderately confident		confident	
	AUC	% of data	AUC	% of data	AUC	% of data
human	71.8	50.7%	80.7	17.0%	92.8	10.0%
mouse female	69.0	59.5%	80.6	18.7%	95.0	10.0%
mouse male	83.5	55.2%	93.7	21.3%	100.0	10.0%
rat male	76.4	37.5%	93.8	15.9%	92.7	10.0%

<sup>a</sup> The subsets of data with moderately confident and confident outcomes are defined as in Table 3.

on the subset of data with confident experimental value and prediction ( $q = 30$ ,  $r = 0.30$ ).

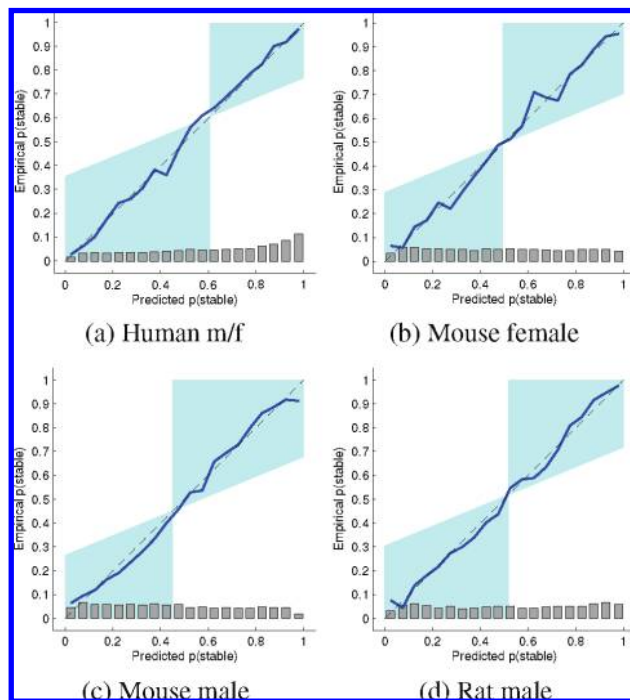
When comparing the results on the training data and that on the blind test data (536 drug candidates from recent projects at Bayer Schering Pharma), we can observe a small drop in performance. Still, the performance remains promisingly high. Note that the blind test data stems from new projects and, 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 a smaller number of compounds that can be predicted with high confidence.

**3.3. Probabilistic Outputs.** Our choice of Gaussian process models was (among other criteria, see section 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.<sup>8</sup> 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 a high probability of being stable, there should be a larger fraction of compounds that actually are stable, versus among the compounds with a low probability of being stable.

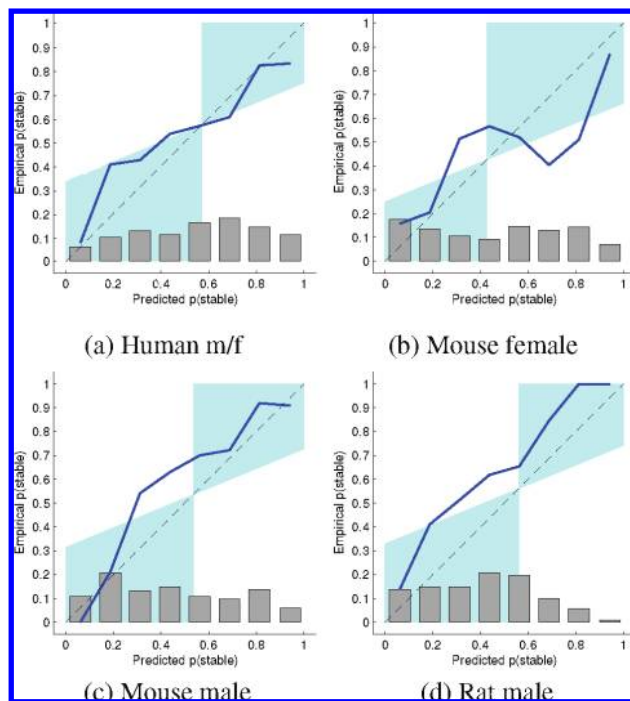
We evaluate the calibration property by means of a calibration curve. Here, we consider groups of compounds for which the predicted probability  $p_{\text{pred}}$  is in bins centered around  $[0.1, 0.2, \dots, 0.9]$ . For each group, we compute the within-bin fraction of stable compounds,  $p_{\text{emp}}$ . 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 plotted the predicted probability  $p_{\text{pred}}$  on the  $x$  axis versus the empirical probability  $p_{\text{emp}}$  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



**Figure 6.** Calibration curves (see section 3.3) of metabolic stability predictions on the training set, evaluated in 2-fold 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.



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

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 overconfident for some compounds that are correctly predicted to be unstable. The curve for “Mouse female” shows nonoptimal 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 pseudoprobabilities can also be computed from methods such as SVMs, by fitting a sigmoid function to the SVM output.<sup>22</sup> However, 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<sup>6,7</sup> deal with directly modeling metabolic stability. Shen et al.<sup>6</sup> used in-house data from GSK, whereas Jensen et al.<sup>7</sup> used a set of data that is publicly available. In the following section, we test our modeling method on the data provided by Jensen et al. They structured their work as follows:

1. A set of 130 compounds was split into a training set (87 compounds) 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 five different ways, each time followed by building a model using 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.

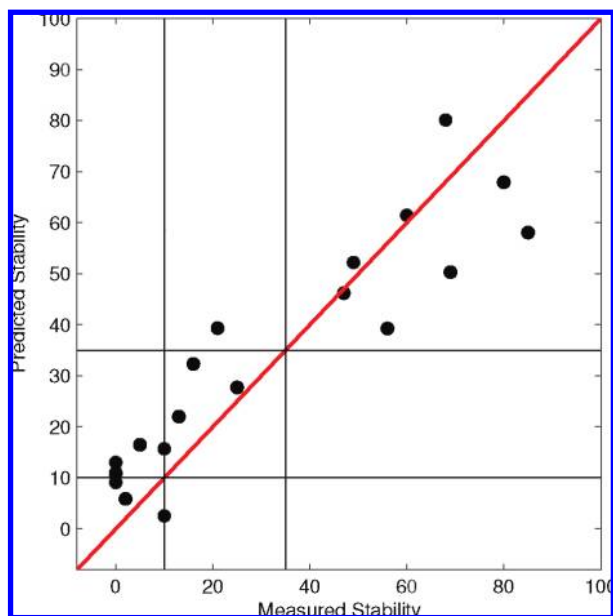
3. From this set of 240 new compounds, 20 compounds were selected where the agreement of the five models built in step 2 was largest. The metabolic stability of these compounds was measured and compared to the predictions of the “consensus model” built from the five regression models.

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 three-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 sets. Their models yield a RMSE between 21 and 16 on the validation set. On the same single split, a GP model yields a 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 than 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





**Figure 8.** Scatterplot for predicted stability versus measured stability on Jensen et al.'s<sup>7</sup> 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.

stability versus measured stability can be found in Figure 8. Overall, we achieve a RMSE of 12.4.

Jensen and co-workers report that their model makes three mispredictions out of the 20-compound test set (with a slightly optimistic interpretation of measurement 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 three-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 three regions, using the same thresholds that Jensen et al. had used. When now counting errors the same way Jensen et al. did, we see that the Gaussian process model mispredicts only 1 out of the 20 validation set compounds (compound M, experimental S9 of 5, but predicted as 17). The mispredictions 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 data set is similar to the performance of Jensen's models. When training a model on all 125 compounds and then applying it to Jensen's 20-compound external validation set, we find that our model generalizes very well. However, it should be noted here that Jensen et al. 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 that predict physicochemical 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 end points, only few off-the-shelf tools are available.

In this work, we presented Gaussian process classification models as a tool 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 were confirmed using a set of publicly available data. Comparing with the work of Jensen et al.,<sup>7</sup> 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 GP 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 retraining tool has been developed, which allows for a model extension whenever new data become available and, thus, constantly enlarges the models' domain of applicability.

## ACKNOWLEDGMENT

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.<sup>7</sup> and the anonymous reviewers for their detailed comments on the first version of this paper.

## APPENDIX

### 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  $\mathbf{X}$  with their labels  $\mathbf{y}$ . To compute this distribution  $p(f_*|\mathbf{X},\mathbf{y},\mathbf{x}_*)$ , 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  $\mathbf{f}$  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:

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

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

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

For GP classification models, the major problem is computing the term in the denominator of eq 9. With the chosen likelihood,  $p(y_i|f(x_i)) = \Phi(y_i \cdot f(x_i))$ , 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"<sup>34</sup> to obtain a local Gaussian approximation for each of the likelihood terms:

$$p(y_i|f(x_i)) \approx Z_i N(f(x_i)|m_i, s_i) \quad (10)$$

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

**A.1. Learning the Family of Latent Functions.** The most important decision in modeling 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",<sup>23</sup> 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:

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

We use  $p(\mathbf{f}|\mathbf{X}, \theta)$  to explicitly denote that the distribution of latent function values depends on a set of parameters  $\theta$  of the covariance function (in the case of eq 7,  $\theta = \{v, w_1, \dots, w_d\}$  for a total of  $d$  descriptors). A gradient ascent method, such as the Broyden–Fletcher–Goldfarb–Shanno method,<sup>35</sup> can now be used to maximize  $L$  with respect to covariance function parameters  $\theta$ .

## B. CHOICE OF MODELS

A large number of experiments were 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 (SVMs<sup>10</sup> and GPC; see section 2.8) and three regression models (linear ridge regression, RR;<sup>9</sup> Gaussian process regression, GPR;<sup>23</sup> and Support Vector regression, SVR<sup>10</sup>). For the non-Bayesian models, parameters were chosen in nested cross-validation on the respective training sets with a 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 nonparametric 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,<sup>23</sup> 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<sup>30</sup> to predict one conformation of the molecule, which in turn is used in Dragon<sup>31</sup> 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 mislabeled 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 modeling approach. Note that, for most methods, there is only a minor (nonsignificant) 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. Among the classification methods, GP classification usually outperforms the SVM.

The result on feature selection seems to contradict the experience of many people working in QSAR modeling, 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 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 a 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 a risk of overfitting.

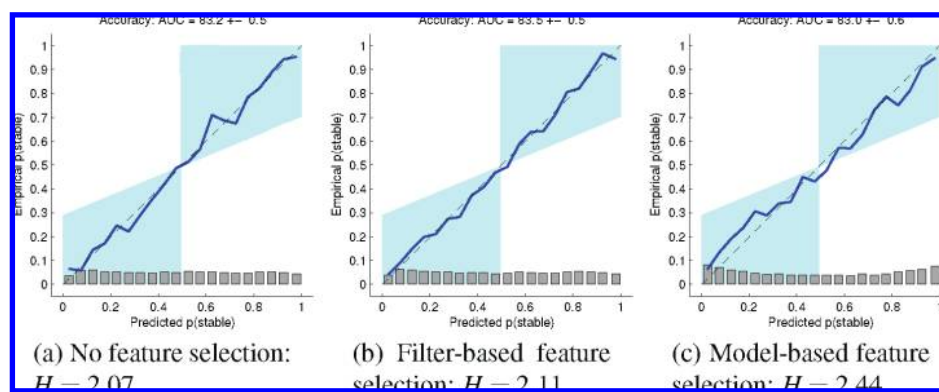
We find our results in line with previous work<sup>33,32</sup> on descriptor selection in QSAR modeling with SVMs. Excellent results can be achieved with a SVM with 2934 descriptors<sup>33</sup> or even more than 100 000 descriptors.<sup>32</sup> By using different methods of feature selection (genetic algorithms<sup>33</sup> or filter-based methods<sup>32</sup>), 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.<sup>32</sup> Reported performance gains<sup>33</sup> 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 corre-

**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<sup>a</sup>

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

<sup>a</sup> The table lists the average performance over five runs of cross-validation ± standard deviation.

**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.**Table 6.** Performance Change when Including or Excluding 3D Descriptors<sup>a</sup>

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

<sup>a</sup> Experiments were conducted without feature selection and on all data, including those with an experimental value around 50% stability. The table lists the average performance (area under the ROC curve, AUC) over five runs of cross-validation ± standard deviation.

sponding 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 overoptimistic 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

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

Smaller loss indicates a better model fit. Values for  $H$  are listed in Figure 9 along with the calibration curves. Just as in 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 allow a meaningful comparison, no feature selection was used. When comparing the performance with and without 3D descriptors, only nonsignificant 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,



**Table 7.** Model Selection: Deciding Whether the Models Shall Be Built on All Data or Only on Those That Have a Clear Experimental Outcome<sup>a</sup>

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

<sup>a</sup> The table lists the average performance (area under the ROC curve, AUC) over five runs of cross-validation ± standard deviation.

including the data with experimental values around 50% is beneficial, in particular for the regression method SVR.

## REFERENCES AND NOTES

- 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.
- 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*, 407–424.
- 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.
- 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.
- 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.
- 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.
- 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.
- Murphy, A. H.; Winkler, R. L. Diagnostic Verification of Probability Forecasts. *Int. J. Forecasting* **1992**, *7*, 435–455.
- Hastie, T.; Tibshirani, R.; Friedman, J. *The Elements of Statistical Learning*; Springer Verlag: New York, 2001.
- Schölkopf, B.; Smola, A. *Learning with Kernels*; MIT Press: Cambridge, MA, 2002.
- Bishop, C. M. *Neural Networks for Pattern Recognition*; Oxford University Press: New York, 1995.
- Orr, G.; Müller, K.-R. *Neural Networks: Tricks of the Trade*; Springer LNCS: New York, 1998; Vol. 1524.
- Gasteiger, J.; Engel, T. *Cheminformatics: A Textbook*; Wiley-VCH: New York, 2003.
- 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.
- Vapnik, V. N. *The Nature of Statistical Learning Theory*; Springer Verlag: New York, 1995.
- 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.
- 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.
- 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.
- 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.
- Netzeva, T. I. Current Status of Methods for Defining the Applicability Domain of (Quantitative) Structure-Activity Relationships. *ATLA, Altern. Lab. Anim.* **2005**, *33*, 1–19.
- 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.
- 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: Cambridge, MA, 1999; pp 61–74.
- Rasmussen, C. E.; Williams, C. K. *Gaussian Processes for Machine Learning*; MIT Press: Cambridge, MA, 2005.
- 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: New York, 1998; Vol. 6, pp 475–501.
- Masimirembwa, C. M.; Bredberg, U.; Andersson, T. B. Metabolic Stability for Drug Discovery and Development: Pharmacokinetic and Biochemical Challenges. *Clin. Pharmacokinet.* **2003**, *42*, 515–528.
- Fox, T.; Kriegl, J. M. Machine Learning Techniques for In Silico Modeling of Drug Metabolism. *Curr. Top. Med. Chem.* **2006**, *6*, 1579–1591.
- Cashman, J. R. Drug Discovery and Drug Metabolism. *Drug Discovery Today* **1996**, *1*, 209–216.
- 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.
- Bursi, R.; de Gooyer, M. E.; Grootenhuys, A.; Jacobs, P. L.; van der Louw, J.; Leysen, D. (Q)SAR Study on the Metabolic Stability of Steroidal Androgens. *J. Mol. Graphics Model.* **2001**, *19*, 552–556.
- Sadowski, J.; Schwab, C.; Gasteiger, J. *Corina*, v.3.1; Molecular Networks GmbH Computerchemie: Erlangen, Germany.
- Todeschini, R.; Consonni, V.; Mauri, A.; Pavan, M. Dragon For Windows and Linux 2006. [http://www.taletmi.it/help/Dragon\\_help/](http://www.taletmi.it/help/Dragon_help/) (accessed 20 Aug 2007).
- Liu, Y. A Comparative Study on Feature Selection Methods for Drug Discovery. *J. Chem. Inf. Comput. Sci.* **2004**, *44*, 1823–1828.
- 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.
- Kuss, M.; Rasmussen, C. E. Assessing Approximate Inference for Binary Gaussian Process Classification. *J. Machine Learning Res.* **2005**, *6*, 1679–1704.
- Zhu, C.; Byrd, R. H.; Nocedal, J. L-BFGS-B: Algorithm 778: L-BFGS-B, FORTRAN routines for large scale bound constrained optimization. *ACM Trans. Math. Software* **1997**, *23*, 550–560.

CI700142C