Gaussian Processes: A Method for Automatic QSAR Modeling of ADME Properties

Olga Obrezanova,*,† Gábor Csányi,‡ Joelle M. R. Gola,† and Matthew D. Segall†

BioFocus DPI, 127 Cambridge Science Park, Milton Road, Cambridge, CB4 0GD, United Kingdom, and Cavendish Laboratory, University of Cambridge, J J Thomson Avenue, Cambridge, CB3 0HE, United Kingdom

Received February 16, 2007

In this article, we discuss the application of the Gaussian Process method for the prediction of absorption, distribution, metabolism, and excretion (ADME) properties. On the basis of a Bayesian probabilistic approach, the method is widely used in the field of machine learning but has rarely been applied in quantitative structure−activity relationship and ADME modeling. The method is suitable for modeling nonlinear relationships, does not require subjective determination of the model parameters, works for a large number of descriptors, and is inherently resistant to overtraining. The performance of Gaussian Processes compares well with and often exceeds that of artificial neural networks. Due to these features, the Gaussian Processes technique is eminently suitable for automatic model generation—one of the demands of modern drug discovery. Here, we describe the basic concept of the method in the context of regression problems and illustrate its application to the modeling of several ADME properties: blood−brain barrier, hERG inhibition, and aqueous solubility at pH 7.4. We also compare Gaussian Processes with other modeling techniques.

1. INTRODUCTION

The importance of optimizing absorption, distribution, metabolism, and excretion (ADME) properties of potential drug molecules is now widely recognized.[1] Considering the ADME properties early in the drug discovery process can reduce the costs of drug development and decrease the attrition rate of drug candidates. In silico predictive modeling offers a possibility to study ADME properties of a molecule before it is even synthesized.

In this article, we present a powerful computational method for the predictive quantitative structure−activity relationship (QSAR) modeling of ADME properties. The method is called Gaussian Processes[2−4] and is based on a Bayesian probabilistic approach. Originating in the machine learning field, it has not yet been widely used in QSAR and ADME modeling. Burden[5] has applied the Gaussian Processes method to create predictive models for three known QSAR data sets: a benzodiazepine data set, a muscarinic data set, and a data set of the toxicity of substituted benzenes to the ciliate *Tetrahymena pyriformis*. Enot et al.[6] and Tiño et al.[7] used this technique to model logP on relatively small data sets, and in a recent paper, Schwaighofer et al.[8] applied Gaussian Processes to model aqueous solubility. Our implementation shares the basic methodology with the above works and introduces a key step which allows the complete automation of model generation with no free parameters requiring human intervention.

This method overcomes many of the problems of existing QSAR modeling techniques:

- Most importantly, it does not require the subjective a priori determination of parameters such as variable importance or network architectures.
- It is suitable for modeling nonlinear relationships.
- The method has a built-in tool to prevent overtraining and does not require cross-validation.
- This technique has an inherent ability to select important descriptors.

The Gaussian Processes technique has been proven to compare well with and often exceed artificial neural networks (ANNs) in performance[5] and has been shown to be equivalent to an ANN with a single hidden layer containing an infinite number of nodes.[9]

The demand of modern drug discovery for fast model (re)-building whenever new data become available gave rise to a trend to develop computational algorithms for automatic model generation.[10,11] The purpose of such algorithms is to save scientists' time, explore more modeling possibilities, and make the process of QSAR model building accessible to nonexperts. The Gaussian Processes technique is sufficiently robust to enable automatic model generation, and because it does not need any subjective input from the user, this method is perfect for an automated process.

A potential disadvantage of the Gaussian Processes technique is that it generates "black box" models, which are difficult to interpret. Although the method has an ability to select the most influential descriptors, it is difficult to extract the contribution of each descriptor to the observed activity or property. This difficulty is inherent to the nature of modeling nonlinear relationships, since the contribution of a descriptor to the model cannot be characterized by a single scalar, unlike in linear modeling. Another drawback of the Gaussian Processes approach is that it can be computationally expensive. The training of a model involves multiple inversions of an $N \times N$ matrix, where N is a number of compounds in the training set; time for direct inversion is of the order of N^3. Methods for approximate matrix inversion were described by Gibbs and MacKay.[12]

* Corresponding author phone: +44(0)1223706177; e-mail: olga.obrezanova@glpg.com.
† BioFocus DPI.
‡ Cavendish Laboratory.

10.1021/ci7000633 CCC: $37.00
Published on Web 06/28/2007

In this article, we present the concepts of the Gaussian Processes approach for regression problems and introduce new techniques of determining model parameters (called *hyperparameters* in the Gaussian Processes framework, see below). In section 2.2.1, we give a set of empirical formulas for the hyperparameters which worked well for a variety of data sets and reduces the number of matrix inversions to just one, an approach that can be used for large data sets. Gradient descent optimization techniques have been used in previous studies,[5,8] but these can fail if the optimized function is multimodal. Nested sampling,[13] a novel technique, which we use in this work, does not suffer from this problem.

To illustrate the application of the described methods to build QSAR models, we will use two in-house ADME data sets—hERG inhibition and aqueous solubility at pH 7.4 data sets. We will also compare the Gaussian Processes technique with other modeling methods by building models for a benzodiazepine data set, modeled by Burden,[5] and a blood–brain barrier data set used by Winkler and Burden.[14]

2. METHODS

2.1. Gaussian Processes for Regression. Our brief description of Gaussian Processes for regression closely follows the work of Mackay.[2] The method is based on casting the problem of building a model for some data in the form of a Bayesian inference. The general logic of Bayesian inference is that one assumes a *prior probability distribution* for the values of some unknown object and updates this probability distribution in the light of observed data to yield the *posterior distribution*. In the case of regression, the unknown object is the model underlying the data (i.e., a function of the input space which ought to describe the data set), and the probability distribution is over all continuous functions of the input space.

Let us denote the training data by $\mathcal{D} = \{\mathbf{Y}, \mathbf{X}\}$, where vector $\mathbf{Y} = \{Y^{(n)}\}_{n=1}^{N}$ is the set of observed property values and matrix $\mathbf{X} = \{\mathbf{x^{(n)}}\}_{n=1}^{N}$ is the set of the molecular descriptors. Here, N is a number of compounds in the training set. We wish to build a model $y(\mathbf{x})$ that predicts the property of a molecule given its descriptor vector $\mathbf{x}$. Then, the Bayesian update rule is

$$P(y(\mathbf{x})|\mathcal{D}) \propto P(\mathbf{Y}|y(\mathbf{x}),\mathrm{X})\, P(y(\mathbf{x}))$$

where $P(y(\mathbf{x})|\mathcal{D})$ is the posterior, $P(\mathbf{Y}|y(\mathbf{x}),\mathbf{X})$ is the likelihood, and $P(y(\mathbf{x}))$ is the prior.

The power of the Gaussian Processes approach lies in that it is parametrized very differently from ordinary nonlinear regression. Rather than choosing fixed functional forms with free parameters, and thus imposing undue limitation and inherent biasing on the model, the parameters of the Gaussian Process control the prior probability distribution of the model function over all possible functions. To underscore this difference, such parameters are traditionally called *hyperparameters*. The natural question immediately arises: "If every continuous function is allowed, how can a few data points guide us in the right direction and, especially, how do we prevent overtraining?". The answer depends on what kind of prior distribution we choose. In practice, the prior distributions will ensure that the prediction function (which we will take as the mean of the posterior distribution) is smooth and matches the observed data as well as possible. Tradeoffs between smoothness and fitting the data can be controlled via the hyperparameters.

The Gaussian Process is defined by stating that, given any set of descriptor vectors $\mathbf{x^{(1)}}$, ..., $\mathbf{x^{(n)}}$, the distribution for the function values is a multidimensional normal distribution with a mean and a covariance matrix which depend on the descriptor vectors. Taking the mean of this distribution to be zero, we have

$$P(y(\mathbf{x^{(1)}}), ..., y(\mathbf{x^{(n)}})) = \mathcal{N}(0,\mathbf{Q}) \quad (1)$$

where $\mathbf{Q}$ is the covariance matrix for the given set of descriptor vectors. The matrix elements are given by the covariance function $C(\mathbf{x^{(n)}},\mathbf{x^{(n')}})$, often referred to as the *kernel*. The role of this function is to define the metric in the input space, that is, the similarity between different molecules. The above expression is equivalent to saying that, for molecules which are not very similar (i.e., the corresponding matrix elements of $\mathbf{C}$ and $\mathbf{C^{-1}}$ are small), there is not much correlation between the function values. Conversely, if two descriptor vectors are similar according to the covariance function, the function values are similar as well.

If each property value is assumed to differ from the corresponding function value by Gaussian noise with zero mean and variance σ_ν^2, then the distribution for the property values is a multivariate normal distribution:

$$P(\mathbf{Y}) = \mathcal{N}(0,\mathbf{C}) \quad (2)$$

with the covariance matrix

$$\mathbf{C} = \mathbf{Q} + \sigma_\nu^2\mathbf{I} \quad (3)$$

We now define our covariance function. It is in this definition that we will be able to tune the above-mentioned tradeoff between the smoothness of y and the fit quality, as well as make prior statements about the relative relevance of different descriptors. Our covariance function will itself be Gaussian, a choice which is most common. We choose the covariance as follows:

$$C(\mathbf{x^{(n)}}, \mathbf{x^{(n')}}) = \theta_1 \exp\left[-\frac{1}{2}\sum_{i=1}^{K} (x_i^{(n)} - x_i^{(n')})^2/r_i^2 \right] + \theta_2 \quad (4)$$

where K is the number of descriptors; $x_i^{(n)}$ is the value of the ith descriptor for the nth molecule; and θ_1, θ_2, and $\{r_i\}_{i=1}^{K}$ are hyperparameters, with the following meanings: The overall scale for the property values is given by θ_1, and the $\{r_i\}$ are a set of length scale parameters, one for each descriptor. A very large value for a given r_i is equivalent to saying that differences in the corresponding descriptor do not influence the property values very much. An overall constant shift in the function away from zero is given by θ_2. We will denote the variance of the assumed noise in the data by $\theta_3 = \sigma_\nu^2$. The last hyperparameter, θ_3 controls the aforementioned tradeoff between smoothness and quality of fit. If θ_3 is taken to be small, that is, the assumed noise is small, the inference procedure will favor more closely fitting models at the expense of smoothness. Too small a value can thus result in overtraining, which is equivalent to not considering enough noise in the data.

The central result we use, which follows from eq 2,[2] is that, given a training set, the posterior distribution for the property value $y' = y(\mathbf{x}')$ for a new molecule with descriptor vector $\mathbf{x}'$ is also Gaussian:

$$P(y(\mathbf{x}')|\mathcal{D}) = \mathcal{N}(\langle y' \rangle, \sigma_{y'}^2) \tag{5}$$

with the following mean and variance

$$\langle y' \rangle = \mathbf{k}^T \mathbf{C}^{-1} \mathbf{Y} \tag{6}$$

where the vector $\mathbf{k}$ with components $k_n = C(x', x^{(n)})$ describes

$$\sigma_{y'}^2 = \kappa - \mathbf{k}^T \mathbf{C}^{-1} \mathbf{k} \tag{7}$$

the similarity of the new molecule to the ones in the training set, $\kappa = C(x', x')$, and $\mathbf{C}$ is given by eq 3.

The case of multiple new molecules is a straightforward extension of eqs 5−7.[3] Although we got an entire distribution as the result of the inference procedure for a new molecule, for the purposes of regression, we take the mean of the Gaussian distribution (eq 6) as the "best" estimate for the prediction. In addition, and quite unlike many other regression methods, the Gaussian Process also provides us with a variance, given by eq 7. The standard deviation $\sigma_{y'}$ can be used as an indicator of where a new molecule lies within the descriptor space of the model. If this standard deviation is very large, it will indicate that the new molecule is well outside the descriptor space covered by the training data. In such a case, the prediction is not to be trusted very much. A crucial point about this variance is that it *only* depends on the descriptor vectors, as can be seen from eq 7, and not at all on the property values $\mathbf{Y}$. A variance for a test set point is obtained by adding parameter θ_3 to the variance $\sigma_{y'}^2$ from eq 7, the former accounting for observational error.

The next question is how to choose hyperparameters $\Theta = \{\theta_1, \theta_2, \theta_3, r_i, (i = 1...K)\}$. In some cases, there is a good reason to estimate these outside the framework of formal Bayesian inference, because, for example, θ_3 represents our knowledge about the inherent noise in the data or equivalently our choice of tradeoff between smoothness and good fit. On the other hand, the length scale parameters r_i ought to be inferred from the data itself: this leads to a way of automatically determining which descriptors are most relevant. In the Bayesian inference framework, estimating the hyperparameters can be done by another application of Bayes' theorem: assume a prior distribution for the hyperparameters, determine the posterior distribution given the data, and then integrate over all hyperparameters:

$$P(y(\mathbf{x}')|\mathcal{D}) = \int P(y(\mathbf{x}')|\Theta, \mathcal{D})\, P(\Theta|\mathcal{D})\, \mathrm{d}\Theta \tag{8}$$

Here, $P(y(\mathbf{x}')|\Theta, \mathcal{D})$ is given by eq 5. The posterior on the hyperparameters $P(\Theta|\mathcal{D})$ is, according to the basic Bayesian update rule,

$$P(\Theta|\mathcal{D}) = \frac{1}{E} P(\mathbf{Y}|\mathbf{X}, \Theta)\, P(\Theta) \tag{9}$$

Here, E is a normalization constant and $P(\mathbf{Y}|\mathbf{X}, \Theta)$ is the marginal likelihood, which plays the role of the likelihood in eq 9.

Of course the integration over the hyperparameters (eq 8) is a formidable task that can be tackled numerically, and we will discuss a method for such integration in section 2.2.4. However, the most common approach is to approximate the integral (eq 8) by using the *most probable* set of hyperparameters Θ_{MP}:

$$P(y(\mathbf{x}')|\mathcal{D}) \simeq P(y(\mathbf{x}')|\Theta_{MP}, \mathcal{D})$$

Finding Θ_{MP} corresponds to finding the maximum of the posterior distribution of the hyperparameters (eq 9), which becomes the same as the maximum likelihood if we assume a uniform prior distribution. Below, we give the formula for the negative logarithm of the marginal likelihood, which thus has to be minimized:

$$\Lambda(\Theta) = -\ln P(\mathbf{Y}|\mathbf{X}, \Theta) = \frac{1}{2}\ln(\det \mathbf{C}) + \frac{1}{2}\mathbf{Y}^T \mathbf{C}^{-1} \mathbf{Y} + \frac{N}{2}\ln 2\pi \tag{10}$$

In the next section, we will discuss approaches for determining the hyperparameters.

2.2. Hyperparameter Tuning. In this section, we present four techniques for determining the hyperparameters, listed in order of increasing computational time that they demand: fixed hyperparameters, forward variable selection procedure, optimization by the conjugate gradient method, and nested sampling. The latter three techniques have the ability to identify and select relevant descriptors. The nested sampling approach is the only technique involving a search in the full hyperparameter space.

2.2.1. Fixed Hyperparameters. Training of the model for each set of hyperparameter values involves inversion of the covariance matrix of size $N \times N$, the most computationally demanding step of the training process which scales as $O(N^3)$. For large data sets ($N > 1000$), the computational time becomes very demanding if all the hyperparameters need to be optimized. This problem led us to search for appropriate fixed values for some hyperparameters which could be suitable for many data sets. We have found that setting length scales and θ_2 in the following way works well for the majority of the data sets we have considered:

$$\theta_2 = \sqrt{N}\sigma_Y, \qquad r_i = r_i^0 \equiv 4\sqrt{K}\sigma_X^i \tag{11}$$

where σ_Y is the standard deviation of $\mathbf{Y}$ values and σ_X^i is the standard deviation of the ith column of matrix of descriptor values $\mathbf{X}$.

With fixed θ_2 and r_i^0, we choose hyperparameters θ_1 and θ_3, which minimize the log marginal likelihood (eq 10). A crude net search in the region $\theta_3 \in (0, 1]$, $\theta_1 \in [(\sqrt{N} - 6)\sigma_Y, (\sqrt{N} + 8)\sigma_Y]$ is usually sufficient.

For the very large training sets, it might be necessary to omit the search for θ_1 and θ_3 hyperparameters and take the following approximation:

$$\theta_1 = (\sqrt{N} + 4)\sigma_Y, \qquad \theta_3 = 0.4$$

In order to estimate appropriate values for the length scales, we looked at typical values of expression $(x_i^{(n)} - x_i^{(n')})^2$ in the covariance function (eq 4). It is easy to show that its average over n and n' equals

$$\frac{1}{N^2}\sum_{n=1}^{N}\sum_{n'=1}^{N}(x_i^{(n)} - x_i^{(n')})^2 = 2s_i^2$$

where s_i is a sample standard deviation of the ith column of matrix **X**. If we take $r_i \ll \sigma_X^i$ for all i, then the exponential term in eq 4 will tend to zero. If we take $r_i \gg \sigma_X^i$ for all i, then the exponential term in eq 4 will be close to 1. In both cases, the information from the training set will not influence the covariance matrix. Therefore, it is reasonable to take $r_i \propto \sqrt{K}\sigma_X^i$. The final values (eq 11) for the length scales and θ_2 were found experimentally on a number of different data sets.

One of the disadvantages of this approach is that some hyperparameters have fixed values which might not be suitable for some data sets or types of descriptors. Also, having fixed length scales means that we lose the ability to differentiate between important and unimportant descriptors.

2.2.2. Forward Variable Selection. Although the above hyperparameter tuning works well with a large pool of descriptors, it does not provide any insight into the structure−activity or structure−property relationship. Identification and selection of relevant descriptors from a large collection is an important step in QSAR modeling. Also, when descriptor calculation is computationally costly, it improves the execution time of the final model.

One common approach to descriptor selection is forward variable selection.[15] The descriptor selection is driven by minimizing the log marginal likelihood (eq 10) and proceeds as follows:

(1) Hyperparameters θ_1, θ_2, and θ_3 are obtained as described in section 2.2.1 and are fixed.

(2) All hyperparameters r_i are set to $r_i = 1000 r_i^0$. This is equivalent to minimizing the importance of all descriptors in the model. The model is trained, and log marginal likelihood Λ is calculated. Let us denote it Λ_{base}.

(3) Each descriptor in turn is brought into the model. This means setting $r_i = r_i^0$ for the ith descriptor, keeping the rest of the length scales unchanged. The descriptor which brings the most improvement to the log marginal likelihood is permanently added to the subset of descriptors. This model will now be the base model, and Λ_{base} is updated.

(4) Step 3 is repeated until there is no improvement in the model.

(5) The best model with the best subset of descriptors will be the one with the smallest log marginal likelihood. In addition, the models created with different subsets of descriptors can be tried with length scales set to $r_i^m = 4\sqrt{m}\sigma_X^i$, where m is a number of descriptors present in the model.

The described approach can be computationally expensive for large training sets with a big pool of descriptors. During variable selection, about $K(K+1)/2$ Gaussian Processes models have to be built; that means $K(K+1)/2$ inversions of the covariance matrix of size $N \times N$ have to be performed. Also, this method has the usual disadvantages associated with forward variable selection. Namely, in the case of a big pool of descriptors, the order of trials will influence which one of a set of correlated descriptors is selected, and this may not be the most chemically meaningful descriptor. Also, serial selection means that descriptors that are important in groups or pairs but not significant as individual descriptors may not be selected.

2.2.3. Optimization of Hyperparameters. It may be possible to determine the hyperparameters by minimization of the log marginal likelihood (eq 10), employing methods such as conjugate gradient optimization[16,17] in the whole hyperparameter space. One difficulty with this approach is the existence of multiple minima of the log marginal likelihood and hence the possibility of being "trapped" in a local minimum during the gradient descent. The main problem with conjugate gradient optimization is that it did not work well when used in the full space of hyperparameters θ and r.

Due to this, we adopted the following approach. We initialize θ_2 and r_i as in eq 11 and obtain θ_1 and θ_3 as described in section 2.2.1. Then, we minimize the log marginal likelihood (eq 10) with respect to hyperparameters r_i by the conjugate gradient Polak−Ribiere method.[17] The formulas for the derivatives of the log marginal likelihood with respect to hyperparameters can be found in the book by MacKay.[2]

After the optimization is converged and the hyperparameters r_i are found, a descriptor selection procedure can be employed. It is based on the idea that if a length scale for a descriptor is very large then this descriptor is irrelevant. This approach is closely related to automatic relevance determination.[9,14,18,19]

Let us denote the optimized length scales as r_i^{opt} ($i = 1...K$). If for a certain descriptor $r_i^{\text{opt}} \gg r_i^0$, then this descriptor is not important and can be omitted from the model. Consider ratio $R_i = r_i^{\text{opt}}/r_i^0$, which reflects the relative importance of the ith descriptor. The smaller R_i is, the more important the corresponding descriptor is. We include in the model only descriptors for which $R_i < a$, where a is a given threshold. The appropriate threshold is chosen by considering all possible values for a and choosing the one which gives the minimum of the log marginal likelihood (eq 10).

It is difficult to say how much computational time this method demands. It depends on details of the implemented conjugate gradient method and on the required tolerance. The conjugate gradient descent process needs to be restarted after K iterations, where K is a number of dimensions in the problem, that is, the number of descriptors. In our experience, this method takes about $\alpha \times K \times 50$ likelihood evaluations, that is, inversions of the covariance matrix, where usually $\alpha \leq 4$.

2.2.4. Finding Hyperparameters by Nested Sampling. Nested sampling was introduced by Skilling[13] as a method to estimate the evidence for a model and generate samples from the posterior distribution (eq 9). It can also be used to find optimal values for the hyperparameters. Briefly, the idea of the method is as follows. The prior space of hyperparameters is sampled randomly. We want to find values of hyperparameters which give the minimum of the log marginal likelihood (eq 10), that is, the maximum of likelihood. Some samples from the prior hyperparameter space which have low likelihood are replaced with new samples with higher likelihood values. At the end of the iterative process, we have points from hyperparameter space which correspond to high likelihood values; that means optimal hyperparameter values.

This approach has numerous advantages: it performs a search in the full space of hyperparameters θ and r, does not get "trapped" in a local minimum of log marginal likelihood, and works well in the case of multiple minima.

It explores a wide prior space of hyperparameters, so there is no danger of missing an important region of that space in contrast to the hyperparameter tuning methods described above. The drawback of the nested sampling method is that it is computationally expensive; for example, for the hERG data set, which will be considered in section 3.3, the nested sampling search took about 30 000 likelihood evaluations, that is, covariance matrix inversions.

We will give a brief description of the nested sampling approach; the details and derivations can be found in papers by Skilling.[13]

The normalization constant for the posterior distribution (eq 9) is also called the evidence; it is given by the equation

$$E = \int P(\mathbf{Y}|\mathbf{X},\Theta)\, P(\Theta)\, \mathrm{d}\Theta$$

where $P(\Theta)$ is a prior distribution for hyperparameters Θ and $P(\mathbf{Y}|\mathbf{X},\Theta)$ is the marginal likelihood; its logarithm is given by eq 10. This integral can be rewritten in the form

$$E = \int_0^1 L\, \mathrm{d}Z$$

with $\mathrm{d}Z = P(\Theta)\, \mathrm{d}\Theta$ and the likelihood $L = L(\Theta) = P(\mathbf{Y}|\mathbf{X},\Theta)$.

Our nested sampling algorithm follows closely algorithms suggested by Skilling[13] and used by Mukherjee et al.[20] The steps are as follows:

(1) Sample I (e.g., 50) points Θ_1, ..., Θ_I randomly from the prior distribution $P(\Theta)$. Evaluate their likelihoods $L(\Theta_1)$, ..., $L(\Theta_I)$. We will call these I points the live points. Initialize $E = 1$ and $Z_0 = 1$.

(2) Repeat until the evidence has been estimated to some desired accuracy (e.g., until condition $\max(L_i)\, Z_j/E < \epsilon$ for $\epsilon = 0.001$ is satisfied), for iterations $j = 1$, ..., J:

(a) Select the point with the lowest of the current likelihood values $L_j = \min_i L(\Theta_i)$.

(b) Set $Z_j = [I/(I+1)]^j$, and increment the evidence by $L_j w_j$, where $w_j = (Z_{j-1} - Z_{j+1})/2$.

(c) Remove the point of the lowest likelihood from the live points, and replace it with a new one drawn from the prior distribution within constraint on likelihood $L(\Theta) > L_i$ for all $i = 1$, ..., I.

(d) The removed point Θ_j gives the jth sample from the posterior distribution $P(\Theta|\mathscr{D})$.

(3) Add the contribution from the live sample points $I^{-1}[L(\Theta_1) + ... + L(\Theta_I)]Z_j$ to the accumulated evidence E.

(4) The sequence of discarded points Θ_j, $j = 1$, ..., J gives us samples from the posterior distribution $P(\Theta|\mathscr{D})$ given by eq 9. An appropriate probability weight should be assigned to each sample:

$$p_j = \frac{L_j w_j}{E} \qquad j = 1, ..., J \tag{12}$$

To find a new sample at step 2c, we have adopted an approach similar to one used by Mukherjee et al.[20] We create a smallest box enclosing the existing live points, expand it in each dimension by a small constant, and sample from the expanded box within the constraint under the prior distribution. Alternative approaches were suggested by Skilling.[13]

At the end of the nested sampling process, the live sample points will shrink to the area of high likelihood. Figure 1 illustrates the evolution of live samples in a one-dimensional case $\Theta = \theta_1$ when the likelihood function has two peaks. At the beginning of the iterative process ($j = 0$), the live points are distributed uniformly on the interval $[-2, 6]$. As the nested sampling process progresses, the live points move to the area of higher likelihood, and at 150 iterations, they concentrate only on one highest peak. At the end of the iterative process, the live points shrink to a small area around the maximum of the likelihood function. An illustration of posterior samples and live points in the multidimensional case will be discussed in section 3.3 (see Figure 2).

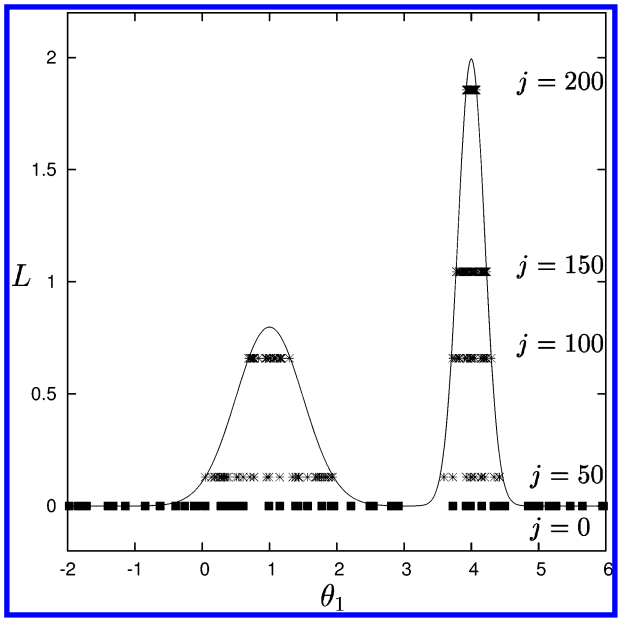

Figure 1. Illustration of the evolution of live samples during nested sampling in a 1D case. Positions of live samples are shown at the beginning of the process ($j = 0$) and after 50, 100, 150, and 200 iterations. The vertical position of points indicates the iteration, and the samples are plotted with an ordinate equal to the minimum of likelihoods of live points $L_{\min} = \min_i (L_i)$ at corresponding iteration j. The likelihood function $L = L(\theta_1)$ is shown as a solid curve.

The live point with the largest likelihood can be considered as an optimal choice for hyperparameters. Alternatively, we are able to use generated posterior samples to marginalize the prediction over all hyperparameter space, that is, the integral (eq 8) can now be estimated by

$$\int P(y(\mathbf{x}')|\Theta,\mathscr{D})\, P(\Theta|\mathscr{D})\, \mathrm{d}\Theta \simeq \sum_{j=1}^{J} P(y(\mathbf{x}')|\Theta_j,\mathscr{D})\, p_j \tag{13}$$

where probability weights p_j are given by eq 12.

The optimized length scales can now be used to automatically identify the relevant descriptors. If a length scale for a descriptor is small (e.g., in comparison with the standard deviation of the descriptor), then this descriptor is important. The descriptor selection procedure described in section 2.2.3 can be employed.

2.3. Data Sets. To make a comparison between the described technique and the other QSAR modeling methods, we have chosen two data sets from the literature.

Benzodiazepine Data Set (BZD). A set of 245 ligands for the benzodiazepine receptor was kindly provided to us by F. Burden. The observed in vitro binding affinities measured by the inhibition of [3H] diazepam binding are expressed as pIC_{50}. Burden[5] used this data set with molecular

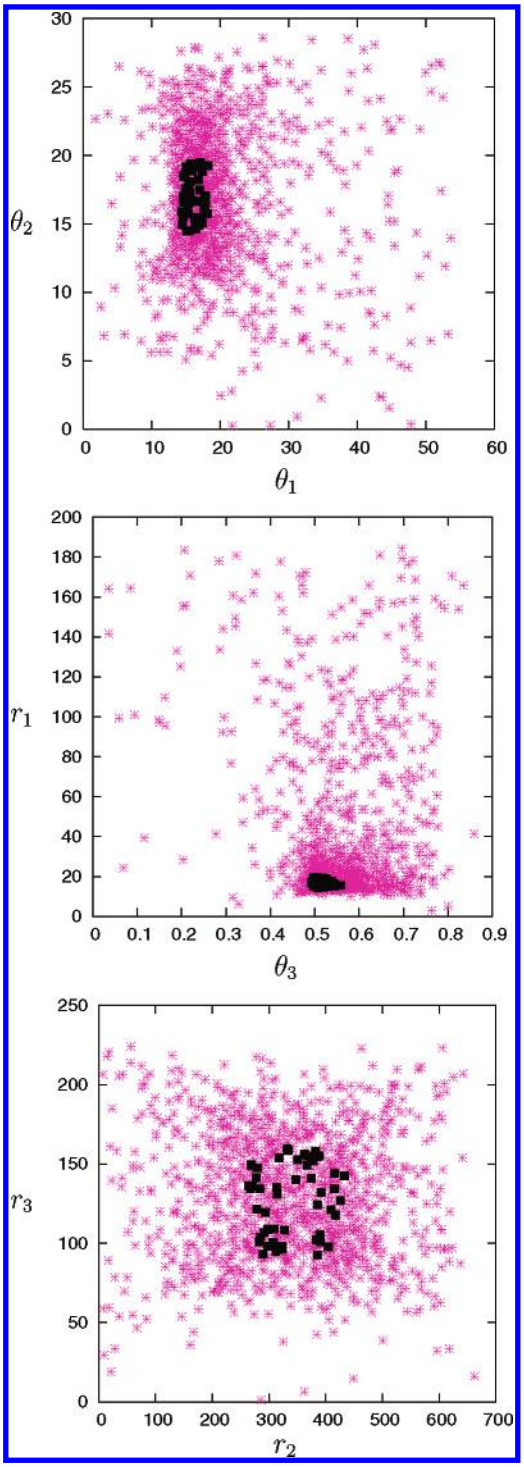

Figure 2. Posterior samples (pink) and live points (black) in a nested sampling model for hERG inhibition.

indices descriptors to produce a number of QSAR models trained on various statistical methods including Gaussian Processes.

Blood−Brain Barrier Data Set (BBB). The ability to predict blood−brain penetration is very important in drug development. For central nervous system (CNS) therapeutic targets, good penetration is an absolute requirement, but for non-CNS targets, blood−brain barrier penetration is undesirable as it is a potential cause of side effects. Because of its importance, many efforts have been given to developing predictive models for the blood−brain barrier.

A BBB partition set of 106 compounds was reported by Rose et al.[21] This data set was modeled by Winkler and Burden[14] using different families of descriptors by Bayesian neural nets. The structures for 106 compounds, their observed log *BB* values, and feature-based descriptors were kindly made available to us by D. Winkler.

We have also built Gaussian Processes models for two additional ADME data sets which were compiled in house.

hERG Inhibition Data Set (hERG). Inhibition of the human ether-a-go-go-related gene (hERG) potassium channel by medications has been linked to QT prolongation, a side effect linked to life-threatening ventricular arrhythmias including Torsade de Pointes. Data on hERG potassium channel blockers were derived from various literature sources. A total of 137 compounds with patch-clamp pIC_{50} values for inhibition of the hERG channel expressed in mammalian cells were carefully selected. This data set is chemically diverse and covers a good range of pIC_{50} values. The references to literature sources and the data set are given in the Supporting Information.

Solubility at pH 7.4 Data Set (Sol74). The solubility of drug compounds is a fundamental property in relation to their in vivo ADME behavior and deserves attention at the early stage of drug discovery. A large number of in silico models have been developed to predict this property and thus assist the drug discovery process.[8] However, the majority of these models predict the solubility of a compound in its neutral form. Predictions of the apparent solubility of neutral and ionized druglike compounds at physiological pH would be a very important addition to the current models for ADME properties. A compilation of high-quality solubility data measured in buffered solution at pH 7.4 (log $S_{7.4}$ with $S_{7.4}$ in μM) was gathered from BioFocus DPI's StARLITe.[22] Only those measurements that were determined between 30 ± 5 °C were considered. A careful search led to a set of 592 diverse druglike compounds.

2.4. Molecular Descriptors. We have modeled BZD and BBB data sets with the same descriptors as those used by Burden and Winkler.[5,14] We will describe these descriptors in section 3.

To model the hERG and Sol74 data sets, we have used 330 in-house molecular descriptors. A total of 321 SMARTS-based descriptors and nine whole molecule properties such as logP, molecular weight, and the McGowan's volume, Vx, were calculated. The SMARTS-based descriptors are counts of atom type (e.g., fluorine atom) and counts of functionalities (e.g., ketone). They also include descriptors explicitly designed to take into account known ligand pharmacophore features. For instance, hERG channel inhibitors often contain positively charged nitrogen, at pH 7.4, as well as a number of hydrophobic moieties in the vicinity.[23] A descriptor was specifically defined to describe such atomic arrangements.

2.5. Set Preparation. Calculated descriptors were subjected to a descriptor selection step that removed descriptors with low variance and low occurrence. More precisely, descriptors with a standard deviation less than 0.0005 and descriptors represented by less than 2−4% of compounds were excluded from the set. Also, highly correlated descriptors were excluded (with pairwise correlation exceeding 0.95), so that just one of the pair remained.

To assess the predictive power of a model, we split data into training and independent test sets. To separate the test

set, we use two approaches. First is a *Y*-based approach—data set is sorted by *Y* values, and every fifth or sixth compound, depending on the percentage needed, is taken to the test set. The second technique is based on a cluster analysis. Compounds are clustered using an unsupervised nonhierarchical clustering algorithm developed by Butina.[24] The cluster analysis of the chemical structures is based on Daylight fingerprints and the Tanimoto similarity index. The algorithm identifies dense clusters where similarity within each cluster reflects the Tanimoto value used for the clustering and singletons, that is, compounds that do not belong to any cluster. Once the clusters are formed, the cluster centroids and singletons are put into the training set. Random selection from the remaining compounds is used to fill the test set.

2.6. Model Validation. To evaluate the quality of the models, we use the following statistics on training and test sets: root-mean-square error

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(Y_i^{\mathrm{obs}} - Y_i^{\mathrm{pred}})^2}$$

where N is the set size and R^2, the coefficient of determination:

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

Although the coefficient of determination R^2 is the most appropriate statistic, the squared correlation coefficient has often been used in QSAR modeling. Therefore, we will also quote $r_{\mathrm{corr}}{}^2$, the squared correlation coefficient between predicted and observed *Y*:

$$r_{\mathrm{corr}}{}^2 = \frac{\left[\sum_{i=1}^{N}(Y_i^{\mathrm{obs}} - \overline{Y_i^{\mathrm{obs}}})(Y_i^{\mathrm{pred}} - \overline{Y_i^{\mathrm{pred}}})\right]^2}{\sum_{i=1}^{N}(Y_i^{\mathrm{obs}} - \overline{Y_i^{\mathrm{obs}}})^2 \sum_{i=1}^{N}(Y_i^{\mathrm{pred}} - \overline{Y_i^{\mathrm{pred}}})^2}$$

The squared correlation coefficient $r_{\mathrm{corr}}{}^2$ measures the general association between $\mathbf{Y^{obs}}$ and $\mathbf{Y^{pred}}$, while the R^2 statistic measures the similarity of the magnitudes of $\mathbf{Y^{obs}}$ and $\mathbf{Y^{pred}}$.[25] In other words, $r_{\mathrm{corr}}{}^2$ describes goodness of fit or prediction around the regression line between $\mathbf{Y^{obs}}$ and $\mathbf{Y^{pred}}$, but the regression line does not have to have a slope equal to 1 and an intercept of 0. R^2 measures the fit around this identity line. For a good QSAR model, these two statistical measures should be close.[26]

3. RESULTS

3.1. Benzodiazepine Data Set. The BZD set of 245 compounds was modeled by Burden by neural network methods and by Gaussian Processes[5] and by Burden et al. by Bayesian neural networks with automatic relevance determination.[19] Burden used a set of computed molecular indices: the Randic index, the valence modification to the Randic index by Kier and Hall, and an atomistic index developed by the author. An additional index was also designed to identify aromatic atoms and hydrogen atom donors and acceptors. Further indices used include counts of rings of various sizes and counts of some functional groups.[5]

Table 1. BZD Set with Molecular Indices Descriptors

Our Results[a]							
		training set			test set		
method	desc.	RMSE	R^2	$r_{\mathrm{corr}}{}^2$	RMSE	R^2	$r_{\mathrm{corr}}{}^2$
PLS	3[b]	0.63	0.32	0.32	0.52	0.51	0.53
GP-Basic	38	0.53	0.51	0.52	0.52	0.51	0.53
GP-FVS	15	0.53	0.51	0.52	0.52	0.52	0.54
GP-Opt	9	0.47	0.62	0.62	0.55	0.47	0.51
GP-Nest	38	0.44 (0.14)[c]	0.67	0.68	0.46 (0.15)[c]	0.63	0.65

Burden Results[5]							
		training set			test set		
method	desc.	SEF[d]	R^2	$r_{\mathrm{corr}}{}^2$	SEP[e]	R^2	$r_{\mathrm{corr}}{}^2$
MLR	39	0.18		0.47	0.20		0.32
ANN	22	0.13		0.73	0.14		0.66
BRANN	39	0.12		0.75	0.12		0.71
GPmodel	39	0.12		0.76	0.14		0.66
GPlinear	39	0.12		0.78	0.13		0.71

[a] Training set 208 compounds, test set 37 compounds. [b] Number of PLS components. [c] In brackets: RMSE for unit interval scaled Y values. [d] SEF = Standard error of fit. [e] SEP = Standard error of prediction.

To compare the present technique with the ones considered by Burden, we calculated the same 59 descriptors. The Randic and Kier—Hall indices were calculated through the E-Dragon software.[27,28] The remaining indices were calculated using SMARTS strings based on the descriptors' definitions given by Burden.[5] After descriptors with low variance and occurrence less than 2% were excluded, 38 descriptors remained.

The actual data split into training and test sets used to model BZD by Burden[5] was not available to us. We have separated the same percentage of total data to the test set, 15% of the compounds, using the *Y*-based split technique.

The results of modeling of the BZD data set are given in Table 1. PLS stands for a model obtained by partial least squares;[29] GP-Basic, GP-FVS, GP-Opt, and GP-Nest are Gaussian Processes models (GP) with fixed hyperparameters (GP-Basic) and hyperparameters obtained by forward variable selection (GP-FVS), by conjugate gradient optimization (GP-Opt), and by nested sampling (GP-Nest). One can see that the performance of the Gaussian Processes technique with nested sampling is superior to that of other models.

Also in Table 1, we give results reported by Burden;[5] they were obtained by using Gaussian Processes, multiple linear regression, backpropagation neural networks (ANN), and Bayesian neural networks (BRANN). Statistics reported by Burden were obtained on *Y* values which were scaled to the (0, 1) interval. We obtained performance statistics on unscaled data, but for comparison, we give in Table 1 RMSE values for scaled data for the Gaussian Processes with nested sampling model. Overall, our Gaussian Processes models slightly underperform those of Burden, although this could be due to a difference in the training and test sets used.

To analyze the influence of the training/test sets split on the modeling results, we have repeated computations for 18

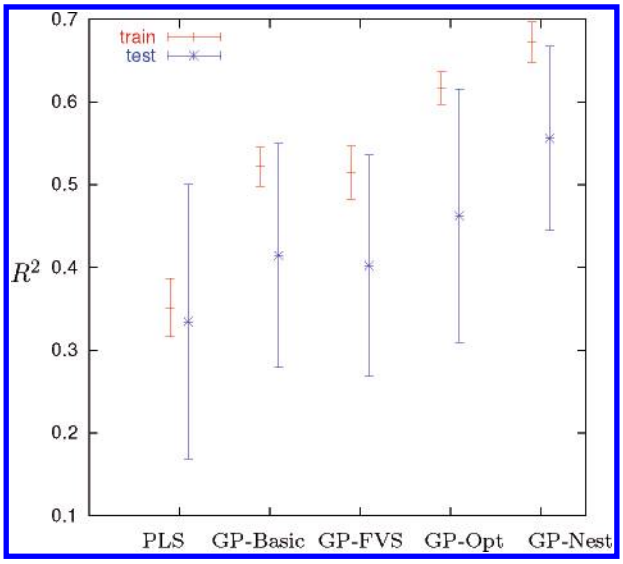

Figure 3. Modeling of BZD data set using different training/test splits. The average R^2 values together with error bars (one standard deviation) are shown for each modeling technique. R^2 statistics for the training set are plotted in red and for the test set in blue. Average values are calculated over 19 different set splits.

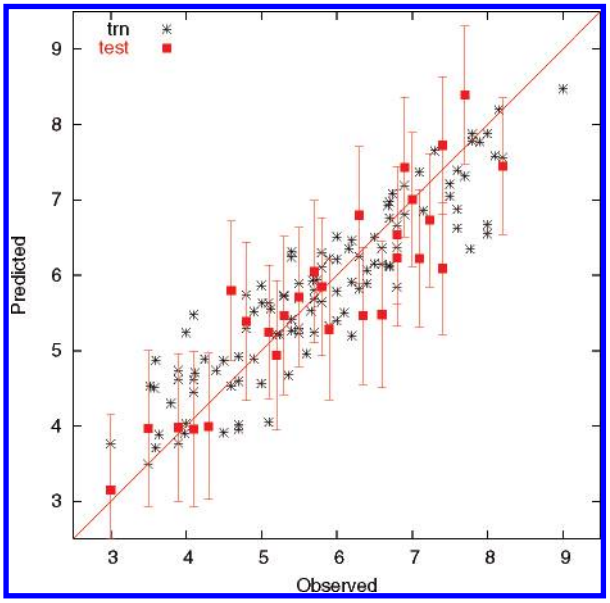

Figure 4. hERG data set. Predicted pIC_{50} values versus those observed for the Gaussian Processes model with optimization. Training set compounds are depicted by stars, test set compounds by squares. On the test set, $R^2 = 0.81$. Test set compounds are shown with error bars.

different splits of the original data set. Half of the splits were performed by the *Y*-based approach and the other half by the cluster analysis method with a different Tanimoto index. The results are given in Figure 3, which shows the average R^2 obtained over 19 available different training/test sets together with standard deviations for each modeling technique (average values and standard deviations for r_{corr}^2 follow closely those for the R^2 statistic). These results confirm great variation in the predictive performance of the models due to the training/test set splits.

Burden reports two Gaussian Processes models—with optimization of the hyperparameters and a "linear" model where the number of hyperparameters to optimize is reduced. The covariance function for these models is the same as that which we have used (eq 4). In the linear Gaussian Processes model, all the length scales are set to the same value $r_i^2 = 0.5$ (the data is mean-centered and scaled) and the noise parameter set $\theta_3 = 0.1$. We have tried and failed to repeat the results of that model. The above values of length scales lead to an overtrained model with no predictive ability. Therefore, we are unable to reproduce this model and, hence, better explain its performance on the data set.

Table 2. BBB Set with Feature-Based Descriptors

Our Results[a]							
		training set			test set		
method	desc.	RMSE	R^2	r_{corr}^2	RMSE	R^2	r_{corr}^2
PLS	3[b]	0.52	0.59	0.59	0.40	0.73	0.74
GP-Basic	7	0.50	0.61	0.62	0.39	0.74	0.77
GP-FVS	3	0.50	0.61	0.61	0.39	0.74	0.75
GP-Opt	7	0.47	0.66	0.66	0.36	0.77	0.78
GP-Nest	7	0.44	0.69	0.70	0.34	0.81	0.82
Winkler & Burden Results[14]							
		training set			test set		
method	desc.	SEE[c]	R^2	r_{corr}^2	SEP[d]	R^2	r_{corr}^2
BNN	7	0.37		0.81	0.54		0.65

[a] Training set 85 compounds, test set 21 compounds. [b] Number of PLS components. [c] SEE = Standard error of estimation. [d] SEP = Standard error of prediction.

Table 3. hERG and Sol74 Data Sets with 2D Descriptors

		training set		test set	
method	desc.	RMSE	R^2	RMSE	R^2
hERG Set[a]					
PLS	2[c]	0.81	0.63	0.69	0.74
GP-Basic	166	0.61	0.79	0.67	0.76
GP-FVS	17	0.65	0.76	0.61	0.80
GP-Opt	26	0.56	0.82	0.60	0.81
GP-Nest	166	0.58	0.81	0.65	0.77
Sol74 Set[b]					
PLS	4[c]	0.86	0.63	0.80	0.60
GP-Basic	174	0.65	0.79	0.73	0.67
GP-FVS	61	0.66	0.79	0.73	0.66
GP-Opt	66	0.49	0.88	0.71	0.68
GP-Nest	174	0.45	0.90	0.68	0.71

[a] Training set 110 compounds, test set 27 compounds. [b] Training set 474 compounds, test set 118 compounds. [c] Number of PLS components.

3.2. Blood−Brain Barrier Data Set. To model the BBB data set, we used seven feature-based descriptors, including the number of hydrogen-bond donors and acceptors, rotatable bonds, hydrophobes, logP, molecular weight, and polar surface area, which were made available to us by D. Winkler. The exact split into training and test sets used by Winkler and Burden[14] in their study was not available to us; we reserved 20% of the initial set for the test set using the *Y*-based set separation technique.

The results of modeling the BBB data set are given in Table 2. Gaussian Processes with nested sampling appears to be the preferential modeling method. For comparison, in Table 2, we have also given the statistics for the best model obtained by Winkler and Burden.[14] They have used Bayesian neural nets to build QSAR models. Our models show a slightly inferior fit to the training set but have better predictive ability on the test set. This difference is due to the fact that three out of the four outliers detected by Rose et al.[21] were included in the training set. Placing some of

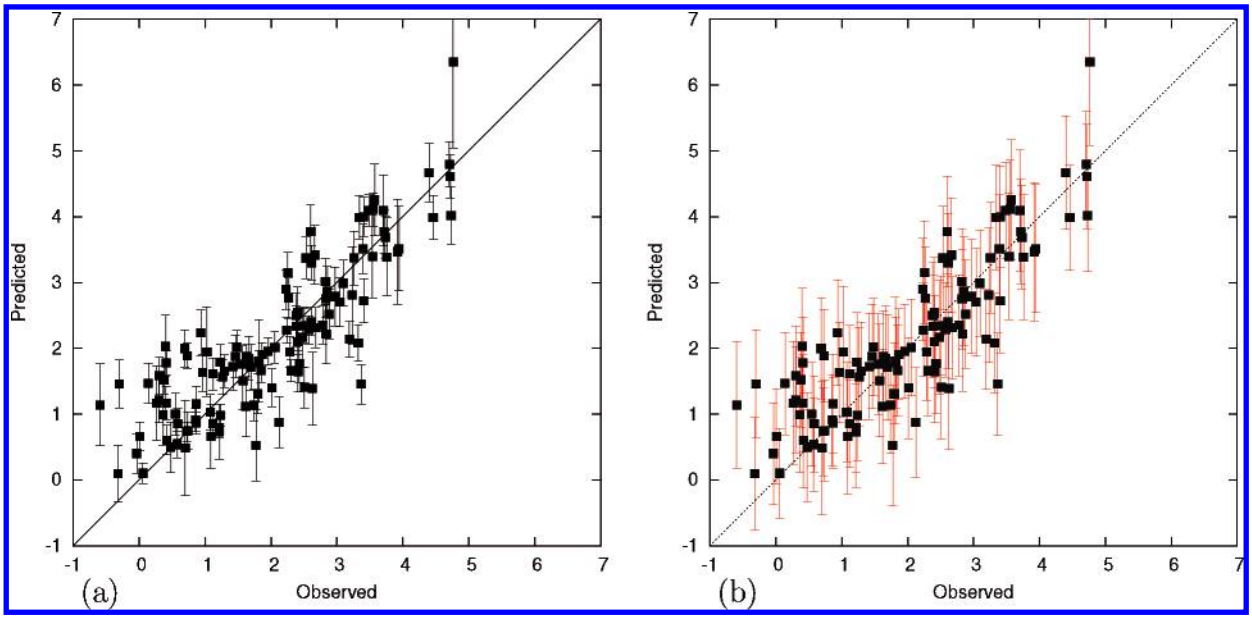

Figure 5. Solubility at pH 7.4 test set. Predicted log S (S in μM) values with error bars versus those observed for the Gaussian Processes model with nested sampling. On the test set, $R^2 = 0.71$. (a) Error bars do not include contribution of the noise variance θ_3. (b) Error bars including the noise variance.

Table 4. Computational Times for hERG Data Set[a]

method	time (min)[b]
PLS	0.2
GP-Basic	2.3
GP-FVS	19
GP-Opt	13
GP-Nest	170

[a] hERG training set—110 compounds, 166 descriptors. [b] The computations were performed on a 2.8 GHz Pentium 4 CPU with a 512 kB cache.

these outliers in the test set leads to a decrease in R^2 for the test set but not a significant improvement in the training set fit; for example, for a training/test set split with the four outliers included in the test set, the nested sampling model gives on the training set $R^2 = 0.77$ ($r_{corr}^2 = 0.77$) and on the test set $R^2 = 0.44$ ($r_{corr}^2 = 0.63$). This result looks similar to the result of Winkler and Burden (the R^2 statistics are not available for their model).

The descriptors which were selected by GP-FVS are the polar surface area, logP, and the number of rotational bonds. This is consistent with results obtained by the automatic relevance determination method by Winkler and Burden.[14]

3.3. hERG Data Set. In this section, we present results of modeling the hERG data set using in-house descriptors (see section 2.4). After preliminary descriptor selection, 166 descriptors remained; the threshold for low occurrence was set at 3%. We kept 20% of the compounds in test set. The set selection was based on a cluster analysis with a Tanimoto level of 0.7. The results of QSAR modeling for the hERG data set are given in Table 3.

The Gaussian Processes technique with conjugate gradient optimization (GP-Opt) produced the best model for this data set. On the test set, it achieved RMSE = 0.60, $R^2 = 0.81$ and $r_{corr}^2 = 0.81$. Figure 4 shows the graph of predicted pIC_{50} values versus those observed for hERG inhibition for this model.

The model for hERG inhibition used 26 descriptors out of the original 174 descriptors. The Gaussian Process model selected meaningful descriptors that describe chemical features known to influence a compound's affinity to the hERG channel. One of the most known pharmacophore features leading to a high pIC_{50} value is the presence of a protonable nitrogen at pH 7.4 surrounded by hydrophobic moieties. The descriptor that was specifically designed to describe this compound feature was selected. The size of the molecule was also found to influence the pIC_{50} value. Other known features such as the presence of aromatic ring bearing hydrophobic substituents such as fluorine atoms or the presence of hydrogen-bond donor and acceptor pairs separated by six bonds were also selected. The current model is also based on descriptors reporting the presence or absence of amide, ketone, and ether functionalities. Again, these are in agreement with published pharmacophore models.[30]

The Gaussian Processes method provides us with the standard deviation for each prediction. Figure 4 shows error bars for the compounds of the test set. Because observed values for the test set compounds contain the same amount of noise as the training set compounds, the error bars include contribution of the noise variance θ_3. These error bars give a good representation of the standard deviation for the difference between predicted and observed properties.

The nested sampling technique has also produced a good model for hERG inhibition with $R^2 = 0.77$ on the test set. For the purpose of illustration, Figure 2 shows the posterior samples and live points at the end of the nested sampling process for hyperparameters θ_1, θ_2, and θ_3 and three length scales r_1, r_2, and r_3. Length scale r_1 corresponds to the descriptor for positively charged nitrogen at pH 7.4 with three neighboring sp^3 carbon and aromatic rings in the vicinities; r_2 corresponds to the descriptor for sp^2 carbon, and r_3 corresponds to the logP descriptor. At the beginning of the iterative process, the live points were uniformly distributed in the shown area (the prior domain for hyperparameters). As the points with lower likelihood become rejected, the live points move to the area of higher likelihood. At the end of the iterative process, the live points shrink to a relatively

small area which pinpoints the location of optimal values of hyperparameters.

Table 4 shows actual computational times used by the different techniques to model the hERG data.

3.4. Solubility at pH 7.4 Data Set. The results of QSAR modeling for the solubility at pH 7.4 data set are given in Table 3. A total of 174 descriptors remained after descriptor selection; the threshold for low occurrence was set at 5%. Descriptor log *P* was not included in the set. The training and test set selection is based on cluster analysis with the Tanimoto level set at 0.7. A total of 20% of the original set was allocated to the test set.

The nested sampling technique produced the best model for this data set. The model has a good predictive ability on the test set where it achieved RMSE = 0.68, $R^2 = 0.71$ and $r_{corr}^2 = 0.72$. Figure 5 shows the graph of predicted solubility log *S* at pH 7.4 values versus those observed for the test set for this model. Error bars in Figure 5a do not include contribution of the noise variance θ_3. One compound (top right) has the largest error bar, which indicates that this compound lies outside the descriptor space of the model. The difference of this compound from the others can be seen more clearly by looking at the error bars not including the noise variance. Error bars in Figure 5b include contribution of the noise variance and give a good representation of the standard deviation for the difference between predicted and observed properties.

4. CONCLUSIONS

We have demonstrated how the Gaussian Processes technique can be applied to building QSAR regression models of ADME properties and explored methods for optimization of these models. The main advantages of this computational method are as follows:

- It does not require subjective a priori determination of parameters such as variable importance or network architectures.
- The method does not need cross-validation. The solution is obtained by minimizing the log marginal likelihood, which directly prevents the model from overtraining.
- The Gaussian Processes method works well for a big pool of descriptors. Most of the techniques for hyperparameter tuning, which we have described, have an inherent ability to select the important descriptors.
- The method is able to model nonlinear relationships.
- This technique furnishes us with the standard deviation for each prediction, which is a measure of how close the new point is to the training points in the descriptor space, that is, how applicable the model is. The best use of this error bar is for the detection of outliers for which the generated model is not expected to work as well. Adjusted to account for the noise present in observed values, these error bars are a good representation of the standard deviation for the difference between predicted and observed property values.
- The Gaussian Processes technique is sufficiently robust for automatic model generation.

If computational resources permit, our technique of choice for finding hyperparameters would be the nested sampling approach, which searches full hyperparameter space in an unbiased way.

The Gaussian Processes method has proved to be comparable to and sometimes to exceed artificial neural networks in performance. All these features make the Gaussian Processes technique a powerful modeling tool for QSAR and ADME problems.

ACKNOWLEDGMENT

We thank Frank Burden and Dave Winkler for making available the benzodiazepine and blood-brain barrier data sets. We are grateful to Frank Burden for discussing with us the procedure he used to obtain QSAR models. We thank Chematica group of BioFocus DPI for making available the solubility data set from their StARLITe database.

Supporting Information Available: Structures in SMILES format and observed values and descriptors for 245 compounds of the benzodiazepine data set; SMILES, observed values, and references to data sources for the 137 compounds of the hERG data set; SMILES and observed values for the 118 test set compounds of the Sol74 data set. This material is available free of charge via the Internet at http://pubs.acs.org.

REFERENCES AND NOTES

(1) Segall, M. D.; Beresford, A. P.; Gola, J. M. R.; Hawksley, D.; Tarbit, M. H. Focus on Success: Using a Probabilistic Approach to Achieve an Optimal Balance of Compound Properties in Drug Discovery. *Expert Opin. Drug Metab. Toxicol.* **2006**, *2*, 325−337.

(2) MacKay, D. J. C. *Information Theory, Inference, and Learning Algorithms*; Cambridge University Press: Cambridge, United Kingdom, 2003.

(3) Rasmussen, C. E.; Williams, C. K. I. *Gaussian Processes for Machine Learning*; The MIT Press: Cambridge, MA, 2006.

(4) Gaussian Processes Resources. The Gaussian Processes Web Site. http://www.gaussianprocess.org (accessed Feb 10, 2007).

(5) Burden, F. R. Quantitative Structure−Activity Relashionship Studies Using Gaussian Processes. *J. Chem. Inf. Comput. Sci.* **2001**, *41*, 830−835.

(6) Enot, D.; Gautier, R.; Le Marouille, J. Gaussian Process: An Efficient Technique to Solve Quantitative Structure−Property Relationship Problems. *SAR QSAR Environ. Res.* **2001**, *12*, 461−469.

(7) Tino, P.; Nabney, I. T.; Williams, B. S.; Losel, J.; Sun, Y. Nonlinear Prediction of Quantitative Structure−Activity Relationships. *J. Chem. Inf. Comput. Sci.* **2004**, *44*, 1647−1653.

(8) Schwaighofer, A.; Schroeter, T.; Mika, S.; Laub, J.; Laak, A. T.; Sulzle, D.; Ganzer, U.; Heinrich, N.; Muller, K. R. Accurate Solubility Prediction with Error Bars for Electrolytes: A Machine Learning Approach. *J. Chem. Inf. Model.* **2007**, *47*, 407−424.

(9) Neal, R. M. *Bayesian Learning for Neural Networks*; Springer: New York, 1996.

(10) Cartmell, J.; Enoch, S.; Krstajic, D.; Leahy, D. E. Automated QSPR through Competitive Workflow. *J. Comput.-Aided Mol. Des.* **2005**, *19*, 821−833.

(11) Zhang, S.; Golbraikh, A.; Oloff, S.; Kohn, H.; Tropsha, A. A Novel Automated Lazy Learning (ALL-QSAR) Approach: Method Development, Applications, and Virtual Screening of Chemical Databases Using Validated ALL-QSAR Models. *J. Chem. Inf. Model.* **2006**, *46*, 1984−1995.

(12) Gibbs, M.; Mackay, D. J. C. Efficient Implementation of Gaussian Processes, 1997. University of Cambridge, Cavendish Laboratory, David MacKay Web site. http://www.inference.phy.cam.ac.uk/mackay/BayesGP.html (accessed Feb 10, 2007).

(13) Skilling, J. Nested Sampling for Bayesian Computations. Presented at Valencia/ISBA 8th World Meeting on Bayesian Statistics, Benidorm, Spain, June 1−6, 2006; University of Cambridge, Cavendish Laboratory, David MacKay Web site. http://www.inference.phy.cam.ac.uk/bayesys/Valencia.pdf (accessed Feb 10, 2007).

(14) Winkler, D. A.; Burden, F. R. Modelling Blood−Brain Barrier Partitioning Using Bayesian Neural Nets. *J. Mol. Graphics Modell.* **2004**, *22*, 499−505.

(15) Everitt, B. S.; Dunn, G. *Applied Multivariate Data Analysis*, 2nd ed.; Arnold: London, 2001.

(16) Shewchuk, J. R. An Introduction to the Conjugate Gradient Method Without the Agonizing Pain, 1994. Carnegie Mellon University, School of Computer Science Web site. http://www.cs.cmu.edu/quake-papers/painless-conjugate-gradient.pdf (accessed Nov 2007).

(17) Press, W. H.; Flannery, B. P.; Teukolsky, S. A.; Vetterling, W. T. *Numerical Recipes in C: The Art of Scientific Computing*; Cambridge University Press: Cambridge, U. K., 1988.
(18) MacKay, D. J. C. Bayesian Methods for Backpropagation Networks. In *Models of Neural Networks III*; Domany, E., Van Hemmen, J.L., Schulten, K., Eds.; Springer-Verlag: New York, 1994.
(19) Burden, F. R.; Ford, M. G.; Whitley, D. C.; Winkler, D. A. Use of Automatic Relevance Determination in QSAR Studies Using Bayesian Neural Networks. *J. Chem. Inf. Comput. Sci.* **2000**, *40*, 1423−1430.
(20) Mukherjee, P.; Parkinson, D.; Liddle, A. R. A Nested Sampling Algorithm for Cosmological Model Selection. 2006, arXiv:astro-ph/0508461v2. arXiv.org ePrint archive. http://arxiv.org/abs/astro-ph/0508461 (accessed Feb 10, 2007).
(21) Rose, K.; Hall, L. H.; Kier, L. B. Modeling Blood−Brain Barrier Partitioning Using the Electrotopological State. *J. Chem. Inf. Comput. Sci.* **2002**, *42*, 651−666.
(22) *StARLITe*, version 0411; BioFocus DPI: London, U. K., 2004 (accessed Jan 15, 2005). This is a knowledge base containing medicinal chemistry and pharmacological data from peer-reviewed journals including Journal of Medicinal Chemistry (1980−2004) and Bioorganic and Medicinal Chemistry Letters (1991−2004). StARLITe is a trademark of BioFocus DPI.
(23) Roche, O.; Trube, G.; Zuegge, J.; Pflimlin, P.; Alanine, A.; Schneider, G. A Virtual Screening Method for Prediction of the HERG Potassium Channel Liability of Compound Libraries. *ChemBioChem* **2002**, *3*, 455−459.
(24) Butina, D. Unsupervised Data Base Clustering Based on Daylight's Fingerprint and Tanimoto Similarity: A Fast and Automated Way to Cluster Small and Large Data Set. *J. Chem. Inf. Comput. Sci.* **1999**, *39*, 747−750.
(25) Gedeck, P.; Rohde, B.; Bartels, C. QSAR—How Good Is It in Practice? Comparison of Descriptor Sets on an Unbiased Cross Section of Corporate Data Sets. *J. Chem. Inf. Model.* **2006**, *46*, 1924−1936.
(26) Golbraikh, A.; Tropsha, A. Beware of q^2! *J. Mol. Graphics Modell.* **2002**, *20*, 269−276.
(27) Tetko, I. V.; Gasteiger, J.; Todeschini, R.; Mauri, A.; Livingstone, D.; Ertl, P.; Palyulin, V. A.; Radchenko, E. V.; Zefirov, N. S.; Makarenko, A. S.; Tanchuk, V. Y.; Prokopenko, V. V. Virtual Computational Chemistry Laboratory - Design and Description. *J. Comput.-Aided Mol. Des.* **2005**, *19*, 453−463.
(28) *E-DRAGON*, version 1.0; VCCLAB, Virtual Computational Chemistry Laboratory: Neuherberg, Germany, 2005. http://www.vcclab.org, 2005 (accessed Dec 6, 2006).
(29) Wold, S.; Sjöström, M.; Eriksson, L. Partial Least Squares Projections to Latent Structures (PLS) in Chemistry. In *The Encyclopedia of Computational Chemistry*; Schleyer, P. v. R., Allinger, N. L., Clark, T., Gasteiger, J., Kollman, P., Schaefer, H. F., III, Schreiner, P. R., Eds.; Wiley: Chichester, U. K., 1998; Vol. 3, pp. 2006−2022.
(30) Song, M.; Clark, M. Development and Evaluation of an in Silico Model for hERG Binding. *J. Chem. Inf. Model.* **2006**, *46*, 392−400.

CI7000633