StructRank: A New Approach for Ligand-Based Virtual Screening

Fabian Rathke,*[,†,‡] Katja Hansen,[†] Ulf Brefeld,[†,§] and Klaus-Robert Müller[†]

Department of Machine Learning, University of Technology, Berlin, Germany, Department of Image and Pattern Analysis, University of Heidelberg, Germany, and Yahoo! Research, Avinguda Diagonal 177, 08018 Barcelona, Spain

Received August 10, 2010

Screening large libraries of chemical compounds against a biological target, typically a receptor or an enzyme, is a crucial step in the process of drug discovery. Virtual screening (VS) can be seen as a ranking problem which prefers as many actives as possible at the top of the ranking. As a standard, current Quantitative Structure−Activity Relationship (QSAR) models apply regression methods to predict the level of activity for each molecule and then sort them to establish the ranking. In this paper, we propose a top-*k* ranking algorithm (StructRank) based on Support Vector Machines to solve the early recognition problem *directly*. Empirically, we show that our ranking approach outperforms not only regression methods but another ranking approach recently proposed for QSAR ranking, RankSVM, in terms of actives found.

INTRODUCTION

High-throughput screening, the physical screening of large libraries of chemicals, is the dominant technique for the identification of lead compounds in drug discovery.[1] In recent years, computational methods, known as Virtual Screening (VS),[2] have gained much attention as an alternative and complementary approach since they can be performed comparatively cheap and fast;[3] the usefulness of *in silico* screenings has been demonstrated in several studies.[4,5]

Virtual screening can be divided into structured-based and ligand-based[6] approaches. Given the drug targets' 3D structure and the 3D structures of ligands, structure-based VS predicts and scores the confirmation and orientation of the ligands within the active site of the receptor.[7] Ligand-based VS on the other hand uses knowledge about a set of ligands that are known to be active for the given drug target. This information is used to identify structurally similar molecules in a database.[7] Different approaches are available depending on the number of known actives; however, all approaches share the common assumption that, with respect to the descriptors, structurally similar molecules are likely to have similar properties.[8] In other words, neighboring molecules are likely to exhibit the same levels of activity.

Given a sufficient number of known actives, one can build a Quantitative Structure Activity Relation (QSAR) model. QSAR models correlate numerical molecular descriptors[9] as physiochemical and topological properties with a biological property such as binding affinity. For each molecule, the former is usually assembled in a *vector of features:* $\mathbf{x} \in \mathbb{R}^d$, while the latter is summarized as label $y \in \mathbb{R}$. Describing molecules and their properties by pairs $(\mathbf{x},y)$ paves the way for machine learned QSAR models. Prominent techniques include *Multiple Linear Regression* (MLR)[10] and *Partial Least Squares* (PLS)[11] and more recently Support Vector Machines for Regression (SVRs), Random Forests, Neural Networks, and Gaussian Processes.[12−15] Various reviews[16−18] offer a detailed overview over these approaches and their application to ligand-based virtual screening.

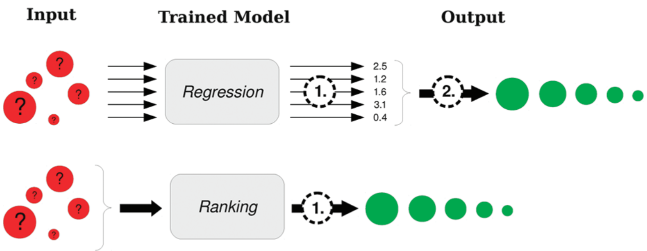


Figure 1. Two different ways to solve the ranking task of virtual screening: (a) State-of-the-art approaches use a two-step approach. In the first step, a regression model is used to predict binding coefficients for all molecules in the library. In a second step, the molecules are sorted according to their predictions. (b) Our ranking approach directly predicts the ranking within a single step.

The task in VS also known as the "early recognition problem"[19,20] can be characterized as follows: Given a library of molecules, the task is to output a ranking of these molecules in terms of their binding coefficient for the investigated drug target, such that the top-*k* molecules can be selected for further investigations. All of the above-mentioned methods solve this task by performing a regression analysis: They learn a function $f: x \rightarrow y$, $f: \mathbb{R}^d \rightarrow \mathbb{R}$ that predicts a label for any molecule given its features. To establish the subset of candidate molecules, predictions are made for all molecules in the database. In a second step, an ordered list is generated on the basis of these predictions. This two-step approach is shown in Figure 1 (top). Finally, the top-*k*-ranked compounds are selected to be investigated in more detail.

However, virtual screening approaches primarily aim to find molecules exhibiting high binding affinities with the target while the predictive accuracy with respect to the labels *y* is only of secondary interest. Although a perfect regression

* To whom correspondence should be addressed: E-mail: fabian.rathke@iwr.uni-heidelberg.de.
[†] University of Technology.
[‡] University of Heidelberg.
[§] Yahoo! Research.

10.1021/ci100308f
Published on Web 12/17/2010

model would also imply a perfect ranking of the molecules of interest, the impact of suboptimal regressors on the ranking is not easily captured, as equal models in terms of their mean squared error could give rise to completely different rankings. Thus, the question arises whether the detour via regression is necessary and whether the task can be addressed in a more natural way. In this article, we propose a top-k-ranking algorithm, **StructRank**, that *directly solves the ranking problem* and that *focuses on the most promising molecules* (cf. Figure 1, bottom).

The driving force for the research of new ranking approaches so far has been the information retrieval community.[21,22] Aiming to improve the results of search engines, documents need to be ranked within the first hits, according to their relevance for a given search query. In the virtual screening community, the use of ranking approaches has been rare. An approach that directly minimizes a ranking loss was applied recently by Wasserman et al.[23] and Agarwal et al.[24] RankSVM[25,26] maximizes the number of correctly ordered pairs of molecules for all ranks. Wassermann et al. report superior performance for RankSVM on classification data sets; Agarwal et al. state that RankSVM performs similarly to baselines tested for QSAR as well as classification data sets.

Whereas RankSVM attempts to optimize the complete ranking, StructRank focuses on the topmost ranks by optimizing the rank loss NDCG.[27] As previously stated by Agarwal et al.,[24] approaches that especially focus on this aspect should be able to outperform RankSVM. Our experiments can confirm this assumption: StructRank outperforms RankSVM as well as Support Vector Regression in terms of actives ranked within the top k. We report results for NDCG as well as two established virtual screening performance measures: Enrichment Factor (EF)[28] and Robust Initial Enhancement (RIE).[29]

The remainder of the article is structured as follows: The next section describes our top-k ranking approach, StructRank, and then briefly reviews the baseline methods RankSVM and SVR. We then introduce the virtual screening data sets in section 3 and the toy example that were used for performance evaluation. We report on empirical results in section 4 and conclude with a discussion in section 5.

METHODS

The formal problem setting of ranking for virtual screening is as follows: Given a set $\mathscr{T}$ consisting of n molecules $(\mathbf{x}_i,y_i)_{i=1}^n$, where $\mathbf{x}_i \in \mathbb{R}^d$ denotes the feature vector of the ith molecule containing the molecular descriptors, and $y_i \in \mathbb{R}$ is a scalar representing the biological/chemical property of that molecule, e.g., binding affinity. We aim at learning a function $f(x)$ which learns to rank the molecules according to their targets y_i. That is, if $y_i > y_j$ for molecules i and j, we want $f(\mathbf{x}_i) > f(\mathbf{x}_j)$. Moreover, as the purpose of virtual screening methods is to rank actives *early* in an ordered list (recall the "early recognition problem"[19,20]), we want the learning machine to focus on the top-k molecules in the ranking.

Our top-k ranking SVM for QSAR utilizes work by Chapelle et al.[30] They build on Structured Support Vector Machines (Structured SVMs),[31] a very flexible learning machine that has been applied to many different learning tasks in information retrieval,[30,32] natural language parsing,[33] and protein sequence alignment.[34]

In the following paragraphs, we describe Structured Support Vector Machines and adjust them to the task of ranking molecules. Additionally, we propose a new method to evaluate QSAR rankings: Normalized Discounted Cumulitive Gain (NDCG).

Evaluating Rankings. To assess the quality of rankings for QSAR, we propose to use a popular ranking measure that originates from the information retrieval community: Normalized Discounted Cumulative Gain (NDCG, see the Appendix for precise definition). Originally, NDCG[27] was introduced to evaluate the results of web searches. It measures how similar a predicted ranking is compared to the true ranking. NDCG has several important properties:

- NDCG$_k$ only evaluates the first k positions of predicted rankings; thus an error on positions below rank k is not punished.
- Furthermore, the first k positions are weighted, which means that errors have a different influence on the final score depending on which position of the ranking they occur. Naturally, position one is the most important, with lower positions discounted by the log of their rank r: $\log_2(1 + r)$.
- Finally, NDCG is normalized; thus if the predicted ranking equals the true ranking, the score is 1. Thus, to translate it into a loss function, we could simply use $\Delta(\mathbf{y},\hat{\mathbf{y}}) = 1 - \text{NDCG}(\mathbf{y},\hat{\mathbf{y}})$.

In summary, NDCG aims at pushing the molecules with the biggest binding affinity to the top of the ranking.

Structured Support Vector Machines for QSAR. We will now briefly describe the framework of Structured SVMs and focus on only the basic concept. For more detailed coverage, we refer to the paper of Tsochantaridis et al.[31]

Our ultimate target is to learn a function f: $\mathscr{X} \rightarrow \mathscr{Y}$: Given a set of molecules $\tilde{\mathbf{x}} = (\mathbf{x}_1, ..., \mathbf{x}_n) \in \mathscr{X}$, f returns a ranking $\mathbf{y} \in \mathscr{Y}$ of this set. In order to establish f, Structured SVMs learn a discriminant function F: $\mathscr{X} \times \mathscr{Y} \rightarrow \mathbb{R}$. F can be thought of as a *compatibility* function that measures how well a certain ranking $\mathbf{y}$ fits the given set of molecules $\tilde{\mathbf{x}}$. The final prediction is given by the ranking $\mathbf{y}$ that achieves the maximal score $F(\tilde{\mathbf{x}},\mathbf{y})$. Thus, we have

$$f(\tilde{\mathbf{x}}) = \underset{\mathbf{y} \in \mathscr{Y}}{\operatorname{argmax}}\, F(\tilde{\mathbf{x}}, \mathbf{y})$$

F is defined over a combined space of sets of molecules and corresponding rankings, a so-called "joint feature space". To be able to learn F directly in that combined space, we define a function Ψ that maps each pair of a set of molecules $\tilde{\mathbf{x}}$ together with a ranking $\mathbf{y}$ (of $\tilde{\mathbf{x}}$) onto one corresponding data point in the joint feature space. Details on the joint feature map used in our approach may be found in the Appendix. Given the joint feature map Ψ, F is defined as a linear function in the joint feature space:

$$F(\tilde{\mathbf{x}}, \mathbf{y}) = \mathbf{w}^T \Psi(\tilde{\mathbf{x}}, \mathbf{y})$$

This way, F is the scalar product of the corresponding joint feature map of $\tilde{\mathbf{x}}$ given a particular ranking $\mathbf{y}$ and the learned parameter vector $\mathbf{w}$.

Modeling F can be cast as follows: Given a set of molecules $\tilde{\mathbf{x}}$, we want the true ranking $\bar{\mathbf{y}}$ to score highest

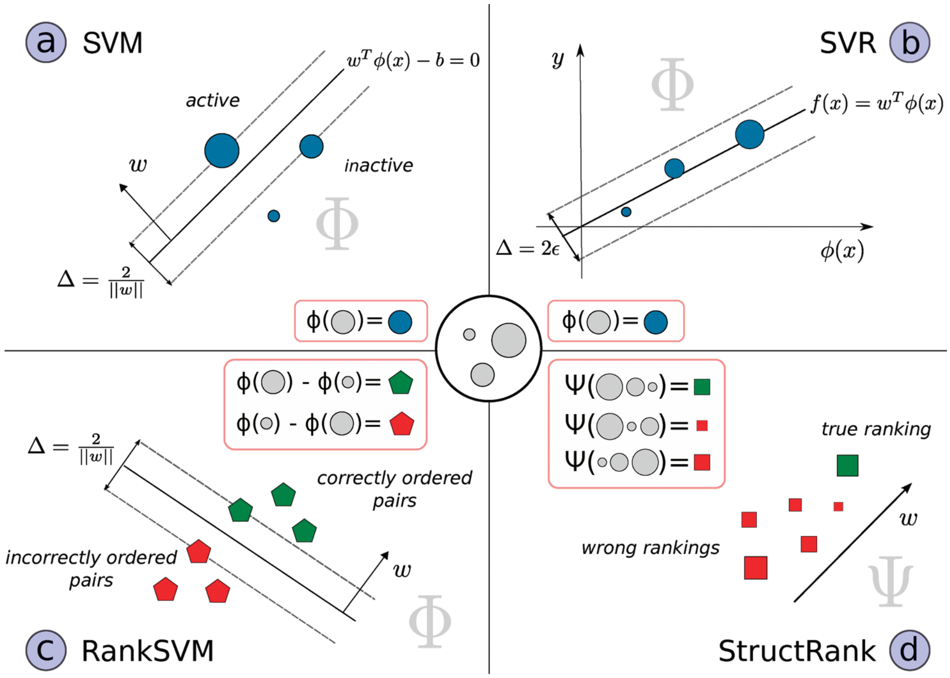


Figure 2. Comparison of different Support Vector Machines. (a) Support Vector Machines for classification learn a linear hyperplane $\mathbf{w}^T\phi(\mathbf{x}) = b$ with maximum margin Δ that optimally separates active from inactive molecules. (b) Support Vector Regression learns a function $\mathbf{w}^T\phi(x)$ that predicts binding affinities for each molecule as correctly as possible. (c) Ranking SVM generates difference vectors of all possible pairs of molecules. Afterward, similar to "a", a linear hyperplane is learned that separates correctly and incorrectly ordered pairs. (d) Ψ takes a set of molecules $\tilde{\mathbf{x}}$ and a ranking $\mathbf{y}$ of this set and maps it onto a point in the joint feature space. StructRank learns a function $\mathbf{w}^T\Psi(\tilde{\mathbf{x}},\mathbf{y})$, which assigns the highest score to the point representing the true ranking.

among all possible rankings $\mathbf{y} \in \mathcal{Y}$ transforming into constraints

$$\mathbf{w}^T(\Psi(\tilde{\mathbf{x}}, \bar{\mathbf{y}}) - \Psi(\tilde{\mathbf{x}}, \mathbf{y})) \geq 0 \quad \forall \mathbf{y} \in \mathcal{Y}\backslash\bar{\mathbf{y}}$$

As with classic Support Vector Machines for Classification,[35] this can be turned into a maximum-margin problem, where we want the difference between the true ranking $\bar{\mathbf{y}}$ and the closest runner-up $\text{argmax}_{\mathbf{y}\neq\bar{\mathbf{y}}}\mathbf{w}^T\Psi(\tilde{\mathbf{x}},\mathbf{y})$ to be maximal (see eq 11 in the Appendix). Also, we want different $\mathbf{y}$'s to be separated according to the degree of their falseness: A predicted ranking with only two ranks interchanged compared to the true ranking is much better than a predicted ranking with all ranks interchanged. We thus require the latter to get further separated with a larger margin from the true ranking than the first one. This is accomplished by replacing the constant margin formulation with the loss-dependent margin (*margin scaling*[31,36]):

$$\mathbf{w}^T(\Psi(\tilde{\mathbf{x}}, \bar{\mathbf{y}}) - \Psi(\tilde{\mathbf{x}}, \mathbf{y})) \geq \Delta(\mathbf{y}, \bar{\mathbf{y}}) \quad \forall \mathbf{y} \in \mathcal{Y}\backslash\bar{\mathbf{y}} \qquad (1)$$

where $1 - \text{NDCG}_k$ is used for $\Delta(\mathbf{y},\bar{\mathbf{y}})$. Furthermore, a *slack variable* ξ is introduced that reflects the maximal error made for the set of constraints in eq 1. Finally, to improve performance, we employ a boosting approach: we randomly draw m different subsets $\tilde{\mathbf{x}}^j$ of molecules from the training set. Applying the methodology described so far to each subset j, we obtain the final optimization problem:

$$\begin{aligned} \min_{w,\xi} \quad & \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{j=1}^{m}\xi^j \\ \text{subject to} \quad & \mathbf{w}^T(\Psi(\tilde{\mathbf{x}}^j, \bar{\mathbf{y}}^j) - \Psi(\tilde{\mathbf{x}}^j, \mathbf{y})) \geq \Delta(\bar{\mathbf{y}}^j, \mathbf{y}) - \xi^j \\ & \qquad \forall j, \forall \mathbf{y} \neq \bar{\mathbf{y}}^j \\ & \xi^j \geq 0 \end{aligned} \qquad (2)$$

Note that there is a very high formal similarity to the original SVM formalization (see eq 8 in the Appendix) with the following differences: (a) margin rescaling, (b) joint feature map, and (c) a very large quantity of constraints. A visualization of the function learned is given in Figure 2d). The corresponding dual form of eq 2 is given in the Appendix (eq 13).

For a set $\tilde{\mathbf{x}}$ with n molecules, there exist $n!$ possible ways of ranking these molecules. Imposing a constraint for each possible ranking would lead to problems becoming too big to be solved. Therefore, Tsochantaridis et al.[31] proposed a cutting plane approach that iteratively adds new constraints which violate the current solution. They show that there exists a polynomially sized subset of constraints whose solution fulfills all constraints of the full optimization problem. Astonishingly, the optimization problem can be solved efficiently; an example is the cutting-plane approach.

Baselines. We compare the novel ranking approach to two algorithms both belonging to the family of Support Vector Machines: Support Vector Regression (SVR), a state-of-the-art regression method, often used for Virtual Screening, and Ranking SVM (RankSVM), another ranking approach.

Support Vector Regression (SVR). Support Vector Regression[37] is an adaptation of classic Support Vector Classifiers for regression. Like its classification counterpart, it follows the Structural Risk Minimization principle introduced by Vapnik,[35] finding a trade-off between model complexity and training error. SVRs learn a linear function f in some chosen kernel feature space.[38] The final predictor is given by

$$f(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i k(\mathbf{x}_i, \mathbf{x}) + b \tag{3}$$

The α's weight the influence of training points $\mathbf{x}_i$ on the prediction $f(\mathbf{x})$. An ε-sensitive loss function is minimized, penalizing only predictions $\hat{y} = f(\mathbf{x})$ that differ more than ε from the true label y.

$$\ell(y, \hat{y}) = |(y - \hat{y})|_\varepsilon = \begin{cases} |(y - \hat{y})| & \text{for } |(y - \hat{y})| > \varepsilon \\ 0 & \text{else} \end{cases} \tag{4}$$

See Figure 2b for a visualization of SVR. Different studies[13,39−41] showed that SVRs can outperform Multiple Linear Regression and Partial Least Squares and perform on par with Neuronal Networks. For implementation, we used LIBSVM together with a Matlab interface available from http://www.csie.ntu.edu.tw/cjlin/libsvm/ (currently in version 3.0, 03.11.2010).

Ranking SVM. As a second baseline, we tested a second ranking approach: Ranking SVM.[25,26] Falling into the category of *pairwise* ranking approaches, it maximizes the performance measure *Kendall's τ*. It measures the number of correctly ordered pairs within a ranking of length n, taking into account all possible $(n(n - 1))/2$ pairs. *Kendall's τ* has two crucial differences compared to NDCG: All positions of the ranking have an influence on the final performance unlike for NDCG, where only the top k positions matter. Additionally, all positions have the same weight, unlike for NDCG, where higher positions are more important. The principle of Ranking SVM is visualized in Figure 2c. We used the implementation of Chapelle (available from http://olivier.chapelle.cc/primal/ranksvm.m, accessed 3/11/2010), which we extended for the use of kernels, according to ref 42.

DATA

We use virtual screening data sets from the Supporting Information of the paper of Sutherland et al.[43] where spline-fitting together with a genetic algorithm was tested to establish a good classifier on five data sets. We selected a subset of three data sets most suitable for regression: the benzodiazepine receptor (BZR) and the enzymes cyclooxygenase-2 (COX-2) and dihydrofolate reductase (DHFR). All data sets were assembled from the literature in order to mimic realistic HTS, i.e., possess high diversity and a low number of actives. Additionally, almost all molecules can be considered drug-like, satisfying Lipinski's rule of five.[44] We will now briefly describe the biological function of each target and give some information about the corresponding data set.

BZR. Being an ion channel located in the membrane of various neurons, BZR inhibits the neuron when bound by its endogenous ligand GABA. Drugs like benzodiazepine can have their own allosteric binding site. They increase the frequency of channel opening, thereby amplifying the inhibitory effect of GABA.[45]

The data set contains 405 molecules that were derived mostly from the work of two research groups (Haefely et al. and Cook et al.). We removed 73 compounds with inexact measurements ($\mathrm{pIC}_{50} < \mathit{value}$ instead of $\mathrm{pIC}_{50} = \mathit{value}$) which are not suitable for regression approaches. The remaining **340 molecules** had labels ranging from 4.27 to 9.47 pIC_{50}.

COX-2. The enzyme COX together with its isoform COX-1[46] takes part in the synthesis of prostanoids. While COX-2 is an adaptive enzyme which is only produced in response to injury or inflammation, COX-1 is a constitutive enzyme which is produced constantly and provides for a physiological level of prostaglandins.[47] Drugs that inhibit COX-2 were shown to reduce gastrointestinal side effects but at the price of increased cardiovascular risk.[48]

The data set consists of 467 COX-2 inhibitors. They were assembled on the basis of the published work of a single research group (Khanna et al.). We again deleted 53 molecules with inexact measurements. The remaining **414 molecules** had labels ranging from 4 to 9 pIC_{50}.

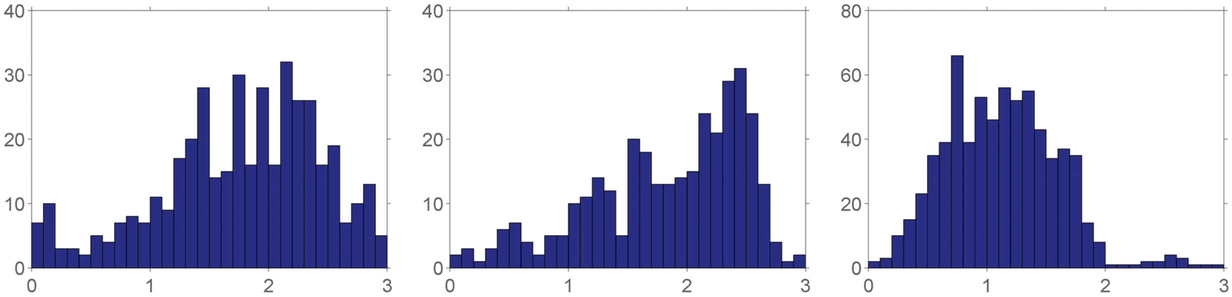
DHFR. The enzyme DHFR is involved in the syntheses of purins (adenine and guanine), pyrimidins (thymine), and some amino acids like glycine. As rapidly dividing cells like cancer cells need high amounts of thymine for DNA synthesis, they are particularly vulnerable to the inhibition of DHFR. Methotrexat, for example, is a DHFR-inhibitor which is used in the treatment of childhood leukemia and breast cancer, among others.[49]

The data set contains a set of 756 inhibitors of dihydrofolate reductase assembled on the basis of the work of one research group (Queener et al.), and we removed 74 compounds with inexact measurement. The remaining **682 molecules** had labels ranging from 3.03 to 10.45 pIC_{50}.

Descriptor Generation and Data Preparation. For descriptor generation, we used Dragon,[50] version 5.5. As in previous studies,[51,52] we used the following subset of Dragon blocks: 1, 2, 6, 9, 12, 15, 16, 17, 18, and 20. This yielded 728−772 descriptors, depending on the data set. We then *normalized* the feature vectors to zero mean and unit variance on the training set. In order to keep the results between data sets in terms of NDCG comparable, we *scaled* binding coefficients for each data set into the range [0,3], as this is a typical range when NDCG is used as scoring function for information retrieval data sets.[27]

If we examine the distribution of binding coefficients for each data set (see Figure 3), we can distinguish different types of distributions: For COX-2 we see a high number of molecules with high binding coefficients; thus we call this data set "dense". DHFR on the other hand has only a low number of molecules with high binding coefficients; thus we call this data set "sparse". BZR is in between, with few molecules possessing very high binding coefficients. We will make use of this distinction later in the Results section.

Test Framework. We used k-fold cross-validation to access performance for the virtual screening data sets. In order to have constant training set sizes (about 225 molecules), we varied the number of folds for each data set: we split BZR into three and COX-2 into two folds. Each fold was used as test set, whereas the other two folds (one fold) were used for training and parameter optimization. This was done by an inner cross-validation with five folds. For DHFR, we also employed three folds but used the single folds for training and the other two as a test set, thus also getting about



(a) COX-2 ("dense") (b) BZR ("medium") (c) DHFR ("sparse")

Figure 3. The distribution of binding coefficients for the virtual screening data sets. The *x* axis shows the binding coefficients (scaled into the range [0,3] for each data set). The *y* axis shows the number of molecules having that certain binding coefficient. Depending on the number of molecules with very high binding coefficients, we can refer to them as "dense" (COX-2), "medium" (BZR), and "sparse" (DHFR).

225 molecules in the training set. These cross-validations were performed seven times for DHFR and BZR, and 10 times for COX-2.

As all three approaches share the same underlying SVM framework, they need to determine the same parameters within the cross-validation loop; for the RBF-kernel:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{(\mathbf{x}_i - \mathbf{x}_j)^T(\mathbf{x}_i - \mathbf{x}_j)}{2d\sigma^2}\right) \qquad (5)$$

the parameters are $\sigma^2 \in \{0.1, 1, 10\}$ and d given by the number of descriptors. The SVM parameter C controlling the model complexity was chosen from the set {0.01, 0.1, 1, 10, 100}. For the SVR, we varied the tube width between {0.01, 0.1, 1}. For our StructRank approach, we also selected the number of ranks over which we optimized using 10, 20, and 30 as parameters.

Alternative Performance Measures. We add two performance measures well-known in the virtual screening community: Enrichment Factor (EF)[28] and Robust Initial Enhancement (RIE).[29] As shown by Truchon and Bayly,[19] the area under the ROC curve is not suitable for the "early recognition problem" of virtual screening.

RIE and ER only distinguish between active and inactive molecules, contrary to NDCG, which takes precise binding affinities into account. Therefore, we have to impose thresholds in order to separate molecules into actives and inactives. To provide for challenging ranking problems (i.e., a low ratio of actives/inactives), we chose 8.5 pIC_{50} (BZR), 8.0 pIC_{50} (COX-2), and 7.5 pIC_{50} (DHFR), respectively According to these thresholds the data sets contain 60, 70, and 38 actives (BZR, COX-2 and DHFR).

The *Enrichment Factor* measures how many more actives are found in an defined fraction ζ of the ordered list, relative to a random distribution. Thus, like NDCG, it only looks at the top k positions of the ranking, but weights each position equally. It is given by

$$\mathrm{EF} = \frac{\sum_{i=1}^{n} \delta_i}{\zeta \cdot n} \qquad (6)$$

where n is the number of actives. δ_i is 1 if the active is ranked within the defined fraction of the list; otherwise it is 0. *Robust Initial Enhancement* measures how much better a given ranking of actives is compared to their random distribution within the ranking. It considers the complete ranking, but like NDCG weights positions in descending order (depending on the parameter α, see eq 7). It is given by

$$\mathrm{RIE} = \frac{\sum_{i=1}^{n} \mathrm{e}^{-\alpha r_i}}{\langle \sum_{i=1}^{n} \mathrm{e}^{-\alpha r_i} \rangle_r} \qquad (7)$$

where r_i is the relative rank (i.e., the rank divided by the length of the ranking) and $1/\alpha$ is the fraction of the list that is most important for the final score, which has a similar meaning as the cutoff k of NDCG. The denominator is the mean score when the actives are distributed randomly across the ordered list.

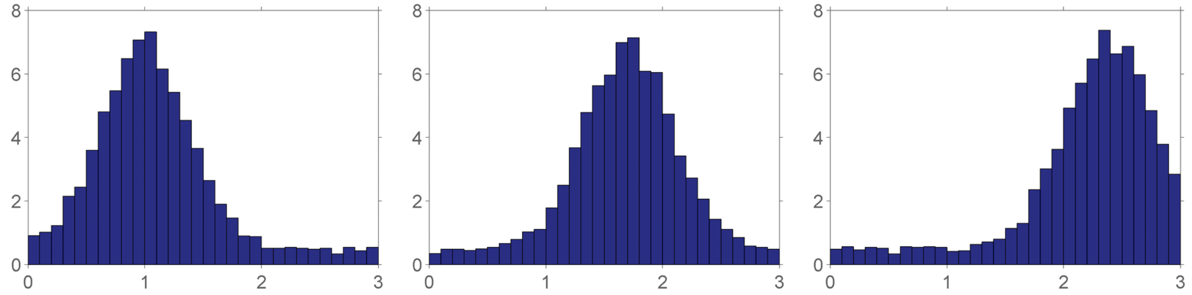
Toy Example. Before analyzing real-world VS data, we designed a toy example to reproduce a set of different label distributions typically found in virtual screening data sets: data sets which possess only a low number of molecules with high binding affinities, and those which contain a medium or high number of those molecules. Therefore, we applied the following approach: We selected 300 training sets (100 of each type) with distribution of labels as outlined above. Each training set consisted of 75 examples. Figure 4 shows the histograms, each averaged over all 100 sets.

The aim is to compare the influences of the different label distributions on ranking performance. We thus draw validation and test sets with uniform label distributions for all three types of training sets: We train models for different parameter combinations and select the optimal parameter combination on a validation set. Using the resulting model, ranking performance was measured using samples from a left out test set. The function we used to generate these data sets was $f(\mathbf{x}) = x_1^4 - x_2^3 - x_3^2 - x_4^4$, randomly drawn from the space of four-dimensional polynomials. We sampled 100.000 times from the four-dimensional unit cube $x \in \{[-1,1]^4\}$. Labels again were scaled into the range [0,3], and the feature vector $\mathbf{x}$ was normalized.

The algorithm, the DHFR data set (including Dragon descriptors) as well as the toy example data may be found at http://doc.ml.tu-berlin.de/structrank/.

RESULTS

We will now report on results obtained for the three virtual screening data sets, published by Sutherland et al.[43] Perfor-



(a) "sparse" (b) "medium" (c) "dense"

Figure 4. The histograms show the average label distribution for all three types of training sets (cf. text). The *y* axis shows the number of elements having a label given by the *x* axis.

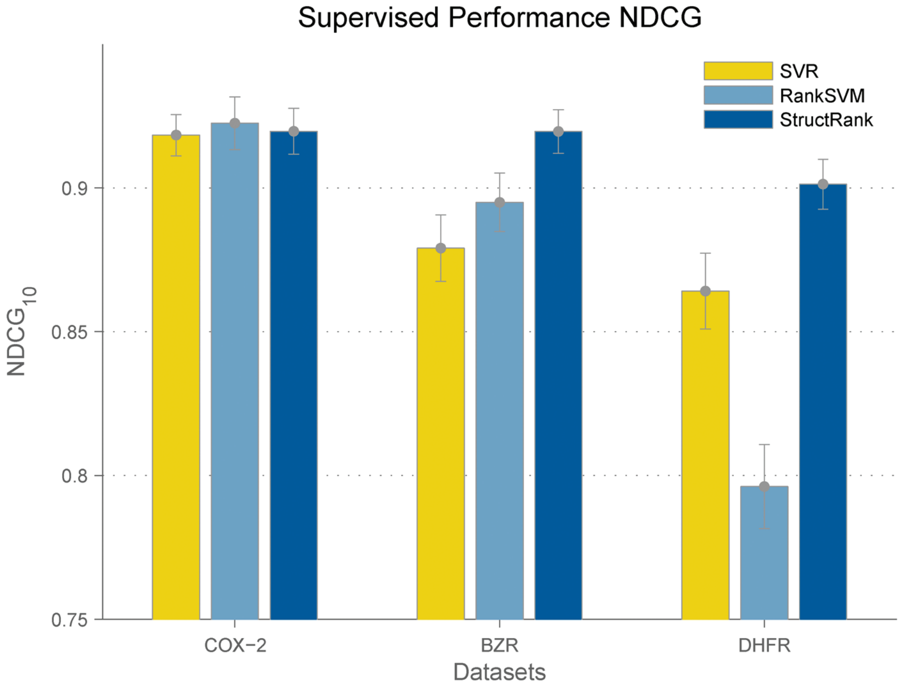


Figure 5. Averaged ranking performance measured in NDCG for the virtual screening data sets. Error bars indicate standard error.

mance is measured for Support Vector Regression (SVR), Ranking SVM, and our proposed StructRank approach. Furthermore, a toy example will shed some light on the results obtained for the virtual screening data sets.

Virtual Screening Data Sets. We measure ranking performance in terms of NDCG, ER, and RIE for both our baselines and our ranking approach StructRank. Performance is measured for the first 10 ranks, which means cutoffs of 10 for $NDCG_{10}$ and ER_{10}, as well as a parameter α for RIE, which puts the most weight on the top 10 ranks. We performed *k*-fold cross-validation as described before, where all three approaches were optimized for NDCG. Figure 5 shows the results in terms of NDCG. Error bars indicate standard error. Table 1 includes results for NDCG as well as ER and RIE. Significant improvements (level of significance 0.05) are indicated by bold numbers over approaches given as a superscript. Additionally, our approach is highlighted in gray. For all three performance measures, higher numbers indicate better rankings.

Starting with the dense data set COX, we observe that all three approaches perform nearly equally well in terms of NDCG, with no approach gaining a significant advantage over the others. These results are confirmed by the two "virtual screening" performance measures ER and RIE. For BZR, which could be classified as "medium" in terms of the high labeled molecules, our approach performs better than both baseline algorithms in terms of NDCG, improving significantly over SVR. RankSVM also can outperform SVR.

Table 1. Results for the Virtual Screening Data Sets for All Baselines and Our Structural Ranking Approach (Highlighted in Gray)[a]

	Method	COX-2	BZR	DHFR
$NDCG_{10}$	SVR	0.920	0.877	**0.872**^{2}
	RankSVM	0.928	**0.901**^{1}	0.798
	StructRank	0.921	**0.919**^{1}	**0.905**^{1,2}
ER_{10}	SVR	5.452	3.955	**16.061**^{2}
	RankSVM	5.583	**4.310**^{1}	13.966
	StructRank	5.326	**4.527**^{1}	**17.168**^{1,2}
RIE	SVR	4.692	3.481	**11.939**^{1}
	RankSVM	4.736	3.575	11.010
	StructRank	4.595	3.698	**12.604**^{1,2}

[a] Bold numbers mark significant improvements with a *p* value ≤ 0.05 over approaches given as superscripts: 1 = SVR and 2 = RankSVM. For all performance measures, higher numbers indicate better results.

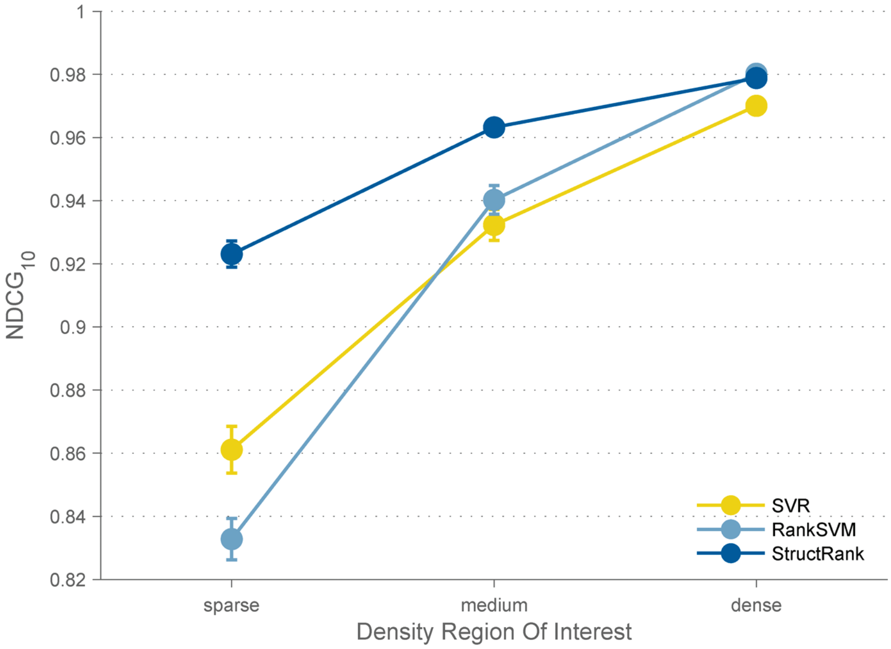


Figure 6. Ranking performance of Support Vector Regression (SVR), Ranking SVM (RankSVM), and Structural Ranking (StructRank) for three different types of training sets. The region with high labeled examples was covered either sparsely, intermediately, or densely. Error bars indicate standard error.

These results are confirmed by ER but not by RIE. Finally, for the "sparse" data set DHFR, our approach can significantly outperform both baseline methods in terms of ranking performance. This results hold for NDCG as well as ER and RIE. RankSVM is outperformed with a p value below 0.001. Furthermore, SVR can outperform RankSVM in terms of both virtual screening ranking measures.

Subsuming our observations, we state that our ranking approaches can outperform both baselines for the BZR and the DHFR set, while for the "dense" data set COX, all approaches perform equally. This data set contains many molecules with high labels; thus the event that one of these molecules is ranked high by chance is very likely. For BZR, we see (Figure 3) that the topmost bins, representing molecules with the highest labels, are sparsely populated. But subsequent bins, representing molecules with slightly lower labels, show a dense population like for COX. But these "sparse" bins seem to make it harder to obtain the perfect ranking, as performance drops in terms of NDCG for SVR and RankSVM. For the "sparse" data set DHFR, we can observe another decline in terms of ranking performance. Containing only very few molecules with high labels, this data set seems to be the hardest but also the most realistic VS scenario. Thus, we observed a continuous decline of performance of the baseline methods with a decreasing number of high labeled molecules.

Toy Example. Three different label distributions are generated as described in the Data section. Performance is again measured for SVR, RankSVM, and StructRank. The results, which are shown in Figure 6, reveal nearly the same behavior as for the real world virtual screening data sets. The "dense"-type data set has a big number of data points with large labels and is therefore comparable to COX-2. Like for COX-2, all approaches perform nearly the same. The "medium"-type data set has fewer data points with large labels and is comparable to BZR. Performance drops for both baselines, whereas StructRank's performance stays nearly the same. Also, like for BZR, RankSVM performs slightly better than SVR.

Table 2. Average CPU Time for Training/Prediction for the Virtual Screening Data Sets

	SVR	RankSVM	StructRank
training	0.18 s	1.71 s	2.32 s
prediction	0.31 s	0.05 s	0.05 s

Finally, the "sparse"-type data set is comparable to DHFR, having the lowest number of data points with large labels. Being the most difficult data set, all approaches display a drop in ranking performance. Nevertheless, for StructRank, the drop is small compared to the baselines, which are both clearly outperformed. Interestingly, SVR and RankSVM display the same behavior as for the virtual screening data sets: while RankSVM has the lead over SVR for the "medium" data set, SVR has a lead over RankSVM for the "sparse" data set.

Run Time Comparison. This section gives an overview of the CPU time needed by each approach for training and prediction. Given values present average values for the virtual screening data sets, i.e., training a model with about 225 molecules, and obtain a prediction for the test set (see Table 2). SVR requires the least CPU time to train a model since it needs to solve only one optimization problem. RankSVM has to solve a much more complex optimization problem, which is reflected in the increased time needed. For StructRank, the optimization problems become too big to be solved within one step. Thus, an iterative cut-and-bound technique[31] is applied, where for each iteration a convex quadratic subproblem has to be solved. This repeated convex optimization step is the reason for the increase of CPU time by a factor of 25 compared to the SVR. For prediction time, we have inverse results with the ranking approach performing fastest.

We also investigated the dependency of time needed to train a model on the training set size. We used the largest

Table 3. CPU Time for Training a Model on DHFR for Different Training Set Sizes

training Set Size	100	200	300	400	500	600	672
CPU time	0.90 s	0.99 s	1.07 s	1.16 s	1.33 s	1.54 s	1.62 s

data set, DHFR, and increased the training set in steps of 100 molecules. It turned out (cf. Table 3) that CPU time scales linearly with the size of the training set. This leaves us very optimistic that StructRank can be applied to much larger virtual screening data sets with reasonable performance.

DISCUSSION AND OUTLOOK

This work investigated the use of ranking approaches when building QSAR models for ligand-based virtual screening. Two ranking approaches, optimizing NDCG (StructRank) and Kendall's τ (RankSVM), were compared to one state-of-the-art approach for virtual screening: Support Vector Regression. The performance was measured using NDCG as well as two established VS metrics: Enrichment Factor and Robust Initial Enhancement.

This was the first time a ranking approach similar to StructRank was used within the field of QSAR modeling. Regarding the mathematical concept, using a ranking approach like StructRank offers two advantages for virtual screening:

1. Direct Optimization of Rankings. StructRank directly optimizes a ranking measure, compared to the indirect optimization of regression approaches, which in the first place optimize a regression performance measure.

2. Focus on Highly Binding Compounds. Because of its composition, NDCG focuses on molecules with high binding coefficients, whereas regression approaches like SVR or ranking approaches like RankSVM pay equal attention to each molecule owing to the structure of their loss functions. Thus, necessary complexity for solving the problem may be wasted uniformly over the data instead of focusing the algorithms' complexity on high rank entries.

Furthermore, runtime seems to be no real obstacle, as it scales linearly with training set size. Thus even for much larger data sets we expect a competitive performance and future work could investigate this assumption.

The evaluation results demonstrate that for data sets which possess only a small or medium number of molecules with high binding coefficients (e.g., BZR and especially DHFR) our approach performs significantly better than the baselines. For data sets which show a high density for these molecules, ranking approaches deliver no real advantage (e.g., for COX-2). These findings are underlined by the toy example: whereas our ranking approach outperforms SVR and RankSVM clearly for the "sparse" type data set, the advantage is lost for the "dense" type data set. An interesting direction for future research would be the experimental confirmation of these findings measured on real data sets.

In conclusion, we note that Structural Ranking represents a promising new approach that is very natural for virtual screening. To facilitate the further use of ranking approaches for virtual screening, we published our source code together with documentation on our webpage: http://doc.ml.tu-berlin.de/structrank/. The algorithm, the DHFR data set (including Dragon descriptors) as well as the toy example data may be found there.

ACKNOWLEDGMENT

This work was supported in part by the DFG Grant MU 987/4-2 and the FP7-ICT Programme of the European Community, under the PASCAL2 Network of Excellence, ICT-216886. We would like to thank talete srl for supporting a public DRAGON test descriptor set.

CLASSIC SUPPORT VECTOR CLASSIFICATION

Originally, Support Vector Machines were formulated by Vapnik[35] to solve classification tasks: Given a set of data points, belonging to either class +1 or −1, how can one separate these classes and additionally maximize the margin around the hyperplane such that $y_i(\mathbf{w}^T\phi(\mathbf{x}_i) + b) \geq 1$ for all $\mathbf{x}_i$? The optimization problem is given by

$$\begin{aligned} \min_{\mathbf{w},b,\xi} \quad & \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{n}\xi_i \\ \text{subject to} \quad & y_i(\mathbf{w}^T\phi(\mathbf{x}_i) + b) \geq 1 - \xi_i \\ & \xi_i \geq 0, i = 1, ..., n \end{aligned} \tag{8}$$

ξ_i are called slack variables and are nonzero for points that violate $y_i(\mathbf{w}^T\phi(\mathbf{x}_i) + b) \geq 1$, i.e., for those that are either misclassified or within the margin ±1 around the hyperplane $w^Tx - b = 0$.

NDCG

Given the true ranking $\bar{\mathbf{y}}$, a predicted ranking $\hat{\mathbf{y}}$, and a cutoff k, NDCG is given by the DCG (Discounted Cumulative Gain) for the predicted ranking normalized by the DCG of the true ranking.

$$\mathrm{NDCG}_k(\bar{\mathbf{y}}, \hat{\mathbf{y}}) = \frac{\mathrm{DCG}_k(\hat{\mathbf{y}})}{\mathrm{DCG}_k(\bar{\mathbf{y}})} \quad \mathrm{DCG}_k(\mathbf{y}) = \sum_{r=1}^{k}\frac{2^{\mathbf{y}(r)} - 1}{\log_2(1 + r)} \tag{9}$$

where $\hat{y}(r)$ is the binding coefficient y_i of the molecule $\mathbf{x}_i$ ranked at position r.

JOINT FEATURE SPACE

For a set of molecules $\tilde{\mathbf{x}} = (\mathbf{x}_1, ..., \mathbf{x}_n)$ and a ranking $\mathbf{y}$ of this set, the joint feature map Ψ is given by

$$\Psi(\tilde{\mathbf{x}}, \mathbf{y}) = \sum_{i=1}^{n}\phi(\tilde{\mathbf{x}}_i)\,A(\mathbf{y}_i) \tag{10}$$

as proposed by Chapelle.[30] ϕ is a mapping into Hilbert space corresponding to a kernel function $k(\mathbf{x}_i,\mathbf{x}_j)$, e.g., the RBF kernel. The new vector in Ψ is a sum of vectors $\phi(\tilde{\mathbf{x}}_i)$ weighted by their ranks according to $A(r) = \max(0,k + 1 - r)$. Only molecules corresponding to the first k ranks are incorporated.

STRUCTURED SUPPORT VECTOR MACHINES

Here, we present a more technical description of Structured Support Vector Machines supplementing the description given in the previous paper. The "naive" maximum-margin problem is given by

$$\begin{aligned} \min_{w,\xi} \quad & \frac{1}{2}\mathbf{w}^T\mathbf{w} \\ \text{subject to} \quad & \mathbf{w}^T(\Psi(\tilde{\mathbf{x}}^j, \bar{\mathbf{y}}^j) - \Psi(\tilde{\mathbf{x}}^j, \mathbf{y})) \geq 1 \ \ \forall j, \forall \mathbf{y} \neq \bar{\mathbf{y}}^j \end{aligned} \tag{11}$$

Keep in mind that each $\tilde{\mathbf{x}}^j$ consists of a set of k molecules $\mathbf{x}_i$, and the corresponding $\mathbf{y}^j$ holds the corresponding true ranking of all molecules within the set. All $k! - 1$ other possible rankings of $\tilde{\mathbf{x}}^j$ are represented by $\mathbf{y}$. After replacing the constant margin ~ 1 with a loss-dependent margin Δ and introducing slack variables ξ^j for each set $\tilde{\mathbf{x}}^j$, we get the final optimization problem

$$\begin{aligned} \min_{w,\xi} \quad & \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{j=1}^{m}\xi^j \\ \text{subject to} \quad & \mathbf{w}^T(\Psi(\tilde{\mathbf{x}}^j, \bar{\mathbf{y}}^j) - \Psi(\tilde{\mathbf{x}}^j, \mathbf{y})) \geq \Delta(\bar{\mathbf{y}}^j, \mathbf{y}) \ - \xi^j \\ & \qquad\qquad\qquad \forall j, \forall \mathbf{y} \neq \bar{\mathbf{y}}^j \\ & \xi^j \geq 0 \end{aligned} \tag{12}$$

The corresponding dual is given by

$$\max_{\alpha} \ -\frac{1}{2}\alpha^T L \alpha + \mathbf{b}^T\alpha \tag{13}$$

$$\text{subject to} \sum_{y \in \mathcal{Y}} \alpha^j_{\mathbf{y}} \leq C, \alpha^j_{\mathbf{y}} \geq 0 \ \ \forall j, \forall \mathbf{y} \neq \bar{\mathbf{y}}^j \tag{14}$$

where we have an α for each possible ranking $\mathbf{y}$ of subset $\tilde{\mathbf{x}}^j$. The matrix L consists of entries $(L)_{i\mathbf{y},j\mathbf{y}'} = (\Psi(\tilde{\mathbf{x}}^i,\bar{\mathbf{y}}^i) - \Psi(\tilde{\mathbf{x}}^i,\mathbf{y}))^T$-$(\Psi(\tilde{\mathbf{x}}^j,\bar{\mathbf{y}}^j) - \Psi(\tilde{\mathbf{x}}^j,\mathbf{y}'))$ and $b_{i\mathbf{y}} = \Delta(\bar{\mathbf{y}}^i,\mathbf{y})$.

ABBREVIATIONS

BZR = Benzodiazepine Receptor
COX-2 = Cyclooxygenase 2
DHFR = Dihydrofolate Reductase
NDCG = Normalized Discounted Cumulitive Gain
QSAR = Quantitive Structure−Activity Relationship
RankSVM = Ranking SVM
StructRank = Structural Ranking
SVM = Support Vector Machine
SVR = Support Vector Regression

REFERENCES AND NOTES

(1) Shoichet, B. K. Virtual screening of chemical libraries. *Nature* **2004**, *432*, 862–865.

(2) Walters, W. P.; Stahl, M. T.; Murcko, M. A. Virtual screening - an overview. *Drug Discovery Today* **1998**, *3*, 160–178.

(3) Waszkowycz, B.; Perkins, T. D. J.; Sykes, R. A.; Li, J. Large-scale virtual screening for discovering leads in the postgenomic era. *IBM Syst. J.* **2001**, *40*, 360–378.

(4) Perola, E.; Xu, K.; Kollmeyer, T. M.; Kaufmann, S. H.; Prendergast, F. G.; Pang, Y. P. Successful virtual screening of a chemical database for farnesyltransferase inhibitor leads. *J. Med. Chem.* **2000**, *43*, 401–408.

(5) Rupp, M.; Schroeter, T.; Steri, R.; Zettl, H.; Proschak, E.; Hansen, K.; Rau, O.; Schwarz, O.; Müller-Kuhrt, L.; Schubert-Zsilavecz, M.; Müller, K.-R.; Schneider, G. From machine learning to natural product derivatives that selectively activate transcription factor PPARγ. *ChemMedChem* **2010**, *5*, 191–194.

(6) Stahura, F. L.; Bajorath, J. New methodologies for ligand-based virtual screening. *Curr. Pharm. Des.* **2005**, *11*, 1189–1202.

(7) Lyne, P. D. Structure-based virtual screening: an overview. *Drug Discovery Today* **2002**, *7*, 1047–1055.

(8) Johnson, M. A.; Maggiora, G. M. *Concepts and Applications of Molecular Similarity*; John Wiley & Sons: New York, 1990.

(9) Todeschini, R.; Consonni, V. *Handbook of Molecular Descriptors*, 1st ed.; Wiley-VCH: New York, 2002.

(10) Hansch, C.; Fujita, T. p-σ-π Analysis. A Method for the Correlation of Biological Activity and Chemical Structure. *J. Am. Chem. Soc.* **1964**, *86*, 1616–1626.

(11) Dunn, W. J.; Wold, S.; Edlund, U.; Hellberg, S.; Gasteiger, J. Multivariate structure-activity relationships between data from a battery of biological tests and an ensemble of structure descriptors: The PLS method. *Quant. Struct.-Act. Relat.* **1984**, *3*, 131–137.

(12) Svetnik, V.; Liaw, A.; Tong, C.; Culberson, J. C.; Sheridan, R. P.; Feuston, B. P. Random forest: a classification and regression tool for compound classification and QSAR modeling. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 1947–1958.

(13) Demiriz, A.; Bennett, K. P.; Breneman, C. M.; Embrechts, M. J. Support Vector Machine Regression in Chemometrics. *Proceedings of the 33rd Symposium on the Interface of Computing Science and Statistics*, Costa Mesa, CA, 2001.

(14) 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.

(15) 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.

(16) Lengauer, T.; Lemmen, C.; Rarey, M.; Zimmermann, M. Novel technologies for virtual screening. *Drug Discovery Today* **2004**, *9*, 27–34.

(17) Melville, J. L.; Burke, E. K.; Hirst, J. D. Machine learning in virtual screening. *Comb. Chem. High Throughput Screen.* **2009**, *12*, 332–343.

(18) Dudek, A. Z.; Arodz, T.; Gálvez, J. Computational methods in developing quantitative structure-activity relationships (QSAR): a review. *Comb. Chem. High Throughput Screen.* **2006**, *9*, 213–228.

(19) Truchon, J.-F.; Bayly, C. I. Evaluating virtual screening methods: good and bad metrics for the "early recognition" problem. *J. Chem. Inf. Model.* **2007**, *47*, 488–508.

(20) Nicholls, A. What do we know and when do we know it. *J. Comput.-Aided Mol. Des.* **2008**, *22*, 239–255.

(21) Liu, T. Y.; Xu, J.; Qin, T.; Xiong, W.; Li, H. LETOR: Benchmark Dataset for Research on Learning to Rank for Information Retrieval. *SIGIR '07: Proceedings of the Learning to Rank workshop in the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, Amsterdam, The Netherlands, 2007.

(22) Le, Q. V.; Smola, A.; Chapelle, O.; Teo, C. H. Optimization of Ranking Measures. *J. Mach. Learn. Res.* **2010**, *1*, 1–48.

(23) Wassermann, A. M.; Geppert, H.; Bajorath, J. Searching for Target-Selective Compounds Using Different Combinations of Multiclass Support Vector Machine Ranking Methods, Kernel Functions, and Fingerprint Descriptors. *J. Chem. Inf. Model.* **2009**, *49*, 582–592.

(24) Agarwal, S.; Dugar, D.; Sengupta, S. Ranking Chemical Structures for Drug Discovery: A New Machine Learning Approach. *J. Chem. Inf. Model.* **2010**, *50*, 716–731.

(25) Herbrich, R.; Graepel, T.; Obermayer, K. Support Vector Learning for Ordinal Regression. *ICANN: Proceedings of the Ninth International Conference on Artificial Neural Networks*, Edinburgh, U.K., 1999.

(26) Joachims, T. Optimizing search engines using clickthrough data. *KDD: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, Alberta, Canada, 2002.

(27) Järvelin, K.; Kekäläinen, J. IR evaluation methods for retrieving highly relevant documents. *SIGIR: Proceedings of the Annual ACM Conference on Research and Development in Information Retrieval*, Athens, Greece, 2000.

(28) Pearlman, D. A.; Charifson, P. S. Improved scoring of ligand-protein interactions using OWFEG free energy grids. *J. Med. Chem.* **2001**, *44*, 502–511.

(29) Sheridan, R. P.; Singh, S. B.; Fluder, E. M.; Kearsley, S. K. Protocols for bridging the peptide to nonpeptide gap in topological similarity searches. *J. Chem. Inf. Comput. Sci.* **2001**, *41*, 1395–1406.

(30) Chapelle, O.; Le, Q.; Smola, A. Large margin optimization of ranking measures. *NIPS: Workshop on Machine Learning for Web Search*, Vancouver, BC, Canada, 2007.

(31) Tsochantaridis, I.; Joachims, T.; Hofmann, T.; Altun, Y. Large Margin Methods for Structured and Interdependent Output Variables. *J. Mach. Learn. Res.* **2005**, *6*, 1453–1484.

(32) Yue, Y.; Finley, T.; Radlinski, F.; Joachims, T. A support vector method for optimizing aver-age precision. *SIGIR: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, Amsterdam, The Netherlands, 2007.

(33) Brefeld, U.; Scheffer, T. Semi-supervised learning for structured output variables. *ICML '06: Proceedings of the 23rd international conference on Machine Learning*, Pittsburgh, Pennsylvania, 2006; pp 145−152.

(34) Yu, C.-N. J.; Joachims, T.; Elber, R.; Pillardy, J. Support Vector Training of Protein Align-ment Models. *J. Comput. Biol.* **2008**, *15*, 867–880.

(35) Vapnik, V. *The Nature of Statistical Learning Theory*; Springer: New York, 1995.

(36) Taskar, B.; Guestrin, C.; Koller, D. Max-margin Markov networks. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, 2003.

(37) Drucker, H.; Burges, C. J. C.; Kaufman, L.; Smola, A.; Vapnik, V. Support vector regression machines. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, 1997.

(38) Schölkopf, B.; Mika, S.; Burges, C. J. C.; Knirsch, P.; Müller, K.-R.; Rätsch, G.; Smola, A. J. Input space versus feature space in kernel-based methods. *IEEE T. Neural Networks* **1999**, *10*, 1000–1017.

(39) Liu, H. X.; Zhang, R. S.; Yao, X. J.; Liu, M. C.; Hu, Z. D.; Fan, B. T. QSAR Study of Ethyl 2-[(3-Methyl-2,5-dioxo(3-pyrrolinyl))amino]-4-(trifluoromethyl) pyrimidine5-carboxylate: An Inhibitor of AP-1 and NF-κB Mediated Gene Expression Based on Support Vector Machines. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 1288–1296.

(40) Yao, X. J.; Panaye, A.; Doucet, J. P.; Zhang, R. S.; Chen, H. F.; Liu, M. C.; Hu, Z. D.; Fan, B. T. Comparative study of QSAR/QSPR correlations using support vector machines, radial basis function neural networks, and multiple linear regression. *J. Chem. Inf. Comput. Sci.* **2004**, *44*, 1257–1266.

(41) Kriegl, J. M.; Arnhold, T.; Beck, B.; Fox, T. Prediction of Human Cytochrome P450 Inhibition Using Support Vector Machines. *QSAR Comb. Sci.* **2005**, *24*, 491–502.

(42) Chapelle, O. Training a support vector machine in the primal. *Neural Comput.* **2007**, *19*, 1155–1178.

(43) Sutherland, J. J.; O'Brien, L. A.; Weaver, D. F. Spline-fitting with a genetic algorithm: a method for developing classification structure-activity relationships. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 1906–1915.

(44) Lipinski, C. A.; Lombardo, F.; Dominy, B. W.; Feeney, P. J. Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. *Adv. Drug Delivery Rev.* **1997**, *23*, 3–25.

(45) Siegel, G. J.; Agranoff, B. W.; Albers, R. W.; Brady, S. T. *Basic Neurochemistry: Molecular, Cellular and Medical Aspects*; Lippincott, Williams and Wilkins: Philadelphia, PA, 1999.

(46) Xie, W. L.; Chipman, J. G.; Robertson, D. L.; Erikson, R. L.; Simmons, D. L. Expression of a mitogen-responsive gene encoding prostaglandin synthase is regulated by mRNA splicing. *Proc. Natl. Acad. Sci. U.S.A.* **1991**, *88*, 2692–2696.

(47) DeWitt, D. L.; Day, J. S.; Sonnenburg, W. K.; Smith, W. L. Concentrations of prostaglandin endoperoxide synthase and prostaglandin I2 synthase in the endothelium and smooth muscle of bovine aorta. *J. Clin. Invest.* **1983**, *72*, 1882–1888.

(48) Jeffrey, M.; Drazen, M. COX-2 Inhibitors - A Lesson in Unexpected Problems. *New Engl. J. Med.* **2005**, *352*, 1131–1132.

(49) Bertino, J. R. Karnofsky memorial lecture. Ode to methotrexate. *J. Clin. Oncol.* **1993**, *11*, 5–14.

(50) *DRAGON for Windows*, version 5.5; talete srl: Milano, Italy, 2007.

(51) Schroeter, T. S.; 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 models: a study on aqueous solubility of drug discovery molecules. *J. Comput.-Aided Mol. Des.* **2007**, *21*, 651–664.

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

CI100308F