Counting-based Output Prediction for Orphan Screening

Katrin Ullrich[1,2], **Christoph Stahr**[1,3], **Thomas Gärtner**[1,4]
[1]Fraunhofer IAIS, Schloss Birlinghoven, 53757 Sankt Augustin
[2]B-IT Research School, [3]University of Magdeburg, [4]University of Bonn
katrin.ullrich@iais.fraunhofer.de, christoph.stahr@student.uni-magdeburg.de,
thomas.gaertner@iais.fraunhofer.de

Abstract

We investigate orphan screening, the search for small molecule ligands of proteins for which no binding ligands are known in advance. Predicting interactions between biologically active molecules is an important step towards effective drug discovery. We propose novel classification and ranking algorithms for orphan screening which are based on counting feature combinations in molecular fingerprints. For the training process we only use positive examples and additional knowledge about the considered proteins and ligands. This knowledge is available in form of protein similarity values a database of molecule compounds. Our algorithms have runtime linear in the number of unlabelled examples.

1 Introduction

We aim at the prediction of ligands for orphan targets. Ligands denote small molecules binding proteins, whereas an orphan target is a protein or protein binding site for which no example ligands are available for learning. More generally, virtual screening denotes the in silico testing of huge databases of molecules for predefined properties, such as the activity against a target. Virtual screening is an important tool to support laboratory testing for the search of ligands. In particular, it is used to preselect promising molecules from the typically very large databases of synthesizable molecules in order to reduce the time and money needed to develop a novel drug. In recent years, machine learning techniques have been very successfully adopted for virtual screening. Whether ligands for the protein under consideration are available beforehand or not, distinguishes the cases of practical relevance. For traditional virtual screening some ligands of the protein are given and others are sought. For orphan screening no ligands of the protein are known and the task is to find some. In the latter case, the protein is called an orphan target.

Virtual screening can be modelled as a classification problem. In contrast to approaches where one is interested in the intensity of the protein ligand bond, in the classification setting we only want to find out whether a small molecule binds a protein or not. Approaches like the one described by Geppert et al. (2008) use support vector machines for non-orphan screening. They employ known ligands as positive training instances and a sample of database molecules as negative training instances. Building on these techniques, Jacob and Vert (2008) try to improve predictive accuracy with a hypothesis that targets several proteins at the same time. Additionally, they carry their virtual screening method over to orphan screening. In particular, they classify protein-ligand pairs as follows: a pair is considered as positive training instance if the ligand binds to the protein and as negative one, otherwise. Later, a somewhat different technique for both orphan and non-orphan screening was developed by Geppert et al. (2009). There, the authors realize their prediction via a combination of multiple hypotheses. While Geppert et al. (2009) show that their approach is more efficient, both approaches for orphan screening still suffer from two drawbacks: On the one hand, the database molecules, which in fact are molecules for which the label is not known, are assumed to be negative training instances. On the other hand, the amount of available database molecules is huge but the time complexity of these algorithms is cubic in the number of used database molecules.

In this paper we develop a novel approach to virtual screening that is (i) applicable to orphan screening, (ii) does not assume database molecules to be negative instances but models them as unlabelled instances, and (iii) has time complexity linear in the number of database molecules. Our approach is derived from the counting-based structured output prediction algorithm proposed in Gärtner and Vembu (2009). Their method works efficiently if sums over features and pairs of features can be computed in linear time. The term structured output refers to the fact that the results of structured output algorithms are more complex than just labels of natural or real numbers. In our case the structured output will be a fingerprint, a feature vector which represents a compound in a database of molecules. Our aim is to adapt the method of Gärtner and Vembu (2009) for the problem of ligand search, as counting features in fingerprints is computationally easy and may lead to good results in this setting. Actually, we achieve the linear runtime complexity by precomputing the sums of feature vectors and their inner products over the database in linear time.

2 Preliminaries

Small molecules can be represented as bit vectors of zeros and ones in $\mathbb{R}^d$ via molecular fingerprints. The developers of these fingerprints assembled a set of d features in order to display the molecules by means of their chemical properties. Fingerprints have already been established as

standard tools in ligand prediction tasks.

Kernel methods are a popular class of learning algorithms. They include techniques like the support vector machine that are well-founded in learning theory and have been applied in many real-world problems. They work by applying a symmetric and positive semidefinite function $k_{\mathcal{X}} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ as a similarity measure for elements of a space $\mathcal{X}$. A function with these properties is also called kernel function and is known to uniquely generate a reproducing kernel Hilbert space $\mathcal{H}_{\mathcal{X}}$ of real-valued functions defined on $\mathcal{X}$ (see Schölkopf et al. (2001)). In recent years, kernel methods have been extended to handle more complex – so-called structured – outputs. For this purpose we want to learn a joint scoring function f which is defined on the cross-product of two spaces $\mathcal{X}$ and $\mathcal{Y}$. The function f should assign a real score to pairs $(x, y) \in \mathcal{X} \times \mathcal{Y}$ according to "how well x and y fit". Applied to our situation, we want to find ligands for proteins from a set $\mathcal{X}$ by screening a huge set of small molecules stored in a database $\mathcal{Y}$. A common choice, which we will also adopt in this work, is to choose the tensor product of the input and output reproducing kernel Hilbert spaces $\mathcal{H} = \mathcal{H}_{\mathcal{X}} \otimes \mathcal{H}_{\mathcal{Y}}$ as hypothesis space for the joint scoring function. The function space $\mathcal{H}$ has the kernel $k = k_{\mathcal{X}} \otimes k_{\mathcal{Y}}$ given by

$$k\big((x, y), (x', y')\big) = k_{\mathcal{X}}(x, x') \cdot k_{\mathcal{Y}}(y, y'),$$

for $(x, y), (x', y') \in \mathcal{X} \times \mathcal{Y}$, where $k_{\mathcal{X}}$ and $k_{\mathcal{Y}}$ are the kernels of $\mathcal{H}_{\mathcal{X}}$ and $\mathcal{H}_{\mathcal{Y}}$, respectively. We will assume that the feature mapping $\phi : \mathcal{Y} \to \mathbb{R}^d$ is low dimensional and given explicitly. This is indeed naturally the case for virtual and orphan screening where we use fingerprint vectors of length d as molecule representations. In practice, the positive semidefinite kernel $k_{\mathcal{X}}$ is calculated essentially as a DNA sequence identity for two proteins and ranges from 0 to 100. Furthermore, we consider the inner product of feature vectors as the kernel $k_{\mathcal{Y}}$ in the sense of

$$k_{\mathcal{Y}}(y, y') = \langle \phi(y), \phi(y') \rangle,$$

for $y, y' \in \mathcal{Y}$.

3 The Method

In the sequel we propose new loss-functions adapted to the problem of ligand prediction with a database of unlabelled examples and few positive training instances. Concerning the solution of the respective optimization problem we will benefit from the properties of reproducing kernel Hilbert spaces. In fact, our minimizer will turn out to be a linear combination of kernel functions evaluated at the training examples. In addition to orphan screening, our algorithm can also be applied to virtual screening for non-orphan targets, i.e., the search for new ligands of proteins with a few known ligands. The main algorithmic difference between both tasks is the utilization of positive training examples. For orphan screening we exploit ligand information of other proteins and a similarity measure between those proteins (used for training) and the orphan target. Contrary to orphan screening, for virtual screening of non-orphan targets we may additionally use the information we already have for the target in consideration.

3.1 Counting-Based Structured Output Prediction

We are considering a set of proteins $\mathcal{X}$ and a database of small molecules $\mathcal{Y}$, where the latter contains the potential ligands. In general, we are looking for a joint scoring function $f : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, which assigns a real value to every pair $(x, y) \in \mathcal{X} \times \mathcal{Y}$. This scoring value should be the higher the better ligand y binds to protein x. The hypothesis space for f shall be the reproducing kernel Hilbert space $\mathcal{H} = \mathcal{H}_{\mathcal{X}} \otimes \mathcal{H}_{\mathcal{Y}}$ with product kernel $k = k_{\mathcal{X}} \otimes k_{\mathcal{Y}}$. Suppose we have positive training examples

$$(x_1, \mathcal{Y}_1), \ldots, (x_m, \mathcal{Y}_m) \in \mathcal{X} \times 2^{\mathcal{Y}},$$

i.e., a set $\mathcal{Y}_i$ of known ligands for every protein $x_i \in \{x_1, \ldots, x_m\} = \mathcal{X}_{train} \subseteq \mathcal{X}$. Moreover, we assume the availability of further knowledge about the candidate space $\mathcal{Y}$. For our algorithm this knowledge consists of a database containing the elements of $\mathcal{Y}$ in form of feature vectors. Note, that we do not consider the exponentially large set of all possible molecule representations. With unlabelled examples we denote every possible combination (x_i, z) such that $x_i \in \mathcal{X}_{train}$ and $z \in \mathcal{Z}_i$, with $\mathcal{Z}_i := \mathcal{Y} \setminus \mathcal{Y}_i$. These pairs are unlabelled examples as we do not know beforehand whether z binds to x_i or not. For orphan screening the considered orphan target $\tilde{x}$ is an element of $\mathcal{X} \setminus \mathcal{X}_{train}$, while a non-orphan target is contained in $\mathcal{X}_{train}$ by construction of the two different tasks. Finally, our goal is a sorted list of the compounds in $\mathcal{Y}$ with respect to a certain protein $\tilde{x}$. Therefore, we want to assign high values of the scoring function f to positive examples and small values to the probably negative unlabelled examples, respectively. We will present different loss-functions which fulfill this requirement (and depend on both positive and unlabelled examples) in the next section. In principle, we want to learn a function $f^* \in \mathcal{H}$ which minimizes the regularized risk functional

$$f^* = \operatorname*{argmin}_{f \in \mathcal{H}} \lambda \|f\|_{\mathcal{H}}^2 + \sum_{i=1}^{m} \mathcal{R}(f, i) \tag{1}$$

where $\mathcal{R}(\cdot, i)$ is the loss-function for the i-th positive training example. Since the quadratic function is strictly monotonically increasing and the regularized risk term $\sum_{i=1}^{m} \mathcal{R}(f, i)$ specified below is a function with inputs (x_i, y) and (x_i, z) such that $x_i \in \mathcal{X}_{train}$, $y \in \mathcal{Y}_i$, and $z \in \mathcal{Z}_i$, $i = 1, \ldots, m$, we may apply the representer theorem (Schölkopf et al., 2001). We obtain for our optimizer f^* that

$$\begin{aligned} f^* \in \mathcal{F} &= \operatorname{span}\Big\{ k_{\mathcal{X}}(\cdot, x) \cdot k_{\mathcal{Y}}(\cdot, z) : x \in \mathcal{X}, z \in \mathcal{Y} \Big\} \\ &= \operatorname{span}\Big\{ k_{\mathcal{X}}(\cdot, x) \cdot \langle \phi(\cdot), \phi(z) \rangle : x \in \mathcal{X}, z \in \mathcal{Y} \Big\}, \end{aligned}$$

where $k_{\mathcal{X}}(\cdot, \cdot)$ is the adequate protein similarity value, and $\langle \phi(\cdot), \phi(\cdot) \rangle$ is the inner product in the output space with dimension d. Usually, $\mathcal{Y}$ is given as a huge database of instances. For this reason we would like to scale down $\mathcal{F}$. In practice the dimension d of the feature space is often much smaller than the number of elements in $\mathcal{Y}$ (e.g. fingerprint *maccs* has $d \approx 160$, while $|\mathcal{Y}|$ is usually greater than 100.000). If we consider the canonical orthonormal

basis $e_1, \ldots, e_d$ of $\mathbb{R}^d$, the transformation

$$\begin{aligned}\mathcal{F} &= \text{span}\left\{k_{\mathcal{X}}(\cdot, x) \cdot \langle \phi(\cdot), \phi(z)\rangle : x \in \mathcal{X}_{train}, z \in \mathcal{Y}\right\} \\ &= \text{span}\Big\{k_{\mathcal{X}}(\cdot, x) \cdot \langle \phi(\cdot), \sum\nolimits_{l=1}^{d} \beta_l^z e_l\rangle : \\ &\qquad\qquad x \in \mathcal{X}_{train}, z \in \mathcal{Y}, \beta_l^z \in \mathbb{R}\Big\} \\ &= \text{span}\Big\{\sum\nolimits_{l=1}^{d} \beta_l^z k_{\mathcal{X}}(\cdot, x) \cdot \langle \phi(\cdot), e_l\rangle : \\ &\qquad\qquad x \in \mathcal{X}_{train}, z \in \mathcal{Y}, \beta_l^z \in \mathbb{R}\Big\}\end{aligned}$$

shows that

$$f^*(x, y) = \sum_{i=1}^{m}\sum_{l=1}^{d} \alpha_{l,i} k_{\mathcal{X}}(x, x_i)\langle \phi(y), e_l\rangle \qquad (2)$$

for appropriate coefficients $\alpha_{l,i} \in \mathbb{R}$. Now we have to optimize f^* over only $m \cdot d$ coefficients (instead of $m \cdot |\mathcal{Y}|$ coefficients as in the initial representation of $\mathcal{F}$). We will find the solution f^* of (1) by minimizing with respect to $\alpha \in \mathbb{R}^{d\times m}$. Following, we propose new loss-functions for ligand prediction which finally determines the optimization problem (1) completely.

3.2 Loss-Functions for Orphan Screening

As mentioned above, we want to learn a function f that maps positive examples to higher values than unlabelled ones. For the associated optimization problem we present different loss-functions and contrast them with each other. The loss-functions are all based on quadratic loss (compare Gärtner and Vembu (2009)). They vary in the treatment of labelled and unlabelled examples, respectively.

Suppose we have a set $\mathcal{X}_{train} = \{x_1, \ldots, x_m\}$ of proteins and a database of molecules $\mathcal{Y}$. Furthermore, for every $x_i \in \mathcal{X}$ we have a set $\mathcal{Y}_i$ of already verified ligands and the complement set of unlabelled database molecules $\mathcal{Z}_i$ with respect to x_i. We will refer to $(x_i, \mathcal{Y}_i)$ as the i-th training unit.

We propose to minimize the regularized risk functional (1) with regularization constant $\lambda > 0$ and training units $1, \ldots, m$, for the following loss-functions $\mathcal{R}_1, \ldots, \mathcal{R}_4$

(ranking simple)
$$\mathcal{R}_1(f, i) = \sum_{y\in\mathcal{Y}_i}\sum_{z\in\mathcal{Z}_i} [1 - f_i(y) + f_i(z)]^2$$

(classification simple)
$$\mathcal{R}_2(f, i) = \sum_{y\in\mathcal{Y}_i} [1 - f_i(y)]^2 + \tfrac{1}{|\mathcal{Z}_i|}\sum_{z\in\mathcal{Z}_i} [1 + f_i(z)]^2$$

(ranking mean)
$$\mathcal{R}_3(f, i) = \sum_{y\in\mathcal{Y}_i} \left[1 - f_i(y) + \tfrac{1}{|\mathcal{Z}_i|}\sum_{z\in\mathcal{Z}_i} f_i(z)\right]^2$$

(classification mean)
$$\mathcal{R}_4(f, i) = \sum_{y\in\mathcal{Y}_i} [1 - f_i(y)]^2 + \left[1 + \tfrac{1}{|\mathcal{Z}_i|}\sum_{z\in\mathcal{Z}_i} f_i(z)\right]^2,$$

where $f_i(y) := f(x_i, y)$.
The loss-functions $\mathcal{R}_1, \ldots, \mathcal{R}_4$ are categorized according to their treatment of unlabelled examples (*simple/mean*) and to their arrangement of examples via classification or ranking (*classification/ranking*). The functions $\mathcal{R}_2$ and $\mathcal{R}_4$ push the positive and unlabelled examples for every protein $x_i \in \mathcal{X}$ to scores of $+1$ or -1, respectively. In contrast, $\mathcal{R}_1$ and $\mathcal{R}_3$ rank ligands in an independent fashion for every training example $x_i \in \mathcal{X}_{train}$. Another qualitative differentiating feature is that $\mathcal{R}_1$ and $\mathcal{R}_2$ are constructed such that every single unlabelled example is ranked or classified. In contrast, $\mathcal{R}_3$ and $\mathcal{R}_4$ handle unlabelled examples as a unit. Only the mean of their values of f is supposed to be lower than the values of f for positive examples. For our data setting we would expect the classification mean approach to work best. We believe this because, on the one hand, we have classification-type data. On the other hand, it should be more appropriate to pull the mean of the values of the scoring function for unlabelled examples down. Actually, the unlabelled examples contain ligands which, by construction, are supposed to have high values of f.

An interesting property of the optimization problems above is that they are equivalent to using the second order Taylor approximation of the exponential loss. For example, consider

$$\min \lambda\|f\|^2 + \sum_{i=1}^{m}\sum_{y\in\mathcal{Y}_i}\sum_{z\in\mathcal{Z}_i} [1 - f_i(y) + f_i(z)]^2$$

and

$$\min \lambda\|f\|^2 + \sum_{i=1}^{m}\sum_{y\in\mathcal{Y}_i}\sum_{z\in\mathcal{Z}_i} \exp[1 - f_i(y) + f_i(z)].$$

Both optimization problems are equivalent except for a removal of multiplicative or additive constants resulting in a shift of the objective function.
In the sequel we will sketch how to obtain compact formulas for all functions necessary for optimization.

3.3 Objective Function and Algorithm

In order to solve the optimization problem in (1), according to the parameterization given by (2), we have to minimize with respect to $\alpha \in \mathbb{R}^{d\times m}$. Let us define the terms $Y \in \mathbb{R}^{m\times d}$, $C \in \mathbb{R}^{d\times d}$, and $\Phi \in \mathbb{R}^d$ by

$$Y_{i\cdot} := \sum_{y\in\mathcal{Y}_i} \phi^T(y)\,,\; C := \sum_{z\in\mathcal{Y}} \phi(z)\phi(z)^T\,,\; \Phi := \sum_{z\in\mathcal{Y}} \phi(z),$$

the kernel matrix of protein similarities, $[K]_{i,j=1}^{m} := k_{\mathcal{X}}(x_i, x_j)$, and the vectors of constants

$$\begin{aligned}V &:= (|\mathcal{Y}_1|, \ldots, |\mathcal{Y}_m|)^T\,, \\ W &:= (|\mathcal{Y}| - 2|\mathcal{Y}_1|, \ldots, |\mathcal{Y}| - 2|\mathcal{Y}_m|)^T\,, \\ S &:= \left(\frac{|\mathcal{Y}_1|}{|\mathcal{Z}_i|^2}, \ldots, \frac{|\mathcal{Y}_m|}{|\mathcal{Z}_m|^2}\right)^T, \\ \overline{S} &:= \left(\frac{1}{|\mathcal{Z}_i|^2}, \ldots, \frac{1}{|\mathcal{Z}_m|^2}\right)^T, \\ U &:= \left(\frac{|\mathcal{Y}_1|}{|\mathcal{Z}_i|}, \ldots, \frac{|\mathcal{Y}_m|}{|\mathcal{Z}_m|}\right)^T, \overline{U} := \left(\frac{1}{|\mathcal{Z}_i|}, \ldots, \frac{1}{|\mathcal{Z}_m|}\right)^T.\end{aligned}$$

As the definitions of C and Φ include counts of fingerprint features or feature combinations in linear time, it becomes clear why our method is called counting-based. These constants can be computed in linear time by a single pass

over the database. With this constants we obtain a compact formulation (independent of the size of the database) of the objective functions $o(\alpha, \mathcal{R}_1), \ldots, o(\alpha, \mathcal{R}_4)$ with the respective loss-functions $\mathcal{R}_1, \ldots, \mathcal{R}_4$

$$
\begin{aligned}
o(\alpha, \mathcal{R}_1) &= \lambda \operatorname{tr}(\alpha K \alpha^T) - |\mathcal{Y}| \operatorname{tr}(Y \alpha K) + V^T K \alpha^T \Phi \\
&\quad + \frac{1}{2} \sum_{i=1}^{m} W_i (\alpha K_{\cdot i})^T C_{\mathcal{Y}_i} (\alpha K_{\cdot i}) \\
&\quad + \frac{1}{2} V^T \operatorname{diag}(K \alpha^T C \alpha K) \\
&\quad - \Phi^T \alpha K \operatorname{diag}(Y \alpha K) + \| \operatorname{diag}(Y \alpha K) \|^2 \\
o(\alpha, \mathcal{R}_2) &= \lambda \operatorname{tr}(\alpha K \alpha^T) - (1 + \overline{U})^T \operatorname{diag}(Y \alpha K) \\
&\quad + \overline{U}^T K \alpha^T \Phi + \frac{1}{2} \overline{U}^T \operatorname{diag}(K \alpha^T C \alpha K) \\
&\quad + \frac{1}{2} \sum_{i=1}^{m} (1 - \overline{U})_i (\alpha K_{\cdot i})^T C_{\mathcal{Y}_i} (\alpha K_{\cdot i}) \\
o(\alpha, \mathcal{R}_3) &= \lambda \operatorname{tr}(\alpha K \alpha^T) - (\mathbf{1} + U)^T \operatorname{diag}(Y \alpha K) \\
&\quad + U^T K \alpha^T \Phi + \frac{1}{2} \sum_{i=1}^{m} (\alpha K_{\cdot i})^T C_{\mathcal{Y}_i} (\alpha K_{\cdot i}) \\
&\quad + \frac{1}{2} \Phi^T \alpha K (S \circ K \alpha^T \Phi) \\
&\quad - \Phi^T \alpha K ((S + \overline{U}) \circ \operatorname{diag}(Y \alpha K)) \\
&\quad + \left\| \left(\sqrt{\frac{1}{2} S + \overline{U}} \right) \circ \operatorname{diag}(Y \alpha K) \right\|^2 \\
o(\alpha, \mathcal{R}_4) &= \lambda \operatorname{tr}(\alpha K \alpha^T) - (\mathbf{1} + \overline{U})^T \operatorname{diag}(Y \alpha K) \\
&\quad + \overline{U}^T K \alpha^T \Phi + \frac{1}{2} \sum_{i=1}^{m} (\alpha K_{\cdot i})^T C_{\mathcal{Y}_i} (\alpha K_{\cdot i}) \\
&\quad + \frac{1}{2} \Phi^T \alpha K (\overline{S} \circ K \alpha^T \Phi) \} \\
&\quad - \Phi^T \alpha K (\overline{S} \circ \operatorname{diag}(Y \alpha K)) \\
&\quad + \left\| \left(\sqrt{\frac{1}{2} \overline{S}} \right) \circ \operatorname{diag}(Y \alpha K) \right\|^2 .
\end{aligned}
$$

With standard techniques we also obtained the gradient of the objective function and the product of the Hessian with a vector $v \in \mathbb{R}^{d \times m}$ which we need for optimization. Due to the positive semidefinite kernel matrix K and the particular structure of the objective function, we deal with a convex optimization problem. Hence, we can use the Newton conjugate gradient method to obtain a solution for our minimization problems.

Our ligand prediction algorithm works as follows for a given target protein x: After learning the scoring function f we calculate the values $f(x, z)$ for every database molecule $z \in \mathcal{Z}_x$ (note, orphan screening and virtual screening for non-orphan targets only vary in the data used for training). Afterwards, all those molecules are sorted with respect to their score. The performance of the algorithm can be measured by recovery rates, i.e., the number of actual ligands among the first s sorted molecules (compare also Geppert et al., 2009), where s is an appropriate threshold.

4 Conclusion and Future Work

In this paper we developed several counting-based structured output prediction algorithms for orphan screening as well as traditional virtual screening. For that we established four loss-functions evaluating the quality of a scoring function f. The algorithms aim at an assignment of scores to small molecules in a database according to how good they bind a target protein, in particular, an orphan target. The loss-functions are constructed such that: After a sorting of the database with respect to the scoring values, ligands of the target protein should be in the top positions of the sorted list of molecules. Furthermore, the design of the loss-functions shows our intention to model the database molecules as unlabelled examples.

We will measure the accuracy of our approach in real-world virtual as well as orphan screening settings. Concretely, we want to test our four methods on a set of proteins, each with a set of known ligands. Furthermore, we plan to vary the algorithm. For example, we want to include actual negative examples, insert an additional variance term for the neutral examples, or invert input and output space. Moreover, an important part of our practical studies will be the comparison of our algorithm with SVMStruct and baseline orphan screening methods. In addition to the classification setting, we want to test regression approaches.

Acknowledgments

This work was supported in part by the German state NRW within the B-IT Research School and by the German Research Foundation (DFG) under the reference number 'GA 1615/1-1'.

References

D. Erhan, P.-J. L'Heureux, S. Y. Yue, and Y. Bengio. Collaborative filtering on a family of biological targets. *Journal of Chemical Informatics and Modelling*, 46: 626–635, 2005.

T. Gärtner and S. Vembu. On structured output training: hard cases and an efficient alternative. *Machine Learning Journal (Special Issue of ECML PKDD)*, 76(2):227–242, 2009.

H. Geppert, T. Horváth, T. Gärtner, S. Wrobel, and J. Bajorath. Support-vector-machine-based ranking significantly the effectiveness of similarity searching using 2d fingerprints and multiple reference compounds. *Journal of Chemical Informatics and Modelling*, 48:742–746, 2008.

H. Geppert, J. Humrich, D. Stumpfe, T. Gärtner, and J. Bajorath. Ligand prediction from protein sequence and small molecule information using support vector machines and fingerprint descriptors. *Journal of Chemical Informatics and Modelling*, 49:767–797, 2009.

L. Jacob and J.-P. Vert. Protein-ligand interaction prediction: an improved chemogenomics approach. *Bioinformatics*, 24(19):2149–2156, 2008.

B. Schölkopf, R. Herbrich, and A. J. Smola. A Generalized Representer Theorem. In *Computational Learning Theory*, volume 2111 of *Lecture Notes in Computer Science*, pages 416–426. Springer, Berlin, 2001.