Ligand Affinity Prediction with Multi-Pattern Kernels

Katrin Ullrich[1,2], Jennifer Mack[1], and Pascal Welke[1]

[1] University of Bonn, Germany
ullrich@iai.uni-bonn.de, mack@iai.uni-bonn.de, welke@uni-bonn.de
[2] Fraunhofer IAIS, Sankt Augustin, Germany

Abstract. We consider the problem of affinity prediction for protein ligands. For this purpose, small molecule candidates can easily become regression algorithm inputs if they are represented as vectors indexed by a set of physico-chemical properties or structural features of their molecular graphs. There are plenty of so-called molecular fingerprints, each with a characteristic composition or generation of features. This raises the question which fingerprint to choose for a given learning task? In addition, none of the standard fingerprints, however, systematically gathers all circular and tree patterns independent of size and the adjacency information of atoms. Since structural and neighborhood information are crucial for the binding capacity of small molecules, we combine the features of existing graph kernels in a novel way such that finally both aspects are covered and the fingerprint choice is included in the learning process. More precisely, we apply the Weisfeiler-Lehman labeling algorithm to encode neighborhood information in the vertex labels. Based on the relabeled graphs we calculate four types of structural features: Cyclic and tree patterns, shortest paths and the Weisfeiler-Lehman labels. We combine these different *views* using different multi-view regression algorithms. Our experiments demonstrate that affinity prediction profits from the application of multiple views, outperforming state-of-the-art single fingerprint approaches.

Keywords: graph kernels, molecular fingerprints, multiple kernel learning, support vector regression, Weisfeiler-Lehman labeling

1 Introduction

In biological organisms small molecular compounds bind to large proteins with protein-ligand-specific affinities. If the real-valued affinity exceeds a given limit the compound is called a *ligand* of the protein. The ligand binding process typically triggers biochemical processes, for example, via a change of conformation or charge at the protein surface. *Ligand (affinity) prediction* is an important and challenging practical problem in the range of *quantitative structure-activity relationship* (QSAR) models as many drugs act as ligands. In practice, for *drug discovery* and *design* millions of compounds can be tested in the laboratory already

quite efficiently with *high-throughput-screening* (HTS) setups. Nevertheless, because of the expensive equipment and the quasi infinite number of synthesizable chemical compounds the process is still very time- and cost-consuming. Therefore, in chemoinformatics *similarity-based virtual screening* uses statistical ranking and machine learning methods in combination with molecular descriptors to train a binding model for the considered protein. Behind these approaches is the similarity assumption that similar compounds show similar binding behavior. As mentioned above, the protein-ligand-docking complex has a certain strength which can be measured as *binding affinity* K_i. For the prediction of affinities the well-known *support vector regression* (SVR) utilizing a vectorial feature representation of small molecules, the so-called *molecular fingerprints*, is the state-of-the-art method and was tested successfully (e.g., [9], [1], [17]). However, other machine learning algorithms like neural networks trained with molecular fingerprint data [10] were applied for the affinity prediction task as well. For a complete overview of approaches we point to the survey of Cherksov et al. [4].

Many publicly available or commercial fingerprint descriptors exist, just to mention a few: Maccs Keys, ECFP/C and FCFP/C fingerprints, GpiDAPH fingerprints, TGD or TGT. The fingerprints list (or count) diverse physico-chemical properties of the molecule, structural properties of their molecular graphs, or 3D information, and can be grouped according to their respective generation of features [2]. Additionally, even more molecular descriptors gathering graph structure information arise from the field of graph theory and have been proven beneficial (e.g., [11], [7]). The variety of data descriptions is a blessing and a curse at the same time. On the one hand, a lot of different information sources for diverse learning tasks are available. On the other hand, this includes the necessity to choose a representation in order to obtain optimal results. Hence, we intend to overcome this problem by using multiple representations simultaneously. There are related QSAR approaches (e.g., [7], [18]) which we would like to complement in this paper. Anyway, the method we will present is applicable for arbitrary graph data with multiple representations and real-valued labels.

Both structural and neighborhood information are crucial for the capacity of small molecules to be a ligand and for the strength of the bond (e.g., [12], [7]). For example, the presence of a benzene ring or that of an alcoholic group and their relative positions influence the chemical properties of the compound at hand. None of the existing fingerprints that collect structural information, however, captures both all circular and tree patterns of the molecular graph independent of size and the adjacency and connectivity information of atoms within the graph structure. Therefore, we propose to combine the feature set of the *cyclic pattern kernel* (CPK) [8] with that of shortest path (SP) kernels [3] and *Weisfeiler-Lehman* (WL) labels [14]. The WL algorithm assigns (new) labels to each vertex in the graph that depend on the surrounding vertices up to a certain distance h. CPK decomposes a graph into the set of contained cycles ($\mathcal{C}$) and remaining tree components ($\mathcal{T}$) of edges that do not belong to cycles. Shortest path features ($\mathcal{P}$) collect the shortest paths from one vertex

to another. Finally, we also consider the labels of the atoms ($\mathcal{L}$) themselves as features. Hence, for each depth h of the WL algorithm, we obtain four types of features $(\mathcal{C}_h, \mathcal{T}_h, \mathcal{P}_h, \mathcal{L}_h)$ for each graph. Each of these $4 \cdot h$ feature sets can possibly be a (weighted) part of a resulting fingerprint. Additionally, a feature vector can either list or count features of a compound (binary or counting feature representation, see [12]). However, it is neither clear which of them to keep in the application scenario of affinity prediction, nor obvious which role the components play for the predictive process. Hence, we apply a systematic process to obtain an optimal combination of the proposed feature sets.

On an abstract level, these components can also be considered multiple views on molecular graphs. Including different views on data to learn one prediction function is known as multi-view learning. In general, a view or feature vector representation Φ is canonically related to a kernel function k via *Mercer's theorem*

$$k(x, x') = \langle \Phi(x), \Phi(x') \rangle \quad , \quad x, x' \in \mathcal{X}, \tag{1}$$

where $\mathcal{X}$ is the instance space of learning objects. In our case $\mathcal{X}$ are the potential ligands and $\Phi(x)$ would be the feature vector. We use the supervised kernel methods *multiple-kernel learning* (MKL) and its solution from Vishwanathan et al. [19] and *learning kernel ridge regression* (LKRR) by Cortes et al. [6]. Both algorithms learn a linear combination of kernel functions corresponding to the provided views. The linear combination of functions is included into a regularized empirical risk functional with ε-insensitive loss function (MKL) or squared loss function (LKRR). Hence, they virtually solve a multi-view SVR and multi-view *regularized least squares regression* (RLSR) problem. We will employ these multi-view kernel methods to find and use a combination of our features that suits the learning process best. Out of the rich set of patterns we aim at finding optimal compositions or combinations of views for the affinity prediction learning task. In the case of the linear kernel this is equivalent with utilizing a novel fingerprint representation with differently weighted pattern components, such that the identification of optimal weights is part of the learning process itself. Finally, this approach allows us to incorporate multiple feature representation for structural and neighborhood information with an automatic weighting that highlights the importance of the pattern group for the affinity prediction task.

2 Regression with Multiple Views

In the practical scenario of ligand affinity prediction we are looking for a real-valued predictor function f defined on an instance space $\mathcal{X}$ of molecules, i.e., $f : \mathcal{X} \to \mathbb{R}$. Here, f should be an element of a certain candidate space of functions $\mathcal{H}$ and the molecules in $\mathcal{X}$ can be regarded as graphs of atoms and bonds (see Fig. 1 below). A particular choice for $\mathcal{H}$ is a *reproducing kernel Hilbert space* (RKHS), a function space that is canonically related to a so-called kernel function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. In the supervised scenario with training examples $(x_1, y_1), \ldots, (x_n, y_n) \in \mathcal{X} \times \mathbb{R}$, a common approach to find a good predictor

is the principle of regularized *empirical risk minimization* (ERM) as a trade-off between empirical risk and function norm. Utilizing the ε-insensitive loss or the squared loss function for the empirical risk we obtain the SVR and RLSR formulation, respectively. For a thorough study of SVR and RLSR as well as their kernelized variants consult, for example, [16] or [5]. When we choose an RKHS $\mathcal{H}$ with reproducing kernel k as function space the *representer theorem* of Schölkopf et al. [15] implies a representation of the solution f^*. In fact, it turns out that f^* must be a linear combination of the kernel function k centered at training points that can be used to parameterize all considered optimization problems. Now let $\Phi_1, \ldots, \Phi_M$ be M views with corresponding kernels $k_1, \ldots, k_M$ according to (1), and respective RKHSs $\mathcal{H}_1, \ldots, \mathcal{H}_M$. Having multiple representations on data we want to apply SVR- and RLSR-related multi-view kernel algorithms. In contrast to using only one single view or kernel in the regularized ERM, the supervised multi-view approaches we consider incorporate a linear combination of kernel functions

$$k(x, x') = \sum_{v=1}^{M} b_v k_v(x, x') = \sum_{v=1}^{M} b_v \left\langle \Phi_v(x), \Phi_v(x') \right\rangle, \tag{2}$$

which itself is a kernel again for $b_v \geq 0$. In order to prevent overfitting, the linear factors $b = (b_1, \ldots, b_M)$ need to be regularized additionally. As the kernel expansion parameters from the representer theorem and linear factors b have to be learned simultaneously, the solution strategies of SVR- and RLSR-related multi-view approaches are different to the ones of single-view SVR and RLSR. On the one hand, the *multiple kernel learning* (MKL) algorithm presented in [19] utilizes the p-norm, $p > 1$, for the regularization of b. Hence, the MKL objective becomes $f_1^*, \ldots, f_M^* =$

$$\underset{f_v \in \mathcal{H}_v, b_v \geq 0}{\operatorname{argmin}} \quad \frac{1}{2} \sum_{v=1}^{M} \|f_v\|_{\mathcal{H}_v}^2 + C \sum_{i=1}^{n} \max\left\{0, |f(x_i) - y_i| - \varepsilon\right\} + \frac{\Lambda}{2} \left(\sum_{v=1}^{M} b_v^p \right)^{\frac{2}{p}},$$

where $f = \sum_{v=1}^{M} b_v f_v$ and $C, \Lambda, \varepsilon > 0$. Actually, there are other variants of MKL which we do not want to consider. On the other hand, the learning kernel ridge regression (LKRR) algorithm from [6] requires b being close to a constant vector b_0. The LKRR objective is

$$\begin{aligned} f_1^*, \ldots, f_M^* = \underset{f_v \in \mathcal{H}_v}{\operatorname{argmin}} \quad & \sum_{v=1}^{M} \|f_v\|_{H_v}^2 + C \sum_{i=1}^{n} |f(x_i) - y_i|^2 \\ \text{s.t.} \quad & \|b - b^0\| \leq \Lambda, b_v \geq 0, \end{aligned}$$

where again $f = \sum_{v=1}^{M} b_v f_v$ as well as $b_v^0, \Lambda > 0$. For the solution of MKL and LKRR we refer to [19] and [6].

3 Patterns for Molecular Graphs

Our contribution in this paper is the intelligent combination of several graph patterns for the task of ligand prediction. To this end we now define four pattern

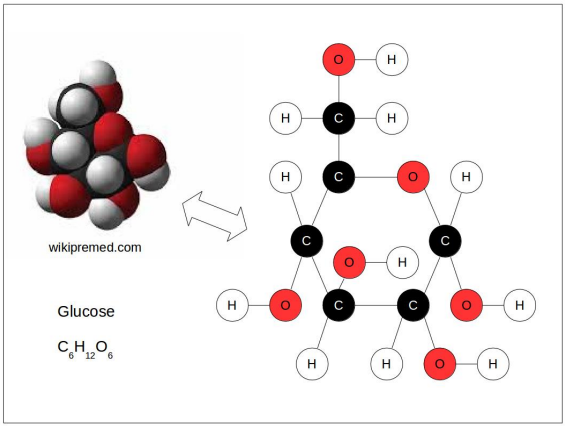

Fig. 1. Glucose molecule in its 3D (left) and graph representation (right). All bonds in glucose are single bonds, hence edge labels are omitted.

sets or classes and corresponding kernels that incorporate structural and neighborhood information. Finally, we define a multi-pattern kernel based on these patterns that allows efficient combinations of patterns based on the methods presented in the previous section.

A *labeled, undirected graph* is a quadruple $G = (V, E, \Sigma, \lambda)$, with V being a finite set of vertices, $E \subseteq \binom{V}{2}$ a set of edges, Σ a finite linearly ordered set of labels, and $\lambda : V \cup E \to \Sigma$ a function assigning a label to each vertex and edge. For the computation of patterns we consider molecules as labeled, undirected graphs such that atoms correspond to vertices (labels: C, O, H, N, ...) and bonds to edges (labels: single, double, and aromatic). An example is shown in Fig. 1. A sequence $w = \{v_0, v_1\}, \{v_1, v_2\}, \ldots, \{v_{k-2}, v_{k-1}\}, \{v_{k-1}, v_k\}$ of edges of a graph is called *simple path* if $v_i \neq v_j$ for all i, j with $1 \leq i < j \leq k$ (the vertex v should not be confused with the view index v). If additionally $v_0 = v_k$ holds true, the sequence is called *simple cycle*. Edges not belonging to any simple cycle are called *bridges*. A *forest* is an undirected graph that does not contain a cycle, a connected (i.e., where any two vertices are connected by a simple path) forest is called a *tree*. Two labeled, undirected graphs $G = (V, E, \Sigma, \lambda)$ and $G' = (V', E', \Sigma', \lambda')$ are *isomorphic*, if there is a bijection $\varphi : V \to V'$ that respects edges and labels, i.e., $\{v, w\} \in E$ if and only if $\{\varphi(v), \varphi(w)\} \in E'$, as well as $\lambda(v) = \lambda'(\varphi(v))$, and $\lambda(\{v, w\}) = \lambda'(\{\varphi(v), \varphi(w)\})$.

In the following we will review four types of graph patterns: Label patterns $\mathcal{L}$, cyclic patterns $\mathcal{C}$, tree patterns $\mathcal{T}$, and shortest path patterns $\mathcal{P}$. These patterns already appear in the definitions of the popular graph kernels Weisfeiler-Lehman kernel (WLK), cyclic pattern kernel (CPK), and shortest path kernel (SPK). First, we introduce the *Weisfeiler-Lehman labeling* of a graph G's vertices as the basis for the WL test of graph isomorphism as well as the WLK of Shervashidze et al. [14], represented by the recursive labeling function $\lambda_G^h : V \to \Sigma^*$ for

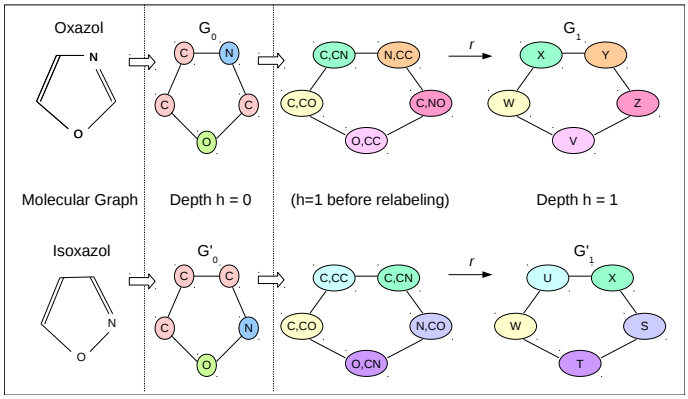

Fig. 2. WL labeling of two molecular graphs.

recursion depth h. Initially, it holds that $\lambda_G^0 = \lambda$. That means for depth $h = 0$ the WL labels are the original vertex labels. For each recursion step we append to each vertex label $\lambda_G^h(v)$, $v \in V$, a sorted list of all labels from adjacent vertices and obtain

$$\lambda_G^{h+1}(v) = r\left(\lambda_G^h(v), sort\left(\left\{\lambda_G^h(w) : (v,w) \in E\right\}\right)\right),$$

where a, b represents the concatenation of strings a and b (using a separating comma), $\{\cdot\}$ is a multiset, and r is a renaming function (see Shervashidze et al. [14]). Based on this definition we denote with $G_h = (V, E, \Sigma, \lambda_G^h)$ the graph G with WL labels of depth h. The edge labels remain unaffected in the sense that $\lambda_G^h(e) = \lambda(e)$ for all $h = 0, 1, \ldots$ and all $e \in E$. Two small examples can be found in Fig. 2.

Definition 1 (Label Patterns). *For a labeled, undirected graph G we define $\mathcal{L}(G)$ to be the set of all vertex labels of G.*

The above definition implies that $\mathcal{L}(G_h)$ are the WL labels of depth $h \geq 0$. Next, *cyclic* and *tree patterns* correspond to the features defined by Horváth et. al. [8] for the CPK.

Definition 2 (Cyclic Patterns). *Given a labeled, undirected graph G we define $S(G)$ to be the set of all simple cycles in G. Then $\mathcal{C}(G)$ denotes the set of canonical representations[3] of $S(G)$.*

That is, $\mathcal{C}(G)$ is equivalent to the set of simple cycles *up to isomorphism*. Hence, if G contains two isomorphic cycles, there will be only one cyclic pattern representing the two. Similarly, the tree patterns of a graph are only considered up to isomorphism.

[3] See [8] for a definition of such a canonical representation

Definition 3 (Tree Patterns). *Given a labeled, undirected graph G we define $T(G)$ to be the set of connected components of the forest that contains only the bridges in G. Then, $\mathcal{T}(G)$ denotes the set of canonical representations of $T(G)$.*

We enhance the expressiveness of the tree patterns $\mathcal{T}(G)$ by computing shortest paths between all pairs of vertices contained in the forest consisting of the bridges in G. To regain connectivity inside the forest, we contract each biconnected component to a single vertex and assign a fixed, unused label to each of those fusion vertices. We call this newly derived tree representation of the original graph *contracted graph* and define the shortest path patterns corresponding to the SPK of Borgwardt et al. [3] as follows.

Definition 4 (Shortest Path Patterns). *Given a labeled, undirected graph G we define $P(G)$ to be the set of shortest paths between all pairs of vertices in the contracted graph of G. With $\mathcal{P}(G)$ we denote the canonical representations of $P(G)$.*

Considering that a WL labeled graph G_h differs from its underlying original graph G only with respect to the labels, we can apply the definitions of cyclic, tree, and shortest path patterns to such graphs as well. This allows us to derive even more detailed patterns $\mathcal{C}(G_h)$, $\mathcal{T}(G_h)$, $\mathcal{P}(G_h)$, and of course $\mathcal{L}(G_h)$ for depths h greater than zero. With $\Phi_v : \mathcal{X} \to \mathbb{R}^{d_v}$, $v \in \{\mathcal{C}, \mathcal{T}, \mathcal{P}, \mathcal{L}\}$, we denote the binary or counting feature representation of the respective patterns. In practice, the feature space dimension d_v depends on the the view v, the considered depth h, and the graph dataset at hand. In Section 4 we will use the term (set) intersection or (set) counting kernel when we refer to the linear kernel on binary or counting feature vectors, respectively. Analogous to the WLK in [14] we define cumulative pattern kernels and a non-cumulative version of them.

Definition 5 (Pattern Kernel). *Let $v \in \{\mathcal{C}, \mathcal{T}, \mathcal{P}, \mathcal{L}\}$ be a graph pattern class and Φ_v its binary or counting feature mapping. For two labeled, undirected graphs G and G' the cumulative pattern kernel k_v^h of depth h is defined as*

$$k_v^h(G, G') = \langle \Phi_v(G_0), \Phi_v(G'_0) \rangle + \cdots + \langle \Phi_v(G_h), \Phi_v(G'_h) \rangle \,, \tag{3}$$

whereas the non-cumulative pattern kernel of depth h is just

$$k_v^h(G, G') = \langle \Phi_v(G_h), \Phi_v(G'_h) \rangle \,. \tag{4}$$

Obviously, (3) is a generalization of the WLK and (4) is an instance of its non-negatively weighted variant. Although we restrict to the linear view kernel $\langle \Phi_v(\cdot), \Phi_v(\cdot) \rangle$ in Definition 5 also other base kernels (compare WLK definition in [14]) could be applied. For the sake of convenience we do not use extra indices for cumulative/non-cumulative or intersection/counting kernels. This will be clear from the context in the practical part below. Finally, we define the multi-pattern kernel (MPK). Interestingly, with minor modifications the molecular fingerprint ECFPx corresponds to the cumulative WL labels of a molecular graph up to depth $h = x/2$.

Definition 6 (Multi-Pattern Kernel). *We consider non-negative weights b_v, $v \in \{\mathcal{C}, \mathcal{T}, \mathcal{P}, \mathcal{L}\}$. The multi-pattern kernel k_{MPK} of two labeled, undirected graphs G and G' is defined as*

$$k_{MPK}(G, G') = \sum_{v \in \{\mathcal{C}, \mathcal{T}, \mathcal{P}, \mathcal{L}\}} b_v \cdot k_v^{h_v}(G, G') \quad , \quad b_v \geq 0,$$

where the WL depth h_v depends on the pattern v and $k_v^{h_v}$ can be a cumulative or non-cumulative pattern kernel.

Now we want to investigate MPKs with multi-view kernel approaches in the context of ligand affinity prediction.

4 Experiments

We provided a considerable number of representation variants for molecular graphs that can be used as views for single- and multi-view kernel approaches which themselves can be parameterized as well in several ways (regularization and kernel type). At first, in the preliminary experiments we intend to extract promising views or view combinations for the practical task of ligand affinity prediction using only the single view regression methods SVR and RLSR. For this purpose, the views are either individual graph pattern vectors or their concatenation. In a second step, we use the best patterns and check whether we can take profit from multi-view kernel methods for regression. We will compare our results with the performance of standard fingerprints.

Target	P23946	Q99895	P09871	P25774	Q9Y5Y6
Number	DS 1	DS 2	DS 3	DS 4	DS 5
Ligands	90	91	92	104	125
Range	5.4-8.9	2.7-8.0	4.8-9.0	4.3-9.8	4.0-10.1
Target	P17655	P42574	P00740	P07384	P07339
Number	DS 6	DS 7	DS 8	DS 9	DS 10
Ligands	128	133	171	189	197
Range	4.8-10.8	4.9-11.9	3.9-8.7	3.1-10.7	4.1-11.0
Target	P08709	P43235	P00750	P07858	P29466
Number	DS 11	DS 12	DS 13	DS 14	DS 15
Ligands	249	252	264	278	310
Range	3.9-9.5	3.9-11.5	2.2-9.5	3.0-10.5	3.1-9.8
Target	P07711	P00747	P00749	P08246	P07477
Number	DS 16	DS 17	DS 18	DS 19	DS 20
Ligands	357	474	600	742	986
Range	3.9-10.6	1.9-11.0	0.3-11.1	2.7-11.2	2.0-10.6

Table 1. Datasets with name, ordinal number, number of ligands, and label range.

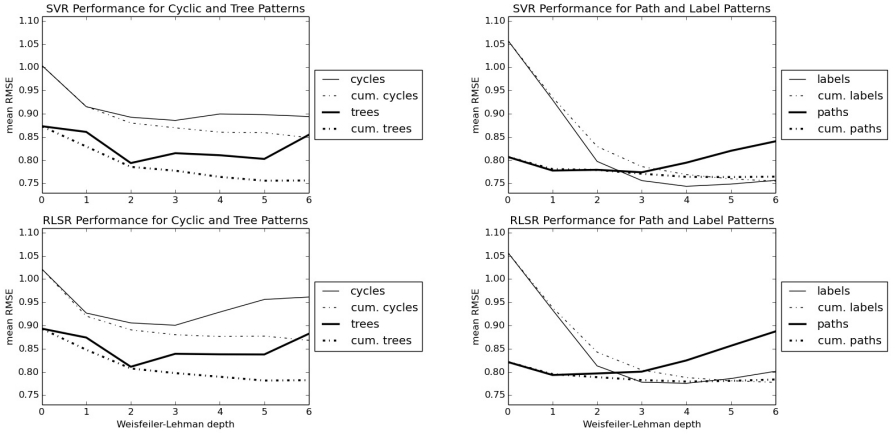

Fig. 3. SVR and RLSR results for the intersection kernel.

4.1 Setup and Datasets

Our experiments will be performed on 20 datasets, each of which representing ligands of one of 20 human proteins. A set contains between 90 and 986 ligands of the respective protein gathered from BindingDB[4]. We ordered the protein datasets according to the number of contained ligands and renamed them (see *Number* in Table 1, DS = dataset). We divided the 20 datasets into two groups: The first group, consisting of odd ordinal numbers, was used for preliminary and parameter tuning experiments. The second group with even numbers was utilized for the main experiments with multi-view kernel methods. Every ligand is a single molecule in the sense of a connected graph and is labeled with its affinity value ($pK_i = -\log_{10} K_i$) towards the protein target. The ligands are given in SMILES-format (Simplified Molecular Input Line Entry Specification) from which the labeled graph structure can be deduced easily, e.g., with the chemistry toolbox Open Babel[5] and the structure data format (SDF). We introduced the edge label *aromatic* by hand using a Hückel's rule heuristic. Thus, for all ligands in all datasets the binary and counting feature vectors for all pattern types up to WL depth $i = 6$ and the standard molecular fingerprints Maccs and ECFP6 were available for our experiments.

We use the SMO-MKL software[6] for efficient MKL that is based on libSVM[7] for both our MKL and SVR experiments. For LKRR we use our own implementation of Algorithm 1 in [6]. For a learned predictor function f we re-

[4] Binding database, `https://www.bindingdb.org/bind/index.jsp`

[5] `openbabel.org`

[6] Available at `http://research.microsoft.com/en-us/um/people/manik/code/smo-mkl/download.html`

[7] `https://www.csie.ntu.edu.tw/~cjlin/libsvm/`

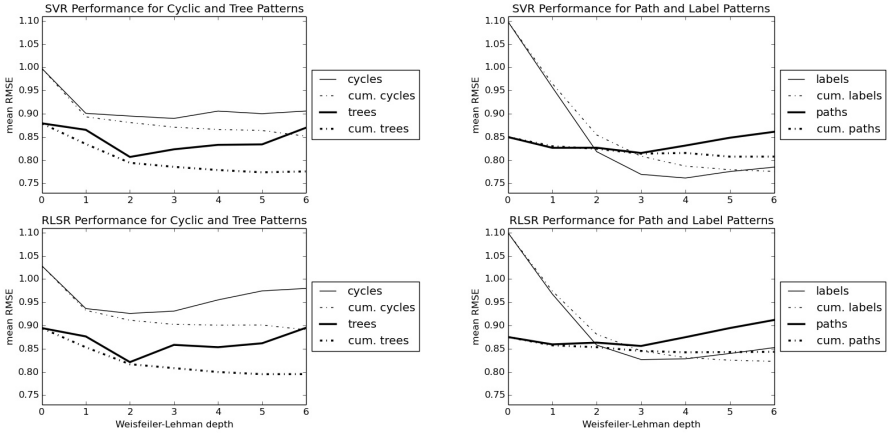

Fig. 4. SVR and RLSR results for the counting kernel.

port root mean squared error $\mathrm{RMSE}(f) = \frac{1}{\sqrt{m}} \|Y_{test} - f(X_{test})\|_2$, i.e., the root mean squared distance between the original label vector Y_{test} of m test instances X_{test} and the corresponding vector of prediction values $Y_{pred} = f(X_{test})$. In our experiments we perform 5-fold cross validation. In every training and parameter tuning fold we randomly sample 80% training data and the remainder as test instances. We use the linear kernel on binary and counting feature vectors $k_v(G, G') = \langle \Phi_v(G), \Phi_v(G') \rangle$. For the reason of calculation stability of the used software, we normalized every kernel matrix initially with its Frobenius matrix norm. For the parameters we chose $C \in [50.0, 100.0]$ for all algorithms and $\Lambda = 1.0$. The trade-off parameter C was generally chosen quite large during the parameter tuning phase (which we account for the kernel normalization), whereas all algorithms seemed to be almost insensible to the choice of Λ. Leaned on the expert knowledge in chemoinformatics with affinity prediction (e.g., [1]) we used $\varepsilon = 0.1$.

4.2 Results

Preliminary Experiments: Initially, we considered the patterns *cycles*, *trees*, *shortest paths*, and *labels* individually applying SVR and RLSR for the prediction of ligand affinities. Therefore, we used a cumulative and non-cumulative feature vector variant which we refer to with "cum. pattern" or "pattern", respectively. For the first variant, we use all features based on all WL depths up to some depth h in a concatenated feature vector. For the second, we only use features of a fixed depth h. The results can be found in Fig. 3 and 4. We report the mean RMSE with respect to all datasets with odd numbers. We observe that the qualitative performance trend is very similar for SVR and RLSR. Obviously, the non-cumulative patterns reach an optimal WL depth and decline for

greater depths. The cumulative ones appear to converge to the optimal performance with increasing WL depth. Nevertheless, the best RMSE is very similar for cumulative and non-cumulative patterns. In general, for the predictive task at hand the performance of labels seems to be very high for an appropriate WL depth, whereas for all WL depths the one for cycles is quite low.

Target	A	B	C	D	E	F
DS 2	1.039	**1.004**	1.010	1.007	1.000	**0.995**
DS 4	0.923	**0.719**	0.737	0.970	0.762	**0.746**
DS 6	1.052	**0.906**	0.915	1.063	0.903	**0.812**
DS 8	0.790	**0.655**	0.675	0.841	**0.618**	0.621
DS 10	0.897	**0.726**	0.755	1.083	**0.906**	0.914
DS 12	1.289	**1.077**	1.111	1.265	1.071	**1.053**
DS 14	1.254	**1.048**	1.078	1.271	1.073	**1.033**
DS 16	1.216	**0.948**	0.996	1.190	0.961	**0.925**
DS 18	1.068	**0.820**	0.834	1.073	0.842	**0.801**
DS 20	1.138	**0.869**	1.045	1.116	0.855	**0.816**

Table 2. RMSE for standard fingerprints. A: SVR with Maccs, B: SVR with ECFP6, C: MKL with Maccs and ECFP6, D: RLSR with Maccs, E: RLSR with ECFP6, F: LKRR with Maccs and ECFP6.

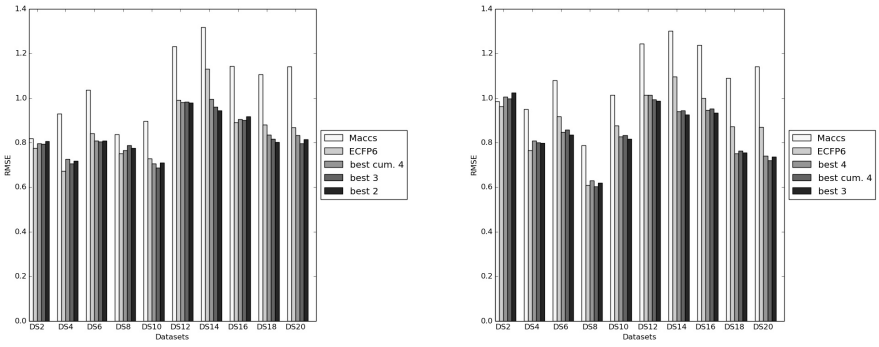

Fig. 5. Graphical visualization of main results for the counting kernel. Left: SVR/MKL results from Table 4, columns A, B, G, H, I. Right: RLSR/LKRR results from Table 6, columns A, B, F, G, H.

As the information for individual datasets is not apparent in the diagrams of Fig. 3 and 4 for each cumulative and non-cumulative pattern variant we chose the best WL depth and extracted the performance for every dataset with odd

Target	A	B	C	D	E	F	G	H	I
DS 2	**1.028**	1.056	1.100	1.084	1.071	1.095	1.088	1.107	1.101
DS 4	0.932	**0.707**	0.726	0.752	0.948	0.777	0.769	0.762	0.743
DS 6	1.068	0.894	**0.842**	0.882	1.041	0.891	0.864	0.881	0.860
DS 8	0.849	0.700	0.715	0.694	0.714	**0.693**	0.697	0.696	0.707
DS 10	0.935	0.744	0.747	0.757	0.819	0.742	0.752	**0.740**	0.745
DS 12	1.307	1.099	1.104	1.069	1.244	1.036	**1.030**	1.016	1.038
DS 14	1.261	1.102	0.931	0.956	1.118	0.932	**0.899**	0.913	0.917
DS 16	1.189	0.946	0.940	0.941	1.070	0.979	0.952	0.952	**0.931**
DS 18	1.103	0.846	0.869	0.838	0.804	**0.784**	0.785	0.787	0.832
DS 20	1.110	0.838	-	0.886	-	-	0.834	0.751	**0.718**

Table 3. RMSE for SVR/MKL experiments with intersection kernel. A: SVR with Maccs, B: SVR with ECFP6, C: SVR with cum. SPK $h = 6$, D: SVR with cum. WLK $h = 6$, E: SVR with cum. CPK $h = 6$, F: MKL with best depth of all patterns, G: MKL with best depth of all cum. patterns, H: MKL with best depths of 3 best patterns, I: MKL with best depths of 2 best patterns, "-" algorithm did not converge.

number. We omit the results for reasons of space, but in essence they show the same impact of the different pattern classes for individual datasets as for the average over the datasets.

Main Experiments: At first, we tested whether we can already take profit from multi-view approaches using standard molecular fingerprints only. The results are presented in Table 2. We find that in the case of RLSR/LKRR there is a performance improvement in favor of the multi-view algorithm. This cannot be verified in the case of SVR/MKL as single-view SVR with fingerprint ECFP6 turns out to be the best method for all datasets. Subsequently, we investigated whether we can take advantage of different graph pattern features in the multi-view setting. Actually, there are too many pattern combinations to test all of them in the scope of multi-view algorithms. Therefore, we chose the most promising combinations and depths from the preliminary experiments (with respect to the RMSE) to compare them with baseline kernels and standard fingerprints. The results are shown in Tables 3 to 6, as well as Fig. 5. In columns A and B the single-view results with standard fingerprints are shown. The columns C to E present the performance of the popular graph kernels SPK, WLK, and CPK, each with its optimal depth taken from the preliminary experiments. Finally, the columns F - I obtain the results for optimal multi-view combinations from the preliminary experiments (see the respective table caption). In the case of SVR/MKL experiments we observe that MKL approaches show the best performances in 7 out of 10 cases (Table 3). If we utilize the counting kernel the MKL approaches outperform the single-view SVR variants with standard molecular fingerprints or standard graph kernels in the majority of cases (Table 4). In the RLSR/LKRR scenario the multi-view approaches exhibit the lowest RMSE for 8 out of 10 or 7 out of 10 datasets when we apply the intersection or counting

Target	A	B	C	D	E	F	G	H	I
DS 2	0.818	**0.776**	0.815	0.787	0.805	0.802	0.796	0.794	0.807
DS 4	0.930	**0.673**	0.675	0.691	0.894	0.728	0.725	0.706	0.718
DS 6	1.037	0.841	**0.763**	0.820	0.995	0.826	0.808	0.805	0.808
DS 8	0.837	**0.751**	0.785	0.761	0.781	0.775	0.764	0.788	0.776
DS 10	0.896	0.728	0.698	0.715	0.807	0.698	0.705	**0.687**	0.710
DS 12	1.231	0.991	0.993	**0.973**	1.212	1.007	0.981	0.984	0.978
DS 14	1.317	1.130	0.990	0.989	1.205	0.978	0.994	0.959	**0.943**
DS 16	1.142	0.891	**0.860**	0.886	1.008	0.912	0.904	0.902	0.917
DS 18	1.106	0.880	0.893	0.877	0.846	0.822	0.834	0.816	**0.803**
DS 20	1.142	0.868	-	-	-	**0.767**	0.833	0.797	0.814

Table 4. RMSE for SVR/MKL experiments with counting kernel. A: SVR with Maccs, B: SVR with ECFP6, C: SVR with cum. SPK $h = 5$, D: SVR with cum. WLK $h = 6$, E: SVR with cum. CPK $h = 4$, F: MKL with best depth of all patterns, G: MKL with best depth of all cum. patterns, H: MKL with best depths of 3 best patterns, I: MKL with best depths of 2 best patterns, "-" algorithm did not converge.

kernel, respectively (Tables 5 and 6). We observe that the single-view approaches SVR and RLSR applying the standard graph kernel features of SPK, WLK, and CPK outperform the other algorithms in very few cases. Most interestingly, the more simple non-cumulative pattern combinations perform very well in comparison to cumulative combinations, even if we do not lift all pattern types cycles, trees, shortest paths, or labels, but rather only 2 or 3 of them. Apparently, it is sufficient to use non-cumulative pattern combinations of few pattern types.

5 Conclusion

We considered the problem of ligand affinity prediction with a variety of different feature vectors representing small molecular compounds and compared single- and multi-view regression approaches for this learning task. We showed that one can profit from the application of linear combinations of multiple views on molecular data in this practical scenario. It turned out that the multi-view approaches based on structural features and neighborhood information outperform the SVR and RLSR algorithm using standard molecular fingerprints or popular graph kernels. This effect was the more visible the greater the dataset sizes were (see Fig. 5). During our experiments we observed that the application of WL labels $\mathcal{L}$ and shortest path patterns $\mathcal{P}$ improved the prediction results particularly. In general, the squared loss-approaches RLSR and LKRR achieved better results than the analogue ε-insensitive loss algorithms SVR and MKL. Hence, using combinations of graph patterns based on WL labels of appropriate depths together with multi-view methods represents a noteworthy alternative to the application of SVR and standard molecular fingerprints which is the state-of-the-art approach for affinity prediction in the field of QSAR modeling.

Target	A	B	C	D	E	F	G	H	I
DS 2	0.991	**0.990**	1.011	1.016	1.082	0.998	1.011	1.018	0.995
DS 4	0.966	0.839	0.827	0.879	1.039	0.807	0.815	0.797	**0.774**
DS 6	1.097	0.974	0.902	0.948	1.055	0.878	**0.870**	0.829	0.784
DS 8	0.820	0.678	0.715	0.684	0.741	0.669	0.688	**0.667**	0.689
DS 10	0.872	**0.697**	0.729	0.725	0.770	0.710	0.715	0.716	0.711
DS 12	1.175	1.028	1.000	0.985	1.210	0.976	0.973	0.954	**0.931**
DS 14	1.251	1.084	0.948	0.980	1.169	0.928	0.940	0.906	**0.887**
DS 16	1.186	0.958	0.933	0.952	1.088	0.918	0.910	0.901	**0.881**
DS 18	1.046	0.828	0.856	0.807	0.770	0.729	0.733	**0.723**	0.738
DS 20	1.160	0.888	0.810	0.890	0.961	0.764	0.767	0.753	**0.742**

Table 5. RMSE for RLSR/LKRR experiments with intersection kernel. A: RLSR with Maccs, B: RLSR with ECFP6, C: RLSR with cum. SPK $h = 4$, D: RLSR with cum. WLK $h = 6$, E: RLSR with cum. CPK $h = 2$, F: LKRR with best depth of all patterns, G: LKRR with best depth of all cum. patterns, H: LKRR with best depths of 3 best patterns, I: LKRR with best depths of 2 best patterns.

Acknowledgements. We want to thank Dr. Martin Vogt from the Department of Life Science Informatics, B-IT, of the university of Bonn for preparing the protein dataset and making it available for us. Furthermore, we thank Dr. Martin Vogt and his colleagues for many valuable discussions on this topic. We would also like to thank Prof. Thomas Gärtner for guidance and advice.

References

1. Balfer, J., Bajorath, J.: Artifacts in Support Vector Regression-Based Compound Potency Prediction Revealed by Statistical and Activity Landscape Analysis PLoS ONE (2015)
2. Bender, A., Jenkins, J. L., Scheiber, J., Sukuru, S. C. K., Glick, M., Davies, J. W.: How Similar Are Similarity Searching Methods? A Principal Component Analysis of Molecular Descriptor Space. J. Chem. Inf. Model. (2009)
3. Borgwardt, K. M., Kriegel, H.-P.: Shortest-Path Kernels on Graphs. Proceedings of ICDM (2005)
4. Cherkasov, A., Muratov, E. N., Fourches, D., Varnek, A., Baskin, I., Cronin, M., et al.: QSAR Modeling: Where Have Your Been? Where Are You Going To?. J. Med. Chem. (2014)
5. Christianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods. Cambridge University Press (2000)
6. Cortes, C., Mohri, M., Rostaminzadeh, A.: L_2 Regularization for Learning Kernels. Proceedings of UAI (2009)
7. Gaüzère, B., Brun, L., Villemin, D.: Treelet kernel incorporating cyclic, stereo and inter pattern information in Chemoinformatics. Pattern Recognition (2014)
8. Horváth, T., Gärtner, T., Wrobel, S.: Cyclic Pattern Kernels for Predictive Graph Mining. Proceedings of KDD (2004)

Target	A	B	C	D	E	F	G	H	I
DS 2	0.985	**0.962**	1.040	1.021	1.034	1.005	0.997	1.024	0.997
DS 4	0.949	**0.766**	0.782	0.826	1.049	0.808	0.801	0.798	0.828
DS 6	1.078	0.918	0.844	0.886	1.034	0.847	0.857	**0.834**	0.860
DS 8	0.787	0.610	**0.599**	0.621	0.676	0.630	0.604	0.620	0.620
DS 10	1.013	0.876	0.838	0.870	0.971	0.826	0.832	**0.816**	0.851
DS 12	1.244	1.013	1.007	1.006	1.254	1.014	0.993	**0.986**	**0.986**
DS 14	1.300	1.096	0.961	0.994	1.130	0.940	0.943	0.926	**0.924**
DS 16	1.238	0.998	0.968	0.988	1.093	0.946	0.951	**0.933**	0.961
DS 18	1.090	0.872	0.899	0.841	0.805	0.751	0.763	0.754	**0.734**
DS 20	1.140	0.869	0.764	0.851	0.912	0.741	**0.719**	0.736	0.785

Table 6. RMSE for RLSR/LKRR experiments with counting kernel. A: RLSR with Maccs, B: RLSR with ECFP6, C: RLSR with cum. SPK $h = 4$, D: RLSR with cum. WLK $h = 6$, E: RLSR with cum. CPK $h = 3$, F: LKRR with best depth of all patterns, G: LKRR with best depth of all cum. patterns, H: LKRR with best depths of 3 best patterns, I: LKRR with best depths of 2 best patterns.

9. Liu, W., Meng, X., Xu, Q., Flower, D. R., Li, T.: Quantitative prediction of mouse class I MHC peptide binding affinity using support vector machine regression (SVR) models. BMC Bioinformatics (2006)
10. Myint, K.-Z., Wang, L., Tong, Q., Xie, X.-Q.: Molecular Fingerprint-Based Artificial Neural Networks QSAR for Ligand Biological Activity Predictions. Mol. Pharmaceutics (2012)
11. Ning, X., Rangwala, H., Karypis, E.: Multi-Assay-based Structure-Activity-Relationship Models: Improving Structure-Activity-Relationship Models by Incorporating Activity Information from Related Targets. J. Chem. Inf. Model. (2009)
12. Ralaivola, L., Swamidass, S. J., Saigo, H., Baldi, P.: Graph kernels for chemical informatics. Neural Networks (2005)
13. Rogers, D., Hahn, M.: Extended Connectivity Fingerprints. J. Chem. Inf. Model. (2010)
14. Shervashidze, N., Schweitzer, P., van Leeuwen, E. J., Mehlhorn, K., Borgwardt, K. M.: Weisfeiler-Lehman Graph Kernels. The Journal of Machine Learning Research (2011)
15. Schölkopf, B., Herbrich, R., Smola, A. J., Williamson, R.: A Generalized Representer Theorem. Proceedings of COLT (2001)
16. Smola, A. J., Schölkopf, B.: A tutorial on support vector regression. Statistics and Computing (2004)
17. Sugaya, N.: Ligand Efficiency-Based Support Vector Regression Models for Predicting Bioactivities of Ligands to Drug Target Proteins. J. Chem. Inf. Model. (2014)
18. Qiu, S., Lane, T.: Multiple Kernel Support Vector Regression for siRNA Efficacy Prediction. IEEE/ACM Transactions on Computational Biology and Bioinformatics (2008)
19. Vishwanathan, S. V. N., Sun, Z., Theera-Ampornpunt, N., Varma, M.: Multiple Kernel Learning and the SMO Algorithm. Proceedings of NIPS (2010)