JOURNAL OF CHEMICAL INFORMATION AND MODELING

Article

pubs.acs.org/jcim

Atom Environment Kernels on Molecules

Hiroshi Yamashita,[†] Tomoyuki Higuchi,[‡,†] and Ryo Yoshida*[,‡,†,§,∥]

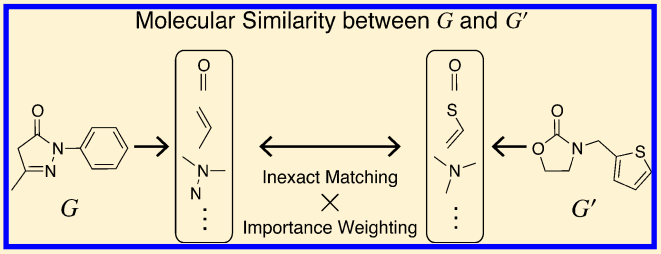
[†]The Graduate University for Advanced Studies, 10-3 Midori-cho, Tachikawa, Tokyo 190-8562, Japan

[‡]The Institute of Statistical Mathematics, 10-3 Midori-cho, Tachikawa, Tokyo 190-8562, Japan

[§]JST CREST, 10-3 Midori-cho, Tachikawa, Tokyo 190-8562, Japan

[∥]JST ERATO, 2-2-2 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0288, Japan

ABSTRACT: The measurement of molecular similarity is an essential part of various machine learning tasks in chemical informatics. Graph kernels provide good similarity measures between molecules. Conventional graph kernels are based on counting common subgraphs of specific types in the molecular graphs. This approach has two primary limitations: (i) only exact subgraph matching is considered in the counting operation, and (ii) most of the subgraphs will be less relevant to a given task. In order to address the above-mentioned limitations, we propose a new graph kernel as an extension of the subtree kernel initially proposed by Ramon and Gärtner (2003). The proposed kernel tolerates an inexact match between subgraphs by allowing matching between atoms with similar local environments. In addition, the proposed kernel provides a method to assign an importance weight to each subgraph according to the relevance to the task, which is predetermined by a statistical test. These extensions are evaluated for classification and regression tasks of predicting a wide range of pharmaceutical properties from molecular structures, with promising results.



1. INTRODUCTION

The definition of an appropriate similarity function between molecules is of crucial importance for many applications in chemical informatics. Common applications include structure−activity relationship[2,3] (SAR) model construction to predict the biochemical activity from a given molecular structure. The structure−activity relationship models rely on the similarity property principle,[4] which states that structurally similar molecules tend to have similar properties. Therefore, the SAR model, derived using an appropriate similarity function, will help guide the synthesis of new molecules.

Graphs are often used as a natural mathematical abstraction to describe the structure of molecules.[5] A molecule is translated to a labeled graph (or molecular graph), in which vertices correspond to atoms and edges correspond to covalent bonds between the atoms. The vertices are labeled with element types (e.g., carbon, oxygen, etc.) while the edges are labeled with bond types (e.g., single, double, etc.). Measurement of the similarity between the molecular graphs requires a method by which to transform any molecular graph G to a feature vector $\phi(G)$. Classically, molecular graphs are transformed into molecular descriptors,[6] which can be thought of as numerical representations that are encoded so as to capture the relevant aspects of molecular structures. The similarity between the molecular descriptors is then measured by a similarity metric, e.g., the Tanimoto coefficient,[7] cosine similarity, or the Gaussian radial basis function. To date, the molecular descriptors are widely applied due to their computational efficiency. However, such a transformation ϕ may cause some loss of structural information on molecules due to the limited dimensional feature space of the molecular descriptors.

Alternatively, molecular graphs can be compared directly in a potentially high or infinite dimensional feature space without the need to perform the explicit transformation, ϕ. This is possible when using a positive definite kernel[8,9] k on a set $\mathcal{G}$ of molecular graphs. The symmetric function $k: \mathcal{G} \times \mathcal{G} \to \mathbb{R}$ is said to be a positive definite kernel on $\mathcal{G}$ if and only if $\sum_{i,j\in\{1,\ldots,n\}} c_i c_j k(G_i,G_j) \geqslant 0$ for all $n \in \mathbb{N}$, $G_1, \ldots, G_n \in \mathcal{G}$, and $c_1, \ldots, c_n \in \mathbb{R}$. For such k, it is known that a map $\phi: \mathcal{G} \to \mathcal{H}$ into a Hilbert space $\mathcal{H}$ exists, such that $k(G,G') = \langle\phi(G),\phi(G')\rangle$ for all $G, G' \in \mathcal{G}$. We suppose that the feature map $\phi(G) = k(\cdot, G) \in \mathcal{H}$ of a kernel function $k(\cdot,G)$ is of substantially the same class as the feature vector of a molecular descriptor. The difference is whether the feature space is defined explicitly or implicitly. The R-convolution kernel[10] provides a framework to construct a wide class of kernel functions for structured objects such as molecular graphs, where each object is (implicitly) decomposed into a set of subgraphs, and the kernel between the objects is defined as the sum of kernel values among the subgraphs. Following this framework, various graph kernels have been proposed in the literature; see Vishwanathan et al.[11] These graph kernels differ with respect to the choice of the subgraph types used to represent the structured objects, such as walks,[12,13] shortest paths,[14] cycles,[15] and trees.[1,16] Mahé et al.[17] introduced two extensions to remove tottering walks and to increase the

Received: July 10, 2013
Published: May 6, 2014

number of different atom labels using the Morgan algorithm. Ralaivola et al. introduced three normalized variants[18] (Tanimoto, MinMax, and Hybrid) of the nontottering walk kernels. Subsequently, the efficient computation schemes for the random walk kernel[19] and the subtree kernel[20] were developed.

The above graph kernels all have two primary limitations. First, these graph kernels rely on exact subgraph matching where a successful match between subgraphs requires strict correspondence in terms of structure and vertex/edge labels. This means that if two subgraphs differ by only a single atom label, then the two subgraphs are considered to be completely different. The requirement for an exact match may reduce the expressivity of the resulting graph kernels. In an effort to address this problem, the elastic tree kernel[21] has been proposed for labeled ordered trees, which allows matching between vertices with different labels. Other similarity measures for inexact matching of subgraphs have been introduced in the optimal assignment kernel.[22] Second, when the number of distinct subgraphs is significantly large, the numerous irrelevant subgraphs for a given task overwhelm the contributions of the relatively few relevant subgraphs. This problem, which is known as the curse of dimensionality, adversely affects the generalization ability of the prediction models built on graph kernels.[23] Possible solutions to this problem include decreasing the contribution of larger subgraphs,[24] using prior knowledge to select relevant subgraphs,[25−27] and increasing the specificity of matching between subgraphs based on consideration of neighborhood information.[22,28]

To tackle the above limitations, we propose a new graph kernel, called the atom environment (AE) kernel, as an extension of the subtree kernel initially proposed by Ramon and Gärtner.[1] The AE kernel regards atoms as vertices labeled with information about the local atom environment. The atom environment labels are derived using an extension[29] of the Burden approach[30] and a variant[31] of the Morgan algorithm.[32] The AE kernel tolerates an inexact match between subgraphs by allowing matching between atoms having similar local environments. In addition, the AE kernel provides a method for assigning an importance weight to each subgraph according to the overall statistical significance of the constituent atoms for a given task.

The remainder of this paper is organized as follows. Section 2 introduces the necessary notation regarding graphs and touches on the subtree kernel on which the proposed kernel is based. Section 3 proposes a new kernel function for molecular graphs by extending the subtree kernel. Section 4 presents the computation of the proposed kernel. Section 5 shows the experimental results. In Section 6 we conclude the paper.

2. PRELIMINARIES

2.1. Graph Terminology and Notation. Let us represent a molecule by a labeled directed graph, $G = (\mathcal{V}, \mathcal{E})$. The graph G is described by a set of vertices $\mathcal{V} = \{v_i\}_{i=1}^{n}$ of size $n = |\mathcal{V}|$ representing the atoms in the molecule and a set of edges $\mathcal{E} = \{(u, v)\} \subseteq \mathcal{V} \times \mathcal{V}$ representing the covalent bonds. In the case of labeled graphs, there is also a set of labels Σ with a labeling function l: $\mathcal{V} \cup \mathcal{E} \rightarrow \Sigma$ that maps vertices and edges to corresponding element types and bond types, respectively. For directed graphs, each edge (u,v) is oriented and is a pair of the initial vertex u and the terminal vertex v. It is assumed that for every edge (u,v) belonging to $\mathcal{E}$ in G, the corresponding opposite edge (v,u) also belongs to $\mathcal{E}$; i.e., G is symmetric. Such symmetric directed graphs can be viewed as undirected graphs. Note that $\mathcal{V}_G$ and $\mathcal{E}_G$ will be used to refer to the vertex and edge sets, respectively, of a specific graph G. We also define a function describing the outgoing neighbors (children) of a vertex v as $\mathcal{N}(v) = \{u | (v, u) \in \mathcal{E}\}$.

A rooted tree $T = (\mathcal{V}_T, \mathcal{E}_T)$ is a directed acyclic graph with a single designated root, in which the edges have a natural orientation away from the root. The size $|T|$ of the tree T is the number of vertices in T, i.e., $|T| = |\mathcal{V}_T|$. The height h of the tree T is the length of the longest path from the root to any other vertex.

2.2. Subtree Kernels. In this section we describe the subtree (ST) kernel initially proposed by Ramon and Gärtner[1] and later extended by Mahé and Vert.[16] Following Mahé and Vert,[16] we start by describing the concept of tree-patterns in a graph. Let $G = (\mathcal{V}_G, \mathcal{E}_G)$ be a graph, and let $T = (\mathcal{V}_T, \mathcal{E}_T)$ with $\mathcal{V}_T = (w_1, ..., w_{|T|})$ be a rooted tree with a designated root w_1. A $|T|$-tuple of vertices $(v_1, ..., v_{|T|}) \in \mathcal{V}_G^{|T|}$ is said to be a tree-pattern of G with respect to T, denoted by $(v_1, ..., v_{|T|}) = \text{pattern}(T)$, if and only if

$$\begin{cases} \forall\, i \in \{1, ..., |T|\} & I(v_i) = I(w_i) \\ \forall\, (w_i, w_j) \in \mathcal{E}_T & (v_i, v_j) \in \mathcal{E}_G \wedge I((v_i, v_j)) = I((w_i, w_j)) \\ \forall\, (w_i, w_j), (w_i, w_k) \in \mathcal{E}_T & j \neq k \Leftrightarrow v_j \neq v_k \end{cases} \tag{1}$$

Note that a vertex in G may appear several times in the tree-pattern, but sibling vertices in the tree-pattern must correspond to distinct vertices in G. With the set of all possible tree-patterns of $G = (\mathcal{V}_G, \mathcal{E}_G)$ with $\mathcal{V}_G = (v_1, ..., v_{|\mathcal{V}_G|})$ arranged in T,

$$\mathcal{P}_T(G) = \{(v_{a_1}, ..., v_{a_{|T|}}) | (a_1, ..., a_{|T|}) \in \{1, ..., |\mathcal{V}_G|\}^{|T|} \wedge (v_{a_1}, ..., v_{a_{|T|}}) = \text{pattern}(T)\} \tag{2}$$

the ST kernel of graphs G and G' is given by

$$k_{\text{ST},h}(G, G') = \sum_{T \in \mathcal{T}_h} \mu(T) \sum_{p \in \mathcal{P}_T(G)} \sum_{p' \in \mathcal{P}_T(G')} I(p \cong p') \tag{3}$$

A set $\mathcal{T}_h$ of all trees up to height h is considered. We assume that $\mathcal{T}_h$ includes the elements of isolated vertices. For each tree $T \in \mathcal{T}_h$, the sets of tree-patterns $\mathcal{P}_T(G)$ and $\mathcal{P}_T(G')$ include all tree-patterns occurring in G and G', which can be arranged in a given tree T. Each tree-pattern pair $(p, p') \in \mathcal{P}_T(G) \times \mathcal{P}_T(G')$ is compared by the indicator function $I(p \cong p')$ that determines their isomorphism to be 1 if p and p' are isomorphic and 0 otherwise. In this case, $I(p \cong p')$ always returns 1 because both $\mathcal{P}_T(G)$ and $\mathcal{P}_T(G')$ include isomorphic tree-patterns. Therefore, the ST kernel counts the weighted number of co-occurrences of tree-patterns in G and G'. Each tree-pattern with respect to T has a weight $\mu(T)$ depending on the tree structure. A typical weight is a function of the tree size $|T|$, for example, $\mu(T) = \lambda^{2|T|}$, and assigns smaller weights to larger tree-patterns, where λ is a nonnegative weight factor that is less than one. Alternative weights have been defined as functions of the structural complexity of the tree.[16]

3. ATOM ENVIRONMENT KERNELS

3.1. Basic Idea. In this section we introduce two extensions to the ST kernel. The first extension, referred to as the inexact

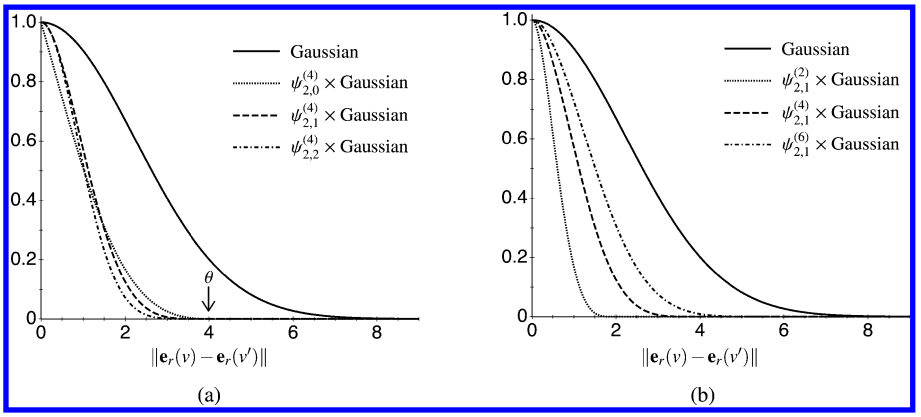


Figure 1. Plots of the Gaussian kernel with a width parameter of $\gamma = 0.1$ (solid line) and the CS kernels $\psi^{(\theta)}_{2,c}$ × Gasussian with respect to (a) the smoothing parameter c and (b) the cutoff distance θ.

match extension, relaxes the requirement for exact tree-pattern matching by allowing matching between atoms with similar local environments, and the second extension, referred to as the importance weight extension, introduces a tree weight function to adjust the contribution of each tree-pattern according to the overall statistical significance of the constituent atoms for a given task. For the inexact match extension, we alter the definition of tree-patterns by omitting the first condition for the exact atom label matching from eq 1 as

$$\begin{cases} \forall\ (w_i, w_j) \in \mathcal{E}_T & (v_i, v_j) \in \mathcal{E}_G \wedge l((v_i, v_j)) = l((w_i, w_j)) \\ \forall\ (w_i, w_j), (w_i, w_k) \in \mathcal{E}_T & j \neq k \Leftrightarrow v_j \neq v_k \end{cases}$$

This alters the definition of the set $\mathcal{P}_T(G)$ in eq 2. Suppose we are given a tree-level kernel $k_{\text{tree}}(p,p')$ to measure the similarity between tree-patterns p and p'. The AE kernel is then given by the weighted sum of $k_{\text{tree}}(p,p')$ over all possible pairs of tree-patterns induced from G and G'

$$k_{\text{AE},h}(G, G') = \sum_{T \in \mathcal{T}_h} \sum_{p \in \mathcal{P}_T(G)} \sum_{p' \in \mathcal{P}_T(G')} w(p)w(p')k_{\text{tree}}(p, p') \tag{4}$$

where $\mathcal{T}_h$ is a set of trees up to height h and $w(p)$ is a weight associated with the tree-pattern p. In the following section we provide the constructions of the tree-level kernel $k_{\text{tree}}(p,p')$ and the tree weight function $w(p)$.

3.2. Inexact Match Extension. Consider a specific form of the tree-level kernel $k_{\text{tree}}(p,p')$ between tree-patterns p and p'

$$k_{\text{tree}}(p, p') = \prod_{(v,v') \in \mathcal{A}(p,p')} k_{\text{atom}}(\mathbf{e}_r(v), \mathbf{e}_r(v'))$$

The atom-level kernel $k_{\text{atom}}(\mathbf{e}_r(v),\mathbf{e}_r(v'))$ measures the soft similarity between atoms v and v' through the atom environment labels $\mathbf{e}_r(v)$ and $\mathbf{e}_r(v')$. The atom environment label $\mathbf{e}_r(v)$ captures the local environment of each atom v in the molecular graph. As will be shown in Section 3.5, $\mathbf{e}_r(v) \in \mathbb{R}^d$ ($d = 2$ in the present study) is derived from the modified Burden matrix[29] of a neighboring substructure of a topological radius r centered at atom v. The tree-level kernel $k_{\text{tree}}(p,p')$ measures the similarity of p and p' as the product of the atom-level kernels over a set $\mathcal{A}(p, p') = \{(p[i], p'[i])\}_{i=1}^{|p|}$ of the aligned atom pairs of p and p', where $p[i]$ is the ith element of a tuple p.

We construct a compactly supported (CS) kernel for k_{atom} by multiplying the Gaussian kernel with a width parameter γ by a Wendland function[33]

$$k_{\text{atom}}(\mathbf{e}_r(v), \mathbf{e}_r(v')) = \psi_{d,c}\left(\frac{\|\mathbf{e}_r(v) - \mathbf{e}_r(v')\|}{\theta}\right) \exp(-\gamma\|\mathbf{e}_r(v) - \mathbf{e}_r(v')\|^2) \tag{5}$$

The Wendland functions $\psi_{d,c}$ are defined for the dimension d of input variables and the smoothing parameter c and tend to 0 when the L_2 distance $\|\mathbf{e}_r(v) - \mathbf{e}_r(v')\|$ is beyond a cutoff distance θ. With this construction, k_{atom} can smoothly decay to 0 at θ without losing positive definiteness.[34]

More specifically, the Wendland functions are defined as

$$\psi_{d,c}(z) = I^c\psi_{\lfloor d/2 \rfloor + c + 1}(z), \qquad c = 0, 1, 2, \ldots$$

with the truncated polynomial

$$\psi_s(z) = (1 - z)^s_+ = \begin{cases} (1 - z)^s & 0 \leqslant z < 1 \\ 0 & z \geqslant 1 \end{cases}$$

and the integral operator

$$I[f](z) = \int_z^\infty xf(x)\, \mathrm{d}x, \qquad z \geqslant 0$$

where $\lfloor \cdot \rfloor$ denotes the largest integer less than or equal to the argument and I^c indicates the I-operator that is applied c times and transforms the function ψ_s to a smoother function. These functions are positive definite on $\mathbb{R}^d$ for $d \leqslant 2s - 1$. We can compute the functions $\psi_{d,c}$ for $d = 2$ and $c = 0, 1, 2$ directly by the explicit form[35]

$$\psi_{2,0}(z) = (1 - z)^2_+$$

$$\psi_{2,1}(z) = (1 - z)^4_+(4z + 1)$$

$$\psi_{2,2}(z) = (1 - z)^6_+\left(1 + 6z + \frac{35}{3}z^2\right)$$

In Figure 1 the Gaussian kernel and the modified kernels with compact support using the Wendland functions for $d = 2$ with varying c and θ are shown. Since c is irrelevant to the sparsity of $\psi_{d,c}$ as shown in Figure 1, we fix $c = 0$ in this paper.

The compact support property of k_{atom} eliminates the redundant matches between atoms that have intrinsically different local environments. This will ensure the detection of pairs of chemically meaningful tree-patterns in two molecular graphs.

dx.doi.org/10.1021/ci400403w | J. Chem. Inf. Model. 2014, 54, 1289−1300

3.3. IMPORTANCE WEIGHT EXTENSION

Another important consideration is to determine the weight $w(p)$ of a tree-pattern p. We assign an importance weight to each tree-pattern according to the overall statistical significance of the constituent atoms for a given classification or regression task.

In the case of a classification task, the chi-squared (χ^2) statistic is used to measure the statistical significance of the atoms. Each atom v is characterized by another atom environment label $a_r(v) \in \mathbb{Z}$. As described later herein, $a_r(v)$ encodes information on a neighboring substructure of a topological radius r centered at atom v using a Morgan type algorithm.[31] Using a two-way contingency table (Table 1),

Table 1. Two-way Contingency Table of Atom Environment Label a and Class Label c[a]

	c	$\neg c$	$\sum$row
a	A	B	$A+B$
$\neg a$	C	D	$C+D$
$\sum$column	$A+C$	$B+D$	N

[a]The rows symbolize the presence and absence of the atom environment label a and the columns are the class labels (positive class c and negative class $\neg c$).

where the rows signify the presence and absence of the atom environment label $a_r(v) = a$ and the columns are the class labels (positive class c and negative class $\neg c$), the association of a with the class labels can be evaluated with the χ^2 statistic

$$\chi^2(a) = \frac{N(AD - BC)^2}{(A + C)(A + B)(B + D)(C + D)} \quad (6)$$

where A is the number of samples in which a and c co-occur, B is the number of samples in which a occurs without c, C is the number of samples in which c occurs without a, D is the number of samples in which neither c nor a occurs, and N is the total number of (training) samples. The value of $\chi^2(a)$ indicates the importance of atoms that have atom environment label a for the task of interest. Thus, the χ^2 statistic allows the identification of atoms with the ability to distinguish between two class labels. The weight of tree-pattern p is then given by

$$w(p) = \prod_{v \in p} \hat{w}(a_r(v))$$

with

$$\hat{w}(a_r(v)) = \begin{cases} \lambda_\alpha & \text{if } \chi^2(a_r(v)) \geq \tau \\ \lambda_\beta & \text{otherwise} \end{cases}, \quad 0 < \lambda_\beta \leq \lambda_\alpha < 1 \quad (7)$$

where τ is a χ^2 threshold. Once τ is given, the significant atoms satisfying $\chi^2(a_r(v)) \geq \tau$ are determined and have weight λ_α, and the other atoms have a relatively small weight λ_β. The importance weight $w(p)$ is expressed as the convolution of weight $\hat{w}(a_r(v))$ over the constituent atoms.

In the case of a regression task, Welch's t-test is used to assess the statistical significance of each atom with atom environment label a. Given two groups 1 and 2 of observations from molecules with and without a, the association of a with the task can be assessed by the t-statistic

$$t(a) = \frac{|\bar{y}_1 - \bar{y}_2|}{(\mathrm{var}(y_1)/n_1 + \mathrm{var}(y_2)/n_2)^{1/2}}$$

where $\bar{y}_i$, $\mathrm{var}(y_i)$, and n_i are the sample mean, sample variance, and sample size in the group i. Using $t(a)$ instead of $\chi^2(a)$ in eq 7, the tree weights for regression can be determined in the same manner as above.

For each tree-pattern p, we denote the number of atoms found to be significant and less significant as $n_\alpha(p)$ and $n_\beta(p)$, respectively. The AE kernel then becomes

$$k_{\mathrm{AE},h}(G, G') = \sum_{T \in \mathcal{T}_h} \sum_{p \in \mathcal{P}_T(G)} \sum_{p' \in \mathcal{P}_T(G')} \lambda_\alpha^{n_\alpha(p)+n_\alpha(p')} \lambda_\beta^{n_\beta(p)+n_\beta(p')} \times \prod_{(v,v') \in \mathcal{A}(p,p')} k_{\mathrm{atom}}(\mathbf{e}_r(v), \mathbf{e}_r(v')) \quad (8)$$

The atom-level kernels preserving positive definiteness are closed under tensor product and non-negative linear combinations.[36] The AE kernel is therefore positive definite.

In the case of unsupervised learning tasks, including cluster analysis and principal components analysis, the AE kernel could be applied by using prior knowledge on the importance of the atoms. In the case where a given pharmacophore set (e.g., hydrogen-bond acceptor and donor, hydrophobic, etc.) is used, if an atom plays the pharmacophore role, the atom is given a higher weight λ_α. Alternatively, subject to a uniform weight $\lambda_\alpha = \lambda_\beta$ in eq 8, the AE kernel can perform unsupervised learning tasks while we still benefit from the importance weight extension.

3.4. Relation to Previous Research. In this section we highlight the differences between the AE kernel (eq 4) and the ST kernel (eq 3). These kernels are both composed of two building blocks: the tree-level kernel and the tree weight function.

The ST kernel relies on the tree-level kernel $I(p \cong p')$, where a successful match between tree-patterns p and p' requires strict correspondence in terms of structure and vertex/edge labels. The AE kernel relaxes the requirement for an exact match of the vertex labels. Instead of $I(p \cong p')$, the AE kernel uses the tree-level kernel $k_{\mathrm{tree}}(p,p')$ with compact support. The $k_{\mathrm{tree}}(p,p')$ tolerates an inexact match between p and p' satisfying the condition $\|\mathbf{e}_r(v) - \mathbf{e}_r(v')\| < \theta$ for all $(v,v' \in \mathcal{A}(p,p))$. The property of compact support eliminates redundant tree-pattern matches.

Another difference lies in the method used to determine tree weights. In the ST kernel, the tree weight function $\mu(T)$ only depends on the tree structure; for example, $\mu(T)$ decreases as the size or complexity of the tree increases. In the AE kernel, the tree weight function $w(p)$ also decreases as the tree size increases. However, this decrease is alleviated by an increase in the number of relevant atoms for the task of interest. In Section 5, we demonstrate how these extended building blocks improve the performance in predicting various pharmaceutical properties of molecules.

3.5. Atom Environment Labels (Continuous). The atom environment label $\mathbf{e}_r(v) \in \mathbb{R}^2$ is derived from a modified Burden matrix[29] of a neighboring substructure of a topological

radius r centered at atom v. The $n \times n$ matrix $\mathbf{B} = (B_{ij})$ for a substructure of size n is given by

$$B_{ij} = \begin{cases} Z_i + 0.1\Delta_i + 0.01\pi_i & \text{if} \quad i = j \\ 0.4d_{ij}^{-1} & \text{if} \quad i \neq j \end{cases}$$

where, for the ith atom, Z_i is the atomic number, Δ_i is the number of non-hydrogen neighbors, π_i is the number of π electrons, and d_{ij} is the length of the shortest paths between the ith and jth atoms. We modify the off-diagonal elements representing edges between the center atom v and another atom to increase the centrality of v in the neighboring substructure; that is, the off-diagonal element B_{ij} is multiplied by 2 if the atom v corresponds to either the ith or the jth atom. This modification is necessary to distinguish between atom v and atoms of the neighboring substructure. The atom environment label is then defined as the concatenation of the smallest eigenvalue $e_{\min}$ and the largest eigenvalue $e_{\max}$ of $\mathbf{B}$, i.e., $\mathbf{e}_r(v) = (e_{\min}, e_{\max})^t$.

3.6. Atom Environment Labels (Discrete). Another atom environment label $a_r(v) \in \mathbb{Z}$ is generated in order to capture information on a neighboring substructure of a topological radius r centered at atom v using a variant[31] of the Morgan algorithm.[32] The variant algorithm consists of r iterations. An initial integer code is first assigned to each atom in such a way that the atomic properties are packed into a single integer value using a hash function. At each iteration, a new integer code of each atom v is generated by combining the current codes of all neighbors and the atom of interest. After r iterations, the final integer code of each atom v is returned as the atom environment label $a_r(v)$. The following atomic properties are considered for the assignment of the initial codes: the number of bonds to heavy atoms, valence minus the number of hydrogens, the atomic number, the atomic mass, the atomic charge, the number of attached hydrogens, and a binary value indicating whether the atom is in a ring.

4. KERNEL COMPUTATION

4.1. Recursive Algorithm. In this section, we derive the recursive formula for computing the AE kernel without enumerating tree-patterns by following Mahé and Vert.[16] Let $G = (\mathcal{V}_G, \mathcal{E}_G)$ and $G' = (\mathcal{V}_{G'}, \mathcal{E}_{G'})$ be two graphs. We first define the set of subsets of neighborhood matching of vertices v and v' by

$$\begin{aligned} \mathcal{M}(v, v') = \{ & R \subseteq \mathcal{N}(v) \times \mathcal{N}(v') \\ & \mid (\forall\ (u, u'), (w, w') \in R\colon u \neq w \wedge u' \neq w') \\ & \wedge (\forall\ (u, u') \in R\colon (l(v, u) = l(v', u')))\} \end{aligned}$$

The AE kernel starts by comparing vertices pairwise in G and G' and then recursively compares their children h times

$$k_{\mathrm{AE},h}(G, G') = \sum_{v \in \mathcal{V}_G} \sum_{v' \in \mathcal{V}_{G'}} k_h(v, v') \tag{9}$$

where k_i, $i = 0,\ldots,h$, is defined as

$$k_i(v, v') = \begin{cases} \hat{w}(a_r(v))\hat{w}(a_r(v'))k_{\mathrm{atom}}(\mathbf{e}_r(v), \mathbf{e}_r(v')) & i = 0 \\ k_0(v, v')\left[1 + \sum_{R \in \mathcal{M}(v,v')} \prod_{(w,w') \in R} k_{i-1}(w, w')\right] & i = 1, \ldots, h \end{cases} \tag{10}$$

The derivation of this recursive formula is presented in the Appendix.

4.2. Complexity. Enumerating all possible matches $\mathcal{M}(v, v')$ of neighbors of vertices v and v' constitutes the main computational bottleneck of the AE kernel. This is due to the unordered nature of tree-patterns induced from molecular graphs. Let d be an upper bound on the out-degree of vertices in the molecular graphs considered herein. The number of operations to compute $k_i(v,v')$ in eq 10 is then bounded above by $\sum_{r=1}^{d} r({}_d\mathrm{P}_r)^2 = O(d^{2d})$ where ${}_d\mathrm{P}_r$ is the number of r-permutations of d. Thus, the worst-case complexity of the AE kernel of G and G' up to tree height h is

$$O(|\mathcal{V}_G| \cdot |\mathcal{V}_{G'}| \cdot h \cdot d^{2d}) \tag{11}$$

In the case of molecular graphs, the factor d^{2d} will be reduced significantly because most vertices have an out-degree of less than four, and the size of $\mathcal{M}$ decreases because of the mismatch in the continuous atom environment label and the edge label. The degree of mismatch between the continuous atom environment labels to be tolerated is controlled by the cutoff distance θ of the CS kernel (eq 5).

5. EXPERIMENTS

To demonstrate the effectiveness of the proposed kernel, we performed retrospective experiments using support vector machines (SVMs) on eleven classification tasks and one regression task. The data sets used herein are summarized in Table 2. The baseline methods to be compared are the subtree (ST) kernel initially proposed by Ramon and Gärtner,[1] the extended subtree (EST) kernel proposed by Mahé and Vert,[16] the Weisfeiler-Lehman subtree (WLST) kernel proposed by Shervashidze and Borgwardt,[20] the extended random walk (ERW) kernel proposed by Mahé et al.,[17] the optimal assignment (OA) kernel proposed by Fröhlich et al.,[22] and the extended-connectivity fingerprint (ECFP).[31] ECFPs are most commonly used in a wide variety of applications[31] in chemical informatics. We compared the effectiveness of the AE kernel and the baseline methods in terms of prediction performance and computational efficiency. We reported the area under the ROC curve (AUC) for classification and the squared correlation coefficient (R^2) between the observed and predicted values for regression using Monte Carlo cross-

Table 2. Basic Information of the Data Sets Used Herein[a]

	samples		
abbrev.	#pos.	#neg.	description
MUTAG	125	63	mutagenic effect on a bacterium
MM	129	207	carcinogenicity, male mice
FM	143	206	carcinogenicity, female mice
MR	152	192	carcinogenicity, male rats
FR	121	230	carcinogenicity, female rats
BBB	276	139	blood-brain barrier penetration
BIO	159	106	human oral bioavailability
BZR	157	149	benzodiazepine receptor ligands
COX2	148	155	cyclooxygenase-2 inhibitors
DHFR	124	269	dihydrofolate reductase inhibitors
ER	181	265	estrogen receptor ligands
SOL	1025		aqueous solubility

[a]#pos.: number of positive samples; #neg.: number of negative samples.

dx.doi.org/10.1021/ci400403w | J. Chem. Inf. Model. 2014, 54, 1289−1300

Table 3. Prediction Performance Comparison of the AE Kernel with the Standard Graph Kernels and Molecular Fingerprint on 12 Benchmarks[a]

data set	AE	ST	EST	WLST	ERW	OA	ECFP
MUTAG	**0.937 ± 0.063**	0.921 ± 0.060	0.924 ± 0.069	0.889 ± 0.127	0.927 ± 0.084	0.896 ± 0.078	0.896 ± 0.081
MM	0.688 ± 0.085	0.598 ± 0.094	0.684 ± 0.072	0.657 ± 0.091	**0.693 ± 0.098**	0.629 ± 0.089	0.636 ± 0.102
FM	**0.673 ± 0.091**	0.562 ± 0.108	0.640 ± 0.092	0.625 ± 0.072	0.658 ± 0.087	0.603 ± 0.070	0.597 ± 0.085
MR	0.672 ± 0.096	0.628 ± 0.105	0.654 ± 0.070	**0.691 ± 0.132**	0.638 ± 0.119	0.636 ± 0.091	0.573 ± 0.124
FR	**0.649 ± 0.096**	0.598 ± 0.117	0.640 ± 0.075	0.610 ± 0.102	0.602 ± 0.088	0.560 ± 0.104	0.545 ± 0.119
BBB	**0.834 ± 0.076**	0.761 ± 0.085	0.823 ± 0.096	0.802 ± 0.064	0.785 ± 0.082	0.744 ± 0.090	0.785 ± 0.088
BIO	**0.767 ± 0.099**	0.688 ± 0.097	0.669 ± 0.111	0.699 ± 0.110	0.675 ± 0.106	0.727 ± 0.112	0.716 ± 0.120
BZR	**0.831 ± 0.064**	0.781 ± 0.075	0.808 ± 0.078	0.766 ± 0.094	0.787 ± 0.091	0.768 ± 0.085	0.812 ± 0.075
COX2	**0.805 ± 0.087**	0.779 ± 0.106	0.788 ± 0.095	0.790 ± 0.096	0.793 ± 0.077	0.760 ± 0.080	0.799 ± 0.073
DHFR	**0.814 ± 0.080**	0.746 ± 0.091	0.798 ± 0.088	0.770 ± 0.086	0.793 ± 0.083	0.758 ± 0.119	0.799 ± 0.092
ER	**0.875 ± 0.050**	0.823 ± 0.067	0.836 ± 0.068	0.841 ± 0.052	0.834 ± 0.075	0.848 ± 0.053	0.855 ± 0.072
SOL	**0.905 ± 0.015**	0.893 ± 0.017	0.891 ± 0.014	0.892 ± 0.019	0.821 ± 0.045	0.875 ± 0.015	0.871 ± 0.024

[a]The areas under the curves (AUC) on 11 data sets (excluding the SOL data set) for classification and the squared correlation coefficients on the SOL data set for regression are shown. Values are expressed as mean value ± standard deviations. The best performance for each data set is shown in bold. The atom environment (AE) kernel is compared to the subtree (ST) kernel, the extended subtree (EST) kernel, the Weisfeiler−Lehman subtree (WLST) kernel, the extended random walk (ERW) kernel, the optimal assignment (OA) kernel, and the extended-connectivity fingerprint (ECFP).

Table 4. Parametrization of the AE Kernel with the Best Performance[a]

	parameters		
data set	tree-pattern	CS kernel	tree weight
MUTAG	$h = 4, r = 2$	$\gamma = 0.1, \theta = 1.40$	$\tau = -, \lambda_\alpha = 0.3, \lambda_\beta = 0.3$
MM	$h = 2, r = 2$	$\gamma = 0.1, \theta = 0.50$	$\tau = 2.7055, \lambda_\alpha = 0.8, \lambda_\beta = 0.4$
FM	$h = 2, r = 1$	$\gamma = 0.1, \theta = 0.50$	$\tau = 1.6424, \lambda_\alpha = 0.5, \lambda_\beta = 0.2$
MR	$h = 1, r = 1$	$\gamma = 0.5, \theta = 0.50$	$\tau = 2.0723, \lambda_\alpha = 0.7, \lambda_\beta = 0.4$
FR	$h = 4, r = 3$	$\gamma = 1.0, \theta = 0.80$	$\tau = -, \lambda_\alpha = 0.2, \lambda_\beta = 0.2$
BBB	$h = 0, r = 1$	$\gamma = 1.0, \theta = 0.20$	$\tau = 6.6349, \lambda_\alpha = 0.8, \lambda_\beta = 0.3$
BIO	$h = 0, r = 1$	$\gamma = 0.1, \theta = 0.05$	$\tau = 2.7055, \lambda_\alpha = 0.6, \lambda_\beta = 0.1$
BZR	$h = 3, r = 1$	$\gamma = 0.5, \theta = 0.05$	$\tau = -, \lambda_\alpha = 0.5, \lambda_\beta = 0.5$
COX2	$h = 0, r = 1$	$\gamma = 0.1, \theta = 1.20$	$\tau = 1.6424, \lambda_\alpha = 0.3, \lambda_\beta = 0.2$
DHFR	$h = 2, r = 1$	$\gamma = 0.1, \theta = 0.20$	$\tau = 3.8415, \lambda_\alpha = 0.4, \lambda_\beta = 0.2$
ER	$h = 4, r = 1$	$\gamma = 1.0, \theta = 1.00$	$\tau = 6.6349, \lambda_\alpha = 0.2, \lambda_\beta = 0.1$
SOL	$h = 1, r = 1$	$\gamma = 1.0, \theta = 1.30$	$\tau = -, \lambda_\alpha = 0.3, \lambda_\beta = 0.3$

[a]For each data set, the parametrization with the best performance is shown. For the tree-patterns, h is the tree height and r is the topological radius of the local environment around each atom. For the CS kernels, γ is the wide parameter of the Gaussian kernel and θ is the cutoff distance. Finally, for the tree weights, in the case of 11 data sets (excluding the SOL data set), τ is the threshold of the χ^2-statistic, and in the case of the SOL data set, τ is the threshold of the t-statistic and λ_α and λ_β are the tree weight factors.

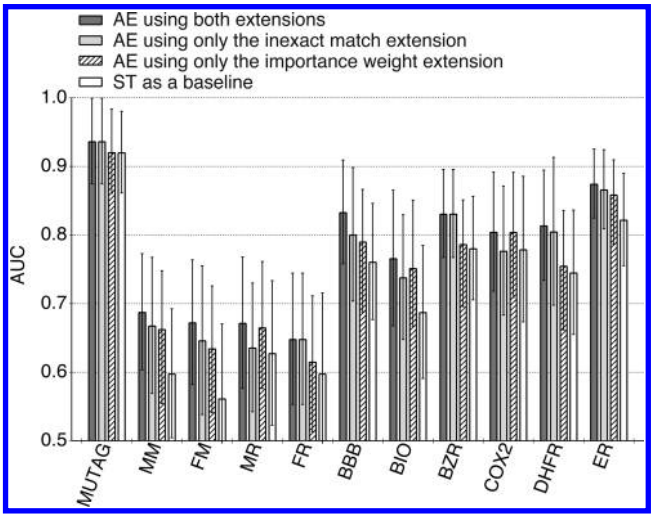


Figure 2. Contributions of the two extensions to the improvement of the classification performance for each data set. The best AUC values for each data set of the AE kernel using both extensions (dark shaded bars), the restricted AE kernel using only the inexact match extension (light shaded bars), the other restricted kernel using only the importance weight extension (hatched bars), and the subtree kernel as a baseline (open bars) are shown. Error bars indicate the standard deviation of the AUC.

validation (MCCV), in addition to the runtime required for the Gram matrix computation.

5.1. Experimental Settings. The following MCCV procedure was performed in all of the experiments:

1. The data set was randomly divided into a learning set $\mathcal{D}_L$ consisting of 90% of the data and a test set $\mathcal{D}_T$ consisting of the remaining 10%.
2. A prediction model based on an SVM with adjustable parameters was constructed to maximize the mean AUC for classification and the mean R^2 for regression over a 10-fold cross-validation on $\mathcal{D}_L$. Application of this model to the test set $\mathcal{D}_T$ yields the AUC for classification and the R^2 for regression.
3. In order to avoid erroneously high accuracy resulting from a lucky partition, the random division of the data into the sets $\mathcal{D}_L$ and $\mathcal{D}_T$ was repeated 20 times, and the mean and standard deviation of the performance metrics over the 20 iterations were evaluated.

We trained the SVMs with a regularization parameter C for classification and SVMs with an ε-insensitive loss function for regression using the LIBSVM implementation.[37] The parameters of the SVMs and the kernels were optimized using the 10-fold cross-validation in step 2. The regularization parameter C was chosen from $\{2^n | n \in \mathbb{N}, -10 \leqslant n \leqslant 14\}$ for SVM classification and regression. The loss function parameter ε for SVM regression was chosen from $\{0.1, 0.5, 1.0, \sigma/10, \sigma/5\}$, where σ is the standard deviation of the response values in $\mathcal{D}_L$. For the AE kernel, the best parameters were found by an exhaustive grid search over the following grid points: for the tree-patterns, the tree height $h \in \{0, 1, 2, 3, 4\}$ and the

dx.doi.org/10.1021/ci400403w | J. Chem. Inf. Model. 2014, 54, 1289−1300

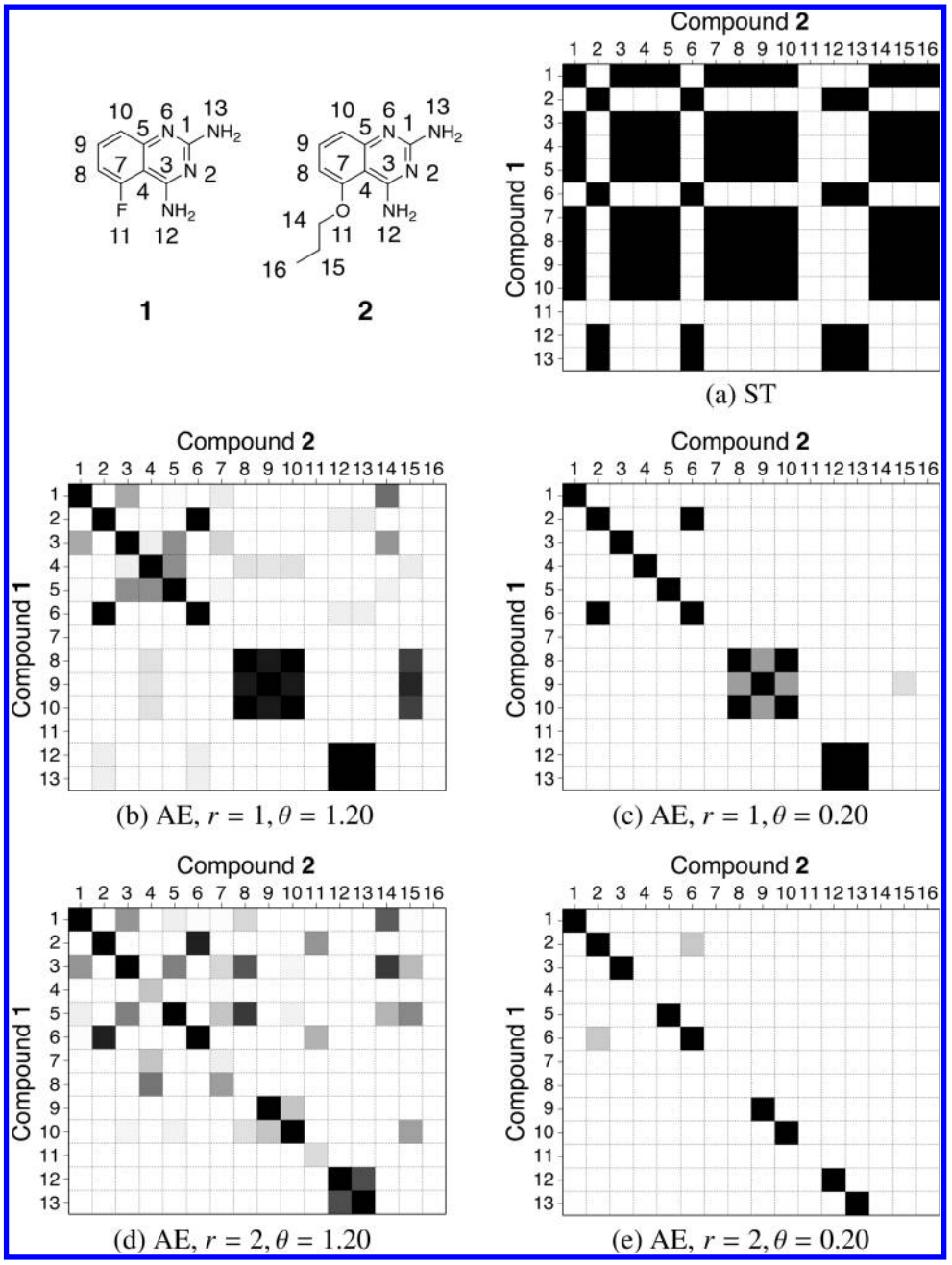


Figure 3. Pairwise atom similarity matrices between compounds **1** and **2** in the DHFR data set using (a) the ST kernel and (b−e) the AE kernels with varying topological radius r and cutoff distance θ, shaded from white to black to indicate increasing similarity. The width parameter γ of the Gaussian kernel is set to an optimized value of 0.1.

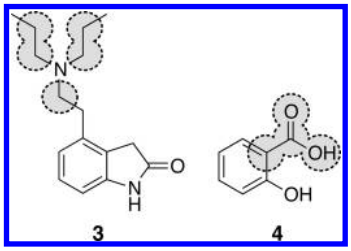


Figure 4. Examples of relevant atoms for the task of predicting the BBB penetration as determined from the χ^2 test. The relevant atoms are enclosed by broken lines. Compound **3** penetrates the BBB, but compound **4** does not. All of the kernel parameters are set to the optimized values shown in Table 4.

topological radius $r \in \{1, 2, 3\}$ of the local environment around each atom; for the CS kernels, the cutoff distance $\theta \in \{0.05, 0.10, 0.20, ..., 1.40\}$, the width parameter of the Gaussian kernel $\gamma \in \{0.1, 0.5, 1.0\}$, and the smoothing parameter $c = 0$; and for the tree weights in the classification task, the tree weight factors $(\lambda_\alpha, \lambda_\beta) \in \{(x,y) \in \{0.1, 0.2, ..., 0.9\}^2 | x \geqslant y\}$ and the χ^2 threshold $\tau \in \{1.3233, 1.6424, 2.0723, 2.7055, 3.8415, 6.6349\}$, the values of which correspond to the 25%, 20%, 15%, 10%, 5%, and 1% significance levels for the χ^2 distribution with one degree of freedom. In the case of the regression task, the Student's t-statistic was used to determine the tree weights at the same significance levels as the χ^2 thresholds. Each component of the atom environment label $\mathbf{e}_r(v)$ was standardized to 0 mean and unit variance within each learning set

dx.doi.org/10.1021/ci400403w | J. Chem. Inf. Model. 2014, 54, 1289−1300

Table 5. Computational Efficiency Comparison of the AE Kernel with the Standard Graph Kernels and Molecular Fingerprint on 12 Benchmarks

	statistics of the data sets[a]				runtime[b]						
data set	max. |G|	avg. |G|	avg. degree	#graphs	AE	ST	EST	WLST	ERW	OA	ECFP
MUTAG	28	17.9	2.2	188	″6	″6	″3	″2	″15	″33	″2
MM	64	14.0	2.0	336	″7	″6	″5	″3	″18	′1″15	″5
FM	64	14.1	2.1	349	″8	″7	″5	″3	″20	′1″21	″6
MR	64	14.3	2.1	344	″8	″7	″5	″3	″20	′1″20	″6
FR	64	14.6	2.1	351	″8	″8	″6	″3	″21	′1″25	″6
BBB	101	21.4	2.1	415	″27	″25	″16	″5	′1″23	′3″19	″12
BIO	36	21.0	2.1	265	″11	″11	″7	″3	″33	′1″18	″5
BZR	33	21.3	2.2	306	″17	″17	″10	″4	″52	′1″44	″7
COX2	36	26.3	2.2	303	″22	″22	″13	″5	′1″33	′2″27	″7
DHFR	39	23.9	2.2	393	″26	″29	″18	″6	′1″54	′3″20	″11
ER	43	21.3	2.2	446	″55	″52	″23	″6	′1″43	′3″48	″12
SOL	47	13.0	2.1	1025	′1″6	′1″2	″40	″9	′2′31	′9″59	″41

[a]max. |G|: maximum size of graphs; avg. |G|: average size of graphs; avg. degree: average degree of graphs; #graphs: number of graphs. [b]Average runtimes for the Gram matrix computation on 12 data sets over 10 runs using the atom environment (AE) kernel, the subtree (ST) kernel, the extended subtree (EST) kernel, the Weisfeiler–Lehman subtree (WLST) kernel, the extended random walk (ERW) kernel, the optimal assignment (OA) kernel, and the extended-connectivity fingerprint (ECFP). The single prime identifies minutes and the double prime indicates seconds.

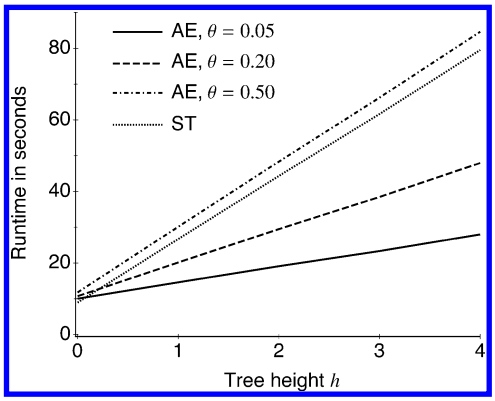


Figure 5. Average runtimes in seconds over 10 runs to compute the 1025 × 1025 Gram matrix on the SOL data set at different tree heights h. We compare the AE kernels with the cutoff distances $\theta = 0.05$ (solid line), 0.20 (dashed line), and 0.50 (dashed-dotted line) and the ST kernel (dotted line).

$\mathcal{D}_L$. In the case of the ST kernel, the tree weight function was given by $\mu(T) = \lambda^{2|T|}$. For the EST kernel, the kernel type was chosen from the set {size-based, branching–based, until-N branching-based}. The tree weight factor λ was chosen from {0.1, 0.2, ..., 0.9} for the ST and EST kernels. For the ST, EST, and WLST kernels, we varied the tree height as $h \in \{0, 1, 2, 3, 4\}$. It should be noted that the tree height follows from our definition. The termination probability for the ERW kernel was chosen from {0.01, 0.05, 0.1, 0.2, ..., 0.9}. For the EST and ERW kernels, the number of the Morgan index iterations was chosen from {1, 2, 3}. In the case of the ST, EST, WLST, and ERW kernels, each atom was labeled with the element type (e.g., carbon, oxygen, etc.) while each edge was labeled with the bond type (single, double, triple, or aromatic). The topological distance for the OA kernel was chosen from {1, 2, 3} and all other parameters were set to default values. For the ECFP, the maximum diameter was chosen from {4, 6}, and information relating to multiple occurrences of substructures was retained. It should be noted that the maximum diameter is essentially equal to twice the tree height number, h, of the tree-patterns.

In a similar manner[18] to the Tanimoto coefficient,[7] the kernels were all normalized as

$$\tilde{k}_{TA}(G, G') = \frac{k(G, G')}{k(G, G) + k(G', G') - k(G, G')}$$

The similarity between M dimensional fingerprints $\mathbf{X} = (x_i)$ and $\mathbf{Y} = (y_i)$ was measured using the MinMax kernel,[18] a variant of the Tanimoto coefficient

$$\tilde{k}_{MM}(\mathbf{X}, \mathbf{Y}) = \frac{\sum_{i=1}^{M} \min(x_i, y_i)}{\sum_{i=1}^{M} \max(x_i, y_i)}$$

Gram matrices $(\tilde{k}_{TA}(G_i, G_j))_{i,j}$ and $(\tilde{k}_{MM}(\mathbf{X}_i, \mathbf{X}_j))_{i,j}$ on each data set were then passed to the SVM solver of LIBSVM.

We measured the runtime of the Gram matrix computation on the 12 data sets to conduct an efficiency comparison of the AE kernel and the baseline methods. In order to perform a fair comparison, we fixed the maximum diameter of the subgraphs used to represent molecular structures to be six, which corresponds to a tree height of three. In the case of all of the tree-based kernels (AE, ST, EST, and WLST), we set the tree height to three. We used a topological distance of three for the OA kernel and a maximum diameter of six for the ECFP. The tree weight factor has less influence on the runtime and was set to 1.0 for the AE, ST, and EST kernels. Each of the other parameters was set to the most frequent value within the optimized values found on the 12 data sets. Specifically, we employed the following parameters: for the AE kernel, the topological radius $r = 1$, the cutoff distance $\theta = 0.5$, and the width parameter of the Gaussian kernel $\gamma = 0.1$; for the EST kernel (the until-N branching-based kernel), the number of the Morgan index iterations was set as 1; for the ERW kernel, a termination probability of 0.2 was used and the number of the Morgan index iterations was also 1.

The AE and ST kernels were implemented in C++ using the OpenBabel toolbox.[38] The EST and ERW kernels were computed using the ChemCpp toolbox.[39] We used a Matlab implementation[40] for the WLST kernel and a Java implementation[41] for the OA kernel. ECFPs were generated using the Pipeline Pilot software.[42] All our experiments were conducted on an Intel Xeon X5570 2.93 GHz system with 32GB of main memory.

5.2. Data Sets. The 12 data sets on mutagenicity, carcinogenicity, blood-brain barrier penetration, bioavailability, bioactivity, and aqueous solubility of chemical compounds, summarized in Table 2, are used. The aqueous solubility data set is a regression task.

In the mutagenesis data set[43] (MUTAG), the task of interest is to learn a classifier to predict whether each of the 188 aromatic and heteroaromatic nitro compounds is able to cause DNA to mutate. The Predictive Toxicology Challenge data set[44] contains compounds labeled according to carcinogenicity in rodents and is divided into male mice (MM), female mice (FM), male rats (MR), and female rats (FR). In the blood-brain barrier (BBB) data set,[45] the objective is to predict BBB penetration of a set of 415 compounds. The Yoshida data set[46] (BIO) classifies the 265 compounds according to their oral bioavailability. The Sutherland data set[47] deals with the binding activity of compounds at the benzodiazepine receptor (BZR), cyclooxygenase-2 (COX2), dihydrofolate reductase (DHFR), and estrogen receptor (ER). The aqueous solubility data set[48] (SOL) contains 1025 compounds with the aqueous solubility values in 20−25 °C expressed in log mol/L. For reasons of computational efficiency, all hydrogen atoms were removed from each compound.

5.3. Results and Discussion. In Table 3 a performance comparison of the AE kernel against the standard graph kernels (ST, EST, WLST, ERW, and OA) and molecular fingerprint (ECFP) for the 12 data sets is shown. It can be seen that the AE kernel outperforms the other methods on 9 of the 11 data sets for classification. The AE kernel achieved a mean AUC value of 0.777 over the 11 data sets, whereas ST, EST, WLST, ERW, OA, and ECFP demonstrated lower values of 0.717, 0.751, 0.740, 0.744, 0.721, and 0.728, respectively. These improvements are significant with p values of 9.8×10^{-4}, 9.8×10^{-4}, 2.9×10^{-3}, 2.0×10^{-3}, 9.8×10^{-4}, and 9.8×10^{-4} for ST, EST, WLST, ERW, OA, and ECFP, respectively, using a Wilcoxon paired two-sided test. The AE kernel performed best on the remaining data set for regression. The best parametrization of the AE kernel for each data set is shown in Table 4. It is worth mentioning that, with respect to AUC, the ECFP gives competitive performance compared to the other methods on the activity data sets (BZR, COX2, DHFR, and ER), which contain compounds with low structural diversity, but poor performance on the carcinogenicity data sets (MM, FM, MR, and FR), which contain compounds with high structural diversity. This is due to the circular substructures that are used for the ECFP to represent the molecular structures. The circular substructures are suitable to discriminate changes in functional groups between molecules with the same scaffold yet have difficulty capturing changes in molecular topology between molecules with different scaffolds. On the other hand, graph kernels based on walks and subtrees are able to capture them successfully.

In order to evaluate the individual contributions of the inexact match extension and the importance weight extension to the improvements seen, we compared the AE kernel, the two reduced variants of the AE kernel, and the ST kernel in terms of AUC (Figure 2). The variants are (i) the restricted AE kernel using only the inexact match extension where the restriction $\lambda_\alpha = \lambda_\beta$ is imposed in eq 8 and (ii) the other restricted AE kernel using only the importance weight extension where exact matching is used instead of k_{atom} in eq 8 for atoms labeled with element types. The figure reveals that, with many of the data sets, obvious improvements are observed through the combination of both of these extensions. We discuss the contribution of each extension in detail next.

One contribution to the improvements arises from the inexact match extension. Figure 3 shows the pairwise similarity matrices of atoms between compounds **1** and **2** in the DHFR data set using the ST kernel (Figure 3a) and the AE kernels (Figure 3b−e) with varying topological radius r and cutoff distance θ. The exact atom matching in the ST kernel causes redundant matches (Figure 3a), where paired atoms have the same element types but are located in different structural environments. In comparison, the inexact atom matching in the AE kernel eliminates such redundant matches by considering the local environment $\mathbf{e}_r(v)$ of each atom v while cutting off the similarity of atoms v and v' if the distance $\|\mathbf{e}_r(v) - \mathbf{e}_r(v')\|$ is larger than the cutoff distance θ (Figure 3b−e). Through comparison of parts b and c or d and e of Figure 3, we find that the decrease in θ reduces the number of nonzero elements in the similarity matrix. This implies that a large value of θ allows exchange between atoms of different elements. In the DHFR data set, the exchange occurs in 0.4% and 10.6% of all atom pairs at the cutoff distances $\theta = 0.20$ and 1.20, respectively. The matching behavior of atoms also depends on the topological radius r. Comparison of parts b and d or c and e of Figure 3 shows that the measurable similarity between the atoms is finer with increasing r. The graded similarity yields a reasonable assignment of atoms from one molecule to those of another by applying an appropriate cutoff distance θ. We note that inexact matching allows the inclusion of reduplicate assignments among atoms with similar local environments and the exclusion of undesirable assignments among atoms with different local environments. As a result, the inexact matching leads to the identification of pairs of chemically meaningful tree-patterns.

The other contribution to the improvements arises from the importance weight extension. In Figure 4 the examples of relevant atoms for prediction of the BBB penetration is shown. The hydrophobic regions of compound **3** and the carboxyl group of compound **4** were recognized as relevant to the task. This is in agreement with prior knowledge that polarity is inversely correlated with the BBB permeability, whereas hydrophobicity is directly correlated with the BBB permeability. It can be seen from Figure 2 that, on 8 out of the 11 data sets, additional increases in AUC, which correspond to the changes from light shaded to dark shaded bars, are obtained by applying the importance weight extension to the AE kernel using only the inexact matching extension. The AUC on the remaining data sets was almost unaffected by the importance weighting. A possible solution is to use different atom environment labels, which encode another substructural feature set (e.g., pharmacophore features). As a result, the importance weighting discloses relevant tree-patterns for the given tasks to the AE kernel, leading to improved performance.

Table 5 lists the runtimes to compute the Gram matrix for each data set. In terms of runtime, the AE kernel was competitive with the ST and EST kernels. In comparison, the WLST kernel outperformed the other methods over all data sets. On smaller data sets excluding the SOL data set, the ECFP was competitive with the WLST kernel but was approximately three times slower than the WLST kernel on the SOL data set. The ERW and OA kernels were slower than the other methods for all of the data sets. In Figure 5 the time taken to compute the 1025×1025 Gram matrix for the SOL data set at different tree heights, h, is shown for the AE kernels with the cutoff distances $\theta = 0.05$, 0.20, and 0.50 and the ST kernel. The

runtimes of both kernels grow linearly with respect to the tree height, h, which is consistent with the complexity given in eq 11, but the AE kernel is more efficient at lower cutoff distances (θ = 0.05 and 0.20). The decrease in θ shortens the runtime of the AE kernel; this is due to redundant matches of atoms being eliminated with decreasing θ, as shown in Figure 3.

6. CONCLUSIONS

In this paper, we tailored a new graph kernel to molecular structures by extending the subtree kernel. First, we permitted inexact tree-pattern matching, while eliminating redundant tree-pattern matches. As a result, the inexact match extension enhanced the identification of pairs of chemically meaningful tree-patterns in two molecular graphs. Second, we introduced the tree weight function to assign an importance weight to each tree-pattern according to the statistical significance for the task of interest. The importance weight extension alleviated the problem of the curse of dimensionality by decreasing the contribution of less significant tree-patterns for the task. As demonstrated, the combination of the two extensions successfully contributed to the improvement of performance for the classification and regression tasks of predicting various pharmaceutical properties. The AE kernel showed comparable or better prediction performance compared to the standard graph kernels and molecular fingerprint.

In future work, we intend to extend the proposed kernel. One possible extension is to allow matching between tree-patterns built on two root vertices and their descendants that contain gaps.[21] The flexible tree-pattern matching will capture new relevant aspects of molecular structures and progressively enrich the feature space induced by the resulting graph kernel. Chemically inspired extensions include matching between molecular fragments (referred to as bioisosteres[49]), which are structurally distinct yet biologically equivalent. It is necessary to condense molecular structures for the bioisostere matching, such that their pharmacophoric features are emphasized using a graph reduction method.[50,51] Another possible extension is to incorporate stereochemical information, such as chiral centers and cis−trans isomers, into the graph kernel.[52]

APPENDIX: DERIVATION OF THE RECURSIVE FORMULA

We derive the recursive form (eq 9) of the AE kernel (eq 4) using the recursive nature of tree construction. Let $G = (\mathcal{V}_G, \mathcal{E}_G)$ and $G' = (\mathcal{V}_{G'}, \mathcal{E}_{G'})$ be two molecular graphs. We first restrict $\mathcal{P}_T(G)$ to tree-patterns rooted at a specified vertex v, i.e.,

$$\mathcal{P}_T^{(v)}(G) = \{(v_{a_1}, ..., v_{a_{|T|}}) | (a_1, ..., a_{|T|}) \in \{1, ..., |\mathcal{V}_G|\}^{|T|} \wedge (v_{a_1}, ..., v_{a_{|T|}}) = \text{pattern}(T) \wedge v_{a_1} = v\}$$

With this set of tree-patterns, the AE kernel in eq 4 between G and G' with respect to any tree $T \in \mathcal{T}_h$ up to height h can be rewritten as

$$\begin{aligned} k_{\text{AE},h}(G, G') &= \sum_{T\in\mathcal{T}_h} \sum_{p\in\mathcal{P}_T(G)} \sum_{p'\in\mathcal{P}_T(G')} w(p)w(p')k_{\text{tree}}(p, p') \\ &= \sum_{v\in\mathcal{V}_G} \sum_{v'\in\mathcal{V}_{G'}} \left(\sum_{T\in\mathcal{T}_h} \sum_{p\in\mathcal{P}_T^{(v)}(G)} \sum_{p'\in\mathcal{P}_T^{(v')}(G')} w(p)w(p')k_{\text{tree}}(p, p') \right) \\ &= \sum_{v\in\mathcal{V}_G} \sum_{v'\in\mathcal{V}_{G'}} \left(\sum_{T\in\mathcal{T}_h} \sum_{p\in\mathcal{P}_T^{(v)}(G)} \sum_{p'\in\mathcal{P}_T^{(v')}(G')} \prod_{(u,u')\in\mathcal{A}(p,p')} \hat{w}(a_r(u))\hat{w}(a_r(u'))k_{\text{atom}}(\mathbf{e}_r(u), \mathbf{e}_r(u')) \right) \end{aligned}$$

The term in brackets in the above equation corresponds to $k_h(v,v')$ in eq 9, i.e.,

$$k_h(v, v') = \sum_{T\in\mathcal{T}_h} \sum_{p\in\mathcal{P}_T^{(v)}(G)} \sum_{p'\in\mathcal{P}_T^{(v')}(G')} \prod_{(u,u')\in\mathcal{A}(p,p')} \hat{w}(a_r(u))\hat{w}(a_r(u'))k_{\text{atom}}(\mathbf{e}_r(u), \mathbf{e}_r(u')) \quad (12)$$

For k_i, where $i = 0, ..., h$, k_0 is reduced to

$$k_0(v, v') = \hat{w}(a_r(v))\hat{w}(a_r(v'))k_{\text{atom}}(\mathbf{e}_r(v), \mathbf{e}_r(v')) \quad (13)$$

For k_i, with $i = 1, ..., h$, $\mathcal{A}(p, p')$, eq 12 always includes the pair (v,v') of root vertices as the first element. Taking the kernel value $k_0(v,v')$ out of the product in eq 12, we have

$$k_i(v, v') = k_0(v, v') \times \left(\sum_{T\in T_i} \sum_{p\in P_T^{(v)}(G)} \sum_{p'\in P_T^{(v')}(G')} \prod_{(u,u')\in A(p,p')\setminus(u_{a_1},u'_{a_1})} k_0(u, u') \right) \quad (14)$$

In the above equation, all pairs of children of the root vertices v and v' appear in the first element of $\mathcal{A}(p, p')\setminus(u_{a_1}, u'_{a_1})$. In other words, the term in brackets in eq 14 compares all pairs of downstream tree-patterns from the root vertices v and v' with respect to $T \in \mathcal{T}_{i-1}$. Thus, eq 14 becomes

$$\begin{aligned} k_i(v, v') &= k_0(v, v') \left[\sum_{R\in M(v,v')+\varnothing} \prod_{(w,w')\in R} \left(\sum_{T\in T_{i-1}} \sum_{p\in P_T^{(w)}(G)} \sum_{p'\in P_T^{(w')}(G')} \prod_{(u,u')\in A(p,p')} k_0(u, u') \right) \right] \quad (15a) \\ &= k_0(v, v') \left[1 + \sum_{R\in M(v,v')} \prod_{(w,w')\in R} \left(\sum_{T\in T_{i-1}} \sum_{p\in P_T^{(w)}(G)} \sum_{p'\in P_T^{(w')}(G')} \prod_{(u,u')\in A(p,p')} k_0(u, u') \right) \right] \quad (15b) \end{aligned}$$

In eq 15a, we take the empty set $\varnothing$ as a special case out of $\mathcal{M}(v, v')$. The product $\prod_{(w,w')\in R}$ is 1 if $R = \varnothing$, in order to treat

dx.doi.org/10.1021/ci400403w | J. Chem. Inf. Model. 2014, 54, 1289−1300

unbalanced trees in the AE kernel. On the other hand, under the convention that the product is 0 if $R = \varnothing$, the AE kernel treats only balanced trees. It is straightforward to obtain eq 15b from eq 15a for the unbalanced trees. The term in parentheses in eq 15b corresponds to $k_{i-1}(w,w')$ in eq 10. With eq 13 for $i = 0$ and eq 15b for $i > 0$, the derivation of the recursive formula is complete.

AUTHOR INFORMATION

Corresponding Author

*E-mail: yoshidar@ism.ac.jp.

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

We would like to thank Takashi Washio, Yukito Iba, and Tetsu Isomura for their enlightening discussions and valuable suggestions and Hideo Kubodera, Masataka Kuroda, and Takanori Ohgaru for their insightful discussions on the chemical aspects of the problems considered herein.

REFERENCES

(1) Ramon, J.; Gärtner, T. Expressivity versus efficiency of graph kernels. In *Proceedings of the 1st International Workshop on Mining Graphs, Trees and Sequences (MTGS 2003)* [Online], Cavtat-Dubrovnik, Croatia, September 22−23, 2003; Washio, T., De Raedt, L., Eds.; University of Osaka, Institute for Scientific and Industrial Research Web site. http://www.ar.sanken.osaka-u.ac.jp/MGTS-2003CFP.html (accessed October 1, 2013).

(2) Hansch, C.; Maloney, P. P.; Fujita, T. Correlation of biological activity of phenoxyacetic acids with hammett substituent constants and partition coefficients. *Nature* **1962**, *194*, 178−180.

(3) Kubinyi, H. Drug research: myths, hype and reality. *Nat. Rev. Drug Discovery* **2003**, *2*, 665−668.

(4) *Concepts and Applications of Molecular Similarity*; Johnson, M. A., Maggiora, G. M., Eds.; Wiley: New York, 1990.

(5) Trinajstić, N. *Chemical Graph Theory*, 2nd ed.; CRC Press: Boca Raton, FL, 1992.

(6) Todeschini, R.; Consonni, V. *Molecular Descriptors for Chemo-informatics* (2 volumes); Wiley-VCH: Weinheim, 2009.

(7) Rogers, D. J.; Tanimoto, T. T. A computer program for classifying plants. *Science* **1960**, *132*, 1115−1118.

(8) Schölkopf, B.; Smola, A. J. *Learning with Kernels*; MIT Press: Cambridge, MA, 2002.

(9) Shawe-Taylor, J.; Cristianini, N. *Kernel Methods for Pattern Analysis*; Cambridge University Press: Cambridge, UK, 2004.

(10) Haussler, D. *Convolution kernels on discrete structures*; Technical Report UCSC-CRL-99-10; University of California Santa Cruz: 1999.

(11) Vishwanathan, S. V. N.; Schraudolph, N. N.; Kondor, R.; Borgwardt, K. M. Graph kernels. *J. Mach. Learn. Res.* **2010**, *99*, 1201−1242.

(12) Gärtner, T.; Flach, P.; Wrobel, S. On graph kernels: Hardness results and efficient alternatives. In *Proceedings of the 16th Annual Conference on Computational Learning Theory and 7th Kernel Workshop (COLT/Kernel 2003)*, Washington, DC, U.S.A., August 24−27, 2003; Schölkopf, B., Warmuth, M. K., Eds.; Springer: Berlin, Germany, 2003; pp 129−143.

(13) Kashima, H.; Tsuda, K.; Inokuchi, A. Marginalized kernels between labeled graphs. In *Proceedings of the 20th International Conference on Machine Learning (ICML 2003)*, Washington, DC, U.S.A., August 21−24, 2003; Fawcett, T., Mishra, N., Eds.; AAAI Press: Chicago, IL, U.S.A., 2003; pp 321−328.

(14) Borgwardt, K. M.; Kriegel, H.-P. Shortest-path kernels on graphs. In *Proceedings of the 5th IEEE International Conference on Data Mining (ICDM 2005)*, Washington, DC, U.S.A., November 27−30, 2005; IEEE Computer Society Press: Washington, DC, U.S.A., 2005; pp 74−81.

(15) Horváth, T.; Gärtner, T.; Wrobel, S. Cyclic pattern kernels for predictive graph mining. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2004)*, Seattle, WA, U.S.A., August 22−25, 2004; ACM Press: New York, NY, U.S.A., 2004; pp 158−167.

(16) Mahé, P.; Vert, J.-P. Graph kernels based on tree patterns for molecules. *Mach. Learn.* **2009**, *75*, 3−35.

(17) Mahé, P.; Ueda, N.; Akutsu, T.; Perret, J.-L.; Vert, J.-P. Extensions of marginalized graph kernels. In *Proceedings of the 21st International Conference on Machine Learning (ICML 2004)*, Banff, Canada, July 4−8, 2004; Greiner, R., Schuurmans, D., Eds.; ACM Press: New York, U.S.A., 2004; pp 552−559.

(18) Ralaivola, L.; Swamidass, S. J.; Saigo, H.; Baldi, P. Graph kernels for chemical informatics. *Neural Networks* **2005**, *18*, 1093−1110.

(19) Vishwanathan, S. V. N.; Borgwardt, K. M.; Schraudolph, N. N. Fast computation of graph kernels. In *Proceedings of the 2006 Conference on Advances in Neural Information Processing Systems 19 (NIPS 2006)*, Vancouver, British Columbia, Canada, December 4−7, 2006; Schölkopf, B., Platt, J., Hoffman, T., Eds.; MIT Press: Cambridge, MA, U.S.A., 2007; pp 131−138.

(20) Shervashidze, N.; Borgwardt, K. M. Fast subtree kernels on graphs. In *Proceedings of the 2009 Conference on Advances in Neural Information Processing Systems 22 (NIPS 2009)*, Vancouver, British Columbia, Canada, December 7−10, 2009; Bengio, Y., Schuurmans, D., Lafferty, J., Williams, C. K. I., Culotta, A., Eds.; MIT Press: Cambridge, MA, U.S.A., 2010; pp 1660−1668.

(21) Kashima, H.; Koyanagi, T. Kernels for semi-structured data. In *Proceedings of the 19th International Conference on Machine Learning (ICML 2002)*, San Francisco, CA, U.S.A., July 8−12, 2002; Sammut, C., Hoffmann, A. G., Eds.; Morgan Kaufmann: 2002; pp 291−298.

(22) Fröhlich, H.; Wegner, J. K.; Sieker, F.; Zell, A. Optimal assignment kernels for attributed molecular graphs. In *Proceedings of the 22nd International Conference on Machine Learning (ICML 2005)*, Bonn, Germany, August 7−11, 2005; de Raedt, L., Wrobel, S., Eds.; Omnipress: Madison, WI, U.S.A., 2005; pp 225−232.

(23) Ben-David, S.; Eiron, N.; Simon, H. U.; Long, M. Limitations of learning via embeddings in Euclidean half spaces. *J. Mach. Learn. Res.* **2002**, *3*, 441−461.

(24) Collins, M.; Duffy, N. Convolution kernels for natural language. In *Proceedings of the 2001 Neural Information Processing Systems Conference (NIPS 2001)*, Vancouver, British Columbia, Canada, December 3−8, 2001; Dietterich, T. G., Becker, S., Ghahramani, Z., Eds.; MIT Press: Cambridge, MA, U.S.A., 2002; pp 625−632.

(25) Cumby, C.; Roth, D. On kernel methods for relational learning. In *Proceedings of the 20th International Conference on Machine Learning (ICML 2003)*, Washington, DC, U.S.A., August 21−24, 2003; Fawcett, T., Mishra, N., Eds.; AAAI Press: Chicago, IL, U.S.A., 2003; pp 107−114.

(26) Suzuki, J.; Isozaki, H.; Maeda, E. Convolution kernels with feature selection for natural language processing tasks. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL 2004)* [Online], Barcelona, Spain, July 21−26, 2004; Scott, D., Daelemans, W., Walker, M. A., Eds.; ACL Web site. http://acl.ldc.upenn.edu/acl2004/main/index.html (accessed October 1, 2013).

(27) Frasconi, P.; Passerini, A.; Muggleton, S.; Lodhi, H. *Declarative kernels*; Technical Report RT 2/2004; Dipartimento di Sistemi e Informatica, Università di Firenze: 2004.

(28) Menchetti, S.; Costa, F.; Frasconi, P. Weighted decomposition kernels. In *Proceedings of the 22nd International Conference on Machine Learning (ICML 2005)*, Bonn, Germany, August 7−11; ACM: New York, NY, U.S.A., 2005; pp 585−592.

(29) Sheridan, R. P. The most common chemical replacements in drug-like compounds. *J. Chem. Inf. Comput. Sci.* **2002**, *42*, 103−108.

(30) Burden, F. R. Molecular identification number for substructure searches. *J. Chem. Inf. Comput. Sci.* **1989**, *29*, 225−227.

(31) Rogers, D.; Hahn, M. Extended-connectivity fingerprints. *J. Chem. Inf. Model.* **2010**, *50*, 742−754.

(32) Morgan, H. L. The generation of a unique machine description for chemical structures: A technique developed at Chemical Abstract Services. *J. Chem. Doc.* **1965**, *5*, 107−113.
(33) Wendland, H. Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. *Adv. Comput. Math.* **1995**, *4*, 389−396.
(34) Gneiting, T. Compactly supported correlation functions. *J. Multivariate Anal.* **2002**, *83*, 493−508.
(35) Gneiting, T. Correlation functions for atmospheric data analysis. *Q. J. R. Meteor. Soc.* **1999**, *125*, 2449−2464.
(36) Berg, C.; Christensen, J. P.; Ressel, P. *Harmonic analysis on semigroups*; Springer-Verlag: New York, 1984.
(37) Chang, C.-C.; Lin, C.-J. LIBSVM: a library for support vector machines. http://www.csie.ntu.edu.tw/~cjlin/libsvm/ (accessed October 1, 2013).
(38) Open Babel: The Open Source Chemistry Toolbox. http://openbabel.org (accessed October 1, 2013).
(39) Perret, J.-L.; Mahé, P.; Vert, J.-P. ChemCpp: an open source C++ toolbox for kernel functions on chemical compounds. http://chemcpp.sourceforge.net (accessed October 1, 2013).
(40) Weisfeiler-Lehman Graph Kernel: a Matlab implementation of the Weisfeiler-Lehman graph kernel. http://mlcb.is.tuebingen.mpg.de/Mitarbeiter/Nino/WL/ (accessed October 1, 2013).
(41) Optimal Assignment Kernel: a Java implementation of the optimal assignment kernel. http://www.ra.cs.uni-tuebingen.de/software/OAKernels/ (accessed October 1, 2013).
(42) *Pipeline Pilot*, version 7.5; Accelrys, Inc.: San Diego, CA, 2008.
(43) Srinivasan, A.; Muggleton, S. H.; Sternberg, M. J. E.; King, R. D. Theories for mutagenicity: A study in first-order and feature-based induction. *Artif. Intell.* **1996**, *85*, 277−299.
(44) Helma, C.; Kramer, S. A survey of the predictive toxicology challenge 2000−2001. *Bioinformatics* **2003**, *19*, 1179−1182.
(45) Li, H.; Yap, C. W.; Ung, C. Y.; Xue, Y.; Cao, Z. W.; Chen, Y. Z. Effect of selection of molecular descriptors on the prediction of blood-brain barrier penetrating and nonpenetrating agents by statistical learning methods. *J. Chem. Inf. Model.* **2005**, *45*, 1376−1384.
(46) Yoshida, F.; Topliss, J. G. QSAR Model for Drug Human Oral Bioavailability. *J. Med. Chem.* **2000**, *43*, 2575−2585.
(47) 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.
(48) Huuskonen, J. Estimation of aqueous solubility for a diverse set of organic compounds based on molecular topology. *J. Chem. Inf. Model.* **2000**, *40*, 773−777.
(49) Burger, A. Isosterism and bioisosterism in drug design. *Prog. Drug Res.* **1991**, *37*, 287−371.
(50) Stiefl, N.; Watson, I. A.; Baumann, K.; Zaliani, A. ErG: 2D pharmacophore descriptions for scaffold hopping. *J. Chem. Inf. Model.* **2006**, *46*, 208−220.
(51) Birchall, K.; Gillet, V. J.; Willett, P.; Ducrot, P.; Luttmann, C. Use of reduced graphs to encode bioisosterism for similarity-based virtual screening. *J. Chem. Inf. Model.* **2009**, *49*, 1330−1346.
(52) Brown, J.; Urata, T.; Tamura, T.; Arai, M.; Kawabata, T.; Akutsu, T. Compound analysis via graph kernels incorporating chirality. *J. Bioinform. Comput. Biol.* **2010**, *8*, 63−81.