

Chemical structure modeling with kernel methods

Hiroshi Yamashita

DOCTOR OF PHILOSOPHY

Department of Statistical Science
School of Multidisciplinary Sciences
The Graduate University for Advanced Studies

2014

Abstract

Small molecules that have the ability to alter the biological response of a cell or disease state are of significant interest as drug candidates. One approach to the design of such molecules using kernel methods is through a map of molecules to a feature space induced by a kernel, where the predictions of the molecular properties are made in a linear manner. The forward mapping helps guide the synthesis of new molecules. Another more direct approach is to solve the so-called pre-image problem, i.e., to reconstruct a corresponding molecule (aka pre-image) from its feature space representation. The inverse mapping is of central importance for the design of new molecules with desired properties. In this thesis, we address two problems in drug design: (I) the forward problem of predicting molecular properties, where we propose a new graph kernel that induces a feature space amenable to the prediction of molecular properties, and (II) the inverse problem of designing new molecules that possess properties required for drug candidates, where we develop a population-based Monte Carlo method to solve the pre-image problem for the molecules. Our respective contributions to these problems are summarized as:

(I) Forward problem of predicting molecular properties

The measurement of molecular similarity is an essential part of predicting molecular properties. Graph kernels provide good similarity measures between molecules. A conventional graph kernel is based on counting common subgraphs of a specific type in molecular graphs. This approach suffers from 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 these 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 lead to promising improvements in classification and regression tasks for predicting a wide range of pharmaceutical properties from the chemical structure of molecules.

(II) Inverse problem of designing new molecules

The *de novo* design of new molecules that yield desired properties has the potential to substantially reduce both the time and cost involved in drug development. Recent developments in graph kernels have enabled us to apply well-established machine learning techniques to molecular data which are internally represented as graphs. However, in order to allow for the design that generates new molecules as a result, it is necessary to solve the pre-image problem for molecules. Unlike the traditional method proposed in Bakır et al. (2004), which is formulated as a nonlinear combinatorial optimization problem, we express the pre-image problem as a sampling problem for molecular graphs. Here, we are not only interested in optimal molecules, but also in near-optimal molecules which are often considered to be good candidates for further chemical synthesis. Therefore, we develop a population-based Monte Carlo method for sampling structurally diversified molecules near the pre-images, which possess good drug-likeness. The key to an efficient sampling method is to use the update of a population by evolutionary operators for the structural alteration of molecules. Furthermore, to penalize non-drug-like molecules, we use the knowledge of drug-likeness commonly considered by medicinal chemists. The effectiveness of the proposed method is

illustrated through experiments to find corresponding molecules from given image points in a feature space induced by a graph kernel.

Acknowledgements

Many thanks go to my thesis supervisor, Ryo Yoshida, and sub-supervisor, Tomoyuki Higuchi, for introducing me to the world of machine learning. Ryo has helped me in innumerable ways. I am also very grateful to the members of the Research and Development Center for Data Assimilation.

It goes without saying that my thesis referees, Yoshihiro Yamanishi, Tomoyuki Higuchi, Yukito Iba, Kenji Fukumizu, and Ryo Yoshida have reviewed this thesis and given me valuable comments to improve upon its content.

Of course, fruitful research is only possible in a friendly and pleasant working environment. Takashi Washio and Yukito Iba—thanks for our scientific discussions. Okimasa Okada and Tetsu Isomura—thanks for all our discussions. Though I could easily continue this list of colleagues, I wish to thank Masataka Kuroda and Takanori Ohgaru who work in the chemoinformatics group at Mitsubishi Tanabe Pharma Corporation.

Contents

Contents	v
1 Introduction	1
1.1 Background	1
1.1.1 Forward Problem of Predicting Molecular Properties	3
1.1.2 Inverse Problem of Designing New Molecules	6
1.2 Outline of the Thesis	8
2 Kernel Methods for Molecules	10
2.1 Kernel Methods	10
2.2 Representing Chemical Structures	13
2.3 Graph Kernels	15
2.3.1 Convolution Kernels	15
2.3.2 Random Walk Kernels	17
2.3.3 Subtree Kernels	18
2.3.4 Other Graph Kernels	20
3 Atom Environment Kernels for the Forward Problem	22
3.1 Basic Idea	22
3.2 Inexact Match Extension	23
3.3 Importance Weight Extension	25

Contents	vi
3.4 Relation to Previous Research	28
3.5 Atom Environment Labels	29
3.5.1 Continuous Labels	29
3.5.2 Discrete Labels	30
3.6 Kernel Computation	31
3.6.1 Recursive Algorithm	31
3.6.2 Complexity	32
3.7 Experiments	32
3.7.1 Experimental Settings	33
3.7.2 Data Sets	36
3.7.3 Results and Discussion	37
3.8 Concluding Remarks	47
4 Fragment Assembly Monte Carlo Methods for the Inverse Problem	49
4.1 Basic Idea	49
4.2 Evolutionary Movements in Chemical Space	53
4.3 Preparation of Molecular Fragments	57
4.4 Regularization of Molecules	58
4.5 Experiments	59
4.5.1 Experimental Settings	59
4.5.2 Results and Discussion	61
4.6 Concluding Remarks	65
5 Conclusions	67
Appendix A Derivation of the Recursive Formula	70
List of Figures	73

Contents	vii
List of Tables	78
Symbols	82
References	83
List of Publications	92

Chapter 1

Introduction

1.1 Background

A primary goal in drug development is to identify new molecules that possess suitable properties required for drug candidates. It is estimated^{3–5} that there are in excess of 10^{60} organic molecules below 500 Da of possible interest for drug development. A comprehensive synthesis of molecules is not a feasible strategy due to this vast chemical space. Therefore, computer-aided molecular design (CAMD) has the potential to substantially reduce both the time and cost involved in the trial-and-error experiments of drug development. The use of quantitative structure–activity relationships, known as QSARs,^{6,7} is a promising approach to CAMD. The first step in this approach is to build a forward QSAR model for predicting biological activities or molecular properties from the structural information of a molecule. A forward QSAR can be established using many different model equations (e.g., multiple linear regression, partial least squares, support vector regression, etc.). Once a model is built, we next invert the QSAR, i.e., now searching for molecules of interest under the model. This is referred to as the inverse problem.

The most commonly used solution to the inverse problem is virtual screening. In this approach, molecules in a database are evaluated using a forward QSAR to identify molecules

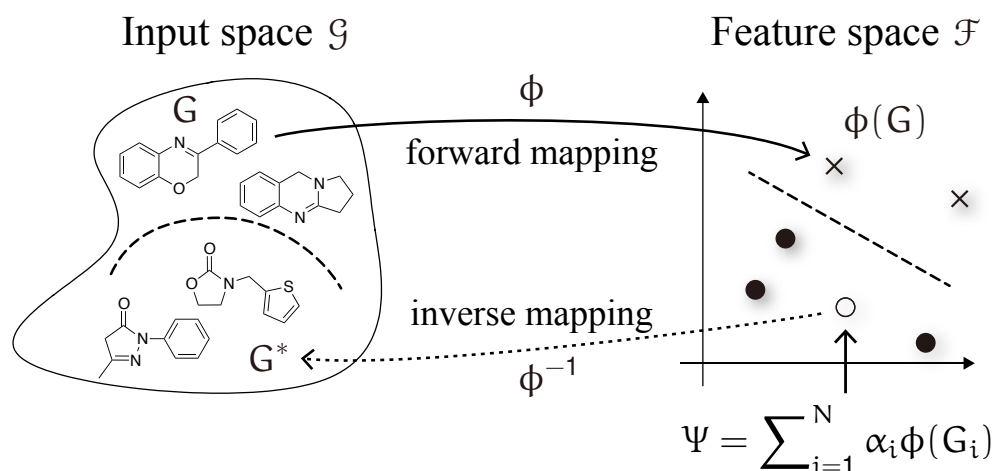


Figure 1.1 Illustration of the pre-image problem in kernel methods. An image point Ψ in the feature space \mathcal{F} is mapped back to the input space \mathcal{G} .

possessing desired properties. However, virtual screening can only identify molecules present in the database, i.e., it cannot suggest new structural molecules that yield better properties than known molecules. *De novo* design solves this problem by building molecules from scratch so as to optimize a scoring function. While the scoring is based on a QSAR model, other factors such as drug-likeness and synthetic accessibility can be used as well. Methods for designing new molecules include graphical enumeration^{8–10} and stochastic optimization^{11–16} in the context of inverse QSAR. The exhaustive enumeration of chemical structures with given constraints is often computationally demanding. In addition, at present, this approach can generate only treelike (acyclic) molecular graphs. A stochastic approach such as simulated annealing, genetic algorithms, or tabu search solves a nonlinear combinatorial optimization problem for chemical structures, where one may be interested in only optimal solutions of the problem.

Kernel methods^{17–20} provide a principle framework for solving the forward and inverse problems in CAMD. The recent development of graph kernel functions²¹ has made it possible to apply kernel methods to various machine learning tasks in chemical informatics,²² where graphs are used to describe the chemical structure of the molecules.²³ Once a suitable kernel function k on a set \mathcal{G} of molecular graphs is defined, the kernel approach to molecules

works successfully for well-established machine learning methods (e.g., support vector machine, logistic regression, K-means clustering, etc.). The basic idea behind the kernel approach is to implicitly map molecular graphs in the input space \mathcal{G} into the high dimensional feature space \mathcal{F} via a possible nonlinear map $\phi : \mathcal{G} \rightarrow \mathcal{F}$ such that for every $G, G' \in \mathcal{G}$ it holds that $k(G, G') = \langle \phi(G), \phi(G') \rangle$ where $\langle \cdot, \cdot \rangle$ denotes the inner product. The forward mapping ϕ is of primary importance in predicting the molecular properties.^{24–26} We next invert the forward mapping ϕ , i.e., the inverse mapping ϕ^{-1} from the feature space \mathcal{F} back to the input space \mathcal{G} . This is known as the pre-image problem. The pre-image problem is of central importance for the design of new molecules with desired properties.^{2,27} Many graph kernels²¹ for solving the forward problem exist, yet solutions^{2,28} to the inverse problem are relatively limited due to its ill-posed nature; it is non-convex, nonlinear, and combinatorial. We describe kernel methods to address the forward and inverse problems in detail below.

1.1.1 Forward Problem of Predicting Molecular Properties

The definition of an appropriate similarity function between molecules is of crucial importance for many applications in chemical informatics. Common applications include QSAR model construction to predict biological activities from structural information of molecules. The quantitative structure-activity relationship models rely on the similarity property principle,²⁹ which states that structurally similar molecules tend to have similar properties. Therefore, the QSAR 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 chemical structure of molecules.²³ 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,³⁰ which can be thought of as numerical representations that are encoded so as to capture the relevant aspects of structural information of molecules. A unique dictionary³⁰ of molecular descriptors lists more than 3,300 descriptors. Popular choices for them include extended-connectivity fingerprints³¹ (ECFPs). The similarity between the molecular descriptors is then measured by a similarity metric, e.g., the Tanimoto coefficient.³² 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 of 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^{19,20} k on a set \mathcal{G} of molecular graphs. The symmetric function $k: \mathcal{G} \times \mathcal{G} \rightarrow \mathbb{R}$ is said to be a positive definite kernel on \mathcal{G} if and only if $\sum_{i,j \in \{1, \dots, n\}} c_i c_j k(G_i, G_j) \geq 0$ for all $n \in \mathbb{N}$, $G_1, \dots, G_n \in \mathcal{G}$, and $c_1, \dots, c_n \in \mathbb{R}$. For such k , it is known that a map $\phi: \mathcal{G} \rightarrow \mathcal{F}$ into a reproducing kernel Hilbert space (RKHS) \mathcal{F} 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 $\phi(G) = k(\cdot, G) \in \mathcal{F}$ of a kernel function k is of substantially the same class as the feature vector of a molecular descriptor. A difference is whether the feature space is defined explicitly or implicitly. The convolution kernel³³ 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..²¹ These graph kernels differ with respect to the choice of the subgraph types used to represent the structured objects, such as walks,^{24,34} shortest paths,³⁵ cycles,³⁶ and trees.^{1,26} Mahé et al.³⁷ introduced

two extensions to remove tottering walks and to increase the number of different atom labels using the Morgan algorithm. Ralaivola et al. introduced three normalized variants³⁸ (Tanimoto, MinMax, and Hybrid) of the non-tottering walk kernels. Subsequently, the efficient computation schemes for the random walk kernel³⁹ and the subtree kernel⁴⁰ 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 kernel⁴¹ 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.⁴² 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.⁴³ Possible solutions to this problem include decreasing the contribution of larger subgraphs,⁴⁴ using prior knowledge to select relevant subgraphs,^{45–47} and increasing the specificity of matching between subgraphs based on consideration of neighborhood information.^{42,48}

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 extension⁴⁹ of the Burden approach^{50,51} and a variant³¹ of the Morgan algorithm.⁵² The AE kernel tol-

erates 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.

1.1.2 Inverse Problem of Designing New Molecules

In this section we consider the inverse problem of reconstructing a corresponding molecular graph from its feature space representation induced by a graph kernel. This is known as the pre-image problem.

Let k be a kernel function on a set \mathcal{G} of molecular graphs. The kernel k induces an RKHS \mathcal{F} , called the feature space, and a map $\phi : \mathcal{G} \rightarrow \mathcal{F}$ such that $k(G, G') = \langle \phi(G), \phi(G') \rangle$ for all $G, G' \in \mathcal{G}$. Given an image point Ψ in \mathcal{F} as an expansion in terms of known molecules $\{G_1, \dots, G_N\} \subseteq \mathcal{G}$, i.e., $\Psi = \sum_{i=1}^N \alpha_i \phi(G_i)$, the pre-image problem amounts to finding a corresponding molecular graph $G^* \in \mathcal{G}$ such that $\Psi = \phi(G^*)$. However, the map ϕ is not






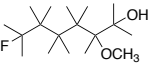
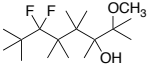
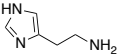
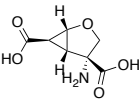
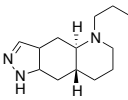
constraints for molecular generation	examples			# molecules
$C_n H_{2n+2}$ $n \leq 13$				799
 $R \in \{150 \text{ functional groups}\}$ $\# \text{ substitutions} \leq 14$				10^{29}
$\text{atom} \in \{C, O, N, S\}$ $\# \text{ atoms} \leq 30$				10^{60} (estimated number)

Table 1.1 The Number of Molecules with Given Structural Constraints.

surjective in general. In this case, it is natural to find an approximate pre-image G^* such that

$$G^* = \arg \min_G \|\Psi - \phi(G)\|^2. \quad (1.1)$$

This is a hard combinatorial optimization problem since there are at least 10^{60} possible organic molecules^{3–5} (see Table 1.1). A general learning-based framework for finding pre-images is reported in ref 27.

Methods to solve the pre-image problem for molecules include combinatorial enumeration²⁸ and stochastic optimization.² Fujiwara et al.²⁸ proposed an enumeration algorithm for treelike chemical structures with given path frequencies using the branch-and-bound method. Enumeration of chemical structures with given constraints is often computationally prohibitive.²⁸ In addition, at present, this approach can generate only treelike (acyclic) chemical structures. Bakır et al.² proposed a stochastic optimization algorithm for finding a corresponding chemical structure from a given image point in \mathcal{F} induced by the random walk kernel.²⁴ The stochastic optimization approach suffers from local minimum trapping, requiring restarts with a new initial guess, and therefore may miss potentially important molecules. Moreover, this approach delivers only local optimal solutions (molecules) of the problem.

In CAMD, medicinal chemists are not only interested in optimal solutions, but also in their neighboring suboptimal solutions. This means that molecules near the optimal solutions are often considered as good candidates for further chemical synthesis since the desirability (e.g., potency, stability, synthesizability, drug-likeness, etc.) of the optimal solutions is usually insufficient. Therefore, the pre-image problem can be expressed as the problem of generating molecular graphs from a target distribution. This is different from the optimization problem (eq 1.1) where only local optimal solutions are of interest. The formulation of the sampling problem begins by defining a target distribution on the molecular graphs as a

Boltzmann distribution at temperature t

$$G^* \sim \pi(G) \propto \exp\{-\left(\|\Psi - \phi(G)\|^2 + \eta R(G)\right)/t\},$$

where $R(G)$ is a regularization function to penalize non-drug-like molecules and η controls the strength of the regularization. In Chapter 4, we will draw molecular graphs from $\pi(G)$ using a population-based Monte Carlo method.^{53–55}

1.2 Outline of the Thesis

This thesis is organized into three remaining chapters, followed by a conclusion.

Chapter 2 discusses graph kernels for molecules that can be naturally represented using a graph. First, a short introduction to kernel methods is given, followed by the necessary notation regarding graphs and trees required for the graph kernel definitions. Finally, state-of-the-art graph kernels for molecules are presented.

Chapter 3 proposes a new graph kernel to address the forward problem as an extension of the subtree kernel initially proposed by Ramon and Gärtner.¹ First, the basic idea for extending the subtree kernel is given. Then, two extensions, called the inexact match extension and the importance weight extension, are introduced. The differences in relation to previous research are then discussed. Atom labels with information regarding the local atom environment are derived. The computation of the proposed kernel is presented thereafter. Finally, application to classification and regression tasks for predicting various pharmaceutical properties from the structure of molecules is described.

Chapter 4 develops a population-based Monte Carlo method for sampling structurally diversified molecules with good drug-likeness. First, the basic idea for the development of the sampling method is given. Then, evolutionary operators (i.e., mutation, crossover, and exchange) for the structural alteration of molecules are introduced. Next, a molecu-

lar fragment database required for the mutation operation is prepared. In order to penalize non-drug-like molecules, existing knowledge of drug-likeness commonly used by medicinal chemists is introduced. Finally, the effectiveness of the proposed sampling method is demonstrated by pre-image reconstruction experiments.

Chapter 2

Kernel Methods for Molecules

This chapter describes the extension of kernel methods to handle structured data such as chemical structures through the convolution kernels proposed by Haussler.³³ We begin with a brief introduction of kernel methods, followed by a description of graph representations of chemical structures, and a review of existing graph kernels.

2.1 Kernel Methods

Traditionally, machine learning algorithms (e.g., support vector machine, logistic regression, K-means clustering, etc.) have been well developed for the linear case. However, real-world problems often require nonlinear algorithms to detect the complex patterns that allow for the successful prediction of properties of interest. The use of a positive definite kernel^{19,20} allows the extension of linear algorithms to nonlinear algorithms (Figure 2.1). The symmetric function $k: \mathcal{G} \times \mathcal{G} \rightarrow \mathbb{R}$ on the domain \mathcal{G} is said to be a positive definite kernel if and only if $\sum_{i,j \in \{1, \dots, n\}} c_i c_j k(G_i, G_j) \geq 0$ for all $n \in \mathbb{N}$, $G_1, \dots, G_n \in \mathcal{G}$, and $c_1, \dots, c_n \in \mathbb{R}$. For such k , it is known that a map $\phi: \mathcal{G} \rightarrow \mathcal{F}$ into a (usually high-dimensional) feature space \mathcal{F} exists, such that $k(G, G') = \langle \phi(G), \phi(G') \rangle$ for all $G, G' \in \mathcal{G}$. In \mathcal{F} , the complex patterns can be found as linear relations. Here, substituting $k(G, G')$ for $\langle \phi(G), \phi(G') \rangle$ is crucial for implicitly mapping the input data into \mathcal{F} without ever knowing $\phi(\cdot)$. Any linear machine

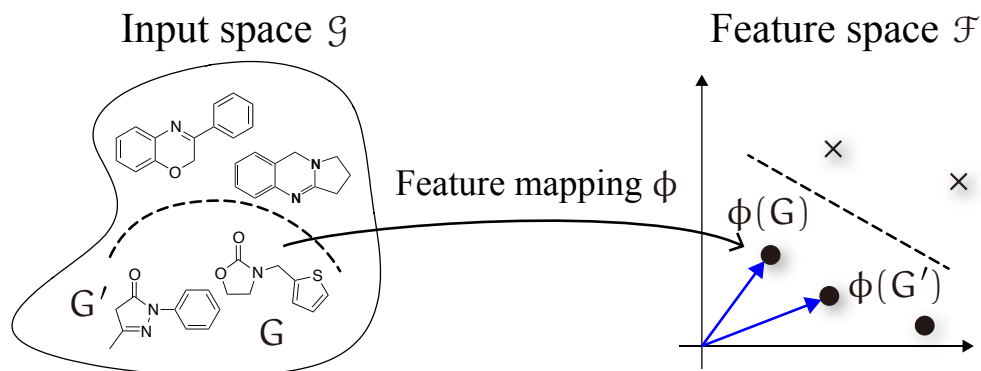


Figure 2.1 The idea of kernel methods. This approach maps the training data in the input space \mathcal{G} into a high-dimensional feature space \mathcal{F} via the feature map $\phi : \mathcal{G} \rightarrow \mathcal{F}$, and applies linear machine learning algorithms such as SVM, which depend on the inner product $\langle \phi(G), \phi(G') \rangle$ between data points $\phi(G)$ and $\phi(G')$. Using the kernel trick $k(G, G') = \langle \phi(G), \phi(G') \rangle$, it is possible to apply them without explicitly mapping ϕ .

learning algorithm formulated as an inner product in \mathcal{F} can be turned into a nonlinear one by the kernel substitution, $k(G, G') = \langle \phi(G), \phi(G') \rangle$. This approach is called the kernel trick. Next we review the nonlinear extension of support vector machines (SVMs) using the kernel trick.

Let us consider a typical binary classification task in chemical informatics as illustrated in Figure 2.1. The goal is to construct a discriminant function to predict whether a new input molecule is effective against a disease. Here, any molecule G in the set \mathcal{G} of all molecules can be transformed into a feature vector $\phi(G) \in \mathbb{R}^p$ containing molecular properties (e.g., molecular weight, molecular hydrophobicity, polar surface area, etc.). Suppose we are given the training data set $\mathcal{D} = \{(\phi(G_i), y_i) \in \mathcal{X} \times \mathcal{Y}\}_{i=1}^n$ where \mathcal{X} is the nonempty set of p -dimensional feature vectors and $\mathcal{Y} \in \{+1, -1\}$ is the set of class labels whose value takes either the presence (+1) or absence (−1) of efficacy against the disease. Assume that \mathcal{D} is separable, i.e., there exists a discriminant function $f : \mathcal{G} \rightarrow \mathcal{Y}$,

$$f(G) = \text{sgn}(\langle \mathbf{w}, \phi(G) \rangle + b), \quad (2.1)$$

where the weight vector $\mathbf{w} \in \mathbb{R}^p$ and the shift coefficient $b \in \mathbb{R}$ are parameters. SVM

determines the hyperplane, which separates the two classes with the largest margin, by solving the constrained optimization problem

$$\underset{\mathbf{w}, b}{\text{minimize}} \quad \frac{1}{2} \|\mathbf{w}\|^2 \quad \text{subject to} \quad y_i(\langle \mathbf{w}, \phi(G_i) \rangle + b) \geq 1 \text{ for all } i = 1, \dots, n. \quad (2.2)$$

Note that $\|\mathbf{w}\|^{-1} f(G_i)$ is the distance from the point $\phi(G_i)$ to the hyperplane $H(\mathbf{w}, b) := \{\phi(G) | \langle \mathbf{w}, \phi(G) \rangle + b = 0\}$. The condition $y_i(\langle \mathbf{w}, \phi(G_i) \rangle + b) \geq 1$ ensures that the margin distance is at least $2\|\mathbf{w}\|^{-1}$. Consequently, minimizing $\|\mathbf{w}\|$ subject to the constraints maximizes the margin of separation. To address this problem one can solve it in dual space, as follows. The Lagrange function of eq 2.2 is given by

$$L(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^n \alpha_i (y_i(\langle \mathbf{w}, \phi(G_i) \rangle + b) - 1), \quad (2.3)$$

where $\alpha_i \geq 0$ is a Lagrange multiplier. L has to be minimized with respect to \mathbf{w} and b and maximized with respect to α_i . At the saddle point, the derivatives of L with respect to the variables \mathbf{w} and b must be equal to zero,

$$\frac{\partial}{\partial b} L(\mathbf{w}, b, \boldsymbol{\alpha}) = 0 \text{ and } \frac{\partial}{\partial \mathbf{w}} L(\mathbf{w}, b, \boldsymbol{\alpha}) = 0,$$

which leads to

$$\sum_{i=1}^n \alpha_i y_i = 0 \quad (2.4)$$

and

$$\mathbf{w} = \sum_{i=1}^n \alpha_i y_i \phi(G_i). \quad (2.5)$$

Substituting eq 2.4 and eq 2.5 into the Lagrangian eq 2.3, we obtain the so-called dual

optimization problem

$$\begin{aligned} \underset{\alpha \in \mathbb{R}^n}{\text{maximize}} \quad & W(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j \langle \phi(G_i), \phi(G_j) \rangle \\ \text{subject to } & \alpha_i \geq 0 \text{ for all } i = 1, \dots, n \text{ and } \sum_{i=1}^n \alpha_i y_i = 0. \end{aligned}$$

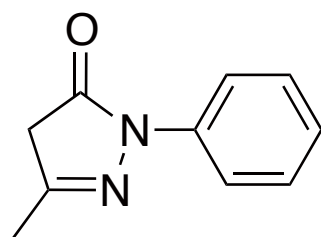
Using eq 2.5, the decision function (eq 2.1) can be written as

$$f(G) = \text{sgn} \left(\sum_{i=1}^n y_i \alpha_i \underbrace{\langle \phi(G), \phi(G_i) \rangle}_{k(G, G_i)} + b \right). \quad (2.6)$$

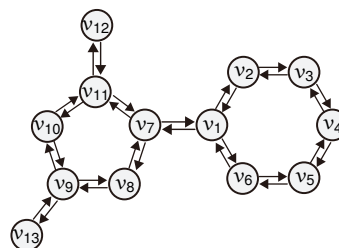
Equation 2.6 can be expressed in terms of the kernel k using the kernel trick $k(G, G_i) = \langle \phi(G), \phi(G_i) \rangle$. The kernel trick allows us to handle the feature vectors $\phi(G)$ in the very high-dimensional feature space with no need to perform the explicit transformation, ϕ . To deal with structured data such as chemical structures, many graph kernels have been developed, as described in Section 2.3. For further information on kernel-based machine learning please see the literature.^{17–20}

2.2 Representing Chemical Structures

Let us represent the chemical structure of a molecule by a labeled directed graph, $G = (\mathcal{V}, \mathcal{E})$, as shown in Figure 2.2. 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. Let $\Sigma_{\mathcal{V}}, \Sigma_{\mathcal{E}}$ be the sets of vertex labels and edge labels, respectively. In the case of labeled graphs, there is also a set of labels $\Sigma = \Sigma_{\mathcal{V}} \cup \Sigma_{\mathcal{E}}$ with a labeling function $\ell: \mathcal{V} \cup \mathcal{E} \rightarrow \Sigma$ 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)



(a) Chemical structure

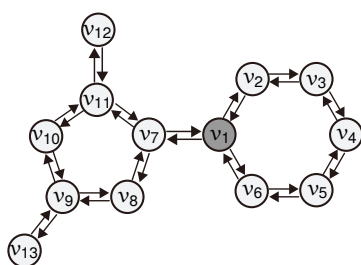


(b) Molecular graph

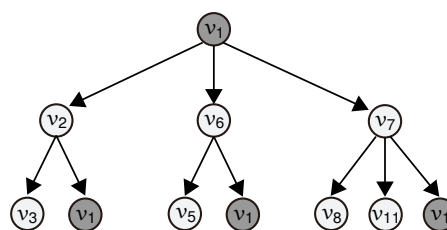
Figure 2.2 A chemical structure (left) can be modeled as a labeled directed graph (right).

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. 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 (see Figure 2.3).



(a) Molecular graph



(b) Subtree patterns

Figure 2.3 A molecular graph (left) and subtree patterns up to the height $h = 2$ rooted at the node v_1 (right). Note that the vertex v_1 appears at a height of 2 again.

2.3 Graph Kernels

The traditional application of machine learning with the kernel method only considers data represented in a single row of a table. However, there are many potential machine learning applications, where this is not the natural representation. For example, such applications include the classification of molecules internally represented as graphs. The best known framework to construct kernels for structured data is the convolution kernel proposed by Haussler.³³ Following this framework, various graph kernels have been proposed over the last decade (see Figure 2.4) and applied successfully to various machine learning tasks in chemical informatics, including the establishment of QSARs. These graph kernels differ with respect to the choice of the subgraph types used to represent the structured objects, such as walks,^{24,34,37,39} paths,³⁵ trees,^{1,26,40,56} cycles,^{36,57} and subgraphs.^{42,48}

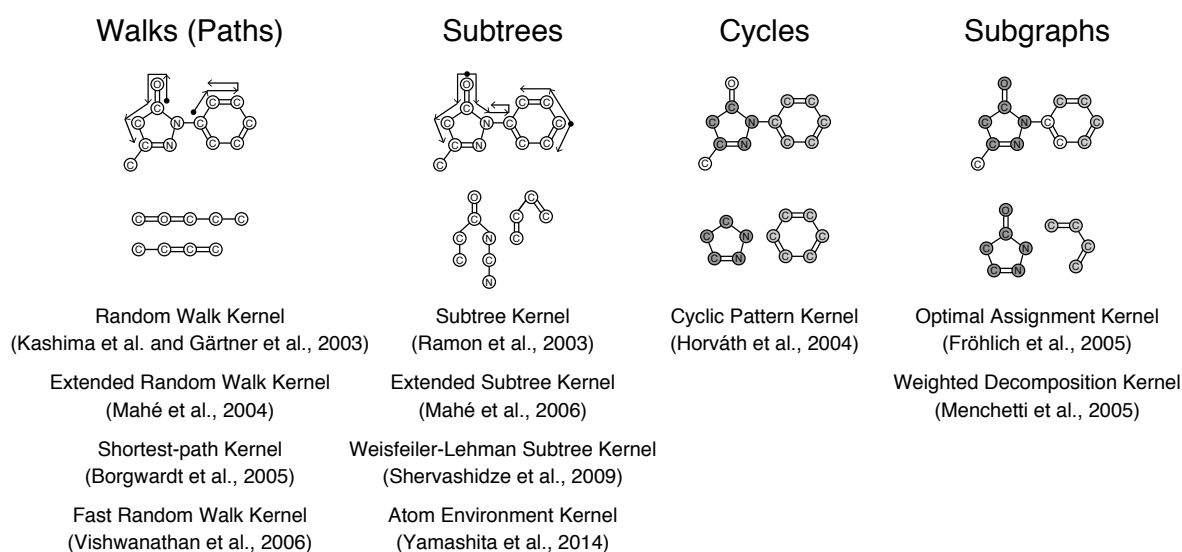


Figure 2.4 The research efforts on graph kernels over the last decade.

2.3.1 Convolution Kernels

The basic idea of the convolution kernels³³ is that each structured object is decomposed into a set of parts, and the kernel between the objects is defined as the sum of the kernel values

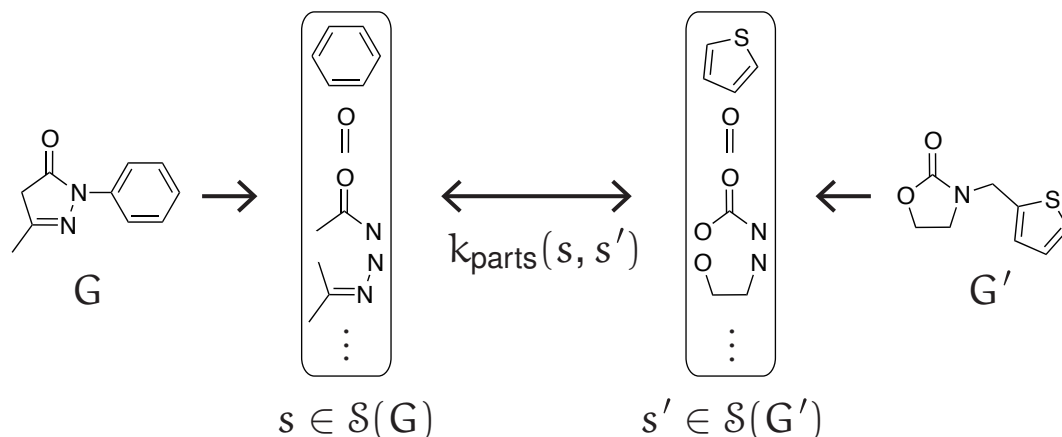


Figure 2.5 A schematic concept of the convolution kernel between the molecular graphs G and G' .

among the parts (see Figure 2.5).

Let $G, G' \in \mathcal{G}$ be the molecular graphs and let $\mathcal{S}(G), \mathcal{S}(G')$ be sets of parts extracted from G, G' . Given an extraction rule, we can define the sets $\mathcal{S}(G)$ and $\mathcal{S}(G')$ of parts. The convolution kernel of G and G' is then defined as

$$k_{\text{conv}}(G, G') = \sum_{s \in \mathcal{S}(G)} \sum_{s' \in \mathcal{S}(G')} w(s)w(s')k_{\text{parts}}(s, s'), \quad (2.7)$$

where the function $w(s)$ returns a weight for the part s and $k_{\text{parts}}(s, s')$ is the kernel function between two parts s and s' . Equation 2.7 is guaranteed to be a valid kernel if k_{parts} is a positive definite kernel. It should be noted that the weight function is not defined in the original definition.³³ The convolution kernel is very general and can be used for many different structured objects (e.g., amino acid sequences, chemical structures, metabolic networks, etc.). To construct a convolution kernel for specific structured objects, we have to design the extraction rule of parts and the kernel function on the parts.

We next describe two established graph kernels: random walk kernels and subtree kernels.

2.3.2 Random Walk Kernels

The idea of the random walk (RW) kernel^{24,34} is to randomly walk on two graphs and compare the label sequences resulting.

Consider a random walk on a graph G , which starts at vertex $x_1 \in \mathcal{V}_G$ with initial probability p_s , goes from x_{i-1} to x_i with transition probability p_t , and ends with probability p_q . The random walk can be represented as a sequence of the vertices traversed of length l , $\mathbf{x} = (x_1, x_2, \dots, x_l)$. The vertex sequence \mathbf{x} in G has probability

$$p(\mathbf{x}|G) = p_s(x_1) \left(\prod_{i=2}^l p_t(x_i|x_{i-1}) \right) p_q(x_l).$$

Let us define a label sequence by another alternating sequence of vertex labels and edge labels

$$\mathbf{h} = (h_1, h_2, \dots, h_{2l-1}) \in (\Sigma_{\mathcal{V}} \Sigma_{\mathcal{E}})^{l-1} \Sigma_{\mathcal{V}}.$$

We then obtain the label sequence associated with \mathbf{x}

$$\mathbf{h}_{\mathbf{x}} = (\ell(x_1)\ell(x_1, x_2), \ell(x_2), \dots, \ell(x_{l-1})\ell(x_{l-1}, x_l), \ell(x_l)).$$

The probability of the label sequence \mathbf{h} is the sum of probabilities of all vertex sequences that generate \mathbf{h}

$$p(\mathbf{h}|G) = \sum_{\mathbf{x}} I(\mathbf{h} \cong \mathbf{h}_{\mathbf{x}}) \cdot \left(p_s(x_1) \prod_{i=2}^l p_t(x_i|x_{i-1}) p_q(x_l) \right),$$

where $I(\mathbf{h} \cong \mathbf{h}_{\mathbf{x}})$ is the indicator function that returns 1 if \mathbf{h} and $\mathbf{h}_{\mathbf{x}}$ are equal and 0 otherwise.

Next, we define a kernel k_z between two label sequences \mathbf{h} and \mathbf{h}' . Suppose we are given two valid kernels k_v between vertex labels and k_e between edge labels. The kernel

for the label sequences of equal length is given by the product of the label kernels

$$k_z(\mathbf{h}, \mathbf{h}') = k_v(h_1, h_1') \prod_{i=1}^{l-1} k_e(h_{2i}, h_{2i}') k_v(h_{2i+1}, h_{2i+1}').$$

For the label sequences \mathbf{h} and \mathbf{h}' of different lengths, $k_z(\mathbf{h}, \mathbf{h}') = 0$. Finally, the random walk kernel is given by the expectation of k_z over all possible \mathbf{h} and \mathbf{h}'

$$k_{RW}(G, G') = \sum_{\mathbf{h}} \sum_{\mathbf{h}'} k_z(\mathbf{h}, \mathbf{h}') p(\mathbf{h}|G) p(\mathbf{h}'|G').$$

This kernel is positive definite for valid k_v and k_e .

The RW kernel exploits an infinite dimensional feature space, spanned by random walks, on molecular graphs. In consequence, the RW kernel gives an alternative to explicit vector representations of molecules (molecular descriptors). However, Gärtner et al.³⁴ indicated the limited expressiveness of the RW kernel based on linear features. To alleviate this limitation, Gärtner et al.³⁴ proposed subtree kernels, as described in the next section.

2.3.3 Subtree Kernels

In this section we describe the subtree (ST) kernel initially proposed by Ramon and Gärtner¹ and later extended by Mahé and Vert.²⁶ Following Mahé and Vert,²⁶ 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 $\mathcal{V}_T = (w_1, \dots, w_{|T|})$ be a rooted tree with a designated root w_1 . A $|T|$ -tuple of vertices $(v_1, \dots, v_{|T|}) \in \mathcal{V}_G^{|T|}$ is said to be a tree-pattern of G with respect to T , denoted by

$(v_1, \dots, v_{|T|}) = \text{pattern}(T)$, if and only if

$$\begin{cases} \forall i \in \{1, \dots, |T|\}, & \ell(v_i) = \ell(w_i), \\ \forall (w_i, w_j) \in \mathcal{E}_T, & (v_i, v_j) \in \mathcal{E}_G \wedge \ell((v_i, v_j)) = \ell((w_i, w_j)), \\ \forall (w_i, w_j), (w_i, w_k) \in \mathcal{E}_T, & j \neq k \Leftrightarrow v_j \neq v_k. \end{cases} \quad (2.8)$$

With the set of all possible tree-patterns of $G = (\mathcal{V}_G, \mathcal{E}_G)$ with $\mathcal{V}_G = (v_1, \dots, v_{|\mathcal{V}_G|})$ arranged in T ,

$$\mathcal{P}_T(G) = \{(v_{a_1}, \dots, v_{a_{|T|}}) | (a_1, \dots, a_{|T|}) \in \{1, \dots, |\mathcal{V}_G|\}^{|T|} \wedge (v_{a_1}, \dots, v_{a_{|T|}}) = \text{pattern}(T)\}, \quad (2.9)$$

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

$$k_{\text{ST},h}(G, G') = \sum_{T \in \mathcal{T}_h} \mu(T) \sum_{p \in \mathcal{P}_T(G)} \sum_{p' \in \mathcal{P}_T(G')} I(p \cong p'). \quad (2.10)$$

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 one if p and p' are isomorphic, and zero otherwise. In this case, $I(p \cong p')$ always returns one 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.²⁶

2.3.4 Other Graph Kernels

There have been various graph kernels proposed in the literature (see Vishwanathan et al.²¹). In this section, we discuss several important kernels in chemical informatics.

Walk-based kernels, one of the first graph kernels, have been independently proposed by Kashima et al.²⁴ and Gärtner et al..³⁴ Unfortunately, random walks on a graph include tottering between two neighboring vertices, which is otherwise known as an immediate re-visiting of a vertex. Such tottering walks are likely to be uninformative. Mahé et al.³⁷ introduced the second-order Markov model to filter tottering walks. Path-based graph kernels³⁵ have been proposed to invalidate the effects of tottering. Another extension³⁷ is to increase the number of different vertex labels using the Morgan index. In the label enrichment, contextual structural information around each vertex is embedded into the vertex label. Subsequently, Vishwanathan et al.³⁹ employed fast methods for solving Sylvester equations as well as conjugate gradient and fixed point iteration methods to speed up walk based kernels.

Ramon and Gärtner¹ defined kernels through the comparison of subtrees instead of walks on the graphs. This alleviates the limited expressiveness of linear features generated by random walks. The subtree kernel was later refined by Mahé and Vert.²⁶ Subsequently, the Weisfeiler-Lehman (WL) subtree kernel⁴⁰ was developed to provide an efficient kernel computation. The WL subtree kernel scales up to large labeled graphs. It uses the Weisfeiler-Lehman isomorphism test, which consists of iterative multiset-label determination, label compression, and relabeling steps.

Horváth et al.^{36,57} proposed cyclic pattern (CP) kernels, which are based on the idea of mapping graphs to the sets of cyclic patterns and tree patterns, which are compared with the intersection kernel. Menchetti et al.⁴⁸ proposed weighted decomposition kernels based on comparing local neighborhoods of vertices. The optimal assignment (OA) kernel,⁴² another graph kernel comparing local neighborhoods, arises from finding the best match

between substructures of graphs. However, the OA kernel is not positive semidefinite in general.⁵⁸ Vishwanathan et al.²¹ suggested a possible remedy to this problem based on an approximation of the tropical semiring.

Other kernels, motivated by applications in chemical informatics, include fingerprint kernels^{38,59} where a molecular graph G is represented by a vector $\phi(G)$ indexed by a set of molecular fragments as illustrated in Figure 2.6, i.e., sequences of atom types and bond types up to a given length. The fingerprint kernels are normalized by three variations of the Tanimoto kernel designed by analogy with the Tanimoto coefficient.

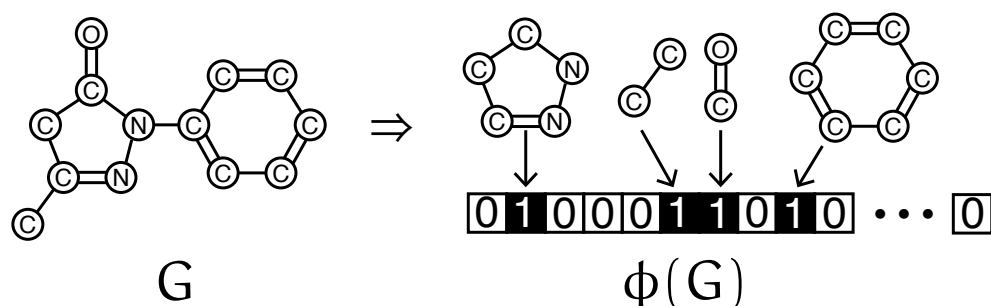


Figure 2.6 Given a molecule graph G , the traditional fingerprint is defined as a binary vector $\phi(G)$ such that it indicates the presence or absence of predefined particular substructures in G .

Chapter 3

Atom Environment Kernels for the Forward Problem

3.1 Basic Idea

In this section we introduce two extensions to the ST kernel. The first extension, referred to as the inexact 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 2.8 as

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

This alters the definition of the set $\mathcal{P}_T(G)$ in eq 2.9. 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 $k_{\text{tree}}(\mathbf{p}, \mathbf{p}')$ over all possible pairs of tree-patterns induced from G and G'

$$k_{\text{AE},h}(G, G') = \sum_{T \in \mathcal{T}_h} \sum_{\mathbf{p} \in \mathcal{P}_T(G)} \sum_{\mathbf{p}' \in \mathcal{P}_T(G')} w(\mathbf{p}) w(\mathbf{p}') k_{\text{tree}}(\mathbf{p}, \mathbf{p}'), \quad (3.1)$$

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

3.2 Inexact Match Extension

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

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

The atom-level kernel $k_{\text{atom}}(\mathbf{e}_r(\mathbf{v}), \mathbf{e}_r(\mathbf{v}'))$ measures the soft similarity between atoms \mathbf{v} and \mathbf{v}' through the atom environment labels $\mathbf{e}_r(\mathbf{v})$ and $\mathbf{e}_r(\mathbf{v}')$. The atom environment label $\mathbf{e}_r(\mathbf{v})$ captures the local environment of each atom \mathbf{v} in the molecular graph. As will be shown in section 3.5.1, $\mathbf{e}_r(\mathbf{v}) \in \mathbb{R}^d$ ($d = 2$ in the present study) is derived from the modified Burden matrix⁴⁹ of a neighboring substructure of a topological radius r centered at atom \mathbf{v} . The tree-level kernel $k_{\text{tree}}(\mathbf{p}, \mathbf{p}')$ measures the similarity of \mathbf{p} and \mathbf{p}' as the product of the atom-level kernels over a set $\mathcal{A}(\mathbf{p}, \mathbf{p}') = \{(\mathbf{p}[\mathbf{i}], \mathbf{p}'[\mathbf{i}])\}_{\mathbf{i}=1}^{|\mathbf{p}|}$ of the aligned atom pairs of \mathbf{p} and \mathbf{p}' , where $\mathbf{p}[\mathbf{i}]$ is the \mathbf{i} th element of a tuple \mathbf{p} .

We construct a compactly supported (CS) kernel for k_{atom} by multiplying the Gaussian kernel with a width parameter γ by a Wendland function⁶⁰

$$k_{\text{atom}}(\mathbf{e}_r(\mathbf{v}), \mathbf{e}_r(\mathbf{v}')) = \psi_{d,c} \left(\frac{\|\mathbf{e}_r(\mathbf{v}) - \mathbf{e}_r(\mathbf{v}')\|}{\theta} \right) \exp(-\gamma \|\mathbf{e}_r(\mathbf{v}) - \mathbf{e}_r(\mathbf{v}')\|^2). \quad (3.2)$$

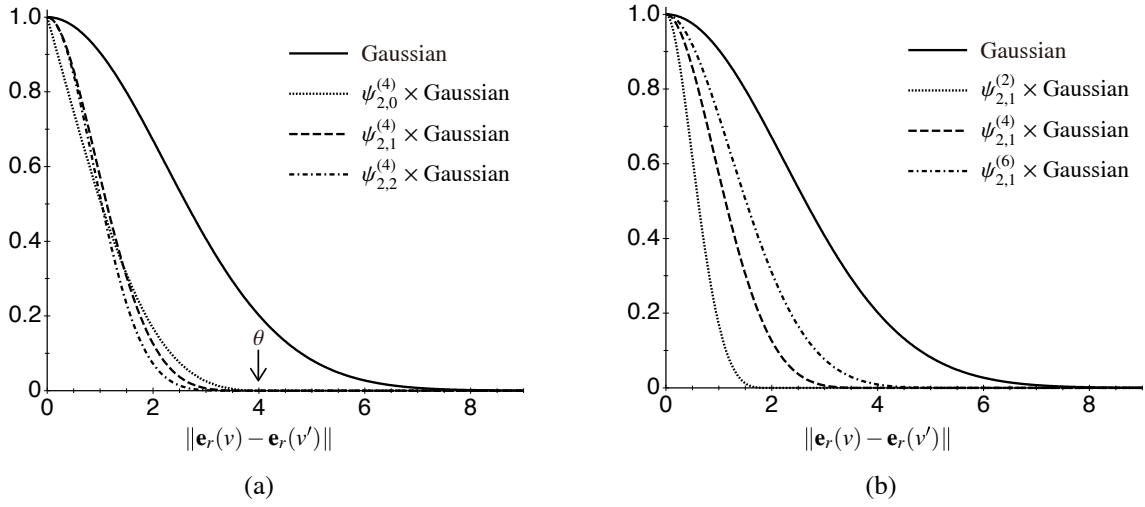


Figure 3.1 Plots of the Gaussian kernel with a width parameter of $\gamma = 0.1$ (solid line) and the CS kernels $\psi_{2,c}^{(\theta)} \times \text{Gaussian}$ with respect to (a) the smoothing parameter c and (b) the cut-off distance θ .

This construction has been proposed in a general machine learning context to yield a sparse Gram matrix without destroying the positive definiteness of any RBF kernel.⁶¹ The Wendland functions $\psi_{d,c}$ are defined for the dimension d of input variables and the smoothing parameter c , and tend to zero when the L_2 distance $\|\mathbf{e}_r(v) - \mathbf{e}_r(v')\|$ is beyond a cut-off distance θ . With this construction, k_{atom} can smoothly decay to zero at θ without losing positive definiteness.⁶²

More specifically, the Wendland functions are defined as

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

with the truncated polynomial

$$\psi_s(z) = (1-z)_+^s = \begin{cases} (1-z)^s, & 0 \leq z < 1, \\ 0, & z \geq 1, \end{cases}$$

and the integral operator

$$I[f](z) = \int_z^\infty xf(x)dx, \quad z \geq 0,$$

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 $d \leq 2s - 1$. We can compute the functions $\psi_{d,c}$ for $d = 2$ and $c = 0, 1, 2$ directly by the explicit form⁶³

$$\begin{aligned}\psi_{2,0}(z) &= (1-z)_+^2, \\ \psi_{2,1}(z) &= (1-z)_+^4(4z+1), \\ \psi_{2,2}(z) &= (1-z)_+^6(1+6z+\frac{35}{3}z^2).\end{aligned}$$

In Figure 3.1 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 3.1, we fix $c = 0$ in this thesis.

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.

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-square (χ^2) statistic is used to measure the statistical significance of the atoms. Each atom v is characterized by another atom environment label $\alpha_r(v) \in \mathbb{Z}$. As described later herein, $\alpha_r(v)$ encodes information on a neighboring

Table 3.1 Two-way Contingency Table of Atom Environment Label a and Class Label c^a

	c	$\neg c$	$\sum \text{row}$
a	A	B	$A + B$
$\neg a$	C	D	$C + D$
$\sum \text{column}$	$A + C$	$B + D$	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 $\neg c$).

substructure of a topological radius r centered at atom v using a Morgan type algorithm.³¹ Using a two-way contingency table (Table 3.1), 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

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

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

$$\hat{w}(a_r(v)) = \begin{cases} \lambda_\alpha, & \text{if } \chi^2(a_r(v)) \geq \tau, \\ \lambda_\beta, & \text{otherwise,} \end{cases}, \quad 0 < \lambda_\beta \leq \lambda_\alpha < 1, \quad (3.3)$$

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. The binarized atomic weights allows for easy visualization of significant atoms in a specific molecule, as seen in later.

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

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

where \bar{y}_i , $\text{var}(y_i)$, 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 3.3, 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

$$k_{\text{AE,h}}(G, G') = \sum_{T \in \mathcal{T}_h} \sum_{p \in \mathcal{P}_T(G)} \sum_{p' \in \mathcal{P}_T(G')} \lambda_\alpha^{n_\alpha(p) + n_\alpha(p')} \lambda_\beta^{n_\beta(p) + n_\beta(p')} \prod_{(v, v') \in \mathcal{A}(p, p')} k_{\text{atom}}(\mathbf{e}_r(v), \mathbf{e}_r(v')). \quad (3.4)$$

The atom-level kernels preserving positive definiteness are closed under tensor product and non-negative linear combinations.⁶⁴ 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 3.4, 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 3.1) and the ST kernel (eq 2.10). 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 $k_{\text{tree}}(p, p')$ with compact support. The $k_{\text{tree}}(p, p')$ 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 3.7, we demonstrate how these extended building blocks improve the performance in predicting various pharmaceutical properties of molecules.

3.5 Atom Environment Labels

3.5.1 Continuous Labels

The atom environment label $\mathbf{e}_r(v) \in \mathbb{R}^2$ is derived from a modified Burden matrix⁴⁹ (Figure 3.2) 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

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

where for the i th 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 i th and j th 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 i th or the j th 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., $\mathbf{e}_r(v) = (e_{\min}, e_{\max})^t$. Among the eigenvalues obtained from \mathbf{B} , the smallest and the largest

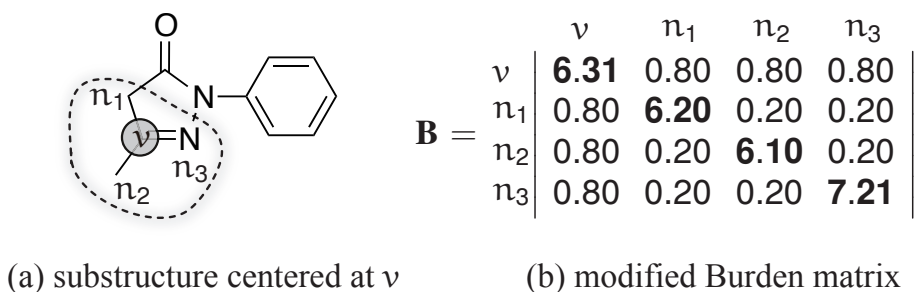


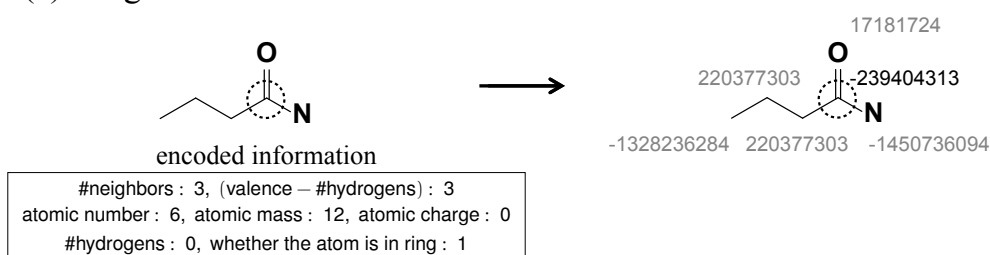
Figure 3.2 The modified Burden matrix of a substructure centered at v .

have been empirically demonstrated to reflect structural relevant aspects of molecules.⁵¹ The smallest eigenvalue reflects the topology of a molecule and on the other hand the largest eigenvalue reflects the atom types.

3.5.2 Discrete Labels

Another atom environment label $\alpha_r(v) \in \mathbb{Z}$ is generated in order to capture information on a neighboring substructure of a topological radius r centered at atom v using a variant³¹ of the Morgan algorithm⁵² (Figure 3.3). 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 $\alpha_r(v)$. The hashed integer code leads to a saving of computational cost and a reduction of memory use. The following atomic properties are considered for

(a) assignment of initial atom identifiers



(b) generation of new atom identifiers

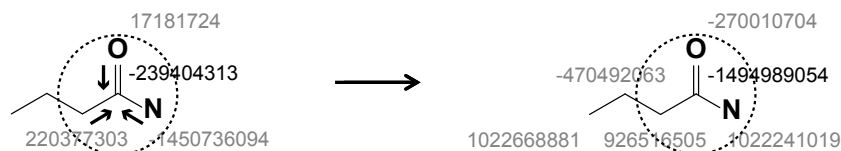


Figure 3.3 In the ECFP algorithm, (a) the assignment of initial atom identifiers, computed by encoding seven atomic properties, (b) the generation of new atom identifiers by performing one iteration.

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.

3.6 Kernel Computation

3.6.1 Recursive Algorithm

In this section, we derive the recursive formula for computing the AE kernel without enumerating tree-patterns by following Mahé and Vert.²⁶ 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

$$\begin{aligned} \mathcal{M}(v, v') = & \{R \subseteq \mathcal{N}(v) \times \mathcal{N}(v') \\ & | (\forall (u, u'), (w, w') \in R : u \neq w \wedge u' \neq w') \\ & \wedge (\forall (u, u') \in R : (\ell(v, u) = \ell(v', u')))\}. \end{aligned}$$

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

$$k_{\text{AE},h}(G, G') = \sum_{v \in \mathcal{V}_G} \sum_{v' \in \mathcal{V}_{G'}} k_h(v, v'), \quad (3.5)$$

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

$$k_i(v, v') = \begin{cases} \hat{w}(\mathbf{a}_r(v)) \hat{w}(\mathbf{a}_r(v')) k_{\text{atom}}(\mathbf{e}_r(v), \mathbf{e}_r(v')), & i = 0, \\ k_0(v, v') \left[1 + \sum_{R \in \mathcal{M}(v, v')} \prod_{(w, w') \in R} k_{i-1}(w, w') \right], & i = 1, \dots, h. \end{cases} \quad (3.6)$$

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

3.6.2 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 3.6 is then bounded above by $\sum_{r=1}^d r(dP_r)^2 = \mathcal{O}(d^{2d})$ where dP_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

$$\mathcal{O}(|\mathcal{V}_G| \cdot |\mathcal{V}_{G'}| \cdot h \cdot d^{2d}). \quad (3.7)$$

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 cut-off distance θ of the CS kernel (eq 3.2).

3.7 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 3.2. 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,²⁶ the Weisfeiler-Lehman subtree (WLST) kernel proposed by Shervashidze and Borgwardt,⁴⁰ the extended random walk (ERW) kernel proposed by Mahé et al.,³⁷ the optimal assignment (OA) kernel proposed by Fröhlich et al.,⁴² and the extended-connectivity fingerprint (ECFP).³¹ ECFPs are most commonly used in a wide variety of applications³¹ 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-validation (MCCV), in addition to the runtime required for the Gram matrix computation.

3.7.1 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.⁶⁵ 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 $\{2^n | n \in \mathbb{N}, -10 \leq n \leq 14\}$ 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 topological radius $r \in \{1, 2, 3\}$ of the local environment around each atom; for the CS kernels, the cut-off distance $\theta \in \{0.05, 0.10, 0.20, \dots, 1.40\}$, 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 $(\lambda_\alpha, \lambda_\beta) \in \{(x, y) \in \{0.1, 0.2, \dots, 0.9\}^2 | x \geq y\}$ 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 zero mean and unit variance within each learning set \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 $\{\text{size-based, branching-based, until-N branching-based}\}$. The tree weight factor λ was chosen from $\{0.1, 0.2, \dots, 0.9\}$ 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 $\{0.01, 0.05, 0.1, 0.2, \dots, 0.9\}$. 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 manner³⁸ to the Tanimoto coefficient,³² 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,³⁸ 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. The measure does not involve the learning phase to optimize kernel parameters. In order to perform a fair comparison, we fixed the maximum diameter of the subgraphs used to represent chemical 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 cut-off 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 one; for the ERW kernel, a termination probability of 0.2 was used and the number of the Morgan index iterations was also one.

The AE and ST kernels were implemented in C++ using the OpenBabel toolbox.^{66,67} The EST and ERW kernels were computed using the ChemCpp toolbox.⁶⁸ We used a Matlab implementation⁶⁹ for the WLST kernel and a Java implementation⁷⁰ for the OA kernel. ECFPs were generated using the Pipeline Pilot software.⁷¹ All our experiments were conducted on an Intel Xeon X5570 2.93GHz system with 32GB of main memory.

3.7.2 Data Sets

The 12 data sets on mutagenicity, carcinogenicity, blood-brain barrier penetration, bioavailability, bioactivity, and aqueous solubility of chemical compounds, summarized in Table 3.2, are used. The aqueous solubility data set is a regression task.

In the mutagenesis data set⁷² (MUTAG), the task of interest is to learn a classifier to

Table 3.2 Basic Information of the Data Sets Used Herein^a

abbrev.	samples		description
	#pos.	#neg.	
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.

predict whether each of the 188 aromatic and heteroaromatic nitro compounds is able to cause DNA to mutate. The Predictive Toxicology Challenge data set⁷³ 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,⁷⁴ the objective is to predict BBB penetration of a set of 415 compounds. The Yoshida data set⁷⁵ (BIO) classifies the 265 compounds according to their oral bioavailability. The Sutherland data set⁷⁶ 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 set⁷⁷ (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.

3.7.3 Results and Discussion

In Table 3.3 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 3.4. 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 chemical 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.

Table 3.3 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 \pm 0.063	0.921 \pm 0.060	0.924 \pm 0.069	0.889 \pm 0.127	0.927 \pm 0.084	0.896 \pm 0.078	0.896 \pm 0.081
MM	0.688 \pm 0.085	0.598 \pm 0.094	0.684 \pm 0.072	0.657 \pm 0.091	0.693 \pm 0.098	0.629 \pm 0.089	0.636 \pm 0.102
FM	0.673 \pm 0.091	0.562 \pm 0.108	0.640 \pm 0.092	0.625 \pm 0.072	0.658 \pm 0.087	0.603 \pm 0.070	0.597 \pm 0.085
MR	0.672 \pm 0.096	0.628 \pm 0.105	0.654 \pm 0.070	0.691 \pm 0.132	0.638 \pm 0.119	0.636 \pm 0.091	0.573 \pm 0.124
FR	0.649 \pm 0.096	0.598 \pm 0.117	0.640 \pm 0.075	0.610 \pm 0.102	0.602 \pm 0.088	0.560 \pm 0.104	0.545 \pm 0.119
BBB	0.834 \pm 0.076	0.761 \pm 0.085	0.823 \pm 0.096	0.802 \pm 0.064	0.785 \pm 0.082	0.744 \pm 0.090	0.785 \pm 0.088
BIO	0.767 \pm 0.099	0.688 \pm 0.097	0.669 \pm 0.111	0.699 \pm 0.110	0.675 \pm 0.106	0.727 \pm 0.112	0.716 \pm 0.120
BZR	0.831 \pm 0.064	0.781 \pm 0.075	0.808 \pm 0.078	0.766 \pm 0.094	0.787 \pm 0.091	0.768 \pm 0.085	0.812 \pm 0.075
COX2	0.805 \pm 0.087	0.779 \pm 0.106	0.788 \pm 0.095	0.790 \pm 0.096	0.793 \pm 0.077	0.760 \pm 0.080	0.799 \pm 0.073
DHFR	0.814 \pm 0.080	0.746 \pm 0.091	0.798 \pm 0.088	0.770 \pm 0.086	0.793 \pm 0.083	0.758 \pm 0.119	0.799 \pm 0.092
ER	0.875 \pm 0.050	0.823 \pm 0.067	0.836 \pm 0.068	0.841 \pm 0.052	0.834 \pm 0.075	0.848 \pm 0.053	0.855 \pm 0.072
SOL	0.905 \pm 0.015	0.893 \pm 0.017	0.891 \pm 0.014	0.892 \pm 0.019	0.821 \pm 0.045	0.875 \pm 0.015	0.871 \pm 0.024

^a The areas under the ROC 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 \pm 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).

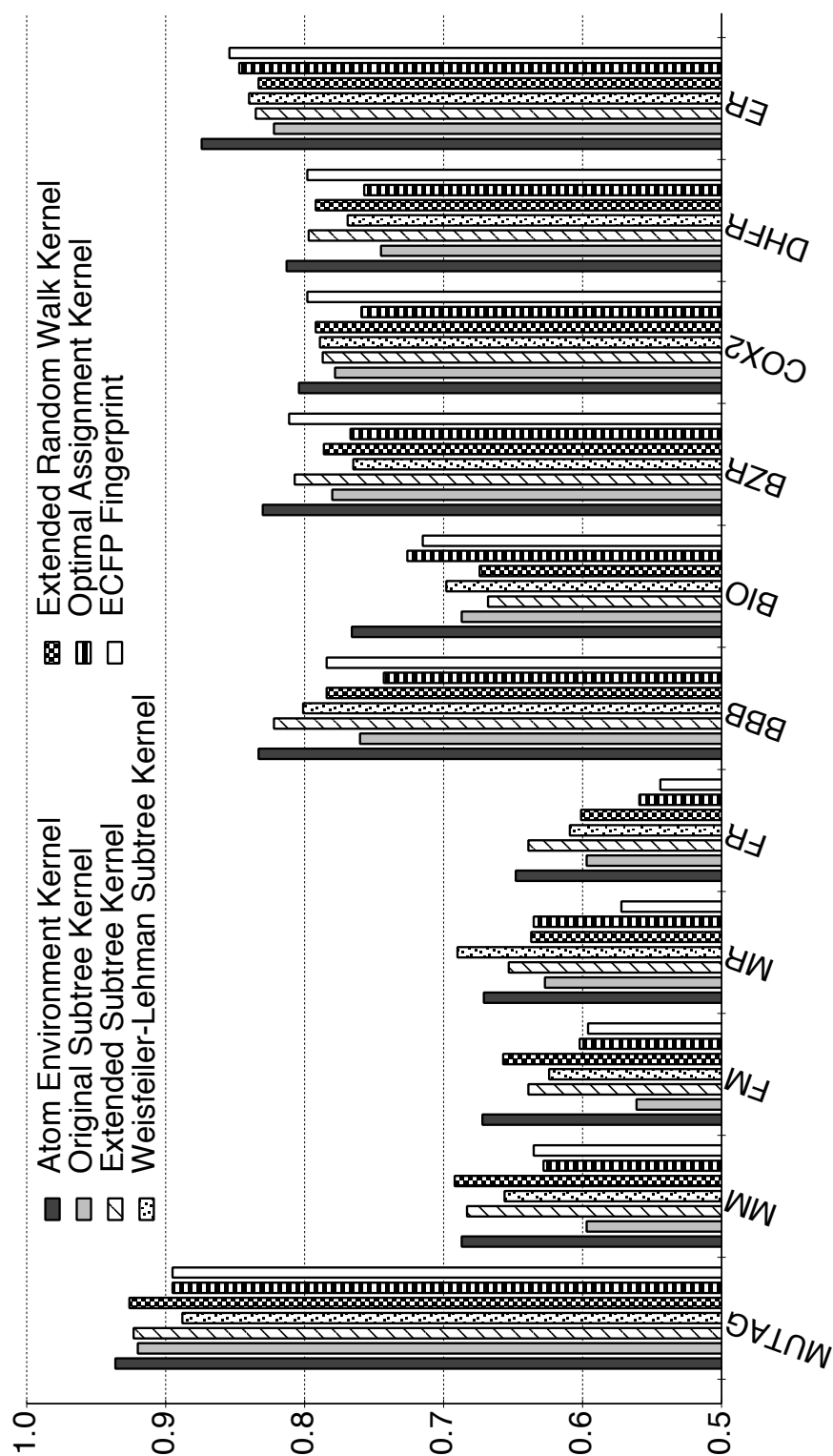


Figure 3.4 Prediction Performance Comparison of the AE Kernel with the Standard Graph Kernels and Molecular Fingerprint on 12 Benchmarks (see Table 3.3 in detail).

Table 3.4 Parametrization of the AE Kernel with the Best Performance^a

data set	parameters		
	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 cut-off 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.

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 3.5). 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 3.4, and (ii) the other restricted AE kernel using only the importance weight extension where exact matching is used instead of k_{atom} in eq 3.4 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.

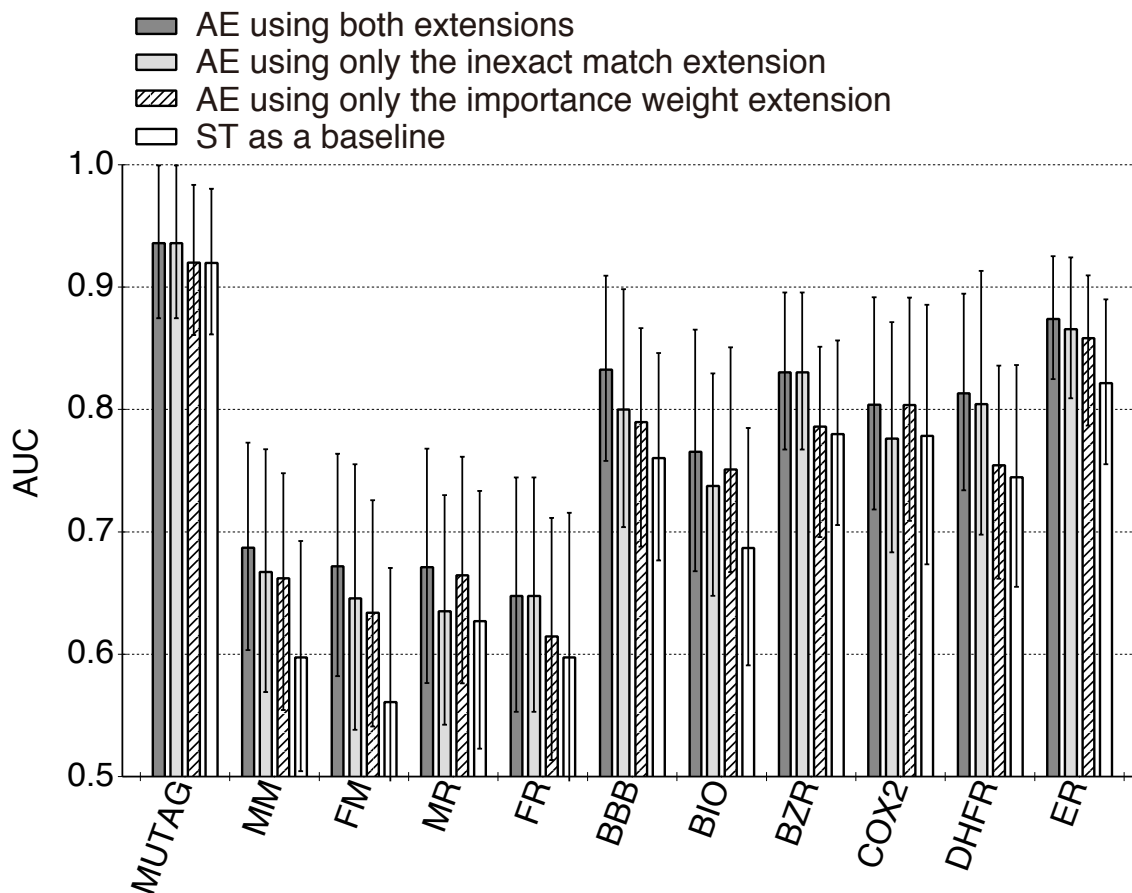


Figure 3.5 Contributions of the two extensions to the improvement of the classification performance for each data set. The best AUC values for each data set of the AE kernel using both extensions (dark shaded bars), the restricted AE kernel using only the inexact match extension (light shaded bars), 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.

We discuss the contribution of each extension in detail next.

One contribution to the improvements arises from the inexact match extension. Figure 3.6 shows the pairwise similarity matrices of atoms between compounds **1** and **2** in the DHFR data set using the ST kernel (Figure 3.6a) and the AE kernels (Figure 3.6b–e) with varying topological radius r and cut-off distance θ . The exact atom matching in the ST kernel causes redundant matches (Figure 3.6a), 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 cut-off distance θ (Figure 3.6b–e). Through comparison of parts b and c or d and e of Figure 3.6, we find that the decrease in θ reduces the number of non-zero 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 cut-off 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 3.6 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 cut-off 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 3.7 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 3.5 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 features (e.g. pharmacophore fea-

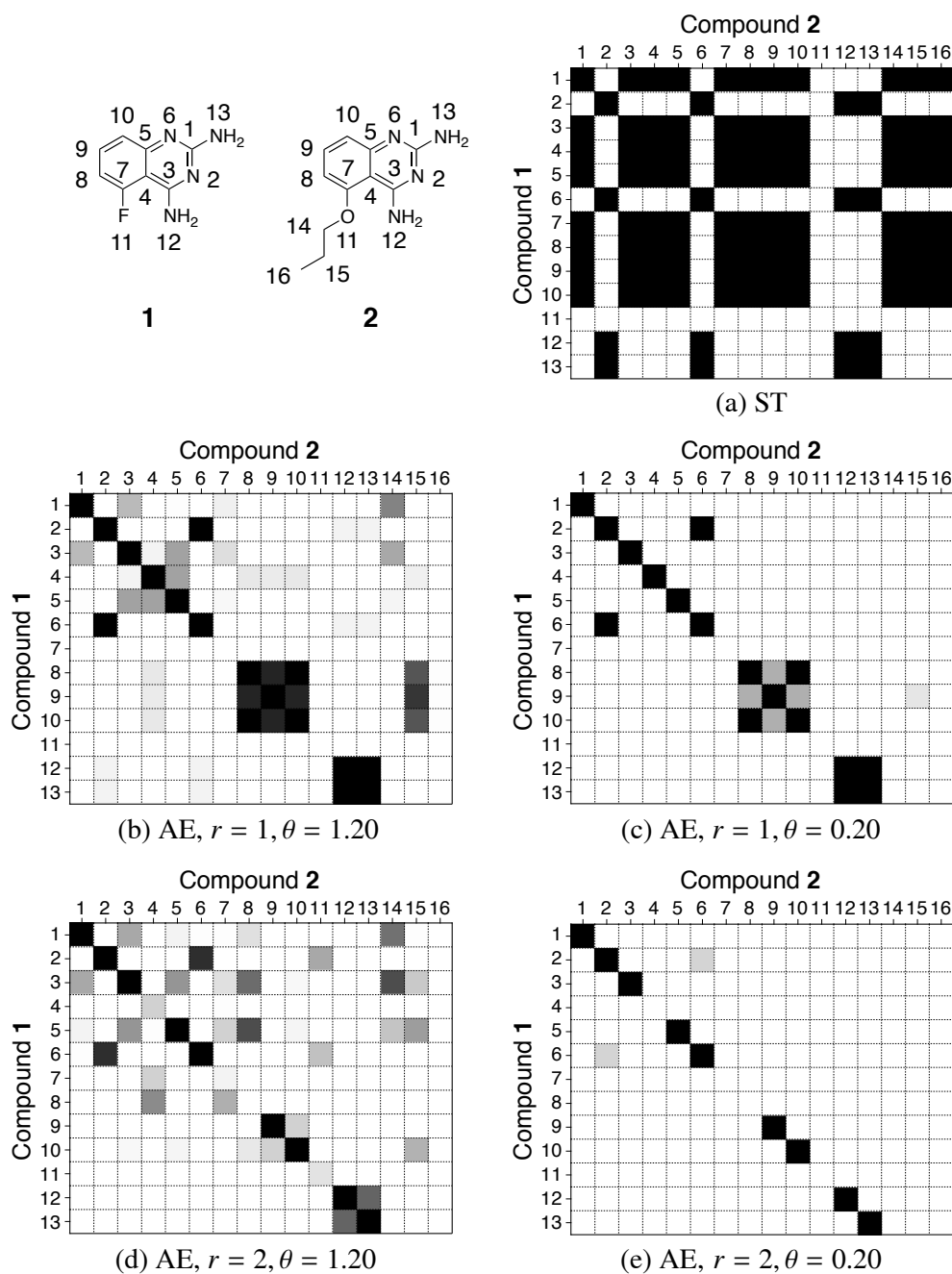


Figure 3.6 Pairwise atom similarity matrices between compounds **1** and **2** in the DHFR data set using (a) the ST kernel and (b)–(e) the AE kernels with varying topological radius r and cut-off distance θ , shaded from white to black to indicate increasing similarity. The width parameter γ of the Gaussian kernel is set to an optimized value of 0.1.

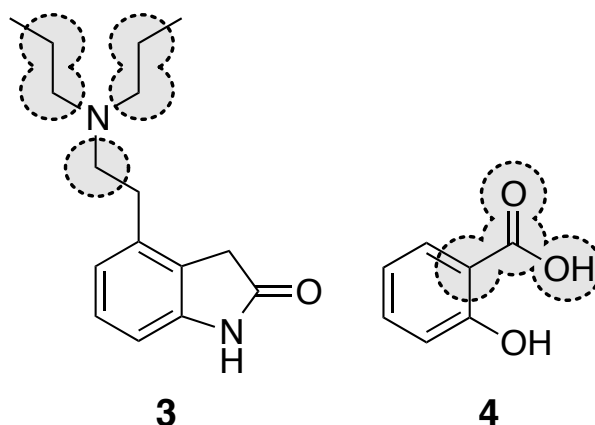


Figure 3.7 Examples of relevant atoms for the task of predicting the BBB penetration as determined from the χ^2 test. The relevant atoms are enclosed by broken lines. Compound **3** penetrates the BBB, but compound **4** does not. All of the kernel parameters are set to the optimized values shown in Table 3.4.

tures). As a result, the importance weighting discloses relevant tree-patterns for the given tasks to the AE kernel, leading to improved performance.

Table 3.5 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 3.8 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 cut-off 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 3.7, but the AE kernel is more efficient at lower cut-off distances ($\theta = 0.05$ 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.

Table 3.5 Computational Efficiency Comparison of the AE Kernel with the Standard Graph Kernels and Molecular Fingerprint on 12 Benchmarks

data set	statistics of the data sets ^a				runtime ^b						
	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.

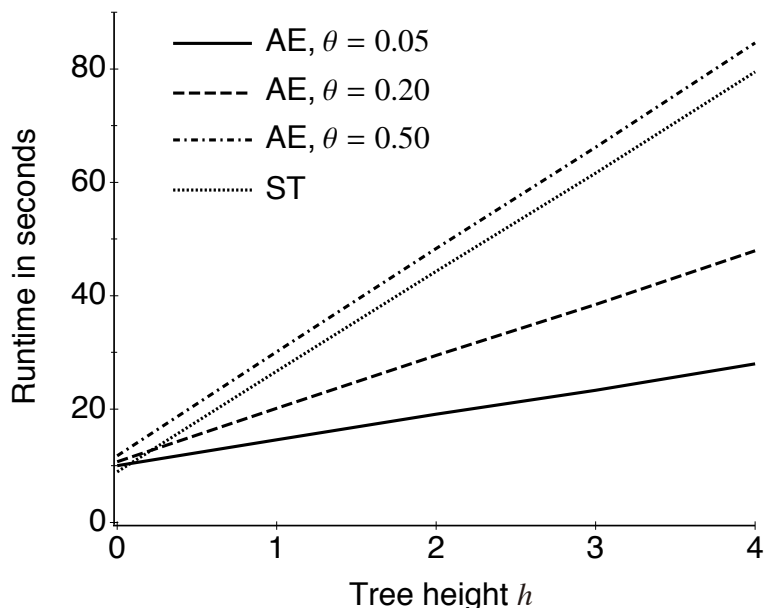


Figure 3.8 Average runtimes in seconds over 10 runs to compute the 1025×1025 Gram matrix on the SOL data set at different tree heights h . We compare the AE kernels with the cut-off distances $\theta = 0.05$ (solid line), 0.20 (dashed line), and 0.50 (dashed-dotted line) and the ST kernel (dotted line).

3.8 Concluding Remarks

We tailored a new graph kernel to molecules by extending the subtree kernel. Firstly, 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. Secondly, 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 proposed 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.⁴¹ The flexible tree-pattern matching will capture new relevant aspects of molecules and progressively enrich the feature space induced by the resulting graph kernel. Chemically inspired extensions include matching between molecular fragments (referred to as bioisosteres⁷⁸), which are structurally distinct yet biologically equivalent. It is necessary to condense the structure of molecules for the bioisostere matching, such that their pharmacophoric features are emphasized using a graph reduction method.^{79,80} Another possible extension is to incorporate stereochemical information, such as chiral centers and cis-trans isomers, into the graph kernel.⁸¹

Interesting related research ideas include the application of the proposed kernel in board games like Go, Shogi, and Checkers. In board games, each board status can be represented by a graph. The proposed kernel could enable us to match with better computer opponents.

Chapter 4

Fragment Assembly Monte Carlo Methods for the Inverse Problem

In this chapter, we are concerned with the problem of reconstructing a corresponding molecular graph (aka pre-image) from its feature space representation induced by a graph kernel, known as the pre-image problem.

4.1 Basic Idea

The pre-image problem is of central importance for the design of new molecules with desired properties.^{2,27} In order to address the pre-image problem, we propose a new sampling method for chemical structures, called the fragment assembly Monte Carlo (FAMC) method. The proposed method was inspired by two different Monte Carlo methods: the fragment assembly method^{82–90} for *de novo* protein structure prediction, which is based on the replacement of residue fragments, and the evolutionary Monte Carlo method⁹¹ for efficient sampling, which incorporates powerful interactions used in genetic algorithms. The FAMC method is based on a population-based Monte Carlo method with evolutionary operators (i.e., mutation, crossover, and exchange) for the fragment-based structural alteration of molecules.

Let $k_{\text{AE},h}$ be the atom environment (AE) kernel on a set \mathcal{G} of molecular graphs. The AE kernel $k_{\text{AE},h}$ induces a map $\phi : \mathcal{G} \rightarrow \mathcal{F}$ into a feature space \mathcal{F} such that $k(G, G') = \langle \phi(G), \phi(G') \rangle$ for all $G, G' \in \mathcal{G}$. Given an image point Ψ in \mathcal{F} as an expansion in terms of known molecules $\{G_1, \dots, G_N\} \subseteq \mathcal{G}$, i.e., $\Psi = \sum_{i=1}^N \alpha_i \phi(G_i)$, the pre-image problem reduces to finding a corresponding molecular graph $G^* \in \mathcal{G}$ such that $\Psi = \phi(G^*)$. However, no such G^* exists for many $\Psi \in \mathcal{F}$ since the map ϕ is not surjective.⁹² A relaxation of the pre-image problem is to find an approximate pre-image G^* , such that the squared distance of Ψ and $\phi(G)$ is minimized,

$$G^* = \arg \min_G (\|\Psi - \phi(G)\|^2 + \eta R(G)) =: \arg \min_G H(G), \quad (4.1)$$

where $R(G)$ is a regularization function to penalize non-drug-like molecules using knowledge of drug-likeness commonly used by medicinal chemists, and η controls the strength of the regularization. We denote the energy function for a molecular graph G by $H(G)$. Using the normalized AE kernel $\tilde{k}_{\text{AE},h}$, we can rewrite $H(G)$ as

$$H(G) = -2 \sum_{i=1}^N \alpha_i \tilde{k}_{\text{AE},h}(G_i, G) + \eta R(G) + C, \quad (4.2)$$

where $C = \sum_{i,j=1}^N \alpha_i \alpha_j \tilde{k}_{\text{AE},h}(G_i, G_j) + 1$ is a constant independent of G .

As mentioned previously, in computer-aided molecular design, medicinal chemists are not only interested in computationally optimal molecules, but also in their neighboring sub-optimal molecules since the suboptimal molecules are often expected to be good candidates for further knowledge-based prioritization. Therefore, we state the pre-image problem as a sampling problem. This is different from the optimization problem (eq 4.1) where only optimal solutions are of interest. The formulation of the sampling problem begins by defining

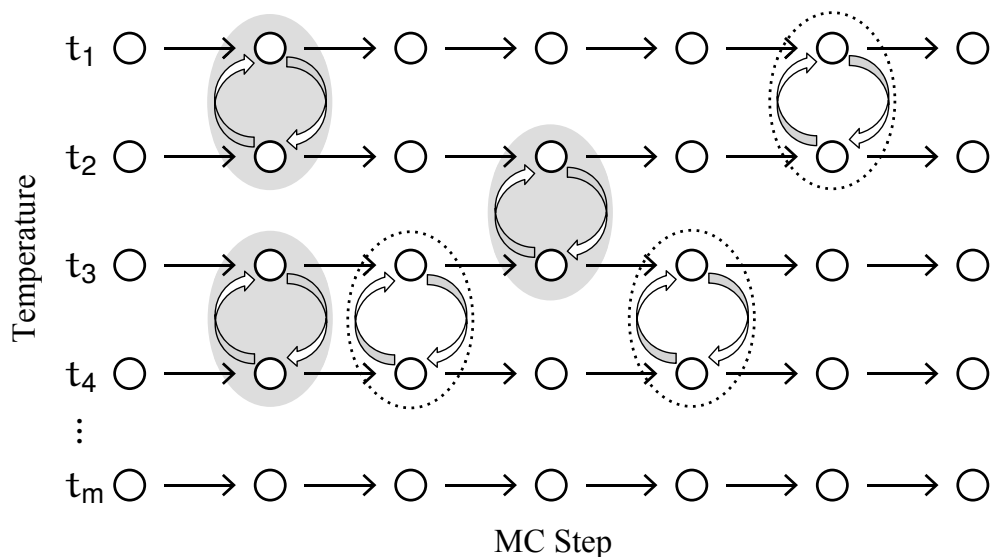


Figure 4.1 A graphical representation of the FAMC method with the parallel tempering scheme. The algorithm works by evolving a population of molecules in parallel, where a different temperature t_i is assigned to each molecule.

a target distribution of interest

$$G^* \sim \pi(G) \propto \exp\{-H(G)/t\}.$$

This is the Boltzmann distribution with energy function $H(G)$ given by eq 4.2 at temperature t . To sample structurally diversified molecules near optimal solutions, we then conduct a random sampling from $\pi(G)$ using the FAMC method. The FAMC method proceeds by evolving a population of molecules in parallel, where a different temperature is assigned to each molecule (Figure 4.1). The population is updated by mutation (partial structure alteration using a molecular fragment database), crossover (partial structure swapping between two molecules), and exchange (whole structure swapping between two molecules) operators (Figure 4.2). Of these operators, the crossover operator typically used in genetic algorithms^{93,94} specifically causes powerful interactions among the molecules in the population,⁹¹ as will be demonstrated in Section 4.5. If the crossover operator is not used, FAMC follows the parallel tempering^{53–55} (PT) algorithm.

Let $\mathbf{G} = \{G_1, G_2, \dots, G_m\}$ denote a population where G_i is a molecular graph called an individual and m is the population size. A set of m different temperatures, $\mathbf{t} = \{t_1, t_2, \dots, t_m\}$, are given and ordered as $t_1 > t_2 > \dots > t_m$. Each individual G_i in the population has a temperature t_i for $i = 1, \dots, m$. The corresponding Gibbs distribution for each individual G_i is

$$\pi_i(G_i) = \frac{1}{Z_i(t_i)} \exp\{-H(G_i)/t_i\},$$

where $Z_i(t_i)$ is the normalizing constant, $Z_i(t_i) = \sum_{\{G_i\}} \exp\{-H(G_i)/t_i\}$. By letting the lowest temperature $t_m = t$, π_m corresponds to the target distribution $\pi(\mathbf{G})$. In FAMC, the target distribution of the population \mathbf{G} is defined as the augmented Boltzmann distribution

$$\pi(\mathbf{G}) = \frac{1}{Z(\mathbf{t})} \exp\left\{-\sum_{i=1}^m H(G_i)/t_i\right\},$$

where $Z(\mathbf{t}) = \prod_{i=1}^m Z_i(t_i)$.

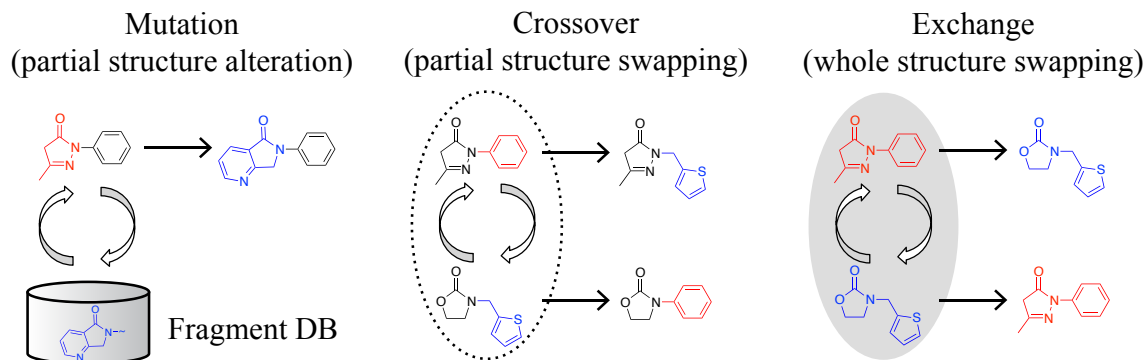


Figure 4.2 Evolutionary operations for the structural alteration of molecules. Mutation: partial structural alteration using a molecular fragment database. Crossover: partial structural swapping between two molecules. Exchange: whole structural swapping between two molecules.

4.2 Evolutionary Movements in Chemical Space

In this section we describe the evolutionary operators (i.e., mutation, crossover, and exchange) for the structural alteration of molecules.

Mutation

The mutation operation is achieved by a Metropolis–Hastings step. An individual G_k is first selected at random from the current population G . G_k is then mutated to G'_k by the following procedure (Figure 4.3):

1. The selected G_k is decomposed into a set $S(G_k)$ of all possible fragments subject to the fragment constraints that the fragments consist of 4–18 heavy atoms and any ring bond cannot be broken. The decomposition is performed using GASTON,^{95–97} an

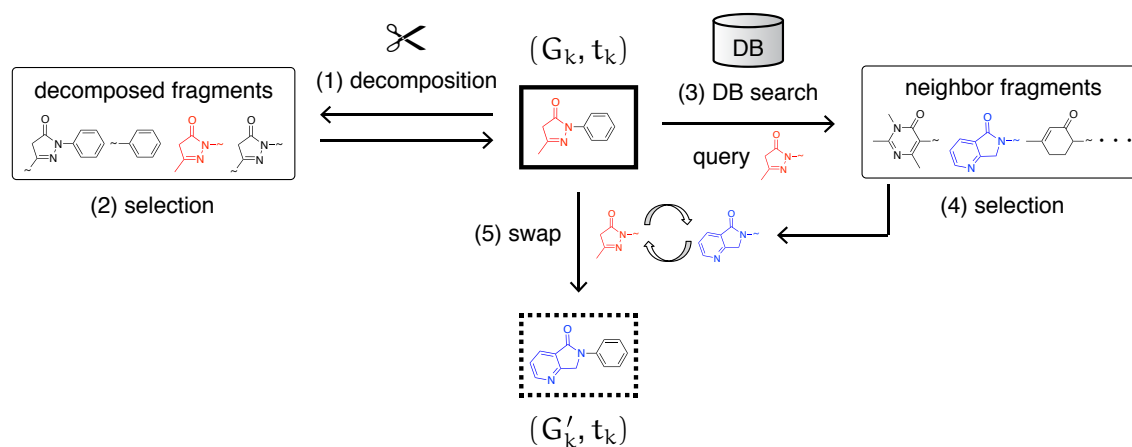


Figure 4.3 A graphical illustration of the mutation operation on molecules. Consider the transition from a molecule G_k to G'_k . The operation involves five steps. (1) The molecule G_k is decomposed into a set $S(G_k)$ of fragments. Let N_{dec} be the number of the decomposed fragments. (2) A fragment is selected at random with probability $\frac{1}{N_{\text{dec}}}$ from $S(G_k)$. This is a candidate to be renewed in G_k . (3) Neighbors of the selected fragment are retrieved from a molecular fragment database prepared in advance. Let N_{nbr} be the number of the retrieved neighbor fragments. (4) A fragment for replacement is randomly selected with probability $\frac{1}{N_{\text{nbr}}}$ from the neighborhoods. (5) A new candidate molecule G'_k is generated by swapping the selected fragment of G_k with the neighboring fragment in the database.

- efficient graph-based substructure mining algorithm. Let N_{dec} be the number of the decomposed fragments.
2. A fragment is selected at random with probability $\frac{1}{N_{\text{dec}}}$ from $\mathcal{S}(G_k)$. This is a candidate to be renewed in G_k .
 3. The ϵ -neighborhoods of the selected fragment are retrieved out of a molecular fragment database prepared in advance. The neighborhoods are identified by a fast similarity search of Morgan fingerprints (ECFP-like fingerprints) stored in a succinct multibit tree.^{98,99} The number N_{nbr} of the neighborhoods depends on a given threshold ϵ for the Tanimoto similarity metric.³² Let N_{nbr} be the number of the retrieved neighbor fragments.
 4. A fragment for replacement is then randomly selected with probability $\frac{1}{N_{\text{nbr}}}$ from the neighborhoods.
 5. A new candidate molecule G'_k is generated by swapping the selected fragment of G_k with the neighboring fragment in the database.

Accordingly, a new population is proposed as $\mathbf{G}' = \{G_1, \dots, G'_k, \dots, G_m\}$, and it is accepted with probability $\min(1, r_m)$, according to the Metropolis–Hastings rule, where

$$r_m = \frac{\pi(\mathbf{G}')T(\mathbf{G}|\mathbf{G}')}{\pi(\mathbf{G})T(\mathbf{G}'|\mathbf{G})} = \exp\{-(H(G'_k) - H(G_k))/t_k\} \frac{N_{\text{dec}} N_{\text{nbr}}}{N'_{\text{dec}} N'_{\text{nbr}}}. \quad (4.3)$$

N'_{dec} and N'_{nbr} are computed in the backward step from \mathbf{G}' to \mathbf{G} . Here, the transition probability $T(\mathbf{G}'|\mathbf{G})$ is asymmetric, i.e., $T(\mathbf{G}'|\mathbf{G}) \neq T(\mathbf{G}|\mathbf{G}')$. It should be mentioned that, in the following experiments, we imposed $N_{\text{dec}} N_{\text{nbr}} / N'_{\text{dec}} N'_{\text{nbr}} = 1$ on the above equation due to the low sampling efficiency which is caused by an imbalance between N_{nbr} and N'_{nbr} . The imbalance is attributed to the heterogeneity of the fragment database. This problem of breaking the detailed balance will be discussed in Section 4.6.

Crossover

First, two parent individuals, G_i and G_j ($i \neq j$), are selected from the current population $\mathbf{G} = \{G_1, \dots, G_i, \dots, G_j, \dots, G_m\}$ according to a roulette wheel selection procedure.¹⁰⁰ Without loss of generality, we assume $H(G_i) \geq H(G_j)$. Two new offspring individuals are then generated from the two parent individuals by the following procedure (Figure 4.4):

1. The selected G_i and G_j are decomposed into two sets, $\mathcal{S}(G_i)$ and $\mathcal{S}(G_j)$, of all possible fragments, respectively, subject to the fragment constraints. The decompositions are performed using GASTON.
2. A list is given of similar fragment pairs between $\mathcal{S}(G_i)$ and $\mathcal{S}(G_j)$. The list is obtained by finding similar fragment pairs from all possible fragment pairs between $\mathcal{S}(G_i)$

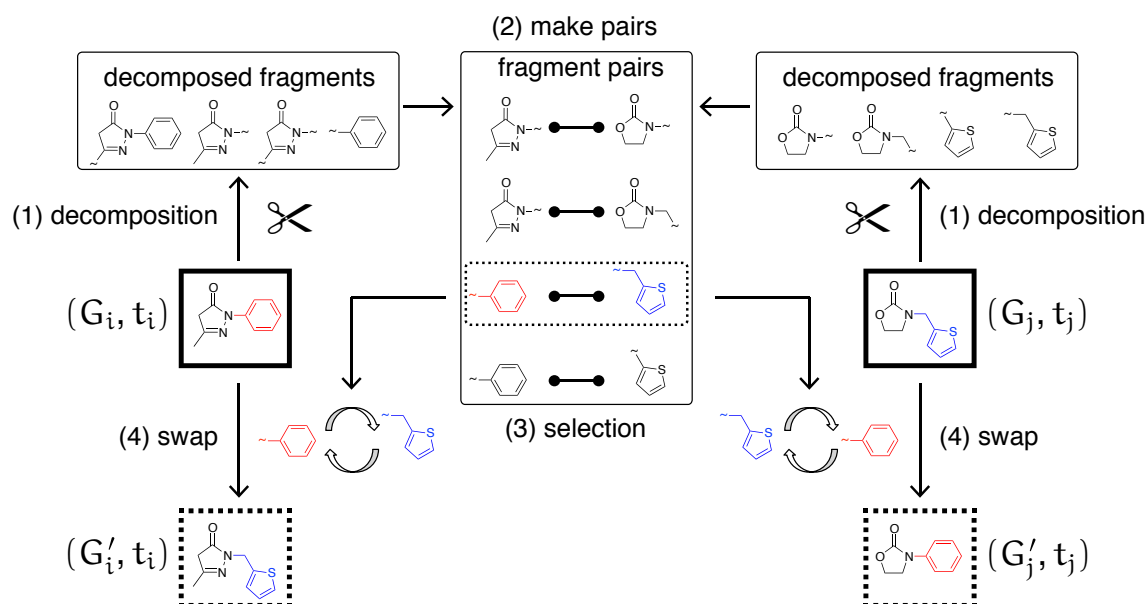


Figure 4.4 A graphical illustration of the crossover operation on molecules. Consider the generation of two new offsprings G'_i and G'_j from one molecular pair G_i and G_j ($i \neq j$). The operation involves four steps. (1) The two molecules G_i and G_j are decomposed into two fragment sets $\mathcal{S}(G_i)$ and $\mathcal{S}(G_j)$, respectively. (2) A list of similar fragment pairs between $\mathcal{S}(G_i)$ and $\mathcal{S}(G_j)$ is given. Let N_{pair} be the number of pairs. (3) One pair is randomly selected with probability $\frac{1}{N_{\text{pair}}}$ from the fragment pair list. (4) Two new offspring G'_i and G'_j are generated by swapping the paired fragments in G_i and G_j .

and $\mathcal{S}(G_j)$ for a given similarity threshold ϵ . The similarity is measured using the Tanimoto coefficient and Morgan fingerprints. Let N_{pair} be the number of the pairs.

3. One pair is randomly selected with probability $\frac{1}{N_{\text{pair}}}$ from the fragment pair list.
4. Two new offspring, G'_i and G'_j , are generated by swapping the above-selected paired fragments in G_i and G_j .

As a result, a new population is proposed as $\mathbf{G}' = \{G_1, \dots, G'_i, \dots, G'_j, \dots, G_m\}$, and it is accepted with probability $\min(1, r_c)$, according to the Metropolis–Hastings rule, where

$$r_c = \frac{\pi(\mathbf{G}')T(\mathbf{G}|\mathbf{G}')}{\pi(\mathbf{G})T(\mathbf{G}'|\mathbf{G})} = \exp\{-(H(G'_i) - H(G_i))/t_i - (H(G'_j) - H(G_j))/t_j\} \frac{N_{\text{pair}}}{N'_{\text{pair}}}.$$

N'_{pair} is computed in the backward step. In this case, the transition probability $T(\mathbf{G}'|\mathbf{G})$ is asymmetric.

Exchange

The exchange operation is used to swap individuals between parallel tempered chains. Given the current population \mathbf{G} and the corresponding temperature ladder \mathbf{t} , $(\mathbf{G}, \mathbf{t}) = \{G_1, t_1, \dots, G_m, t_m\}$, we first select two adjacent individuals G_i and G_j at random and then attempt

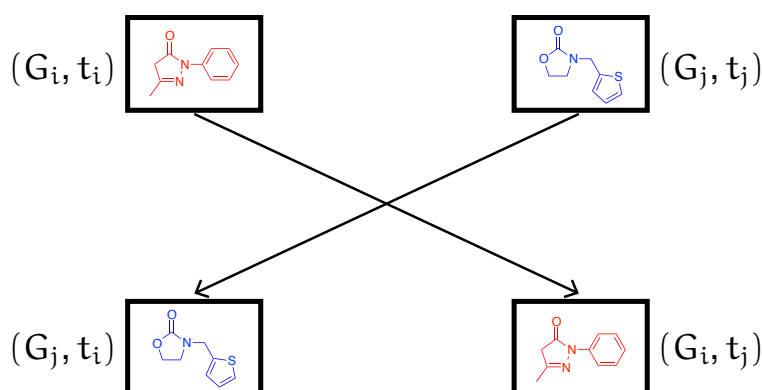


Figure 4.5 A graphical illustration of the exchange operation on molecules. This operation is conducted by swapping neighboring chains randomly chosen from the molecule population.

to move $(\mathbf{G}, \mathbf{t}) = \{G_1, t_1, \dots, G_i, t_i, G_j, t_j, \dots, G_m, t_m\}$ to $(\mathbf{G}', \mathbf{t}) = \{G_1, t_1, \dots, G_j, t_j, G_i, t_i, \dots, G_m, t_m\}$ (see Figure 4.5). This move is accepted with probability $\min(1, r_e)$, according to the Metropolis-Hastings rule, where

$$r_e = \frac{\pi(\mathbf{G}')T(\mathbf{G}|\mathbf{G}')}{\pi(\mathbf{G})T(\mathbf{G}'|\mathbf{G})} = \exp\{(H(G_i) - H(G_j))(\frac{1}{t_i} - \frac{1}{t_j})\}.$$

Here, the transition probability $T(\mathbf{G}'|\mathbf{G})$ is symmetric.

Algorithm

In summary, using the evolutionary operations described above, the FAMC algorithm works as follows. Given an initial population $\mathbf{G} = \{G_1, G_2, \dots, G_m\}$ and a temperature ladder $\mathbf{t} = \{t_1, t_2, \dots, t_m\}$, FAMC iterates over the following two steps:

1. Apply either the mutation or crossover operator to the current population with probability q_m and $1 - q_m$, respectively. q_m is called the mutation rate.
2. Apply the exchange step. An individual G_i is first selected at random from \mathbf{G} and is subjected to exchange with one of its neighbors. This operation is iterated N_m times.

4.3 Preparation of Molecular Fragments

We prepared a database for the mutation operation containing over four million molecular fragments. The database was prepared as follows (Figure 4.6): molecular fragments are extracted from all compounds in the ChEMBL database¹⁰¹ (release 17) using GASTON,^{95–97} subject to the fragment constraints. We then removed duplicated and undesirable fragments. That is, we first removed molecules with reactive, toxic, and otherwise undesirable structural motifs^{102–104} (e.g. aldehydes, Michael acceptors, imines, etc.) and then filtered the remaining molecules for their fragment-likeness.^{105,106} The resultant molecular fragments were converted to Morgan fingerprints (similar to ECFP fingerprints³¹) using

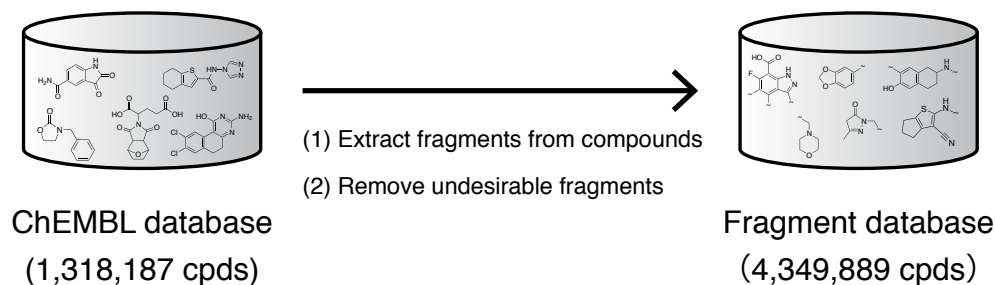
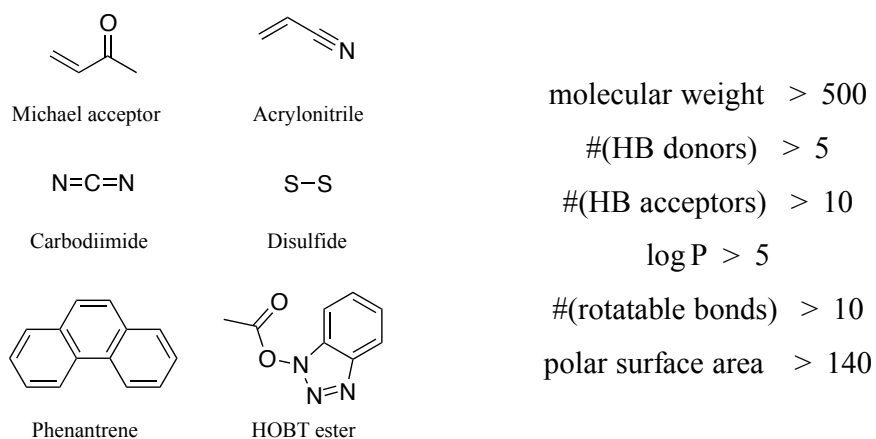


Figure 4.6 Procedure for the preparation of molecular fragments. Molecular fragments with attachment points are extracted from the compounds in the ChEMBL database and subsequently filtered to remove fragments with undesirable substructures or properties.

the RDKit,¹⁰⁷ which were stored in a succinct multibit tree^{98,99} for fast similarity searching. They were also converted to the SMILES chemical structure line notations^{108,109} using OpenBabel,^{66,67} which were then stored in a NoSQL database¹¹⁰ for fragment structure retrieval.

4.4 Regularization of Molecules

We collected a set \mathcal{R} of rules to identify non-drug-like molecules from the literature^{102–104,111–113} (see Figure 4.7). In order to penalize non-drug-like molecules sampled from the target dis-



(a) undesirable substructures (573 rules) (b) undesirable properties (21 rules)

Figure 4.7 A set of (a) undesirable substructures and (b) undesirable properties commonly used by medicinal chemists. This set is used to penalize non-drug-like molecules.

tribution $\pi(G)$, we give a regularization function

$$R(G) = \sum_{r \in \mathcal{R}} M(G, r),$$

where $M(G, r)$ is 1 if the molecule G matches the rule r and 0 otherwise.

4.5 Experiments

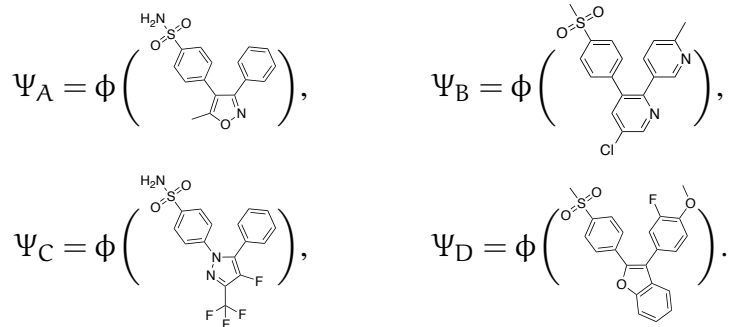
We demonstrate the effectiveness of the FAMC method by conducting experiments to find corresponding molecules from given image points in the feature space \mathcal{F} induced by the AE kernel. The pre-image reconstruction capability of the FAMC method is then presented. The efficiency of the crossover operation is also evaluated.

4.5.1 Experimental Settings

Molecular Reconstruction

We first performed a simple experiment to reconstruct a corresponding molecule $G^* \in \mathcal{G}$ such that $\Psi = \phi(G^*)$ from a given image point Ψ in \mathcal{F} .

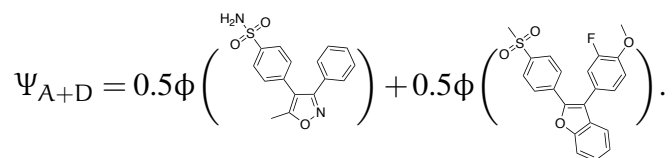
The reconstruction capability of the FAMC methods were tested with four different image points, Ψ_A , Ψ_B , Ψ_C , and Ψ_D ,



These four image points were implicitly mapped from the four COX-2 inhibitors by the AE kernel, respectively.


Molecular Interpolation

In addition, we performed an alternative experiment to design intermediate molecules between two known COX-2 inhibitors. This experiment was conducted by considering the image point Ψ_{A+D} ,



This image point is defined by a linear combination of the two COX-2 inhibitors.

Algorithm Settings

In all of the simulations, the FAMC scheme was implemented using the following settings: FAMC was run for 10,000 iterations with a population size of 32. The initial individuals in the population are all set to the molecular graph of toluene, . The inverse temperature ladder was configured as $\{1, 5, 9, \dots, 125\}$. The strength η of molecular regularization is set to a large value of 10000. The similarity threshold ϵ is set to 0.38. The mutation rate q_m was chosen from $\{0.75, 1.00\}$. Setting $q_m = 1.00$, FAMC is reduced to parallel tempering (PT). All of the AE kernel parameters used herein were set to the optimized values which were obtained when building the forward prediction model on the COX-2 data set (see Table 3.4). The simulation results were analyzed over the 10,000 sampling points in the chain with sampled molecules ($1/t = 125$).

The FAMC algorithm was coded in C++ using the OpenBabel toolbox.^{66,67} The simulations were conducted on an Intel Xeon X5570 2.93GHz system with 32GB of main memory.

The communication between the processors was achieved using the Open Message Passing Interface¹¹⁴ (Open MPI), which is an open source MPI implementation.

4.5.2 Results and Discussion

Molecular Reconstruction

We present the simulation results for the molecular reconstruction experiments.

Figure 4.8 shows the time series plots of the squared distance of each target point to the corresponding sample points in \mathcal{F} . In all cases, the distance tended to zero until at least 6,000 iterations. The achievement of zero distance indicates that the molecular reconstruction is complete. It can also be seen from the figure that the chain with the lowest temperature fluctuates around the given image point after the achievement of zero distance. In all experiments, FAMC was run for 10,000 iterations within 14 hours. The hot spot of

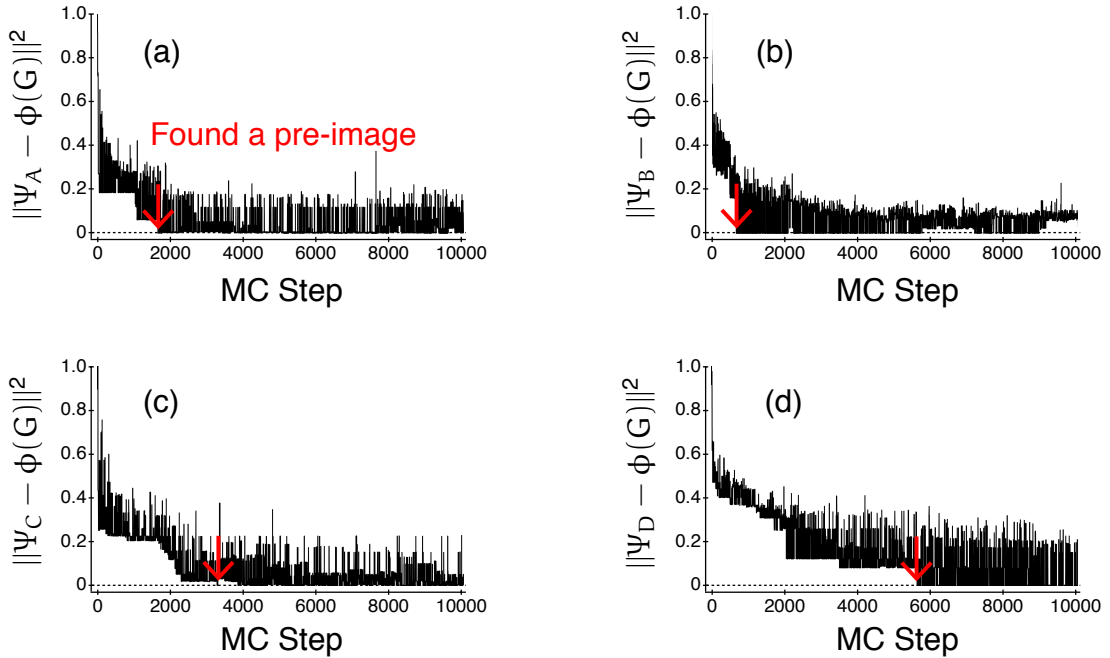


Figure 4.8 Time series plots of the squared distance between four target points and the corresponding sampling points in \mathcal{F} . We consider four target points, (a) Ψ_A , (b) Ψ_B , (c) Ψ_C , and Ψ_D . The red arrow indicates the position where an exact pre-image was found.

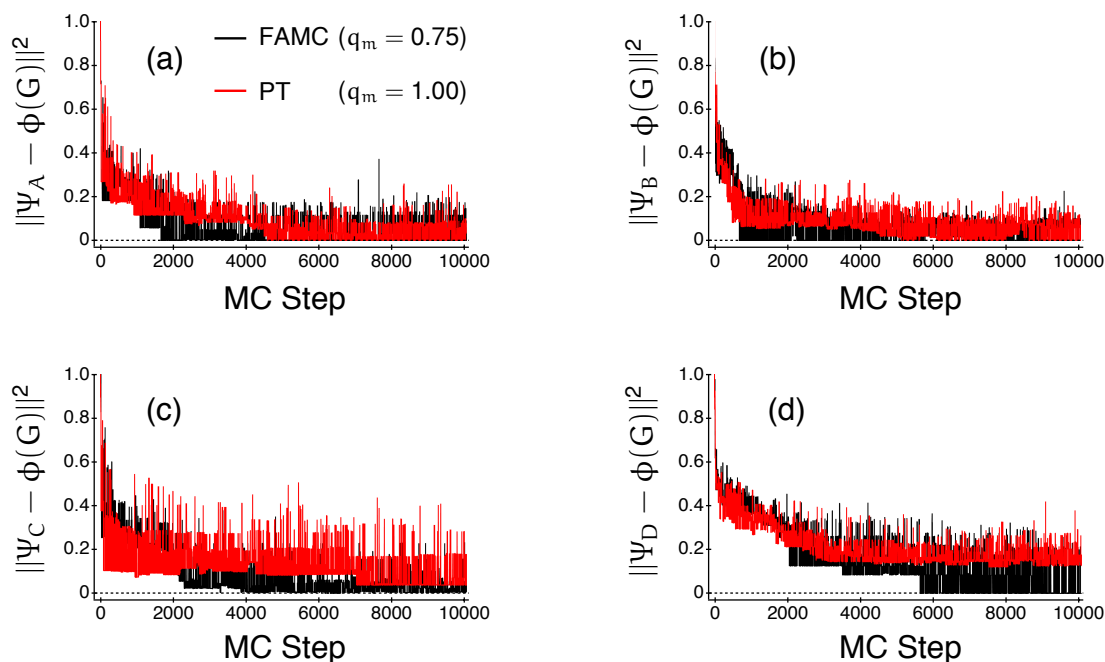


Figure 4.9 Comparison of the pre-image reconstruction capability for given feature vectors, Ψ_A , Ψ_B , Ψ_C , and Ψ_D , between FAMC (black line) and PT (red line). In all experiments, FAMC found the pre-images more quickly than PT. Unfortunately, PT failed on the two experiments.

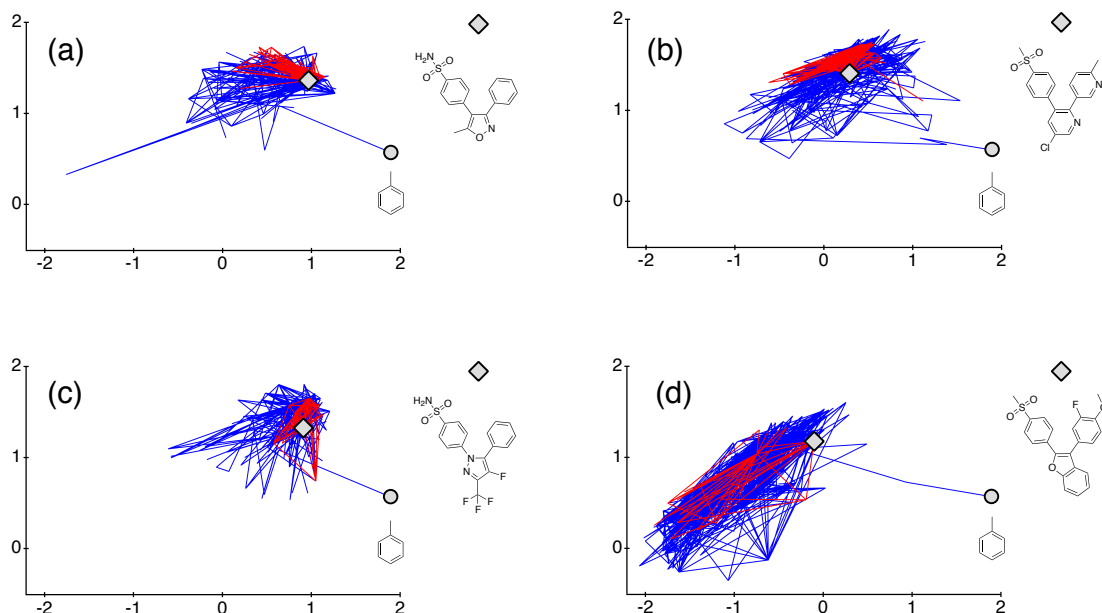


Figure 4.10 Sampling paths from toluene (grey filled circle) to the four known COX-2 inhibitors (grey filled diamond) in \mathcal{G} . The paths of the last 2000 steps are drawn in red.

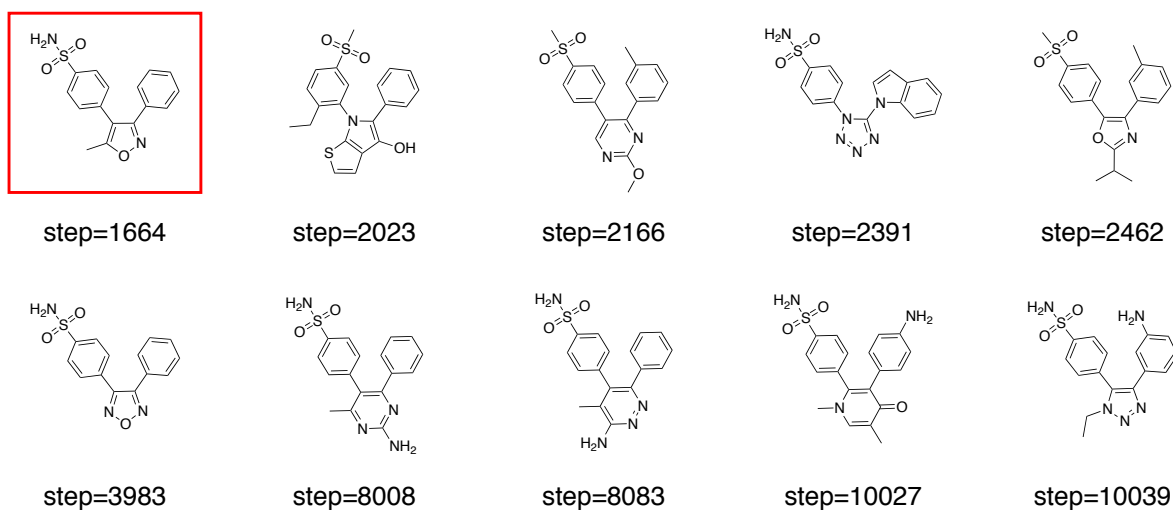



Figure 4.11 Structurally diversified molecules near the pre-image, which are sampled by FAMC in order to reconstruct a corresponding molecule (pre-image) from the feature vector Ψ_A . FAMC found the pre-image, highlighted by a red frame, at step 1664.

FAMC, where the most time was spent during the execution, is the neighbor retrieval from the fragment database, which is performed in the mutation operation. The time taken for the retrieval could be reduced by clustering fragments into groups (more details in Section 4.6).

In Figure 4.9 the importance of the crossover operator is shown by comparing the simulation results of FAMC with crossover ($q_m = 0.75$) and PT without it ($q_m = 1.00$). In all experiments, FAMC succeeded in reconstructing the pre-image more quickly than PT. Unfortunately, PT failed in two cases (Figure 4.9c and d). The success of the crossover operator arises from powerful interactions among molecules in the population. As a result, the crossover operator is beneficial to find the pre-images for molecules in FAMC.

Figure 4.10 displays the time-series sequence of sampled molecules resulting from the 10,000 iterations of FAMC. Here, the sampled molecules were projected into the chemical space obtained by the kernel ISOMAP.^{115,116} This figure shows that, despite starting from the distant solution (i.e., ) , the chain of sampled molecules succeeded in reaching the pre-image point for any four experiments. Furthermore, as shown in Figure 4.11, the sampled molecules exhibit considerable structural diversity while sharing some structural features

with the target molecule, and also maintaining drug-likeness.

Molecular Interpolation

We present the simulation results for the design of intermediate molecules between the two COX-2 inhibitors.

Trajectory plots of molecules both in \mathcal{F} and \mathcal{G} , sampled by FAMC, are shown in Figure 4.12. As shown in Figure 4.12a, the squared distance of the target point Ψ_{A+D} to every sample point $\phi(G)$ never reaches zero. One explanation for the failure is that no exact pre-image exists in the limited chemical space of drug-like molecules. Figure 4.12b shows the transition of the sampled molecules in the chemical space dimensionally reduced by the kernel ISOMAP.^{115,116} Once the sample sequence reached a near-optimal solution, it remained within the region between the two COX-2 inhibitors. Figure 4.13 shows that FAMC samples structurally diversified molecules with good drug-like characteristics, in which the structural features of the two COX-2 inhibitors are intricately intertwined. The molecular interpolation approach has potential applications to drug design. For example, if we suppose there are two molecules with different desired properties, intermediate molecules are

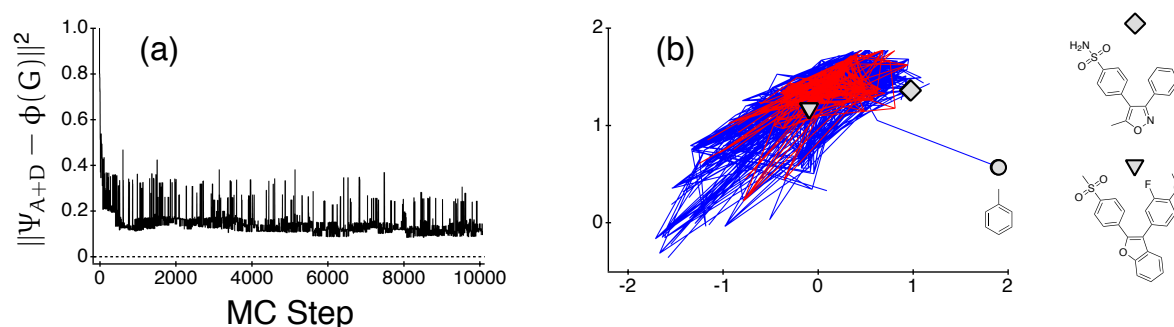


Figure 4.12 Trajectory plots of molecules sampled by FAMC for the experiment to interpolate between two COX-2 inhibitors using the linear combination Ψ_{A+D} of the two feature vectors Ψ_A and Ψ_D . (a) A time series plot of the squared distance between the target point Ψ_{A+D} and every sample point in \mathcal{F} . Note that the distance never reached zero. (b) A sampling path from the toluene (grey filled circle) to the midpoint between two COX-2 inhibitors (grey filled diamond and inverted triangle) in \mathcal{G} . The path of the last 2000 steps is drawn in red.

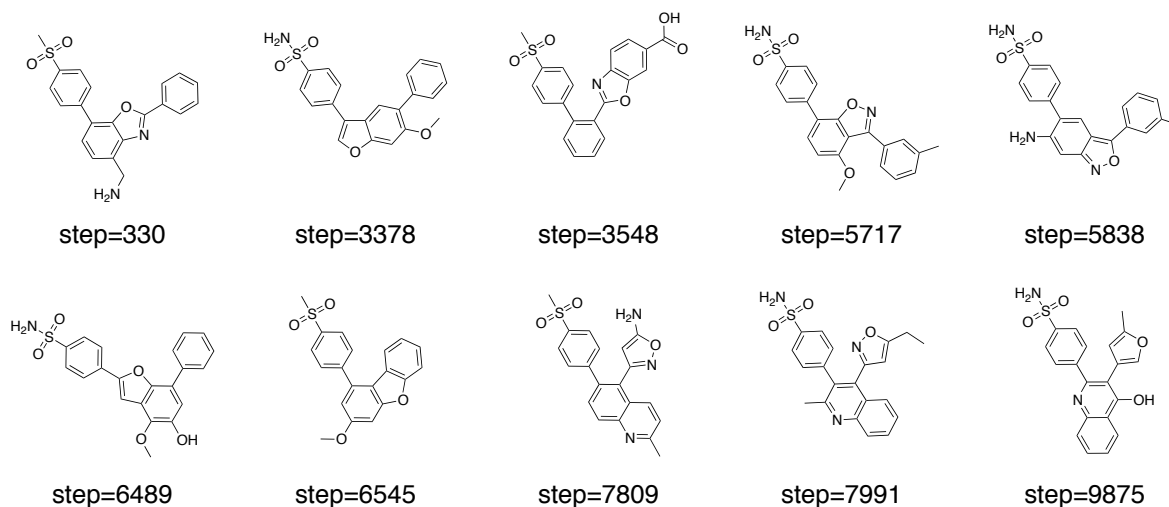


Figure 4.13 Structurally diversified molecules sampled so as to synthesize the structural features of the two COX-2 inhibitors.

expected to be promising candidates possessing both properties.

In comparison to previous research^{2,28} into graph enumeration and optimization, the proposed method has the following characteristics: (i) it is designed to generate not only optimal molecules but also structurally diversified molecules near the optimal molecules. (ii) The generated molecules have good drug-like properties to be acceptable for the common structural filters and physicochemical filters that were unconsidered in previous research.

4.6 Concluding Remarks

We developed a population-based Monte Carlo method to solve the pre-image problem for molecules. All simulation results demonstrated the effectiveness of the proposed method to sample structurally diversified molecules near the pre-images, which possess good drug-like characteristics. The effectiveness of the proposed method was due to two main reasons. For efficient sampling, we used evolutionary operators of the genetic algorithms for efficient structural alteration of molecules. Simulation with the crossover operator led to faster reconstruction of given pre-images than those without it. In addition, in order to penalize

non-drug-like molecules, we used the knowledge of drug-likeness commonly considered by medicinal chemists.

One possible extension to the FAMC method is to satisfy the detailed balance condition while improving computational efficiency. What matters here is the heterogeneity of the fragment database for mutation. The heterogeneity causes an imbalance between N_{nbr} and N'_{nbr} in eq 4.3. In order to compensate the imbalance, it is necessary that the fragments are uniformly distributed across the database. We intend to redesign the database such that $N_{\text{nbr}} = N'_{\text{nbr}}$ by the following steps: The fragments are first mapped into two-dimensional grid cells using the self-organizing map¹¹⁷ (SOM). We then build a uniform database by adding dummy fragments to each cell such that the number of fragments is equal in all cells. If a dummy fragment is chosen for replacement in a mutation trial, the trial is always rejected. Another important extension is to penalize molecules with difficult synthetic accessibility.¹¹⁸

Chapter 5

Conclusions

The primary aim of this thesis was to develop a data-driven method for the *de novo* design of new molecules that yield suitable properties required for drug candidates. *De novo* design remains a computationally challenging problem due to the ill-posed nature (i.e., non-convex, nonlinear, and combinatorial) of the problem, which, if solved, could help accelerate the development of new drugs. The major hurdles for *de novo* design are: the accurate prediction of molecular properties of interest and the efficient exploration of the chemical space of possible molecules with good drug-likeness and synthetic accessibility. Failure to overcome these hurdles could require costly efforts to synthesize and test new complex molecules, which ultimately may not have the desired properties. In this thesis, we developed a kernel-based strategy to address these hurdles. The strategy involved two steps. First, we embedded molecules into a feature space amenable to the prediction of molecular properties. This embedding was achieved by defining a new graph kernel specialized for molecules. Second, we designed target molecules through the reconstruction of corresponding molecules (aka pre-images) from an image point in the prepared feature space, reflecting desired properties. This so-called pre-image problem is of central importance to *de novo* design. We expressed the pre-image problem as a sampling problem in order to retrieve structurally diversified molecules near the pre-images. We next summarize each step in the *de novo* design strategy and sketch some directions for future work.

To construct a feature space where it is easy to predict molecular properties, we tailored a new graph kernel to molecules by extending the existing subtree kernel.¹ The proposed kernel tackled two primary limitations of conventional graph kernels: (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. The proposed kernel first permitted an 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. In addition, we introduced the tree weight function to assign an importance weight to each tree-pattern according to the statistical significance of 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 to the task. The proposed kernel either outperformed, or was at least competitive with, existing standard graph kernels and molecular fingerprints over all the learning tasks we considered.

To suggest new molecules with desired properties, it is necessary to solve the pre-image problem. Unlike a traditional method proposed in ref 2 which relies on nonlinear combinatorial optimization, we expressed the pre-image problem as a sampling problem, where we are interested not only in optimal molecules, but also in near-optimal molecules. Therefore, we developed a population-based Monte Carlo method for sampling structurally diversified molecules with good drug-likeness near the pre-images. The key to efficient sampling was to use the update of a population by evolutionary operators for the structural alteration of molecules. In addition, in order to penalize non-drug-like molecules, we used the knowledge of drug-likeness commonly considered by medicinal chemists. The effectiveness of the proposed method was illustrated through experiments to find pre-images for several molecules. Inherently, the synthetic accessibility has to be evaluated for sampled molecules, but we leave this for future work.

An important extension to this work that is yet to be developed is to construct a pipeline between the forward prediction of chemical properties and the inverse prediction of chemical structure designs. A promising line of research for this future work is to devise the energy function (eq 4.2). Given a forward prediction model built by, for example, the support vector regression

$$\hat{y}(G) = \sum_{G' \in SV} \alpha_i \tilde{k}_{AE,h}(G, G') + b,$$

we can define an alternative energy function of the form

$$H_r(G) = (y^* - \hat{y}(G))^2 + \gamma R(G),$$

where $y^* \in \mathbb{R}$ is set to the desired value of the output variable of interest. Similarly, if $\hat{y}(G)$ is a support vector machine classifier, we have the energy function for classification problems

$$H_c(G) = -y^* \hat{y}(G) + \gamma R(G),$$

with a given class label $y^* \in \{-1, 1\}$. Furthermore, given two regression models $\hat{y}_A(G)$ and $\hat{y}_B(G)$, we have the energy function for multi-objective problems

$$H_m(G) = \alpha(y_A^* - \hat{y}_A(G))^2 + (1 - \alpha)(y_B^* - \hat{y}_B(G))^2 + \gamma R(G),$$

where $y_A^*, y_B^* \in \mathbb{R}$ are desired property values and α is a weighting coefficient in the range $0 \leq \alpha \leq 1$. We can then design new molecules with desired properties by sampling molecules from the target distribution with the energy function $H_r(G)$ for regression problems, $H_c(G)$ for classification problems, or $H_m(G)$ for multi-objective problems.

We believe that our contributions have the potential to explore new avenues in data-driven drug design.

Appendix A

Derivation of the Recursive Formula

We derive the recursive form (eq 3.5) of the AE kernel (eq 3.1) 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.,

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

With this set of tree-patterns, the AE kernel in eq 3.1 between G and G' with respect to any tree $T \in \mathcal{T}_h$ up to height h can be rewritten as

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

The term in brackets in the above equation corresponds to $k_h(v, v')$ in eq 3.5, i.e.,

$$k_h(v, v') = \sum_{T \in \mathcal{T}_h} \sum_{p \in \mathcal{P}_T^{(v)}(G)} \sum_{p' \in \mathcal{P}_T^{(v')}(G')} \prod_{(u, u') \in \mathcal{A}(p, p')} \hat{w}(a_r(u)) \hat{w}(a_r(u')) k_{\text{atom}}(e_r(u), e_r(u')). \quad (\text{A.1})$$

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

$$k_0(v, v') = \hat{w}(a_r(v)) \hat{w}(a_r(v')) k_{\text{atom}}(e_r(v), e_r(v')). \quad (\text{A.2})$$

For k_i , with $i = 1, \dots, h$, $\mathcal{A}(p, p')$ in eq A.1 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 A.1, we have

$$k_i(v, v') = k_0(v, v') \left(\sum_{T \in \mathcal{T}_i} \sum_{p \in \mathcal{P}_T^{(v)}(G)} \sum_{p' \in \mathcal{P}_T^{(v')}(G')} \prod_{\substack{(u, u') \in \mathcal{A}(p, p') \\ \setminus (v, v')}} k_0(u, u') \right). \quad (\text{A.3})$$

In the above equation, all pairs of children of the root vertices v and v' appear in the first element of $\mathcal{A}(p, p') \setminus (v, v')$. In other words, the term in brackets in eq A.3 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 A.3 becomes

$$\begin{aligned} k_i(v, v') &= k_0(v, v') \\ &\times \left[\sum_{R \in \mathcal{M}(v, v') + \emptyset} \prod_{(w, w') \in R} \left(\sum_{T \in \mathcal{T}_{i-1}} \sum_{p \in \mathcal{P}_T^{(w)}(G)} \sum_{p' \in \mathcal{P}_T^{(w')}(G')} \prod_{(u, u') \in \mathcal{A}(p, p')} k_0(u, u') \right) \right] \end{aligned} \quad (\text{A.4a})$$

$$\begin{aligned} &= k_0(v, v') \\ &\times \left[1 + \sum_{R \in \mathcal{M}(v, v')} \prod_{(w, w') \in R} \left(\sum_{T \in \mathcal{T}_{i-1}} \sum_{p \in \mathcal{P}_T^{(w)}(G)} \sum_{p' \in \mathcal{P}_T^{(w')}(G')} \prod_{(u, u') \in \mathcal{A}(p, p')} k_0(u, u') \right) \right]. \end{aligned} \quad (\text{A.4b})$$

In eq A.4a, we take the empty set \emptyset as a special case out of $\mathcal{M}(v, v')$. The product $\prod_{(w, w') \in R}$ is one if $R = \emptyset$, in order to treat unbalanced trees in the AE kernel. On the other hand, under the convention that the product is 0 if $R = \emptyset$, the AE kernel treats only balanced trees. It is straightforward to obtain eq A.4b from eq A.4a for the unbalanced trees. The term in parentheses in eq A.4b corresponds to $k_{i-1}(w, w')$ in eq 3.6. With eq A.2 for $i = 0$ and eq A.4b for $i > 0$, the derivation of the recursive formula is complete.

List of Figures

1.1	Illustration of the pre-image problem in kernel methods. An image point Ψ in the feature space \mathcal{F} is mapped back to the input space \mathcal{G}	2
2.1	The idea of kernel methods. This approach maps the training data in the input space \mathcal{G} into a high-dimensional feature space \mathcal{F} via the feature map $\phi : \mathcal{G} \rightarrow \mathcal{F}$, and applies linear machine learning algorithms such as SVM, which depend on the inner product $\langle \phi(G), \phi(G') \rangle$ between data points $\phi(G)$ and $\phi(G')$. Using the kernel trick $k(G, G') = \langle \phi(G), \phi(G') \rangle$, it is possible to apply them without explicitly mapping ϕ	11
2.2	A chemical structure (left) can be modeled as a labeled directed graph (right).	14
2.3	A molecular graph (left) and subtree patterns up to the height $h = 2$ rooted at the node v_1 (right). Note that the vertex v_1 appears at a height of 2 again.	14
2.4	The research efforts on graph kernels over the last decade.	15
2.5	A schematic concept of the convolution kernel between the molecular graphs G and G'	16
2.6	Given a molecule graph G , the traditional fingerprint is defined as a binary vector $\phi(G)$ such that it indicates the presence or absence of predefined particular substructures in G	21

3.1	Plots of the Gaussian kernel with a width parameter of $\gamma = 0.1$ (solid line) and the CS kernels $\psi_{2,c}^{(\theta)} \times \text{Gaussian}$ with respect to (a) the smoothing parameter c and (b) the cut-off distance θ	24
3.2	The modified Burden matrix of a substructure centered at v	29
3.3	In the ECFP algorithm, (a) the assignment of initial atom identifiers, computed by encoding seven atomic properties, (b) the generation of new atom identifiers by performing one iteration.	30
3.4	Prediction Performance Comparison of the AE Kernel with the Standard Graph Kernels and Molecular Fingerprint on 12 Benchmarks (see Table 3.3 in detail).	40
3.5	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.	42
3.6	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 cut-off 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.	44
3.7	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 3.4. . .	45

- 3.8 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 cut-off distances $\theta = 0.05$ (solid line), 0.20 (dashed line), and 0.50 (dashed-dotted line) and the ST kernel (dotted line). 47
- 4.1 A graphical representation of the FAMC method with the parallel tempering scheme. The algorithm works by evolving a population of molecules in parallel, where a different temperature t_i is assigned to each molecule. . . . 51
- 4.2 Evolutionary operations for the structural alteration of molecules. Mutation: partial structural alteration using a molecular fragment database. Crossover: partial structural swapping between two molecules. Exchange: whole structural swapping between two molecules. 52
- 4.3 A graphical illustration of the mutation operation on molecules. Consider the transition from a molecule G_k to G'_k . The operation involves five steps. (1) The molecule G_k is decomposed into a set $\mathcal{S}(G_k)$ of fragments. Let N_{dec} be the number of the decomposed fragments. (2) A fragment is selected at random with probability $\frac{1}{N_{\text{dec}}}$ from $\mathcal{S}(G_k)$. This is a candidate to be renewed in G_k . (3) Neighbors of the selected fragment are retrieved from a molecular fragment database prepared in advance. Let N_{nbr} be the number of the retrieved neighbor fragments. (4) A fragment for replacement is randomly selected with probability $\frac{1}{N_{\text{nbr}}}$ from the neighborhoods. (5) A new candidate molecule G'_k is generated by swapping the selected fragment of G_k with the neighboring fragment in the database. 53

- 4.4 A graphical illustration of the crossover operation on molecules. Consider the generation of two new offsprings G'_i and G'_j from one molecular pair G_i and G_j ($i \neq j$). The operation involves four steps. (1) The two molecules G_i and G_j are decomposed into two fragment sets $\mathcal{S}(G_i)$ and $\mathcal{S}(G_j)$, respectively. (2) A list of similar fragment pairs between $\mathcal{S}(G_i)$ and $\mathcal{S}(G_j)$ is given. Let N_{pair} be the number of pairs. (3) One pair is randomly selected with probability $\frac{1}{N_{\text{pair}}}$ from the fragment pair list. (4) Two new offspring G'_i and G'_j are generated by swapping the paired fragments in G_i and G_j 55
- 4.5 A graphical illustration of the exchange operation on molecules. This operation is conducted by swapping neighboring chains randomly chosen from the molecule population. 56
- 4.6 Procedure for the preparation of molecular fragments. Molecular fragments with attachment points are extracted from the compounds in the ChEMBL database and subsequently filtered to remove fragments with undesirable substructures or properties. 58
- 4.7 A set of (a) undesirable substructures and (b) undesirable properties commonly used by medicinal chemists. This set is used to penalize non-drug-like molecules. 58
- 4.8 Time series plots of the squared distance between four target points and the corresponding sampling points in \mathcal{F} . We consider four target points, (a) Ψ_A , (b) Ψ_B , (c) Ψ_C , and Ψ_D . The red arrow indicates the position where an exact pre-image was found. 61
- 4.9 Comparison of the pre-image reconstruction capability for given feature vectors, Ψ_A , Ψ_B , Ψ_C , and Ψ_D , between FAMC (black line) and PT (red line). In all experiments, FAMC found the pre-images more quickly than PT. Unfortunately, PT failed on the two experiments. 62

- 4.10 Sampling paths from toluene (grey filled circle) to the four known COX-2 inhibitors (grey filled diamond) in \mathcal{G} . The paths of the last 2000 steps are drawn in red. 62
- 4.11 Structurally diversified molecules near the pre-image, which are sampled by FAMC in order to reconstruct a corresponding molecule (pre-image) from the feature vector Ψ_A . FAMC found the pre-image, highlighted by a red frame, at step 1664. 63
- 4.12 Trajectory plots of molecules sampled by FAMC for the experiment to interpolate between two COX-2 inhibitors using the linear combination Ψ_{A+D} of the two feature vectors Ψ_A and Ψ_D . (a) A time series plot of the squared distance between the target point Ψ_{A+D} and every sample point in \mathcal{F} . Note that the distance never reached zero. (b) A sampling path from the toluene (grey filled circle) to the midpoint between two COX-2 inhibitors (grey filled diamond and inverted triangle) in \mathcal{G} . The path of the last 2000 steps is drawn in red. 64
- 4.13 Structurally diversified molecules sampled so as to synthesize the structural features of the two COX-2 inhibitors. 65

List of Tables

1.1	The Number of Molecules with Given Structural Constraints.	6
3.1	Two-way Contingency Table of Atom Environment Label α and Class Label c^a	26
3.2	Basic Information of the Data Sets Used Herein ^a	36
3.3	Prediction Performance Comparison of the AE Kernel with the Standard Graph Kernels and Molecular Fingerprint on 12 Benchmarks ^a	39
3.4	Parametrization of the AE Kernel with the Best Performance ^a	41
3.5	Computational Efficiency Comparison of the AE Kernel with the Standard Graph Kernels and Molecular Fingerprint on 12 Benchmarks	46

Symbols

Roman Symbols

\mathcal{G} an input space (or chemical space).

\mathcal{F} an RKHS (or feature space).

\mathcal{V}_G a set of vertices including in molecular graph G .

\mathcal{E}_G a set of edges including in molecular graph G .

G a molecular graph, $G = (\mathcal{V}_G, \mathcal{E}_G)$.

u, v vertices, $u, v \in \mathcal{V}_G$.

(u, v) the edge between two vertices u and v , $(u, v) \in \mathcal{E}_G$.

T a rooted tree, $T = (\mathcal{V}_T, \mathcal{E}_T)$.

$\mathcal{P}_T(G)$ a set of all possible tree-patterns of G arranged in T .

p a tree-pattern, $p \in \mathcal{P}_T(G)$.

\mathcal{T}_h a set of all trees up to height h .

h the height of tree T , i.e., the length of the longest path from the root to any other vertex.

$\mathcal{A}(p, p')$ a set of the aligned atom pairs of p and p' .

$\mathcal{M}(v, v')$ a set of subsets of neighborhood matching of vertices v and v' .

$\mathcal{N}(v)$ an outgoing neighborhood set of vertex v .

$\mathcal{S}(G)$ a set of parts extracted from molecular graph G .

$\alpha_r(v) \in \mathbb{Z}$ the atom environment label derived from a neighboring substructure of a topological radius r centered at vertex v using a variant of the Morgan algorithm.

$\mathbf{e}_r(v) \in \mathbb{R}^2$ the atom environment label derived from a neighboring substructure of a topological radius r centered at vertex v using an extension of the Burden approach.

r the topological radius for atom environment labels.

$I(p \cong p')$ the indicator function determines the isomorphism of tree-patterns p and p' .

$k_{\text{AE},h}(G, G')$ the AE kernel of molecular graphs G and G' , considering all trees up to height h .

$\tilde{k}_{\text{AE},h}(G, G')$ the normalized AE kernel.

$k_{\text{atom}}(\mathbf{e}_r(v), \mathbf{e}_r(v'))$ the atom-level kernel of atom environment labels $\mathbf{e}_r(v)$ and $\mathbf{e}_r(v')$.

$k_{\text{conv}}(G, G')$ the convolution kernel of molecular graphs G and G' .

$k_{\text{tree}}(p, p')$ the tree-level kernel of tree-patterns p and p' .

$w(p)$ a weight of tree-pattern p .

$\hat{w}(\alpha_r(v))$ a weight associated with the atom environment label $\alpha_r(v)$ of vertex v .

\mathbf{G} a population of m molecular graphs, $\mathbf{G} = \{G_1, G_2, \dots, G_m\}$.

\mathbf{t} a set of m different temperatures, $\mathbf{t} = \{t_1, t_2, \dots, t_m\}$.

$H(G)$ the energy function for molecular graph G .

$R(G)$ the regularization function to penalize non-drug-like molecules.

Greek Symbols

γ the width parameter of the Gaussian kernel.

λ_α a weight of significant atoms.

λ_β a weight of the other atoms.

ϕ a feature map.

$\psi_{d,c}(\cdot)$ a Wendland function for the dimension d of input variables and the smoothing parameter c .

Σ_V a set of vertex labels.

Σ_E a set of edge labels.

Σ a set of vertex and edge labels, $\Sigma_V \cup \Sigma_E$.

τ the threshold of χ^2 statistic for the determination of significant atoms.

θ the cut-off distance for the CS kernel.

η the strength of the regularization to penalize non-drug-like molecules.

$\pi(G)$ a target distribution of interest.

Ψ an image point in \mathcal{F} for the pre-image problem.

Other Symbols

\mathbb{N} the set of all natural numbers.

\mathbb{R} the set of all real numbers.

\mathbb{Z} the set of integer numbers.

Acronyms / Abbreviations

AE atom environment.

AUC area under the ROC curve.

CP cyclic pattern.

CS compactly supported.

ECFP extended-connectivity fingerprint.

ERW extended random walk.

EST extended subtree.

MCCV Monte Carlo cross-validation.

OA optimal assignment.

RKHS reproducing kernel Hilbert space.

ROC receiver operating characteristic.

RW random walk.

ST subtree.

WLST WeisfeilerLehman subtree.

FAMC fragment assembly Monte Carlo.

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] Bakır, G. H.; Zien, A.; Tsuda, K. Learning to find graph pre-images. *Lecture notes in computer science* **2004**, 253–261.
- [3] Bohacek, R. S.; McMartin, C.; Guida, W. C. The art and practice of structure-based drug design: A molecular modeling perspective. *Med. Res. Rev.* **1996**, *16*, 3–50.
- [4] Anonymous, The numbers game. *Nat. Rev. Drug Discov.* **2002**, *1*, 929.
- [5] Dobson, C. M. Chemical space and biology. *Nature* **2004**, *432*, 824–828.
- [6] Hansch, C.; Maloney, P. P.; Fujita, T. Correlation of biological activity of phenoxy-acetic acids with hammett substituent constants and partition coefficients. *Nature* **1962**, *194*, 178–180.
- [7] Kubinyi, H. Drug research: myths, hype and reality. *Nat. Rev. Drug Discovery* **2003**, *2*, 665–668.
- [8] Kier, L. B.; Hall, L. H.; Frazer, J. W. Design of molecules from quantitative structure-activity relationship models. 1. Information transfer between path and vertex degree counts. *J. Chem. Inf. Comput. Sci.* **1993**, *33*, 143–147.
- [9] Skvortsova, M. I.; Baskin, I. I.; Slovokhotova, O. L.; Palyulin, V. A.; Zefirov, N. S. Inverse problem in QSAR/QSPR studies for the case of topological indexes characterizing molecular shape (Kier indices). *J. Chem. Inf. Comput. Sci.* **1993**, *33*, 630–634.
- [10] Faulon, J.-L.; Churchwell, C. J.; Visco, D. P. The signature molecular descriptor. 2. Enumerating molecules from their extended valence sequences. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 721–734.
- [11] Brown, N.; McKay, B.; Gilardoni, F.; Gasteiger, J. A graph-based genetic algorithm and its application to the multiobjective evolution of median molecules. *J. Chem. Inf. Comput. Sci.* **2004**, *44*, 1079–1087.

- [12] Douguet, D.; Thoreau, E.; Grassy, G. A genetic algorithm for the automated generation of small organic molecules: Drug design using an evolutionary algorithm. *J. Comput.-Aided Mol. Des.* **2000**, *14*, 449–466.
- [13] Kvasnička, V.; Pospíchal, J. Simulated annealing construction of molecular graphs with required properties. *J. Chem. Inf. Comput. Sci.* **1996**, *36*, 516–526.
- [14] Lin, B.; Chavali, S.; Camarda, K.; Miller, D. C. Computer-aided molecular design using Tabu search. *Comput. Chem. Eng.* **2005**, *29*, 337–347.
- [15] Marcoulaki, E.; Kokossis, A. Molecular design synthesis using stochastic optimisation as a tool for scoping and screening. *Comput. Chem. Eng.* **1998**, *22*, S11–S18.
- [16] Venkatasubramanian, V.; Chan, K.; Caruthers, J. M. Evolutionary design of molecules with desired properties using the genetic algorithm. *J. Chem. Inf. Comput. Sci.* **1995**, *35*, 188–195.
- [17] Vapnik, V. N. *Statistical learning theory*; New York; Wiley-Interscience, 1998.
- [18] Cristianini, N.; Shawe-Taylor, J. *An introduction to support vector machines and other kernel-based learning methods*; Cambridge Univ. Press: Cambridge, U.K., 2000.
- [19] Schölkopf, B.; Smola, A. J. *Learning with Kernels*; MIT Press: Cambridge, MA, 2002.
- [20] Shawe-Taylor, J.; Cristianini, N. *Kernel Methods for Pattern Analysis*; Cambridge University Press: Cambridge, UK, 2004.
- [21] Vishwanathan, S. V. N.; Schraudolph, N. N.; Kondor, R.; Borgwardt, K. M. Graph kernels. *J. Mach. Learn. Res.* **2010**, *99*, 1201–1242.
- [22] Lodhi, H. M.; Yamanashi, Y. *Chemoinformatics and Advanced Machine Learning Perspectives: Complex Computational Methods and Collaborative Techniques*, 1st ed.; IGI Global: Hershey, PA, USA, 2010.
- [23] Trinajstić, N. *Chemical Graph Theory*, 2nd ed.; CRC Press: Boca Raton, Fla., 1992.
- [24] 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.
- [25] Mahé, P.; Ueda, N.; Akutsu, T.; Perret, J.-L.; Vert, J.-P. Graph kernels for molecular structure-activity relationship analysis with support vector machines. *J. Chem. Inf. Model.* **2005**, *45*, 939–951.
- [26] Mahé, P.; Vert, J.-P. Graph kernels based on tree patterns for molecules. *Mach. Learn.* **2009**, *75*, 3–35.
- [27] Bakır, G. H.; Weston, J.; Schölkopf, B. Learning to find pre-images. *Adv. Neural Inf. Process. Syst.* **2004**, *16*, 449–456.

- [28] Fujiwara, H.; Wang, J.; Zhao, L.; Nagamochi, H.; Akutsu, T. Enumerating treelike chemical graphs with given path frequency. *J. Chem. Inf. Model.* **2008**, *48*, 1345–1357.
- [29] Johnson, M. A., Maggiora, G. M., Eds. *Concepts and Applications of Molecular Similarity*; Wiley: New York, 1990.
- [30] Todeschini, R.; Consonni, V. *Molecular Descriptors for Chemoinformatics (2 volumes)*; Wiley-VCH: Weinheim, 2009.
- [31] Rogers, D.; Hahn, M. Extended-connectivity fingerprints. *J. Chem. Inf. Model.* **2010**, *50*, 742–754.
- [32] Rogers, D. J.; Tanimoto, T. T. A computer program for classifying plants. *Science* **1960**, *132*, 1115–1118.
- [33] Haussler, D. *Convolution kernels on discrete structures*; Technical Report UCSC-CRL-99-10, UC Santa Cruz, 1999.
- [34] 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.
- [35] 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.
- [36] 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.
- [37] 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.
- [38] Ralaivola, L.; Swamidass, S. J.; Saigo, H.; Baldi, P. Graph kernels for chemical informatics. *Neural Netw.* **2005**, *18*, 1093–1110.
- [39] 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.

- [40] 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.
- [41] 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.
- [42] 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.
- [43] 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.
- [44] 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.
- [45] 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.
- [46] 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).
- [47] Frasconi, P.; Passerini, A.; Muggleton, S.; Lodhi, H. Declarative kernels. Technical Report RT 2/2004; Dipartimento di Sistemi e Informatica, Università di Firenze, 2004.
- [48] 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.
- [49] Sheridan, R. P. The most common chemical replacements in drug-like compounds. *J. Chem. Inf. Comput. Sci.* **2002**, 42, 103–108.
- [50] Burden, F. R. Molecular identification number for substructure searches. *J. Chem. Inf. Comput. Sci.* **1989**, 29, 225–227.
- [51] Validation, M. the Receptor-Relevant Subspace Concept Pearlman, RS; Smith. *J. Chem. Inf. Comput. Sci.* **1999**, 39, 28–35.

- [52] Morgan, H. L. The generation of a unique machine description for chemical structures—a technique developed at chemical abstract service. *J. Chem. Doc.* **1965**, *5*, 107–113.
- [53] Geyer, C. J. Markov chain Monte Carlo maximum likelihood. In *Computing Science and Statistics Proceedings of the 23rd Symposium on the Interface*, American Statistical Association, New York. 1991; pp 156–163.
- [54] Hukushima, K.; Nemoto, K. Exchange Monte Carlo method and application to spin glass simulations. *J. Phys. Soc. Jpn.* **1996**, *65*, 1604–1608.
- [55] Iba, Y. Extended ensemble monte carlo. *Int. J. Mod. Phys. C* **2001**, *12*, 623–656.
- [56] Yamashita, H.; Higuchi, T.; Yoshida, R. Atom environment kernels on molecules. *J. Chem. Inf. Model.* **2014**, *54*, 1289–1300.
- [57] Horváth, T. In *Proceedings of the 9th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2005)*, Hanoi, Vietnam, May 18–20, Vol. 3518. *Lecture Notes in Computer Science*, Springer; Springer, 2005; pp 791–801.
- [58] Vert, J.-P. *The optimal assignment kernel is not positive definite*; Technical Report HAL-00218278, Centre for Computational Biology, Mines ParisTech, Paris, France, 2008.
- [59] Swamidass, S. J.; Chen, J.; Bruand, J.; Phung, P.; Ralaivola, L.; Baldi, P. Kernels for small molecules and the prediction of mutagenicity, toxicity and anti-cancer activity. *Bioinformatics* **2005**, *21*, i359–i368.
- [60] Wendland, H. Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. *Adv. Comput. Math.* **1995**, *4*, 389–396.
- [61] Zhang, H. H.; Genton, M. G.; Liu, P. Compactly supported radial basis function kernels. Available at <http://www4.stat.ncsu.edu/~hzhzhang/research.html>.
- [62] Gneiting, T. Compactly supported correlation functions. *J. Multivariate Anal.* **2002**, *83*, 493–508.
- [63] Gneiting, T. Correlation functions for atmospheric data analysis. *Q. J. Roy. Meteor. Soc.* **1999**, *125*, 2449–2464.
- [64] Berg, C.; Christensen, J. P.; Ressel, P. *Harmonic analysis on semigroups*; Springer-Verlag: New York, 1984.
- [65] 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).
- [66] O’Boyle, N.; Banck, M.; James, C.; Morley, C.; Vandermeersch, T.; Hutchison, G. Open Babel: An open chemical toolbox. *J. Cheminform.* **2011**, *3*, 1–14.
- [67] Open Babel: The Open Source Chemistry Toolbox. <http://openbabel.org> (accessed October 1, 2013).

- [68] 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).
- [69] 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).
- [70] Optimal Assignment Kernel: a Java implementation of the optimal assignment kernel. <http://www.ra.cs.uni-tuebingen.de/software/OAKernels/> (accessed October 1, 2013).
- [71] *Pipeline Pilot* version 7.5; Accelrys, Inc.: San Diego, CA, 2008.
- [72] 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.
- [73] Helma, C.; Kramer, S. A survey of the predictive toxicology challenge 2000–2001. *Bioinformatics* **2003**, *19*, 1179–1182.
- [74] 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.
- [75] Yoshida, F.; Topliss, J. G. QSAR model for drug human oral bioavailability. *J. Med. Chem.* **2000**, *43*, 2575–2585.
- [76] 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.
- [77] Huuskonen, J. Estimation of aqueous solubility for a diverse set of organic compounds based on molecular topology. *J. Chem. Inf. Comput. Sci.* **2000**, *40*, 773–777.
- [78] Burger, A. Isosterism and bioisosterism in drug design. *Prog. Drug Res.* **1991**, *37*, 287–371.
- [79] Stiefl, N.; Watson, I. A.; Baumann, K.; Zaliani, A. ErG: 2D pharmacophore descriptions for scaffold hopping. *J. Chem. Inf. Model.* **2006**, *46*, 208–220.
- [80] 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.
- [81] 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.
- [82] Baker, D.; Sali, A. Protein structure prediction and structural genomics. *Science* **2001**, *294*, 93–96.

- [83] Lesk, A. M.; Lo Conte, L.; Hubbard, T. J. Assessment of novel fold targets in CASP4: Predictions of three-dimensional structures, secondary structures, and interresidue contacts. *Proteins* **2001**, *45*, 98–118.
- [84] Simons, K. T.; Kooperberg, C.; Huang, E.; Baker, D. Assembly of protein tertiary structures from fragments with similar local sequences using simulated annealing and Bayesian scoring functions. *J. Mol. Biol.* **1997**, *268*, 209–225.
- [85] Jones, D. T. Predicting novel protein folds by using FRAGFOLD. *Proteins* **2001**, *45*, 127–132.
- [86] Jones, D. T.; Bryson, K.; Coleman, A.; McGuffin, L. J.; Sadowski, M. I.; Sodhi, J. S.; Ward, J. J. Prediction of novel and analogous folds using fragment assembly and fold recognition. *Proteins* **2005**, *61*, 143–151.
- [87] Chikenji, G.; Fujitsuka, Y.; Takada, S. A reversible fragment assembly method for de novo protein structure prediction. *J. Chem. Phys.* **2003**, *119*, 6895–6903.
- [88] Chikenji, G.; Fujitsuka, Y.; Takada, S. Shaping up the protein folding funnel by local interaction: Lesson from a structure prediction study. *Proc. Natl. Acad. Sci. USA* **2006**, *103*, 3141–3146.
- [89] Lee, J.; Kim, S.-Y.; Joo, K.; Kim, I.; Lee, J. Prediction of protein tertiary structure using PROFESY, a novel method based on fragment assembly and conformational space annealing. *Proteins* **2004**, *56*, 704–714.
- [90] Lee, J.; Kim, S.-Y.; Lee, J. Protein structure prediction based on fragment assembly and parameter optimization. *Biophys. Chem.* **2005**, *115*, 209–214.
- [91] Liang, F.; Wong, W. H. Evolutionary Monte Carlo: Applications to C_p model sampling and change point problem. *Stat. Sinica* **2000**, *10*, 317–342.
- [92] Mika, S.; Schölkopf, B.; Smola, A. J.; Müller, K.-R.; Scholz, M.; Rätsch, G. Kernel PCA and de-noising in feature spaces. In *Proceedings of the 1998 Conference on Advances in Neural Information Processing Systems II (NIPS 1998)*, Denver, Colorado, U.S.A., November 30–December 5, 1998; Michael, J. K., Sara, A. S., David, A. C., Eds.; MIT Press: Cambridge, MA, 1999; pp 536–542.
- [93] Goldberg, D. E.; Korb, B.; Deb, K. Messy genetic algorithms: Motivation, analysis, and first results. *Complex Systems* **1989**, *3*, 493–530.
- [94] Holland, J. H. *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence.*; University of Michigan Press, Ann Arbor, MI, 1975.
- [95] Nijssen, S.; Kok, J. N. A quickstart in frequent structure mining can make a difference. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*. 2004; pp 647–652.
- [96] Nijssen, S. Gaston: a unified graph, sequences and tree extraction algorithm. <http://www.liacs.nl/~snijssen/gaston/> (accessed October 1, 2013).

- [97] Kazius, J.; Nijssen, S.; Kok, J.; Bäck, T.; IJzerman, A. P. Substructure Mining Using Elaborate Chemical Representation. *J. Chem. Inf. Model.* **2006**, *46*, 597–605.
- [98] Tabei, Y. *Workshop on Algorithms in Bioinformatics (WABI) ALGO*; Springer, 2012; pp 201–213.
- [99] Yasuo, T. smbt: Succinct Multibit Tree. <http://code.google.com/p/smbt/> (accessed October 1, 2013).
- [100] Goldberg, D. E. *Genetic Algorithms in Search, Optimization and Machine Learning*, 1st ed.; Addison-Wesley Longman Publishing Co., Inc.: Boston, MA, USA, 1989.
- [101] Gaulton, A.; Bellis, L. J.; Bento, A. P.; Chambers, J.; Davies, M.; Hersey, A.; Light, Y.; McGlinchey, S.; Michalovich, D.; Al-Lazikani, B.; Overington, J. P. ChEMBL: a large-scale bioactivity database for drug discovery. *Nucleic Acids Res.* **2012**, *40*, D1100–D1107.
- [102] Hann, M.; Hudson, B.; Lewell, X.; Lifely, R.; Miller, L.; Ramsden, N. Strategic pooling of compounds for high-throughput screening. *J. Chem. Inf. Comp. Sci.* **1999**, *39*, 897–902.
- [103] Irwin, J. J.; Shoichet, B. K. ZINC-a free database of commercially available compounds for virtual screening. *J. Chem. Inf. Model.* **2005**, *45*, 177–182.
- [104] Baell, J. B.; Holloway, G. A. New substructure filters for removal of pan assay interference compounds (PAINS) from screening libraries and for their exclusion in bioassays. *J. Med. Chem.* **2010**, *53*, 2719–2740.
- [105] Congreve, M.; Carr, R.; Murray, C.; Jhoti, H. A ‘Rule of Three’ for fragment-based lead discovery? *Drug Discov. Today* **2003**, *8*, 876–877.
- [106] Köster, H.; Craan, T.; Brass, S.; Herhaus, C.; Zentgraf, M.; Neumann, L.; Heine, A.; Klebe, G. A small nonrule of 3 compatible fragment library provides high hit rate of endothiapepsin crystal structures with various fragment chemotypes. *J. Med. Chem.* **2011**, *54*, 7784–7796.
- [107] RDKit: Open-source cheminformatics. <http://www.rdkit.org> (accessed October 1, 2013).
- [108] Weininger, D. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *J. Chem. Inf. Comput. Sci.* **1988**, *28*, 31–36.
- [109] Weininger, D.; Weininger, A.; Weininger, J. L. SMILES. 2. Algorithm for generation of unique SMILES notation. *J. Chem. Inf. Comput. Sci.* **1989**, *29*, 97–101.
- [110] Hirabayashi, M. Tokyo Cabinet: a modern implementation of DBM. <http://fallabs.com/tokyocabinet/index.html> (accessed October 1, 2013).
- [111] Lipinski, C. A. Drug-like properties and the causes of poor solubility and poor permeability. *J. Pharmacol. Toxicol.* **2000**, *44*, 235–249.
- [112] Rishton, G. M. Nonleadlikeness and leadlikeness in biochemical screening. *Drug Discov. Today* **2003**, *8*, 86–96.

- [113] Petrova, T.; Chuprina, A.; Parkesh, R.; Pushechnikov, A. Structural enrichment of HTS compounds from available commercial libraries. *Med. Chem. Commun.* **2012**, *3*, 571–579.
- [114] Open MPI: Open Source High Performance Computing. <http://www.open-mpi.org> (accessed October 1, 2013).
- [115] Choi, H.; Choi, S. Kernel isomap on noisy manifold. In *Proceedings of 4th IEEE International Conference on Development and Learning*, IEEE Computer Society Press, Los Alamitos. 2005; pp 208–213.
- [116] Choi, H.; Choi, S. Robust kernel isomap. *Pattern Recogn.* **2007**, *40*, 853–862.
- [117] Kohonen, T. *Self-Organizing Maps, 3rd Edition, volume 30*; Springer: Berlin, Heidelberg, New York, 2001.
- [118] Ertl, P.; Schuffenhauer, A. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *J. Cheminformatics* **2009**, *1*, 8.

List of Publications

- [1] Yamashita, H.; Higuchi, T.; Yoshida, R. Atom environment kernels on molecules. *Journal of Chemical Information and Modeling* **2014**, *54*, 1289–1300.
- [2] Yamashita, H.; Kidera, A. Environmental influence on electron scattering from a molecule. *Acta Crystallographica Section A: Foundations of Crystallography* **2001**, *57*, 518–525.
- [3] Yamashita, H.; Endo, S.; Wako, H.; Kidera, A. Sampling efficiency of molecular dynamics and Monte Carlo method in protein simulation. *Chemical physics letters* **2001**, *342*, 382–386.