The Pharmacophore Kernel for Virtual Screening with Support Vector Machines

Pierre Mahé,*,† Liva Ralaivola,‡ Véronique Stoven,† and Jean-Philippe Vert†

Center for Computational Biology, Ecole des Mines de Paris, 35 rue Saint Honoré, 77305 Fontainebleau, France, and Laboratoire d'Informatique Fondamentale, University Provence/Aix-Marseille, 139 rue Joliot-Curie, F-13453 Marseille Cedex 13, France

Received April 13, 2006

We introduce a family of positive definite kernels specifically optimized for the manipulation of 3D structures of molecules with kernel methods. The kernels are based on the comparison of the three-point pharmacophores present in the 3D structures of molecules, a set of molecular features known to be particularly relevant for virtual screening applications. We present a computationally demanding exact implementation of these kernels, as well as fast approximations related to the classical fingerprint-based approaches. Experimental results suggest that this new approach is competitive with state-of-the-art algorithms based on the 2D structure of molecules for the detection of inhibitors of several drug targets.

1. INTRODUCTION

Virtual screening refers to the process of inferring biological properties of molecules in silico and plays an increasingly important role at the early stages of the drug discovery process to select candidate molecules with promising druglikeness, including good toxicity and pharmacokinetics properties, as well as the potential to bind and inhibit a target protein of interest.[1] In this context, structure−activity relationship (SAR) analysis is commonly used to build predictive models for the property of interest from a description of the molecules, using statistical procedures to build these models from the analysis of molecules with known properties.[2]

It is widely accepted that several druglike properties can be efficiently deduced from the 2D structure of the molecule, that is, the description of a molecule as a set of atoms and their covalent bonds. For example, Lipinski's "rule of five" remains a widely used standard for the prediction of intestinal absorption,[3] and the prediction of mutagenicity from 2D molecular fragments is an accurate state-of-the-art approach.[4] In the case of target binding prediction, however, the molecular mechanisms responsible for the binding are known to depend on a precise 3D complementarity between the drug and the target, from both the steric and electrostatic perspectives.[5] For this reason, there has been a long history of research on the prediction of these interactions from the 3D representation of molecules, that is, their spatial conformation in the 3D space. If the 3D structure of the target is known, the strength of the interaction can be directly evaluated by docking techniques, which quantify the complementarity of the molecule to the target in terms of energy.[6] In the general case where the 3D structure of the target is unknown, however, the docking approach is not possible anymore, and the modeler must resort to creating a predictive model from available data, typically a pool of molecules with a known affinity to the target; this approach is usually referred to as the *ligand-based* approach to virtual screening.

Most approaches to ligand-based virtual screening require the representation and comparison of 3D structures of molecules. The comparison of 3D structures can, for example, rely on optimal alignments in the 3D space,[7] or on the comparison of features extracted from the structures.[8] Features of particular importance in this context are subsets of two to four atoms together with their relative spatial organization, also called *pharmacophores*. Discovering pharmacophores common to a set of known inhibitors to a drug target can be a powerful approach to the screening of other candidate molecules containing the pharmacophores, as well as a first step toward the understanding of the biological phenomenon involved.[9,10] Alternatively, the use of pharmacophore fingerprints, that is, bitstrings representing a molecule by the pharmacophores it contains, has emerged as a potential approach to apply statistical learning methods for SARs, although sometimes with mixed results.[11−13]

We focus in this paper on an extension of the fingerprint representation of molecules for building SAR models with support vector machines (SVMs). SVM is an algorithm for learning a classification or regression rule from labeled examples,[14,15] which has recently been subject to much investigation for SAR applications in chemoinformatics.[16−19] Although SVMs can be trained from a vector or bitstring representation of molecules, they can also take advantage of a mathematical trick to only rely on a measure of similarity between molecules, known as a *kernel*. This trick, common to other algorithms called kernel methods,[20] was, for example, used in refs 18 and 21 to build SAR models from a 2D fingerprint of molecules of virtually infinite length. Here, we investigate the possibility of using this trick in the context of 3D SAR modeling. We propose a measure of similarity between 3D structures, which we call the *pharmacophore kernel*, based on the comparison of pharmacophores present in the structures. It satisfies the mathematical properties required to be a valid kernel, and it therefore allows the use of SVM for model building. This kernel bears

* Corresponding author phone: (+33) 1 64 69 49 94; e-mail: pierre.mahe@ensmp.fr.
† Ecole des Mines de Paris.
‡ University Provence/Aix-Marseille.

10.1021/ci060138m CCC: $33.50
Published on Web 08/12/2006

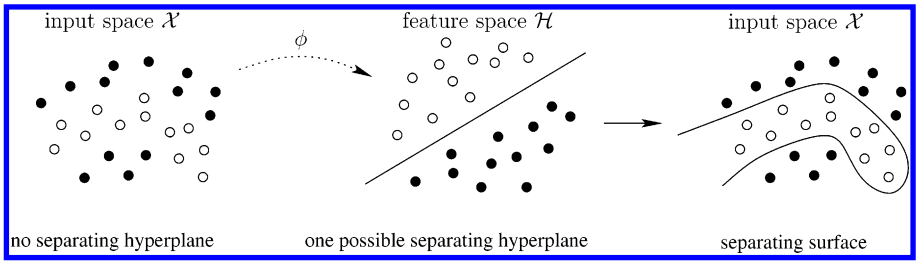

Figure 1. The kernel trick. Instead of looking for, e.g., a separating hyperplane directly in the input space $\mathcal{X}$, training patterns (white and black disks) are mapped into a *feature space* $\mathcal{H}$ through a function ϕ in which a hyperplane is computed; this hyperplane might correspond to a complex surface in the input space. Using a proper positive definite kernel k when carrying out the computation to derive the separating hyperplane is equivalent to directly working with the images of the training samples by some mapping ϕ from $\mathcal{X}$ to some feature space $\mathcal{H}$; the existence of ϕ: $\mathcal{X} \rightarrow \mathcal{H}$ is guaranteed by eq 3.

some similarity with pharmacophore fingerprint approaches, although it produces more general models. In fact, we show that a fast approximation of this kernel, based on pharmacophore fingerprints, leads to significantly lower performance on a benchmark data set. The overall good performance of the approach on this benchmark supports its relevance as a potentially effective tool for 3D SAR modeling.

This paper is organized as follows. A light introduction to SVM and kernel methods is provided in section 2, followed by the definition of the pharmacophore kernel (section 3). The exact computation of this kernel is presented in section 4, followed by a discussion about the connection between the pharmacophore kernel and recently introduced graph kernels (section 5) and the presentation of a fast approximation (section 6). Experimental results on a benchmark data set for inhibitor prediction are then presented in section 7, followed by a short discussion.

2. SUPPORT VECTOR MACHINES

In this section, we briefly review the basics of support vector machines.[14,15] The interested reader is invited to refer to refs 20, 22, and 23 for further details. In its simplest form, SVM is an algorithm to learn a binary classification rule from a set of labeled examples. More formally, suppose one is given a set of examples with a binary label attached to each example, that is, a set $\mathcal{S} = \{(x_1, y_1), ..., (x_\ell, y_\ell)\}$ where $(x_i, y_i) \in \mathcal{X} \times \{-1, +1\}$ for $i = 1, ..., \ell$. Here, $\mathcal{X}$ is an innerproduct space (e.g., $\mathbb{R}^d$), equipped with inner product $\langle \cdot, \cdot \rangle$, which represents the space of data to be analyzed, typically molecules represented by d-dimensional fingerprints, and the labels $+1$ and -1 are meant to represent two classes of objects, such as inhibitors or noninhibitors of a target of interest. The purpose of SVM is to learn from $\mathcal{S}$ a classification function $f: \mathcal{X} \rightarrow \{-1, +1\}$ that can be used to predict the class of new unlabeled examples x as $f(x)$.

In the case of SVM, the classification function is simply of the form $f(x) = \text{sign}(\langle w, x \rangle + b)$, where sign$(\cdot)$ is the function returning the sign, $+1$ or -1, of its argument. Geometrically speaking, this means that f outputs a prediction for a pattern x depending upon which side of the hyperplane $\langle w, x \rangle + b = 0$ it falls in. More precisely, SVM learns a separating hyperplane from $\mathcal{S}$ defined by a vector w that is a linear combination of the training vectors $w = \sum_{i=1}^{\ell} \alpha_i x_i$, for some $\alpha_i \in \mathbb{R}$, $i = 1, ..., \ell$, obtained by solving a linearly constrained quadratic problem meant to optimize a tradeoff between finding a hyperplane that correctly separates all of the points and being as far as possible from each point. The linear classifier f can consequently be rewritten as

$$f(x) = \text{sign}\left(\sum_{i=1}^{\ell} \alpha_i \langle x_i, x \rangle + b\right) \tag{1}$$

However, when dealing with nonlinearly separable problems, such as the one depicted in Figure 1 (left), the set of linear classifiers may not be rich enough to provide a good classification function, no matter what the values of the parameters $w \in \mathcal{X}$ and $b \in \mathbb{R}$ are. The purpose of the *kernel trick*[14,24] is precisely to overcome this limitation by applying a linear approach to the transformed data $\phi(x_1), ..., \phi(x_\ell)$ rather than the original data, where ϕ is an embedding from the input space $\mathcal{X}$ to the *feature space* $\mathcal{H}$, usually, but not necessarily, a high-dimensional space, equipped with dot product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$. Thus, according to eq 1, the separating function f writes as

$$f(x) = \text{sign}\left(\sum_{i=1}^{\ell} \alpha_i \langle \phi(x_i), \phi(x) \rangle_{\mathcal{H}} + b\right) \tag{2}$$

The key ingredient in the kernel approach is to replace the dot product in $\mathcal{H}$ with a kernel, using the definition of positive definite kernels.

Definition 1 (Positive Definite Kernel). Let $\mathcal{X}$ be a nonempty space. Let K: $\mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ be a symmetric function. K is said to be a positive definite kernel if and only if, for all positive integers ℓ, for all $x_1, ..., x_\ell \in \mathcal{X}$, the square $\ell \times \ell$ matrix $\mathbf{K} = [K(x_i, x_j)]_{1 \le i,j \le \ell}$ is positive semidefinite, that is, all of its eigenvalues are non-negative.

For a given set $\mathcal{S}_x = \{x_1, ..., x_\ell\}$, $\mathbf{K}$ is the *Gram matrix* of K with respect to $\mathcal{S}_x$. A fundamental property of positive definite kernels that underlies the kernel trick is the fact that each such kernel can be represented as an inner product in some space. More precisely, it can be shown[25] that, for any positive definite kernel function K, there exists a space $\mathcal{H}$, equipped with the inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$, and a mapping ϕ: $\mathcal{X} \rightarrow \mathcal{H}$ such that

$$\forall u, v \in \mathcal{X}\ K(u,v) = \langle \phi(u), \phi(v) \rangle_{\mathcal{H}} \tag{3}$$

The kernel trick consists of replacing all occurrences of $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ in eq 2 by a positive definite kernel K such that the corresponding decision function f, for an input pattern x, is given by

$$f(x) = \mathrm{sign}\left(\sum_{i=1}^{\ell} \alpha_i K(x_i,x) + b\right) \quad (4)$$

For SVM as well as for other kernel methods, the knowledge of the Gram matrix suffices to obtain the coefficients α_i.

For any given positive definite kernel, applying the kernel trick turns out to be equivalent to transforming the input patterns x_1, ..., x_ℓ into the corresponding vectors $\phi(x_1)$, ..., $\phi(x_\ell) \in \mathcal{H}$ and looking for hyperplanes in $\mathcal{H}$, as illustrated in Figure 1 (middle). The decision surface in input space $\mathcal{X}$ corresponding to the selected separating hyperplane in $\mathcal{H}$ might be quite complex (see Figure 1, right).

A noteworthy feature of support vector machines and more generally of kernel methods[23] is that, because ready-to-use libraries to derive separating hyperplanes are available, the only requirement for them to be applied to a specific classification problem is to have at hand a proper kernel function to assess the similarity between patterns of the input space considered. As a result, they can be used in classification problems involving structured data such as chemical compounds, provided some kernel function has been derived. The rest of the paper is devoted to the construction and analysis of such a kernel for 3D structures of molecules.

3. THE PHARMACOPHORE KERNEL

A *pharmacophore* is usually defined as a three-dimensional arrangement of atoms—or groups of atoms—responsible for the biological activity of a drug molecule.[26] The present work focuses on *three-point* pharmacophores, composed of three atoms whose arrangement, therefore, forms a triangle in the 3D space (Figure 2). With a slight abuse, we refer to pharmacophore below as *any* possible configuration of three atoms or classes of atoms arranged as a triangle and present in a molecule, representing therefore a *putative* configuration responsible for the biological property of interest.

Throughout this paper, we represent the 3D structure of a molecule as a set of points in $\mathbb{R}^3$. These points correspond to the 3D coordinates of the atoms of the molecule (for a given arbitrary basis of the 3D Euclidean space), and they are labeled with some information related to the atoms. More formally, we define a molecule m as

$$m = \{(x_i,l_i) \in \mathbb{R}^3 \times \mathcal{L}\}_{i=1,\ldots,|m|}$$

where $|m|$ is the number of atoms that composes the molecule and $\mathcal{L}$ denotes the set of atom labels. The label is meant to contain the relevant information to characterize a pharmacophore based on atoms. It might for instance be defined by the type of atom (C, N, O, ...) or various physicochemical atomic properties (e.g., partial charge). The three-point pharmacophores considered in this work correspond to triplets of distinct atoms of the molecules. The set of pharmacophores of the molecule m can therefore be formally defined as

$$\mathcal{P}(m) = \{(p_1,p_2,p_3) \in m^3, p_1 \neq p_2, p_1 \neq p_3, p_2 \neq p_3\} \quad (5)$$

More generally, the set of all possible pharmacophores is naturally defined as $\mathcal{P} = (\mathbb{R}^3 \times \mathcal{L})$,[3] to ensure the inclusion $\mathcal{P}(m) \subset \mathcal{P}$. We can now define a general family of kernels for molecules on the basis of their pharmacophore content.

Definition 2. For any positive definite kernel for pharmacophores $K_{\mathcal{P}}$: $\mathcal{P} \times \mathcal{P} \rightarrow \mathbb{R}$, we define a corresponding pharmacophore kernel for any pair of molecules m and m' by

$$K(m,m') = \sum_{p \in \mathcal{P}(m)} \sum_{p' \in \mathcal{P}(m')} K_{\mathcal{P}}(p,p') \quad (6)$$

with the convention that $K(m, m') = 0$ if either $\mathcal{P}(m)$ or $\mathcal{P}(m')$ is empty.

The fact that the pharmacophore kernel defined in eq 6 is a valid positive definite kernel on the set of molecules, as soon as $K_{\mathcal{P}}$ is itself a valid positive definite kernel on the set of pharmacophores, is a classical result (see, e.g., ref 27, Lemma 1). The problem of constructing a pharmacophore kernel for molecules therefore boils down to the simpler problem of defining a kernel between pharmacophores. A chemically relevant measure of similarity between pharmacophores should obviously quantify at least two features: first, similar pharmacophores should be made of similar atoms [where the notion of similarity can for instance be based on atom types or property(ies) and, more generally, on pharmacophoric types], and second, the atoms should have similar relative positions in the 3D space. It is therefore natural to study kernels for pharmacophores that decompose as follows:

$$K_{\mathcal{P}}(p,p') = K_{\mathrm{I}}(p,p') \times K_{\mathrm{S}}(p,p') \quad (7)$$

where K_{I} is a kernel function assessing the similarity between the triplets of basis atoms of the pharmacophores (their so-called *intrinsic* similarity) and K_{S} is a kernel function introduced to quantify their *spatial* similarity.

We can furthermore investigate intrinsic and spatial kernels that factorize themselves as products of more basic kernels between atoms and pairwise distances, respectively. Triplets of atoms are indeed globally similar if the three corresponding pairs of atoms are simultaneously similar, and triangles are similar if the lengths of their edges are pairwise similar. For any pair of pharmacophores $p = [(x_1,l_1),(x_2,l_2),(x_3,l_3)]$ and $p' = [(x'_1,l'_1),(x'_2,l'_2),(x'_3,l'_3)]$, this suggests the definition of kernels as follows:

$$K_{\mathrm{I}}(p,p') = \prod_{i=1}^{3} K_{\mathrm{Feat}}(l_i,l'_i) \quad (8)$$

$$K_{\mathrm{S}}(p,p') = \prod_{i=1}^{3} K_{\mathrm{Dist}}(||x_i - x_{i+1}||,||x'_i - x'_{i+1}||) \quad (9)$$

where $||\cdot||$ denotes the Euclidean distance, the index $i + 1$ is taken modulo 3, and K_{Feat} and K_{Dist} are kernel functions introduced to compare pairs of labels from $\mathcal{L}$ and pairs of distances, respectively. It suffices now to define the kernels K_{Feat} on $\mathcal{L} \times \mathcal{L}$ and K_{Dist} on $\mathbb{R} \times \mathbb{R}$ in order to obtain, by eqs 6−9, a pharmacophore kernel for molecules. The first one compares the atom labels, while the second compares the distances between atoms in the pharmacophores. Intuitively, they define the basic notions of similarity involved in the pharmacophore comparison, which in turn defines the overall similarity between molecules.

Note from the definition in eq 5 that, because of permutations, every distinct triplet of atoms of the molecule m gives

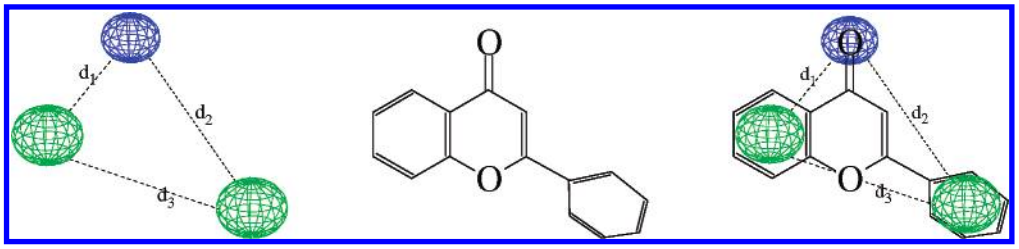

Figure 2. Left: A three-point pharmacophore made of one hydrogen-bond acceptor (top-most sphere) and two aromatic rings, with distances d_1, d_2, and d_3 between the features. Middle: The molecule of flavone. Right: Match between flavone and the pharmacophore.

rise to six pharmacophores in $\mathscr{P}(m)$. In the general case, these six pharmacophores are considered as different in the pharmacophore kernel. However, because of the definition of the notion of similarity between pharmacophores, some of these six pharmacophores will be seen as identical if the triplet of atoms is made of atoms of the same type. This phenomenon is even emphasized if the triplet of atoms exhibits some kind of spatial symmetry. For example, the six pharmacophores associated with a triplet of identical atoms arranged as an equilateral triangle are identical.

The kernel we use for K_{Dist} is the Gaussian radial basis function (RBF) kernel, known to be a safe default choice for SVM working on real numbers or vectors:[20]

$$K_{\text{Dist}}^{\text{RBF}}(x,y) = \exp\left(-\frac{||x-y||^2}{2\sigma^2}\right) \tag{10}$$

where $\sigma > 0$ is the bandwidth parameter that will be optimized as part of the training of the classifier (see section 7.2).

Various kernels K_{Feat} between labels can be chosen depending on the atom labels definition. With these labels belonging, in principle, to a finite set of possible labels, for example, the set of atom types with their charges (C, C^+, C^-, N, ...), the following *Dirac kernel* is a natural default choice to compare a pair of atom labels l, $l' \in \mathscr{L}$:

$$K_{\text{Feat}}^{\text{Dirac}}(l,l') = \begin{cases} 1 & \text{if } l = l' \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

Alternatively, it might be relevant for the pharmacophore definition to compare atoms not only on the basis of their types and partial charges but also in terms of other physicochemical parameters such as their size, polarity, and electronegativity. Formally, a physicochemical parameter for an atom with label l is a real number $f(l)$. In that case, the Gaussian RBF kernel (eq 10) could be applied directly to the parameter values to compare labels. We discuss this issue in section 8.

Note finally that the Gaussian (eq 10) and Dirac (eq 11) kernels are known to be definite positive,[20] and it follows from the closure properties of the family of kernel functions that the kernel between pharmacophores $K_{\mathscr{P}}$ is valid for these choices of the kernels K_{Feat} and K_{Dist}.

4. KERNEL COMPUTATION

We are now left with the task of computing the pharmacophore kernel (eq 6) for a particular choice of feature and distance kernels K_{Feat} and K_{Dist}. In this section, we provide a simple analytical formula for this computation.

For any pair of molecules $m = \{(x_i,l_i) \in \mathbb{R}^3 \times \mathscr{L}\}_{i=1,\ldots,|m|}$ and $m' = \{(x'_i,l'_i) \in \mathbb{R}^3 \times \mathscr{L}\}_{i=1,\ldots,|m'|}$, let us define a square matrix $\mathbf{M}$ of size $n = |m| \times |m'|$, whose dimensions are indexed by the Cartesian product of m and m'. In other words, to each index $i \in [1,n]$ corresponds a unique couple of indices $(i_1,i_2) \in [1,|m|] \times [1,|m'|]$, and to each dimension of the matrix $\mathbf{M}$ corresponds a distinct pair of points taken from the molecules m and m'. Denoting by $\mathbf{1}(\cdot)$, the indicator function equal to one if its argument is true, and zero otherwise, the entries of $\mathbf{M}$ are defined by

$$\begin{aligned}\mathbf{M}[i,j] &= \mathbf{M}[(i_1,i_2),(j_1,j_2)] \\ &= K_{\text{Feat}}(l_{i_1}, l'_{i_2}) \times K_{\text{Dist}}(||x_{i_1} - x_{j_1}||,||x'_{i_2} - x'_{j_2}||) \times \\ &\quad \mathbf{1}(i_1 \neq j_1) \times \mathbf{1}(i_2 \neq j_2)\end{aligned} \tag{12}$$

The value of the pharmacophore kernel between m and m' can now be deduced from the matrix $\mathbf{M}$ by the following result:

Proposition 1. The pharmacophore kernel (eq 6) between a pair of molecules m and m' is equal to

$$K(m,m') = \text{trace}(\mathbf{M}^3)$$

where $\mathbf{M}$ is the square matrix of dimensions $|m| \times |m'|$ constructed from m and m' by eq 12.

Proof. Developing the matrix products involved in the expression of $\mathbf{M}^3$, we get

$$\text{trace}(\mathbf{M}^3) = \sum_{i,j,k=1}^{n} \mathbf{M}[i,j]\,\mathbf{M}[j,k]\,\mathbf{M}[k,i]$$

where $n = |m| \times |m'|$ is the size of $\mathbf{M}$. Using the fact that the indices of $\mathbf{M}$ range over the Cartesian product of the set of indices $[1,|m|]$ and $[1,|m'|]$, we can rewrite this expression as

$$\begin{aligned}\text{trace}(\mathbf{M}^3) = \sum_{i_1,j_1,k_1=1}^{|m|} \sum_{i_2,j_2,k_2=1}^{|m'|} &\mathbf{M}[(i_1,i_2),(j_1,j_2)] \\ &\mathbf{M}[(j_1,j_2),(k_1,k_2)]\,\mathbf{M}[(k_1,k_2),(i_1,i_2)]\end{aligned}$$

Substituting with the definition of $\mathbf{M}$ given in eq 12, we obtain

$$\begin{aligned}\mathrm{trace}(\mathbf{M}^3) = &\sum_{i_1,j_1,k_1=1}^{|m|}\sum_{i_2,j_2,k_2=1}^{|m'|}\mathbf{1}(i_1\neq j_1)\,\mathbf{1}(j_1\neq k_1)\\ &\mathbf{1}(k_1\neq i_1)\times\mathbf{1}(i_2\neq j_2)\,\mathbf{1}(j_2\neq k_2)\,\mathbf{1}(k_2\neq i_2)\\ &\times K_{\mathrm{Feat}}(l_{i_1},l'_{i_2})\times K_{\mathrm{Dist}}(||x_{j_1}-x_{i_1}||,||x'_{j_2}-x'_{i_2}||)\\ &\times K_{\mathrm{Feat}}(l_{j_1},l'_{j_2})\times K_{\mathrm{Dist}}(||x_{k_1}-x_{j_1}||,||x'_{k_2}-x'_{j_2}||)\\ &\times K_{\mathrm{Feat}}(l_{k_1},l'_{k_2})\times K_{\mathrm{Dist}}(||x_{i_1}-x_{k_1}||,||x'_{i_2}-x'_{k_2}||)\\ =&\sum_{i_1,j_1,k_1=1}^{|m|}\sum_{i_2,j_2,k_2=1}^{|m'|}\mathbf{1}(i_1\neq j_1\neq k_1)\times\\ &\mathbf{1}(i_2\neq j_2\neq k_2)\times K_{\mathscr{P}}[[(x_{i_1},l_{i_1}),(x_{j_1},l_{j_1}),(x_{k_1},l_{k_1})],\\ &\qquad[(x'_{i_2},l'_{i_2}),(x'_{j_2},l'_{j_2}),(x'_{k_2},l'_{k_2})]]\\ =&\sum_{\substack{i_1,j_1,k_1=1,\\ i_1\neq j_1\neq k_1}}^{|m|}\sum_{\substack{i_2,j_2,k_2=1\\ i_2\neq j_2\neq k_2}}^{|m'|}K_{\mathscr{P}}[[(x_{i_1},l_{i_1}),(x_{j_1},l_{j_1}),(x_{k_1},l_{k_1})],\\ &\qquad[(x'_{i_2},l'_{i_2}),(x'_{j_2},l'_{j_2}),(x'_{k_2},l'_{k_2})]]\\ =&\sum_{p\in\mathscr{P}(m)}\sum_{p'\in\mathscr{P}(m')}K_{\mathscr{P}}(p,p')\\ =&K(m,m')\end{aligned}$$

If we let u be the cost of evaluating the basis kernels K_{Feat} and K_{Dist} and consider that the cost of the addition and product operations is a small constant, the complexity of the kernel between pharmacophores $K_{\mathscr{P}}$ is $6u$. Because the cardinality of the set of pharmacophores $\mathscr{P}(m)$ of the molecule m is $|m|^3$, the complexity of the direct computation of the pharmacophore kernel given in definition 2 is $(|m|\times|m'|)^3\times 6u$. On the other hand, the computation given in proposition 1 is a two-step process: First is the initialization of the matrix $\mathbf{M}$; each of the $(|m|\times|m'|)^2$ entries is initialized by the product of a kernel K_{Feat} with a kernel K_{Dist}, for a complexity of $(|m|\times|m'|)^2\times 2u$. Second is the computation of the trace of $\mathbf{M}^3$, which has a complexity of $(|m|\times|m'|)^3$. The global complexity of the matrix-based computation of the kernel is therefore $(|m|\times|m'|)^3+(|m|\times|m'|)^2\times 2u$ or, equivalently, $(|m|\times|m'|)^3\times[1+2u/(|m|\times|m'|)]$. In comparison with the direct approach, the matrix-based implementation proposed in proposition 1 reduces the number of basis kernels K_{Dist} and K_{Feat} to be computed and is therefore more efficient.

In any case, the complexity of the pharmacophore kernel computation is therefore $\mathscr{O}[(|m|\times|m'|)^3]$. Even for relatively small molecules (on the order of 50 atoms), this complexity becomes in practice a serious issue when the size of the data set increases to thousands or tens of thousands of molecules. However, we can note from the definition given in eq 12 that the lines of $\mathbf{M}$ corresponding to pairs of points $(x,l)\in m$ and $(x',l')\in m'$ for which $K_{\mathrm{Feat}}(l,l')=0$ are filled with zeros. On the basis of this consideration, we observe that the cost of computing the kernel can be reduced by limiting the size of the matrix $\mathbf{M}$, according to the following proposition.

Proposition 2. If we let $\mathbf{M}_2$ be the reduced version of a square matrix $\mathbf{M}_1$, where the null lines and the corresponding columns are removed, then $\mathrm{trace}(\mathbf{M}_2{}^3)=\mathrm{trace}(\mathbf{M}_1{}^3)$.

Proof. Let n_1 (respectively n_2) be the size of $\mathbf{M}_1$ (respectively $\mathbf{M}_2$), and define P (respectively N) as the subset of the set of indices $[1,n_1]$ that corresponds to the non-null (respectively null) lines of $\mathbf{M}_1$. By definition, we have

$$\begin{aligned}\mathrm{trace}(\mathbf{M}_1{}^3)&=\sum_{i=1}^{n_1}\mathbf{M}_1{}^3[i,i]\\ &=\sum_{i,j,k=1}^{n_1}\mathbf{M}_1[i,j]\,\mathbf{M}_1[j,k]\,\mathbf{M}_1[k,i]\end{aligned}\qquad(13)$$

Moreover, if $i\in N$, then $\mathbf{M}_1[i,j]=0\ \forall j\in[1,n_1]$. As a consequence, the term $\mathbf{M}_1[i,j]\,\mathbf{M}_1[j,k]\,\mathbf{M}_1[k,i]$ in the summations over i, j, and k in eq 13 is zero as soon as at least one index i, j, or k is in the set N. It follows that

$$\begin{aligned}\mathrm{trace}(\mathbf{M}_1{}^3)&=\sum_{i,j,k\in P}\mathbf{M}_1[i,j]\,\mathbf{M}_1[j,k]\,\mathbf{M}_1[k,i]\\ &=\sum_{i,j,k=1}^{n_2}\mathbf{M}_2[i,j]\,\mathbf{M}_2[j,k]\,\mathbf{M}_2[k,i]\\ &=\mathrm{trace}(\mathbf{M}_2{}^3)\end{aligned}$$

Proposition 2 implies that the Cartesian product of m and m' involved in the matrix $\mathbf{M}$ defined in eq 12 can be restricted to the pairs of points for which the label kernel K_{Feat} is nonzero. In the case of the Dirac kernel (eq 11) for discrete labels, this boils down to introducing a dimension in $\mathbf{M}$ for any pair of atoms having the same label. This result can have important consequences in practice. Consider for example the case where the atoms of the molecules m and m' are uniformly distributed in k classes of atom labels. In this case, the size of the matrix $\mathbf{M}$ is equal to $k(|m|/k\times|m'|/k)=|m|\times|m'|/k$. The complexity of the kernel computation is therefore $\mathscr{O}[(|m|\times|m'|/k)^3]=\mathscr{O}[(1/k^3)(|m|\times|m'|)^3]$. It is therefore reduced by a factor k^3 in comparison with the original implementation. More generally, this shows that important gains in memory and computation can be expected when the set of labels is increased. Section 7.3 discusses such a case in more detail when the partial charges of atoms are included or not in the labels. Note finally that, in a similar way, the kernel K_{Dist} to compare distances can be set to a compactly supported kernel instead of the Gaussian RBF kernel (eq 10). This has the effect of introducing sparsity in the matrix $\mathbf{M}$, allowing the kernel computation to benefit from sparse matrix algorithms. This possibility was not further explored in this work.

5. RELATION WITH GRAPH KERNELS

In this section, we show that the pharmacophore kernel can be seen as an extension of the walk-count graph kernels[28] to the 3D representation of molecules. The walk-count graph kernel is based on the representation of a molecule m as a labeled graph $m=(\mathscr{V},\mathscr{E})$, defined by a set of vertices $\mathscr{V}$, a set of edges $\mathscr{E}\subset\mathscr{V}\times\mathscr{V}$ connecting pairs of vertices, and a labeling function l: $\mathscr{V}\cup\mathscr{E}\rightarrow\mathscr{A}$, assigning a label $l(x)$ in an alphabet $\mathscr{A}$ to any vertex or edge x. In the case of molecules, the set of vertices $\mathscr{V}$ corresponds to the atoms of the molecule, and the edges of the graph are usually defined as the covalent bonds between the atoms of the molecules.[18,21,28] To extend this 2D representation to a graph structure capturing 3D information, we propose to introduce an edge

between any pair of vertices of the graph. Molecules are therefore seen as complete, atom-based graphs. If we now define a walk of length n as a succession of $n + 1$ connected vertices, it is easy to see that there is a one-to-one correspondence between the set of pharmacophores $\mathscr{P}(m)$ of a molecule m and its set of self-returning walks of length three, which we call $\mathscr{W}_3(m)$. We can therefore write the pharmacophore kernel (eq 6) as a walk-based graph kernel:

$$K(m,m') = \sum_{p\in\mathscr{P}(m)} \sum_{p'\in\mathscr{P}(m')} K_{\mathscr{P}}(p,p') = \sum_{w\in\mathscr{W}_3(m)} \sum_{w'\in\mathscr{W}_3(m')} K_{\text{Walk}}(w,w')$$

where $K_{\text{Walk}}(w,w') = K_{\mathscr{P}}(p,p')$ for the pair of walks (w,w') corresponding to the pair of pharmacophores (p,p'). More precisely, consider a pair of pharmacophores $p = [(x_1,l_1),(x_2,l_2),(x_3,l_3)]$ and $p' = [(x'_1,l'_1),(x'_2,l'_2),(x'_3,l'_3)]$ and a corresponding pair of walks $w = (w_1,w_2,w_3,w_1)$ and $w' = (w'_1,w'_2,w'_3,w'_1)$. There is a direct equivalence between $K_{\mathscr{P}}$ and K_{Walk} if we choose to label the vertices of the graphs by the atom labels involved in the pharmacophore characterization and to label the edges by the Euclidian distance between the atoms they connect. Indeed, in this case, we can write

$$\begin{aligned} K_{\mathscr{P}}(p,p') &= \prod_{i=1}^{3} K_{\text{Feat}}(l_i,l'_i)\, K_{\text{Dist}}(||x_i - x_{i+1}||,||x'_i - x'_{i+1}||) \\ &= \prod_{i=1}^{3} K_{\text{Feat}}[l(w_i),l(w'_i)]\, K_{\text{Dist}}[l((w_i,w_{i+1})),l((w'_i,w'_{i+1}))] \\ &= K_{\text{Walk}}(w,w') \end{aligned}$$

A striking point of this kernel between walks is that it can be factorized along the edges of the walks:

$$\begin{aligned} K_{\text{Walk}}(w,w') &= \prod_{i=1}^{3} K_{\text{Feat}}[l(w_i),l(w'_i)]\, K_{\text{Dist}}[l((w_i,w_{i+1})), l((w'_i,w'_{i+1}))] \\ &= \prod_{i=1}^{3} K_{\text{Step}}[(w_i,w_{i+1}),(w'_i,w'_{i+1})] \end{aligned} \quad (14)$$

The pharmacophore kernel therefore formulates as a walk-based graph kernel, with a walk kernel factorizing along the edges of the walks. It follows from ref 21 that it can be computed by the formalism based on product graphs and powers of the adjacency matrix proposed in ref 28, if the adjacency matrix of the product graph is weighted by the walk-step kernels K_{Step} (eq 14). Consequently, the matrix **M**, defined in eq 12 and upon which is based the kernel computation of proposition 1, can be seen as a weighted adjacency matrix of a product graph defined on complete, atom-based, molecular factor graphs.

Note moreover that, in its way to characterize the molecular structure, the pharmacophore kernel bears some similarity with cyclic patterns kernels[29] where a molecule is represented by graph cycles, even though general cycles instead of cycles of size three are considered in this latter approach, and the graph corresponds to the 2D structure of the molecule.

6. FAST APPROXIMATION

As an alternative to the costly computation presented in section 4, we introduce in this section a fast approximation to the pharmacophore kernel based on a discretization of the pharmacophore space.

Our definition of pharmacophores is based on the atoms' 3D coordinates, but they can equivalently be characterized by the pairwise distances between atoms. To define discrete pharmacophores, we restrict ourselves to discrete sets of atom labels (e.g., the set of atom types), and we discretize uniformly the range of distances between atoms into a predefined number of bins. For example, if the interatomic distances lie in the 0−20 Å range, and we consider 10 bins to discretize the distances, the bins will correspond to the intervals 0−2, 2−4, ..., and 18−20 Å. Each distance is then mapped to the index of the bin into which it falls, and a discrete pharmacophore is defined by a triplet of atom labels together with a triplet of bin indices. More formally, if the distance range is discretized into p bins, the set of discrete pharmacophores is a finite set defined as $\mathscr{T}_3 = \mathscr{L}^3 \times [1,p]^3$, where $\mathscr{L}$ is the set of atom labels.

Consider the mapping $\phi^{3\text{pt}}$ from the set of molecules to the set of discrete pharmacophores $\mathscr{T}_3$, defined for the molecule m as $\phi^{3\text{pt}}(m) = [\phi_t(m)]_{t\in\mathscr{T}_3}$, where $\phi_t(m)$ is the number of times the pharmacophore t is found in the molecule m. This mapping leads to the following kernel definition.

Definition 3 (Three-Point Spectrum Kernel). For a pair of molecules m and m', we define the three-point spectrum kernel $K^{3\text{pt}}_{\text{Spec}}$ as

$$K^{3\text{pt}}_{\text{Spec}}(m,m') = \langle \phi^{3\text{pt}}(m),\phi^{3\text{pt}}(m') \rangle = \sum_{t\in\mathscr{T}_3} \phi_t(m)\, \phi_t(m') \quad (15)$$

Note that, if we define the mapping d: $\mathscr{P} \rightarrow \mathscr{T}_3$, such that $d(p)$ is the discretized version of the pharmacophore $p \in \mathscr{P}$, we can explicitly write the three-point spectrum kernel as a particular pharmacophore kernel (eq 6):

$$K^{3\text{pt}}_{\text{Spec}}(m,m') = \sum_{p\in\mathscr{P}(m)} \sum_{p'\in\mathscr{P}(m')} \mathbf{1}[d(p) = d(p')] \quad (16)$$

This equation shows that this is a crude pharmacophore kernel, based on a kernel for pharmacophores that simply checks if two given pharmacophores have identical discretized versions or not.

In addition, we consider a "two-point pharmacophore" version of the kernel (eq 15), based on pairs, instead of triplets, of atoms.[30] Letting $\mathscr{T}_2$ be the set of all possible two-point pharmacophores, that is, pairs of atom types together with the bin index of the edge connecting them, and letting $\phi^{2\text{pt}}(m) = [\phi_t(m)]_{t\in\mathscr{T}_2}$ be the mapping of the molecule m to $\mathscr{T}_2$, corresponding to $\phi^{3\text{pt}}(m)$, we define the following *two-point spectrum* kernel.

Definition 4 (Two-Point Spectrum Kernel). For a pair of molecules m and m', we define the two-point spectrum kernel $K^{2\text{pt}}_{\text{Spec}}$ as

$$K^{2pt}_{Spec}(m,m') = \langle \phi^{2pt}(m), \phi^{2pt}(m') \rangle = \sum_{t \in \mathcal{T}_2} \phi_t(m)\, \phi_t(m') \tag{17}$$

This kernel shows strong similarities with recently introduced kernels for 3D structures of molecules[30] and is introduced as a baseline to validate the three-point pharmacophore characterization of molecules.

The kernels in eqs 15 and 17 are directly expressed as dot products and are consequently positive definite, which justifies their use with SVM. Moreover, these kernels can be computed efficiently using an algorithm derived from that used in the implementation of the spectrum string kernel.[31] We describe this algorithm for the three-point version of the kernel, its extension to the two-point kernel being straightforward. Following the notation of section 5, we represent molecules by complete, atom-based labeled graphs, with the difference that the set of atom labels $\mathcal{L}$ defining the vertices labels is considered to be discrete (e.g., the atom types), and the edges are now labeled by the bin index of the corresponding interatomic distance. We consider the problem of computing the Gram matrix **K** associated with such a set of molecular graphs $\{G_i = (\mathcal{V}_{G_i}, \mathcal{E}_{G_i})\}_{i=1,\ldots,n}$ for the kernel in eq 15. The alphabet $\mathcal{A}$, involved in the graph labeling function l of section 5, is defined as $\mathcal{A} = L_V \cup L_E$, where L_V is the set of vertex labels, corresponding to the set of atom labels $\mathcal{L}$, and L_E is the set of edges labels, corresponding to the set of distance bins indices.

The algorithm is based on the manipulation of sets of walk pointers within each graph, according to a tree transversal process. If we let n and p be the cardinalities of L_V and L_E, respectively, we define a rooted, depth-four tree structuring the space of pharmacophores $\mathcal{T}_3$ as follows:

- The root node has n sons, corresponding to the n possible vertex labels.
- The depth-one and depth-two nodes have $n \times p$ sons, corresponding to the $n \times p$ possible pairs of edge and vertex labels.
- The depth-three nodes have p sons, corresponding to the p possible edge labels, a leaf node being implicitly associated with the vertex label of its depth-one ancestor. A path from the root to a leaf node therefore corresponds to a triplet of distinct vertex labels, together with a triplet of distinct edge labels. There is therefore a one-to-one correspondence between the leaf nodes and the pharmacophores of $\mathcal{T}_3$. The principle of the algorithm is to recursively transverse this tree until each leaf node (i.e., each potential pharmacophore) is visited. During this process, a set of walk pointers is maintained within each molecule. The pointers are recursively updated such that the pointed walks correspond to the pharmacophores under construction in the tree-transversal process. When reaching a leaf node, the pointed walks correspond to the occurrences of a particular pharmacophore t in the molecules. The mapping $\phi_t(G_i)$ can therefore be computed for the molecular graphs $\{G_i\}_{i=1,\ldots,n}$, and the kernel matrix **K** can be updated by adding the products $\phi_t(G_i)\, \phi_t(G_j)$ to its (i,j) entries.

A pseudo code of the algorithm is given in algorithms 1−4 (Chart 1). Algorithm 1 is the main program in charge of the tree-transversal process, and algorithms 2, 3, and 4 are subroutines, introduced to initialize the walk pointers, extend the pointed walks, and update the Gram matrix, respectively. This pseudocode relies on the abstract types *Pointer* and *Label*, to represent the walk pointers involved in the algorithm and the generic vertices and edges labels, belonging to L_V and L_N, respectively. Formally, a *Pointer* object consists of two graph vertices: a *start* and a *current* vertex, representing the first and the current vertices of the pointed walk being extended. To maintain walk pointers within each molecule, we introduce a matrix of pointers walkPointers = Pointer[][]: this matrix is initially empty, and during the walk extension process, walkPointers[i][j] corresponds to the jth pointer of the molecular graph G_i. The stopping criterion of the recursion is controlled by an integer variable *depth* corresponding to the depth in the tree during the transversal process. It is initialized to zero and incremented at each recursive call. When depth is three, a depth-three node is reached in the tree, which corresponds to pointers on length-two walks in the graphs. In the subsequent recursive step, depth is four, and the pointers are updated to ensure that the extended walks correspond to self-returning ones. A leaf node is then reached, and the recursion terminates, leading to an update of the Gram matrix. Note, however, that the recursion is aborted whenever the set of walk pointers becomes empty for all graphs, because we only need to reach the leaf nodes corresponding to the pharmacophores truly present in the set of graphs.

Computing the Gram matrix **K** simply requires a call to the *COMPUTE* function of algorithm 1 with the following arguments: *Pointers*, the empty Pointer matrix; *depth* initialized to zero; and **K**, the $n \times n$ Gram matrix filled with zeros.

The cost of this algorithm depends on the number of leaf nodes visited and is therefore bounded by the total number of leaves of the tree, that is, $(np)^3$ if the number of distinct vertex labels is n and the number of distance bins is p. However, the maximum number of distinct pharmacophores that can be found in the molecule m is $|m|^3$, and we do not need to exhaustively transverse the tree. This means that, to compute the kernel between the molecules m and m', at most, $\min(|m|^3, |m'|^3)$ leaves, corresponding to the common pharmacophores of m and m', need to be visited. The complexity of the algorithm is therefore $\mathcal{O}[\min[(np)^3, \min(|m|^3, |m'|^3)]]$. For small molecules, the cost of the kernel will therefore depend on their number of atoms, while it will depend on the size of the discrete pharmacophore space for large molecules.

Note finally that, although we omit the details, the previous algorithm and complexity analysis hold for the two-point version of the kernel: the tree involved in the recursive transversal process is smaller (a depth-two tree, with n^2p leaf nodes), and the complexity is reduced to $\mathcal{O}[\min[n^2p, \min(|m|^2, |m'|^2)]]$.

7. EXPERIMENTS

We now turn to the experimental section. The problem considered here consists of building predictive models to distinguish *active* from *inactive* molecules on several protein targets. This problem is naturally formulated as a supervised binary classification problem that can be solved by SVM.

7.1. Data Sets. We tested the pharmacophore kernel on several data sets used in a recent SAR study.[32] More precisely, we considered the following four publicly available

Chart 1. Algorithms 1−4

Algorithm 1 main program

```
COMPUTE(Pointer[][] walkPointers, Integer depth, Float[][] K)
  depth = depth + 1
  if depth = 1 then
    for label ∈ L_V do
      walkPointers = initPointers(label)
      compute(walkPointers, depth, K)
    end for
  else
    for label_1 ∈ L_V do
      for label_2 ∈ L_E do
        walkPointers = extendPointers(walkPointers, depth, label_1, label_2)
        if walkPointers ≠ [][] then
          if depth = 4 then
            updateGram(walkPointers, K)
          else
            compute(walkPointers, depth, K)
          end if
        end if
      end for
    end for
  end if
```

Algorithm 2 Sub-routine 1 : initialize walks pointers

```
INITPOINTERS(Label label)
  walkPointers = Pointer[][]
  for i = 1, ..., n do
    for v ∈ V_{G_i} do
      if l(v) = label then
        walkPointers[i].addPointer(start = v, current = v)
      end if
    end for
  end for
  return walkPointers
```

Algorithm 3 Sub-routine 2 : extend walks pointers

```
EXTENDPOINTERS(Pointer[][] walkPointers_in, Integer depth, Label label_1, Label label_2)

  walkPointers_out = Pointer[][]
  for i = 1, ..., n do
    for ptr ∈ walkPointers_in[i] do
      for (ptr.current, v) ∈ E_{G_i} do
        if l(v) = label_1 ∧ l((ptr.current, v)) = label_2 then
          if depth ≠ 4 ∨ v = ptr.start then
            walkPointers_out[i].addPointer(start = ptr.start, current = v)
          end if
        end if
      end for
    end for
  end for
  return walkPointers_out
```

Algorithm 4 Sub-routine 3 : update Gram matrix

```
UPDATEGRAM(Pointer[][] walkPointers, Float [][] K)
  for i = 1, ..., n do
    for j = 1, ..., n do
      if walkPointers[i] ≠ [] ∧ walkPointers[j] ≠ [] then
        K[i][j] = K[i][j] + walkPointers[i].size() × walkPointers[j].size()
        if i ≠ j then
          K[j][i] = K[j][i] + walkPointers[i].size() × walkPointers[j].size()
        end if
      end if
    end for
  end for
```

data sets (available as Supporting Information of the original study at http://pubs.acs.org/journals/jcisd8/): the BZR data set, a set of 405 ligands for the benzodiazepine receptor; the COX data set, a set of 467 cyclooxygenase-2 inhibitors; the DHFR data set, a set of 756 inhibitors of dihydrofolate reductase; and the ER data set, a set of 1009 estrogen receptor

Table 1. Basic Information about the Data Sets Considered

	train		test	
	positive	negative	positive	negative
BZR	94	87	63	62
COX	87	91	61	64
DHFR	84	149	42	118
ER	110	156	70	110

ligands. These data sets contain the 3D structures of the molecules, together with a quantitative measure of their ability to inhibit a biological mechanism. The reference paper[32] presents a data preparation scheme sought to mimic a real virtual screening application: data sets were first filtered to prevent structural redundancy in the compounds considered and were further split into training and test sets such that the compounds used for testing are as structurally different as possible from those used for training. To have a reference result to which to compare, we kept this particular data preparation scheme. Table 1 gathers basic information about the data sets involved in the study.

7.2. Experimental Setup. We investigated in this study a simple labeling scheme to describe each atom (hydrogen atoms were systematically removed) and, therefore, the potential pharmacophores: the label of an atom is composed of its type (e.g., *C*, *O*, *N*, ...) and the sign of its partial charge (+, −, or 0). Hence, the set of labels can be expanded as $\mathcal{L} = \{C^+, C^0, C^-, O^+, O^0, O^-, ...\}$. The partial charges account for the contribution of each atom to the total charge of the molecule and were computed with the QuacPAC software developed by OpenEye (http://www.eyesopen.com/products/applications/quacpac.html). It is important to note that, contrary to the physicochemical properties of atoms, partial charges depend on the molecule and describe the spatial distribution of charges. Although the partial charges take continuous values, we simply kept their signs for the labeling as basic indicators of charges in the description of pharmacophores. We call *categorical kernel* the kernel resulting from this labeling, where the kernel between labels K_{Feat} is the Dirac kernel (eq 11) and the kernel between distances K_{Dist} is the Gaussian RBF kernel (eq 10).

Alternatively, we tested several variants of this basic categorical kernel. On one hand, we tested the effect of the partial charges by removing them from the labels and keeping the same Dirac and Gaussian RBF kernels for the labels and distances, respectively. In this case, the label of an atom reduces to its type. On the other hand, we tested the fast approximation and its two-point counterpart mentioned in section 6 with our original labeling scheme, that is, atoms labeled by their types and the sign of their partial charges.

In addition, we tested the state-of-the-art Tanimoto kernel based on the 2D structure of molecules[19] to evaluate the potential gain obtained by including 3D information. This kernel is defined as the Tanimoto coefficient between fingerprints indicating the presence or absence of all possible molecular fragments of a length up to eight in the 2D structure of the molecule, where a fragment refers to a sequence of atoms connected by covalent bonds. We note that this fingerprint is similar to classical 2D fingerprints such as the Daylight representation (http://www.daylight-.com/dayhtml/doc/theory/theory.toc.html), with the difference that our implementation does not require the folding of the fingerprint into a small-size vector.[18]

The different kernels were implemented in C++ within the open-source ChemCpp toolbox (available at http://chemcpp.sourceforge.net), and the SVM experiment was conducted with the open-source Python machine learning package PyML (http://pyml.sourceforge.net). The SVM prediction is obtained by taking the sign of a score function (eq 1). However, by varying this zero decision threshold, it is possible to compute the evolution of the true-positive rate versus the false-positive rate in a curve known as the receiver operating characteristic (ROC) curve. The area under the ROC curve (AUC) is known to be a safer indicator of the quality of a classifier than its accuracy,[33] being 1 for an ideal classifier and 0.5 for a random classifier. For each experiment, all parameters of the kernel and the SVM were optimized over a grid of possible choices on the training set only, to maximize the mean AUC over an internal 10-fold cross-validation.

The results on the test set correspond to the performance of the SVM with the selected parameters only. The optimized parameters include the width $\sigma \in \{0.1, 1, 10\}$ (in angstroms) of the Gaussian RBF kernel used to compare distances, the soft-margin parameter of the SVM over the grid {0.1, 0.5, 1, 1.5, ..., 20}, and the number of bins used to discretize the distances for the fast approximations over the grid {4, 6, 8, ..., 30}.

7.3. Results. Table 2 shows the results of classification for the different kernel variants. Each line corresponds to a kernel and reports several statistics: the accuracy (fraction of correctly classified compounds), sensitivity (fraction of positive compounds that were correctly classified), specificity (fraction of negative compounds that were correctly classified), and AUC. The first line corresponds to the basic categorical kernel. The following three lines show the results of the variants of the categorical kernel: the reduction of the atom labels to their types (i.e., categorical kernel without partial charges) and the fast approximation of the kernel (i.e., three-point spectrum kernel), together with its two-point counterpart. Finally, we added the performance obtained by the state-of-the-art 2D Tanimoto kernel, based on the 2D structure of the molecules, and the best results reported in the reference publication.[32] This latter method, labeled "Sutherland" in Table 2, is based on descriptors inherited from the 2D structure and the atomic composition of the molecules, which are selected using a genetic algorithm.

The results of parameter optimization on the training set often led to similar choices for different kernels. For example, the width of the Gaussian RBF kernel to compare distances was usually selected at 0.1 Å, which corresponds to a very strong constraint on the pharmacophore matching. Finally, the number of bins selected by the fast approximations to discretize the distances was usually between 20 and 30 bins.

We can first observe from Table 2 that removing the partial charges from atom labels decreases the accuracy by 2−4%, corresponding to a relative variation of 3−5%, on all data sets except COX. This superiority in accuracy of the categorical kernel is significant at a *p* value of $p = 0.125$, according to the one-sided Wilcoxon signed-rank test for paired data[34] based on the accuracy statistic, which suggests that the partial charge information is important for the definition of pharmacophores.

Morevover, the fast pharmacophore kernel obtained by applying a Dirac kernel to check when pairs of candidate

Table 2. Classification of the Test Sets, after Model Selection on the Training Set

	BZR				COX				DHFR				ER			
	acc.	sens.	spec.	AUC	acc.	sens.	spec.	AUC	acc.	sens.	spec.	AUC	acc.	sens.	spec.	AUC
categorical	76.4	74.0	78.9	82.1	69.8	69.8	69.8	75.1	81.9	63.3	88.8	84.8	79.8	72.0	84.7	86.8
categorical, no partial charges	74.3	73.6	75.0	81.5	70.0	68.5	70.9	74.6	78.1	65.2	82.7	82.2	77.6	71.7	81.4	87.2
three-point spectrum	75.4	74.4	76.3	81.3	67.0	64.4	69.5	75.9	76.9	70.9	79.0	81.9	78.6	78.3	78.8	87.4
two-point spectrum	71.4	61.3	81.6	80.3	68.9	70.2	67.7	74.7	67.7	67.4	67.9	72.3	78.7	75.9	80.4	84.5
2D-Tanimoto	71.2	71.9	70.5	80.8	63.0	67.5	58.6	69.8	76.9	73.8	78.0	83.0	77.1	69.3	82.1	83.6
Sutherland, ref 32	75.2	70.0	81.0		73.6	75.0	72.0		71.9	74.0	71.0		78.9	77.0	80.0	

Table 3. Computation Times in Minutes Needed to Compute the Different Kernel Matrices on the BZR Training Set[a]

	exact	discrete
with charges	20′	6′
without charges	249′	7′

[a] The first column refers to the computation of the exact kernel (eq 7) and the second one to the approximate kernel (eq 15).

pharmacophores fall in the same bin of the discretized space (three-point spectrum kernel) systematically degrades the accuracy by 1−5%, corresponding to a relative variation of 1−6%, over all four data sets compared to the categorical kernel. This is significant at a p value of $p = 0.062$ and suggests that the gain in computation time obtained by discretizing the space and computing a 3D-fingerprint-like representation of molecules has a cost in terms of accuracy of the final model. A particular limitation of the fingerprint-based method is that two pharmacophores could remain unmatched if they fall into two different bins, although they might be very similar but close to the bins' boundaries. In the case of the pharmacophore kernel, such pairs of similar pharmacophores would always be matched.

We observe finally that, except for the COX data set, the discrete kernel based on two-point pharmacophores leads to worse accuracy results than its three-point counterpart. This tends to highlight the benefits of the three-point pharmacophore characterization of the molecular structure, but this is only significant at a p value of $p = 0.312$.

For each data set, the results obtained with the 2D-Tanimoto kernel are significantly worse than those of the categorical kernel, with a decrease ranging from 3 to 7%, corresponding to a relative variation of 3−10%, on the different data sets. This is significant at a p value of $p = 0.062$ and confirms the relevance of 3D information for drug activity prediction, which motivated this work. Finally, we note that, on all but the COX data set, the categorical kernel outperforms the best results of ref 32. This tends to confirm the competitiveness of our method compared to state-of-the-art methods, but these latter results are only significant at a p value of $p = 0.312$.

Regarding the computational complexity of the different methods, Table 3 shows the time required to compute the kernel matrices on the BZR training set for different kernels, on a desktop computer, equipped with a Pentium 4 3.6 GHz processor with 1 GB of RAM. In the discrete version, the distance range was split into 24 bins, and as expected, the kernels based on the discretization of the pharmacophore space are faster than their counterparts by a factor of 4−35, depending on the type of labels used (with or without the partial charge information). In the exact kernel computation, the effect of removing the partial charges from the labels is to induce more matches between atoms and, therefore, as discussed in section 4, to drastically slow the computation by a factor of 12, consistent with the theoretical estimate that dividing the size of the label classes by k increases the speed by a factor of k^3.

8. DISCUSSION AND CONCLUSION

This paper presents an attempt to extend the application of recent machine learning algorithms for classification to the manipulation of 3D structures of molecules. This attempt is mainly motivated by applications in drug activity prediction, for which 3D pharmacophores are known to play important roles. Although previous attempts to define kernels for 3D structures (similar in fact to the two-point spectrum kernel we tested) led to mixed results,[30] we obtained performance competitive with that of state-of-the-art algorithms for the categorical kernel based on the comparison of pharmacophores contained in the two molecules to be compared. This kernel is not an inner product between fingerprints and, therefore, fully exploits the mathematical trick that allows SVM to manipulate measures of similarities rather than explicit vector representations of molecules, as opposed to other methods such as neural networks. We even observed that, for the closest fingerprint-based approximation obtained by discretizing the space of possible pharmacophores (three-point spectrum kernel), the performance significantly decreases. This highlights the benefits that can be gained from the use of kernels, which provide a satisfactory answer to the common issue of choosing a "good" discretization of the pharmacophore space to make fingerprints: once discretized, pharmacophores falling on different sides of bins' edges do not match, although they might be very close. We notice that approaches based on fuzzy fingerprints,[35] for example, aim at correcting this effect by matching pharmacophores based on different distance bins.

Concerning the practical use of our approach for the screening of large data sets, Table 3 shows that, even for the fastest variants, the approach based on kernel methods can be computationally demanding even for relatively small data sets. In practice, however, the time to train the SVM can be smaller than the times presented in Table 3 because not all entries of the matrix are required. Speeding up SVM and kernel methods for large data sets is currently a topic of interest in the machine learning community, and applications in virtual screening on large databases of molecules will certainly benefit from the advances in this field.

Among the possible extensions to our work, a promising direction that is likely to be relevant for many real-world applications is to take into account different conformers of each molecule. Indeed, it is well-known that the biological activity to be predicted is often due to one out of several

conformers for a given molecule, which suggests the representation of a molecule not as a single 3D structure but as a set of structures. This problem, known as multi-instance learning, has been drawing considerable interest in the machine learning community since its initial formulation.[36] The SVM and kernel approaches lend themselves particularly well to this extension, thanks to the possibility of defining kernels between sets of structures from a kernel between structures,[37] and extensions of the SVM algorithm.[38] A second direction would be to test and validate different definitions and labeling for the vertices of the pharmacophores. We limited ourselves to the simplest possible three-point pharmacophores based on single atoms annotated by their types and partial charges. The method could be improved by testing other schemes known to be relevant features as basic components of pharmacophores. It is, for example, possible to consider groups of atoms forming functional units instead of single atoms to form pharmacophores. Alternatively, the atom labels considered in this work may be enriched with the introduction of various physicochemical properties known to account for the steric and electrostatic behavior of atoms. As a first step in this direction, we investigated a labeling scheme based on a set of four physicochemical properties (namely, the atomic van der Waals and covalent radii, electronegativity, and first ionization energy), but the corresponding results were not convincing: they were globally similar to those obtained with atom types labels without partial charges. This is actually not really surprising because these properties are deduced from the atom types and, therefore, bear little additional information, contrary to the partial charges which depend on the molecular conformation. A third possible extension is to generalize this work to pharmacophores with more points, for example, four or five. Although several results will not remain valid in this case, such as the expression of the kernel as the trace of a matrix, this could lead to more accurate models in cases where the binding mechanism is well-characterized by such pharmacophores. Finally, we note that several approaches were recently proposed to derive a measure of similarity between structured objects from the similarity of their substructures.[39−41] These approaches could generalize the present work to alternative measures of similarity between 3D structures based on pharmacophore similarity.

Note Added after ASAP Publication. This article was released ASAP on August 12, 2006, with extraneous text in section 2. The correct version was posted on August 18, 2006.

REFERENCES AND NOTES

(1) Manly, C.; Louise-May, S.; Hammer, J. The impact of informatics and computational chemistry on synthesis and screening. *Drug Discovery Today* **2001**, *6* (21), 1101−1110.

(2) Butina, D.; Segall, M. D.; Frankcombe, K. Predicting ADME properties in silico: Methods and models. *Drug Discovery Today* **2002**, *7* (Suppl. 11), S83−S88.

(3) Lipinski, C. A.; Lombardo, F.; Dominy, B. W.; Feeney, P. J. Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. *Adv. Drug. Delivery Rev.* **2001**, *46* (1−3), 3−26.

(4) King, R. D.; Muggleton, S. H.; Srinivasan, A.; Sternberg, M. J. E. Structure−activity relationships derived by machine learning: The use of atoms and their bond connectivities to predict mutagenicity by inductive logic programming. *Proc. Natl. Acad. Sci. U.S.A.* **1996**, *93* (1), 438−442.

(5) Böhm, H. J.; Schneider, G.; Mannhold, R.; Kubinyi, H.; Folkers, G. *Protein−Ligand Interactions*; Wiley: New York, 2003.

(6) Halperin, I.; Ma, B.; Wolfson, H.; Nussinov, R. Principles of docking: An overview of search algorithms and a guide to scoring functions. *Proteins* **2002**, *47* (4), 409−443.

(7) Lemmen, C.; Lengauer, T. Computational methods for the structural alignment of molecules. *J. Comput.-Aided Mol. Des.* **2000**, *14* (3), 215−232.

(8) Xue, L.; Bajorath, J. Molecular descriptors in chemoinformatics, computational combinatorial chemistry, and virtual screening. *Comb. Chem. High. Throughput Screening* **2000**, *3* (5), 363−372.

(9) Holliday, J. D.; Willett, P. Using a genetic algorithm to identify common structural features in sets of ligands. *J. Mol. Graphics Modell.* **1997**, *15* (4), 221−232.

(10) Finn, P.; Muggleton, S. H.; Page, D.; Srinivasan, A. Pharmacophore discovery using the inductive logic programming language *Prog. Machine Learning* **1998**, *30* (2−3), 241−270.

(11) Matter, H.; Pötter, T. Comparing 3D pharmacophore triplets and 2D fingerprints for selecting diverse compound subsets. *J. Chem. Inf. Model.* **1999**, *39* (6), 1211−1225.

(12) Brown, R. D.; Martin, Y. C. The information content of 2D and 3D structural descriptors relevant to ligand−receptor binding. *J. Chem. Inf. Comput. Sci.* **1997**, *37* (1), 1−9.

(13) Bajorath, J. Selected concepts and investigations in compound classification, molecular descriptor analysis, and virtual screening. *J. Chem. Inf. Comput. Sci.* **2001**, *41* (2), 233−245.

(14) Boser, B. E.; Guyon, I. M.; Vapnik, V. N. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory*; ACM Press: New York, 1992; pp 144−152.

(15) Vapnik, V. N. *Statistical Learning Theory*; Wiley: New York, 1998.

(16) Burbidge, R.; Trotter, M.; Buxton, B.; Holden, S. Drug design by machine learning: Support vector machines for pharmaceutical data analysis. *Comput. Chem.* **2001**, *26* (1), 4−15.

(17) Byvatov, E.; Fechner, U.; Sadowski, J.; Schneider, G. Comparison of support vector machine and artificial neural network systems for drug/nondrug classification. *J. Chem. Inf. Comput. Sci.* **2003**, *43* (6), 1882−1889.

(18) 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* (4), 939−951.

(19) Ralaivola, L.; Swamidass, S. J.; Saigo, H.; Baldi, P. Graph kernels for chemical informatics. *Neural Networks* **2005**, *18* (8), 1093−1110.

(20) Schölkopf, B.; Smola, A. J. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*; MIT Press: Cambridge, MA, 2002.

(21) Kashima, H.; Tsuda, K.; Inokuchi, A. Kernels for graphs. In *Kernel Methods in Computational Biology*; Schölkopf, B., Tsuda, K., Vert, J. P., Eds.; MIT Press: Cambridge, MA, 2004; pp 155−170.

(22) Burges, C. J. C. A tutorial on support vector machines for pattern recognition. *Data Min. Knowledge Discovery* **1998**, *2* (2), 121−167.

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

(24) Aizerman, M. A.; Braverman, E. M.; Rozonoér, L. I. Theoretical foundations of the potential function method in pattern recognition learning. *Autom. Remote Control (Engl. Trans.)* **1964**, *25*, 821−837.

(25) Aronszajn, N. Theory of reproducing kernels. *Trans. Am. Math. Soc.* **1950**, *68*, 337−404.

(26) Güner, O. F. *Pharmacophore Perception, Development, and Use in Drug Design*; International University Line: La Jolla, CA, 2000; Vol. 2, IUL Biotechnology Series.

(27) Haussler, D. *Convolution Kernels on Discrete Structures*; Technical Report UCSC-CRL-99-10, U. C. Santa Cruz: Santa Cruz, CA, 1999.

(28) Gärtner, T.; Flach, P.; Wrobel, S. On graph kernels: Hardness results and efficient alternatives. In *Proceedings of the Sixteenth Annual Conference on Computational Learning Theory and the Seventh Annual Workshop on Kernel Machines*; Schölkopf, B., Warmuth, M., Eds.; Heidelberg: Heidelberg, Germany, 2003; Vol. 2777, Lecture Notes in Computer Science, pp 129−143.

(29) Horváth, T.; Gärtner, T.; Wrobel, S. Cyclic pattern kernels for predictive graph mining. In *Proceedings of the Tenth International Conference on Knowledge Discovery and Data Mining*; ACM Press: New York, 2004; pp 158−167.

(30) 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* (Suppl. 1), i359−i368.

(31) Leslie, C.; Eskin, E.; Noble, W. S. The spectrum kernel: A string kernel for SVM protein classification. In *Proceedings of the Pacific Symposium on Biocomputing 2002*; Altman, R. B, Dunker, A. K.,

Hunter, L., Lauerdale, K., Klein, T. E., Eds.; World Scientific: River Edge, NJ, 2002; pp 564−575.
(32) 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* (6), 1906−1915.
(33) Fawcett, T. *ROC graphs: Notes and practical considerations for data mining researchers*; Technical Report 2003-4; HP Laboratories: Palo Alto, CA, 2003.
(34) Demšar, J. Statistical Comparisons of Classifiers over Multiple Data Sets. *J. Machine Learning Res.* **2006**, *7*, 1−30.
(35) Horvath, D.; Jeandenans, C. Neighborhood behavior of in silico structural spaces with respect to in vitro activity spaces—A novel understanding of the molecular similarity principle in the context of multiple receptor binding profiles. *J. Chem. Inf. Comput. Sci.* **2003**, *43* (2), 680−690.
(36) Dietterich, T. G.; Lathrop, R. H.; Lozano-Perez, T. Solving the multiple instance problem with axis-parallel rectangles. *Artif. Intell.* **1997**, *89* (1−2), 31−71.
(37) Gärtner, T.; Flach, P. A.; Kowalczyk, A.; Smola, A. J. Multi-instance kernels. In *Proceedings of the Nineteenth International Conference on Machine Learning*; Sammut, C., Hoffmann, A., Eds.; Morgan Kaufmann: San Francisco, CA, 2002; pp 179−186.
(38) Andrews, S.; Hofmann, T.; Tsochantaridis, I. Multiple instance learning with generalized support vector machines. In *Proceedings of the Eighteenth National Conference on Artificial Intelligence*; American Association for Artificial Intelligence: Edmonton, Alberta, Canada, 2002; pp 943−944.
(39) Cuturi, M.; Fukumizu, K.; Vert, J. P. Semigroup kernels on measures. *J. Machine Learning Res.* **2005**, *6*, 1169−1198.
(40) Wolf, L.; Shashua, A. Learning over sets using kernel principal angles. *J. Machine Learning Res.* **2003**, *4*, 913−931.
(41) Jebara, T.; Kondor, R.; Howard, A. Probability Product Kernels. *J. Machine Learning Res.* **2004**, *5*, 819−844.

CI060138M