TITLE:

# Kernel methods in computational biology

AUTHOR(S):

Vert, Jean-Philippe; 阿久津, 達也

CITATION:

Vert, Jean-Philippe ...[et al]. Kernel methods in computational biology. 物性研究 2003, 81(1): 142-155

ISSUE DATE:

2003-10-20

URL:

http://hdl.handle.net/2433/97608

RIGHT:

京都大学学術情報リポジトリ
KURENAI 紅
Kyoto University Research Information Repository

# Kernel methods in computational biology

Jean-Philippe Vert (Ecole des Mines de Paris)

（日本語要旨）

## 計算生物学におけるカーネル法

原稿作成： 阿久津達也（京大化学研究所）

### 1. 緒言

計算生物学の研究目的の一つは、実験的研究により生成される大量のデータを解析し、生物学的に有用な仮説を自動的に導くための計算手法を開発することである。また、生物学においては多種多様なデータが生成されるため、それらを統合して扱うことのできる数学的枠組みを見出すことは重要な課題の一つである。

計算生物学において対象となるデータには、遺伝子配列データ、化学構造データ、遺伝子発現データなどがあるが、ここでは、これらを統一的に扱うことを可能とするカーネル法について説明する。カーネル法はここ十年間に機械学習分野において発展してきた手法であり、生物学を含む数多くの問題に応用されている。

### 2. Mercer カーネル

集合 X の直積から実数へ関数 K(x,y) が、対称性 (K(x,y)=K(y,x)) を満たし、さらに、正定値性を満たす場合に、関数 K(.,.) は Mercer カーネルと呼ばれる。K(.,.) が Mercer カーネルである場合、あるヒルベルト空間Φ、および、X からΦへの関数 $\phi$ (x) が存在し、K(x,y) は $\phi$ (x) と $\phi$ (y) の内積となる。より、厳密には RKHS(reproducing kernel Hilbert space) と呼ばれるヒルベルト空間を用いることにより、Mercer カーネルと RKHS を対応づけすることができる。また、RKHS の重要な性質として、RKHS が無限次元空間であっても、ある条件下で正則化された関数の最小化が有限個の点のみを考慮することで行えるということがあげられる。

### 3. カーネル法

カーネル法の大きな利点の一つとして、ヒルベルト空間へ写像すること無しに種々の計算が行えることがあげられ、このことはカーネルトリックと呼ばれる。簡単な例としてはヒルベルト空間における 2 点間の距離がカーネル関数の簡単な組み合わせで求めることができる。より有用な例として、統計解析の主要手法の一つである主成分分析 (PCA) が、カーネルを用いた場合にも、ヒルベルト空間における計算なしに行える。カーネルを用いた正準相関分析 (CCA) は固有値計算問題に帰着することができ、二種類のデータを統合した解析を行うのに有用である。

サポートベクターマシン (SVM) はカーネル法に基づく（教師あり）機械学習のための手法で、正負の例が与えられた時、正負の例を分離し、かつ、最近点までの距離（マージン）が最大となる超平面を計算する。実際には、正負の例を完全に分離することが不可能である場合が多いので、分類誤差と距離をトレードオフしたものを最適化する。SVM では、カーネルトリックにより、最適な分離超平面が（多くの場合には少ないサイズの）正負の例の部分集合に対するカーネルの組み合わせにより表現される。

### 4. タンパク質データに対するカーネル法

カーネル法を生物学データに適用するため、タンパク質や関連するデータに対するカーネル関数が提案されている。特に、配列（文字列）に対するカーネル関数はよく研究されている。長さ k の部分文字列の出現頻度のベクトルを用いることにより、文字列からユークリッド空間へのカーネル関数を定義できるが、この手法は spectrum カーネルと呼ばれている。また、配列解析に広く利用されている確率モデルである隠れマルコフモデル (HMM) などから情報を抽出することによりカーネル関数を定義する、Fisher カーネルも提案されている。

配列データ以外には、遺伝子発現データ、Phylogenetic Profile などを扱うためのカーネルや、グラフ構造に関する diffusion カーネルとカーネル CCA を組み合わせ代謝パスウェイと発現データの相関を抽出する研究などが行われている。

カーネルの組み合わせに関する研究も行われており、半正定値計画法による、カーネルの線形結合の最適化などが研究されている。

# Kernel methods in computational biology

Jean-Philippe Vert
Ecole des Mines de Paris
35 rue Saint-Honor
77300 Fontainebleau
Jean-Philippe.Vert@mines.org

## 1 Introduction

Computational biology in the post-genomics era aims at providing a computational environment for biological research, where huge amounts of data generated by high-throughput technologies can be integrated, and where biological hypotheses resulting from the automatic analysis of these data can be generated. Besides obvious data management issues, core problems in computational biology concern the definition of a coherent and useful mathematical framework to integrate data.

The purpose of this paper is to propose a mathematical framework to model large sets of objects, such as the set of all genes in a genome or of all chemicals in a cell, and to represent various types of information about these objects (such as the sequences, structures and expression of genes) in a unified framework. Moreover, a number of tools for analysis and inference are provided in this framework. It is based on the theory of Mercer kernels and reproducing kernel Hilbert spaces, while analysis and inference tools are derived from the recently developed theory of kernel methods which has attracted a lot of attention during the last decade in the machine learning community.

This framework should obviously neither be considered a unique nor a final solution to the problem of building a mathematical framework for post-genomics. There is probably no single answer to this need, but rather a collection of possible frameworks with different advantages and drawbacks. The one proposed in this paper is certainly very limited in terms of its capacity to represent complex relationships among objects, such as evolutionary or regulatory relationships among genes. However, the tools for statistical analysis and inference in this framework are reasonably powerful and have been shown to be useful in many applications, such as predicting the functions of genes. Hence we can say that this formalism is slightly biased toward deriving powerful inference tools, at the expense of modeling complex biological relationships.

The organization of the paper is the following. We first describe the mathematical theory of Mercer kernels and review some of their properties, as well as the family of kernel methods they underlie. We then turn our attention to biological systems such as the set of all genes of an organism, and argue that various types of information about the system can be encoded as Mercer kernels and can be considered as various realizations of a more abstract model of the systems.

The application of kernel methods in computational biology has recently been subject to much investigation. Even though we mention arbitrary references to illustrate some of our statements, this short paper is not a review of this field and we apologize for all the interesting recent contributions which we don't mention here.

## 2 Mercer kernels

Let $\mathcal{X}$ be a set supposed to represent objects one wants to analyze, e.g., the set of all genes in a given genome. We propose to represent any form of information about the objects by a Hilbert space structure on the set $\mathcal{X}$. Intuitively, the Hilbert product between two objects will be defined as a measure of similarity of the objects with respect to the information available. Obviously, every measure of similarity does not define a valid Hilbert product. For a function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ to be a valid product, it must be symmetric (i.e., $K(x,y) = K(y,x)$ for any $x, y \in \mathcal{X}$) and positive definite in the sense that for any $n \in \mathbb{N}$, any $(a_1, \ldots, a_n) \in \mathbb{R}^n$ and any $(x_1, \ldots, x_n) \in \mathcal{X}^n$, the following holds:

$$\sum_{i,j=1}^{n} a_i a_j K(x_i, x_j) \geq 0.$$

In that case we say that $K$ is a Mercer kernel. It is easy to see that if $\mathcal{X}$ is a Hilbert space with Hilbert product $< .,. >$, then $K(x,y) =< x, y >$ is a Mercer kernel on $\mathcal{X}$. Conversely, if $\mathcal{X}$ is a set and $K(.,.)$ a Mercer kernel on $\mathcal{X}$, then there exists a Hilbert space $\Phi$ and a mapping $\phi : \mathcal{X} \to \Phi$ such that the kernel between two points be the Hilbert product between their images: $K(x,y) =< \phi(x), \phi(y) >$.

An interesting construction of such a mapping $\phi$ is the reproducing kernel Hilbert space (RKHS), which we now define. Simply speaking, a (real) RKHS on $\mathcal{X}$ is a Hilbert space $\mathcal{H}$ of real-valued functions on $\mathcal{X}$ with the property that, for each $x \in \mathcal{X}$, the evaluation functional $L_x$ which associates any $f \in \mathcal{H}$ with $L_x f = f(x)$, is a bounded linear functional. The boundedness means that for any $x \in \mathcal{X}$, there exists a constant $M_x \in \mathbb{R}$ such that

$$\forall f \in \mathcal{H}, \quad |L_x f| = |f(x)| \leq M_x ||f||_{\mathcal{H}}.$$

We remark that this definition implies that the functions of the RKHS must be defined pointwise and thus that the familiar space $\mathcal{L}_2(\mathcal{X})$, if $\mathcal{X}$ is measurable, is not a RKHS. The link with Mercer kernels is given in the following theorem which shows an equivalence between RKHS and Mercer kernels [SS02]:

**Theorem 1** *Let $(\mathcal{H}, < .,. >)$ be a RKHS on a set $\mathcal{X}$. Then there exists a unique Mercer kernel $K$ on $\mathcal{X}$ with the following properties:*

1. *$K$ spans $\mathcal{H}$, i.e., $\mathcal{H}$ is the completion of span$\{K(x,.), x \in \mathcal{X}\}$.*
2. *$K$ has the reproducing property, i.e.,*

$$\forall (f,x) \in \mathcal{H} \times \mathcal{X}, \quad < f, K(x,.) >= f(x);$$

*in particular,*

$$\forall (x,x') \in \mathcal{X} \times \mathcal{X}', \quad < K(x,.), K(x',.) >= K(x,x'). \tag{1}$$

*Conversely, if $\mathcal{X}$ is a set and $K$ is a Mercer kernel on $\mathcal{X}$, then a these equations define uniquely a RKHS on $\mathcal{X}$.*

Equation 1 shows that when $\mathcal{X}$ is a set endowed with a Mercer kernel, then the mapping $\phi : x \to \phi(x) = K(x,.)$ from $\mathcal{X}$ to the RKHS defined by $K$ is a valid mapping from $\mathcal{X}$ to a Hilbert space that satisfies $K(x,y) =< \phi(x), \phi(y) >$, and justifies the fact that any Mercer kernel defines a Hilbert space structure on $\mathcal{X}$.

A number of interesting properties make Mercer kernels a very powerful tool in real-world applications. First, Mercer kernels can be designed in such a way that the norm in the RKHS be of interest for various purposes. As an example, suppose $\mathcal{X} = \mathbb{R}^d$, and take the Gaussian RBF kernel:

$$K(x, y) = \exp\left(-\frac{||x - y||^2}{2\sigma^2}\right),$$

which is a valid Mercer kernel. Then the norm in the associated RKHS is given by:

$$||f||_{\mathcal{H}}^2 = \frac{1}{2\pi\sigma^2} \int |\hat{f}(\omega)|^2 \exp\left(\frac{\sigma^2||\omega||^2}{2}\right) d\omega, \tag{2}$$

where $\hat{f}$ is the Fourier transform of $f$ (here $\mathcal{H}$ is the set of functions with a Fourier transform such that (2) be finite). In that case $||f||_{\mathcal{H}}$ is a smoothing functional, in the sense that it is small when $f$ is smooth, and large otherwise. This can typically be useful as a term for regularization, when one wants to minimize a function $\Omega(f)$ on $\mathcal{H}$ with simultaneously imposing some smoothness constraints on $f$ (otherwise the problem might be ill-posed). In that case, instead of minimizing $\Omega(f)$, one might want to minimize $\Omega(f) + \lambda||f||_{\mathcal{H}}$, where $\lambda$ controls the trade-off between the minimization of $\Omega$ and the smoothness of $f$.

A second remarkable propertiy of RKHS is that even though they are typically infinite-dimensional, a large class of regularized minimization problems in $\mathcal{H}$ can be solved exactly in a lower-dimensional space. The precise statement is contained in the following representer theorem [KW71, SS02]

**Theorem 2** *Let $\omega : [0, \infty) \to \mathbb{R}$ a strictly monotonic increasing function, $\mathcal{X}$ a set with a Mercer kernel $K$ and associated RKHS $\mathcal{H}$. Let $(x_1, \ldots, x_n) \in \mathcal{X}^n$ be a set of points in $\mathcal{X}$ and $\Omega$ be a function on $\mathcal{H}$ expressed in terms of the $x_i$ and $f(x_i)$ only, i.e., of the form:*

$$\forall f \in \mathcal{H}, \quad \Omega(f) = \Omega(x_1, f(x_1), \ldots, x_n, f(x_n)).$$

*Then each minimizer $f \in \mathcal{H}$ of the regularized function:*

$$\Omega(f) + \omega(||f||_{\mathcal{H}})$$

*admits a representation of the form*

$$f(x) = \sum_{i=1}^{n} \alpha_i K(x_i, .).$$

The Representer theorem has tremendous consequences on the design of methods and algorithms, including such techniques as splines [Wah90] and kernel methods [SS02]. It means that minimizing a regularized form of a functional which only depends on the value of $f$ on a finite number of points can be done by working in a finite-dimensional space, and reduces to finding coefficients $\alpha_i$ associated with each point $x_i$. A whole variety of algorithms can be deduced from this theorem by varying the function to be optimized and the Mercer kernels, ranging from generalized forms of principal component analysis to support vector machines for classification and regression.

# 3 Kernel methods

In this section we briefly review a number of algorithms known as kernel methods based on the Representer theorem. They all take as input a finite number of points in a set endowed with a Mercer kernels, such as the set of genes where a Mercer kernel has been defined as a similarity measure with respect to some sort of information, and provide useful tools to perform various analysis of these points. For each of these methods, two complementary points of views can help get an intuition of how they work. First, the direct point of view consists in viewing each point mapped by a function $\phi$ to some Hilbert space. The algorithms consist them in working implicitly in the resulting Hilbert space. Second, from a dual point of view, the Mercer kernel defines a RKHS. The algorithms consist then in optimizing a regularized functional in the RKHS. The equivalence between these two points of view is ensured by the construction of RKHS and the Representer theorem.

## 3.1 Computing a Euclidean distance

Suppose you want to define a Euclidean distance between two points $x$ and $x'$ of a space $\mathcal{X}$ endowed with a Mercer kernel $K$. Then an obvious solution is to consider the Hilbert distance between their images in a Hilbert space defined by $K$, i.e., to consider the following square distance:

$$\begin{aligned} d(x,x')^2 &= ||\phi(x) - \phi(x')||^2 \\ &=< \phi(x) - \phi(x'), \phi(x) - \phi(x') > \\ &=< \phi(x), \phi(x) > + < \phi(x'), \phi(x') > -2 < \phi(x), \phi(x') > \\ &= K(x,x) + K(x',x') - 2K(x,x'). \end{aligned}$$

The result is expressed in terms of $K$ only. This shows than once a kernel is given on a set $\mathcal{X}$, it defines naturally a Euclidean distance which can be obtained without computing the images of the points in the Hilbert space associated with $K$. This is an example of the *kernel trick*, which consists in working implicitly in a complex Hilbert space without ever seeing a point in that space.

## 3.2 Computing the distance between a point and the center of a cloud of points

Suppose you have a set of points $(x_1, \ldots, x_n)$ in a set $\mathcal{X}$ endowed with a Mercer kernel, and you want to compute how each point is "far" from the "average" of all points. One way to formalize this intuitive goal is to consider the set of points mapped to a Hilbert space associated with $K$ by a mapping $\phi$. Then a natural definition of "average" of all points is the center of mass of all all points, i.e.,

$$m = \frac{1}{n} \sum_{i=1}^{n} \phi(x_i),$$

and the distance between a point $x_i$ and the average can be quantified by the square Euclidean distance between $\phi(x_i)$ and $m$, which can be computed by:

$$\begin{aligned} ||\phi(x_i) - m||^2 &=< \phi(x_i), \phi(x_i) > + < m, m > -2 < \phi(x_i), m > \\ &= K(x_i, x_i) - \frac{2}{n} \sum_{j=1}^{n} K(x_i, x_j) + \frac{1}{n^2} \sum_{j,k=1}^{n} K(x_j, x_k). \end{aligned}$$

Once again, the kernel trick enables to perform the computation implicitly in the feature space, and to obtain the result in terms of the function $K$ between the input points only.

### 3.3 Kernel principal component analysis

Suppose you want to analyze the variations among points $(x_1, \ldots, x_n)$ of a set $\mathcal{X}$ with a Mercer kernel $K$. A classical and useful tool in mathematical statistics is principal component analysis (PCA), which is defined when $\mathcal{X} = \mathbb{R}^d$. It is a powerful technique for extracting structure form possibly high-dimensional data sets, by projecting the points on the so-called principal axes of the data, defined as the directions which capture the largest amounts of variance [Jol96] (see Figure 1).

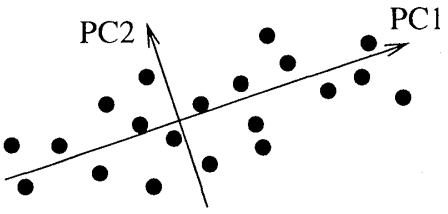


Figure 1: PCA detects the directions of largest variations among a family of points in a vector space, called principal directions.

The standard PCA algorithm is readily performed by solving an eigenvalue problem, i.e., by diagonalizing the covariance matrix:

$$C = \frac{1}{n} \sum_{i=1}^{n} x_i x_i^T,$$

where the data have been centered first ($\sum_{i=1}^{m} x_i = 0$). Suppose now that $\mathcal{X}$ is not a vector space, but simply a set endowed with a Mercer kernel. Then a useful way to analyze variations among the points and extract features is to perform a PCA on the points $\phi(x_i)$ mapped to a Hilbert space associated with $K$. As shown in [SSM99, SS02], this can be performed in a dual form by centering the data in the Hilbert space and diagonalizing the matrix

$$K^0_{i,j} =< \phi^0(x_i), \phi^0(x_j) >,$$

where $\phi^0(x_i)$ represents the point $\phi(x_i)$ after centering. Observe that while the size of the covariance matrix $C$ is $d \times d$, where $d$ is the dimension of the vector space $\mathcal{X}$ in classical PCA, the dimension of $K^0$ is $n \times n$, where $n$ is the number of points to analyze. Hence it becomes tractable to perform PCA in a possibly infinite-dimensional space, because the principal components are in fact always in the finite-dimensional space spanned by the points. Once again, the centering of the matrix, the computation of the principal directions and the projections of the points can be computed implicitly by only working with the function $K$.

### 3.4 Extracting correlations between two kernels

Suppose that two different Mercer kernels $K_1$ and $K_2$ are defined on the same set $\mathcal{X}$. This can be the case for instance when different notions of similarity are defined on the same objects, such as sequence similarity and co-expression for genes. The two kernels define two different mappings $\phi_1$ and $\phi_2$ to two Hilbert spaces. In order to compare the two kernels, and extract correlations between them, a useful tool in statistics is canonical correlation

analysis (CCA). When a set of points is given in simultaneously two finite-dimensional vector spaces, classical CCA consists in extracting pairs of directions, one in each space, such that the projections of the points on these directions be as correlated as possible [Hot36] (see Figure 2). Recently it was pointed out that a regularized form of CCA can be

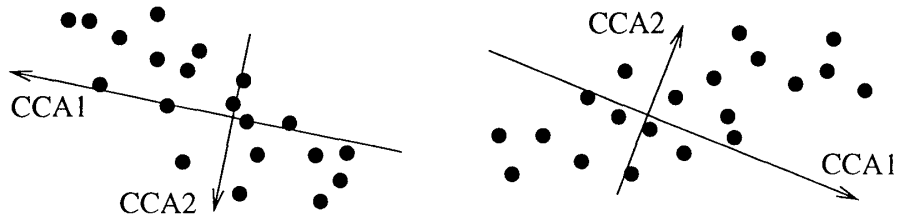


Figure 2: CCA detects directions simultaneously in two vector spaces such that the projections of the points be maximally correlated.

performed implicitly between two sets of points mapped to Hilbert spaces associated with Mercer kernels [BJ02], and that performing CCA in this context is equivalent to solving the following generalized eigenvalue problem:

$$\begin{pmatrix} 0 & K_1K_2 \\ K_2K_1 & 0 \end{pmatrix} \vec{\xi} = \rho \begin{pmatrix} K_1^2 & 0 \\ 0 & K_2^2 \end{pmatrix} \vec{\xi}$$

This provides a useful tool to extract correlations between two forms of informations encoded in two kernels, and has recently been applied in the context of natural language processing [AVC03], to extract language-independent semantic representations of texts, and in computational biology [VK, YVNK03], to compare gene networks and gene expressions or detect operons in prokaryote genomes.

### 3.5 Support vector machines

Support vector machines (SVM) are certainly the best-known kernel methods. Their popularity is due to their remarkable performances on many real-world problem, and they have been a main topic of investigation in the machine learning community during the last decade. First introduced by Vapnik and coworkers [BGV92, Vap98], SVM are a family of algorithms useful for classification and regression. In the simplest binary classification case, points $(x_1, \ldots, x_n)$ from a set $\mathcal{X}$ with a Mercer kernel $K$ are given together with a binary label associated with each point, denoted by $(y_1, \ldots, y_n)$ where $y_i \in \{-1, 1\}$. The goal of classification is to find a discrimination between positive and negative points. The solution implemented in SVM, motivated by theoretical results of statistical learning theory, is to take a linear discrimination between the positive and negative points mapped to the Hilbert space associated with $K$, such that the distance between the separating hyperplane and the closest point (called the margin) be the largest possible. This problem has a unique solution when the points are linearly separable, i.e., when there exists at least one hyperplane which separates positive points from negative ones. In the general situation, SVM finds an hyperplane that optimizes a trade-off between the number of misclassified points and the margin (see Figure 3). Once again this problem can be expressed in a dual formulation and results in the following optimization problem:

$$\begin{cases} \max_{\vec{\alpha}} \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j K(x_i, x_j), \\ \forall i = 1, \ldots, n \quad 0 \leq \alpha_i \leq C, \\ \sum_{i=1}^n \alpha_i y_i = 0, \end{cases}$$

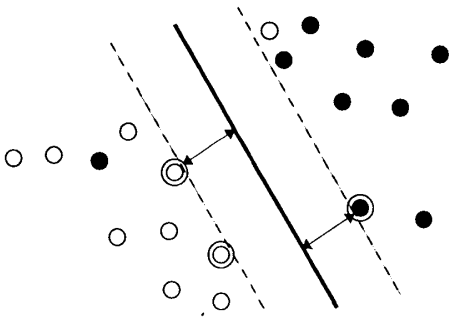

Figure 3: SVM finds a linear discrimination rule between two classes of points, by optimizing a functions which achieves a trade-off between the number of misclassified points (a small as possible) and the margin of the classification, i.e., the distance between the separating hyperplane and the closest point (which should be as large as possible).

where $C$ is the parameter which controls the trade-off between large margin and good classification. This algorithm is one of the most powerful algorithms availablenowadays on many real-world problems. In particular, as those equations suggest, if $\mathcal{X} = \mathbb{R}^D$, then the dimension $d$ does not matter too much on the performance of SVM (thanks to the choice of the largest margin linear classifier), or at least it seems to matter less for SVM than for many other classification algorithms which often perform poorly in large dimensions (this is part of a phenomenon called the curse of dimensionality).

SVM have been applied with success to many real-world problems. In the field of computational biology, a non-exhaustive list of applications using SVM include gene functional classification from microarrays [BGL+00, PWCG01], tissue samples classification from microarrays [MTM+98, FDC+00, GWBV02], protein family prediction from amino-acid sequences [JDH00], or protein secondary structure and localization prediction from amino-acid sequences [HS01b, HS01a].

## 4 Kernelizing the proteome

This quick overview of the theory of Mercer kernels, RKHS and kernel methods aimed at convincing the reader that they provide a framework with sound theoretical foundations, and which is associated with a panoply of powerful methods thanks to the simple Representer theorem. In this section, we now turn our attention to the question of the relevance of this framework to post-genomics.

One of the main challenges of computational biology in the post-genomics era is to integrate and process a variety of data generated by high-throughput technologies about biological objects, and use them to generate biological hypothesis. In the sequel we focus on problems related to functional genomics and proteomics, where one wants to infer knowledge about genes such as their functions, their regulation and their interactions from several different sources of information. Concretely, genes and proteins can be characterized by a number of different points a view: they can be defined by nucleotides or amino-acid sequences, by the structure of their promoter region in DNA, by their expression measured with DNA microarrays, by the secondary, tertiary and quaternary structure of the proteins they encode, by their role in metabolic and signaling pathways, by their interactions etc... Each characterization incorporates some partial representation of a more abstract concept of gene.

One approach to integrate these complementary characterizations in a unified frame-

work is to encode each type of information into a Mercer kernel defined on the set $\mathcal{X}$ of all genes (which can be considered a finite set when one focuses on the genes of a given genome, but can be extended to more complex structures such as the set of all finite-lengths nucleotide sequences for instance). This suggests a more abstract representation of a set of genes, as for instance a probability density on the set of kernel functions (e.g., the empirical measure corresponding to a set of kernel functions obtained by encoding different sources of informations). We won't develop further this avenue in this paper, but further investigations of the shape of this empirical measure is likely to provide interesting information about the relationships between different types of information. This could also provide an interesting notion of complexity of a biological system, as the entropy of the probability measure on the set of Mercer kernels used to represent it.

To illustrate this discussion we review in the following sections several recent attempts to encode biological information into kernel functions, as well as several attempts to explore the space of Mercer kernels obtained by different sources of information. By lack of space we will conclude with these onsiderations, but believe that exploring this framework is likely to result in new interesting conceptualizations of biological systems, as well as powerful tools to handle large amounts of heterogeneous data.

### 4.1 String kernel

Biological databases are full of sequences. A gene can be characterized as a subsequence of the DNA molecule (hence as a finite string over a four-letter alphabet), or by the sequence of amino-acids which forms the protein it encodes (hence as a finite string in over 20-letter alphabet). In both cases, the set of genes is a finite set of sequences, with a particular hidden structure. Many genes are known to share common ancestors, which often indicates functional or structural relationships. This evolutionary relationship can be detected to some extent by comparing the sequences of genes, because two genes sharing a common ancestor were similar a long time ago, but have evolved in different environments through mutations or insertion/deletion of letters or subsequences. As a result, measuring the "similarity" between gene sequences gives some information about the evolutionary, functional and structural relationships between genes.

It this context, a number of initiatives have been taken to engineer kernel functions for sequences to transform the sequence similarity relationship into a Hilbert space structure on the set of genes. Examples include the spectrum kernel [LEN02] which explicitly maps any sequence to its $k$-spectrum ($k > 0$) i.e., the number of appearance of each $k$-mer in the sequence, and defines the inner product between $k$-spectra as Hilbert product between sequences. This is motivated by the idea that even though the global appearance of a sequence might be strongly modified by several mutations during evolution, its $k$-spectrum might be more stable. This kernel was later improved in the mismatch kernel [LEWN03] which consists in convolution the $k$-kernel with a kernel that accepts up to $m < k$ mismatches, and provides a powerful tool to detect remote homology between proteins.

Another general approach for string kernel engineering is the Fisher kernel [JH99] which is a two-step process. First, a parametric statistical model for sequences, i.e., a family of probability densities $\{p_\theta(.), \theta \in \Theta \subset \mathbb{R}^k\}$ on the set of finite sequences, is built (e.g., a hidden Markov models [DEKM98]), and a parameter $\hat{\theta} \in \Theta$ is estimated to fit a set of sequences of interest. Second, each sequence $x$ is mapped to the score vector $\partial\theta \log p_\theta(x)$, i.e., to a vector in the tangent space at the point $p_{\hat{\theta}}$ of the Riemannian manifold formed by the statistical model. This manifold has a natural Riemannian metric [AN01] and a natural positive definite quadratic form is therefore available in the tangent space where sequences

are mapped. This ingenious approach is particularly useful because a lot of work has been devoted in the early days of bioinformatics to the development of parametric statistical models for biological objects, and the Fisher kernel provides a principled way to derive a kernel from virtually any such model. We can also mention other approaches to make kernels from probabilistic models, such as convolution kernels [Hau99, Wat00, SVUA03] which are based on probabilistic models for pairs of sequences, and can be computed efficiently using finite-state automata.

## 4.2 Expression profiles

With the development of DNA microarrays, it becomes increasingly simple to characterize a gene by an expression profile, i.e., a vector of real numbers which indicate its level of expression in different experiments. As genes are supposed to be expressed when the cell needs the proteins they encode, genes with similar profiles are likely to have related functions. Hence it might be useful to define a Hilbert product from these vectors, which can easily be achieved as a large number of kernel for real-valued vector have been known for a long time, such as the simple inner product or the Gaussian RBF kernel.

## 4.3 Other kernels

Here we briefly mention a non-exhaustive list of Mercer kernels which have been developed recently in the context of computational biology. As the number of sequenced organisms increases, classical homology detection tools can detect whether a given gene is present or absent in every fully sequenced organism. Hence a gene can be characterized by a vector of 0/1, which indicates its presence or absence in all sequenced organisms. This vector is called the phylogenetic profile of the gene [PMT+99]. Two genes with similar phylogenetic profiles are present and absent in the same organisms, hence chances are high that they have related function or participate to common structural complexes, for instance. While the simple dot product between vectors of bits provides a valid Mercer kernel, a more specific kernel for phylogenetic profiles was proposed in [Ver02b] and shown to be more relevant than the simple dot kernel. This kernel uses prior knowledge about the evolutionary relationships among sequenced organisms, and is one instance of a general approach to derive a kernel from probabilistic models when structural knowledge exists among the random variables [Ver02a].

An other interesting approach is the diffusion kernel [KL02] which is a kernel for the nodes of a graph. It is a discrete equivalent of the Gaussian RBF kernel, which can be computed as the matrix exponential of the opposite of the graph Laplacian. Intuitively, the kernel between two nodes increases when there are many short paths between them; more precisely, the kernel $K(x, y)$ is the probability of reaching the node $y$ by a random walk on the graph starting from the node $x$, and killed with some probability at each step. This kernel is known as the heat kernel [Chu97] in spectral graph theory, and is particularly relevant to define a kernel from a network of genes, for example interaction or regulation network. It was recently applied in combination with kernel canonical correlation analysis to extract correlations between metabolic pathways (which can be represented as a network of genes) and gene expression data [VK, VK03].

This short list is far from being exhaustive, and the list of kernels specifically developed for particular objects is increasing very quickly, providing a panoply of Mercer kernels to represent various types of information about genes.

### 4.4 Kernel operations

We showed in the previous sections that a number of kernels can be engineered to represent specific knowledges about genes. As the class of Mercer kernel is the class of symmetric positive definite functions, a variety of new kernels can be computed from a family of basic kernels using the closure properties of the class of symmetric positive definite functions. For example, this class is a closed convex cone, i.e., it is stable by addition, multiplication by a positive constant, and pointwise limit. Some kernels, known as infinitely divisible, can be taken to an arbitrary non-negative exponent and remain positive definite [Hau99, SS02].

As an example, observe that if $K_1$ is a string kernel for protein sequences, and $K_2$ is a Gaussian RBF kernel for the expressions of the corresponding genes, then $K_1 + K_2$ is a kernel that incorporates both sequence and expression similarity. Hence the addition of two kernels is a simple yet powerful way to combine two types of informations, which has been for example successfully used in the context of gene function prediction [PWCG01]. More generally, if $K_1, \ldots, K_p$ is a family of kernels on the same space, then a valid Mercer kernel obtained as a linear combination of the basic kernel which optimizes some linear function can be found by semi-definite programming, which has been explored for instance in [LCG+02]. Alternatively, one can use information geometric properties of the set of positive definite matrix to generate new kernels from basic kernels [KT].

## 5 Conclusion

In this short overview of recent advances in kernel methods and their applications in computational biology, we tried to present kernel methods as a relevant framework to manipulate biological data and perform computations and inference from them. This has been a very active and exciting field or research in the last few years, as it turns out that a wide range of kernels can be engineered and adapted to particular data, and that the machinery of kernel methods provide a large family of algorithms to extract information from kernel representations. Moreover, mathematical operations on kernels provide a interesting starting point to develop more satisfactory frameworks for modeling living systems. As an example, we mentioned the possibility we intend to study in the future to represent a biological system as a probability distribution on the set of Mercer kernels, in such a way that the Mercer kernels built from various sources of information would be different realizations of a more abstract random kernels. Such developments would certainly require more research or more integration of theoretical results about positive definite kernels, and might lead to new families of algorithms to fit the need of post-genomics.

## 6 Acknowledgments

I am grateful to Kenji Ueno and Tsuyoshi Kato who offered me the opportunity to participate to the workshop "Mathematical aspects of molecular biology: toward new mathematics" that was held in Nara, Japan, on January 24-27, 2003. This text is based on the talk I gave there.

## References

[AN01] Shunichi Amari and Hiroshi Nagaoka. *Methods of information geometry.* AMS vol. 191, 2001.

[AVC03] John Shawe-Taylor Alexei Vinokourov and Nello Cristianini. Inferring a semantic representation of text via cross-language correlation analysis. In Suzanna Becker, Sebastian Thrun, and Klaus Obermayer, editors, *Advances in Neural Information Processing Systems*. MIT Press, 2003.

[BGL+00] Michael P. S. Brown, William Noble Grundy, David Lin, Nello Cristianini, Charles Walsh Sugnet, Terence S. Furey, Jr. Manuel Ares, and David Haussler. Knowledge-based analysis of microarray gene expression data by using support vector machines. *Proc. Natl. Acad. Sci. USA*, 97:262–267, 2000.

[BGV92] B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the 5th annual ACM workshop on Computational Learning Theory*, pages 144–152. ACM Press, 1992.

[BJ02] Francis R. Bach and Michael I. Jordan. Kernel independent component analysis. *Journal of Machine Learning Research*, 3:1–48, 2002.

[Chu97] Fan R.K. Chung. *Spectral graph theory*, volume 92 of *CBMS Regional Conference Series*. American Mathematical Society, Providence, 1997.

[DEKM98] Richard Durbin, Sean Eddy, Anders Krogh, and Graeme Mitchison. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, 1998.

[FDC+00] Terrence S. Furey, Nigel Duffy, Nello Cristianini, David Bednarski, Michel Schummer, and David Haussler. Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics*, 16(10):906–914, 2000.

[GWBV02] Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1/3):389–422, Jan 2002.

[Hau99] David Haussler. Convolution kernels on discrete structures. Technical report, UC Santa Cruz, 1999.

[Hot36] H. Hotelling. Relation between two sets of variates. *Biometrika*, 28:322–377, 1936.

[HS01a] Sujun Hua and Zhirong Sun. A novel method of protein secondary structure prediction with high segment overlap measure: Support vector machine approach. *Journal of Molecular Biology*, 308(2):397–407, April 2001.

[HS01b] Sujun Hua and Zhirong Sun. Support vector machine approach for protein subcellular localization prediction. *Bioinformatics*, 17(8):721–728, 2001.

[JDH00] Tommi Jaakkola, Mark Diekhans, and David Haussler. A discriminative framework for detecting remote protein homologies. *Journal of Computational Biology*, 7(1,2):95–114, 2000.

[JH99] Tommi S. Jaakkola and David Haussler. Exploiting generative models in discriminative classifiers. In *Proc. of Tenth Conference on Advances in Neural Information Processing Systems*, 1999.

[Jol96] I.T. Jolliffe. *Principal component analysis*. Springer-Verlag, New-York, 1996.

[KL02] R. I. Kondor and J. Lafferty. Diffusion kernels on graphs and other discrete input. In *ICML 2002*, 2002.

[KT] K. Asai K. Tsuda, S. Akaho. Approximating incomplete kernel matrices by the EM algorithm. Poster presented at the 6th kernel machines workshop, 2002.

[KW71] G.S. Kimeldorf and G. Wahba. Some results on tchebycheffian spline functions. *Journal of Mathematical Analysis and Applications*, 33:82–95, 1971.

[LCG+02] Gert R.G. Lanckriet, Nello Cristianini, Laurent El Ghaoui, Peter Bartlett, and Michael I. Jordan. Learning the kernel matrix with semidefinite programming. Technical report, UC Berkeley, 2002.

[LEN02] Christina Leslie, Eleazar Eskin, and William Stafford Noble. The spectrum kernel: a string kernel for svm protein classification. In Russ B. Altman, A. Keith Dunker, Lawrence Hunter, Kevin Lauerdale, and Teri E. Klein, editors, *Proceedings of the Pacific Symposium on Biocomputing 2002*, pages 564–575. World Scientific, 2002.

[LEWN03] Christina Leslie, Eleazar Eskin, Jason Weston, and William Stafford Noble. Mismatch string kernels for svm protein classification. In Suzanna Becker, Sebastian Thrun, and Klaus Obermayer, editors, *Advances in Neural Information Processing Systems 15*. MIT Press, 2003.

[MTM+98] S. Mukherjee, P. Tamayo, J.P. Mesirov, D. Slonim, A. Verri, and T. Poggio. Support vector machine classification of microarray data. Technical Report 182, C.B.L.C., 1998. A.I. Memo 1677.

[PMT+99] Matteo Pellegrini, Edward M. Marcotte, Michael J. Thompson, David Eisenberg, and Todd O. Yeates. Assigning protein functions by comparative genome analysis: Protein phylogenetic profiles. *Proc. Natl. Acad. Sci. USA*, 96:4285–4288, April 1999.

[PWCG01] Paul Pavlidis, Jason Weston, Jinsong Cai, and William Noble Grundy. Gene functional classification from heterogeneous data. In *Proceedings of the Fifth Annual International Conference on Computational Biology*, pages 249–255, 2001.

[SS02] Bernhard Schölkopf and Alexander J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, 2002.

[SSM99] Bernhard Schölkopf, Alexander J. Smola, and Klaus-Robert Müller. Kernel principal component analysis. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*, pages 327–352. MIT Press, 1999.

[SVUA03] Hiroto Saigo, Jean-Philippe Vert, Nobuhisa Ueda, and Tatsuya Akutsu. Protein homology detection using string alignment kernels. To appear, 2003.

[Vap98] Vladimir N. Vapnik. *Statistical Learning Theory*. Wiley, New-York, 1998.

[Ver02a] Jean-Philippe Vert. Support vector machine prediction of signal peptide cleavage site using a new class of kernels for strings. In Russ B. Altman, A. Keith Dunker, Lawrence Hunter, Kevin Lauerdale, and Teri E. Klein, editors, *Proceedings of the Pacific Symposium on Biocomputing 2002*, pages 649–660. World Scientific, 2002.

[Ver02b] Jean-Phlippe Vert. A tree kernel to analyze phylogenetic profiles. *Bioinformatics*, 18:S276–S284, 2002.

[VK] Jean-Philippe Vert and Minoru Kanehisa. Graph-driven features extraction from microarray data. Preprint arXiv physics/0206055, June 2002.

[VK03] Jean-Philippe Vert and Minoru Kanehisa. Graph-driven features extraction from microarray data using diffusion kernels and kernel cca. In Suzanna Becker, Sebastian Thrun, and Klaus Obermayer, editors, *Advances in Neural Information Processing Systems*. MIT Press, 2003.

[Wah90] G. Wahba. *Spline Models for Observational Data*, volume 59 of *CBMS-NSF Regional Conference Series in Applied Mathematics*. SIAM, Philadelphia, 1990.

[Wat00] C. Watkins. Dynamic alignment kernels. In A.J. Smola, P.L. Bartlett, B. Schölkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*, pages 39–50. MIT Press, Cambridge, MA, 2000.

[YVNK03] Y. Yamanishi, J.-P. Vert, A. Nakaya, and M. Kanehisa. Extraction of correlated gene clusters from multiple genomic data by generalized kernel canonical correlation analysis. *Bioinformatics*, 2003. To appear.