Graph-driven features extraction from microarray data

Jean-Philippe Vert and Minoru Kanehisa
Bioinformatics Center
Institute for Chemical Research
Kyoto University
Uji, Kyoto 611-0011, Japan
Jean-Philippe.Vert@mines.org
kanehisa@kuicr.kyoto-u.ac.jp

June 15, 2002

Abstract

Gene function prediction from microarray data is a first step toward better understanding the machinery of the cell from relatively cheap and easy-to-produce data. In this paper we investigate whether the knowledge of many metabolic pathways and their catalyzing enzymes accumulated over the years can help improve the performance of classifiers for this problem.

The complex network of known biochemical reactions in the cell results in a representation where genes are nodes of a graph. Formulating the problem as a graph-driven features extraction problem, based on the simple idea that relevant features are likely to exhibit correlation with respect to the topology of the graph, we end up with an algorithm which involves encoding the network and the set of expression profiles into kernel functions, and performing a regularized form of canonical correlation analysis in the corresponding reproducible kernel Hilbert spaces.

Function prediction experiments for the genes of the yeast *S. Cerevisiae* validate this approach by showing a consistent increase in performance when a state-of-the-art classifier uses the vector of features instead of the original expression profile to predict the functional class of a gene.

Keywords: microarray, gene expression, network, pathway, diffusion kernel, kernel CCA, feature extraction, function prediction.

1 Introduction

Following the near completion of many genome sequencing projects and the identification of genes coding for proteins in these genomes, the research paradigm is shifting toward a better understanding of the functions of the genes and their interactions. This discipline, broadly called *functional genomics* is expected to provide new insights into the machinery of the cell and suggest new therapeutic targets by better focusing on the precise molecules or processes responsible for a given disease.

Functional genomics has been boosted since the mid 1990's by the introduction of the DNA microarray technology [SSDB95, BB00], which enables the monitoring of the quantity of messenger RNA (mRNA) present in a cell for several thousands genes simultaneously, at a given instant. As mRNA is the intermediate molecule between the blueprint of a protein on the DNA strand and the protein itself, it is expected that the quantity of mRNA reflects the quantity of the protein itself, and that variations in the quantity of mRNA when a cell is confronted to various experimental conditions reflects the genetic regulation process. Consequently functional characterization of a protein from its expression profile as measured by several microarray hybridation experiments is supposed to be possible to some extent, and initial experiments confirmed that many genes with similar function yield similar expression patterns [ESBB98]. As data accumulate the incentive to develop precise methods to assign functions to genes from expression profiles increases.

Proteins can have many structural or functional roles. In particular proteins known as enzymes catalyze chemical reactions which enable cells to acquire energy and materials from its environment, and to utilize them to maintain their own biochemical network. Decades of careful experiments have helped characterize many reactions taking place in the cell together with some of the genes playing a role in their control, and this information has now been integrated into several databases including WIT [OLP$^+$00] or KEGG [KGKN02]. Such databases provide a view of the set of proteins as the nodes of a large and complex network, where two genes are linked when they catalyze two successive reactions.

The question motivating this paper is whether this network can help improve the performance of function prediction algorithms based on microarray data only. To this end we propose a graph-driven feature extraction process from the expression profiles, based on the idea that patterns of expression which correspond to actual biological events, such as the activation of a series of chemical reactions forming a *chemical pathway*, are likely to be shared by genes close to each other with respect to the network topology. Translating this idea mathematically we end up with a features extraction process equivalent to performing a generalization of canonical correlation analysis (CCA) between the representations of the genes in two different reproducing kernel Hilbert spaces, defined respectively by a diffusion kernel [KL02] on the gene graph and by a linear kernel on the expression profiles. The CCA can be performed in these RKHS using the kernel-CCA algorithm presented in [BJ01].

Relationships between expression profiles and biochemical pathways have been subject to much investigation in the recent years. As microarray data are much cheaper to produce than precise pathway data, pathway reconstruction or validation from expression data has been attracting much attention since the availability of public microarray data [FLNP00, AMK00]. Extraction of co-clusters, i.e., clusters of genes in the network which have similar expression has also been investigated recently [NGK01, HZZL02]. On the technical point of view the integration of several sources of data has been investigated with different approaches, e.g., combining expression data and genomic location information in a Bayesian framework [HGJY02], combining expression data with phylogenetic profiles by kernel operations [PWCG01], or defining distances between genes by combining distances measured from different data types [MPT$^+$99].

This paper is organized as follows. Section 2 translates mathematically the feature extraction problem and contains basic notations and definitions, followed by a short review of some properties of RKHS relevant for our purpose in Section 3. Sections 4 and 5 describe respectively how two important properties of features can be expressed in terms of norms in RKHS, and Section 6 describes the feature extraction process. Experimental results are presented in Section 7, followed by a discussion in Section 8.

2 Problem definition

2.1 Setting and notations

Before focusing on expression profiles and biochemical pathways, we first formulate in a more abstract way the problem we are dealing with. The set of genes is represented by a finite set $\mathcal{X}$ of cardinality $|\mathcal{X}| = n$, where each element $x \in \mathcal{X}$ represents a gene. The information provided by the microarray experiments and the pathway database are represented respectively as:

- a mapping $e : \mathcal{X} \rightarrow \mathbb{R}^p$, where $e(x)$ is the expression profile for the gene x, for any x in $\mathcal{X}$, and p is the number of measurements available. In the sequel we assume that the profiles have been centered, i.e.:

$$\sum_{x \in \mathcal{X}} e(x) = 0. \tag{1}$$

- A simple graph $\Gamma = (\mathcal{X}, \mathcal{E})$ (without loops and multiple edges) whose vertices are the genes $\mathcal{X}$ and whose edges $\mathcal{E}$ represent the links between genes, as extracted from the biochemical pathway database.

The notation $x \sim y$ for any $(x, y) \in \mathcal{X}^2$ means that there is an edge between x and y, i.e., $\{x, y\} \in \mathcal{E}$. Our goal in the sequel is to use the graph Γ in order to extract features from the expression profiles e relevant for the functional classification of the genes. In this context we formally define a feature to be a mapping $f : \mathcal{X} \rightarrow \mathbb{R}$, and we denote by $\mathcal{F} = \mathbb{R}^{\mathcal{X}}$ the set of possible features. The set of centered features is denoted by $\mathcal{F}_0 = \left\{ f \in \mathcal{F} : \sum_{x \in \mathcal{X}} f(x) = 0 \right\}$. For any feature $f \in \mathcal{F}$ the same notation is used to represent the n-dimensional vector $f = (f(x))_{x \in \mathcal{X}}$ indexed by the elements of $\mathcal{X}$, and f' denotes its transpose. The constant unit vector is denoted $\mathbf{1} = (1, \ldots, 1)$.

2.2 Feature relevance

Features can be derived from the mapping e. As an example, projecting e to a given direction $v \in \mathbb{R}^p$ gives the feature $f_{e,v}$ defined for any x in $\mathcal{X}$ by:

$$f_{e,v}(x) = v'e(x). \tag{2}$$

If v represents a particular expression pattern, then $f_{e,v}$ quantifies how each gene correlates with this pattern. In this paper we restrict ourselves to such linear features, and denote by $\mathcal{G} = \{f_{e,v}, v \in \mathbb{R}^p\} \subset \mathcal{F}$ the set of linear features. Observe that by hypothesis (1), each linear feature is also centered by (2), i.e., $\mathcal{G} \subset \mathcal{F}_0$.

Biological events such as synthesis of new molecules or transport of substrates usually require the coordinated actions of many proteins. Genes encoding such proteins are therefore likely to share particular patterns of expression over different experimental conditions, e.g. simultaneous overexpression or inhibition. A vector $v \in \mathbb{R}^p$ representing this pattern should therefore be particularly correlated (positively or negatively) with the genes participating in the biological process. As a result, linear features $f_{e,v}$ corresponding to biologically relevant patterns $v \in \mathbb{R}^d$ are more likely to have a larger variance than those corresponding to patterns unrelated to any biological event, where the variance is defined by:

$$\forall f_{e,v} \in \mathcal{G}, \quad V(f_{e,v}) = \frac{\sum_{x \in \mathcal{X}} f_{e,v}(x)^2}{||v||^2}. \tag{3}$$

On the other extreme a pattern $v \in \mathbb{R}^p$ orthogonal to all profiles leads to a feature $f_{e,v}$ with null variance, and is clearly unlikely to be related to any biological process requiring gene expression. It follows that the variance (3) captured by a feature is a first indicator of its biological pertinence. In order to prevent confusion with other criteria in the sequel, we will call a feature *relevant* if it captures much variations between expression profiles in the sense of (3), and *irrelevant* otherwise. The reader can observe that searching for the most relevant features can be done by performing a principal component analysis (PCA) [Jol96] of the profiles, the first principal components corresponding to the most relevant features; however we now show that relevance is not the only criterion which can be used to select features.

2.3 Feature smoothness

Relevance as defined in Section 2.2 is an intrinsic properties of the set of profiles, as it is defined in terms of variation captured, and no other information about the relationships between genes is used.

Independently of any microarray experiment, many metabolic pathways have been experimentally characterized over the years. These collections of chemical reactions involve proteins as enzymes, whose presence or absence plays a major role in monitoring the reaction. Actual biological event usually involve series of such reactions, also called *pathways*. Genes involved in consecutive reactions of pathways are likely to share particular patterns of expression, corresponding to the activation or not of the corresponding pathway.

As a result a pattern $v \in \mathbb{R}^p$ which corresponds to a true biological event, such as the activation or inhibition of a pathway, is likely to be shared by clusters of genes in the graph of genes where two genes are linked if they participate in consecutive reactions. On a more global scale, such a feature is more likely to vary smoothly on the graph of genes, in the sense that variations between linked genes be as small as possible, than a noisy pattern unrelated to any biochemical event which would not exhibit any particular correlation between genes linked to each other in the graph.

Such features are called *smooth* in the sequel, by opposition to *rugged* features which vary a lot with respect to the graph topology. These notions are formalized and quantified in terms of a norm in a Hilbert space in Section 4, but before developing these technicalities we can already sketch a feature extraction process based on this intuitive definition.

2.4 Problem formulation

From the discussions in Sections 2.2 and 2.3 two criteria appear to characterize "good" candidate features : their relevance on the one hand (Section 2.2) based on a statistical analysis of the set of profiles, and their smoothness on the other hand (Section 2.3) which results from the analysis of the variations of the feature with respect to the topology of the graph of genes.

Good candidate features are smooth and relevant in the same time. These two properties are however not always correlated: it might be possible to find many relevant but rugged features, as well as smooth but irrelevant features. A reasonable approach to extract meaningful features is therefore to try to find a compromise between these two criteria, and to extract features which are as smooth and relevant in the same time as possible.

Although this statement can be translated mathematically in many different ways, we investigate in the sequel the following formulation:

Problem 1 *Extract pairs of features* $(f_1, f_2) \in \mathcal{F}_0 \times \mathcal{G}$ *such that:*

- f_1 *be smooth,*
- f_2 *be relevant,*
- f_1 *and* f_2 *be correlated.*

These three goals are usually contradictory and a trade-off must be found between them. Observe that if either the smoothness or the relevance conditions are removed, the problem is likely to be ill-posed. For instance, if the smoothness requirement is removed then any relevant feature f_2 is perfectly correlated with itself; on the other hand if the relevance conditions disappears then many smooth features f_1 can probably be correlated with linear features which are not necessarily relevant (this possibility increases when the dimension p of the profiles increases, as the set of linear features increases too).

Let us now formulate Problem 1 mathematically. The correlation between any two centered features $(f_1, f_2) \in \mathcal{F}_0^2$ is equal to:

$$c(f_1, f_2) = \frac{f_1' f_2}{\sqrt{f_1' f_1} \sqrt{f_2' f_2}}. \tag{4}$$

As already mentioned the maximization of $c(f_1, f_2)$ over $\mathcal{F}_0 \times \mathcal{G}$ is an ill-posed problem.

Suppose we can define a smoothness functional $h_1 : \mathcal{F} \rightarrow \mathbb{R}^+$ for any feature, and a relevance functional $h_2 : \mathcal{G} \rightarrow \mathbb{R}^+$ for linear features, in such a way that lower values of the functional h_1 (resp. h_2) corresponds to smoother (resp. more relevant) features. Then one way to formalize the trade-off between correlation and relevance / smoothness is to solve the following maximization problem:

$$\max_{(f_1, f_2) \in \mathcal{F}_0 \times \mathcal{G}} \frac{f_1' f_2}{\sqrt{f_1' f_1 + \delta h_1(f_1)} \sqrt{f_2' f_2 + \delta h_2(f_2)}}, \tag{5}$$

where δ is a regularization parameter. When $\delta = 0$ we recover the ill-posed problem of maximizing the correlation (4), and the larger δ the smoother (resp. the more relevant) the feature f_1 (resp. f_2) which solves (5). As a result, a solution (f_1, f_2) of (5) is a reasonable solution to Problem 1, with δ controlling the trade-off between correlation on the one hand, smoothness and relevance on the other hand.

Equation (5) is therefore the problem we consider is the sequel. In order to solve it we need to 1) express the relevance and smoothness functional h_1 and h_2 mathematically and 2) solve the maximization problem (5) with these functionals. These two steps are not independent. In particular there is an incentive to express mathematically h_1 and h_2 in such a way that (5) be computationally solvable.

If f_1 and f_2 were restricted to be linear functionals obtained by projecting two different vector representations of the genes on particular directions, then the maximization of (4) would be the exactly the first *canonical correlation* between f_1 and f_2 [Hot36], as obtained by classical canonical correlation analysis (CCA). Linear algebra algorithms involving eigenvector decomposition exist to perform CCA. However f_1 is not restricted to be a linear feature, and (4) is consequently ill-posed.

Formulated as (5), however, we recover a slight generalization of CCA introduced in [BJ01] and called *kernel-CCA*. More precisely, kernel-CCA is formulated as:

$$\max_{(f_1, f_2) \in \mathcal{H}_1 \times \mathcal{H}_2} \frac{f_1' f_2}{\sqrt{f_1' f_1 + \delta ||f_1||_{\mathcal{H}_1}} \sqrt{f_2' f_2 + \delta ||f_2||_{\mathcal{H}_2}}}, \tag{6}$$

where $\mathcal{H}_1$ and $\mathcal{H}_2$ are two reproducible kernel Hilbert spaces (see Section 3) on the space $\mathcal{X}$. Problem (6) is equivalent to a generalized eigenvalue problem

[BJ01] and can be solved iteratively to extract several pairs of features (see Section 6.2).

In order to use the algorithm of [BJ01] we therefore need to restate (5) in terms of optimization in RKHS like (6). This involves 1) expressing $\mathcal{F}_0$ as a RKHS whose norm is a smoothness functional (Section 2.3), 2) expressing $\mathcal{G}$ as a RKHS whose norm is a relevance functional (Section 5), and 3) solving the resulting problem (6).

3 Reproducing kernel Hilbert space

Before carrying out the program sketched in Section 2.4 we first recall some definitions and basic properties of RKHS in order to make this paper as self-contained as possible. Good introductions on RKHS can be found in [Aro50, Sai88, Wah90, SS02] from which we borrow most of the materials presented in this section.

3.1 Basic definitions

Let $\mathcal{X}$ be a set (which we don't necessarily assume to be finite in this section), and $K : \mathcal{X} \to \mathbb{R}$ a symmetric positive definite function, in the sense that for every $l \in \mathbb{N}$ and $(x_1, \ldots, x_l) \in \mathcal{X}^l$ the $l \times l$ Gram matrix $K_{i,j} = K(x_i, x_j)$ be positive semidefinite.

Then it is known that the linear span of set of functions $\{K(.,x), x \in \mathcal{X}\} \subset \mathbb{R}^{\mathcal{X}}$ can be completed into a Hilbert space $\mathcal{H} \subset \mathbb{R}^{\mathcal{X}}$ which satisfies the following "reproducing property":

$$\forall (f,x) \in \mathcal{H} \times \mathcal{X}, \quad f(x) = \langle K(.,x), f \rangle_{\mathcal{H}}, \tag{7}$$

where $< .,. >_{\mathcal{H}}$ represents the inner product of $\mathcal{H}$. In particular, by plugging $f = K(.,x')$ in (7) we obtain:

$$\forall (x,x') \in \mathcal{X}^2, \quad \langle K(.,x), K(.,x') \rangle_{\mathcal{H}} = K(x,x'). \tag{8}$$

The Hilbert space $\mathcal{H}$ is called a *reproducing kernel Hilbert space* [Aro50] to emphasize the property (7). In order to make this rather abstract result clearer, let us show how the space $\mathcal{H}$ can be built when $\mathcal{X}$ is finite, which is the case of interest in this paper.

Let us therefore take $\mathcal{X}$ to be the finite set of genes, and suppose first that the $n \times n$ Gram matrix $K_{x,y} = K(x,y)$ for any $(x,y) \in \mathcal{X}^2$ is *positive definite*, i.e., that its eigenvalues are all positive. It can then be diagonalized as follows:

$$K = \sum_{i=1}^{n} \lambda_i \phi_i \phi_i', \tag{9}$$

where the eigenvalues satisfy $0 < \lambda_1 \leq \ldots \leq \lambda_n$ and the set $(\phi_1, \ldots, \phi_n) \in \mathcal{F}^n$ is an associated orthonormal basis of eigenvectors.

We can now take the Hilbert space to be $\mathcal{H} = \mathcal{F}$, and define the inner product in $\mathcal{H}$ in terms of the decomposition of any $f \in \mathcal{H}$ in the basis of eigenvectors:

$$f = \sum_{i=1}^{n} a_i \phi_i, \tag{10}$$

as follows:

$$\left\langle \sum_{i=1}^{n} a_i \phi_i, \sum_{i=1}^{n} b_i \phi_i \right\rangle_{\mathcal{H}} = \sum_{i=1}^{n} \frac{a_i b_i}{\lambda_i}. \tag{11}$$

It is easy to check that the Hilbert space defined by (11) satisfies the reproducing property (7), and is therefore a RKHS associated with the kernel $K(.,.)$.

The columns of the Gram matrix being independent, any feature $f \in \mathcal{H}$ can be uniquely represented as follows:

$$f(.) = \sum_{x \in \mathcal{X}} \alpha(x) K(x,.), \tag{12}$$

or in an equivalent matrix form:

$$f = K\alpha. \tag{13}$$

This representation is called the *dual* representation of f, and the vector $\alpha = (\alpha(x))_{x \in \mathcal{X}} \in \mathcal{F}$ is called the *dual coordinate* of f.

The dual representation is useful to express the inner product in the Hilbert space $\mathcal{H}$. Indeed, using (12) and (8) it is easy to check that the inner product between two features $(f,g) \in \mathcal{F}^2$ with dual coordinates $(\alpha, \beta) \in \mathcal{F}^2$ respectively is given by:

$$\langle f, g \rangle_{\mathcal{H}} = \sum_{(x,y) \in \mathcal{X}^2} \alpha(x)\beta(y)K(x,y) = \alpha' K \beta.$$

In particular the $\mathcal{H}$-norm of a feature $f \in \mathcal{F}$ with dual coordinates $\alpha \in \mathcal{F}$ is given by:

$$||f||_{\mathcal{H}}^2 = \alpha' K \alpha. \tag{14}$$

The inner product in the original space $L^2(\mathcal{X})$ can also simply be expressed with the dual representation: for any $(f,g) \in \mathcal{F}^2$ with dual coordinates (α, β) respectively we have by (13) and using the fact that K is symmetric:

$$f'g = \sum_{x \in \mathcal{X}} f(x)g(x) = \alpha' K^2 \beta.$$

In case the kernel K is just positive semidefinite, with r being the multiplicity of 0 as eigenvalue, then we can follow the same construction with the index i ranging from $r+1$ to n in (9), (10) and (11). In that case the RKHS $\mathcal{H}$ is the linear span of $\{\phi_{r+1}, \ldots, \phi_n\}$, of dimension $n-r$. The dual representation still makes sense but is defined up to an element of $\{\alpha \in \mathbb{R}^{\mathcal{X}}, K\alpha = 0\}$.

3.2 RKHS and smoothness functional

One classical application of the theory of RKHS is regularization theory to solve ill-posed problems [TA77, Iva76, Wah90, GJP95]. Indeed it is well known that for many choices of kernels $K(.,.)$ on continuous spaces $\mathcal{X} \subset \mathbb{R}^N$ the norm in the corresponding RKHS $||f||_{\mathcal{H}}$ is intimately related to the smoothness properties of the functions $f \in \mathcal{H}$.

The following classical example is relevant for us. Consider a set $\mathcal{X} \subset \mathbb{R}^N$ and a translation-invariant kernel of the form $K(x,y) = k(x-y)$ for any $(x,y) \in \mathcal{X}^2$. Then the RKHS $\mathcal{H}$ is composed of the functions $f \in L^2(\mathcal{X})$ such that:

$$||f||_{\mathcal{H}} = \int_{\mathbb{R}^N} \frac{|\hat{f}(\omega)|^2}{\nu(\omega)} d\omega < \infty, \tag{15}$$

where $\hat{f}(\omega)$ is the Fourier transform of f and $\nu(\omega)$ is the Fourier transform of $k(.)$ [GJP95, SSM98]. Functionals of the form (15) are known to be smoothness functionals (in which case smoothness is defined in terms of Fourier transform, i.e., smooth functions are functions with few energy at high frequency), where the rate of decrease to zero of ν controls the smoothness properties of the function in the RKHS. For example, for the Gaussian radial basis function $k(x-y) = \exp(-||x-y||^2/2\sigma^2)$ the norm in the RKHS takes the form:

$$||f||_{\mathcal{H}} = \left(2\pi\sigma^2\right)^{-\frac{p}{2}} \int_{\mathbb{R}^N} e^{\frac{\sigma^2}{2}||\omega||^2} |\hat{f}(\omega)|^2 d\omega. \tag{16}$$

Equation (16) shows that the energy of f at a frequency ω should decrease at least as $\exp(-\sigma^2||\omega||^2/2)$ for its $\mathcal{H}$-norm to be finite. Functions with much energy at high-frequency have a large norm in $\mathcal{H}$, which therefore acts as a smoothness functional.

We refer the reader to [TA77, Iva76, Wah90, GJP95] for more details on the connections between RKHS and smoothness functionals, as well as for applications to solve ill-posed problems. In the sequel we will adapt these approaches to discrete spaces $\mathcal{X}$ in order to fulfill the program sketched in Section 2.4

4 Smoothness functional on a graph

As pointed out in Sections 2.4 our interest is now to derive a "smoothness functional" for features $f \in \mathcal{F}$ with respect to the graph Γ expressed as a norm in a RKHS.

4.1 Fourier transform on graphs

Equation (15) shows that the norm in a RKHS on a continuous space associated with a translation-invariant kernel is defined in terms of Fourier transform. A natural approach to adapt the construction of smoothing functional to functions defined on a graph is therefore to adapt the Fourier transform to that context.

As a matter of fact Fourier transforms on graphs have been extensively studied in spectral graph theory [Chu97, Moh91, Moh97, Sta96], as we now recall.

Let D be the $n \times n$ diagonal matrix of vertex degrees of the graph Γ, i.e.,

$$\forall (x,y) \in \mathcal{X}^2, \quad D_{x,y} = \begin{cases} 0 & \text{if } x \neq y, \\ deg(x) & \text{if } x = y, \end{cases}$$

where $deg(x)$ is the number of edges involving x in Γ, and let A be the adjacency matrix defined by:

$$\forall (x,y) \in \mathcal{X}^2, \quad A_{x,y} = \begin{cases} 1 & \text{if there is an edge between } x \text{ and } y \text{ in } \Gamma, \\ 0 & \text{otherwise .} \end{cases}$$

Then the $n \times n$ matrix:

$$L = D - A$$

is called the (discrete) *Laplacian* of Γ. The discrete Laplacian L is a central concept in spectral graph analysis [Moh97]. It shares many important properties with the familiar differential operator

$$-\Delta(.) = div(grad(.))$$

on Riemannian manifolds. It is symmetric, semidefinite positive, and singular. The eigenvector $(1, \ldots, 1)$ belongs to the eigenvalue $\lambda_1 = 0$, whose multiplicity is equal to the number of connected components of Γ.

Let us denote by

$$0 = \lambda_1 \leq \ldots \leq \lambda_n$$

the eigenvalues of L and $\{\phi_i, i = 1, \ldots, n\}$ an orthonormal set of associated eigenvectors. Just like the Fourier basis functions are eigenfunctions of the continuous Laplacian on $\mathbb{R}^N$, the eigenvectors of L can be regarded as a discrete Fourier basis on the graph Γ [Sta96], with frequency increasing with their eigenvalues.

Although the term “frequency” is not well defined for functionals on a graph, the reader can get an intuition of the fact that the functions $(\phi_i, i = 1, \ldots, n)$ “oscillates” more and more on the graph as i increases through the following two well-known results:

- Applying the classical equality [Moh97]:

  $$\forall f \in \mathcal{F}, \quad f' L f = \sum_{x \sim y} \left(f(x) - f(y)\right)^2,$$

  to an eigenfunction ϕ of L with eigenvalue λ gives the following equality:

  $$\sum_{x \sim y} \left(\phi(x) - \phi(y)\right)^2 = \lambda. \qquad (17)$$

  Equation (17) confirms that the larger λ, the more the associated eigenfunction varies between adjacent vertices of the graph.

- An other classical result concerns the number of maximal connected components of the graph where a feature has a constant sign. The first eigenfunction being constant, it has only one such component, namely the whole graph. For the other eigenfunctions, the discrete nodal domain theorem which translate Courant's famous nodal theorem for elliptic operators on Riemannian manifolds [Cha84] to the discrete settings [dV93, Fri93, vdH96, DGL$^+$01] states that the number of maximal connected subsets of $\mathcal{X}$ where ϕ_i does not change sign is equal to i in the case where all eigenvalues have multiplicity 1 (see a more general statement in [DGL$^+$01]). Together with the fact that each eigenfunction ϕ_i for $i > 1$ has zero mean (because it is orthogonal to the constant function ϕ_1) this shows that ϕ_i "oscillates" more and more on the graph, in the sense that it changes sign more and more often as i increases.

By similarity with the continuous case the basis $(\phi_i)_{i=1,\ldots,n}$ is called a *Fourier basis*, higher eigenvalues corresponding to higher frequencies. Any feature $f \in \mathcal{F}$ can be expanded in terms of this basis:

$$f = \sum_{i=1}^{n} \hat{f}_i \phi_i, \tag{18}$$

where $\hat{f}_i = \phi_i' f$ and $\hat{f} = \left(\hat{f}_1, \ldots, \hat{f}_n\right)$ is called the *discrete Fourier transform* of f. This provides a way to analyze features in the frequency domain, and in particular to measure their smoothness as we now show.

4.2 Graph smoothness functional

The Laplacian matrix L is semidefinite positive and can therefore be used as a Kernel Gram matrix. The multiplicity of 0 as eigenvalue is the number of connected components of the graph, and the associated eigenvectors are the functions constant on each connected components. Following Section 3 the associated RKHS $\mathcal{H}$ has dimension $n - r$ and is made of the set of features with zero mean on each connected component. By (11) the norm of any function $f \in \mathcal{H}$ is given by:

$$||f||_{\mathcal{H}}^2 = \sum_{i=r+1}^{m} \frac{\hat{f}_i^2}{\lambda_i}, \tag{19}$$

where $\hat{f}$ is the Fourier transform of f (18) and λ is the ordered set of eigenvalues of L.

However, as shown in Section 4.1, the smoothness of ϕ_i decreases with i; because λ_i increases with i, the norm (19) in the RKHS associated with the kernel L increases with smoothness, and is therefore a "ruggedness functional" instead of a smoothness functional in the sense defined in Section 3. To illustrate this we can observe that:

$$\forall i \in \{r+1, \ldots, n\}, \quad ||\phi_i||_{\mathcal{H}} = \frac{1}{\sqrt{\lambda_i}},$$

hence $||\phi_i||_{\mathcal{H}}$ decreases with i.

Transforming this ruggedness functional into a smoothness functional can be performed by a simple operation on the kernel as follows:

Definition 1 *For any decreasing mapping $\zeta : \mathbb{R}^+ \to \mathbb{R}^+ \backslash \{0\}$, we define the ζ-kernel $K_\zeta : \mathcal{X}^2 \to \mathbb{R}$ by:*

$$\forall (x,y) \in \mathcal{X}^2, \qquad K_\zeta(x,y) = \sum_{i=1}^n \zeta(\lambda_i)\phi_i(x)\phi_i(y),$$

where $0 = \lambda_1 \leq \ldots \leq \lambda_n$ are the eigenvalues of the graph Laplacian and $(\phi_1, \ldots, \phi_n)$ an associated orthonormal Fourier basis.

The mapping ζ being assumed to take only positive values, the matrix K_ζ is definite positive and is therefore a valid kernel, with associated RKHS $\mathcal{H} = \mathcal{F}$. From the discussion above it is now clear that:

Proposition 1 *The norm $||.||_\zeta$ in the RKHS associated with the kernel K_ζ is a smoothing functional, given for any feature $f \in \mathcal{F}$ with Fourier transform $\hat{f} \in \mathbb{R}^n$ by:*

$$||f||_\zeta^2 = \sum_{i=1}^n \frac{\hat{f}_i^2}{\zeta(\lambda_i)}. \tag{20}$$

Proof Equation (20) is a direct consequence of Definition 1 and (11). The fact that $||.||_\zeta$ is a smoothing functional is simply a translation of the fact that $\zeta(\lambda_i)$ decreases with i, hence the relative contribution of the Fourier components in (20) increases with their frequency.

Proposition 1 shows that the smoothness functional associated with a function ζ is controlled by its rate of decrease to 0. An example of valid ζ function with rapid decay is the following:

$$\forall x \in \mathbb{R}^+, \quad \zeta(x) = e^{-\tau x}, \tag{21}$$

where τ is a parameter. In that case we recover the *diffusion kernel* introduced and discussed in [KL02]. The authors of this paper show that the diffusion kernel shares many properties with the continuous Gaussian kernel $K(x,y) = \exp(-||x-y||^2/2\sigma^2)$ on $\mathbb{R}^p$, and can therefore be considered as its discrete version.

Combining (20) and (21) we obtain that the norm in the RKHS associated with the diffusion kernel is given by:

$$\forall f \in \mathcal{F}, \qquad ||f||_\zeta = \sum_{i=1}^n e^{\tau \lambda_i} \hat{f}_i^2, \tag{22}$$

hence the high frequency energy of f is strongly penalized by this kernel, and the penalization increases with the parameter τ.

Before continuing we should observe that in concrete applications the computation of the kernel K_ζ for a given ζ can be performed by diagonalizing the Laplacian matrix as:

$$L = \Phi' \Lambda \Phi,$$

where Λ is a diagonal matrix with diagonal element $\Lambda_{i,i} = \lambda_i$, and computing:

$$K_\zeta = \Phi' \zeta(\Lambda) \Phi,$$

where $\zeta(\Lambda)$ is a diagonal matrix with diagonal element $\zeta(\Lambda)_{i,i} = \zeta(\lambda_i)$. We can also observe that the diffusion kernel can be written using the matrix exponential as:

$$K_\zeta = e^{-\tau L}.$$

Although other choices of ζ lead to other kernels, discussing them would be beyond the scope of this paper so we will restrict ourselves to using the diffusion kernel as a smoothing functional in the sequel. The conclusion of this section is that by using the diffusion kernel we can build a RKHS $\mathcal{H} = \mathcal{F}$ whose norm $||.||_\mathcal{H}$ is a smoothness functional.

5 Relevance functional

Let us now consider the problem of defining a relevance functional. First observe that any direction $v \in \mathbb{R}^p$ with orthogonal projection v_0 on the linear span of $\{e(x), x \in \mathcal{X}\}$ satisfies $f_{e,v} = f_{e,v_0}$. As a result the search of linear features $f_{e,v}$ can be restricted to directions belonging to this linear span, which can be parametrized as:

$$v = \sum_{x \in \mathcal{X}} \beta(x) e(x), \tag{23}$$

where $\beta \in \mathcal{F}$ is called the dual coordinate of v (defined up to an element of $\{\beta \in \mathcal{F}, K\beta = 0\}$).

The positive semidefinite Gram matrix $K_{x,y} = e(x)'e(y)$, singular due to the centering of profiles (1), defines a RKHS $\mathcal{H} \subset \mathcal{F}$ which consists of features of the form:

$$\begin{aligned} f(.) &= \sum_{x \in \mathcal{X}} \gamma(x) K(x,.) \\ &= \sum_{x \in \mathcal{X}} \gamma(x) e(x)' e(.) \\ &= \left(\sum_{x \in \mathcal{X}} \gamma(x) e(x) \right)' e(.), \end{aligned}$$

where $\gamma \in \mathcal{F}$. Equation (23) shows that $\mathcal{H}$ is exactly the set of linear features $\mathcal{G}$, and by (14) the semi-norm of $\mathcal{H}$ is given by:

$$\forall f_{e,v} \in \mathcal{G}, \quad ||f_{e,v}||_\mathcal{H} = \beta' K \beta, \tag{24}$$

where β is the dual coordinate of v defined by (23).

On the other hand, combining (2), (3) and (23) shows that the variance of a feature $f_{e,v} \in \mathcal{G}$ can be expressed in terms of the dual coordinate β of v as follows:

$$\begin{aligned} V(f_{e,v}) &= \frac{\sum_{x\in\mathcal{X}} f_{e,v}(x)^2}{||v||^2} \\ &= \sum_{x\in\mathcal{X}} \frac{\left(v'e(x)\right)^2}{v'v} \\ &= \frac{\beta' K^2 \beta}{\beta' K \beta}. \end{aligned}$$

From this we see that the larger the ratio between $\beta' K^2 \beta$ and $\beta' K \beta$ the more relevant the feature $f_{e,v}$, where v has dual coordinates β. By observing that $f_{v,e} = K\beta$ and therefore $f'_{e,v} f_{e,v} = \beta' K^2 \beta$, and by (24) we see that a natural relevance functional to plug into (5) in order to counterbalance the effect of $f'_1 f_1$ is the following:

$$h_2(f_{e,v}) = \beta' K \beta = ||f_{e,v}||_{\mathcal{H}}. \tag{25}$$

Indeed the larger $h_2(f_{e,v})$ compared to $f'_{e,v} f_{e,v}$ the smaller $V(f_{e,v})$, and therefore the less variation is captured by $f_{e,v}$. The functional (25) is defined on $\mathcal{G}$ as the norm of a RKHS, which was the goal assigned in Section 2.4.

6 Extracting smooth correlations

6.1 Dual formulation

Let us now put together the elements we have developed up to now. In Section 4 we have shown that any feature $f \in \mathcal{F}$ can be represented as:

$$f = K_1 \alpha,$$

where K_1 is the diffusion kernel Gram matrix derived from the Laplacian matrix L by $K_1 = \exp(-\tau L)$, and α is the dual coordinate vector of f in the corresponding RKHS $\mathcal{H}_1 = \mathcal{F}$. Moreover, we defined a smoothness functional as:

$$\forall f \in \mathcal{F}, \quad h_1(f) = ||f||_{\mathcal{H}_1} = \alpha' K_1 \alpha.$$

In Section 5 we showed that every linear feature $f_{e,v} \in \mathcal{G}$ can also be represented in a dual form:

$$f_{e,v} = K_2 \beta,$$

where K_2 is the kernel Gram matrix $K_2(x,y) = e(x)'e(y)$ for any $(x,y) \in \mathcal{X}^2$ and β is the dual coordinate vector in the corresponding degenerate RKHS $\mathcal{H}_2 = \mathcal{G}$. Moreover a relevance functional was defined as:

$$\forall v \in \mathbb{R}^p, \quad h_e(f_{e,v}) = ||f||_{\mathcal{H}_2} = \beta' K_1 \beta.$$

Plugging these results into (5) leads to the following formulation of the initial problem in terms of dual coordinates:

$$\max_{(\alpha,\beta)\in\mathcal{F}^2} \gamma(\alpha,\beta), \tag{26}$$

with

$$\gamma(\alpha,\beta) = \frac{\alpha' K_1 K_2 \beta}{\left(\alpha'\left(K_1^2+\delta K_1\right)\alpha\right)^{\frac{1}{2}}\left(\beta'\left(K_2^2+\delta K_2\right)\beta\right)^{\frac{1}{2}}}. \tag{27}$$

Observe that this is the dual formulation of (5) except that the optimization is done in $\mathcal{F}\times\mathcal{G}$ instead of $\mathcal{F}_0\times\mathcal{G}$. Moreover, in order keep the interpretation of $||.||_{\mathcal{H}_1}$ as a smoothing functional the kernel K_1 should not be centered in the feature space, as in usual kernel CCA [BJ01] and kernel PCA [SSM99]. As the following Proposition shows, this is however not a problem because the features whose dual coordinates maximize (26) are centered anyway, and the optimization in for $f_1 \in \mathcal{F}$ is therefore equivalent to the maximization for $f \in \mathcal{F}_0$:

Proposition 2 *For any $(\alpha,\beta)\in\mathcal{F}^2$, let α_0 be the dual coordinate of the centered version of $f = K_1\alpha$, i.e.:*

$$\begin{cases} \exists\epsilon\in\mathbb{R}, \quad K_1\alpha_0 = K_1\alpha + \epsilon\mathbf{1}, \\ \sum_{x\in\mathcal{X}} K_1\alpha_0(x) = 0. \end{cases}$$

Then the following holds:

$$\gamma(\alpha_0,\beta) \geq \gamma(\alpha,\beta),$$

with equality if and only if $\alpha = \alpha_0$. In particular, the features whose dual coordinates α and β solve (26) are centered.

Proof Because the profiles $\{e(x), x\in\mathcal{X}\}$ are supposed to be centered we have $K_2\mathbf{1} = 0$, and therefore:

$$\alpha' K_1 K_2 \beta = (\alpha_0' K_1 + \epsilon\mathbf{1}')K_2\beta = \alpha_0' K_1 K_2 \beta.$$

Let $(\phi_1,\ldots,\phi_n)$ denote an orthonormal Fourier basis, where ϕ_1 is constant. Then any feature $f = K_1\alpha$ is centered by removing the contribution of ϕ_1 in its Fourier expansion, i.e.,

$$f_0 = K_1\alpha_0 = \sum_{i=2}^{n} \hat{f}_i\phi_i.$$

As a result we obtain from (11):

$$
\begin{aligned}
\alpha' K_1 \alpha &= ||f||_{\mathcal{H}_1} \\
&= \sum_{i=1}^{n} \frac{\hat{f}_i^2}{\lambda_i} \\
&\geq \sum_{i=2}^{n} \frac{\hat{f}_i^2}{\lambda_i} \\
&= ||K\alpha_0||_{\mathcal{H}_1} \\
&= \alpha_0 K_1 \alpha_0,
\end{aligned}
$$

where the inequality on the third line is an equality if and only if $\hat{f}_1 = 0$, i.e., f is centered. Moreover, using Pythagorean equality in $L^2(\mathcal{X})$ for the orthogonal vectors $\mathbf{1}$ and $K\alpha_0$ we easily get:

$$
\begin{aligned}
\alpha' K_1^2 \alpha &= ||f||_{L^2(\mathcal{X})} \\
&= ||K_1\alpha_0 + \epsilon \mathbf{1}||^2_{L^2(\mathcal{X})} \\
&= ||K_1\alpha_0||^2_{L^2(\mathcal{X})} + ||\epsilon \mathbf{1}||^2_{L^2(\mathcal{X})} \\
&\geq ||K_1\alpha_0||^2_{L^2(\mathcal{X})} \\
&= \alpha_0' K_1^2 \alpha_0
\end{aligned}
$$

Combining this inequalities with the definition of γ (26) proves the Lemma.

6.2 Features extraction

Stated as (26) the problem is similar to the kernel canonical correlation problem studied in [BJ01]. In particular, by differentiating with respect to α and β we see that (α, β) is a solution of (26) if and only if it satisfies the following generalized eigenvalue problem:

$$
\begin{pmatrix} 0 & K_1 K_2 \\ K_2 K_1 & 0 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \rho \begin{pmatrix} K_1^2 + \delta K_1 & 0 \\ 0 & K_2^2 + \delta K_2 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \tag{28}
$$

with ρ the largest possible. The reader is referred to [BJ01] for details about the derivation of (28). Let $\bar{n} = \min(n, p)$. As pointed out in this paper solving (28) provides a series of pairs of features:

$$
\{(\alpha_i, \beta_i), i = 1, \ldots, \bar{n}\}
$$

with decreasing values of $\gamma(\alpha_i, \beta_i)$ for which the gradient $\nabla_{\alpha,\beta}\gamma$ is null, equivalent to the extraction of successive canonical directions with decreasing correlation in classical CCA. The resulting features $f_{1,i} = K_1\alpha_i$ and $f_{2,i} = K_2\beta_i$ are therefore a set of features likely to have decreasing biological relevance when i increases, and are the features we propose to extract in this paper.

The classical way to solve a generalized eigenvalue problem $B\rho = \lambda C\rho$ is to perform a Cholesky decomposition of C as $C = E'E$, to define $\mu = E\rho$ and to solve the standard eigenvector problem $(E')^{-1}BE^{-1}\mu = \lambda\mu$. However the matrix $K_2^2 + \delta K_2$ is singular so it must be regularized for this approach to be numerically stable. Following [BJ01] this can be done by adding $\delta^2/4$ on the diagonal, and observing that:

$$K^2 + \delta K + \frac{\delta^2}{4} I = \left(K + \frac{\delta}{2} I \right)^2 ,$$

leads to the following regularized problem:

$$\begin{pmatrix} 0 & K_1K_2 \\ K_2K_1 & 0 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \rho \begin{pmatrix} (K_1 + \delta' I)^2 & 0 \\ 0 & (K_2 + \delta' I)^2 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix}, \quad (29)$$

where $\delta' = \delta/2$. If (α, β) is an generalized eigenvector solution of (29) belonging to the generalized eigenvalue ρ, then $(-\alpha, \beta)$ belong to $-\rho$. As a result the spectrum of (29) is symmetric : $(\rho_1, -\rho_1, \dots, \rho_n, -\rho_n)$ with $\rho_1 \geq \dots \geq \rho_n$, $\rho_i = 0$ for $i > p$.

6.3 Feature extraction process

Solving (29) results in two sets of features $\{K_1\alpha_i, i = 1, \dots, \bar{n}\}$ and $\{K_2\beta_i, i = 1, \dots, \bar{n}\}$. Features of the form $K\alpha_1$ are computed from the position of the genes in the gene graph, while features of the form $K_2\beta$ are computed from the expression profiles.

In concrete applications, the position of a still uncharacterized gene in the gene graph is not known, while its expression profile can be measured. As a result the only way to extract features for such a gene is to use the features $\{K_2\beta_i, i = 1, \dots, \bar{n}\}$. These features are obtained by projecting the expression profiles to the respective directions:

$$v_i = \sum_{x \in \mathcal{X}} \beta_i(x) e(x), \quad i = 1, \dots, \bar{n}. \quad (30)$$

Therefore features can be extracted from any expression profile e by projections on these directions. We can now summarize a typical use of the the feature extraction process presented in this paper as follows:

- The set of genes $\mathcal{X}$ is supposed to be the disjoint union of two subsets $\mathcal{X}_1$ and $\mathcal{X}_2$. Expression profiles are measured for all genes, but only genes in $\mathcal{X}_1$ are present in the gene network $\mathcal{G} = (\mathcal{X}_1, \mathcal{E})$. Hence $\mathcal{X}_1$ is the set of genes which have been assigned a precise role in a pathway, while $\mathcal{X}_2$ is the set of uncharacterized genes.

- Use the set $\mathcal{X}_1$ to extract features from the set of expression profiles $\{e(x), x \in \mathcal{X}_1\}$ using the graph $\mathcal{G}$, by solving (29).

- Derive a set of expression patterns by (30).

- Extract features from the expression profiles $\{e(x), x \in \mathcal{X}_2\}$ by projecting them on the derived expression patterns.

This process provides a way to replace the expression patterns of an uncharacterized gene by a vector of features which hopefully are more biologically relevant than the raw profiles themselves. Any data mining algorithms, e.g. clustering of functional classification methods, can then be applied on this new representation.

7 Experiments

In order to evaluate the relevance of the pathway-driven features extraction process presented in this paper we performed functional classification experiments with the genes of the yeast *Saccharomyces Cerevisiae*. The main goal of these experiments is to test whether a state-of-the-art classifier, namely a support vector machine, performs best by working directly with the expression profiles of the genes, or by using the vectors of features.

7.1 Pathway data

The LIGAND database of chemical compounds and reactions in biological pathways [GOH$^+$02, GNK98] is part of the Kyoto Encyclopedia of Genes and Genomes (KEGG) [KGKN02, Kan97]. As of February 2002 it consists of a curated set of 3579 metabolic reactions known to take place in some organisms, together with the substrates involved and the classification of the catalyzing enzyme as an EC number.To each reaction are associated one or several EC numbers, and to each EC number are associated one or several genes of the yeast genome. Using this information we created a graph of genes by linking two genes whenever they were assigned two EC number known to catalyze two reactions which share a common main compound (secondary compounds such as water or ATP are discarded).

In other words two genes are linked in the resulting graph if they have the possibility to catalyze two successive reactions, the main product of the first one being the main substrate of the second one. Although it is far from being certain that all the genes candidates to catalyze a given reaction (because they are assigned an EC number supposed to represent a family of potential enzymes catalyzing the reaction) actually catalyze it in the cell, these data nevertheless provide a global picture of the possible relationships between genes in terms of catalyzing properties. In particular a path in this graph corresponds to a possible series of reactions catalyzed by the successive genes met along the path.

The resulting graph involves 774 genes of *S. Cerevisiae*, linked with 16,650 edges.

7.2 Microarray data

Publicly available microarray expression data were collected from the Stanford Microarray Database [SHBK+01]. The data include yeast response to various experimental conditions, including metabolic shift from fermentation to respiration [DIB97], alpha-factor block release, cdc15 block release, elutriation time course, cyclin over-expression [SSZ+98], sporulation [CDE+98], adaptive evolution [FBBR99], stress response [GSK+00], manipulation in phosphate level [ODB00], cell cycle [ZSV+00], growth conditions of excess copper or copper deficiency [GKI+00], DNA damage response [GHM+01], and transfer from a fermentable to a nonfermentable carbon source [KDBS01].

Combining these data results in 330 data points available for 6075 genes, i.e., almost all known or predicted genes of *S. cerevisiae.* Each data point produced by a DNA microarray hybridation experiment represents the ratio of expression levels of a particular gene under two experimental conditions. Following [ESBB98, BGL+00] we don't work directly with this ratio but rather with its normalized logarithm defined as:

$$\forall (x,i) \in \mathcal{X} \times \{1,\dots,330\}, \quad e(x)_i = \frac{\log E_{x,i}/R_{x,i}}{\sqrt{\sum_{j=1}^{330} \log^2 E_{x,i}/R_{x,i}}},$$

where $E_{x,i}$ is the expression level of gene x in experiment i and R_i is the expression level in the corresponding reference state. Missing values were estimated with the software KNNimpute [TCS+01].

7.3 Functional classes

The January 10, 2002, version of the functional classification catalogue of the Comprehensive Yeast Genome Database (CYGD) [MFG+02] is a comprehensive classification of 3936 yeast genes into 259 functional classes organized in a hierarchy. The classes vary in size between 1 and 2258 genes (for the class "subcellular localization"), and not all of them are supposed to be correlated with gene expression [BGL+00]. Only classes with at least 20 genes (after removing the genes present in the gene graph, see next Section) are considered as benchmark datasets for function prediction algorithm in the sequel, which amounts to 115 categories.

7.4 Gene function prediction

Following the general approach presented in Section 6.3 the gene prediction experiment involves two steps:

- The 669 genes in the gene graph derived from the pathway database with known expression profiles are used to perform the feature extraction process by solving (30).

- The resulting linear features are extracted from the expression profiles of the disjoint set of 2688 genes which are in the CYGD functional catalogue but not in the pathway database. Systematic evaluation of the performance of support vector machines to predict each CYGD class either from the expression profiles themselves [BGL$^+$00] or from the features extracted is then performed on this set of genes using 3-fold cross-validation averaged over 10 iterations.

Support vector machine (SVM) [Vap98, CST00, SS02] is a class of machine learning algorithms for supervised classification which has been shown to perform better that other machine learning techniques, including Fisher's linear discriminant, Parzen windows and decision trees on the problem of gene functional classification from expression profiles [BGL$^+$00]. We therefore use SVM as a state-of-the-art learning algorithm to assess the gain resulting from replacing the original expression profiles by vectors of features.

Experiments were carried out with SVM Light [Joa99], a public and free implementation of SVMs. To ensure a comparison as fair as possible between different data representations, all vectors were scaled to unit length before being sent to the SVM, and all SVM used a radial basis kernel with unit width, i.e., $k(x, y) = \exp(-||x - y||^2)$. The trade-off parameter between training error and margin was set to its default value (namely 1 in the case where all vectors have unit length), and the cost factor by which training errors on positive examples outweigh errors on negative examples was set equal to the ratio of the number of positive examples and the number of negative examples in the training set.

We compared the performance of SVM working directly on the expression profiles. as in [BGL$^+$00], with SVM working on the vectors of features extracted by the procedure described in this paper, for various choices of regularization parameters δ, width of the diffusion kernel τ and numbers of features selected.

For each experiment the performance is measured by the ROC index, defined as the area under the ROC curve, i.e., the plot of true positives versus false positives, and normalized to 100 for a perfect classifier. The ROC curve itself is obtained by varying a threshold and classify genes by comparing the score output by the SVM with this threshold. A random classifier has an average ROC index of 50.

7.5 Setting the parameters

Our feature extraction process contains two free parameters, namely the width τ of the diffusion kernel and the regularization parameter δ. Intuitively, the larger τ and δ, the smoother and more relevant the features extracted, at the expense of a decrease between their correlations. As pointed out in [BJ01] the parameter δ is expected to decrease linearly with n, and a reasonable value is $\delta = 0.001$ for n of the order of 1000. An initial value of $\tau = 1$ was chosen.

We varied independently δ and τ in order to check their influence. For a fixed $\delta = 0.001$ we tested the performance of SVM based on the features extracted with the parameter $\tau \in \{0.5, 1, 2, 5\}$, where all 330 features are used.

Table 1: Performance comparison for various τ

δ	τ	Average ROC	Percentage of classes best predicted
0.001	0.5	61.4	37
0.001	1	61.4	35
0.001	2	60.0	20
0.001	5	55.2	8

Table 2: Performance comparison for various δ

δ	τ	Average ROC	Percentage of classes best predicted
0.0005	1	61.4	17
0.001	1	61.4	18
0.002	1	61.4	25
0.005	1	61.6	39

Table 1 shows the ROC index averaged over all 115 classes with more than 20 genes for each of the four SVM, as well as the percentage of classes best predicted by each method. The best performance is reached for $\tau = 1$, with an important deterioration when τ increases to 5. A larger τ means by (22) that rugged features are more strongly penalized, so larger τ tend to generate smoother features. The deterioration when τ increases shows the importance of not excessively penalizing ruggedness.

We also checked the influence of the regularization parameter δ, which controls the trade-off between correlation on the one hand, smoothness and relevance on the other hand. Table 2 compares the performances of SVM based on the features extracted with the parameters $\tau = 1$ and $\delta \in \{0.0005, 0.001, 0.002, 0.005\}$. This shows a small (in terms of ROC index increase) but consistent (in terms of number of classes best predicted) increase in performance when δ increases from 0.0005 to 0.005. This illustrates the importance of regularization, and therefore the improvement gained by imposing some smoothness and relevance constraints to the features.

7.6 Number of features

From now on we fix the parameters to $\tau = 1$ and $\delta = 0.001$. As the feature extraction process is supposed to extract up to $p = 330$ features by decreasing biological relevance, one might ask if classification performance could increase by only keeping the most relevant features, and hopefully removing noise by discarding the remaining ones. To check this we measured the performance of SVM using an increasing number of features. Results are shown on Table 3, and show that it is on average more interesting to use all features as the performance increases with the number of features used. Exceptions to this average principle include classes such as fermentation, ionic homeostasis, assembly of protein complexes, vacuolar transport, phosphate metabolism or nucleus organization, which are better predicted with less than 100 features as shown on Figure 7.6

Table 3: Performance comparison for various numbers of features, with $\delta = 0.001$ and $\tau = 1$

Number of features	Average ROC	Percentage of classes best predicted
50	55.3	3
100	57.9	10
150	58.9	9
200	59.9	7
250	60.6	17
300	61.2	17
330	61.4	37

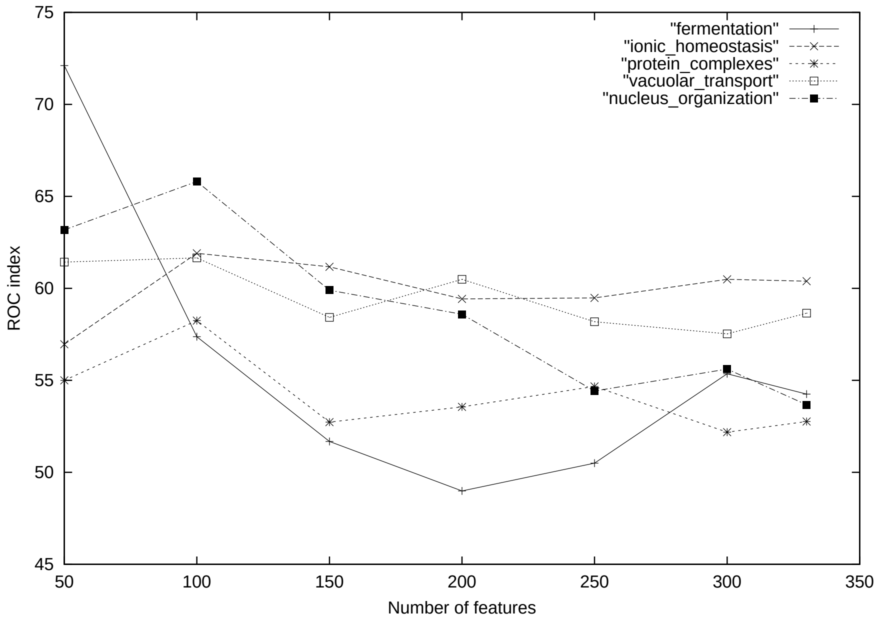

Figure 1: Classification performance for various classes

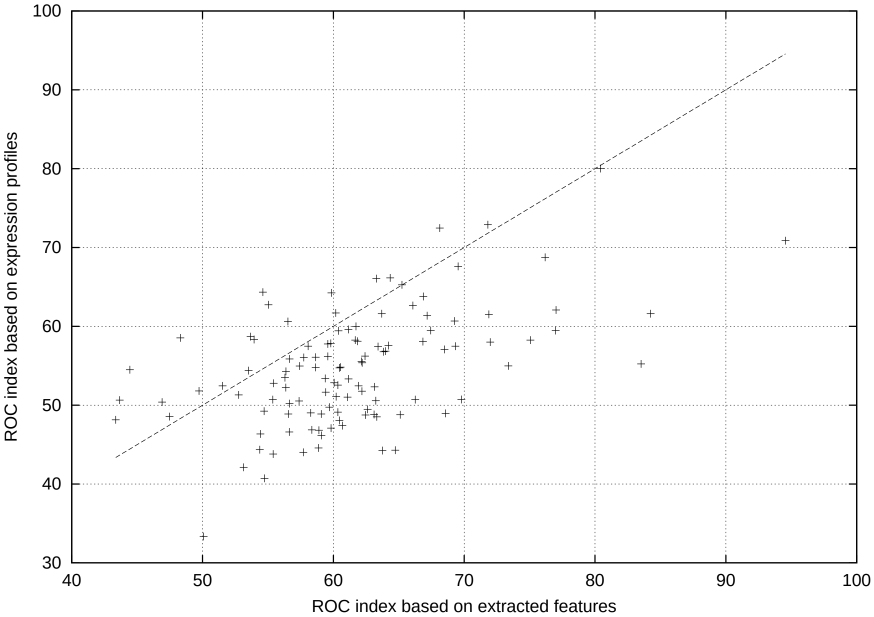

Figure 2: Comparison of the classification performance of SVM based on expression profiles (y axis) or extracted features (x axis). Each point represents one functional class.

7.7 Functional classification performance

In order to check whether the features extraction provides any advantage over the direct use of expression profiles for gene function prediction we finally compared the performance of a SVM using all features extracted with the parameters $\delta = 0.001$ and $\tau = 1$, with the performance of a SVM using directly the gene expression profiles. Figure 7.7 shows the ROC index obtained by each of the two methods for all 115 functional classes. Except for a few classes, there is a clear improvement in classification performance when the genes are represented as vectors of features, and not directly as expression profiles.

Table 4 shows that the ROC index averaged over all classes increases significantly between the two representations (from 54.9 to 61.2). Moreover Figure 7.7 shows that most of the classes seem almost impossible to learn from their expression profiles only (when the ROC index is around 45 - 55, i.e. not better than a random classifier), but can somehow be learned by their vectors of features, as the ROC index jumps in the range 55-65 for many of those classes. Some classes exhibit a dramatic increase in ROC index, as shown in Table 5 which lists the classes largest absolute increase in ROC index between the two experiments.

Table 4: ROC index averaged over 115 functional classes by SVM using different representations of the data

Data representation	Average ROC
Expression profiles	54.6
Vector of features	61.4

Table 5: ROC index for the prediction of categories based on expression profiles or features vectors. The categories listed are the one which exhibit the largest increase in ROC index between these two representations.

Class	Expression	Features	Increase
Heavy metal ion transporters (Cu, Fe, etc.)	55.2	83.5	+28.3
Ribosome biogenesis	70.9	94.6	+23.7
Protein synthesis	61.6	84.3	+22.7
Directional cell growth (morphogenesis)	44.3	64.7	+20.4
Regulation of nitrogen and sulphur utilization	49.0	68.6	+19.6
Nitrogen and sulfur metabolism	44.3	63.8	+19.5
Translation	50.7	69.8	+19.1
Cytoplasm	55.0	73.4	+18.4
Endoplasmic reticulum	59.5	77.0	+17.5
Amino acid transport	75.1	58.3	+16.8

8 Discussion and conclusion

This paper proposes an algorithm to extract features from gene expression profiles based on the knowledge of a biochemical network linking a subset of genes. Based on the simple idea that relevant features are likely to exhibit correlation with respect to the topology of the network, we end up with a formulation which involves encoding the network and the set of expression profiles into to kernel functions, and performing a regularized canonical correlation analysis in the corresponding reproducible kernel Hilbert spaces.

Results presented in Section 7 are encouraging and confirm the intuition that incorporating valuable information, such as the knowledge of the precise position of many genes in a biochemical network, helps extracting relevant informations from expression profiles. While this problem has still attracted relatively few attention because the number of expression data has always been small compared to the number of genes until recently, it is expected to be more and more important as the production of expression data becomes cheaper and the underlying technology more widespread.

A detailed analysis of the experimental results reveals that functional categories related to metabolism, protein synthesis and subcellular localization benefit the most from the representation of genes as vectors of features. In the case of metabolism and protein synthesis related categories, this can be explained by the fact that many pathways related to this process are present in the pathway database, so relevant features have probably been extracted. The case of subcellular localization proteins is more surprising, as they seem to be more related to structural properties than functional properties of the genes, but certainly reflects the functional role of the organelles themselves. As an example a sudden need of energy might promote the activity in mitochondria and require the synthesis of proteins to be directed to this location, even though they might not be directly involved as enzymes.

On the technical point of view the approach developed in this paper can be seen as an attempt to encode various types of information about genes into kernels. The diffusion kernel K_1 encodes the gene network, and the linear kernel K_2 summarizes the expression profiles. Recent research shows that this approach can in fact be generalized to many other sources of information about genes, as many kernels have been engineered and continue to be developed for particular types of data. Apart from classical kernels for finite-dimensional real-valued vectors [Vap98] which can be used to encode any vectorial gene representation, e.g. expression profiles, and from diffusion kernels which can encode any gene network, e.g. network derived from biochemical pathway or protein interaction networks, relevant examples of recently developed kernels include the Fisher kernel to encode how the amino-acid sequence of a protein is related to a given hidden Markov model [JDH00] or to encode the arrangement of transcription factor binding site motifs in its promoter region [PWCG01], several string kernels to encode the information present in the amino-acid sequence itself [Hau99, Wat00, LEN02, Ver02a, LSST$^+$02], or a tree kernel to encode the phylogenetic profile of a protein [Ver02b]. This increasing list suggests a unified framework to represent various types of informations, which is obtained by "kernelizing the proteome", i.e., tranforming any type of information into an adequate kernel.

Parallel to the apparition of new kernels recent years have witnessed the development of new methods, globally referred to as *kernel methods*, to perform various data mining algorithm from the knowledge of the kernel matrix only. Apart from the most famous support vector machine algorithm for classification and regression [BGV92, Vap98], other kernel methods include principal component analysis [SSM99], clustering [BHHSV01], Fisher discriminants [MRW$^+$99] or independent component analysis [BJ01].

These recent developments open the door to new analysis opportunities which we believe can be particularly suited to the new discipline of proteomics whose central concepts, genes or proteins, are defined through a variety of different points of view (as sequences, structures, expression patterns, position in networks, ...), the integration of which promises to unravel some of the secrets of life.

9 Acknowledgements

We would like to thank Yasushi Okuno for help and advices with the pathway data, and Olivier Bousquet for simulating discussions. This work was supported by the Research for the Future Program of the Ministry of Education, Culture, Sports, Science and Technology, Japan.

References

[AMK00] T. Akutsu, S. Miyano, and S. Kuhara. Inferring qualitative relations in genetic networks and metabolic pathways. *Bioinformatics*, 16(8):727–734, 2000.

[Aro50] N. Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68:337 – 404, 1950.

[BB00] P.O. Brown and D. Botstein. Exploring the new world of the genome with dna microarrays. *Nature Genetics*, 21:33–37, 2000.

[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.

[BHHSV01] Asa Ben-Hur, David Horn, Hava T. Siegelmann, and Vladimir Vapnik. Support vector clustering. *Journal of Machine Learning Research*, 2:125–137, 2001.

[BJ01] F. R. Bach and M. I. Jordan. Kernel independent component analysis. Technical Report UCB//CSD-01-1166, UC Berkeley, 2001.

[CDE+98] S. Chu, J. DeRisi, M. Eisen, J. Mulholland, D. Botstein, P.O. Brown, and I. Herskowitz. The transcriptional program of sporulation in budding yeast. *Science*, 282:699–705, 1998.

[Cha84] I. Chavel. *Eigenvalues in Riemannian geometry.* Academic Press, Orlando, Fl., 1984.

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

[CST00] Nello Cristianini and John Shawe-Taylor. *An introduction to Support Vector Machines and other kernel-based learning methods.* Cambridge University Press, 2000.

[DGL$^+$01] E. B. Davies, G. M. L. Gladwell, J. Leydold, , and P. F. Stadler. Discrete nodal domain theorems. *Lin. Alg. Appl.*, 336:51–60, 2001.

[DIB97] Joseph L. DeRisi, Vishwanath R. Iyer, and Patrick O. Brown. Exploring the metabolic and genetic control of gene expression on a genomic scale. *Science*, 278(5338):680–686, 1997.

[dV93] Y.C. de Verdière. Multiplicités des valeurs propres Laplaciens discrets et Laplaciens continus. *Rendiconti di Matematica*, 13:433–460, 1993.

[ESBB98] Michael B. Eisen, Paul T. Spellman, Patrick O. Brown, and David Botstein. Cluster analysis and display of genome-wide expression patterns. *Proc. Natl. Acad. Sci. USA*, 95:14863–14868, Dec 1998.

[FBBR99] Tracy L. Ferea, David Botstein, Patrick O. Brown, and R. Frank Rosenzweig. Systematic changes in gene expression patterns following adaptive evolution in yeast. *Proc. Natl. Acad. Sci. USA*, 96(17):9721–9726, 1999.

[FLNP00] Nir Friedman, Michal Linial, Iftach Nachman, and Dana Pe'er. Using bayesian networks to analyze expression data. *Journal of Computational Biology*, 7(3-4):601–620, 2000.

[Fri93] J. Friedman. Some geometric aspects of graphs and their eigenfunctions. *Duke MAthematical journal*, 69:487–525, March 1993.

[GHM$^+$01] A.P. Gasch, M. Huang, S. Metzner, D. Botstein, S.J. Elledge, and P.O. Brown. Genomic expression responses to DNA-damaging agents and the regulatory role of the yeast ATR homolog Mec1p. *Mol. Biol. Cell*, 12(10):2987–3003, 2001.

[GJP95] Frederico Girosi, Michael Jones, and Tomaso Poggio. Regularization theory and neural networks architectures. *Neural Computation*, 7(2):219–269, 1995.

[GKI$^+$00] C. Gross, M. Kelleher, V.R. Iyer, P.O. Brown, and D.R. Winge. Identification of the copper regulon in saccharomyces cerevisiae by DNA microarrays. *J. Biol. Chem.*, 275(41):32310–32316, 2000.

[GNK98] S. Goto, T. Nishioka, and M. Kanehisa. LIGAND: chemical database for enzyme reactions. *Bioinformatics*, 14:591–599, 1998.

[GOH$^+$02] S. Goto, Y. Okuno, M. Hattori, T. Nishioka, and M. Kanehisa. LIGAND: database of chemical compounds and reactions in biological pathways. *Nucleic Acid Research*, 30:402–404, 2002.

[GSK$^+$00] Audrey P. Gasch, Paul T. Spellman, Camilla M. Kao, Orna Carmel-Harel, Michael B. Eisen, Gisela Storz, David Botstein, and Patrick O. Brown. Genomic expression programs in the response of yeast cells to environmental changes. *Mol. Biol. Cell*, 11:4241–4257, Dec 2000.

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

[HGJY02] A.J. Hartemink, D.K. Gifford, T.S. Jaakkola, and R.A. Young. Using graphical models and genomic expression data to statistically validate models of genetic regulatory networks. 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 422–433. World Scientific, 2002.

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

[HZZL02] D. Hanisch, A. Zien, R. Zimmer, and T. Lengauer. Co-clustering of biological networks and gene expression data. *Bioinformatics*, 2002.

[Iva76] V.V. Ivanov. *The theory of approximate methods and their application to the numerical solution of singular integral equations.* Nordhoff International, Leiden, 1976.

[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.

[Joa99] Thorsten Joachims. Making large-scale svm learning practical. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*, pages 169–184. MIT Press, 1999.

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

[Kan97] M. Kanehisa. A database for post-genome analysis. *Trends Genet.*, 13:375–376, 1997.

[KDBS01] K.M. Kuhn, J.L. DeRisi, P.O. Brown, and P. Sarnow. Global and specific translational regulation in the genomic response of Saccharomyces cerevisiae to a rapid transfer from a fermentable to a nonfermentable carbon source. *Mol. Cell. Biol.*, 21(3):916–927, 2001.

[KGKN02] M. Kanehisa, S. Goto, S. Kawashima, and A. Nakaya. The KEGG databases at GenomeNet. *Nucleic Acid Research*, 30:42–46, 2002.

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

[LSST+02] Huma Lodhi, Craig Saunders, John Shawe-Taylor, Nello Cristianini, and Chris Watkins. Text classification using string kernels. *Journal of Machine Learning Research*, 2:419–444, 2002.

[MFG+02] H.W. Mewes, D. Frishman, U. Güldener, G. Mannhaupt, K. Mayer, M. Mokrejs, B. Morgenstern, M. Münsterkoetter, S. Rudd, and B. Weil. MIPS: a database for genomes and protein sequences. *Nucleic Acid Research*, 30(1):31–34, 2002.

[Moh91] B. Mohar. The laplacian spectrum of graphs. In Y. Alavi, G. Chartrand, O. Ollermann, and A. Schwenk, editors, *Graph theory, combinatorics, and applications*, pages 871–898, New-York, 1991. John Wiley and Sons, Inc.

[Moh97] B. Mohar. Some applications of laplace eigenvalues of graphs. In G. Hahn and G. Sabidussi, editors, *Graph Symmetry: Algebraic Methods and Applications*, volume 497 of *NATO ASI Series C*, pages 227–275. Kluwer, Dordrecht, 1997.

[MPT+99] Edward M. Marcotte, Matteo Pellegrini, Michael J. Thompson, Todd O. Yeates, and David Eisenberg. A combined algorithm for genome-wide prediction of protein function. *Nature*, 402:83–86, November 1999.

[MRW+99] S. Mika, G. Rätsch, J. Weston, B. Schölkopf, and K.R. Müller. Fisher discriminant analysis with kernels. In Y.-H. Hu, J. Larsen, E. Wilson, and S. Douglas, editors, *Neural Networks for Signal Processing IX*, pages 41–48. IEEE, 1999.

[NGK01] A. Nakaya, S. Goto, and M. Kanehisa. Extraction of correlated gene clusters by multiple graph comparison. In *Genome Informatics 2001*, pages 44–53. Universal Academy Press, Tokyo, Japan, 2001.

[ODB00] Nobuo Ogawa, Joseph DeRisi, and Patrick O. Brown. New components of a system for phosphate accumulation and polyphosphate metabolism in saccharomyces cerevisiae revealed by genomic expression analysis. *Mol. Biol. Cell*, 11:4309–4321, Dec 2000.

[OLP+00] Ross Overbeek, Niels Larsen, Gordon D. Pusch, Mark D'Souza, Evgeni Selkov Jr, Nikos Kyrpides, Michael Fonstein, Natalia Maltsev, and Evgeni Selkov. WIT: integrated system for high-throughput genome sequence analysis and metabolic reconstruction. *Nucleic Acid Research*, 28:123–125, 2000.

[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.

[Sai88] S. Saitoh. *Theory of reproducing Kernels and its applications*. Longman Scientific & Technical, Harlow, UK, 1988.

[SHBK$^+$01] G. Sherlock, T. Hernandez-Boussard, A. Kasarskis, G. Binkley, J.C. Matese, S.S. Dwight, M. Kaloper, S. Weng, H. Jin, C.A. Ball, M.B. Eisen, and P.T. Spellman. The stanford microarray database. *Nucleic Acid Research*, 29(1):152–155, Jan 2001.

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

[SSDB95] M. Schena, D. Shalon, R.W. Davis, and P.O. Brown. Quantitative monitoring of gene expression patterns with a complimentary DNA microarray. *Science*, 270:467–470, 1995.

[SSM98] A.J. Smola, B. Schölkopf, and K.-R. Müller. The connection between regularization operators and support vector kernels. *Neural Networks*, 11(4):637–649, 1998.

[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.

[SSZ$^+$98] Paul T. Spellman, Gavin Sherlock, Michael Q. Zhang, Vishwanath R. Iyer, Kirk Anders, Michael B. Eisen, Patrick O. Brown, David Botstein, and Bruce Futcher. Comprehensive identification of cell cycle-regulated genes of the yeast saccharomyces cerevisiae by microarray hybridization. *Mol. Biol. Cell*, 9:3273–3297, 1998.

[Sta96] Peter F. Stadler. Landscapes and their correlation functions. *J. Math. Chem.*, 20:1–45, 1996.

[TA77] A.N. Tikhonov and V.Y. Arsenin. *Solutions of ill-posed problems*. W.H. Winston, Washington, D.C., 1977.

[TCS$^+$01] Olga Troyanskaya, Michael Cantor, Gavin Sherlock, Pat Brown, Trevor Hastie, Robert Tibshirani, David Botstein, and Russ B. Altman. Missing value estimation methods for NA microarrays. *Bioinformatics*, 17:520–525, 2001.

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

[vdH96] H. van der Holst. *Topological and spectral graph characterizations.* PhD thesis, Universiteit van Amsterdam, 1996.

[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*, 2002. To appear.

[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.

[ZSV$^+$00] Gefeng Zhu, Paul T. Spellman, Tom Volpe, Patrick O. Brown, David Botstein, Trisha N. Davis, and Bruce Futcher. Two yeast forkhead genes regulate the cell cycle and pseudohyphal growth. *Nature*, 406:90–94, 2000.