Framework for kernel regularization with application to protein clustering

Fan Lu[†], Sündüz Keleş[†‡], Stephen J. Wright[§], and Grace Wahba[†‡§¶]

Departments of [†]Statistics, [‡]Biostatistics and Medical Informatics, and [§]Computer Sciences, University of Wisconsin, Madison, WI 53706

Contributed by Grace Wahba, June 27, 2005

We develop and apply a previously undescribed framework that is designed to extract information in the form of a positive definite kernel matrix from possibly crude, noisy, incomplete, inconsistent dissimilarity information between pairs of objects, obtainable in a variety of contexts. Any positive definite kernel defines a consistent set of distances, and the fitted kernel provides a set of coordinates in Euclidean space that attempts to respect the information available while controlling for complexity of the kernel. The resulting set of coordinates is highly appropriate for visualization and as input to classification and clustering algorithms. The framework is formulated in terms of a class of optimization problems that can be solved efficiently by using modern convex cone programming software. The power of the method is illustrated in the context of protein clustering based on primary sequence data. An application to the globin family of proteins resulted in a readily visualizable 3D sequence space of globins, where several subfamilies and subgroupings consistent with the literature were easily identifiable.

classification | convex cone programming | dissimilarity information | trace penalty | sequence data

It has long been recognized that symmetric positive definite kernels (hereafter "kernels") play a key role in function estimation (1, 2), clustering and classification, dimension reduction, and other applications. Such kernels can be defined on essentially any conceivable domain of interest (3), originally function spaces and, more recently, finite (but possibly large) collections of trees, graphs, images, DNA and protein sequences, microarray gene expression chips, and other objects. A kernel defines a distance metric between pairs of objects in the domain that admits an inner product. Thus, they play a key role in the implementation of classification algorithms [by means of support vector machines (SVMs)] and clustering (via *k*-means algorithms, for example), along with their more classical role in function approximation and estimation and the solution of ill-posed inverse problems (4). Since the mid-1990s, when the key role of these kernels became evident in SVMs (5–8), a massive body of literature has grown related to the use and choice of kernels in many domains of application, including, notably, computational biology (9). A Google search as of the date of this writing gave >3 million results on the phrase "Kernel Methods," along with an ad from Google soliciting job applications from computer scientists.

Mathematically defined kernels, for example, spline kernels, radial basis functions, and related positive definite functions defined on Euclidean space, have long been the workhorses in the field, generally with one or a few free parameters estimated from the data (see, for example, ref. 10). A recent work (11) proposes estimating a kernel by optimizing a linear combination of prespecified kernels through a semidefinite programming approach. Recent literature on kernel construction and use in various contexts is available at the NIPS2004 web site (http://books.nips.cc/nips17.html) or in ref. 12.

It is frequently possible to use expert knowledge or other information to obtain dissimilarity scores for pairs of objects, which serve as pseudodistances between the objects. There are two problem types of interest. The first is to estimate full relative position information for a (training) set of objects in a space of preferably low dimension to visualize the data or to conduct further processing, typically, classification or clustering. One traditional approach for this purpose is multidimensional scaling (13), which continues to be an active research area. The second problem is to place new objects in the space, given some dissimilarity information between them and some members of the training set, in the coordinate space of the training set.

This work proposes regularized kernel estimation (RKE), a unified framework for solving both problems by fitting a positive definite kernel from possibly crude, noisy, incomplete, inconsistent, weighted, repetitious dissimilarity information, in a fully nonparametric approach, by solving a convex optimization problem with modern convex cone programming tools. The basic idea is to solve an optimization problem that trades off goodness of fit to the data and a complexity (shrinkage) penalty on the kernel that is used to fit the data, analogous to the well known bias–variance tradeoff in the spline and ill-posed inverse literature but not exactly the same. Within this framework, we provide an algorithm for placing new objects in the coordinate space of the training set.

The method can be used instead of multidimensional scaling to provide a coherent set of coordinates for the given objects in few or many dimensions without problems with local minima or (some) missing data. It also can be used to solve problems discussed in ref. 11 but in a fully nonparametric way.

The feasibility of the RKE approach is demonstrated in the context of protein sequence clustering, by applying the method to global pairwise alignment scores of the heme-binding protein family of globins. In this example, we are already able to visualize the known globin subfamilies from a 3D plot of the training sequence coordinates that are obtained by the regularized kernel estimate. Furthermore, apparent subclusterings and outliers of the known globin subfamilies from the 3D plot reveal interesting observations consistent with the literature. Clustering of protein sequences from a family to identify subfamilies or clustering and classification of protein domains to determine protein function present one major application area for the previously undescribed framework presented here. However, we envision many more applications involving clustering and classification tasks in biological and nonbiological data analysis; some of these applications are detailed in *Discussion*.

In *Dissimilarity Information and RKE*, we present the general formulation of the problem and define the family of RKEs. *Numerical Methods for RKE* describes the formulation of RKE problems and the task of placing test data in the coordinate space of training data as general convex cone problems. Also included is a brief discussion on tuning the parameters of the estimation procedure. *Protein Clustering and Visualization with RKE* pre-

Freely available online through the PNAS open access option.

Abbreviations: RKE, regularized kernel estimation; SVM, support vector machine; MSVM, multicategory SVM; HMM, hidden Markov model.

[¶]To whom correspondence should be addressed. E-mail: wahba@stat.wisc.edu.

www.pnas.org/cgi/doi/10.1073/pnas.0505411102

sents an application to the globin protein family to identify subfamilies and discusses the biological implication of the results. Examples of placing test data points in the coordinate system of training protein sequences are illustrated here. We conclude with a summary and possible future directions in *Discussion*.

Dissimilarity Information and RKE

Given a set of N objects, suppose we have obtained a measure of dissimilarity, d_{ij}, for certain object pairs (i, j). We introduce the class of RKEs, which we define as solutions to optimization problems of the following form:

$$\min_{K \in S_N} \sum_{(i,j) \in \Omega} L(w_{ij}, d_{ij}, \hat{d}_{ij}(K)) + \lambda J(K), \qquad [1]$$

where S_N is the convex cone of all real nonnegative definite matrices of dimension N, Ω is the set of pairs for which we use dissimilarity information, L is some reasonable loss function, where $\hat{d}_{\mathrm{ij}}$ is the dissimilarity induced by K and L is convex in K. J is a convex kernel penalty (regularizing) functional, and λ is a tuning parameter balancing fit to the data and the penalty on K. The induced dissimilarity, which is a real squared distance admitting of an inner product, is $\hat{d}_{ij} = K(i, i) + K(j, j) - 2K(i, j) = B_{ij} \cdot K$, where $K(i, j)$ is the (i, j) entry of K; B_{ij} is a symmetric matrix of dimension N with all elements 0 except $B_{ij}(i, i) = B_{ij}(j, j) = 1$, $B_{ij}(i, j) = B_{ij}(j, i) = -1$; and the inner (dot) product of two matrices of the same dimensions is defined as $A \cdot B = \Sigma_{i,j} A(i, j) \cdot B(i, j) \equiv \mathrm{trace}(A^T B)$. The w_{ij} are weights that may, if desired, be associated with particular (i, j) pairs. There are essentially no restrictions on the set of pairs other than requiring that the graph of the objects with pairs connected by edges be connected. A pair may have repeated observations, which just yield an additional term in Eq. **1** for each separate observation. If the pair set induces a connected graph, then the minimizer of Eq. **1** will have no local minima.

Although it is usually natural to require the observed dissimilarity information $\{d_{ij}\}$ to satisfy $d_{ij} \geq 0$ and $d_{ij} = d_{ji}$, the general formulation above does not require these properties to hold. The observed dissimilarity information may be incomplete (with the restriction noted); it may not satisfy the triangle inequality; or it may be noisy. It also may be crude, as, for example, when it encodes a small number of coded levels such as "very close," "close," "distant," and "very distant."

In this work, we consider two special cases of the formulation in Eq. **1**, the first for its use in the application to be discussed.

Numerical Methods for RKE

Here, we describe a specific formulation of the approach in *Dissimilarity Information and RKE* based on a linearly weighted l_1 loss and use the trace function in the regularization term to promote dimension reduction. The resulting problem is as follows:

$$\min_{K \succeq 0} \sum_{(i,j) \in \Omega} w_{ij} |d_{ij} - B_{ij} \cdot K| + \lambda\ \mathrm{trace}(K). \qquad [2]$$

We show how this formulation can be posed as a general conic optimization problem and also describe a "newbie" formulation in which the known solution to Eq. **2** for a set of N objects is augmented by the addition of one more object together with its dissimilarity data. A variant of Eq. **2**, in which a quadratic loss function is used in place of the l_1 loss function, is described in *Supporting Text*, which is published as supporting information on the PNAS web site. We remark that trace was used as a regularizer in ref. 11 in a different approach to obtain K, which limited K to a linear combination of prespecified kernels.

General Convex Cone Problem. We specify here the general convex cone programming problem. This problem, which is central to modern optimization research, involves some unknowns that are vectors in Euclidean space and others that are symmetric matrices. These unknowns are required to satisfy certain equality constraints and also are required to belong to cones of a certain type. The cones have the common feature that they all admit a self-concordant barrier function, which allows them to be solved by interior-point methods that are efficient in both theory and practice.

To describe the cone programming problem, we define some notation. Let $\mathcal{R}^p$ be Euclidean p-space, and let P_p be the nonnegative orthant in $\mathcal{R}^p$, that is, the set of vectors in $\mathcal{R}^p$ whose components are all nonnegative. We let Q_q be the second-order cone of dimension q, which is the set of vectors $x = (x(1), \ldots, x(q)) \in \mathcal{R}^q$ that satisfy the condition $x(1) \geq [\Sigma_{i=2}^{q} x(i)^2]^{1/2}$. We define S_s to be the cone of symmetric positive semidefinite $s \times s$ matrices of real numbers.

The general convex cone problem is then

$$\begin{aligned} &\min_{X_j, x_i, z} \sum_{j=1}^{n_s} C_j \cdot X_j + \sum_{i=1}^{n_q} c_i \cdot x_i + g \cdot z \\ &\text{s.t.} \sum_{j=1}^{n_s} A_{rj} \cdot X_j + \sum_{i=1}^{n_q} a_{ri} \cdot x_i + g_r \cdot z = b_r,\ \ \forall_r \qquad [3] \\ &X_j \in S_{s_j}\ \forall_j;\ x_i \in Q_{q_i}\ \forall_i;\ z \in P_p. \end{aligned}$$

Here, C_j, A_{rj} are real symmetric matrices (not necessarily positive semidefinite) of dimension s_j; $c_i, a_{ri} \in \mathcal{R}^{q_i}$; $g, g_r \in \mathcal{R}^p$; $b_r \in \mathcal{R}^1$.

The solution of a general convex cone problem can be obtained numerically by using publicly available software such as SDPT3 (14) and DSDP5 (15).

RKE with l_1 Loss. To convert the problem of Eq. **2** into a convex cone programming problem, without loss of generality, we let Ω contain m distinct (i, j) pairs, which we index with $r = 1, 2, \ldots, m$. Define I_N to be the N-dimensional identity matrix and $e_{m,r}$ to be vector of length $2m$ consisting of all zeros except for the rth element being 1 and $(m + r)$-th element being -1. If we denote the rth element of Ω as $(i(r), j(r))$, and with some abuse of the notation let $i = i(r)$, $j = j(r)$, and $w \in P_{2m}$ with $w(r) = w(r + m) = w_{i(r), j(r)}$, $r = 1, \ldots, m$, we can formulate the problem of Eq. **2** as follows:

$$\begin{aligned} &\min\nolimits_{K \succeq 0, u \geq 0} w \cdot u + \lambda I_N \cdot K \\ &\text{s.t. } d_{ij} - B_{ij} \cdot K + e_{m,r} \cdot u = 0,\ \ \forall_r, \qquad [4] \\ &K \in S_N,\ u \in P_{2m}. \end{aligned}$$

Newbie Formulation. We now consider the situation in which a solution K_N of Eq. **2** is known for some set of N objects. We wish to augment the optimal kernel (by one row and column), without changing any of its existing elements, to account for a new object. That is, we wish to find a new "pseudo-optimal" kernel $\tilde{K}_{\mathrm{n}+1}$ of the form

$$\tilde{K}_{N+1} = \begin{bmatrix} K_N & b^T \\ b & c \end{bmatrix} \succeq 0, \qquad [5]$$

(where $b \in \mathcal{R}^N$ and c is a scalar) that solves the following optimization problem:

$$\begin{aligned} &\min\nolimits_{c \geq 0, b} \Sigma_{i \in \Psi}\, w_i |d_{i,N+1} - B_{i,N+1} \cdot K_{N+1}| \\ &\text{s.t. } b \in \ \mathrm{Range}(K_N),\ c - b^T K_N^{+} b \geq 0, \end{aligned} \qquad [6]$$

where K_N^+ is the pseudoinverse of K_N, and Ψ is a subset of $\{1, 2, \ldots, N\}$ of size t. The quantities w_i, $i \in \Psi$ are the weights assigned to the dissimilarity data for the new point. The constraints in this problem are the necessary and sufficient conditions for $\tilde{K}_{N+1}$ to be positive semidefinite.

Suppose that K_N has rank $p < N$ and let $K_N = \Gamma\Lambda\Gamma^T$, where $\Gamma_{N\times p}$ is the orthogonal matrix of nonzero eigenvectors and Λ is the $p \times p$ matrix of positive eigenvalues of K_N. By introducing the variable $\tilde{b}$ and setting $b = \Gamma\Lambda^{1/2}\tilde{b}$, we can ensure that the requirement $b \in \text{Range}(K_N)$ is satisfied. We also introduce the scalar variable $\tilde{c}$, and enforce $c \geq \tilde{c}^2$ by requiring that

$$Z \overset{\text{def}}{=} \begin{bmatrix} 1 & \tilde{c} \\ \tilde{c} & c \end{bmatrix} \in S_2. \qquad \textbf{[7]}$$

By using these changes of variable, the condition $c - b^T K_N^+ b \geq 0$ is implied by the condition

$$x \overset{\text{def}}{=} [\tilde{c}\ \tilde{b}^T]^T \in Q_{p+1}.$$

Further, we define the $N \times (p + 1)$ matrix $\Sigma \overset{\text{def}}{=} [0_N\ 2\Gamma\Lambda^{1/2}]$, where 0_N is the zero vector of length N, and let $\Sigma_{i\cdot}$ be the row vector consisting of the $p + 1$ elements of row i of Σ. We use $K_N(i, i)$ to denote the iith entry of K_N and define the weight vector $w \in P_{2t}$ with components $w(r) = w(t + r) = w_{i(r)}$, $r = 1, \ldots, t$. We then replace the Problem **6** by the following equivalent convex cone program:

$$\min_{Z\succeq 0, u\geq 0, x} \quad w\cdot u$$

$$\text{s.t.} \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}\cdot Z = 1,$$

$$\begin{bmatrix} 0 & 0.5 \\ 0.5 & 0 \end{bmatrix}\cdot Z - \begin{bmatrix} 1 \\ 0_p \end{bmatrix}\cdot x = 0,$$

$$d_{i,N+1} - K_N(i, i) - \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}\cdot Z$$

$$+ \Sigma_{i\cdot}\cdot x + e_{t,r}\cdot u = 0, \ \forall_{r=1,2,\ldots,t},$$

$$Z \in S_2, x \in Q_{p+1}, u \in P_{2t},$$

where $i = i(r)$ as before. Note that the constraints on Z ensure that it has the form of Eq. **7**.

Choosing Elements of Ω. If the dissimilarity information is symmetric (i.e., $d_{ij} = d_{ji}$), we can choose Ω to be the subset of $\{(i, j) : i < j\}$ for which information is available. However, the codes we use for solving the formulation in Eq. **4** (14, 15) require $O(m^2)$ storage (where m is the size of Ω), which is prohibitive for the application we describe in *Protein Clustering and Visualization with RKE*. Hence, we define Ω by randomly selecting a subset of the available dissimilarity information in a way that ensures that each object i appears with roughly the same frequency among the (i, j) pairs of Ω. Specifically, for each i, we choose a fixed number k of pairs (i, j) with $j \neq i$ (we call the objects j "buddies" of i) and add either (i, j) or (j, i) to the set Ω, reordering if necessary to ensure that the first index of the pair is smaller than the second. [It is possible that (j, i) has been placed in Ω at an earlier stage.] We choose the parameter k sufficiently large that the solution of Eq. **4** does not vary noticeably with different random subsets.

The newbie formulation in Eq. **6** is comparatively inexpensive to solve, so we take Ψ to be the complete set of objects for which dissimilarity information $d_{i,N+1}$ is available.

Eigenanalysis, Tuning, Truncation. The left five images of Fig. 1 illustrate the effect of varying λ on the eigenvalues of the regularized estimate of K obtained by solving Eq. **4**. The data are from the example to be discussed in *Protein Clustering and Visualization with RKE*, with $N = 280$ objects and $k = 55$ buddies for each of the N objects. Note that the vertical scale is in units of $\log_{10} \lambda$. As λ increases, the smaller eigenvalues begin to shrink, although in this example there is a very broad range of values of λ, spanning several orders of magnitude, where the sensitivity to λ is barely visible. At $\lambda = 10^{-8}$, the condition number of K is $\approx 10^3$. As λ goes much past 200 in this example, the penalty on K dominates, and the dissimilarity information in the data is suppressed.

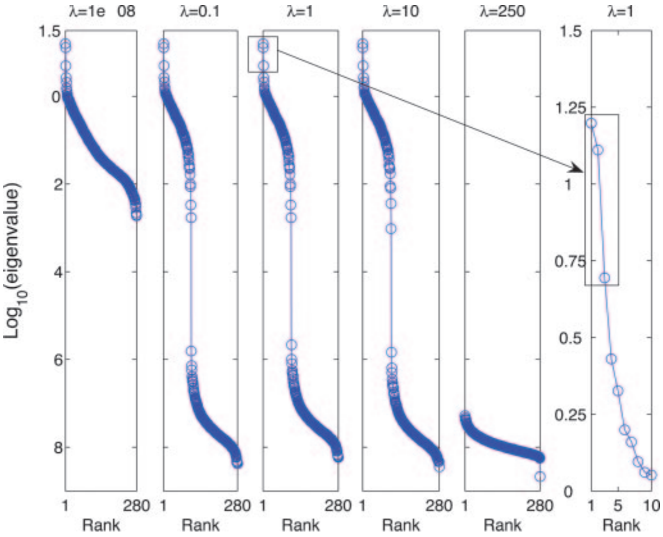

Fig. 1. The effect of varying λ on the eigenvalues of the regularized estimate of K. The left five images show log-scale eigensequence plots for five values of λ. As λ increases, smaller eigenvalues begin to shrink. The rightmost image shows the first 10 eigenvalues of the $\lambda = 1$ case displayed on a larger scale.

It is desirable to have a kernel with rank as low as possible while still respecting the data to an appropriate degree. Even if the rank of the regularized kernel estimate is not low, a low rank approximation obtained by setting all but a relatively small number of the largest eigenvalues to zero might retain enough information to provide an efficient way of doing classification or clustering.

In the work described here, as well as in various simulation studies, we started with a very small positive λ, increased λ in a coarse log scale, and then experimented with retaining various numbers of eigenvalues to get a low rank approximation to K. The rightmost image in Fig. 1 shows the first 10 eigenvalues for the $\lambda = 1$ case in an expanded log scale. Natural breaks appear after both the second and the third eigenvalues. Setting all of the eigenvalues of K after the largest p to 0 results in the νth coordinates of the jth object as $x_j(\nu) = \sqrt{\lambda_\nu}\phi_\nu(j)$, $\nu = 1, 2, \ldots, p$, where the λ_ν, ϕ_ν are the first p eigenvalues and eigenvectors of K and $\phi_\nu(j)$ is the j component of ϕ_ν. We remark that the coordinates of the N objects are always centered at the origin because the RKE estimate of K always has the constant vector as a 0 eigenvector. In the example discussed in *Protein Clustering and Visualization with RKE* below with four classes of labeled objects, the choice of $\lambda = 1$ and $p = 3$ provided plots with a clear, informative clustering on the labels, that was verified from the science of the subject matter. We note that using the estimated K or a low-rank version of it as the kernel in an SVM will result in linear classification boundaries in the object coordinates, [piecewise linear in the case of the multicategory SVM (MSVM) of ref. 16]. It will be seen in the plots for labeled objects in *Protein Clustering and Visualization with RKE* that piecewise linear classification boundaries in $p = 3$ coordinates would apparently do quite well. However, that will not necessarily always be the case, and a more flexible workhorse kernel in the p object coordinates can be used. The MSVM (16) comes with a cross-validation based method for choosing the MSVM tuning parameter(s) in a labeled training set. In principle, the parameters λ and p here can be incorporated in that method or other related methods as additional MSVM parameters. Further examination of principled methods of choosing λ and p along with the MSVM tuning parameter(s) is needed.

Protein Clustering and Visualization with RKE

Background. One of the challenging problems of contemporary biology is inferring molecular functions of unannotated proteins. A widely used successful method of protein function prediction is based on sequence similarity. Statistically significant sequence similarity, which is typically based on a pairwise alignment score between two proteins, forms the basis for inferring the same function. Two major related problems exist for predicting function from sequence. The first problem is the clustering of a large number of unlabeled protein sequences into subfamilies for the purpose of easing database searches and grouping similar proteins together. The second problem is concerned with assigning new unannotated proteins to the closest class, given the labeled or clustered training data. A substantial amount of literature exists for addressing these two problems. In particular, ref. 17 employs profile hidden Markov models (HMMs) for both problems. Clustering of proteins is obtained by a mixture of profile HMMs, whereas assignment of new protein sequences to the clusters/classes is based on the likelihood of the new sequence under each of the cluster-specific HMMs. Later, ref. 18 addresses the second problem first by obtaining an explicit vector of features (Fisher scores) for each protein sequence and then by using a variant of SVMs, based on a kernel called the Fisher kernel for classification purposes. The feature vector for each protein sequence is based on the likelihood scores of the input sequence evaluated at the corresponding maximum likelihood estimates of the HMM parameters fitted on the training data. More recently, ref. 19 similarly uses SVMs for protein classification. However, in contrast to obtaining a feature vector by likelihood scores, they define a feature vector for each protein sequence as a vector of its pairwise sequence similarity scores to all other proteins. Alternatively, ref. 20 represents protein sequences as vectors in a high-dimensional feature space by using a string-based feature map and train an SVM based on these vectors by using a mismatch kernel. These latter works clearly illustrate the advantage of representing each protein sequence by a high-dimensional feature vector in some coordinate system and the power of kernel methods for protein classification. The RKE methodology presented here provides an efficient way to represent each protein sequence by a feature vector in a chosen coordinate system using the pairwise dissimilarity between protein sequences.

Data. We illustrate the utility of RKE methodology by using a data set of globins that was first analyzed in ref. 17 by a profile HMM approach. The data set, distributed with the HMMER2 software package (21), has a total of 630 globin sequences. The globin family is a large family of heme-containing proteins with many subfamilies. It is mainly involved in binding and/or transportation of oxygen. For illustration purposes, we randomly choose 280 sequences from these data so that three large subclasses of the globin family (α-chains, β-chains, myoglobins) are included along with a heterogeneous class containing various types of chains. This selection resulted in a total of 112 "α-globins," 101 "β-globins," 40 "myoglobins," and 27 "globins" (the heterogeneous class), according to the SwissProt database annotation (25). The proportion of sequences in each class was taken to be proportional to the class sizes in the original data set.

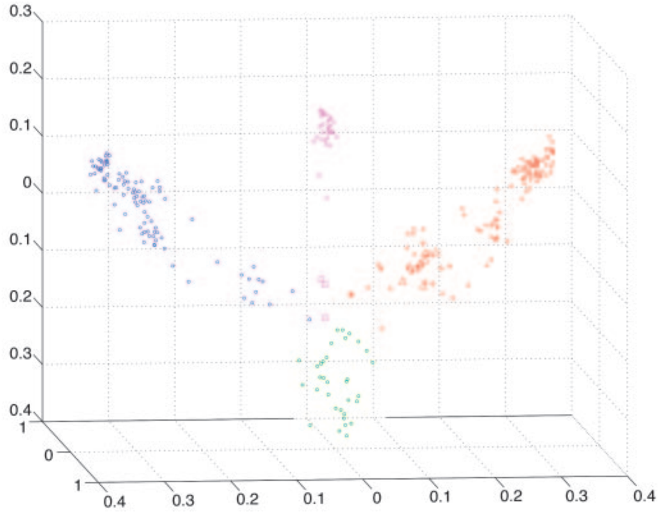

Fig. 2. A 3D representation of the sequence space for 280 proteins from the globin family. Different subfamilies are encoded with different colors: Red symbols are α-globin subfamily, blue symbols are β-globins, purple symbols represent myoglobin subfamily, and green symbols, scattered in the middle, are a heterogeneous group encompassing proteins from other small subfamilies within the globin family. Here, hemoglobin ζ-chains are represented by the symbol +, fish myoglobins are marked by □, and the diverged α-globin HBAM_RANCA is shown by $\ast$. Hemoglobin α-D chains, embedded within the α-globin cluster, are highlighted by the symbol △.

Implementation of RKE. We used the RKE formulation of *RKE with* l_1 *Loss* for this application. The BIOCONDUCTOR package PAIRSEQSIM (22) was used to obtain global pairwise alignment scores for all pairs of $N = 280$ sequences. This procedure gave a total of $N(N - 1)/2 = 39{,}060$ similarity scores, which we then normalized to map into the interval [0, 1]. We used one minus each of these numbers as the dissimilarity measure for each pair of sequences. During this process, alignment parameters were taken to be equal to the BLAST server (23) defaults. To construct the active index pair set Ω, we used the procedure described in *Choosing Elements of* Ω with $k = 55$ randomly chosen buddies for each protein sequence. The set Ω thus contained $\approx$14,000 sequence pairs, corresponding to $\approx$36% of the size of the complete index set. Replicated runs with $k = 55$ proved to be nearly indistinguishable, as judged by examination of eigenvalue and 3D plots and the measure: $\Sigma_{i<j}|\hat{d}_{ij1} - \hat{d}_{ij2}|/\Sigma_{i<j}(1/2)(\hat{d}_{ij1} + \hat{d}_{ij2})$, where the third subscript in $\hat{d}_{ijk}$ indexes different replicates (the above measure is typically $\approx 5\%$ for each pairwise comparison). The tuning parameter λ is set to 1 in the plots that follow later in this section.

Visualization of the Globin Sequence Space and Results. Fig. 2 displays the 3D representation of the sequence space of 280 globins. This figure shows that the first three coordinates of the protein sequence space, corresponding to three largest eigenvalues, is already quite informative. The four main classes of the globin family are visually identifiable: The four colors red, blue, purple, and green represent α-globins, β-globins, myoglobins, and globins, respectively.

Further investigation of this 3D plot reveals several interesting results. First, we observe that the five hemoglobin ζ-chains, namely HBAZ_HORSE, HBAZ_HUMAN, HBAZ_MOUSE, HBAZ_PANTR, and HBAZ_PIG, shown by +, are located close to each other and are embedded within the α-globin cluster. ζ-Globin chains are α-like polypeptides and are synthesized in the yolk sac of the early embryo. It is well known that human ζ-globin polypeptide is more closely related to other mammalian embryonic α-like globins (i.e., ζ-globins) than to human α-globins (24). Furthermore, the ζ-globin gene in humans is a member of the α-globin gene cluster. Second we note that HBAM_RANCA, which is represented by * and is a hemoglobin α-type chain, seems to be isolated from the rest of the α-globin sequences. A possible explanation might be found in the structure of this protein. Ref. 26 notes that the gene encoding this protein appeared through a gene duplication of hemoglobin, which took place near the time of the duplication that generated the α- and β-chains. Our third observation is that the myoglobins MYG_MUSAN, MYG_THUAL, and MYG_GALJA, denoted by open squares, which are all fish myoglobins [*Mustelus antarcticus* (Gummy shark), *Thunnus albacares* (Yellowfin tuna), and *Galeorhinus japonicus* (shark)], appear to be slightly separated from the rest of the myoglobin cluster. This observation is quite remarkable because fish myoglobins are known to be structurally distinct from the mammalian myoglobins (27), and the RKE method nicely highlights this distinction on the basis of primary sequence data only. The 3D plot also reveals subclusters in the α-globin cluster. For example, all of the 10 hemoglobin α-D chains (shown by open triangles in Fig. 2) are clustered together within the α-globin cluster.

In a recent work (28), the authors provided a 3D plot of the protein structure space of 1,898 chains. These authors used multidimensional scaling to project protein structures to a lower-dimensional space based on the pairwise structural dissimilarity scores derived from 3D structures of proteins. Our application of RKE to the globin family, which is a few levels down from the top level of the protein structure hierarchy considered by ref. 28, provide an analogous 3D plot for the sequence space of the globin family. It is quite encouraging that subprotein domains of this family are readily distinguishable from the 3D embedding of the protein sequences. It is also worth mentioning that our current application is concerned only with pairwise sequence similarity, which can be obtained efficiently. However, clustering at the higher levels of the protein structure hierarchy is known to benefit enormously from 3D structural similarities (see *Discussion* for details).

Classification of New Protein Sequences. We next illustrate how the newbie algorithm can be used to visualize unannotated protein sequences in the coordinate space of training data obtained by RKE. We used the following sequences as our test data: (*i*) HBAZ_CAPHI, hemoglobin ζ-chain from goat *Capra hircus*; and (*ii*) HBT_PIG, hemoglobin θ-chain from pig *Sus scrofa*. Fig. 3 displays the positions of these two test sequences with respect to 280 training sequences. We observe that HBAZ_CAPHI (filled circle) clusters nicely with the rest of the hemoglobin ζ-chains, whereas HBT_PIG (filled star), which is an embryonic β-type chain, is located closer to β-globins. Additionally, we also used 17 leghemoglobins (filled triangles) as test data and found that these cluster tightly within the heterogeneous globin group. This observation is consistent with the results of ref. 17, whose authors also found a heterogeneous globin cluster with a tight subclass of leghemoglobins among their seven clusters obtained by a mixture of HMMs. These results indicate that RKE together with newbie algorithm provide a powerful means for clustering and classifying proteins.

Discussion

We have described a framework for estimating a regularized kernel (RKE methodology) from general dissimilarity information by means of the solution of a convex cone optimization problem. We have presented an application of the RKE methodology (including the newbie algorithm) to homology detection in the globin family of proteins. The most striking result here is perhaps the fact that a simple 3D plot is sufficient for visual identification of the subfamily information. However, in other applications, the plot coordinates (or higher-dimensional coordinate vectors obtained by retaining more eigenvalues) may be used to build an automatic classification algorithm by means of the (principled) MSVM (16). That algorithm comes with a tuning method; it partitions the attribute space into regions for each training category, and it also comes with a method for signaling "none of the above." Multicategory penalized likelihood estimates also may be used if there is substantial overlap of the data from different classes (10, 29–31).

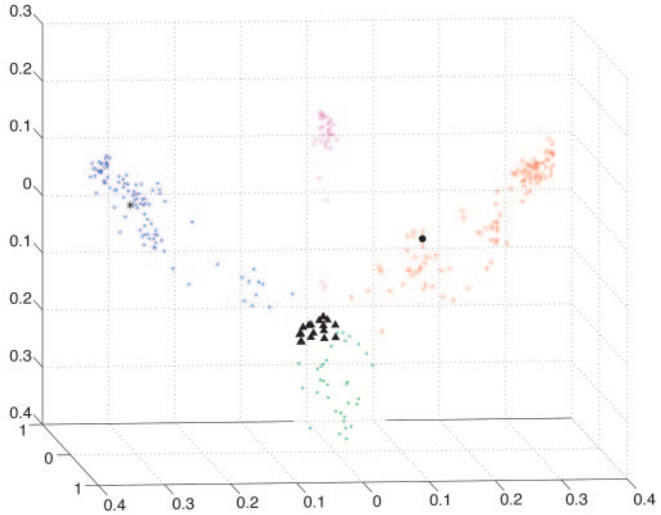

Fig. 3. Positioning test globin sequences in the coordinate system of 280 training sequences from the globin family. The newbie algorithm is used to locate 1 hemoglobin ζ-chain (filled circle), 1 hemoglobin θ-chain (filled star), and 17 leghemoglobins (filled triangles) into the coordinate system of the training globin sequence data.

A much more difficult problem in the context of protein classification and clustering is remote homology detection, that is, detecting homology in the presence of low sequence similarity. Because our framework accommodates an arbitrary notion of dissimilarities, we can easily take advantage of various types of dissimilarities such as presence or absence of discrete sequence motifs (32) and dissimilarities based on the primary, secondary, and tertiary structure (33) and obtain optimal kernels from each piece of information or data set. Without using labeled training sets, relations between a pair of kernels from different sources of information (or their lower rank approximations) can be quantified in various ways. A simple example is a measure of correlation: $\Sigma_{ij}\hat{d}^{s/2}_{ij\alpha}\hat{d}^{s/2}_{ij\beta}/((\Sigma_{ij}\hat{d}^{s}_{ij\alpha})^{1/2}(\Sigma_{ij}\hat{d}^{s}_{ij\beta})^{1/2})$, where α and β index the different sources of information and s is a real number to be chosen. With labeled data, these kernels can be examined further and combined in an optimal way, as, for example, in ref. 11, in the context of classification. As emphasized above, a striking feature of the presented methodology is the fact that it can exploit any type of dissimilarity measure and data sets with missing information. These properties are clearly beneficial in biological data analysis, because many biologically relevant dissimilarities may not naturally result in positive semidefinite kernels (pairwise alignment scores, for example), which are essential for powerful classification methods such as SVMs.

Homology detection is one type of computational biology problem for which our framework offers rigorous, flexible tools. However, there are many other computational biology applications that can naturally be handled within this framework. Clustering of transcription factor position weight matrices (binding profiles) is one such application. With the increasing growth of transcription factor-binding site databases, such as ref. 34, a need for characterizing the space of DNA-binding profiles and for developing tools to identify the class of newly estimated/studied profiles is emerging. A characterization of all available experimentally verified binding profiles such as in ref. 34 might provide invaluable information regarding the possible class of binding profiles. Such information then can be used in supervised motif-finding methods such as ref. 35. A natural dissimilarity measure for binding profiles is the Kullback–Leibler divergence. Clustering of the experimentally verified binding profiles based on a regularized kernel estimate of such dissimilarity measure might group binding profiles in a way that is consistent with the DNA binding domains of the transcription factors. We envision that this technique might generate a "protein binding profile space," as the work of ref. 28 generated a "protein structure space."

Potential topics for further exploration include both extension of the methodology and extension of the applications; in biology, the clustering of proteins at the top level of the protein hierarchy; and in other contexts, medical images in particular; other choices of loss and penalty functionals in the noisy manifold unfolding problem; and examining the properties of alternatives provided here and their application in other contexts.

G.W. thanks Richard Gill, S. V. N. Vishwanathan, and David Callan for helpful conversations. This work was partly supported by National Science Foundation Grant DMS0072292 and National Institutes of Health Grant EY09946 (to F.L. and G.W.); a Wisconsin Alumni Research Foundation start-up grant from University of Wisconsin at Madison (to S.K.); National Science Foundation Grants ATM-0296033, CNS-0127857, CCF-0113051, ANI-0330538, DMS-0427689, and CCF-0430504 (to S.J.W.); and Department of Energy Grant DE-FG02-04ER25627 (to S.J.W.).

1. Aronszajn, N. (1950) *Trans. Am. Math. Soc.* **68,** 337–404.
2. Kimeldorf, G. & Wahba, G. (1971) *J. Math. Anal. Appl.* **33,** 82–95.
3. Wahba, G. (1990) *Spline Models for Observational Data*, CBMS-NSF Regional Conference Series in Applied Mathematics (Soc. for Industrial and Applied Mathematics, Philadelphia), Vol. 59.
4. Wahba, G. (1977) *SIAM J. Numer. Anal.* **14,** 651–667.
5. Wahba, G. (1999) in *Advances in Kernel Methods: Support Vector Learning*, eds. Schölkopf, B., Burges, C. & Smola, A. (MIT Press, Cambridge, MA), pp. 69–88.
6. Evgeniou, T., Pontil, M. & Poggio, T. (2000) *Adv. Comput. Math.* **13,** 1–50.
7. Cristianini, N. & Shawe-Taylor, J. (2000) *An Introduction to Support Vector Machines* (Cambridge Univ. Press, Cambridge, U.K.).
8. Schölkopf, B. & Smola, A. J. (2002) *Learning with Kernels* (MIT Press, Cambridge, MA).
9. Schölkopf, B., Tsuda, K. & Vert, J-P. (2004) *Kernel Methods in Computational Biology* (MIT Press, Cambridge, MA).
10. Wahba, G. (2002) *Proc. Natl. Acad. Sci. USA* **99,** 16524–16530.
11. Lanckriet, G., Cristianini, N., Bartlett, P., Ghaoui, L. E. & Jordan, M. (2004) *J. Mach. Learn. Res.* **5,** 27–72.
12. Shawe-Taylor, J. & Cristianini, N. (2004) *Kernel Methods for Pattern Analysis* (Cambridge Univ. Press, Cambridge, U.K.).
13. Buja, A. & Swayne, D. (2002) *J. Classification* **19,** 7–43.
14. Tütüncü, R. H., Toh, K. C. & Todd, M. J. (2003) *Math. Progr.* **95,** 189–217.
15. Benson, S. J. & Ye, Y. (2004) DSDP5: *A Software Package Implementing the Dual-Scaling Algorithm for Semidefinite Programming* (Argonne National Laboratory, Argonne, IL), Technical Report ANL/MCS-TM-255.
16. Lee, Y., Lin, Y. & Wahba, G. (2004) *J. Am. Stat. Assoc.* **99,** 67–81.
17. Krogh, A., Brown, M., Mian, I. S., Sjölander, K. & Haussler, D. (1994) *J. Mol. Biol.* **235,** 1501–1531.
18. Jaakkola, T., Diekhans, M. & Haussler, D. (2000) *J. Comput. Biol.* **7,** 95–114.
19. Liao, L. & Noble, W. S. (2003) *J. Comput. Biol.* **10,** 857–868.
20. Leslie, C., Eskin, E., Cohen, A., Weston, J. & Noble, W. S. (2004) *Bioinformatics* **20,** 467–476.
21. Eddy, S. R. (1998) *Bioinformatics* **14,** 755–763.
22. Gentleman, R. C., Carey, V. J., Bates, D. J., Bolstad, B. M., Dettling, M., Dudoit, S., Ellis, B., Gautier, L., Ge, Y., Gentry, J., *et al.* (2004) *Genome Biol.* **5,** R80.
23. Atschul, S. F., Gish, W., Miller, W., Myers, E. W. & Lipman, D. (1990) *J. Mol. Biol.* **215,** 403–410.
24. Clegg, J. B. & Gagnon, J. (1981) *Proc. Natl. Acad. Sci. USA* **78,** 6076–6080.
25. Gasteiger, E., Gattiker, A., Hoogland, C., Ivanyi, I., Appel, R. D. & Bairoch, A. (2003) *Nucleic Acids Res.* **31,** 3784–3788.
26. Maeda, N. & Fitch, W. M. (1982) *J. Biol. Chem.* **257,** 2806–2815.
27. Cashon, R., Vayda, M. & Sidell, B. D. (1997) *Comp. Biochem. Physiol. B* **117,** 613–620.
28. Hou, J., Jun, S., Zhang, C. & Kim, S. (2005) *Proc. Natl. Acad. Sci. USA* **102,** 3651–3656.
29. Lin, X. (1998) Ph.D. thesis (Univ. of Wisconsin, Madison, WI); Tech. Rep. 1003 (Department of Statistics, Univ. of Wisconsin, Madison, WI).
30. Wahba, G., Gu, C., Wang, Y. & Chappell, R. (1995) in *The Mathematics of Generalization, Santa Fe Institute Studies in the Sciences of Complexity Proceedings*, ed. Wolpert, D. (Addison–Wesley, Reading, MA), Vol. XX, pp. 329–360.
31. Zhu, J. & Hastie, T. (2004) *Biostatistics* **5,** 329–340.
32. Ben-Hur, A. & Brutlag, D. (2003) *Bioinformatics* **19,** i26–i33.
33. Tang, C. L., Xie, L., Koh, I. Y. Y., Posy, S., Alexov, E. & Honig, B. (2003) *J. Mol. Biol.* **334,** 1043–1062.
34. Sandelin, A., Alkema, W., Engström, P., Wasserman, W. & Lenhard, B. (2004) *Nucleic Acids Res.* **32,** D91–D94.
35. Keleş, S., van der Laan, M. J., Dudoit, S., Xing, B. & Eisen, M. B. (2003) *Stat. Appl. Genet. Mol. Biol.* **2,** 5.