Genomic Similarity and Kernel Methods I: Advancements by Building on Mathematical and Statistical Foundations

Daniel J. Schaid

Division of Biomedical Statistics and Informatics, Mayo Clinic, Rochester, Minn., USA

Key Words

Distance • Eigenvalue decomposition • Nonparametric regression • Regularization • Similarity kernel • Support vector machine

Abstract

Measures of genomic similarity are the basis of many statistical analytic methods. We review the mathematical and statistical basis of similarity methods, particularly based on kernel methods. A kernel function converts information for a pair of subjects to a quantitative value representing either similarity (larger values meaning more similar) or distance (smaller values meaning more similar), with the requirement that it must create a positive semidefinite matrix when applied to all pairs of subjects. This review emphasizes the wide range of statistical methods and software that can be used when similarity is based on kernel methods, such as nonparametric regression, linear mixed models and generalized linear mixed models, hierarchical models, score statistics, and support vector machines. The mathematical rigor for these methods is summarized, as is the mathematical framework for making kernels. This review provides a framework to move from intuitive and heuristic approaches to define genomic similarities to more rigorous methods that can take

advantage of powerful statistical modeling and existing software. A companion paper reviews novel approaches to creating kernels that might be useful for genomic analyses, providing insights with examples [1].

Copyright © 2010 S. Karger AG, Basel

Introduction

Over the past few years there have been numerous papers that present flexible regression methods and score statistics to evaluate the association of traits with genetic information among subjects, all connected by a common underlying theme that uses a matrix of genetic similarities for all pairs of subjects [2–9]. The association of a genetic similarity matrix with trait information is often evaluated in a regression framework, typically by variance component methods that treat the similarity matrix as a general covariance structure.

Fortunately, similarity matrices are the basis of a wide array of flexible statistical models, including nonparametric regression, smoothing spline models, and support vector machines (SVMs). The purpose of this paper is to provide a synthesis of developments that are relevant to genetic studies. Most of this exposition uses single-nucle-

otide polymorphisms (SNPs) for illustrative purposes. It is important to recognize, however, that a strength of these types of similarity methods is their ability to integrate different types of data, for example SNPs, gene expression, and protein measures, as well as epidemiological data and environmental exposure measurements. We review the statistical foundation of similarity measures, their role in penalized loss functions, and a variety of ways to create similarity measures. The main strengths and limitations of using genetic similarities to evaluate the association of genetic variation with trait variation are discussed, while highlighting both the enormous expressive power of similarity 'kernels' and areas for future research.

Genomic Similarity in Genetic Epidemiology

Measures of genetic similarity are the foundation of many statistical genetic analyses, ranging from measures of identity-by-descent (IBD) for pairs of relatives from pedigrees [10–12], to measures of IBD for pairs of subjects from a population [13], to sharing scores for pairs of haplotypes (e.g. haplotype sharing methods), to more general measures of similarity for pairs of subjects based on genetic markers or gene expression assays. Yet, the approaches to devise measures of similarity are often ad hoc and unstructured. With the high-density SNP arrays currently available, the large-scale arrays for gene expression, and the anticipated arrival of DNA sequence data for a large number of subjects, measures of genetic similarity for complex data types will likely play a greater role in genetic analyses. To illustrate a variety of ways similarity measures have been created for genetic epidemiology studies, a brief review is provided, which also motivates attention to the mathematical basis of similarity measures.

Some examples of similarity measures for haplotype sharing are: the fraction of alleles alike-in-state (AIS) between two aligned haplotypes [14, 15], the length of haplotype similarity around a given marker [16–19], measures that down-weight matches on common alleles [20, 21], and measures that allow for a few mismatches due to mutation or genotyping errors [22]. Nonlinear sharing measures have been proposed, such as the hyperbolic cosine function of the haplotype frequencies to create a weighted haplotype dissimilarity score [23]. Wessel and Schork [6] considered similarity for pairs of subjects according to a variety of measures: AIS of alleles, with some scores weighted by allele frequencies (rare alleles increas-

ing likelihood of IBD), some weighted by the likelihood that the genetic variants are functional, some weighted by nucleotide conservation across species, and some weighted according to statistics for single-locus associations with a trait. They also considered measures of similarity based on haplotype similarity (weighted or unweighted), or based on ancestry determined by phylogenetic analyses. Such a wide array of similarity measures illustrates ample creative opportunities.

Simple measures of similarity based on genetic markers (i.e. ignoring haplotype phase) have been widely used, such as the fraction of matching genotypes across all markers. Yet, the range of possible similarity measures could be broader. One could select from many different similarity measures for categorical variables. For example, Selinski and colleagues [24–26] evaluated the utility of a variety of similarity measures for categorical data to cluster SNPs (i.e. similarity for pairs of SNPs, not pairs of subjects); others have considered such comparisons for data mining algorithms [27]. In a parallel approach, measures of similarity between groups of subjects, based on similarities of group-specific allele frequencies, have been widely used to classify taxonomic units [28]. These types of measures could be adapted to measure similarity between pairs of subjects, but with caution, as we emphasize later.

To illustrate a novel strategy to measure similarity for pairs of subjects according to uninterrupted stretches of SNP genotype matches, consider the scoring proposed by Gonzalez-Recio et al. [29]. They took advantage of the ordered sequence of SNP genotypes on chromosomes to create dissimilarity scores for subsequences of genotypes; contiguous genotype matches receive small dissimilarities, while interruptions by nonmatched genotypes increase dissimilarities. Furthermore, smaller dissimilarities were given to matches of rare genotypes. This score, based on a dynamic programming algorithm, is similar to that used for matching amino acid sequences [30]. They then converted from dissimilarity (d) to similarity, $s = \exp(-d)$.

The above examples illustrate a variety of ways to construct a similarity matrix for all pairs of subjects. The resulting similarity matrix can be used in a variety of ways. One way is to visualize structure in the data, such as by multidimensional scaling. This approach extracts a lower-dimensional space from the matrix, often for plotting in two or three dimensions. In contrast to factor analysis, which extracts lower dimensions from the correlation matrix, multidimensional scaling works directly with a similarity matrix for pairs of subjects. Cluster analysis is

another approach that uses similarity (or dissimilarity) to classify subjects into different groups. This is achieved by partitioning a data set into subsets, so that the data in each subset are close to each other according to the defined similarity or dissimilarity measure. Although there are a variety of methods to create clusters from a similarity matrix [31], an approach relevant to our general discussion is based on spectral graph theory. This approach transforms the similarity matrix to a new matrix, such as dividing the elements of the similarity matrix by their marginal row sums. This standardization provides the interpretation of transition probabilities – the probability of moving from one cluster to another. Clusters can then be calculated by using the spectral decomposition of the new matrix to extract a small number of eigenvectors upon which clustering methods can be applied, such as k-means clustering [32]. These clustering methods can be useful to screen for outliers, or to cluster subjects into homogeneous groups to control for population confounding from ethnic differences (e.g. population stratification; [33]).

Measures of genetic similarity are often used to evaluate the association of genetic similarity with trait similarity. For relatives, IBD is the foundation of genetic linkage analyses. For unrelated subjects, a variety of measures of genetic similarity have been used to evaluate the association of genetic similarity with trait similarity, based on U-statistics [34] or analogous methods [35, 36], or regression models [6–9, 37–40].

Although many different measures of similarity or dissimilarity have been used for analysis of genetic data, most have been ad hoc with little attention to whether the resulting measures preserve mathematical notions of distance, and whether they can be used in standard statistical analyses (e.g. nonparametric regression, SVMs, etc.). Our focus will be on kernel methods. A kernel function converts information for a pair of subjects to a quantitative value representing either similarity (larger values meaning more similar) or distance (smaller values meaning more similar), with the requirement that the function must create a positive semidefinite (psd) kernel matrix when applied to all pairs of subjects. If K is a matrix and c is a vector, then K is psd if $c'Kc \geq 0$ for all $c \neq 0$. Furthermore, a psd matrix has only nonnegative eigenvalues. This requirement of a symmetric psd similarity matrix is key to its use in many statistical analyses. Fortunately, a rigorous developmental approach is provided by the large body of literature on kernel methods and on how they relate to similarity, dissimilarity, and distance [41–44]. In a nutshell, a kernel matrix is a symmetric psd matrix,

most often based on a measure of similarity. Examples of kernel functions, and how to construct them, will be discussed after we illustrate how general kernel matrices can be used in statistical models and score statistics.

Concepts and Utility of Similarity, Dissimilarity, and Distance

Because kernel matrices are based on measures of similarity between pairs of subjects according to their measured attributes, it is worthwhile to review notions of similarity, dissimilarity, and distance. A measure of similarity for a pair of subjects has the obvious notion of an increasing value as the measured attributes become more alike, with dissimilarity going in the opposite direction. So, similarity (s) and dissimilarity (d) can be viewed as mapping from one to the other, such as first standardizing similarity to take values over $[0, 1]$, and then using $d = 1 - s$, or $d = \sqrt{1 - s}$, or $d = \sqrt{1 - s^2}$, depending on the similarity measure. We restrict distance, however, to a more formal mathematical structure. When measuring distances based on multiple attributes, different types of distances can be measured. The most familiar is euclidean distance: the shortest distance between two points is a straight line. The euclidean distance between two subjects with the attribute vectors x and y is

$$d(x, y) = \|x - y\| = \sqrt{\sum_{k=1}^p (x_k - y_k)^2},$$

also called the L_2 norm. In contrast, the ‘Manhattan’ distance is

$$d(x, y) = \sum_{k=1}^p |x_k - y_k|,$$

also called the L_1 norm. This ‘city-block’ distance represents the total distance traveled on streets laid out in a grid (no diagonal shortcuts that the euclidean traveler is allowed to take).

A general concept of ‘metric’ distance between two points can be characterized as satisfying four properties:

- nonnegativity: $d(x, y) \geq 0$,
- identity: $d(x, y) = 0$ only when x and y are completely identical,
- symmetry: $d(x, y) = d(y, x)$,
- triangle inequality: $d(x, z) \leq d(x, y) + d(y, z)$.

The first three properties are intuitively obvious. The triangle inequality is based on the geometry of a triangle with x , y , and z at the three vertices.

The choice of an appropriate distance often depends on the goals of a study. For example, it is common to use euclidean distances for SNPs. This is achieved by coding genotypes as 0, 1, or 2 for the number of minor alleles a subject possesses, and then measuring distances between pairs of subjects. These distances are often used to cluster subjects for plotting, or for principal components analyses to correct for population stratification [45, 46]. Recently, Manhattan distances have been successfully used with high-density SNPs to determine whether an index subject is likely to have its DNA sample in a complex genomic DNA mixture. The distance of the index to the mixture pool was compared with the distance of the index to a reference population [47]. Whether Manhattan distance is a better choice than euclidean distance is an interesting question.

A good example of the logic for choosing a particular distance metric is the preference for a type of Manhattan distance for ecological studies. Ecological studies typically measure the distance between two communities according to the number of different species in each community. A problem with euclidean distance for counts of species is that it depends on both the counts per species and how these counts vary across species. This can cause two communities with small counts to have a close distance, even if there is no overlap of the types of species. For this reason, the Bray-Curtis distance is preferred,

$$d_{ij} = (\sum_k |x_{ik} - x_{jk}|) / (\sum_k x_{ik} + x_{jk}),$$

where x_{ik} is the number of species of type k in community i . This distance, however, does not satisfy the triangle inequality, and can lead to negative eigenvalues of a distance matrix – a problem we later emphasize. Note, however, that using the absolute value, $|x_{ik} - x_{jk}|$, does not mean that all such distances will cause negative eigenvalues. A simple fix for the Bray-Curtis distance is to replace the counts of species with proportions such that the proportions sum to one in each community. Other fixes to convert nonmetric distances to metric distances are discussed in the section Making Similarity Kernels.

Dissimilarity measures that can be nonmetric are those that lead to non-zero values for identity, the diagonal values of the dissimilarity matrix. In a parallel vein, variation in the diagonal elements of a similarity matrix can be a problem, because rescaling the similarity matrix by its maximum value will not lead to diagonal values of complete identity. These types of nonmetric measures can cause difficulties for kernel methods. Gower [48] gives an extensive list of commonly used dissimilarity coefficients, and illustrates whether or not they satisfy the four metric properties.

From Linear Regression to Kernel Methods

The power of similarity matrices comes from the flexibility they provide for both regression models and ‘machine learning’ methods. Although regression and machine learning methods have much in common, different terms between the regression literature and the machine learning literature can cause confusion. In regression settings, the independent variable is often called a ‘trait’; in machine learning, a binary group variable is called a ‘label’. In regression, variables used to predict the outcome are called ‘predictors’ or ‘covariates’; in machine learning, ‘attributes’. We will use ‘attributes’ for our exposition because it has a more generic connotation.

To understand how kernels can be used in regression, it helps to review key concepts in linear regression. Although a quantitative trait is used for illustration, similar ideas follow for generalized linear models (e.g. logistic regression for binary traits). Let y_i denote a measured trait for the i -th subject and let x_i denote a set of attributes. The usual approach of parametric linear regression is to define functions that map the measured attributes into a regression design matrix. Let X denote a design matrix, with n rows for subjects and p columns that contain the results from applying the functions that map attributes to the design matrix. For example, a binary attribute might map to the design matrix with a function that has values of 0 or 1; another function might map a quantitative attribute to a linear effect, $f(x) = x$; another function to a quadratic effect, $f(x) = x^2$, etc. These mapping functions are called basis functions, because they are ultimately linearly combined, using the vector of regression coefficients, β , as weights to compute expected values: $E[y_i] = x_i' \beta$. To gain flexibility on how the attributes influence the trait, particularly nonlinear influences, spline functions are widely used. Spline functions are often piecewise polynomials where the pieces are joined by knots such that the resulting spline function is continuous, has a specified number of derivatives (determining its smoothness), and perhaps satisfies boundary conditions [49]. For now, assume that only a few knots are used, so that traditional statistical methods can be used to fit a regression model. Later we discuss how greater flexibility can be gained by linking spline-smoothing with kernels. To create more complicated models, additional basis functions could be used, such as the product of main effects to allow pair-wise interactions. The main point is that the attributes, x , can be mapped into a design matrix X by explicitly defined basis functions. The complexity of this approach grows as the dimension of the attributes grows,

or when the attributes themselves are so complex that it is difficult to specify basis functions to map attributes into a design matrix. This is where the kernel method can realize its benefits, by avoiding the need to explicitly define the basis functions.

Before we leap to the kernel method, it is worthwhile to review the role of euclidean distance in the design matrix, and its dual information for distance between pairs of columns versus distance between pairs of rows. Consider the linear regression model $Y = Ab + e$, where Y is a vector of length n , the matrix A is $n \times m$, the regression coefficient vector b is of length m , and the residual error vector e has the length n . Although linear regression typically specifies A to be a design matrix, with columns as the various predictor variables, at this point we have not specified what A is, but rather leave it somewhat general. The goal is to find a coefficient vector b that minimizes the residual sums of squares, $\sum_i (y_i - a'_i b)^2$, where a'_i is the i -th row of A (a' denotes the transpose of a). This minimization problem can be expressed as a loss function in vector notation, $loss(b) = (Y - Ab)'(Y - Ab)$. The least squares solution can be found by taking derivatives of $loss$ with respect to b , setting the result equal to zero, and then solving for b . This leads to the normal equations, $(A'A)b = A'Y$. In the usual formulation, A is the $n \times p$ design matrix X , and b is the regression vector β . This *primal* formulation leads to the least squares solution $\hat{\beta} = (X'X)^{-1} X'Y$, with fitted value $\hat{Y} = X(X'X)^{-1} X'Y$. Note that this primal formulation uses the $p \times p$ matrix $X'X$.

An alternative formulation – called the *dual* formulation – is to let $A = XX'$, an $n \times n$ matrix. In this case, we want to determine b that minimizes $loss(b) = (Y - XX'b)'(Y - XX'b)$. The least squares solution leads to $X'\hat{b} = \hat{\beta}$, with fitted value $\hat{Y} = XX'\hat{b} = X\hat{\beta}$. An important consequence is that whether we use the design matrix X in the primal formulation to estimate $\hat{\beta}$, or use XX' in the dual formulation to estimate $\hat{b}$, we get the same fitted values, and hence the same loss!

Some important characteristics of the primal versus dual formulations of the least squares solutions are worth emphasizing, because these ideas will help to understand the role of the kernel matrix used in statistical models or score statistics. First, suppose that the columns of X are centered about their means. Then, the $p \times p$ matrix $X'X$ (from the primal formulation) can provide a measure of the correlation between *pairs of attributes*, averaged over the n subjects. This type of matrix is sometimes referred to as an R matrix, and principal components analysis can be based on this matrix (for example, see the function *princomp* in the R software environment for statistical

computing). In contrast, the $n \times n$ matrix XX' provides a measure analogous to the correlation between *pairs of subjects*, averaged over the p covariates. This type of matrix, sometimes referred to as a Q matrix, can also be used in principal components analysis (for example, see the R function *prcomp*). Note that $X'X$ and XX' are both based on euclidean distances, the former for pairs of attributes and the latter for pairs of subjects. Importantly, these two matrices contain the same information in terms of their underlying vector spaces, emphasized by their eigenvalue decompositions:

- they are both symmetric with nonnegative eigenvalues (i.e. psd matrices);
- they have identical rank (i.e. the number of non-zero eigenvalues);
- the eigenvectors of one matrix can be obtained from the eigenvectors of the other by a simple linear mapping. To see this, recall the singular value decomposition of matrix $X = UDV'$, where D is a diagonal matrix of the singular values. Then, $X'X = VD^2V'$ and $XX' = UD^2U'$ (note that U and V are each orthonormal matrices, so that $U'U$ is an identity matrix, as is $V'V$). This illustrates that $X'X$ and XX' share the same non-zero eigenvalues, D^2 , and hence the same percentage of the covariance structure explained by the eigenvalues (i.e. the principal component explained variation).

To summarize, the dual formulation of ordinary least squares shows that it can be viewed as a kernel regression method where the kernel matrix that measures similarity for pairs of subjects is XX' . An advantage of this dual formulation is that it can be generalized by replacing XX' with a symmetric kernel matrix. This is referred to as the 'kernel trick': a dot product (e.g. XX') is replaced with a kernel matrix. The kernel function that creates a kernel matrix can be quite general, capturing nonlinear measures of similarity between pairs of subjects, but with the requirement that the kernel function results in a psd symmetric kernel matrix. As we later discuss in detail, a psd kernel matrix is implicitly defined in terms of dot products (euclidean distances between pairs of subjects), allowing computational algorithms to easily determine optimal solutions. This allows use of nonlinear kernels to measure similarities of pairs of subjects without having to specify the columns of a design matrix. To appreciate this trick, concepts about vector spaces and what happens when a continuous kernel function is used to determine the kernel matrix are reviewed in Appendix 1.

In summary, attributes are the direct measures on subjects, these attributes are fed into a kernel function to create a symmetric psd kernel matrix. Because the kernel

matrix is psd, we are assured that the underlying vector space of the matrix has nonnegative eigenvalues and corresponding eigenvectors, which means that the kernel matrix determines an underlying ‘feature space’ which has a distance property analogous to euclidean distances. The feature space allows optimization algorithms to rely on the underlying euclidean distances, even when the kernel matrix of similarities is a nonlinear mapping of the attributes to similarities.

We have emphasized the importance of nonnegative eigenvalues. An obvious question is: why should we avoid negative eigenvalues? First, euclidean distances no longer exist, so the notion that lesser distances imply greater similarities is violated. A subject can have zero distance to self, giving maximum similarity, yet lesser distance to other less similar subjects (due to negative squared euclidean distance). Second, eigenvalues are interpreted as the percentage of the variation explained by the corresponding eigenvectors, but a negative eigenvalue implying negative variance is not sensible. Third, there is no guarantee that the optimization algorithms that work for psd kernels will work when there are negative eigenvalues – the solution might be a local optima or a stationary point [50]. Finally, the mathematical foundation moves from real numbers to complex numbers.

Fitting Kernel Models by Regularization

The theory outlined in Appendix 1 regarding reproducing kernel Hilbert space (RKHS) provides a general and rigorous foundation to fit kernel models, where a model is comprised of a kernel matrix, a loss function, and a penalty function. Suppose a kernel is specified, and we want to estimate nonparametric functions, something like the form $y_i = f(x_i) + e_i$, where y_i is a trait, $f(x_i)$ a nonparametric function for attribute x_i , and e_i is random error. The number of parameters can be quite large, as large as n . To avoid over-fitting the parameter estimates to the analyzed data, additional information is introduced into the solution, such as assumptions about the smoothness of the solution. This smoothing process, known as regularization, is achieved by adding a penalty function to the loss function; minimizing a penalized loss function controls the model complexity. The loss and penalty functions can be quite general, and if we keep within the framework of RKHS, the representer theorem of Kimeldorf and Wahba [51] proves that the solution can be found by minimizing a penalized loss function of the form

$$\text{loss}(y_i, f(x_i); i = 1, \dots, n) + \text{penalty}(\|f\|_{H_K}^2),$$

where $\text{loss}(\cdot)$ is an arbitrary loss function, $\text{penalty}(\cdot)$ is a positive increasing penalty function, and $\|f\|_{H_K}^2$ is the squared-norm in RKHS ($\|f\|_{H_K}^2 = \sum f_i^2$). Restricting to a convex penalized loss function ensures a global minimum, emphasizing the importance of this theorem [44]. The penalized loss function can then be expressed according to a scalar penalty parameter, λ ,

$$\text{loss}(y_i, f(x_i); i = 1, \dots, n) + \lambda(\|f\|_{H_K}^2).$$

The scalar parameter λ is a tuning parameter that balances between how well the model fits the data and the complexity of the model. The representer theorem proves that the solution has the form

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

where x represents the function argument, x_i^* ($i = 1, \dots, n$) represents fixed values (e.g. in the training data) and the α_i 's are unknown parameters to be estimated.

Note that $f(x)$ is a linear combination of basis functions of the form $K(x_i^*, x)$; this basis function is known as the ‘representer of evaluation at x ’. The representer theorem shows that the fitted functions are $\hat{f} = K\hat{\alpha}$, where $\hat{\alpha}$ is the estimated coefficient vector that minimizes the penalized loss. Intuitively, this means that the nonparametric function for a new vector of attributes, x , is a weighted linear combination of the similarity of x with the attributes of all other subjects, with weights determined by the $\hat{\alpha}$ coefficients. For example, consider subject i with attribute vector x_i . If we plug x_i into $\hat{f}(x_i)$, and if $\hat{\alpha}_j = 0$, then this means the similarity of subject j with subject i does not influence the nonparametric function $\hat{f}(x_i)$. When the kernel method is used with traits, $\hat{\alpha}_j = 0$ means that the similarity of genomic information between subjects j and i does not influence the trait for subject i . Because there can be a large number of α 's to estimate, regularization by the penalty function is critical. As we shall see later, this is a strength of SVMs, which attempt to estimate a small set of α 's that have the largest impact on classification for a binary trait, or for regression for a quantitative trait.

These results are very powerful because they allow general loss functions and any psd kernel matrix, subsuming many statistical models. For example, if the kernel matrix is created from a design matrix as $K = XX'$, and the loss function for a quantitative trait is squared-error loss (i.e. the usual linear regression least-squares), then the resulting solution is the usual ridge regression. Commonly used loss and penalty functions for quantitative and binary traits are summarized in table 1. It illustrates

Table 1. Commonly used ‘loss + penalty’ functions^a

Trait	Model	Known as	Loss function	Penalty function ^b
Quantitative	linear fixed effects	ridge regression	squared-error $\ Y - X\beta\ _2^2$	L_2 $\lambda \ \beta \ _2^2$
		lasso regression	squared-error $\ Y - X\beta\ _2^2$	L_1 $\lambda \ \beta \ _1$
	linear mixed model	empirical Bayes	squared-error $\ Y - X\beta - Zu\ _2^2$	L_2 $\lambda \ u \ _2^2$
		lasso, random effects	squared-error $\ Y - X\beta - Zu\ _2^2$	L_1 $\lambda \ u \ _1$
	nonparametric	kernel regression	squared-error $\ Y - f(X)\ _2^2$	L_2 $\lambda \ f \ _{H_K}^2$
		support vector regression	L_1 ε -insensitive $\sum y_i - f(x_i) _\varepsilon$ where $ a _\varepsilon = \begin{cases} 0 & \text{if } a < \varepsilon \\ a - \varepsilon & \text{if } a \geq \varepsilon \end{cases}$ Larger ε results in less resolution and ‘simpler’ models	L_2 $\lambda \ f \ _{H_K}^2$
Binary ($y = \pm 1$)		support vector classification, soft margin, hinge-loss	soft margin $\sum sm(1 - y_i f(x_i))$ where soft margin is $sm(a) = \begin{cases} a & \text{if } a > 0 \\ 0 & \text{if } a \leq 0 \end{cases}$	L_2 $\lambda \ f \ _{H_K}^2$
		support vector classification, L_1 penalty [110]	soft margin $\sum sm(1 - y_i f(x_i))$	L_1 $\lambda \ f \ _{H_{K,1}}$
		support vector classification, hard margin	hard margin $\sum hm(1 - y_i f(x_i))$ where hard margin (Heaviside, unit step function) is $hm(a) = \begin{cases} 1 & \text{if } a > 0 \\ 0 & \text{if } a \leq 0 \end{cases}$	L_2 $\lambda \ f \ _{H_K}^2$
		support vector, misclassification loss	hard margin $\sum hm(-y_i f(x_i))$	L_2 $\lambda \ f \ _{H_K}^2$
	Logistic regression	support vector regression	Bernoulli loglikelihood $-\sum \log\{1 + \exp[-y_i f(x_i)]\}$	L_2 $\lambda \ f \ _{H_K}^2$
General trait, exponential family	generalized linear models	GLM nonparametric [70, 71]	negative log likelihood $-\sum \ln \text{like}[y_i, f(x_i)]$	L_2 $\lambda \ f \ _{H_K}^2$
		GLM lasso [72]	negative log likelihood $-\sum \ln \text{like}(y_i, x_i; \beta)$	L_1 $\lambda \ \beta \ _1$

^a If some model parameters are penalized and others not, the space for the penalized parameters must be independent from the space for the unpenalized parameters, achieved by projecting the unpenalized space onto the penalized space with the projection matrix P (e.g. the residuals from the unpenalized space are used in the penalized space). This is achieved, for example, by using the penalty function $\lambda \| Pf \|_{H_K}^2$.

^b L_1 -norm: $\|Z\|_1 = \sum |z_i|$; L_2 -norm squared: $\|Z\|_2^2 = \sum z_i^2$; $H_K = \text{RKHS}$. Many formulations multiply the penalty function by 1/2, but we absorb this factor into λ for ease of presentation.

examples of both L_1 and L_2 loss functions and L_1 and L_2 penalty functions, emphasizing the generality of the representer theorem and a wide array of methods that mix different types of loss and penalty functions.

A nice feature of the kernel method is that it provides a bayesian interpretation. For the L_2 penalty, the f functions arise from a gaussian process with mean zero and prior covariance function K , $f \sim N(0, \sigma_K^2 K)$, where σ_K^2 is the amount of variation of y explained by the kernel ma-

trix. For the L_1 penalty, $f \sim \text{Laplace}(0, \sigma_K^2 K)$, also called the double exponential distribution. The normal and Laplace density functions, with mean μ and scale σ , are illustrated below to emphasize the L_2 term $[(x - \mu)^2]$ and the L_1 term $(|x - \mu|)$:

$$N(\mu, \sigma^2): d(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{(x - \mu)^2}{2\sigma^2}\right\}$$

$$\text{Laplace}(\mu, \sigma^2): d(x) = \frac{1}{2\sigma} \exp\left\{-\frac{|x - \mu|}{\sigma}\right\}.$$

This bayesian perspective of the kernel matrix as a prior covariance structure can serve as a useful guide when making kernels for analyses. Some specific loss + penalty functions that have been used for genetic studies are discussed in subsequent sections.

The methods required to fit the penalized loss function depend on both the loss function and the penalty function. For quantitative traits with squared-error loss and the L_2 quadratic penalty, $\lambda \|f\|_{H_K}^2$, the familiar ridge and empirical Bayes methods result from a constrained system of linear equations, allowing the penalized loss function to be translated to a linear mixed model (LMM) formulation. This powerful result allows the use of mixed model software to estimate all parameters, including the penalty parameter λ (discussed below).

In contrast, the L_1 ε -insensitive-loss for quantitative traits, which ignores errors in the fitted values that are less than ε (called support vector regression), and the hinge-loss for binary traits (called support vector classification), are discontinuous functions. For a given value of the penalty parameter λ , the problem can be formulated as a constrained optimization problem; the constraints are handled by adding extra lagrangian ‘parameters’ that enforce the constraints. Fortunately, the fitted α values in expression (1) can be found by solving a quadratic programming problem, meaning that the solution leads to a global minimum of the penalized function; it is a significant advantage to know that the solution is not a local minimum.

The lasso-type of models that combine an L_2 loss function with an L_1 penalty function can be fit with the least angle regression algorithm that efficiently finds all possible solutions over all possible values of the penalty parameter [52, 53]. Remarkably faster methods based on coordinate descent have recently been developed [54–56], making large-scale analyses feasible.

To choose a value of the scalar penalty λ that minimizes the penalized loss function, group cross-validation can be used [53]. Alternatively, Akaike information criterion, Bayes information criterion, or generalized cross-validation [57] can be used. Generalized cross-validation numerically approximates the prediction error for cross-validation when one observation at a time is left out of the training set.

Developing methods for the L_1 penalty function, of which lasso regression [58] is an important special case, is a highly active area of research, too much to fully summarize here (see [59] or [53]). A few key points, however, are worth mentioning. Because the L_1 penalty is discontinuous, it can shrink regression coefficients to exactly

zero, providing a method to both shrink and select regression coefficients (in contrast to the L_2 penalty which only shrinks, but not to zero). A limitation of lasso, however, is that it gives the same penalty parameter to all regression coefficients. This can cause the true large coefficients to be shrunk too much. Furthermore, in the presence of several large coefficients, the penalty parameter cannot become too large, otherwise the true coefficients would be excluded. Yet, a smaller penalty can cause pure noise attributes to be included in the model. So, there can be a conflict between prediction (using cross-validation to estimate the shrinkage penalty parameter) and model selection. To overcome this, a number of modified penalty functions have been proposed, such as a ‘smoothly clipped absolute deviation’ [60], or an adaptive lasso that uses data-dependent weights to scale the attributes before applying the lasso algorithm [61]. Another approach to select ‘grouped’ attributes, based on their correlation structure, is to use both L_1 and L_2 penalties, called ‘elastic net’ [62]. This might be particularly useful for SNPs with high linkage disequilibrium. To model genomic data that is linearly ordered along chromosomes, Tibshirani and colleagues developed ‘fused’ lasso [63, 64], which encourages regression coefficients to have similar values when they represent the effects of genetic variants close on the same chromosome. These examples demonstrate the broad applications, and the novelty, of using L_1 penalties for model selection. For high-dimensional data, in our case high-dimensional kernels, the ability to both shrink and select coefficients (whether in the attribute space or the feature space) is appealing. For example, the advantage of the L_1 penalty was recently realized in a method to simultaneously analyze all SNPs from a genome-wide association study [65]. Although they did not include a kernel to measure SNP similarities, it seems that including such information would likely increase the power of these types of shrinkage and selection methods.

The consistency of the lasso model selection, however, depends on the correlation of the attributes. Roughly, if the irrelevant attributes are not correlated with the relevant attributes, then models selected by lasso tend to be consistent, in the sense of selecting the true model for large sample sizes [66]. This active research on model selection and shrinkage will likely provide more powerful ways in which kernels can be used in penalized loss functions.

The generality of RKHS has important consequences for linking semi-parametric regression models with LMMs [8, 67], for linking kernels, penalized splines, LMMs, and empirical Bayes (ridge) shrinkage [68], for SVMs that use kernels [41, 69], for penalized likelihood

methods for generalized linear models [70–72], and for smoothing spline regression [57, 68, 73]. A few of these topics are summarized in the subsequent sections.

Support Vector Machines

One of the primary uses of SVMs is to classify subjects into two groups (e.g. diseased vs. nondiseased). Although SVMs can be used to classify entities into more than two categories and to analyze quantitative traits (support vector regression), we discuss only the binary classification problem. A goal of classification is to determine an optimal boundary according to the measured attributes that minimizes classification error. When dealing with linear features, if the two classes are separable (i.e. linear separation without points on the wrong-side), a hyperplane can be found that creates the largest margin between the training points for the two classes. This can be achieved by using a ‘hard-margin’ loss function (see table 1). Furthermore, because the SVM loss functions are discontinuous, the largest margin is primarily influenced by the points closest to the hyperplane – distant points on either side of the hyperplane can be removed without changing classification errors. Although separability might not be completely achievable, ‘slack’ variables can be introduced to allow for a few points on the wrong side of the classifier, resulting in a ‘soft-margin’ loss function. Sometimes, allowing a few misclassified points in the training data can result in a wider margin separating the groups, providing increased accuracy for predicting future observations [43]. Because logistic regression is frequently used to build a classifier, it is worthwhile to note that the hinge-loss function (see soft-margin loss function in table 1) of SVMs has similar tails as the negative log-likelihood loss function for logistic regression, yet very different tails from the squared-error loss for a binary trait [53], suggesting that the hinge loss and negative log-likelihood for Bernoulli data have similar properties.

A significant advantage of kernel functions is that they can be nonlinear measures of similarities between subjects, yet we can rely on the RKHS to create the necessary underlying linear feature space. This allows us to transform back and forth between the kernel’s nonlinear measures on the attributes and the feature space that relies on linear measures of distance in the optimization algorithm. Hence, the optimization problem is moved to the nice computational advantages of the feature space, where the problem is to determine an optimal hyper-plane (e.g. euclidean metrics) that maximizes the boundary be-

tween the two groups. The underlying feature vectors that are closest to the optimal hyper-plane are called support vectors. These support vectors have non-zero α ’s and play the major role in SVM classification – the vectors further away have much smaller impact on the optimal separation, and so their α ’s can be set to zero. A significant advantage of the SVM algorithm is that determining the maximum margin hyper-plane reduces to a quadratic programming problem, whose convex objective function can always be maximized efficiently under given constraints.

The output from a SVM classification algorithm is a function that is ‘learned’ from the training set,

$$f(x) = \sum y_i k(x, x_i) \hat{\alpha}_i + b,$$

where x is a generic set of attributes (i.e. the argument of the function), perhaps for a new subject, x_i are the attributes and y_i are the class labels ($y_i \pm 1$) for the training points ($i = 1, \dots, n$), $\hat{\alpha}_i$ is the estimated coefficient for the i -th support vector, and b is analogous to the intercept, or offset, in logistic regression (called ‘bias’ in machine learning literature). The classification of a new point with attributes x is based on the sign of $f(x)$. A critical aspect of f is that it is based on a linear combination of the kernel evaluated between a new vector of attributes, x , and the attributes of all subjects in the training set. This means that the function is a smoothing of the similarity of a generic set of attributes with the attributes of all subjects in the training set. This concept of smoothing the similarity of a subject with all other subjects is also fundamental to nonparametric regression and Bayes shrinkage.

The SVM method offers several significant advantages. First, the kernel matrix of similarities is often a nonlinear mapping of attributes to similarities, yet SVM uses euclidean distances of the implicit underlying feature space to create an optimal hyper-plane boundary between groups, providing enormous flexibility to determine an optimal boundary. The kernel is often high-dimensional, making it easier to determine a hyper-plane without the need of ‘slack’ variables to allow for points on the wrong side of the boundary. Second, focusing on the hyper-plane that creates the maximum margin provides a ‘safety margin’ for classification, reducing classification error rates. Third, because only the vectors closest to the margin, the support vectors, determine the optimal boundary, the others could be removed from the training set without loss of information. This means that only the support vectors have non-zero α coefficients, resulting in a sparse solution in feature space, controlling the risk of over-fitting the boundary to the training data. Accessible

introductions to SVMs are Ben-Hur et al. [43] and Cristianini and Schölkopf [74], with extensive treatment in several books [41, 53, 69].

Nonparametric Regression

In linear regression, the objective is to estimate regression coefficients that best describe the data, typically by the least squares estimates that are known to be unbiased and minimize the residual error sums of squares. To avoid over-fitting the parameter estimates when there are many attributes, or to avoid unstable estimates when the columns of the design matrix are highly correlated (collinear), ridge regression can be fit by using the penalty function $\lambda \sum_i \beta_i^2$; the ‘loss + penalty’ function to minimize is $loss(\beta) = (Y - X\beta)'(Y - X\beta) + \lambda \sum_i \beta_i^2$. The resulting ridge regression coefficient estimates are biased, yet their smaller variance can provide better predictions (i.e. smaller prediction error) than ordinary least squares. This form of ridge regression has been widely applied and discussed in the statistical literature, and has recently been proposed as a way to account for linkage disequilibrium among SNP ‘predictor’ variables in genetic association studies with quantitative traits [5].

Ridge regression, however, can be generalized further to give stronger weight to prior information about the expected relationships of the regression coefficients. Although this generalization can be based solely on the choice of penalty function, an equivalent mathematical solution is motivated from a bayesian viewpoint. Consider the standard regression model, $Y = X\beta + e$, with $e \sim N(0, \sigma_e^2)$. Furthermore, assume the prior distribution $\beta \sim N(0, \sigma_K^2 K)$, where K is a fixed correlation matrix for the β 's and σ_K^2 is a variance component. It is well known [75] that the Bayes posterior distribution of β is also normal with posterior mean

$$\hat{\beta} = [V^{-1} + (\sigma_K^2 K)^{-1}]^{-1} (1/\sigma_e^2) X'Y,$$

where $V = \sigma_e^2 (X'X)^{-1}$ is the usual variance matrix for the ordinary least squares estimates, $\hat{\beta} = (X'X)^{-1} X'Y$. After replacing V with $\sigma_e^2 (X'X)^{-1}$, using the variance ratio $\lambda = \sigma_e^2/\sigma_K^2$, and simplifying, the posterior mean can be expressed as

$$\hat{\beta} = (X'X + \lambda K^{-1})^{-1} X'Y, \quad (2)$$

which is the same as a ridge regression solution when using the penalty function $\lambda \beta' K^{-1} \beta$. This reveals several key ideas. First, the kernel matrix K represents a prior correlation structure for the expected relationships among the

β 's. Second, the penalty parameter, λ , can be interpreted as a variance ratio: $\lambda = \sigma_e^2/\sigma_K^2$. When the kernel explains little or none of the trait variation, σ_K^2 is much smaller than σ_e^2 . This implies that λ gets very large, in which case $\hat{\beta} \rightarrow 0$. In contrast, when the kernel explains the majority of the trait variation, $\lambda \rightarrow 0$ and $\hat{\beta}$ approaches the ordinary least-squares unpenalized estimate.

In the preceding formulation, the design matrix X imposed an assumed parametric structure. To progress to nonparametric regression, suppose that we replace X with the identity matrix, and replace β with an unspecified function, f , so that the regression model is $Y = f + e$. The preceding penalized loss function now reduces to nonparametric kernel regression [8, 67, 76]. Substituting the identity matrix I for X in expression (2) reveals

$$\begin{aligned} \tilde{f} &= (I + \lambda K^{-1})^{-1} Y \\ &= K[K(I + \lambda K^{-1})]^{-1} Y \\ &= K(K + \lambda I)^{-1} Y. \end{aligned}$$

This emphasizes that when there is no explicitly defined design matrix, the flexible nonparametric modeling depends only on the kernel matrix K and the penalty coefficient λ .

Kwee et al. [8] used this nonparametric approach to simultaneously model SNP genotypes while adjusting for known covariates. They used the vector x_i to denote covariates (including the intercept), G_i to denote the set of measured genetic markers, and $f(G)$ to denote an unknown centered smoothed function that depends on the genetic markers. The following semi-parametric regression model was used

$$y_i = x_i' \beta + f(G_i) + e_i,$$

where $x_i' \beta$ is parametric while $f(G)$ is nonparametric. Based on least-square kernel machines (LSKM), Liu et al. [76] showed that β and $f(\cdot)$ can be estimated by maximizing a penalized likelihood function, equivalent to minimizing the following loss + penalty function,

$$L(\beta, f) = \frac{1}{2} \sum_{i=1}^n [y_i - x_i' \beta - f(G_i)]^2 + \frac{1}{2} \lambda \|f\|_{H_K}^2$$

Here, H_K is a RKHS space that is determined by a psd kernel function, $k(G_i, G_j)$ – see Appendix 1. The parameters can be estimated as

$$\begin{aligned} \hat{\beta} &= [X'(K + \lambda I)^{-1} X]^{-1} X'(K + \lambda I)^{-1} Y, \\ \hat{f} &= K(K + \lambda I)^{-1} (Y - X\hat{\beta}). \end{aligned}$$

They further showed that this type of semi-parametric regression model can be expressed as a LMM, so that the parameters can be determined in standard software for mixed models. This comes from the connection between

the penalized likelihood for the kernel machine and the LMM likelihood. That is, when it is assumed that $f \sim N(0, \sigma_K^2 K)$ and $e \sim N(0, \sigma_e^2 I)$, the marginal variance matrix of Y is $V = \sigma_K^2 K + \sigma_e^2 I$, a variance components model with σ_K^2 as the variance component attributed to the similarity of SNP genotypes (i.e. the kernel matrix) and σ_e^2 as the residual environmental variance. The penalized model (with penalty parameter λ) and the mixed model are connected by $\lambda = \sigma_e^2 / \sigma_K^2$.

It is well known that software for mixed models can be used to fit penalized models, including smoothing splines, nonparametric kernel regression, and empirical Bayes shrinkage [77–80]; the connection between kernel smoothing and mixed models is becoming better appreciated [67, 76]. The advantage of framing the kernel regression in terms of a LMM is that standard software can be used, particularly restricted maximum likelihood (REML) for quantitative traits. Why REML? The maximum likelihood estimator of variance components in a linear model can be biased downwards because it does not adjust for the degrees of freedom that are lost by estimating the fixed effects regression coefficients. REML corrects this by maximizing the likelihood of a set of residual contrasts, often with better performance than the usual maximum likelihood estimators.

Gianola and van Kaam [67] used a semi-parametric model similar to that of Kwee et al., but with a slight extension to analyze data from pedigrees. They added a kinship coefficient matrix to the model, essentially another kernel matrix, to account for additive genetic variance due to unmeasured polygenes. Gianola et al. provide a nice review of the main concepts of RKHS and how it relates to penalized likelihood and LMMs. They also provide the following key ideas. First, in the classical polygenic model, the kinship coefficient matrix induces an exchange of information of the polygenotypes among relatives, in the sense of correlated polygenotypes leading to correlated phenotypes. In a similar manner, the kernel matrix K for genomic similarity according to SNPs induces an exchange of information among unrelated subjects. This can be seen in the preceding formulation of the model by Kwee et al.: $\hat{f} = K\hat{\alpha}$, where $\hat{\alpha} = (K + \lambda I)^{-1}(Y - X\hat{\beta})$. Second, Gianola et al. illustrate how to extend the nonparametric model to pairs of chromosomes by allowing different nonparametric functions for the different chromosomes. Third, Gianola et al. discuss choice of kernel, particularly for discrete genotypes. Fourth, they discuss the potential importance of filtering a large number of SNPs to a smaller number to improve prediction. Liu et al. [76] emphasize that basing a kernel on a subset of

SNPs results in a new kernel, and so SNP subset selection can be viewed as kernel selection. For a small number of genetic markers, Liu et al. were able to perform kernel regression for all possible subsets, and use Akaike information criterion to select a nonparametric model. Further research is needed on kernel selection for a large number of SNPs, particularly to distinguish between fewer SNPs for kernels useful to describe data versus more SNPs for kernels useful to predict future observations with minimum error.

Kernels, Regularization, and Mixed Models

The preceding developments for shrinkage and nonparametric regression with respect to kernels can be formulated more generally in the usual LMM framework. This is useful to see how standard concepts in LMM's tie with kernels. Suppose we assume a standard LMM, $Y = X\beta + Z\gamma + e$, where X and Z are matrices for predictor variables, β is a vector of fixed effects, and γ is a vector of random effects, with the usual assumed normal distributions for the residual errors [$e \sim N(0, \sigma_e^2 I)$] and the γ 's [$\gamma \sim N(0, \sigma_K^2 K)$]. The marginal mean of Y is $E[Y] = X\beta$ and its marginal variance matrix is $V = \sigma_K^2 ZKZ' + \sigma_e^2 I$. Maximizing the mixed model log-likelihood is equivalent to minimizing a squared-error loss function plus an L_2 penalty function,

$$\text{loss}(\beta, \gamma) = (Y - X\beta - Z\gamma)'(Y - X\beta - Z\gamma) + \lambda \gamma'K^{-1}\gamma,$$

whose solution can be found by taking derivatives with respect to β and γ , setting the derivative equations to zero, and solving for β and γ . These equations are equivalent to Henderson's mixed-model equations. The resulting estimators are

$$\begin{aligned}\hat{\beta} &= (X'V^{-1}X)^{-1}X'V^{-1}Y, \\ \hat{\gamma} &= KZ'[ZKZ' + (\sigma_e^2/\sigma_K^2)I]^{-1}(Y - X\hat{\beta}).\end{aligned}$$

When σ_e^2 and σ_K^2 are known, the estimators $\hat{\beta}$ are the best linear unbiased estimators (BLUE) and $\hat{\gamma}$ are the best linear unbiased predictors (BLUP); when σ_e^2 and σ_K^2 are estimated, $\hat{\beta}$ and $\hat{\gamma}$ are called *empirical* BLUE and BLUP, respectively.

Hierarchical Models

In the previous formulations for shrinkage, nonparametric regression, and mixed models, the random effects had a prior mean of zero. If this is not the case, the prior

mean can be absorbed into the fixed effects. However, it is often more informative to use a hierarchical approach to formulate models by allowing the first-stage random effects, say γ , to depend on covariates. For example, consider the random effects model $Y = Z\gamma + e$ where $E[\gamma] = W\pi$. In this context, attributes measured on subjects are used to create the first-stage design matrix, Z , while higher-level attributes that describe the γ 's are used to create the second-stage design matrix, W . This two-stage hierarchical model has been advocated for analyzing epidemiological studies [81, 82], and recently for genetic studies where the columns of the Z matrix represent SNP genotypes and the columns of the W matrix represent prior information about the SNPs. Prior SNP information could include degree of evolutionary conservation, functional information, tagging information or linkage scores from prior studies [4], or linkage disequilibrium structure, such as indicator variables for different haplotype blocks [3].

A distinction between hierarchical models and marginal models is that the covariance matrices at each level of the hierarchical model are assumed to be psd, while only the marginal covariance is required to be psd for the marginal approach. Nonetheless, the hierarchical approach provides an intuitive way to build a complex model, while using the marginal model formulation to fit the model by variance components methods [82].

Hierarchical models can be fit in several ways, such as a two-stage approach, jointly iteratively reweighted least squares, a penalized quasi-likelihood, or a mixed-model formulation [83]. Because the two-stage approach has been widely used in genetic studies, it is worthwhile to review this strategy, and some of its limitations. In the first stage, the usual maximum likelihood parameters are estimated ($\hat{\gamma}$). In the second stage, the $\hat{\gamma}$'s are treated as independent variables and weighted least-squares is used to estimate $\hat{\pi}$. Then, the first-stage $\hat{\gamma}$'s are combined with the second-stage predicted values, $W\hat{\pi}$, using a variance-weighted average. A limitation of this two-stage fitting, however, occurs when there are many columns of the second-stage design matrix W , such as when there are a large number of different attributes for SNPs, or when the sample size is small. For small sample sizes, more efficient methods are available, such as jointly estimating the first-stage (γ) and second-stage (π) parameters, or using a penalized likelihood [83]. Other limitations of the two-stage approach are: (1) the first-stage is typically based on estimating the effect of only one SNP at a time (the marginal effect of each SNP), ignoring correlations among SNPs; when there is a large number of SNPs, SNP subset selec-

tion must be performed [4]; (2) it is merely an approximation, which can perform poorly when data is sparse or when the trait is nonnormal, such as a binary trait.

For traits that are not quantitative, such as binary data, hierarchical models can still be used, but in the framework of generalized linear mixed models (GLMMs). Maximization of the log-likelihood can be computationally challenging because of the need to numerically integrate over multidimensional integrals of the distribution of the random effects. A potential advantage of using the kernel matrix to represent the prior covariance matrix of the random effects is that there are only two variance components in the model, σ_K^2 and σ_e^2 . Furthermore, the variance structure for the γ 's, $V(\gamma) = \sigma_K^2 K$, suggests a simple reparameterization to simplify numerical integrations: use the generalized Cholesky decomposition $K = LDL'$, where L is a lower-triangular matrix with 1's along the diagonal and D is a diagonal matrix with nonnegative diagonal elements. The transformation matrix $LD^{1/2}$ can be used to make transformed parameters independent, so multidimensional integrals can reduce to products of lower-dimensional integrals, facilitating numerical integrations.

Alternative approximate methods to fit GLMMs have been developed. The popular penalized likelihood approximation for GLMMs in GLIMMIX (SAS software, www.sas.com) [84] iteratively fits a LMM to a modified dependent variable [85]. These types of approximations, however, do not work well for binary data with small clusters [86–88]. How these methods would work with general kernel matrices is unknown, but based on the need of 'not too small clusters', it seems reasonable to expect that kernel matrices would work well as long as they do not generate many small patchy clusters (e.g. many small clusters of subsets with high within-cluster similarities). An indication of possibly poor approximation of the penalized likelihood method is when the variation attributed to the random effects, σ_K^2 , is small [83]. In these situations, Markov chain Monte Carlo (MCMC) methods that perform numerical integrations, such as in the popular BUGS software (www.mrc-bsu.cam.ac.uk/bugs), would be preferred. Importantly, the two-stage approximation can give parameter estimates that differ in sign from MCMC estimates when σ_K^2 is small [4], again emphasizing the need to use MCMC methods for sparse data when anticipated effect sizes are small.

Note that it is common to assume an exchangeable prior for the second stage, which means that the covariance matrix $V(\gamma)$ does not change if the γ 's are interchanged in their vector. This makes sense when a sec-

ond-stage W matrix is used to model attributes of the SNPs, so that $V(\gamma)$ captures residual correlations among the γ 's not explained by W . But, exchangeability might fail if we mismodel the second stage. Furthermore, if there are many columns in the second-stage W matrix, and some are highly correlated, then this could cause numerical problems. An alternative is to use a nonparametric approach at the second stage by using a kernel matrix to encode the similarities of the SNPs. This would be particularly attractive if the SNP attributes tend to be qualitative and difficult to encode into the second-stage design matrix.

Because the LMM and GLMM formulations are based on a normal distribution for the second-stage random effects, they implicitly impose an L_2 penalty function, similar to ridge regression. This highlights a general limitation: there is no subset selection for choosing the 'important' predictors. Whether this is advantageous for complex traits needs further evaluation and real-data applications. As discussed elsewhere, a bayesian view on the usual regression subset selection is that it is optimal if in fact there are only a few predictors with large effects, in contrast to ridge regression, principal components regression, and partial least squares that are optimal when there are many predictors of small effects such that many of them jointly influence the response, but no small subset of them stands out [89]. For the genetics of complex traits, attempting to interpret individual regression coefficients might be misleading, and a more thorough understanding might be provided by a linear combination of a subset of predictors.

Score Statistics

The statistical association of phenotype similarity with genotype similarity can be evaluated by using kernels in score statistics. A substantial benefit of score statistics is less computation time compared to maximum likelihood estimation and likelihood ratio statistics. Score statistics are based on a kernel representing a structured covariance matrix for the random effects [8, 29, 67, 76, 90], with derived score statistics for variance components in LMMs [8, 76] or GLMMs [7, 91–94]. Published work on this topic can be a bit confusing. The algebraic formulations differ, some statistics were developed to test lack-of-fit of fixed-effects models [91], some publications allow general kernels [67] while others are more specialized [7, 8, 92], some develop methods that account for kurtosis for nonnormal data [91, 92, 95], others do not [7,

8], and some allow for over- or underdispersion for binomial and Poisson data [95]. To keep focus on the use of kernel matrices in score statistics, we summarize a few main results here, with some intuitive guidelines, and provide more details in Appendix 2.

Assume measurements are made on a trait value y_i for subject i ($i = 1, \dots, n$), where y_i can be binary, quantitative, or any type of trait that can be modeled by generalized linear models with an appropriately chosen link function (we restrict to canonical link functions). Furthermore, assume measures are made on m genetic markers, such as SNPs, and let G_i denote the set of measured genetic markers for subject i . To evaluate the association of y_i with G_i , we use a kernel matrix K that is based on the measured genetic markers. In addition, it is sometimes appropriate to adjust for known covariates, treating them as fixed effects. These fixed effects are often nongenetic, although sometimes it can be advantageous to adjust out the effects of known genetic factors when searching for new genetic influences. With the traits in vector Y , the fitted values based on the fixed-effects model denoted $\hat{\mu}$, and kernel matrix K for pairs of subjects based on their genetic similarity, the score statistic can be expressed as

$$T_K = \frac{1}{2\hat{\phi}^2} (Y - \hat{\mu})' K (Y - \hat{\mu}),$$

(where $\hat{\phi}$ is an estimated scale parameter. This emphasizes that T_K is a measure of the correlation of the residuals from the fixed-effects model with the structured kernel matrix.

For a quantitative trait with a normal distribution, the residual mean squared error from a regression on the fixed effects can be used to estimate the scale, $\hat{\phi} = \hat{\sigma}_{mse}^2$. But, it is important to account for estimating $\hat{\phi}$ when approximating the distribution of T_K . Kwee et al. [8] used a similar approach, but used ϕ in place of ϕ^2 in the denominator of T_K ; this did not matter, because the derived moments of T_K under the null hypothesis accounted for this difference. For binomial and Poisson data $\phi = 1$, although ϕ can deviate from 1 when there is over- or underdispersion. Overdispersion can occur when covariates are missing or mismodeled (e.g. assuming linear when there are quadratic effects, or there are missing interaction terms), when the wrong link function is used (e.g. binomial when probit is better for rare events), when there are large outliers, or when there is unaccounted correlation among subjects (perhaps due to cryptic relatedness). For Bernoulli data, however, there is no information on ϕ unless the data is clustered. So, for most case-control studies, $\phi = 1$.

Statistical Power of Kernels

To choose a kernel, it would help to know the theoretical statistical power for competing methods. Unfortunately, this theory does not exist because the distribution of the score statistic is difficult to determine (e.g. a complex mixture of χ^2 -distributions under the null, and not well defined under an alternative, partly because the kernel matrix K is not necessarily a block-diagonal matrix). However, to gain insight into how the kernel matrix likely impacts power, we consider the following crude approximation. Intuitively, the power should depend on how close the chosen kernel is to the ‘true’ kernel, in the sense of the percentage of the variation of a trait that is explained by the kernel matrix. For a quantitative trait and a random effects model (i.e. no fixed effects), the marginal variance of Y is $V(Y) = \sigma_K^2 K + \sigma_e^2 I$. If we could appeal to the central limit theorem for independent vectors (meaning $V(Y)$ is block-diagonal), then the power to test whether $\sigma_K^2 > 0$ would be determined by classical likelihood ratio theory for variance components [80]. Under the alternative hypothesis that $\sigma_K^2 > 0$, the classical likelihood ratio statistic has a noncentral χ^2 distribution, so power depends on the noncentrality parameter. Following the steps outlined elsewhere [96], we derived the noncentrality parameter as

$$ncp = \frac{h_K^2}{2} [tr(KK) - 2tr(K) + n]$$

where $h_K = \sigma_K^2 / (\sigma_K^2 + \sigma_e^2)$. Hence, we interpret h_K as the percentage of the total variation of Y attributed to the kernel matrix K (the ‘heritability’ for the kernel matrix). This can be simplified a bit by replacing K with its eigendecomposition, $K = UDU'$, where D is a diagonal matrix of eigenvalues and U a matrix of corresponding eigenvectors. Noting that $tr(K) = tr(UDU') = tr(DU'U) = tr(D)$, because U is orthonormal (i.e. $U'U$ results in the identity matrix), and $tr(KK) = tr(D^2)$, the noncentrality parameter can be expressed as

$$ncp = \frac{h_K^2}{2} \left[\sum_{i=1}^n (d_i - 1)^2 \right]$$

where d_i is the i -th eigenvalue. This illustrates that the noncentrality parameter depends on the square of the ‘heritability’ h_K and the variation of the eigenvalues about 1. If the chosen kernel grossly differs from the true kernel, then h_K will shrink towards zero, likely weakening power. In practice, it might be useful to compare competing kernels according to the variation of their eigenvalues about 1, although actual power would require simulations.

Similarity Kernels as Prior Information

Throughout the previous developments, particularly mixed models and hierarchical models, the kernel matrix served as a way to structure the correlation matrix for the random effects. Although the statistical value of a kernel depends on the study goals and subject matter, it is useful to consider a kernel as a generalized covariance for pairs of subjects; psd kernels in fact arose as covariance kernels of stochastic processes. Exchangeable priors are typically used in Bayes analyses, where permutation of the elements of the random effects, the γ vector, does not change its mean or covariance matrix. For this reason, the identity matrix is often used as a prior covariance matrix. Goeman et al. advocated this type of prior for a GLMM score statistic [94], and Chapman and Whittaker [97] found that Goeman’s score statistic had higher power than competing multi-SNP analysis methods for case-control data. Goeman’s analytic work showed that the score statistic is expected to have greater power than the maximum of univariate regression test statistics when there are a large number of tests and only a few true non-zero effects, and surprisingly even when there is only one true non-zero effect [94].

This provides a framework to think of a kernel matrix as a covariance matrix with prior assumptions about its structure, as a bayesian informative prior [98]. Properly defined informative kernels, even nonexchangeable, could potentially increase power of the kernel score statistics. As emphasized elsewhere, the potential strength of the bayesian perspective is the ability to include strong prior information when appropriate [99]. These intuitions might provide guidance on how to create kernels appropriate to the scientific aims of a study, and how much extraneous information is relevant to build a kernel.

Although the kernel methods described in this review do not require adherence to the bayesian philosophy, the distinction between Bayes and frequentist approaches to data analysis become blurred when extraneous ‘prior’ information is used in the analysis. From this perspective, the kernel matrix serves as the link between the chosen analytic methods and a bayesian perspective.

Distance-Based Regression for Nonmetric Similarities

When the kernel matrix is not psd, such as when non-metric measures of similarity are used in ecology studies, McArdle and Anderson [100] [see also 101–103] devel-

oped a distance-based regression to test for the association of one or more traits with a matrix of either similarities or dissimilarities. Schork and colleagues advocate this approach for genetic studies [6, 104–106].

A distance-based regression requires a matrix of genomic similarities between pairs of subjects. It then swaps the roles of traits and genomic similarities by treating the genomic similarities as the outcome variable and the traits as the predictor variables. To see how this works, consider the usual least square setting for multivariate analysis of variance. Let Y denote an $n \times t$ matrix of t traits, and X an $n \times p$ matrix of p predictor variables. McArdle and colleagues suggested that a pseudo- F statistic could be used to test the null hypothesis of no association of Y with X ,

$$F = \frac{\text{tr}(\hat{Y}'\hat{Y})/(p-1)}{\text{tr}(R'R)/(n-p)}$$

which reduces to the usual univariate F statistic for ordinary linear regression when there is a single trait ($t = 1$). The numerator represents the model-fit sums of squares,

$$\hat{Y} = X\hat{\beta} = X(X'X)^{-1}X'Y = HY,$$

where H is the usual ‘hat’ matrix, and the denominator represents the residual sums of squares,

$$R = Y - Y' = (I - H)Y.$$

Because $\text{tr}(AB) = \text{tr}(BA)$, the pseudo F -statistic does not change if we replace $\text{tr}(\hat{Y}'\hat{Y})$ with $\text{tr}(\hat{Y}\hat{Y}') = \text{tr}(HYY'H)$, and replace $\text{tr}(R'R)$ with $\text{tr}(RR') = \text{tr}[(I - H)YY'(I - H)]$. Now, suppose we interchange the roles of Y and X : put the traits in the X matrix to then create the H hat matrix, and use the kernel trick by replacing YY' with a genomic distance matrix. This ‘swap’ can be used in the pseudo F -statistic.

To create a distance matrix, a genetic similarity matrix, S , is standardized to range over 0–1, with 1’s along the diagonal (i.e. maximum similarity for self-matching), and then converted to a dissimilarity matrix, $D = 1 - S$. The dissimilarity matrix is then converted to an ‘association’ matrix, A , with elements $a_{ij} = -d_{ij}^2/2$. This is a key step towards attempting to convert non-euclidean dissimilarity D to a euclidean distance in A . Another step is to ‘center’ the A matrix. In euclidean geometry, the distance between two points does not change if the central location of all points is moved. However, when A is decomposed into its eigenvectors and eigenvalues, the distance between points according to their eigenvectors is dominated by two aspects: (1) the magnitude of the eigenvalues, and (2) large values of $(e_{ki} - e_{kj})^2$, where e_{ki} is the i -th value of the k -th eigenvector. This means that if the

elements of an eigenvector have little variation, then the eigenvector contributes little to the distance. Often, the largest eigenvalue corresponds to the central location of a matrix, and its corresponding eigenvector has little variation. Hence, removing this effect by centering the A matrix reduces noise in the underlying measured distances. One could also argue that the eigenvectors with small eigenvalues should be removed, to further reduce noise, as is commonly done in principal components regression (i.e. use only the principal components with large eigenvalues because they explain most of the covariance structure of the matrix), although this was not proposed. To center the matrix A , for each element subtract the row mean, the column mean, and add back in the overall mean. This can be expressed in matrix notation as

$$G = \left(I - \frac{1}{n}J\right)A\left(I - \frac{1}{n}J\right),$$

where J is a matrix of 1’s. Although G is not guaranteed to be psd, if it is, it has euclidean distances.

The above G matrix can then be used in the pseudo F -statistic to evaluate the association of measured traits (treated as predictor variables) with pair-wise genetic distances (treated as dependent variables) according to

$$F = \frac{\text{tr}(HGH)/(p-1)}{\text{tr}[(I - H)G(I - H)]/(n-p)}$$

Because the distribution of F is unknown, permutations of the rows of X (the trait variables) is required to compute p values. An advantage of using permutations is that it allows for negative eigenvalues of the G matrix, and so allows for distance metrics that do not satisfy the triangle inequality. This does, however, complicate interpretations because eigenvalues are interpreted as the amount of variation explained by each eigenvector, and having negative variances is awkward. Although McArdle and Anderson suggest partitioning analyses into the positive and negative eigenvalues, interpretation remains problematic.

Although this approach allows for nonmetric similarity (or dissimilarity) measures, it has a number of features that can limit its utility: (1) it treats the genotype similarity matrix as the dependent variable and traits as predictors, making it difficult to adjust for true covariates and to extend to more complicated modeling (e.g. gene-environment interactions, genetic subset selections, etc.); (2) it is very computer-intensive to compute p values; (3) its main focus on p values overlooks parameter estimation, confidence intervals, and model selection; (4) it does not easily generalize to more complex traits, such as censored

Table 2. Kernel math for a new kernel, $K(x_i, x_j)$, from existing kernels, $K_1(x_i, x_j)$ and $K_2(x_i, x_j)$

$K(x_i, x_j) = x_i'x_j = \langle x_i, x_j \rangle$	vector dotproduct
$K(x_i, x_j) = f(x_i)f(x_j)$	function product
$K(x_i, x_j) = x_i'Bx_j$	vector dotproduct, weighted by symmetric psd matrix B
$K(x_i, x_j) = K_1(x_i, x_j) + K_2(x_i, x_j)$	kernel addition
$K(x_i, x_j) = aK_1(x_i, x_j)$	scalar multiplication ($a > 0$)
$K(x_i, x_j) = K_1(x_i, x_j) \cdot K_2(x_i, x_j)$	kernel multiplication (element-wise matrix multiplication)
$K(x_i, x_j) = \sum_{i=0}^p a_i K_1(x_i, x_j)^i$	general polynomial of order p ($a_i \geq 0$)
$K(x_i, x_j) = (\langle x_i, x_j \rangle)^p$	homogenous polynomial of dotproduct
$K(x_i, x_j) = (\langle x_i, x_j \rangle + c)^p$	inhomogenous polynomial of dotproduct ($c > 0$)
$K(x_i, x_j) = \exp[K_1(x_i, x_j)]$	exponentiation of kernel
$K(x_i, x_j) = \exp(-\ x_i - x_j\ ^2/\sigma^2)$	gaussian kernel (a type of radial basis function; see [41], p. 46 for more general forms); $\ x_i - x_j\ ^2 = \sum_k (x_{ik} - x_{jk})^2$ is a measure of distance, σ^2 scales the distance, and the exponential converts to a measure of similarity.

survival endpoints, and (5) it does not attempt to reduce the underlying dimension of the similarity matrix, which might be useful when some of the eigenvalues are small. In summary, although this distance-based regression method offers a way to handle negative eigenvalues, it seems excessively restrictive. Perhaps a better choice of kernel that generates psd matrices would be a better approach, allowing a wider range of regularized regression methods to be used.

Making Similarity Kernels

A significant advantage of the kernel method is its expressive power to capture domain knowledge in a general manner. Yet, this generality can also make it difficult to begin to know how to make a good kernel for a specific problem. The basic requirement is a symmetric function that generates a psd matrix for all sample sizes of subjects (from 1 to infinity). To provide some guidance on how to make kernels, it is important to know that new kernels can be created by using other kernels as building blocks, hence the expressive power of kernels. Roughly speaking, there are mainly three mathematical operations that can be used to create new kernels from existing kernels: multiplying by a positive scalar, adding kernels, and multiplying kernels (element-wise). Yet, these three operations can be used to create an enormous variety of kernels, some of which are summarized in table 2. Building kernels from

other kernels can also be used to prove that a kernel is psd. To understand how to verify if a new kernel is psd, we outline proofs for seven different kernels in Appendix 3, showing how to build some proofs on others.

To illustrate how to prove that a kernel is valid by showing it is equivalent to a known valid kernel, consider the commonly used measure of similarity for SNP genotype data – the total number of matching genotypes for a pair of subjects (or the proportion of matches). To show that this generates a psd kernel, consider a general categorical attribute with C categories, and let x_i be a vector of length C , where $x_{ic} = 1$ if the i -th subject falls in category level c , and 0 otherwise. Subjects i and j have a match score of 1 if they have the same level c , and 0 otherwise, captured by the dot product, $\langle x_i, x_j \rangle$. So, for a single categorical attribute, the dot product is a valid kernel (table 2). Summing these kernels across all categorical attributes will also result in a valid kernel, and dividing this sum by the number of attributes leads to a valid kernel, implying that kernels that measure either the total number of matching SNP genotypes for a pair of subjects, or the proportion of matches, are valid kernels.

Summary

The notion of genomic similarity has been the basis of many statistical analytic methods, ranging from IBD for pedigree data, to genetic markers AIS for pairs of subjects

or pairs of haplotypes, to DNA/RNA/protein sequence analyses. Although measures of similarity are often based on intuition and heuristics, a strong mathematical foundation can be used to create measures of similarity that generate psd kernel matrices. This in turn provides a wide range of statistical methods and software that can be used to perform sophisticated statistical analyses, such as non-parametric regression, LMMs and GLMMs, hierarchical regression models, score statistics, and SVMs. This review of the mathematical rigor provided by reproducing kernel Hilbert spaces and regularization by general loss functions, as well as general penalty functions, provides a solid framework for further advancements to fit complex kernels. Furthermore, this review of the mathematical framework for validating kernels provides the mathematical tools to move from intuition and heuristics to more rigorous evaluation of similarity measures. The challenges of mapping biological and genomic information to measures of similarity are discussed in the companion paper [1].

Acknowledgements

This research was supported by the U.S. Public Health Service, National Institutes of Health, contract grant number GM065450.

Appendix 1: Mathematical and Statistical Ideas of Vector Space, Euclidean Distance, and Hilbert Space

To develop a general approach to the analysis of kernels that are continuous functions, a certain level of mathematical abstraction provides a rigorous foundation to extend metric distances between vectors to metric distances between functions. A Hilbert space provides the right type of structure, but to understand a Hilbert space, it helps to review the more familiar ideas of the mathematics of vectors of real numbers. A *vector space* (also called

linear space) is a set of vectors that can be manipulated by adding vectors together, multiplying vectors by scalar constants, or performing both operations; any newly constructed vectors are also in the vector space. A vector space can be viewed as a set of points, where each point is a vector, and the elements of a vector indicate where the point resides in the space. For example, if vectors have one element, the space is the real line; two elements, a plane; three elements, a cube or sphere. Finite dimensional vector spaces also have a vector *norm*, $\|x\| = \sqrt{x_1^2 + \dots + x_p^2}$, which gives the euclidean distance from the vector point x to the origin, a consequence of the pythagorean theorem. A norm can also be used to measure the *euclidean distance* between points x and y , $d(x, y) = \|x - y\| = \sqrt{(x_1 - y_1)^2 + \dots + (x_p - y_p)^2}$.

As we move to higher-dimensional vectors, the concepts of a norm and distance can still be used, although care must be taken when moving from vectors of finite dimension to vectors with infinite dimension, which occurs when using continuous kernel functions. In this case, we can view the vector space to have points that are functions, in place of the usual idea of a vector at each point. But, the functions cannot be completely arbitrary, otherwise the mathematics to compute norms and distances might fall apart. The first step toward a restricted vector space is a ‘Banach space’, which is a vector space that has a norm and is complete. We illustrated that a norm can be used to measure distance. Complete means that the distance between points (the functions in the space) becomes arbitrarily small in the mathematical sense of functions converging to some point, with convergence measured by a normed distance (convergence of a Cauchy sequence). Intuitively, a complete space eliminates some types of discontinuous functions that would cause ‘holes’ in the space, with ‘missing points’ around some aberrant function.

Restricting a Banach space further by requiring the space to have an inner product results in a Hilbert space. An inner product generalizes the concept of euclidean geometry. In euclidean space, the inner product of vectors x and y , (also called the ‘dot product’) is just the sum of the products of the elements of the vectors, $\langle x, y \rangle = x_1 y_1 + \dots + x_p y_p$ (also $x'y$ in vector notation). The norm and inner product are related by $\|x\| = \langle x, x \rangle^{1/2}$. An inner product provides a vector space with intuitive geometrical ideas, such as the angle between vectors or length of vectors. It also provides structure to define when vectors are linearly independent. In euclidean geometry, the dot product, norm, and angle (θ) between vectors x and y are all related, $\langle x, y \rangle = \|x\| \|y\| \cos \theta$. So, there are

Appendix Table 1. Generalized linear model functions for different distributions

	Normal	Poisson	Binomial
$g(\mu_i) = X_i' \beta + Z_i' \gamma$	μ_i identity	$\log(\mu_i)$	$\log \text{it}(\mu_i) = \log(\mu_i) - \log(1 - \mu_i)$
$\nu(\mu_i)$	1	μ_i	$\mu_i (1 - \mu_i)$
ϕ	σ_e^2	1 ^a	1 ^a
$(\mu_{4i} - 3\mu_{2i}^2)^b$ kurtosis	0	μ_i	$\mu_i (1 - \mu_i) (1 + 6\mu_i^2 - 6\mu_i)$

^aIf no over/under dispersion. ^b μ_{4i} and μ_{2i} are the 4th and 2nd central moments, and $(\mu_{4i} - 3\mu_{2i}^2)$ measures amount of kurtosis departing from a normal distribution.

several ways to say that vectors x and y are linearly independent: their inner product is 0 ($\langle x, y \rangle = 0$); they are perpendicular, so the angle between them is $\theta = 90^\circ$ ($\cos(90^\circ) = 0$); they are orthogonal. From a statistical perspective, when x and y are centered about their mean values, the norm represents the standard deviation (e.g. length of vectors in units of standard deviations), the dot product represents the covariance of x and y , and $\cos\theta$ represents the correlation of x and y . These concepts remain intact for an infinite dimensional vector space when we require the space to have an inner product.

In summary, a Hilbert space is a complete normed vector space with an inner product. When a kernel function defines a Hilbert space, it allows use of intuitive geometrical ideas of distances between points, even when the kernel that defines this space is nonlinear with respect to the attributes measured on subjects. But, the kernel function cannot be completely arbitrary, because it cannot leave ‘holes’ in the underlying vector space, and it must be psd in order to rely on the underlying metric distances. A psd kernel has eigenvalues that are nonnegative, a key aspect. A few other key parallel concepts from matrix algebra about a square symmetric psd matrix, K , are: the quadratic form $c'Kc \geq 0$ for all $c \neq 0$; the determinant of a matrix ($|K|$) is the product of its eigenvalues, $|K| = \prod \lambda_i$, implying that $|K| \geq 0$. If there is strict inequality (replace ‘ $\geq$ ’ with ‘ $>$ ’), then the matrix K is positive definite and nonsingular.

Reproducing Kernel Hilbert Space

At first glance, dealing with infinite vector spaces might seem more abstract than necessary. More careful scrutiny reveals that the power of this approach comes from the property of closure for Hilbert spaces, which means that arbitrary finite samples are also contained in the Hilbert space, so we are assured that all the general properties of an infinite dimensional Hilbert space will apply to any size sample.

Let us now delve deeper into some of these ‘spaces’. An intuitive guide from linear algebra will help. Recall that a non-zero vector e is called an eigenvector of matrix K if $Ke = \lambda e$, where λ is the corresponding scalar eigenvalue. Going further, the eigendecomposition of a finite dimensional matrix is

$$K = \sum_{i=1}^n \lambda_i e_i e_i'.$$

From this, we can see that K defines the eigenvectors and eigenvalues – there is no need to explicitly define them. Eigenvectors are sometimes called latent vectors for a good reason – they are not readily apparent! Another key feature is that the eigendecomposition is based on a linear combination (using λ_i ’s as weights) of inner products.

The RKHS follows a similar idea, but instead of linear combinations of vectors, it is based on linear combinations of functions, and moves from a finite number of elements to an infinite number of functions. An advantage of this approach is that we only need to define the kernel function. If the kernel function $k(x, y)$ is symmetric and psd, then Mercer’s theorem [41, 42, 44, 107] provides two powerful results. First, it allows the kernel function to be decomposed into its eigenvalues and orthonormal eigenfunctions [denoted $\phi(x)$],

$$k(x, y) = \sum_{i=1}^{\infty} \lambda_i \phi_i(x) \phi_i(y).$$

Second, it provides an inner product, which relates the kernel function to an inner product of the implicitly defined eigenfunctions, $k(x, y) = \langle \phi(x), \phi(y) \rangle$. The good news is that we do not need to define the ϕ functions, much like we do not need to explicitly define latent vectors to know that they exist. In the machine learning literature, this imbedded space of ϕ functions is often called the ‘feature space’.

To understand the ‘reproducing’ property of RKHS, suppose we use a kernel function k to create a kernel matrix K (lower-case k denotes a kernel function; upper-case K a kernel matrix). Let K_s and K_t be two columns of the kernel matrix. This means that we can view the column vector K_s as a vector of functions, say $K_s = K(s, \cdot)$, where the column vector s is fixed and ‘ $\cdot$ ’ a place-holder for the function argument (e.g. K_s measures the similarity of an input attribute x with each ‘item’ in the s -th column). Then define the inner product of K_s and K_t as $\langle K_s, K_t \rangle = k(s, t)$. This inner product is key, because if two functions (say f and g) in the RKHS have a close distance defined by the inner product, then the output values from these functions, $f(x)$ and $g(x)$, will also be close. Furthermore, this inner product completely defines the set of functions in the corresponding RKHS. For example, if the set of functions in the RKHS is denoted H_K , then any function $f(x)$ in H_K can be expressed as a linear combination of the basis functions,

$$f(x) = \sum_{i=1}^{\infty} a_i \phi_i(x).$$

Suppose that g is also in H_K , represented as

$$g(x) = \sum_{i=1}^{\infty} b_i \phi_i(x).$$

Taking their inner product, using weights $1/\lambda_i$, results in

$$\langle f, g \rangle = \sum_{i=1}^{\infty} \frac{a_i b_i}{\lambda_i}.$$

If we take the (weighted) inner product of $K(x, \cdot)$ with $f(\cdot)$, and note that the $\phi(\cdot)$ functions are orthogonal, it can be shown that

$$\langle K(x, y), f(y) \rangle = \sum_{i=1}^{\infty} \frac{a_i \lambda_i \phi_i(x)}{\lambda_i} = \sum_{i=1}^{\infty} a_i \phi_i(x) = f(x),$$

illustrating the ‘magical’ reproducing property of the kernel: $\langle K(x, y), f(y) \rangle = f(x)$. More rigorous definitions of Mercer’s theorem and the mathematics of RKHS are available elsewhere [41, 42, 44, 107].

Appendix 2: Score Statistics and Kernel Matrices

Denote the mean of y_i conditional on random effects $E[y_i | \gamma] = \mu_i$, and the conditional variance as $\text{Var}(y_i | \gamma) = V(\mu_i) = a_i^{-1} \phi(\mu_i)$, where a_i is a known weight, such as a denominator for binomial data, ϕ is the scale parameter and $\nu(\mu_i)$ is a variance function; see Appendix table 1 for commonly used functions.

If the SNP genotypes are encoded in a design vector Z_i , then the usual GLMM can be expressed hierarchically as

$$\begin{aligned} g(\mu_i) &= x_i' \beta + z_i' \gamma \\ \gamma &\sim N(0, \sigma_K^2 K) \end{aligned}$$

where $g(\mu_i)$ is a link function, β is a p -dimensional vector of fixed effects for the covariates, γ is a vector of random effects of the SNPs, K is a kernel matrix representing the structured covariance of the γ 's – their 'prior' similarities, and σ_K^2 is a variance component for this kernel. For a LMM of a quantitative trait, the corresponding marginal model is $Y \sim N(X'\beta, \sigma_K^2 ZKZ' + \sigma_e^2 I)$ where X is a $n \times p$, Z is a $n \times q$, and K is a $q \times q$ kernel matrix. Note that we used σ_e^2 in place of ϕ , to emphasize that the scale parameter measures the residual variance.

In this setup, the genotype similarity between pairs of subjects is captured by $S = ZKZ'$; Z maps the SNPs measured on subjects (rows) to coded effects on the trait (columns), and the kernel matrix K measures similarity for pairs of columns of Z , the SNPs. With genotypes structured as haplotypes, Tzeng and Zhang [7] used Z to count distinguishable haplotypes and K to measure the similarity between pairs of haplotypes. Alternatively, Z could be the identity matrix, in which case the modeling of the SNP effects is completely nonparametric and fully specified by the kernel matrix. Kwee et al. chose this approach by specifying a kernel matrix according to the number of alleles AIS at tagSNPs for pairs of subjects [8]. Since we use K to denote the matrix of similarities for pairs of columns of Z , we use $S = ZKZ'$ to denote the matrix of similarities for pairs of subjects. When Z is the identity matrix, S is a direct measure of similarity for pairs of subjects.

The null hypothesis of no effect of G on Y , $H_0: \gamma = 0$, is equivalent to testing no variance component, $H_0: \sigma_K^2 = 0$, meaning that the marginal covariance matrix $V(Y)$ does not depend on G . The derivation of the score statistic for a GLMM depends on replacing $(y_i = \mu_i)$ with its linear approximation, since μ_i can depend on a nonlinear link function. This is achieved by $\delta_i = g'(\mu_i)$, the derivative of $g(\mu_i)$ with respect to μ_i . The derivation also entails use of weighted least squares, because the variance of Y is not necessarily constant. This is achieved by $w_i = [V(\mu_i)\{g'(\mu_i)\}^2]^{-1}$, the generalized linear model working weight in the iteratively weighted least squares algorithm. Using the diagonal matrices $\Delta = \text{diag}\{\delta_i\}$ and $W = \text{diag}\{w_i\}$, the score statistic to test $H_0: \sigma_K^2 = 0$ can be expressed as

$$T_K = \frac{1}{2}(Y - \hat{\mu})' \Delta W S W \Delta (Y - \hat{\mu})$$

where $\hat{\mu} = g^{-1}(X\hat{\beta})$, $\hat{\beta}$ is the maximum likelihood estimate under the null, $\Delta = \text{diag}[g'(\hat{\mu}_i)]$, $W = \text{diag}(w_i)$, with $w_i = \{\hat{\phi} a_i^{-1} \nu(\hat{\mu}_i) [g'(\hat{\mu}_i)]^2\}^{-1}$, and with $\hat{\phi}$ a REML type of estimate. Details of the derivations can be found elsewhere [7, 95]. A computational advantage of this approach is that one only needs to fit a generalized linear model for the fixed effects.

The score statistic can be expressed more simply by noting that ΔW is a diagonal matrix with elements $a_i / [\phi \nu(\mu_i) g'(\mu_i)]$, and that $g'(\mu_i) = 1/\nu(\mu_i)$, so that $\Delta W = \text{diag}(a_i)/\phi$. If all weights are $a_i = 1$, a common situation, then

$$T_K = \frac{1}{2\phi^2}(Y - \hat{\mu})' S (Y - \hat{\mu})$$

For a quantitative trait with a normal distribution, the residual mean squared error from a regression on the fixed effects can be used, $\hat{\phi} = \hat{\phi}_{mse}^2$. For binomial and Poisson data, $\phi = 1$.

The T_K score statistic has a complicated mixture of χ_1^2 distributions. For a LMM with only an intercept, the mixing probabilities depend on the sample size, while for more general models with covariates, and for GLMMs, the mixing probabilities are dif-

Appendix Table 2. Moments of T_K and U_ϕ under the null hypothesis^a

$$\begin{aligned} E[T_K] &= \frac{1}{2} \text{tr}(PK) \\ \text{Var}(T_K) &= \frac{1}{2} \text{tr}(PKPK) + \frac{1}{4} \sum_{i=1}^n A_{ii}^2 (\mu_{4i} - 3\mu_{2i}^2) (\phi a_i^{-1})^3 \\ \text{Var}(U_\phi) &= \frac{1}{2\phi^2} \text{tr}(PW^{-1}) + \frac{1}{4\phi^2} \sum_{i=1}^n B_{ii}^2 (\mu_{4i} - 3\mu_{2i}^2) (\phi a_i^{-1})^3 \\ \text{Cov}(T_K, U_\phi) &= \frac{1}{2\phi} \text{tr}(PK) + \frac{1}{4\phi} \sum_{i=1}^n A_{ii} B_{ii} (\mu_{4i} - 3\mu_{2i}^2) (\phi a_i^{-1})^3 \\ \text{Cov}(T_{K_1}, T_{K_2}) &= \frac{1}{2} \text{tr}(PK_1 PK_2) + \frac{1}{4} \sum_{i=1}^n A_{1,ii} A_{2,ii} (\mu_{4i} - 3\mu_{2i}^2) (\phi a_i^{-1})^3 \end{aligned}$$

^a T_K the score statistic to test $H_0: \sigma_K^2 = 0$.

$\Delta = \text{diag}[g'(\hat{\mu}_i)]$; $W = \text{diag}(w_i)$, with $w_i = \{\hat{\phi} a_i^{-1} \nu(\hat{\mu}_i) [g'(\hat{\mu}_i)]^2\}^{-1}$

$P = W - WX(X'WX)^{-1}X'W$, the null projection matrix

K, K_1, K_2 are kernel matrices

$A = \Delta PKP\Delta$ and $B = \Delta P\Delta$

μ_{4i} and μ_{2i} are the 4th and 2nd central moments.

ficult to obtain [108]. To approximate the distribution of T_K , the Satterthwaite procedure is used [7, 8, 76, 91, 92], which equates the null mean and variance of T_K with a scaled χ^2 distribution.

Tzeng and Zhang [7] derived the mean and variance of the score equations under the null hypothesis, but their derivations are correct only if the trait has a normal distribution (they ignore the influence of kurtosis). In contrast, others have derived moments of the score statistics to account for nonnormal data [80, 91, 95]. A key ingredient to this derivation is the covariance of two quadratic forms. If A and B are matrices, then

$$\begin{aligned} \text{Cov}(Y'AY, Y'BY) \\ = 2\text{tr}(AVBV) + \sum_{i=1}^n A_{ii} B_{ii} (\mu_{4i} - 3\mu_{2i}^2) (\phi a_i^{-1})^3, \end{aligned} \quad (1)$$

(where $V = \text{Var}(Y)$ and $\text{tr}(\cdot)$ is the trace of a matrix. The second summand in equation (1) accounts for kurtosis (the peakedness of the distribution) through the 4th-order cumulant, $\kappa_{4i} = (\mu_{4i} - 3\mu_{2i}^2)(\phi a_i^{-1})^3$, where μ_{4i} and μ_{2i} are the 4th and 2nd central moments. The factor $(\phi a_i^{-1})^3$ is needed only when there is cluster sampling and over-/underdispersion. This kurtosis correction is not needed for normally distributed data ($\kappa_{4i} = 0$). In contrast, for binomial data, $(\mu_{4i} - 3\mu_{2i}^2) = \mu_i(1 - \mu_i)(1 + 6\mu_i^2 - 6\mu_i)$, which is negative over the approximate range $0.2 \leq \mu_i \leq 0.8$, reaching its most extreme when $\mu_i = 0.5$. This means that correction for kurtosis is most needed for case-control data with approximately equal numbers of cases and controls. For Poisson data, $(\mu_{4i} - 3\mu_{2i}^2) = \mu_i$.

The moments of the score statistics under the null hypothesis are summarized in Appendix table 2. These moments account for the expected values and variances of both the kernel score statistic and the scale score statistic, as well as their covariances. Terms involving the scale score statistic are used to correct for estimating the scale variable. When the GLMM scale variable is not esti-

mated, say $\phi = 1$ for case-control data, the moments for T_K can be used directly from Appendix table 2. But, when $\hat{\phi}$ is estimated, such as for a quantitative normal trait, the efficient variance for T_K that accounts for the covariance between the score equation T_K and the score equation for ϕ (denoted U_ϕ) is $\text{Var}(T_K) - \text{Cov}(T_K, U_\phi)^2 / \text{Var}(U_\phi)$ (see Appendix table 2 for these terms).

To use the Satterthwaite procedure to approximate the distribution of T_K with a scaled χ^2 distribution, the expected value and variance of T_K under the null, e_0 and ν_0 , are set equal to the mean and variance of a scaled χ^2 random variable, $\mu = \delta d$ and $\sigma^2 = 2\delta^2 d$, respectively. This results in scale parameter $\delta = \nu_0 / 2e_0$ and degrees of freedom $d = 2e_0^2 / \nu_0$. The scaled score statistic, $T_K^* = T_K / \delta$, can then be compared to a χ^2 distribution with d degrees of freedom.

When analyzing data with multiple kernel matrices, it is important to account for the correlations of the resulting score statistics. Appendix table 2 presents the covariance between score statistics T_{K_1} and T_{K_2} based on two different kernels, K_1 and K_2 . When using multiple kernel matrices on the same data, this provides a way to build a covariance matrix of all score statistics in order to account for their dependencies when performing hypothesis tests and correcting for multiple testing. For example, it may be desirable to create kernels for different parts of the genome, perhaps different chromosomes, or even a series of kernels for a sliding window over a chromosome. Another example is creating kernels for different biological pathways, some of which could have overlapping SNPs. To approximate the distribution of multiple kernel score statistics, the procedure outlined by le Cessie and van Houwelingen can be used [91]. After scaling the score statistics and estimating their degrees of freedom, say $T_{K_i}^*$ and d_i for the i -th kernel matrix, the following transformation has an approximate standard normal distribution,

$$z_{K_i}(T_{K_i}^*, d_i) = \left[\left(\frac{T_{K_i}^*}{d_i} \right)^{1/3} + \frac{2}{9d_i} - 1 \right] \left(\frac{9d_i}{2} \right)^{1/2}.$$

Using a Taylor-series expansion, it can be shown that the correlation of the z_{K_i} 's is approximately the same as the correlation for the $T_{K_i}^*$'s. This allows use of a multivariate normal distribution to compute the distribution of the most extreme z_{K_i} , either numerically or by repeated sampling from a multivariate normal distribution.

A nice feature of the score statistic, emphasized by others [7, 8, 76, 91, 92], is that it can be expressed as

$$\begin{aligned} T_K &= \frac{1}{2\phi^2} (Y - \hat{\mu})' S (Y - \hat{\mu}) \\ &= \frac{1}{2\phi^2} \sum_i \sum_j (y_i - \hat{\mu}_i) S_{ij} (y_j - \hat{\mu}_j) \\ &= \frac{1}{2\phi} \sum_i \frac{(y_i - \hat{\mu}_i)}{\sqrt{\phi}} \sum_j S_{ij} \frac{(y_j - \hat{\mu}_j)}{\sqrt{\phi}}. \end{aligned} \quad (2)$$

So, recognizing $(y_i - \hat{\mu}_i) / \sqrt{\phi}$ as a standardized residual, we can view

$$\sum_j S_{ij} \frac{(y_j - \hat{\mu}_j)}{\sqrt{\phi}}$$

as a smoothed standardized residual, smoothed by the kernel matrix, in the neighborhood of the i -th subject. Large values of this smoothed residual could indicate highly influential subjects on the score statistic. Furthermore, the second step in expression (2) illustrates that the score statistic measures the association of the covariance of a trait, $(y_i - \hat{\mu}_i)(y_j - \hat{\mu}_j)$, with terms in the similarity matrix, suggesting that plotting the values of the similarity matrix (above the diagonal) against the corresponding values of $(y_i - \hat{\mu}_i)(y_j - \hat{\mu}_j)$ is a nice way to view how the association behaves, and whether there are large influential points.

Appendix 3: Proofs for Building Positive Semidefinite Kernel Matrices

To ensure that a newly devised kernel matrix is psd – i.e.

$$\sum_i \sum_j c_i c_j K(x_i, x_j) \geq 0$$

for all real vectors c – this appendix illustrates some of the tricks of the trade by reviewing proofs of psd for seven different kernels, showing that a ‘proof’ may merely require showing that a new kernel can be expressed in the form of a kernel already proven to be psd.

For the first proof, consider the dot-product kernel, and the following steps that show that it is psd:

$$\begin{aligned} \sum_i \sum_j c_i c_j K(x_i, x_j) &= \sum_i \sum_j c_i c_j \langle x_i, x_j \rangle \\ &= \sum_i \sum_j c_i c_j \sum_k x_{ik} x_{jk} = \sum_k \left(\sum_i c_i x_{ik} \right) \left(\sum_j c_j x_{jk} \right) = \sum_k \left(\sum_i c_i x_{ik} \right)^2 \geq 0 \end{aligned}$$

The final inequality follows because of the sum of squared terms.

To prove that a product of functions is psd (second proof), note that if we replace $x_{ik} x_{jk}$ in the preceding dot product kernel with a product of real functions, $f_k(x_{ik}) f_k(x_{jk})$ (allowing the function to differ across attributes), we would still have a sum of squared terms, completing the proof.

For the third proof, note that if we use a function to map all attributes to a single real number, say $f(x_i)$, then the proof is similar to that above:

$$\begin{aligned} \sum_i \sum_j c_i c_j K(x_i, x_j) &= \sum_i \sum_j c_i c_j f(x_i) f(x_j) \\ &= \left(\sum_i c_i f(x_i) \right) \left(\sum_j c_j f(x_j) \right) = \left(\sum_i c_i f(x_i) \right)^2 \geq 0 \end{aligned}$$

For the fourth proof, note that the dot-product kernel can be generalized by using a symmetric psd matrix B to weight the dot product, $K(x_i, x_j) = x_i' B x_j$. If B is psd, then the generalized Cholesky decomposition is $B = LL'$, so $x_i' B x_j = x_i' L L' x_j = z_i' z_j$, also a dot product, completing the proof. This approach provides a method to weight the attributes that go into the total similarity across all attributes, either individual weights per attribute, (using a diagonal matrix for B), even setting some weights to 0 to exclude some attributes, or weights for linear combinations of attributes. This type of weighting, for example, has been used to create weighted distances for cluster analysis [109].

The fifth proof, for the polynomial kernel, is simple: polynomials obey the operations of multiplying kernels by positive scalars and multiplying kernels together (i.e. raised to a power), so polynomials are valid kernels.

The sixth proof, taking the exponent of a kernel, uses the polynomial proof. The power series of e^z is an infinite polynomial, with coefficients $a_i = 1/i!$. The limit of this series exists because kernels are closed for point-wise limits [44]. So, because $\exp\{K(x_i, x_j)\}$ can be expressed as a polynomial, it too is a valid kernel.

The seventh and final proof is for the gaussian kernel [44], $K(x_i, x_j) = \exp(-\|x_i - x_j\|^2/\sigma^2)$. First note that

$$\begin{aligned}\|x_i - x_j\|^2 &= \sum (x_{ik} - x_{jk})^2 \\ &= \sum x_{ik}^2 + \sum x_{jk}^2 - 2\sum x_{ik}x_{jk} \\ &= \|x_i\|^2 + \|x_j\|^2 - 2\langle x_i, x_j \rangle\end{aligned}$$

Multiplying both sides of the equation by $-1/\sigma^2$ results in

$$-\|x_i - x_j\|^2/\sigma^2 = -\|x_i\|^2/\sigma^2 - \|x_j\|^2/\sigma^2 + 2\langle x_i, x_j \rangle/\sigma^2.$$

Applying the exponent function to both sides results in

$$\begin{aligned}\exp(-\|x_i - x_j\|^2/\sigma^2) \\ = \exp(-\|x_i\|^2/\sigma^2)\exp(-\|x_j\|^2/\sigma^2)\exp(2\langle x_i, x_j \rangle/\sigma^2).\end{aligned}$$

The first product on the right-hand side, $K_1(x_i - x_j) = \exp(-\|x_i\|^2/\sigma^2)\exp(-\|x_j\|^2/\sigma^2)$, is a valid kernel, because it is a product of functions. The third term, $K_2(x_i - x_j) = \exp(2\langle x_i, x_j \rangle/\sigma^2)$, is a valid kernel, because the dot product (a valid kernel) is scaled by a positive value, and the exponent function results in a valid kernel. Finally, the product of kernels, K_1K_2 , creates a valid kernel.

References

- Schaid D: Genomic similarity and kernel methods II: methods for genomic information. *Hum Hered* 2010;70:132–140.
- Li Y, Sung WK, Liu JJ: Association mapping via regularized regression analysis of single-nucleotide polymorphism haplotypes in variable-sized sliding windows. *Am J Hum Genet* 2007;80:705–715.
- Conti D, Witte J: Hierarchical modeling of linkage disequilibrium: genetic structure and spatial relations. *Am J Hum Genet* 2003;72:351–363.
- Chen GK, Witte JS: Enriching the analysis of genomewide association studies with hierarchical modeling. *Am J Hum Genet* 2007;81:397–404.
- Malo N, Libiger O, Schork NJ: Accommodating linkage disequilibrium in genetic-association analyses via ridge regression. *Am J Hum Genet* 2008;82:375–385.
- Wessel J, Schork NJ: Generalized genomic distance-based regression methodology for multilocus association analysis. *Am J Hum Genet* 2006;79:792–806.
- Tzeng JY, Zhang D: Haplotype-based association analysis via variance-components score test. *Am J Hum Genet* 2007;81:927–938.
- Kwee LC, Liu D, Lin X, Ghosh D, Epstein MP: A powerful and flexible multilocus association test for quantitative traits. *Am J Hum Genet* 2008;82:386–397.
- Yu J, Pressoir G, Briggs WH, Vroh Bi I, Yamasaki M, Doebley JF, McMullen MD, Gaut BS, Nielsen DM, Holland JB, Kresovich S, Buckler ES: A unified mixed-model method for association mapping that accounts for multiple levels of relatedness. *Nat Genet* 2006;38:203–208.
- Thompson E: *Pedigree Analysis in Human Genetics*. Baltimore, The Johns Hopkins University Press, 1986.
- Whittemore AS, Halpern J: Probability of gene identity by descent: computation and applications. *Biometrics* 1994;50:109–117.
- Whittemore AS: Genome scanning for linkage: an overview. *Am J Hum Genet* 1996;59:704–716.
- Weir BS, Anderson AD, Hepler AB: Genetic relatedness analysis: modern data and new challenges. *Nat Rev Genet* 2006;7:771–780.
- Tzeng JY, Devlin B, Wasserman L, Roeder K: On the identification of disease mutations by the analysis of haplotype similarity and goodness of fit. *Am J Hum Genet* 2003;72:891–902.
- Tzeng JY, Byerley W, Devlin B, Roeder K, Wasserman L: Outlier detection and false discovery rates for whole-genome DNA matching. *J Am Statist Assoc* 2003;98:236–246.
- Bourgain C, Genin E, Ober C, Clerget-Darpoux F: Missing data in haplotype analysis: a study on the MILC method. *Ann Hum Genet* 2002;66:99–108.
- Bourgain C, Genin E, Holopainen P, Mustalahti K, Maki M, Partanen J, Clerget-Darpoux F: Use of closely related affected individuals for the genetic study of complex diseases in founder populations. *Am J Hum Genet* 2001;68:154–159.
- Bourgain C, Genin E, Quesneville H, Clerget-Darpoux F: Search for multifactorial disease susceptibility genes in founder populations. *Ann Hum Genet* 2000;64:255–265.
- Zhang S, Sha Q, Chen HS, Dong J, Jiang R: Transmission/disequilibrium test based on haplotype sharing for tightly linked markers. *Am J Hum Genet* 2003;73:566–579.
- Yu K, Zhang S, Borecki I, Kraja A, Xiong C, Myers R, Province M: A haplotype similarity based transmission/disequilibrium test under founder heterogeneity. *Ann Hum Genet* 2005;69:455–467.
- Durrant C, Zondervan KT, Cardon LR, Hunt S, Deloukas P, Morris AP: Linkage disequilibrium mapping via cladistic analysis of single-nucleotide polymorphism haplotypes. *Am J Hum Genet* 2004;75:35–43.
- Lange EM, Boehnke M: The haplotype runs test: the parent-parent-affected offspring trio design. *Genet Epidemiol* 2004;27:118–130.
- Yuan A, Yue Q, Apprey V, Bonney G: Detecting disease gene in DNA haplotype sequences by nonparametric dissimilarity test. *Hum Genet* 2006;120:253–261.
- Müller T, Selinski S, Ickstadt K: Cluster analysis: a comparison of different similarity measures for SNP data: Technical Report 14095, available at <http://opuszbw-kiel.de/volltexte/2005/3389/pdf/tr14-05pdf> (accessed July 23, 2008), Universität Dortmund, 2005.
- Selinski S: Similarity measures for clustering SNP and epidemiological data: Technical Report, available at <http://www.sfb475.uni-dortmund.de/berichte/tr25-06pdf> (accessed July 23, 2008), Universität Dortmund, 2006, p 34.
- Selinski S, Ickstadt K: Similarity measures for clustering SNP data: Technical Report, available at <http://opuszbw-kiel.de/volltexte/2006/3976/> (accessed July 23, 2008), Universität Dortmund, 2006.
- Boriah S, Chandola V, Kumar V: Similarity measures for categorical data: a comparative evaluation. *Proceedings of the 2008 SIAM International Conference on Data Mining*. Atlanta, GA, Society for Industrial and Applied Mathematics, available at http://www.siam.org/proceedings/datamining/2008/dm08_22_Boriah.pdf (accessed August 7, 2008), 2008.
- Reif J, Melchinger A, Frisch M: Genetical and mathematical properties of similarity and dissimilarity coefficients applied in plant breeding and seed bank management. *Crop Sci* 2005;45:1–7.

- 29 Gonzalez-Recio O, Gianola D, Long N, Weigel KA, Rosa GJ, Avendano S: Nonparametric methods for incorporating genomic information into genetic evaluations: an application to mortality in broilers. *Genetics* 2008;178:2305–2313.
- 30 Needleman SB, Wunsch CD: A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J Mol Biol* 1970;48:443–453.
- 31 Everitt B, Landau S, Leese M: *Cluster Analysis*, ed 4. London, Arnold, 2001.
- 32 von Luxburg U: A tutorial on spectral clustering: Technical Report No. TR-149, available at http://www.wkybmgde/publications/attachments/Luxburg06_TR_%5B0%5Dpdf (accessed October 15, 2008). Tuebingen, Max-Planck Institute for Biological Cybernetics, 2006.
- 33 Roeder K, Lee A, Luca D: Modeling ancestry in structured populations using spectral analysis (abstract 59). Abstracts 58th Annu Meet Amer Soc Hum Genet, Philadelphia, November 2008, p 46.
- 34 Schaid DJ, McDonnell SK, Hebring SJ, Cunningham JM, Thibodeau SN: Nonparametric tests of association of multiple genes with human disease. *Am J Hum Genet* 2005;76:780–793.
- 35 Beckmann L, Thomas D, Fischer C, Chang-Claude J: Haplotype sharing analysis using mantel statistics. *Hum Hered* 2005;59:67–78.
- 36 Sha Q, Chen HS, Zhang S: A new association test using haplotype similarity. *Genet Epidemiol* 2007;31:577–593.
- 37 Tzeng J, Chang S, Zhang D, Thomas D, Davidian M: Regression-based multi-marker analysis for genome-wide association studies using haplotype similarity. Available at <http://www4.stat.ncsu.edu/~tzeng/research.html> (accessed August 1, 2008).
- 38 Grimes L, Harvey W: Estimation of genetic variances and covariances using symmetric differences squared. *J Anim Sci* 1980;50:634–644.
- 39 Thomas SC: The estimation of genetic relationships using molecular markers and their efficiency in estimating heritability in natural populations. *Philos Trans R Soc Lond B Biol Sci* 2005;360:1457–1467.
- 40 Ritland K: A marker-based method for inferences about quantitative inheritance in natural populations. *Evolution* 1996;50:1062–1073.
- 41 Schölkopf B, Smola A: *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge MA, MIT Press, 2002.
- 42 Cristianini N, Shawe-Taylor J: *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge, UK, Cambridge University Press, 2000.
- 43 Ben-Hur A, Ong CS, Sonnenburg S, Schölkopf B, Rätsch G: Support vector machines and kernels for computational biology. *PLoS Comput Biol* 2008;4:e1000173.
- 44 Hofmann T, Schölkopf B, Smola A: Kernel methods in machine learning. *Ann Stat* 2008;36:1171–1220.
- 45 Price AL, Patterson NJ, Plenge RM, Weinblatt ME, Shadick NA, Reich D: Principal components analysis corrects for stratification in genome-wide association studies. *Nat Genet* 2006;38:904–909.
- 46 Patterson N, Price AL, Reich D: Population structure and eigenanalysis. *PLoS Genet* 2006;2:e190.
- 47 Homer N, Szelinger S, Redman M, Duggan D, Tembe W, Muehling J, Pearson JV, Stephan DA, Nelson SF, Craig DW: Resolving individuals contributing trace amounts of DNA to highly complex mixtures using high-density SNP genotyping microarrays. *PLoS Genet* 2008;4:e1000167.
- 48 Gower J: Metric and euclidean properties of dissimilarity coefficients. *J Classification* 1986;3:5–48.
- 49 Wahba G: Smoothing splines in nonparametric regression; in El-Shaarawi A, Piegorsch W (eds): *Encyclopedia of Environmental Metrics*. New York, Wiley, 2001, vol 4, pp 2099–2112.
- 50 Haasdonk B: Feature space interpretation of SVMs with indefinite kernels. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 2005;27:482–492.
- 51 Kimeldorf G, Wahba G: Some results on tchebycheffian spline functions. *J Math Anal Appl* 1971;33:82–95.
- 52 Efron B, Hastie T, Johnstone I, Tibshirani R: Least angle regression. *Ann Statist* 2004;32:407–499.
- 53 Hastie T, Tibshirani R, Friedman J: *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York, Springer, 2001.
- 54 Friedman J, Hastie T, Tibshirani R: Regularization paths for generalized linear models via coordinate descent. Available at <http://www-statstanford.edu/~jhf/ftp/glmnetpdf> (accessed October 16, 2008).
- 55 Friedman J, Hastie T, Hofling H, Tibshirani R: Pathwise coordinate optimization. *Ann Appl Stat* 2007;1:302–332.
- 56 Wu T, Lange K: Coordinate descent algorithms for lasso penalized regression. *Ann Appl Stat* 2008;2:224–244.
- 57 Wahba G: *Spline Models for Observational Data*. Philadelphia, Society for Industrial and Applied Mathematics, 1990.
- 58 Tibshirani R: Regression shrinkage and selection via the lasso. *J R Statist Soc B* 1996;58:267–288.
- 59 Hesterberg T, Choi N, Meier L, Fraley C: Least angle and L1 regression: a review. *Statistics Surveys* 2008;2:61–93.
- 60 Fan J, Li R: Variable selection via nonconcave penalized likelihood and its oracle properties. *J Am Stat Assoc* 2001;96:1348–1360.
- 61 Zou H: The adaptive lasso and its oracle properties. *Am Statist Assoc* 2006;101:1418–1429.
- 62 Zou H, Tibshirani R: Regularization and variable selection via the elastic net. *J R Statist Soc B* 2005;67:301–320.
- 63 Tibshirani R, Saunders M, Rosset S, Zhu J, Knight K: Sparsity and smoothness via the fused lasso. *J R Statist Soc B* 2005;67:91–108.
- 64 Tibshirani R, Wang P: Spatial smoothing and hot spot detection for CGH data using the fused lasso. *Biostatistics* 2008;9:18–29.
- 65 Hoggart CJ, Whittaker JC, De Iorio M, Balding DJ: Simultaneous analysis of all SNPs in genome-wide and re-sequencing association studies. *PLoS Genet* 2008;4:e1000130.
- 66 Zhao P, Yu B: On model selection consistency of lasso. *J Machine Learning Research* 2006;7:2542–2563.
- 67 Gianola D, van Kaam JB: Reproducing kernel Hilbert spaces regression methods for genomic assisted prediction of quantitative traits. *Genetics* 2008;178:2289–2303.
- 68 Pearce N, Wand M: Penalised splines and reproducing kernel methods. *Am Stat* 2006;60:233–240.
- 69 Cristianini N, Shawe-Taylor J: *An Introduction to Support Vector Machines (and Other Kernel-Based Learning Methods)*. Cambridge, UK, Cambridge University Press, 2000.
- 70 O'Sullivan F, Yandell B, Raynor W: Automatic smoothing of regression functions in generalized linear models. *J Am Stat Assoc* 1986;81:96–103.
- 71 Green P, Yandell B: Semi-parametric generalized linear models; in Brillinger D, Fienberg S, Gani J, Hartigan J, Krickeberg K (eds): *Lecture Notes in Statistics: Generalized Linear Models: Proceedings of the GLIM 85 Conference*. Berlin, Springer-Verlag, 1985, vol 32, pp 44–55.
- 72 Park MY, Hastie T: L1-regularization path algorithm for generalized linear models. *J R Statist Soc B* 2007;69:659–677.
- 73 Berline A, Thomas-Agnan C: *Reproducing Kernel Hilbert Spaces in Probability and Statistics*. Boston, Kluwer Academic Publishers, 2004.
- 74 Cristianini N, Schölkopf B: Support vector machines and kernel methods: the new generation of learning machines. *AI Magazine* 2002;23:31–41.
- 75 Hsiang T: A bayesian view on ridge regression. *Statistician* 1975;24:267–268.
- 76 Liu D, Lin X, Ghosh D: Semiparametric regression of multidimensional genetic pathway data: Least-squares kernel machines and linear mixed models. *Biometrics* 2007;63:1079–1088.
- 77 Wand M: Smoothing and mixed models. *Comput Stat* 2003;18:223–249.
- 78 Ngo L, Wand M: Smoothing with mixed model software. *J Stat Softw* 2004;9:1–54.
- 79 Laird N, Ware J: Random-effects models for longitudinal data. *Biometrics* 1982;38:963–974.
- 80 Searle S, Casella G, McCulloch C: *Variance Components*. New York, John Wiley and Sons, Inc., 1992.

- 81 Greenland S: Hierarchical regression for epidemiologic analyses of multiple exposures. *Environ Health Perspect* 1994;102:33–39.
- 82 Greenland S: Principles of multilevel modeling. *Int J Epidemiol* 2000;29:158–167.
- 83 Greenland S: Second-stage least squares versus penalized quasi-likelihood for fitting hierarchical models in epidemiologic analyses. *Stat Med* 1997;16:515–526.
- 84 Witte JS, Greenland S, Kim LL, Arab L: Multilevel modeling in epidemiology with GLIMMIX. *Epidemiology* 2000;11:684–688.
- 85 Wolfinger R, O'Connell M: Generalized linear mixed models: a pseudo-likelihood approach. *J Stat Comp Simul* 1993;15:1294–1310.
- 86 Breslow N, Clayton D: Approximate inference in generalized linear mixed models. *J Am Stat Assoc* 1993;88:9–25.
- 87 Lin X, Breslow NE: Bias correction in generalized linear mixed models with multiple components of dispersion. *J Am Stat Assoc* 1996;91:1007–1016.
- 88 Breslow N, Lin X: Bias correction in generalized linear mixed models with a single component of dispersion. *Biometrika* 1995;82:81–91.
- 89 Frank I, Freidman J: A statistical view of some chemometrics regression tools. *Technometrics* 1993;35:109–148.
- 90 Gianola D, Fernando RL, Stella A: Genomic-assisted prediction of genetic value with semiparametric procedures. *Genetics* 2006;173:1761–1776.
- 91 le Cessie S, van Houwelingen HC: Testing the fit of a regression model via score tests in random effects models. *Biometrics* 1995;51:600–614.
- 92 Goeman JJ, van de Geer SA, de Kort F, van Houwelingen HC: A global test for groups of genes: testing association with a clinical outcome. *Bioinformatics (Oxford, England)* 2004;20:93–99.
- 93 Goeman JJ, Oosting J, Cleton-Jansen AM, Anninga JK, van Houwelingen HC: Testing association of a pathway with survival using gene expression data. *Bioinformatics (Oxford, England)* 2005;21:1950–1957.
- 94 Goeman J, van de Geer S, van Houwelingen H: Testing against a high dimensional alternative. *J R Stat Soc Ser B* 2006;68:477–493.
- 95 Lin X: Variance component testing in generalised linear models with random effects. *Biometrika* 1997;84:309–326.
- 96 Williams J, Blangero J: Power of variance component linkage analysis to detect quantitative trait loci. *Ann Hum Genet* 1999;63:545–563.
- 97 Chapman J, Whittaker J: Analysis of multiple SNPs in a candidate gene or region. *Genetic Epidemiol* 2008;32:560–566.
- 98 Sollich P: Bayesian methods for support vector machines: evidence and predictive class probabilities. *Mach Learn* 2002;46:21–52.
- 99 Gelman A, Carlin J, Stern H, Rubin D: *Bayesian Data Analysis*, ed 2. New York, Chapman & Hall, 2004.
- 100 McArdle B, Anderson M: Fitting multivariate models to community data: a comment on distance-based redundancy analysis. *Ecology* 2001;82:290–297.
- 101 Anderson MJ: A new method for non-parametric multivariate analysis of variance. *Austral Ecology* 2001;26:32–46.
- 102 Anderson MJ: Distance-based tests for homogeneity of multivariate dispersions. *Biometrics* 2006;62:245–253.
- 103 Legendre P, Anderson M: Distance-based redundancy analysis: Testing multispecies responses in multifactorial ecological experiments. *Ecol Mongr* 1999;69:1–24.
- 104 Wessel J, Zapala MA, Schork NJ: Accommodating pathway information in expression quantitative trait locus analysis. *Genomics* 2007;90:132–142.
- 105 Nievergelt CM, Libiger O, Schork NJ: Generalized analysis of molecular variance. *PLoS Genet* 2007;3:e51.
- 106 Zapala MA, Schork NJ: Multivariate regression analysis of distance matrices for testing associations between gene expression patterns and related variables. *Proc Natl Acad Sci USA* 2006;103:19430–19435.
- 107 Wahba G: An introduction to model building with reproducing kernel spaces: Technical Report No 1020, Dept. of Statistics, University of Washington, 2000.
- 108 Fitzmaurice GM, Lipsitz SR, Ibrahim JG: A note on permutation tests for variance components in multilevel generalized linear mixed models. *Biometrics* 2007;63:942–946.
- 109 Gnanadesikan R, Kettenring J: Weighting and selection of variables for cluster analysis. *J Classification* 1995;12:113–136.
- 110 Zhu J, Rosset S, Hastie T, Tibshirani R: *1-Norm Support Vector Machines*; Neural Information Processing Systems. Cambridge, MA, MIT Press, 2003.