Supplemental Material for Multi-Kernel Linear Mixed Models for Complex Phenotype Prediction

Supplemental tables

Table S1: **Sample sizes and number of SNPs for the evaluated WTCCC data sets**. The diseases are CD (Crohn's disease), T1D (type 1 diabetes), BD (bipolar disorder), RA (rheumatoid arthritis), T2D (type 2 diabetes), CAD (coronary artery disease) HT (hypertension) and UC (ulcerative colitis). Control individuals are divided into the UK national blood service (NBS) controls and the 1958 British birth cohort (C58) controls.

	#SNPs	#cases	#NBS controls	#C58 controls
CD	285650	1720	1451	1474
T1D	286237	1957	1454	1741
BD	284208	1856	1450	1745
RA	287884	1850	1453	1742
T2D	286339	1906	1453	1742
CAD	288544	1910	1451	1474
HT	281898	1932	1455	1470
UC	458560	2697	2801	2851

Table S2: **Analysis results on WTCCC data**. The table is similar to Table 1 in the main text, but also includes mean negative out of sample log likelihood (NOOS LL) results, computed heuristically as the probability of having a non-negative phenotype according to the posterior phenotype distribution of each individual. We emphasize that such values should be regarded with caution, as they do not take the ascertainment procedure into account. Traits marked with an asterisk are ones where the MHC region was excluded from the analysis.

Trait	MKLMM-Adapt		MKLMM-Poly2		AMB		GBLUP	
	AUC	NOOS LL	AUC	NOOS LL	AUC	NOOS LL	AUC	NOOS LL
CD	**0.667±0.010**	**0.645±0.005**	0.650±0.010	**0.651±0.005**	0.645±0.011	0.657±0.004	0.582±0.013	0.680±0.003
T1D	**0.886±0.004**	**0.434±0.006**	**0.885±0.003**	**0.437±0.005**	0.883±0.004	0.443±0.006	0.601±0.008	0.670±0.004
BD	0.563±0.011	0.681±0.004	0.571±0.012	0.677±0.005	0.568±0.011	0.681±0.004	0.578±0.011	0.679±0.002
RA	0.750±0.009	0.592±0.007	0.749±0.009	0.589±0.006	0.752±0.010	0.589±0.007	0.671±0.009	0.643±0.003
T2D	0.634±0.007	0.655±0.004	0.632±0.008	0.656±0.004	0.634±0.007	0.655±0.004	0.598±0.009	0.672±0.003
CAD	0.698±0.014`	0.624±0.007	0.697±0.014	0.621±0.007	0.699±0.013	0.621±0.007	0.701±0.015	0.621±0.008
HT	0.611±0.003	0.662±0.002	0.610±0.003	0.660±0.002	0.611±0.003	0.658±0.003	0.576±0.005	0.675±0.001
UC	**0.601±0.007**	**0.676±0.004**	0.590±0.003	0.689±0.001	0.585±0.002	0.683±0.001	0.583±0.004	0.684±0.001
CD*	**0.668±0.009**	**0.646±0.004**	**0.653±0.009**	**0.652±0.004**	0.646±0.010	0.658±0.004	0.580±0.013	0.680±0.003
T1D*	0.607±0.006	0.666±0.003	0.613±0.008	0.666±0.003	0.612±0.007	0.665±0.002	0.564±0.009	0.679±0.003
RA*	0.650±0.007	0.651±0.003	0.644±0.009	0.651±0.003	0.650±0.007	0.651±0.003	0.652±0.010	0.651±0.003

Table S3: Results of permutation tests comparing MKLMM-Adapt and AMB on the WTCCC data sets. Each test was performed with 100,000 permutations. The P-value is the percentage of permutations wherein the evaluated method had a larger advantage over AMB than observed in the original data.

	MKLMM-Adapt		MKLMM-Poly2	
Trait	**AUC**	**NOOS LL**	**AUC**	**NOOS LL**
CD	5.00×10^{-5}	$< 10^{-5}$	1.18×10^{-1}	6.30×10^{-4}
T1D	1.82×10^{-2}	$< 10^{-5}$	1.75×10^{-2}	5.00×10^{-5}
BD	9.11×10^{-1}	7.63×10^{-1}	6.11×10^{-1}	2.13×10^{-2}
RA	7.30×10^{-1}	6.81×10^{-1}	5.32×10^{-1}	4.04×10^{-1}
T2D	3.47×10^{-1}	5.29×10^{-2}	2.14×10^{-1}	2.97×10^{-1}
CAD	5.16×10^{-1}	7.05×10^{-1}	5.21×10^{-1}	6.12×10^{-1}
HT	9.26×10^{-1}	7.97×10^{-1}	9.45×10^{-1}	4.89×10^{-1}
UC	1.20×10^{-4}	$< 10^{-5}$	2.96×10^{-1}	9.99×10^{-1}

Table S4: **Analysis results on WTCCC data, using quantitative phenotype measures**. The reported values are the root mean square error (RMSE), and Pearson correlation (Corr). Results marked in bold text indicate a statistically significant advantage over AMB.

Trait	MKLMM-Adapt		MKLMM-Poly2		AMB		GBLUP	
	RMSE	Corr	RMSE	Corr	RMSE	Corr	RMSE	Corr
CD	1.259±0.006	**0.285±0.016**	1.263±0.004	**0.263±0.018**	1.259±0.004	0.235±0.012	1.261±0.003	0.134±0.019
T1D	1.224±0.006	**0.620±0.011**	1.232±0.006	**0.619±0.010**	1.224±0.006	0.607±0.013	1.224±0.006	0.170±0.014
BD	1.238±0.003	0.137±0.020	1.243±0.003	**0.145±0.017**	1.241±0.003	0.133±0.023	1.237±0.002	0.125±0.015
RA	1.253±0.005	0.418±0.015	1.263±0.006	0.422±0.010	1.253±0.005	0.422±0.016	1.253±0.005	0.295±0.015
T2D	1.236±0.005	0.230±0.015	1.242±0.004	0.231±0.017	1.236±0.005	0.233±0.016	1.235±0.003	0.166±0.014
CAD	1.265±0.002	0.356±0.019	1.275±0.005	0.355±0.019	1.267±0.002	0.356±0.019	1.261±0.004	0.359±0.023
HT	1.233±0.003	0.206±0.014	1.243±0.004	0.202±0.003	1.233±0.003	0.211±0.015	1.233±0.003	0.134±0.006
UC	1.304±0.004	**0.206±0.014**	10.63±2.36	0.057±0.005	1.304±0.002	0.145±0.008	1.303±0.002	0.131±0.002

Table S5: **Analysis results on WTCCC data, without omitting the top 10 principal components.** The table is similar to Table S1, but reports results for data that includes the top 10 principal components, and may thus be susceptible to spurious results due to confounding. The reported values are the area under ROC curve (AUC; higher is better), and average negative out of sample log likelihood (NOOS LL; lower is better). Results marked in bold text indicate a statistically significant advantage over AMB.

Trait	MKLMM-Adapt		MKLMM-Poly2		AMB		GBLUP	
	AUC	NOOS LL	AUC	NOOS LL	AUC	NOOS LL	AUC	NOOS LL
CD	**0.679±0.013**	**0.640±0.005**	0.675±0.010	**0.641±0.005**	0.673±0.010	0.643±0.005	0.628±0.014	0.666±0.004
T1D	0.891±0.006	**0.422±0.009**	**0.892±0.007**	**0.422±0.010**	0.890±0.006	0.428±0.009	0.672±0.008	0.642±0.004
BD	0.640±0.006	0.660±0.001	0.638±0.006	0.661±0.002	0.641±0.006	0.658±0.002	0.641±0.008	0.660±0.003
RA	0.763±0.009	0.581±0.006	0.759±0.008	0.581±0.006	0.761±0.009	0.581±0.006	0.690±0.007	0.634±0.003
T2D	0.640±0.005	0.652±0.003	0.639±0.006	0.653±0.004	0.641±0.005	0.653±0.003	0.602±0.008	0.671±0.003
CAD	0.693±0.014	0.624±0.008	0.692±0.014	0.628±0.008	0.692±0.014	0.625±0.008	0.694±0.014	0.623±0.008
HT	0.623±0.005	0.655±0.002	0.623±0.004	0.657±0.003	0.623±0.005	0.655±0.002	0.591±0.006	0.671±0.002
UC	0.626±0.005	0.670±0.005	0.631±0.003	0.671±0.005	0.629±0.005	0.663±0.003	0.623±0.003	0.667±0.003

Table S6: Analysis results on WTCCC data, using an additional controls group. The table is similar to Table S1, but uses the C58 controls group in addition to the national blood service control group. The reported values are the area under ROC curve (AUC; higher is better), and average negative out of sample log likelihood (NOOS LL; lower is better). Results marked in bold text indicate a statistically significant advantage over AMB.

Trait	MKLMM-Adapt		MKLMM-Poly2		AMB		GBLUP	
	AUC	NOOS LL	AUC	NOOS LL	AUC	NOOS LL	AUC	NOOS LL
CD	**0.688±0.004**	**0.618±0.002**	**0.668±0.011**	**0.620±0.003**	0.644±0.008	0.625±0.004	0.584±0.008	0.650±0.002
T1D	**0.917±0.007**	**0.384±0.005**	**0.914±0.007**	**0.394±0.010**	0.882±0.009	0.442±0.011	0.621±0.011	0.652±0.003
BD	0.621±0.009	0.643±0.004	0.626±0.009	**0.636±0.004**	0.620±0.009	0.644±0.003	0.617±0.012	0.650±0.003
RA	0.745±0.008	0.573±0.003	0.750±0.007	0.569±0.004	0.746±0.007	0.573±0.004	0.668±0.007	0.626±0.003
T2D	0.675±0.011	0.619±0.003	0.672±0.007	0.622±0.004	0.674±0.013	0.621±0.005	0.603±0.005	0.655±0.002
CAD	0.718±0.006	0.597±0.003	0.717±0.011	0.598±0.003	0.717±0.007	0.599±0.004	0.698±0.003	0.611±0.002
HT	0.632±0.003	0.637±0.002	0.633±0.007	0.634±0.003	0.635±0.009	0.635±0.005	0.603±0.011	0.657±0.003
UC	**0.628±0.004**	**0.605±0.002**	0.617±0.003	0.608±0.003	0.613±0.003	0.607±0.002	0.583±0.004	0.684±0.001
CD*	**0.687±0.004**	**0.620±0.003**	**0.667±0.008**	**0.628±0.005**	0.645±0.007	0.644±0.003	0.583±0.008	0.661±0.003
T1D*	0.611±0.005	0.664±0.003	0.613±0.006	0.665±0.004	0.613±0.007	0.661±0.004	0.588±0.008	0.664±0.006
RA*	0.653±0.006	0.650±0.004	0.655±0.010	0.645±0.003	0.653±0.007	0.649±0.003	0.651±0.011	0.641±0.005

Table S7: Analysis results on WTCCC data, using additional MKLMM methods. The table is similar to Supplemental Table 1, but reports results for an MKLMM formulation that uses a weighted combination of linear and an SP kernel for each region (MKLMM-SP), and an MKLMM formulation that uses a weighted combination of a linear and a radial basis function kernel for each region (MKLMM-RBF). Results marked in bold text indicate a statistically significant advantage over AMB.

Trait	MKLMM-Adapt		MKLMM-SP		MKLMM-RBF		AMB	
	AUC	NOOS LL	AUC	NOOS LL	AUC	NOOS LL	AUC	NOOS LL
CD	**0.667±0.010**	**0.645±0.005**	**0.666±0.010**	**0.645±0.005**	**0.667±0.011**	**0.648±0.006**	0.645±0.011	0.657±0.004
T1D	**0.886±0.004**	**0.434±0.006**	0.886±0.003	**0.434±0.006**	0.885±0.005	0.445±0.010	0.883±0.004	0.443±0.006
BD	0.563±0.011	0.681±0.004	0.572±0.008	**0.676±0.003**	0.572±0.009	0.682±0.004	0.568±0.011	0.681±0.004
RA	0.750±0.009	0.592±0.007	0.750±0.010	0.590±0.007	0.751±0.010	0.590±0.007	0.752±0.010	0.589±0.007
T2D	0.634±0.007	0.655±0.004	0.632±0.007	0.655±0.003	0.633±0.007	0.660±0.003	0.634±0.007	0.655±0.004
CAD	0.698±0.014	0.624±0.007	0.697±0.013	0.621±0.007	0.696±0.014	0.629±0.008	0.699±0.013	0.621±0.007
HT	0.611±0.003	0.662±0.002	0.609±0.004	0.658±0.003	0.614±0.004	0.660±0.003	0.611±0.003	0.658±0.003
UC	**0.601±0.007**	**0.676±0.004**	0.587±0.006	**0.678±0.001**	0.592±0.003	**0.676±0.002**	0.585±0.002	0.683±0.001

Supplemental figures

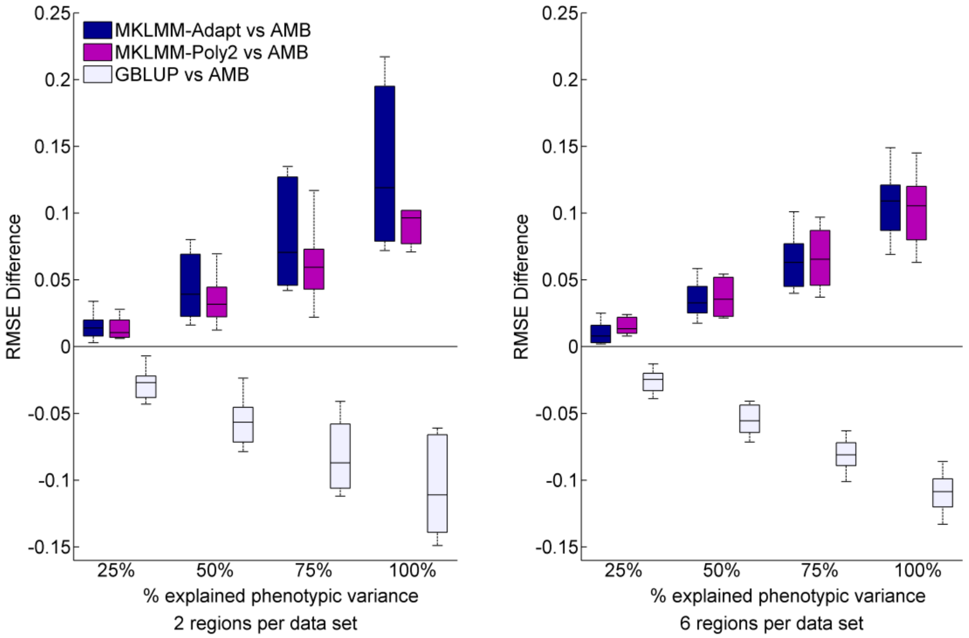

Figure S1: Comparison of the evaluated methods on synthetic data sets with various ratios of explained to total phenotypic variance. The advantage of the MKLMM methods over AMB, and of MKLMM-Adapt over MKLMM-Poly2, increases with the percentage of explained variance.

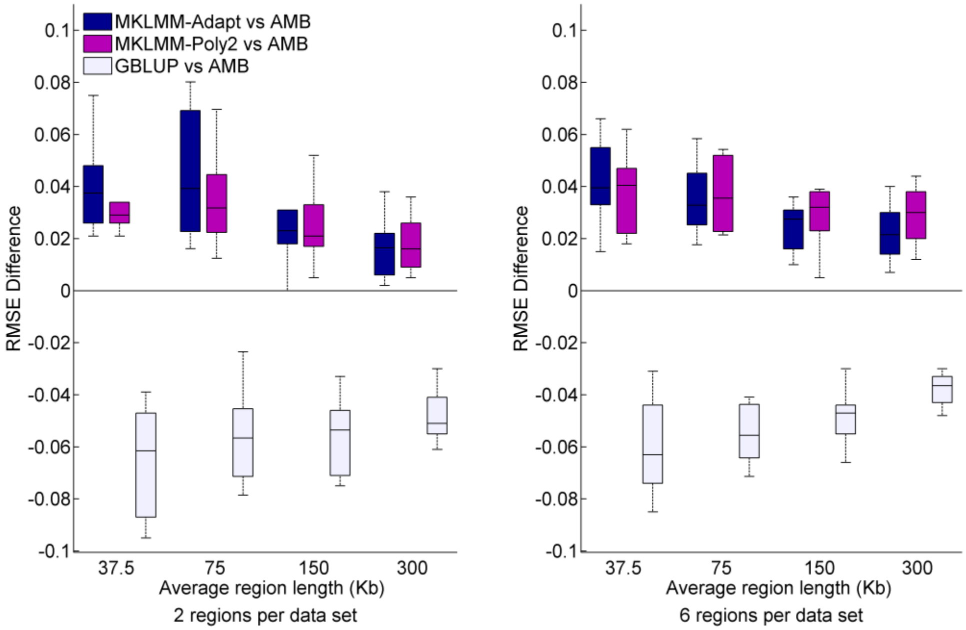

Figure S2: Comparison of the evaluated methods on synthetic data sets with various region lengths. The advantage of the kernel-based methods over the linear methods is greater under shorter regions, indicating that interactions can be better captured over short distances. Linear methods do not capture interactions and are thus less sensitive to the region length.

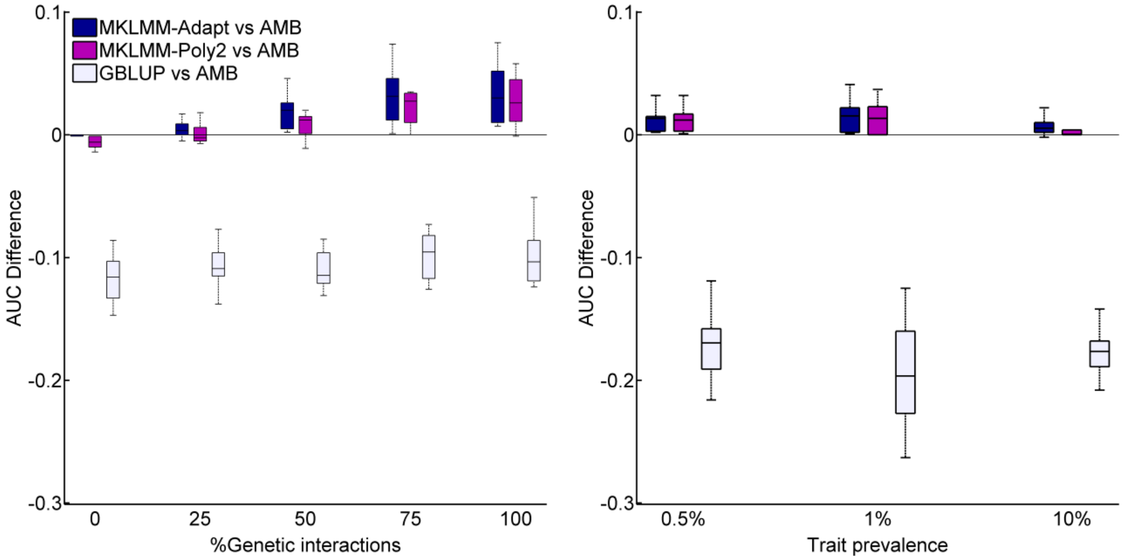

Figure S3: Comparison of the evaluated methods on synthetic data sets with binary phenotypes. The left pane shows performance for randomly ascertained data sets with real genotypes, and the right pane shows performance for ascertained data sets with synthetic genotypes and an equal number of cases and controls, as a function of the trait prevalence in the population. Each data set contains two genomic regions harboring interacting variants. The MKLMM methods outperform AMB under all settings.

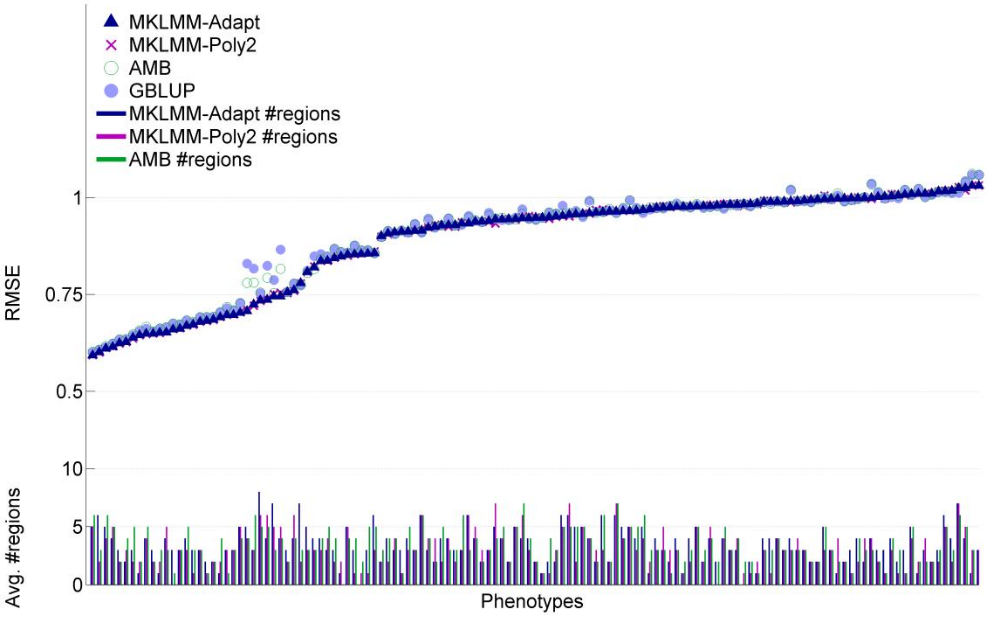

Figure S4: Absolute performance of the evaluated methods in prediction of mouse phenotypes, according to the root mean square error (RMSE) measure.

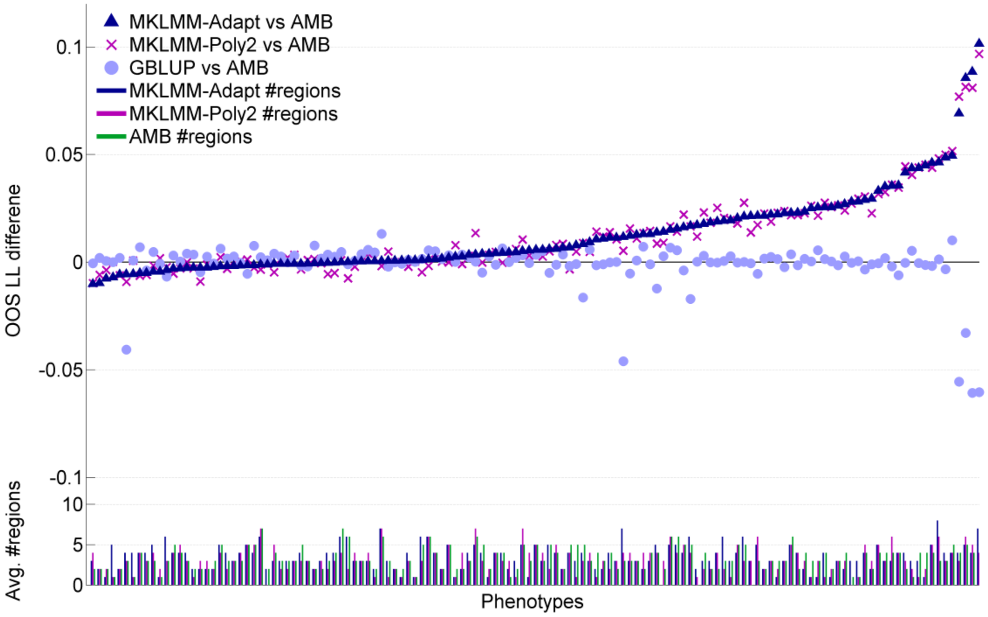

Figure S5: Relative performance of the evaluated methods in prediction of mouse phenotypes, according to the OOS LL measure. For each phenotype, the figure shows the difference between the prediction performance of the evaluated methods and AMB, according to OOS LL. MKLMM-Adapt outperformed AMB across 96 phenotypes.

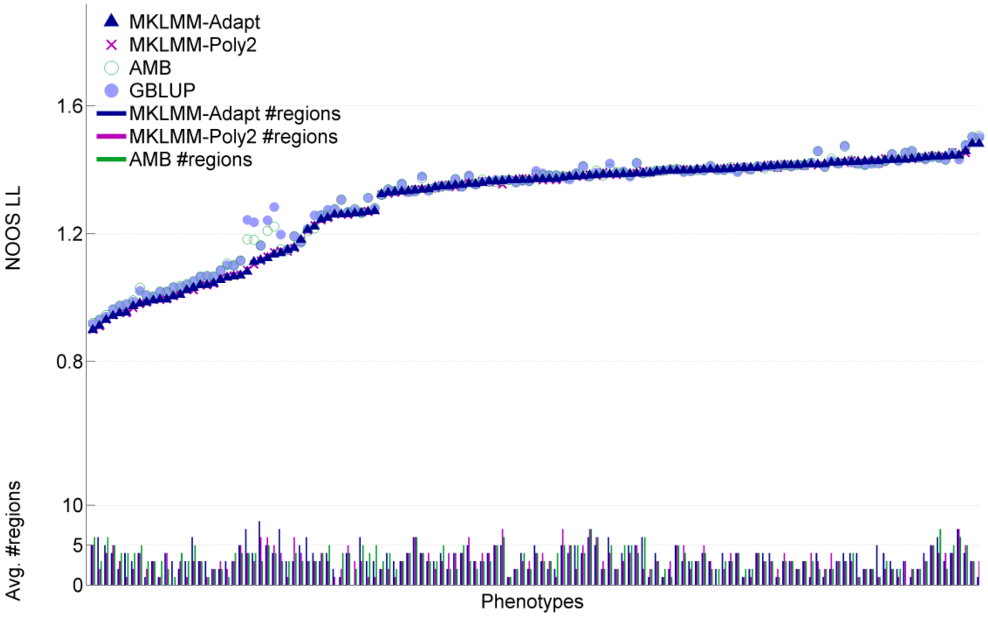

Figure S6: Absolute performance of the evaluated methods in prediction of mouse phenotypes, according to negative out of sample log likelihood (NOOS LL) measure.

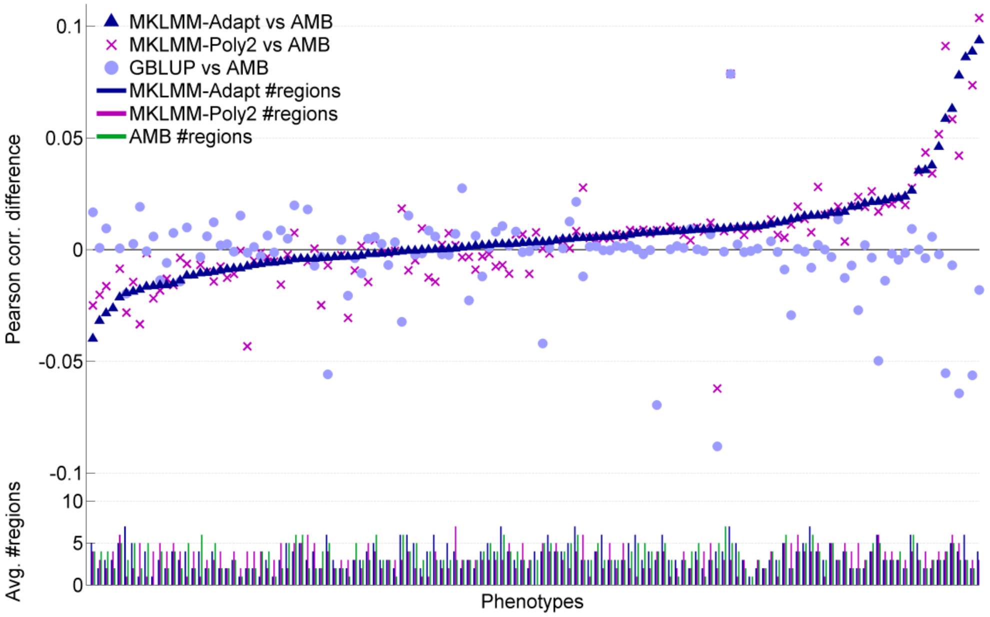

Figure S7: Relative performance of the evaluated methods in prediction of mouse phenotypes, according to the Pearson correlation measure. For each phenotype, the figure shows the difference between the prediction performance of the evaluated methods and AMB. MKLMM-Adapt outperformed AMB across 83 phenotypes.

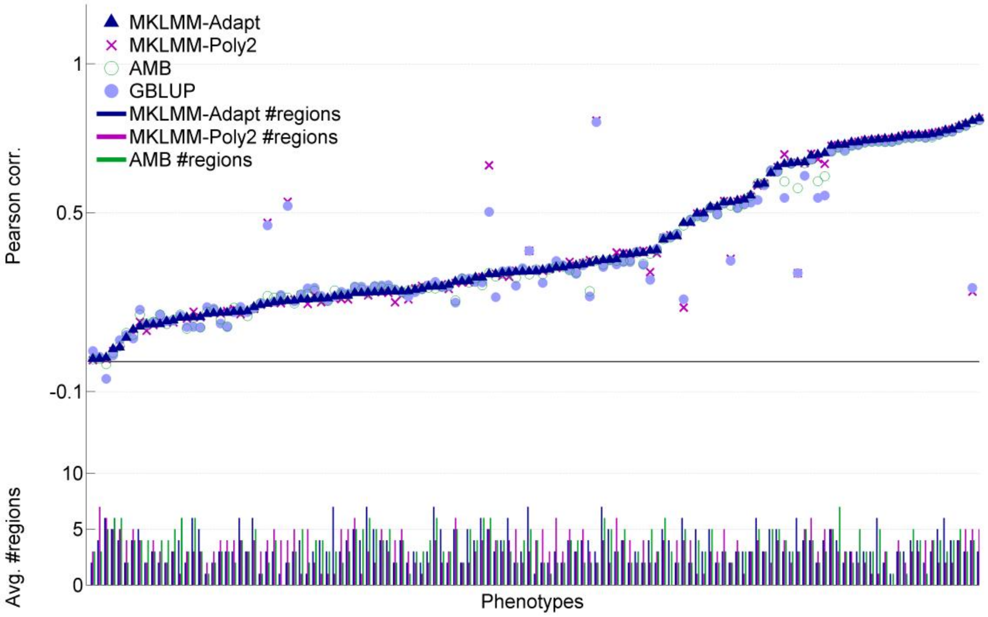

Figure S8: Absolute performance of the evaluated methods in prediction of mouse phenotypes, according to the Pearson correlation measure.

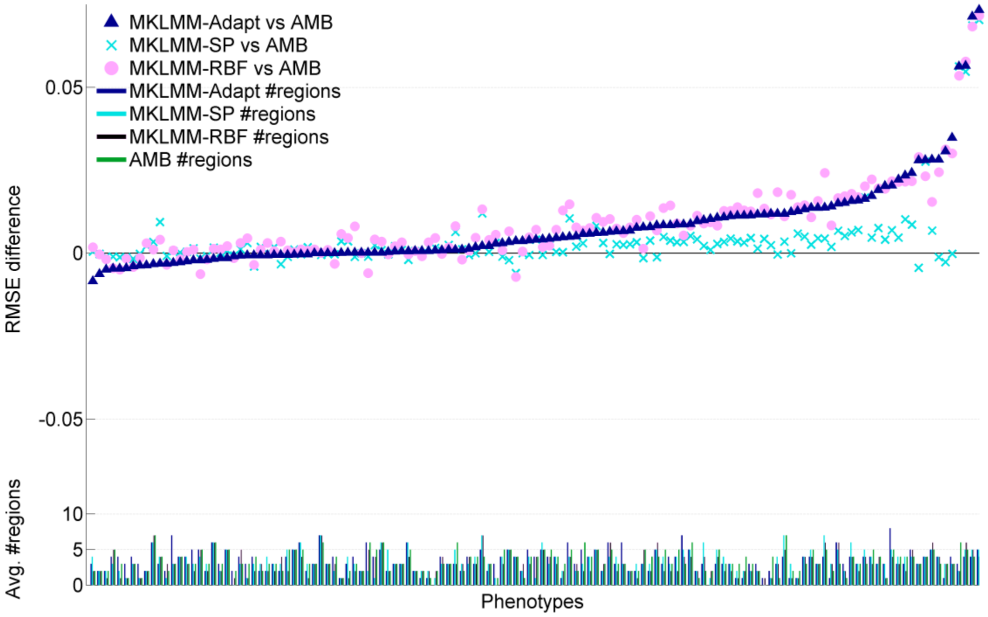

Figure S9: Relative performance of additional evaluated methods in prediction of mouse phenotypes, according to the RMSE measure. The figure is similar to Figure S3 in the main text, but reports results for additional MKLMM formulations that assign a weighted combination of a linear and a radial basis function kernel for each region (MKLMM-RBF), or a weighted combination of a linear and a saturating pathways kernel for each region (MKLMM-SP). Higher values indicate a greater advantage for a method over AMB. MKLMM-SP and MKLMM-RBF outperformed AMB across 96 and 112 phenotypes, respectively.

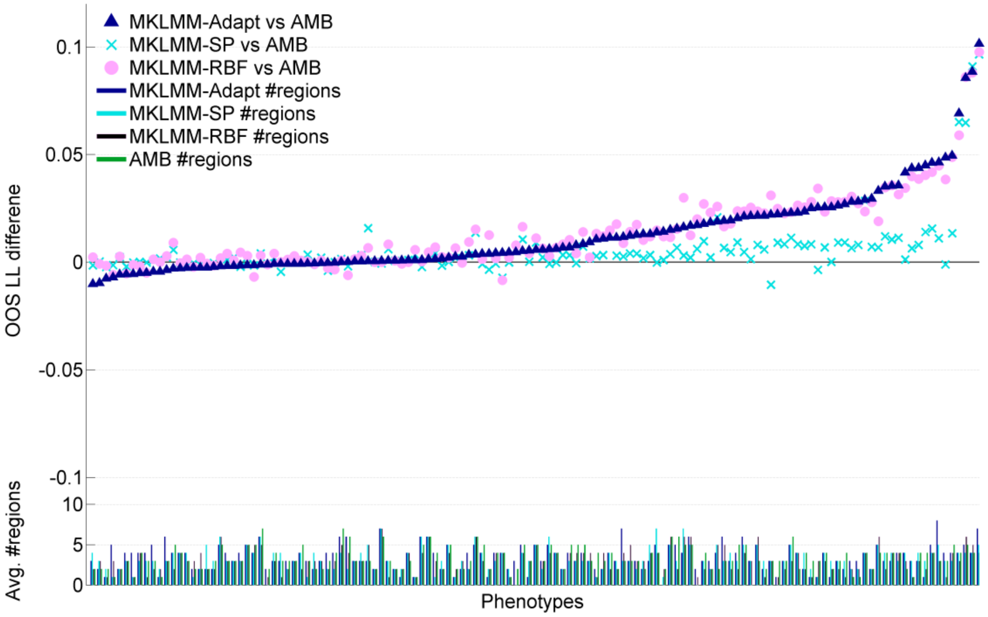

Figure S10: Relative performance of additional evaluated methods in prediction of mouse phenotypes, according to the OOS LL measure. The figure is similar to Figure S9, but uses OOS LL to measure prediction performance. MKLMM-SP and MKLMM-RBF outperformed AMB across 97 and 113 phenotypes, respectively.

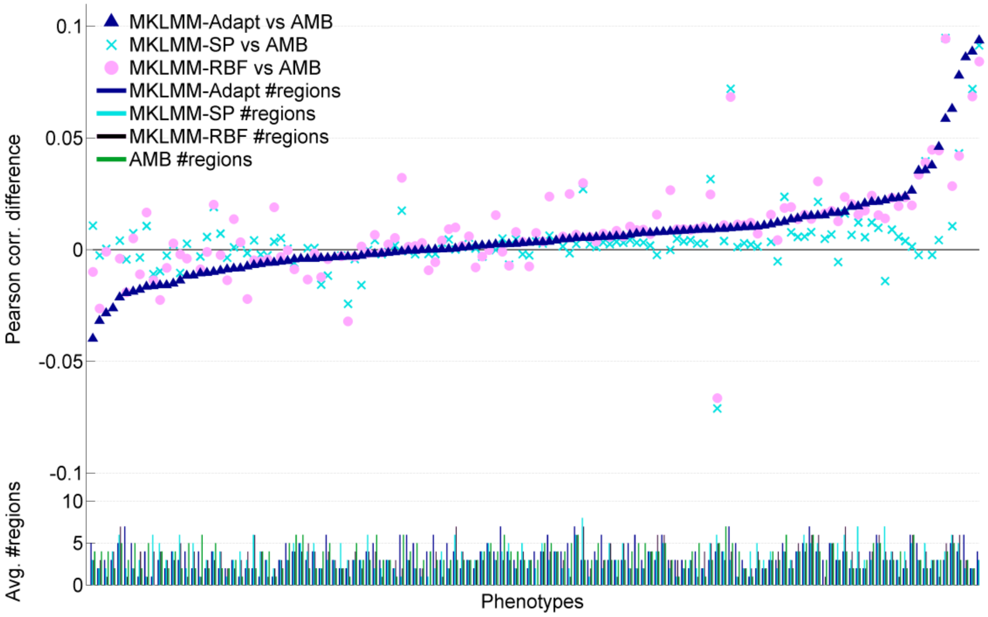

Figure S11: Relative performance of additional evaluated methods in prediction of mouse phenotypes, according to the Pearson correlation measure. The figure is similar to Figure S9, but uses Pearson correlation to measure prediction performance. MKLMM-SP and MKLMM-RBF both outperformed AMB across 89 phenotypes.

Investigating tagging of ungenotyped variants

A potential concern with MKLMM is that it may improve prediction performance over AMB due to better tagging of ungenotyped variants, rather than modeling of true interactions. To investigate this possibility, we generated data sets based on true genotypes of control individuals from the Wellcome Trust Case Control Consortium 2 (WTCCC2) with one chromosome-wide linear kernel and an additional linear kernel for a region consisting of 2,4,6 or 8 SNPs. Afterwards, we excluded the causal SNPs in the small region and their 0,2,4 or 6 closest flanking SNPs from the analysis.

In all cases MKLMM-Adapt selected only linear kernels, indicating that even in the situation which is most favorable to the tagging hypothesis, where untyped causal variants have a strong linear effect, and typed variants have varying levels and patterns of LD with these causative variants, the tagging did not create phantom interactions that were captured by MKLMM-Adapt. Furthermore, prediction performance for MKLMM-RBF and MKLMM-SP was almost the same as that of AMB (mean Pearson correlation difference <0.0005, and maximum Pearson correlation difference <0.03 in all cases), again indicating that improved tagging of causal SNPs was not exploited to improve prediction performance.

Evaluated kernels

This work makes use of several kernel types under the MKLMM framework. In the following, we describe the evaluated kernels and their underlying assumptions. The kernels surveyed here are used as building blocks for a composite kernel that is a weighted-sum of region-specific kernels, as detailed in the main text. Throughout this section, $\boldsymbol{X} \in R^{n \times m}$ denotes a matrix of m variants measured for n individuals, where all variants are standardized to have a zero mean and unit variance. We consider four kernels: A linear kernel, a polynomial kernel of degree 2, a radial basis function (RBF) kernel and a saturating pathways (SP) kernel.

The linear kernel is given by

$$\boldsymbol{G}(\boldsymbol{X},\theta)_{k,l} = \frac{1}{m}\boldsymbol{X}_k^T\boldsymbol{X}_l, \qquad (1)$$

where $\boldsymbol{G}(\boldsymbol{X},\theta)_{k,l}$ is the genotypic covariance between individuals k and l. This kernel is equivalent to the kernel defined in Equation 3 in the main text, with the exception that the scaling factor (commonly known as σ_g^2 in the presence of a single genomewide kernel) is now considered a parameter of the composite kernel. Therefore, the kernel presented here is not associated with any parameter. The linear kernel corresponds to the identity transformation, and encodes the assumption that genetic variants have a linear effect on the phenotype.

The polynomial kernel of degree 2 is given by

$$\boldsymbol{G}(\boldsymbol{X},\theta) = \frac{1}{m^2}(\boldsymbol{X}_k^T\boldsymbol{X}_l)^2 \qquad (2)$$

This kernel encodes the assumption that products of pairs of variants have a linear effect on the phenotype, as described in the main text. It corresponds to a transformation that projects each genotype vector $\boldsymbol{X}_k$ into a new vector in which there is an entry for the product of every pair of variants.

The RBF kernel, which has received considerable attention in the plant and animal breeding literature (Morota and Gianola 2014), generalizes the polynomial kernel to model interactions of an arbitrary order. This kernel involves a single positive parameter θ, and is given by

$$\boldsymbol{G}(\boldsymbol{X},\theta)_{k,l} = \exp\left(-\frac{1}{2\theta m}\sum_i \left(X_k^i - X_l^i\right)^2\right). \qquad (3)$$

The RBF kernel corresponds to an infinite-dimensional transformation, and can therefore capture rich interaction patterns. To gain some intuition into its underlying assumptions, we consider the explicit underlying transformation of the RBF kernel. It can be shown that this transformation associates every non-negative integer number j

and every set of non-negative integers $t_1, \ldots, t_m$ such that $\sum_{i=1}^{m} t_i = j$ with a unique entry, given by (Shashua 2009)

$$\frac{\exp\left(\frac{-\|X_k\|_2^2}{2mj\theta}\right)}{\sqrt{j!}^{1/j}\, m^{j/2}\theta^{j/2}} \binom{j}{t_1, \ldots, t_m}^{1/2} (X_k^1)^{t_1} \cdots (X_k^m)^{t_m}\,. \tag{4}$$

Equation 4 demonstrates that the RBF kernel generalizes the polynomial kernel to also capture higher order interactions, because every possible polynomial combination of parameters is represented in its underlying transformation.

Finally, we also consider the saturating pathways kernel, described in detail below.

The saturating pathways kernel

Here we present a kernel known in the Machine learning community as the neural network kernel (Neal 1996). In our context it has an attractive biological interpretation as assuming an interaction model of saturating pathways, as described next. In this description we assume a single kernel and omit the fixed effects for ease of presentation.

Consider a phenotype that is affected by R biological processes in an additive manner. The phenotype for individual k is given by

$$y_k(\boldsymbol{X}_k; \boldsymbol{\theta}^p, \boldsymbol{\theta}^w) = \sum_{i=1}^{R} \theta_i^p h((\theta_i^w)^T \boldsymbol{X}_k) + \epsilon, \tag{5}$$

where $h(z)$ is a monotone function in the range $[-1,1]$, and $\epsilon \sim N(0, \sigma_e^2)$ (Supplemental Figure S12). We refer to each term $h((\theta_i^w)^T \boldsymbol{X}_k)$ as a saturating pathway, because the bounded range of $h(z)$ encodes saturation dynamics, which are common in biological systems (Zuk et al. 2012). Interactions are naturally encoded in this model via the saturating functions. The proposed model can be seen as a generalization of the standard linear kernel. It is easy to show that when there is only a single pathway

$(R = 1)$ and $h(z) = z$, Equation 5 can be described by a linear kernel, by setting $\theta_i^p = 1, \theta_i^w \sim N\left(0, \frac{\sigma_g^2}{m} I\right)$, and using the Bayesian interpretation of LMMs (see below).

We now extend Equation 5 by considering the limit of an infinite number of pathways. Assuming that $\boldsymbol{\theta}^w, \boldsymbol{\theta}^p$ are vectors of iid zero mean random variables with $\operatorname{var}(\theta_i^p) = 1/R$, taking the limit $R \to \infty$, and applying the central limit theorem, we obtain that y_k follows a zero mean normal distribution with the covariance matrix

$$\operatorname{cov}(y_k, y_l) = \operatorname{cov}\big(h((\boldsymbol{\theta}^w)^T \boldsymbol{X}_k), h((\boldsymbol{\theta}^w)^T \boldsymbol{X}_l)\big). \qquad (6)$$

Different choices of the saturation function $h(z)$ and the distribution of $\boldsymbol{\theta}^w$ lead to different kernels. A common choice is $h(z) = \operatorname{erf}(z) = 2\pi^{-1/2} \int_0^z e^{-t^2} dt$ and $\boldsymbol{\theta}^w \sim N(0, \sigma_w^2 \boldsymbol{I})$, which leads to a kernel defined by a single parameter (Rasmussen and Williams 2006):

$$\boldsymbol{G}(\boldsymbol{X}; \theta)_{k,l} = 2\pi^{-1} \sin^{-1}\left(\frac{\frac{1}{m}(\boldsymbol{X}_k)^T \boldsymbol{X}_l}{\sqrt{(\theta + \|\boldsymbol{X}_k\|_2^2/m)(\theta + \|\boldsymbol{X}_l\|_2^2/m)}}\right), \qquad (7)$$

where $\theta = 1/2\sigma_w^2$. Typically the genotype matrix $\boldsymbol{X}$ is augmented with a column with the value $\sqrt{m}$ for numerical stability. Each pathway can model a sharp step function by taking the limit $\sigma_w^2 \to \infty$. It is worthy to note that the presented model (when not invoking the central limit theorem) is known to be a universal approximator (Hornik 1993), which can approximate any arbitrary function to an arbitrary degree of accuracy as the sample size tends to infinity.

We also note that there is some resemblance between the proposed model and the well known limiting pathways model (Zuk et al. 2012), that can be obtained from Equation 5 by defining $h(z) = z$ and replacing the summation with a minimization operator. However, unlike the SP model, the limiting pathways model results in a non-normal phenotype distribution.

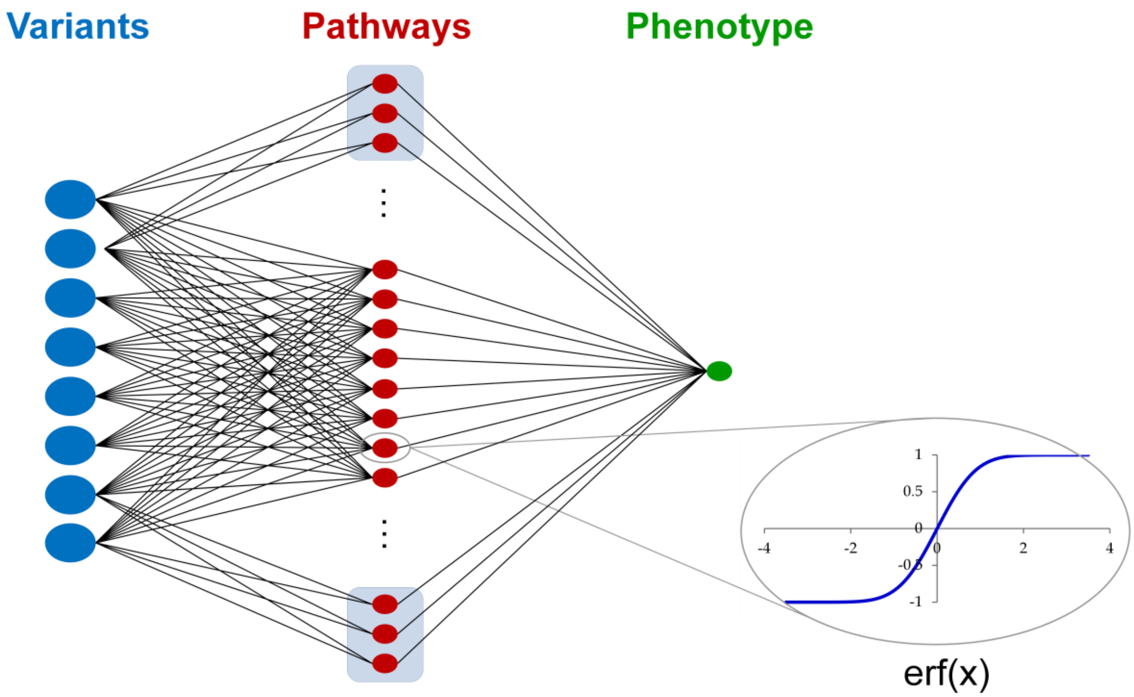

Figure S12: **A schematic representation of the saturating pathways kernel**. Blue circles are genetic variants, red circles are pathways and the green circle represents the phenotype. Every pathway is associated with a subset of variants and with a kernel. The figure shows one genome-wide SP kernel and two region-specific SP kernels, highlighted at the top and bottom of the middle column. The top two and bottom two variants belong to regions with region-specific kernels. The output of each pathway is a saturating function (erf(x)) of a linear combination of its variants, and the phenotype is a linear combination of the pathway values. The linear combination coefficients of each pathway and of the phenotype are drawn from a normal distribution.

MKLMM parameters estimation

Here we describe maximum likelihood (ML) and restricted maximum likelihood (REML) parameter estimation for MKLMM. Under the LMM, the log likelihood of the phenotypes vector $\mathbf{y}$ for n individuals is given by:

$$LL(\boldsymbol{\beta},\boldsymbol{\theta},\sigma_e^2) = -\frac{1}{2}((\boldsymbol{y}-\boldsymbol{C\beta})^T\boldsymbol{R}(\boldsymbol{X};\boldsymbol{\theta},\sigma_e^2)^{-1}\,(\boldsymbol{y}-\boldsymbol{C\beta}) + \mathrm{nlog}(2\pi) \\ + \log|\boldsymbol{R}(\boldsymbol{X};\boldsymbol{\theta},\sigma_e^2)|) \qquad (8)$$

where $\boldsymbol{X}$ is a matrix of genotyped variants, $\boldsymbol{C}$ is a matrix of covariates, $\boldsymbol{\beta}$ is a vector of fixed effects, and $\boldsymbol{R}(\boldsymbol{X};\boldsymbol{\theta},\sigma_e^2) = \boldsymbol{G}(\boldsymbol{X};\boldsymbol{\theta}) + \sigma_e^2\boldsymbol{I}$ is the overall covariance matrix composed of the genetic covariance matrix $\boldsymbol{G}(\boldsymbol{X};\boldsymbol{\theta})$ and an environmental term with variance σ_e^2.

Maximum likelihood estimation

Maximum likelihood estimation amounts to finding the parameters $\boldsymbol{\beta}$,$\boldsymbol{\theta}$ and σ_e^2 that maximize Equation 8. Importantly, the normalization term of the normal distribution serves as a regularization term that discourages overly complex models with a very high variance. This distinguishes LMMs and their extensions from fixed effect models. Parameter estimation can be performed via conjugate gradient ascent, as described below. The maximum likelihood estimate of $\boldsymbol{\beta}$ given $\boldsymbol{\theta}$ and σ_e^2 is given by (Lippert et al. 2011):

$$\widehat{\boldsymbol{\beta}} = (\boldsymbol{C}^T\boldsymbol{R}(\boldsymbol{X};\boldsymbol{\theta},\sigma_e^2)^{-\mathbf{1}}\boldsymbol{C})^{-1}\boldsymbol{C}^T\boldsymbol{R}(\boldsymbol{X};\boldsymbol{\theta},\sigma_e^2)^{-\mathbf{1}}\boldsymbol{y}. \qquad (9)$$

To infer $\boldsymbol{\theta}$ and σ_e^2, we perform conjugate gradient ascent using the gradient of Equation 8, given by (Rasmussen and Williams 2006):

$$\frac{\partial}{\partial\theta_j}LL(\boldsymbol{\beta},\boldsymbol{\theta},\sigma_e^2) = \frac{1}{2}\mathrm{tr}\left((\boldsymbol{\alpha\alpha}^T - \boldsymbol{R}(\boldsymbol{X};\boldsymbol{\theta},\sigma_e^2)^{-1})\frac{\partial\boldsymbol{R}(\boldsymbol{X},\sigma_e^2;\boldsymbol{\theta})}{\partial\theta_j}\right) \qquad (10)$$

$$\frac{\partial}{\partial\sigma_e^2}LL(\boldsymbol{\beta},\boldsymbol{\theta},\sigma_e^2) = \frac{1}{2}\mathrm{tr}(\boldsymbol{\alpha\alpha}^T - \boldsymbol{R}(\boldsymbol{X};\boldsymbol{\theta},\sigma_e^2)^{-1}), \qquad (11)$$

where $\boldsymbol{\alpha} = \boldsymbol{R}(\boldsymbol{X};\boldsymbol{\theta},\sigma_e^2)^{-1}(\boldsymbol{y}-\boldsymbol{C}\widehat{\boldsymbol{\beta}})$, $\mathrm{tr}(\cdot)$ is the trace operator, and $\frac{\partial\boldsymbol{R}(\boldsymbol{X};\boldsymbol{\theta},\sigma_e^2)}{\partial\theta_j}$ is a matrix of elementwise partial derivatives.

The computational complexity scales cubically with the sample size due to the matrix inversion, as in standard LMMs. All other operations scale quadratically with the sample size. Importantly, the complexity is independent of the number of parameters. After inverting this matrix, the computation of Equations 8-11 can be parallelized. Efficient approximations can potentially scale this procedure up to hundreds of thousands of individuals at a modest loss of accuracy (Snelson and Ghahramani 2007; Hensman et al. 2013; Yang et al. 2015).

Restricted maximum likelihood estimation

The restricted log likelihood is comprised of the log likelihood and three additional terms (Lippert et al. 2011):

$$LL_R(\boldsymbol{\beta}, \boldsymbol{\theta}, \sigma_e^2) = LL(\boldsymbol{\beta}, \boldsymbol{\theta}, \sigma_e^2) + \frac{1}{2}(d\log(2\pi) + \log|\boldsymbol{C}^T\boldsymbol{C}| - \log|\boldsymbol{C}^T\boldsymbol{R}(\boldsymbol{X}; \boldsymbol{\theta}, \sigma_e^2)^{-1}\boldsymbol{C}|). \qquad (12)$$

The restricted maximum likelihood value of $\boldsymbol{\beta}$ given $\boldsymbol{\theta}$ and σ_e^2 is the same as for non-restricted likelihood (Equation 9). The gradient of $LL_R(\boldsymbol{\beta}, \boldsymbol{\theta}, \sigma_e^2)$ with respect to the other parameters is given by

$$\frac{\partial}{\partial\theta_j}\mathrm{LL_R}(\boldsymbol{\beta}, \boldsymbol{\theta}, \sigma_e^2) = \frac{\partial}{\partial\theta_j}LL(\boldsymbol{y}) + \frac{1}{2}\mathrm{tr}\left((\boldsymbol{\gamma}\boldsymbol{C})^{-1}\boldsymbol{\gamma}\frac{\partial\boldsymbol{R}(\boldsymbol{X}; \boldsymbol{\theta}, \sigma_e^2)}{\partial\theta_j}\boldsymbol{\gamma}^T\right) \qquad (13)$$

$$\frac{\partial}{\partial\sigma_e^2}\mathrm{LL_R}(\boldsymbol{\beta}, \boldsymbol{\theta}, \sigma_e^2) = \frac{\partial}{\partial\sigma_e^2}LL(\boldsymbol{y}) + \frac{1}{2}\mathrm{tr}((\boldsymbol{\gamma}\boldsymbol{C})^{-1}\boldsymbol{\gamma}\boldsymbol{\gamma}^{\boldsymbol{T}}), \qquad (14)$$

where $\boldsymbol{\gamma} = \boldsymbol{C}^T\boldsymbol{R}(\boldsymbol{X}; \boldsymbol{\theta}, \sigma_e^2)^{-1}$.

The overall computational complexity remains the same as before after neglecting the cubic dependency on the number of fixed effects, which is typically very small compared to the sample size.

Estimation procedure

Parameter estimation was carried out via a Polak-Ribiere conjugate gradient procedure, based on the code from the GPML toolbox (Rasmussen and Nickisch 2010), and using REML computations. Such an optimization procedure is available in many optimization packages, and requires only a function to compute the gradient. Alternatively, one could use average information REML (Gilmour et al. 1995; Lee and van der Werf 2006), though this procedure was not carried out here. Empirical comparison with Adaptive MultiBLUP (Speed and Balding 2014), which uses average information REML, indicated that both methods tend to have very similar performance.

The optimization for each fold consisted of 100 conjugate gradient steps. The initial values for the fixed effects were determined via a multivariate linear regression. The initial scaling factor of each kernel was $0.5\mathrm{var}(\boldsymbol{y})/r,$ where $\boldsymbol{y}$ is the phenotype vector and r is the number of kernels. The initial value of the environmental effect variance σ_e^2 was $0.5\mathrm{var}(\boldsymbol{y})$. Although the likelihood surface of MKLMM is not convex, we did not encounter convergence problems, as determined via global optimization routines based on simulated annealing.

MKLMM-Adapt procedure

Here we describe the MKLMM-Adapt model training procedure in detail. The MKLMM-Adapt procedure is composed of two steps: Ranking of genomic regions, and evaluation of models of increasing complexity to select the best one.

Ranking of regions

MKLMM-Adapt ranks genomic regions similarly to MultiBLUP (Speed and Balding 2014). The full procedure is carried as follows.

1. Divide the genomic into overlapping sub-regions spanning 75kb, where the minimum distance between the first base pair of two consecutive sub-regions is 10kb.
2. Assign a score to each sub-region, corresponding to the restricted log-likelihood (Equation 12) when using an LMM with a single linear kernel spanning only the variants in the sub-region. Such a score can be computed efficiently using the

Fast-LMM procedure for fitting LMMs with a low-rank kernel (Lippert et al. 2011).

3. Discard all sub-regions whose restricted log-likelihood is below the 95% percentile of obtained values.
4. Merge consecutive undiscarded sub-regions into regions. The score assigned to each merged region is the maximum score among all its sub-regions.
5. Rank regions according to their score in descending order. Region 0 is always considered as the genome-wide region that spans all variants.

Training an MKLMM-Adapt model

The MKLMM-Adapt model training procedure is as follows. We evaluate several models with increasing complexity, $M^{(0)}, M^{(1)}, \dots, M^{(B)}$, where B is a user defined parameter described in the main text, and where each model $M^{(t)}$ uses a sum of region-specific kernels for regions 0 … t. The difference between the models lies in the form of the covariance matrix. The covariance matrix for model $M^{(t)}$ is given by

$$\boldsymbol{G}(\boldsymbol{X};\theta)^{(t)} = \sum_{i=0}^{t} \boldsymbol{G}_i(\boldsymbol{X}_i;\theta_i),$$

where $\boldsymbol{X}_i$ is the matrix of variants in region i, and $\boldsymbol{G}_i(\boldsymbol{X}_i;\theta_i)$ is the covariance matrix induced by region i. We consider four possible forms for $\boldsymbol{G}_i(\boldsymbol{X}_i;\theta_i)$:

1. Linear kernel only: $\boldsymbol{G}_i(\boldsymbol{X}_i;\theta) = \theta_{i,w}^{\text{linear}} \boldsymbol{G}_{\boldsymbol{i}}^{\text{linear}}(\boldsymbol{X}_i)$
2. Linear and poly2 kernels: $\boldsymbol{G}_i(\boldsymbol{X}_i;\theta) = \theta_{i,w}^{\text{linear}} \boldsymbol{G}_{\boldsymbol{i}}^{\text{linear}}(\boldsymbol{X}_i) + \theta_{i,w}^{\text{poly2}} \boldsymbol{G}_{\boldsymbol{i}}^{\text{poly2}}(\boldsymbol{X}_i)$
3. Linear and RFB kernels: $\boldsymbol{G}_i(\boldsymbol{X}_i;\theta) = \theta_{i,w}^{\text{linear}} \boldsymbol{G}_{\boldsymbol{i}}^{\text{linear}}(\boldsymbol{X}_i) + \theta_{i,w}^{\text{RBF}} \boldsymbol{G}_{\boldsymbol{i}}^{\text{RBF}}\left(\boldsymbol{X}_i;\theta_{i,k}^{\text{RBF}}\right)$
4. Linear and SP kernels: $\boldsymbol{G}_i(\boldsymbol{X}_i;\theta) = \theta_{i,w}^{\text{linear}} \boldsymbol{G}_{\boldsymbol{i}}^{\text{linear}}(\boldsymbol{X}_i) + \theta_{i,w}^{\text{SP}} \boldsymbol{G}_{\boldsymbol{i}}^{\text{SP}}\left(\boldsymbol{X}_i;\theta_{i,k}^{\text{SP}}\right)$,

where $\theta_{i,w}^{\text{type}}$ is the weight assigned to a kernel of the specified type in region i, and $\theta_{i,k}^{\text{type}}$ is the internal parameter of a kernel of the specified type in region i.

When selecting the kernel type for model $M^{(t)}$, the parameters for all regions 0 … t are jointly inferred by maximizing the restricted log likelihood in Equation 12 via conjugate gradient ascent. However, the type of the selected kernel for region t (out of the four alternatives listed above) is only selected once, and will be used when evaluating all other models $M^{(s>t)}$. In order to select the kernel type for region t we perform three likelihood ratio tests, each comparing a model with one of the three composite kernel types to a model that uses only a linear kernel for region t. These tests are carried out by attempting to reject the null hypothesis $H_0 \colon \theta_{i,w}^{\text{type}} = 0$ for each of the three non-linear kernel types. Note that each hypothesis test re-estimates all parameters for all regions $0 \dots t$. A step by step description of the model selection procedure is now provided.

1. Iterate over the index t in the range [0..B]:
    a. Iterate over the evaluated kernel types (in the present study, these include a linear, linear+Poly2, linear+RBF, linear+SP kernels). For each evaluated kernel:
        i. Train an MKLMM model whose covariance matrix is a weighted combination of kernels assigned to regions $0 \dots t$. The kernel type assigned to regions $0 \dots t-1$ is the one selected at previous iterations. The kernel assigned to region t is the one evaluated at the current iteration.
        ii. Evaluate the restricted log likelihood (Equation 12) using the trained model.
        iii. Assign a score to the evaluated model by performing a likelihood ratio test, where the null model uses only a linear kernel for region t. The null distribution is approximated by a $0.5\chi_0^2 \colon 0.5\chi_d^2$ distribution, where d is the number of additional estimated parameters in the alternative model over the null model. This null distribution accounts for the fact that the tested parameters are on the edge of the boundary space in the null model.
    b. If no model has a statistically significant advantage over the linear kernel model after multiple testing correction, select the linear kernel for region t. Otherwise, select the model with the smallest P-value.
    c. Evaluate the predictive performance of the selected model on an external validation set.
2. Select the model with the best predictive performance on an external validation set.

Computational complexity of MKLMM

The computational complexity of MKLMM is similar to that of a standard LMM, and is determined by the sample size, the number of kernels, the number of variants assigned to each kernel and the number of iterations carried out for parameter estimation. We address the computational complexity required for parameter estimation and for phenotype prediction separately.

Parameter estimation requires repeatedly evaluating each of Equations 9 and 12-14 for a specified number of iterations, or until convergence. The computational complexity of each iteration is formally dominated by the computation of $\boldsymbol{G}(\boldsymbol{X};\boldsymbol{\theta})$, which typically scales linearly with the number of parameters p, the number of variants m and the matrix size n^2 (where n is the sample size), yielding a computational complexity of $O(pmn^2)$. However, it is often possible to perform a single $O(mn^2)$ computation for each parameter p and cache the result, thus avoiding dependence on m during the estimation procedure. For example, for a linear kernel, one can compute the matrix XX^T (where X is a matrix of variants) only once and then multiply this matrix by a scaling factor at each iteration. Such a computational shortcut is also possible for polynomial kernels, radial basis function kernels and saturating pathways kernels (see below). MKLMM therefore only requires performing a small finite number of $O(mn^2)$ computations, similarly to standard LMMs. The computational complexity of each iteration in the estimation procedure is thus independent of m.

Assuming we can avoid $O(m)$ operations at each iteration, the computation at each iteration is dominated by the inversion of the overall covariance matrix $\boldsymbol{R}(\boldsymbol{X};\boldsymbol{\theta},\sigma_e^2)$, which scales as $O(n^3)$. Afterwards, Equations 9 and 12-14 can be evaluated concurrently. At each iteration, a different instance of Equation 13 is evaluated for each kernel parameter separately, along with a single instance of Equations 9, 12 and 14. Given the inverted matrix, the computational complexity of Equation 12 is $O(n^2 + n^2d + d^2n + d^3)$, where d is the number of fixed effects, and the $O(n^2)$ element stems from the definition of the normal density. Similarly, the computational complexity for Equation 9 is $O(n^2d + d^2n + d^3)$. Denoting p as the number of kernel parameters, the combined computational complexity of Equations 13-14 is $O(pn^2 + pnd^2 + pnd + d^3)$. This analysis exploits the fact that traces of matrix products can be computed efficiently by only computing the diagonal of the product. Combining all these terms together and neglecting terms that cannot dominate the complexity, the computational complexity for parameter estimation is $O\big(pmn^2 + i(n^3 + n^2d + d^3 + pn^2 + pnd^2)\big)$, where i is the number of estimation iterations.

Assuming that $p < n$, $d < n$ and $pd^2 < n^2$, the asymptotic complexity is $O(pmn^2 + in^3)$, similarly to the asymptotic complexity of REML estimation for standard LMMs with multiple variance components. Typical use of MKLMM will only require a single genome-wide kernel, with the other kernels using a small number of variants (typically less than 500), yielding $O(mn^2 + in^3)$ complexity.

Using similar considerations, the computational complexity of phenotype prediction for a single individual (Equation 2 in the main text) is $O(d + mn + n^2)$. When using the techniques described in the privacy preservation section, the complexity is given by $O(d + M^2)$, where M is the approximated dimensionality of the projection induced by the kernel.

Efficient kernel computations

Here we describe how to efficiently cache kernel computations, so that the computational complexity of kernel evaluation in each iteration of the estimation procedure becomes independent of the number of variants m.

The linear kernel can be trivially cached by pre-computing the quantity $(\boldsymbol{X}_k)^T\boldsymbol{X}_l$ for every pair of individuals k and l. The polynomial kernel can similarly be cached by pre-computing the quantity $((\boldsymbol{X}_k)^T\boldsymbol{X}_l)^2$ for every pair of individuals.

The RBF kernel can be computed efficiently by caching the squared distances between every pair of individuals, given by $D_{kl} = \sum_i \left(X_k^i - X_l^i\right)^2$. Given the matrix D, the RBF kernel can then be computed efficiently via $\boldsymbol{G}(\boldsymbol{X}, \theta)_{k,l} = \exp\left(-\frac{1}{2\theta m} D_{kl}\right)$. Similarly, the SP kernel can be efficiently computed by caching the quantity $\|\boldsymbol{X}_k\|_2^2$ for every individual k, and the product $(\boldsymbol{X}_k)^T \boldsymbol{X}_l$ and for every pair of individuals k and l.

Empirical run time measurement

To evaluate the empirical complexity of MKLMM, we measured estimation run-times for an analysis of Crohn's disease (CD; using 3,171 individuals and 285,650 variants) and ulcerative colitis (UC; using 5,498 individuals and 458,560 variants) with various numbers of kernels, under various MKLMM models. All regions used kernels of the same type. The first region was the genome-wide region, and subsequent regions were selected according to the MKLMM-Adapt region selection procedure. The analysis was performed on a 2GHz Linux workstation using a single core. All analyses avoided direct dependence on the number of variants m by caching the covariance matrices, as described above. The average run time for each estimation procedure (using 100 estimation iterations) is reported below, using the average of five independent estimation procedures (one for each cross validation fold). The times are reported in minutes. The first row reports the average computation time for the genome-wide kernel, as this kernel was computed only once and then cached for subsequent use.

	CD **Linear kernels**	**CD** **SP kernels**	**UC** **Linear kernels**	**UC** **SP kernels**
Kernel creation	5.1	5.1	26.6	32.0
1 kernel	4.6	4.3	20.5	26.1
2 kernels	4.6	5.5	21.4	29.3
3 kernels	4.9	7.0	23.8	34.2
4 kernels	5.1	8.0	28.3	36.6
5 kernels	5.6	9.2	28.6	42.6
6 kernels	6.2	10.3	30.2	47.5
7 kernels	6.2	11.5	30.3	54.7
8 kernels	6.7	12.6	32.3	55.0
9 kernels	6.7	16.8	35.7	58.3
10 kernels	6.8	17.0	36.5	63.0

MKLMM for binary phenotypes

Binary LMMs model the distribution of a binary trait y. To adapt MKLMM to the binary case, we adopt the liability threshold model (Dempster and Lerner 1950; Golan and Rosset 2014), which associates every individual i with a latent normally distributed variable l_i called the *liability*, such that cases are individuals whose liability exceeds a given cutoff t. It is typically assumed that the liability has a unit variance, in which case t is determined according to the trait prevalence K, $t = \Phi^{-1}(1-K)$, where Φ is the standard normal cumulative density. The liability is typically assumed to arise as the sum of two independent normally distributed terms, $l_i = g_i + e_i$, where g_i is called the *genetic effect*, and the zero mean normal variable e_i is called the *environmental effect*.

Given a sample of individuals with a variants matrix $\boldsymbol{X} = [\boldsymbol{X}_1\ \boldsymbol{X}_2 \ldots \boldsymbol{X}_n]^T$**,** a covariates matrix $\boldsymbol{C} = [\boldsymbol{C}_1\ \boldsymbol{C}_2 \ldots \boldsymbol{C}_n]^T$**,** a phenotypes vector $\boldsymbol{y} = [y_1, y_2, \ldots, y_n]^T$ and a genetic effects vector $\boldsymbol{g} = (g_1, g_2, \ldots, g_n)^T$, The posterior liability for a tested individual given $\boldsymbol{g}$ is normally distributed, $l_* | \boldsymbol{X}, \boldsymbol{C}, \boldsymbol{g}, \boldsymbol{C}_*, \boldsymbol{X}_* \sim N(\mu_*(\boldsymbol{g}), \sigma_*^2 + \sigma_e^2)$, where σ_e^2 is the variance of e_i. Following Equation 2 in the main text, the distribution parameters are given by

$$\mu_*(\boldsymbol{g}) = \boldsymbol{C}_*^T\boldsymbol{\beta} + \boldsymbol{g}_*^T(\boldsymbol{\theta})\boldsymbol{G}(\boldsymbol{X};\boldsymbol{\theta})^{-1}(\boldsymbol{g} - \boldsymbol{C}\boldsymbol{\beta})$$

$$\sigma_*^2 = \ g_{**}(\boldsymbol{\theta}) - \boldsymbol{g}_*^T(\boldsymbol{\theta})\boldsymbol{G}(\boldsymbol{X};\boldsymbol{\theta})^{-1}\boldsymbol{g}_*(\boldsymbol{\theta}). \qquad (15)$$

Therefore, conditional on the genetic effects $\boldsymbol{g}$ of the training individuals, risk prediction can be computed via a closed form formula,

$$\begin{aligned} P(y_* = 1 | \boldsymbol{X}, \boldsymbol{C}, \boldsymbol{g}, \boldsymbol{C}_*, \boldsymbol{X}_*; \boldsymbol{\beta}, \boldsymbol{\theta}, \sigma_e^2) &= P(l_* \geq t | \boldsymbol{X}, \boldsymbol{C}, \boldsymbol{g}, \boldsymbol{C}_*, \boldsymbol{X}_*; \boldsymbol{\beta}, \boldsymbol{\theta}, \sigma_e^2) \\ &= \Phi\left((\mu_*(\boldsymbol{g}) - t)/\sqrt{\sigma_*^2 + \sigma_e^2}\right). \qquad (16) \end{aligned}$$

In practice, the vector $\boldsymbol{g}$ is not observed, and thus risk estimation is more involved. When only the training set phenotypes vector $\boldsymbol{y}$ is given, the estimated risk is given by

$$\begin{aligned} &P(y_* = 1 | \boldsymbol{X}, \boldsymbol{C}, \boldsymbol{y}, \boldsymbol{C}_*, \boldsymbol{X}_*; \boldsymbol{\beta}, \boldsymbol{\theta}, \sigma_e^2) \\ &\qquad = \int_{\boldsymbol{g}} P(\boldsymbol{g} | \boldsymbol{X}, \boldsymbol{C}, \boldsymbol{y}; \boldsymbol{\beta}, \boldsymbol{\theta}) \Phi\left((\mu_*(\boldsymbol{g}) - t)/\sqrt{\sigma_*^2 + \sigma_e^2}\right) d\boldsymbol{g}. \qquad (17) \end{aligned}$$

We conclude that risk estimation for binary phenotypes under LMMs amounts to computing a high dimensional integral, which cannot be solved analytically. Nevertheless, several effective approximation methods exist. One approach is to approximate the integral via Gibbs sampling, as recently proposed (Golan and Rosset 2014). Another approach is to approximate the posterior distribution $P(\boldsymbol{g}|\boldsymbol{X},\boldsymbol{C},\boldsymbol{y};\boldsymbol{\beta},\boldsymbol{\theta},\sigma_e^2)$ via a normal distribution, which can render Equation 17 tractable. Such approximations have recently gained considerable interest in both the machine learning (Nickisch and Rasmussen 2008) and Bayesian statistics (Rue et al. 2009) communities. In particular, the Laplace approximation, which approximates the posterior distribution via a second order Taylor approximation around the maximum a posteriori value, is known to be computationally efficient on the one hand and highly accurate on the other (Nickisch and Rasmussen 2008). It is therefore possible to efficiently approximate Equation 17 with a high degree of accuracy.

Parameter estimation for binary phenotypes

While binary phenotypes prediction is relatively simple when the model parameters $\boldsymbol{\beta},\boldsymbol{\theta},\sigma_e^2$ are known, estimating the model parameters poses a greater challenge. Here, it is important to distinguish between randomly ascertained and ascertained samples, wherein cases are oversampled relative to the trait prevalence.

Under a randomly ascertained sample, the liability threshold model states that $\boldsymbol{g}$ is normally distributed in the sample. In such cases, one can efficiently approximate the maximum-likelihood estimate by approximating the likelihood $P(\boldsymbol{y}|\boldsymbol{X},\boldsymbol{C};\boldsymbol{\beta},\boldsymbol{\theta},\sigma_e^2)$ via the Laplace approximation, and inferring the parameters via conjugate gradient ascent (Nickisch and Rasmussen 2008).

In the presence of ascertainment, the genetic effects vector $\boldsymbol{g}$ is no longer normally distributed in the sample (Golan and Rosset 2014), and the assumptions of the Laplace approximation are therefore no longer accurate. When a single linear kernel is used, one possible approach is estimating the fixed effects via logistic regression to estimate the affection thresholds, and then employing a Taylor approximation-based moments estimator for the covariance matrix parameters, which takes the ascertainment procedure into account (Golan et al. 2014). However, the Taylor approximation is accurate only when the entries of the covariance matrix $\boldsymbol{G}(\boldsymbol{X};\boldsymbol{\theta})$ are small, which may not be the case under more complex kernels.

Another option is to treat the phenotype as if it were normally distributed and estimate the model parameters as described in the main text. This is the approach adopted by several recently proposed methods, which suggested treating binary phenotypes as if they were normally distributed (Zhou et al. 2013; Speed and Balding 2014; Moser et al. 2015).

We have carried out an empirical evaluation of the three parameter estimation approaches on both simulated and real data sets. Our evaluation found that when complex kernels are being used, the third approach outperforms the other two in the majority of cases, in spite of its inaccurate assumptions (results not shown). The moments estimator approach is not robust to the large matrix entries that are sometimes encountered in the presence of complex kernels, while the Laplace approximation yields estimates that are almost identical to the ones of the third approach, at a substantially increased computational cost. We conclude that efficient parameter estimation for binary LMMs under ascertainment remains an open research problem.

Binary phenotype simulations

To simulate ascertained data for a binary trait with prevalence K, we used the assumptions of the liability threshold model. Namely, we first generated a large data set with 500,000 synthetic genotypes and phenotypes, and then determined the affection cutoff as the $1-K$ empirical percentile of the phenotypes. Afterwards, 1,400 individuals with phenotype exceeding this cutoff were designated as cases, and 1,401 of the other individuals were designated as controls. Genotypes were generated by treating each SNP as a Binom(2) distributed random variable, using the empirical minor allele frequencies of the 2,801 individuals used in all other experiments.

The Bayesian interpretation of MKLMM

MKLMM readily admits a Bayesian interpretation. Under this interpretation, MKLMM is a linear regression model wherein effect sizes are iid normally distributed. Recall that under MKLMM, the phenotype is normally distributed, $\boldsymbol{y}|\boldsymbol{X},\boldsymbol{C}\sim N(\boldsymbol{C\beta},\boldsymbol{G}(\boldsymbol{X};\theta)+\sigma_e^2\boldsymbol{I})$. Further recall that according to the Mercer theorem, every covariance matrix is associated with a transformation function $\varphi: R^m\rightarrow R^M$ that projects genotype vectors into a high dimensional space, such that $\boldsymbol{G}(\boldsymbol{X};\boldsymbol{\theta})=\varphi(\boldsymbol{X})\varphi(\boldsymbol{X})^T$, where φ is invoked on each row of the matrix $\boldsymbol{X}$ (each individual) separately. Using basic properties of the normal distribution, the (normal) density of $\boldsymbol{y}$ is given by

$$\mathrm{P}(\mathbf{y}|\mathbf{X},\mathbf{C};\boldsymbol{\beta},\boldsymbol{\theta},\sigma_e^2)=\int \phi(\mathbf{y};\ \boldsymbol{C\beta}+\varphi(\boldsymbol{X})\boldsymbol{\gamma},\sigma_e^2\boldsymbol{I})\phi\left(\boldsymbol{\gamma};\ \mathbf{0},\frac{1}{M}\boldsymbol{I}\right)\mathrm{d}\boldsymbol{\gamma}, \qquad (18)$$

where ϕ is the normal density, and the parameters $\boldsymbol{\theta}$ are used implicitly by φ. We conclude that MKLMM can be written as the following Bayesian model:

$$\boldsymbol{y}=\boldsymbol{C\beta}+\varphi(\boldsymbol{X})\boldsymbol{\gamma}+\boldsymbol{\epsilon}$$

$$\boldsymbol{\gamma}\sim N\left(\mathbf{0},\frac{1}{M}\boldsymbol{I}\right)$$

$$\boldsymbol{\epsilon}\sim N(0,\sigma_e^2\boldsymbol{I}).$$

We now describe how the linear kernel can be derived via a slight transformation of the saturating pathways kernel in Equation 5. To derive the linear kernel from Equation 5, we slightly rearrange the model for the linear kernel as follows:

$$\boldsymbol{y}=\boldsymbol{C\beta}+X\boldsymbol{\gamma}+\boldsymbol{\epsilon}$$

$$\boldsymbol{\gamma}\sim N\left(\mathbf{0},\frac{\theta}{m}\boldsymbol{I}\right)$$

$$\boldsymbol{\epsilon}\sim N(0,\sigma_e^2\boldsymbol{I}).$$

The linear kernel can be directly derived from the saturating pathways kernel in Equation 5, by setting $R=1$, $h(z)=z$, $\theta_i^p=1$, $\theta_i^w=\boldsymbol{\gamma}$.

Privacy-preserving phenotype prediction

A key feature of MKLMM is its ability to perform genetic-similarity based prediction without having to store the genotypes and phenotypes of the training sample. Exact computations are possible for kernels with a finite-dimensional underlying transformation, while the computations for infinite-dimensional kernels can be approximated to an arbitrary degree of accuracy. We first explain how privacy preservation is achieved, and then provide mathematical proofs for our claims.

Privacy preserving prediction for finite-dimensional kernels

We begin by describing exact privacy preserving prediction for finite-dimensional kernel transformations. Our main tool is the fact that the vector $\boldsymbol{g}_*(\boldsymbol{\theta})$ in Equation 2 in the main text, which describes genotypic covariance between each training individual and the tested individual, can be factored as $\boldsymbol{g}_*(\boldsymbol{\theta}) = \varphi^U(\boldsymbol{X};\boldsymbol{\theta})\varphi^V(\boldsymbol{X}_*;\boldsymbol{\theta})$, where M is the dimension of the feature space, the function $\varphi^V: R^m \to R^M$ transforms genotype vectors, and the function $\varphi^U: R^{n\times m} \to R^{n\times M}$ transforms each row of a genotypes matrix (each individual) separately. To simplify notation, in the remainder of this section we write $\varphi(\boldsymbol{X};\boldsymbol{\theta})$ for both transformation types, because the type of φ can always be inferred from its argument. The Mercer theorem states that for every possible kernel there exists a corresponding function φ. For example, under the linear kernel we have $M = m$, $\boldsymbol{g}_*(\theta) = \theta\frac{1}{m}\boldsymbol{X}\boldsymbol{X}_*$, and the corresponding function is $\varphi(\boldsymbol{Z};\boldsymbol{\theta}) = \sqrt{\frac{\theta}{m}}\boldsymbol{Z}$. The function φ enables computing the posterior distribution of a predicted phenotype without storing genotypes or phenotypes of training individuals, as we now demonstrate.

Using the factorization of $\boldsymbol{g}_*(\boldsymbol{\theta})$, the predicted mean and variance in Equation 2 in the main text can be rewritten as follows:

$$\mu_* = \boldsymbol{C}_*^T\boldsymbol{\beta} + \varphi(\boldsymbol{X}_*;\boldsymbol{\theta})^T\varphi(\boldsymbol{X};\boldsymbol{\theta})^T(\boldsymbol{G}(\boldsymbol{X};\boldsymbol{\theta}) + \sigma_e^2\boldsymbol{I})^{-1}(\boldsymbol{y} - \boldsymbol{C\beta})$$

$$\sigma_*^2 = g_{**}(\boldsymbol{\theta}) - \varphi(\boldsymbol{X}_*;\boldsymbol{\theta})^T\varphi(\boldsymbol{X};\boldsymbol{\theta})^T(\boldsymbol{G}(\boldsymbol{X};\boldsymbol{\theta}) + \sigma_e^2\boldsymbol{I})^{-1}\varphi(\boldsymbol{X};\boldsymbol{\theta})\varphi(\boldsymbol{X}_*;\boldsymbol{\theta}). \qquad (19)$$

Equation 19 can now be rewritten as follows:

$$\mu_* = \boldsymbol{C}_*^T \boldsymbol{\beta} + \varphi(\boldsymbol{X}_*; \boldsymbol{\theta})^T \boldsymbol{H}$$

$$\sigma_*^2 = \ g_{**}(\boldsymbol{\theta}) - \varphi(\boldsymbol{X}_*; \boldsymbol{\theta})^T \boldsymbol{W} \varphi(\boldsymbol{X}_*; \boldsymbol{\theta}), \qquad (20)$$

where the vector $\boldsymbol{H}$ and the matrix $\boldsymbol{W}$ are independent of the tested genotype, and are given by

$$\boldsymbol{H} = \varphi(\boldsymbol{X}; \boldsymbol{\theta})^T (\boldsymbol{G}(\boldsymbol{X}; \boldsymbol{\theta}) + \ \sigma_e^2 \boldsymbol{I})^{-1} (\boldsymbol{y} - \boldsymbol{C}\boldsymbol{\beta}) \qquad (21)$$

$$\boldsymbol{W} = \varphi(\boldsymbol{X}; \boldsymbol{\theta})^T (\boldsymbol{G}(\boldsymbol{X}; \boldsymbol{\theta}) + \ \sigma_e^2 \boldsymbol{I})^{-1} \varphi(\boldsymbol{X}; \boldsymbol{\theta}). \qquad (22)$$

Equation 20 has an intuitive interpretation under the Bayesian view of LMMs, described above. Under this view, LMMs are equivalent to a linear regression model, wherein all effect sizes are iid normally distributed. The vector $\boldsymbol{H}$ can therefore be viewed as the posterior mean of the effect sizes vector, whereas the matrix $\boldsymbol{W}$ is given by $\boldsymbol{W} = \boldsymbol{I} - \boldsymbol{\Sigma}$, where $\boldsymbol{\Sigma}$ is the covariance matrix of the posterior effect sizes distribution.

We conclude that phenotype prediction can be performed by computing $\boldsymbol{H}$ and $\boldsymbol{W}$ only once, and then discarding the original genotypes matrix $\boldsymbol{X}$ and phenotypes vector $\boldsymbol{y}$**.** It is clear that $\boldsymbol{X}$ and $\boldsymbol{y}$ cannot be recovered from $\boldsymbol{H}$**,** because there are infinitely many such matrix-vector pairs leading to the same vector $\boldsymbol{H}$ (see proof below)**.** It is also easy to show that $\boldsymbol{X}$ cannot be recovered from $\boldsymbol{W}$, since it is invariant to rotations of $\varphi(\boldsymbol{X}; \boldsymbol{\theta})$**,** indicating that there are infinitely many matrices leading to the same matrix $\boldsymbol{W}$ (see proof below). We note that the matrix $\boldsymbol{W}$ has dimensions $M \times M$, which can make its storage unwieldy. However, it can be decomposed into a product of matrices of dimensions $n \times M$ (where n is the training set size) which alleviates this concern, as described below.

When using a composite kernel that is a weighted sum of simpler kernels, we define a composite kernel transformation which concatenates matrices horizontally and vectors vertically. For example, for the kernel $\boldsymbol{G}(\boldsymbol{X}; \boldsymbol{\theta}_1, \boldsymbol{\theta}_2) = \boldsymbol{G}_1(\boldsymbol{X}; \boldsymbol{\theta}_1) + \boldsymbol{G}_2(\boldsymbol{X}; \boldsymbol{\theta}_2)$ we define

$\boldsymbol{g}_*(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2) = [\varphi_1(\boldsymbol{X}; \boldsymbol{\theta}_1)\ \varphi_2(\boldsymbol{X}; \boldsymbol{\theta}_2)] \begin{bmatrix} \varphi_1(\boldsymbol{X}_*; \boldsymbol{\theta}_1) \\ \varphi_2(\boldsymbol{X}_*; \boldsymbol{\theta}_2) \end{bmatrix}$, where φ_1, φ_2 are the underlying transformations for kernels $\boldsymbol{G}_1$ and $\boldsymbol{G}_2$, respectively.

Privacy preserving prediction for infinite-dimensional kernels

When the kernel transformation φ has an infinite dimensionality (as in the SP kernel), the procedure above cannot be used, because explicit computation of $\varphi(\boldsymbol{Z}; \boldsymbol{\theta})$ is impossible. However, the kernel transformation for many kernel types can be approximated as a finite-dimensional transformation. Approximation of kernels via finite transformations is an active research topic, and many recent works have shown that finite approximations can substantially simplify kernel methods with a negligible loss of accuracy (Le et al. 2013; Yang et al. 2015).

The saturating pathways kernel is particularly suitable for a finite approximation, as it is explicitly derived by applying the central limit theorem for an asymptotic expansion of an infinite number of saturating pathways. It is therefore straightforward to approximate the underlying transformation of this kernel by sampling a finite but large number of pathways. The central limit theorem guarantees that the approximation error is proportional to the square root of the number of sampled pathways, indicating that accuracy can be increased to an arbitrary degree by sampling additional pathways without compromising genomic privacy.

Proofs of privacy-preserving claims

Here we provide proofs for several claims regarding privacy-preserving phenotype prediction. We base the following proofs on the spectral decomposition of $\varphi(\boldsymbol{X}; \boldsymbol{\theta})$, which we now derive. We first rewrite the matrix $\boldsymbol{G}(\boldsymbol{X}; \boldsymbol{\theta})$ as a matrix product in a high dimensional space, $\boldsymbol{G}(\boldsymbol{X}; \boldsymbol{\theta}) = \varphi(\boldsymbol{X}; \boldsymbol{\theta})\varphi(\boldsymbol{X}; \boldsymbol{\theta})^T$. Equations 21 and 22 can now be written as

$$\boldsymbol{H} = \varphi(\boldsymbol{X}; \boldsymbol{\theta})^T(\varphi(\boldsymbol{X}; \boldsymbol{\theta})\varphi(\boldsymbol{X}; \boldsymbol{\theta})^T + \sigma_e^2\boldsymbol{I})^{-1}(\boldsymbol{y} - \boldsymbol{C\beta}) \qquad (23)$$

$$\boldsymbol{W} = \varphi(\boldsymbol{X}; \boldsymbol{\theta})^T(\varphi(\boldsymbol{X}; \boldsymbol{\theta})\varphi(\boldsymbol{X}; \boldsymbol{\theta})^T + \sigma_e^2\boldsymbol{I})^{-1}\varphi(\boldsymbol{X}; \boldsymbol{\theta}). \qquad (24)$$

Next, we rewrite Equations 23 and 24 via the singular value decomposition (SVD) of $\varphi(\boldsymbol{X};\boldsymbol{\theta}) = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T$. Using the orthonormality of $\boldsymbol{U}$ and denoting $\operatorname{diag}(\boldsymbol{S}) = \boldsymbol{s}$ (where a lower case $\boldsymbol{s}$ indicates a vector rather than a diagonal matrix), Equations 23 and 24 can be rewritten as follows:

$$\boldsymbol{H} = \boldsymbol{V}\boldsymbol{S}\boldsymbol{U}^T\left((\boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T)(\boldsymbol{V}\boldsymbol{S}\boldsymbol{U}^T) + \sigma_e^2\boldsymbol{I}\right)^{-1}(\boldsymbol{y} - \boldsymbol{C}\boldsymbol{\beta})$$

$$= \boldsymbol{V}\boldsymbol{S}\boldsymbol{U}^T\,(\boldsymbol{U}\boldsymbol{S}^2\boldsymbol{U}^T + \sigma_e^2\boldsymbol{U}\boldsymbol{U}^T)^{-1}(\boldsymbol{y} - \boldsymbol{C}\boldsymbol{\beta})$$

$$= \boldsymbol{V}\boldsymbol{S}\boldsymbol{U}^T[\boldsymbol{U}(\boldsymbol{S}^2 + \sigma_e^2\boldsymbol{I})\boldsymbol{U}^T]^{-1}(\boldsymbol{y} - \boldsymbol{C}\boldsymbol{\beta})$$

$$= \boldsymbol{V}\boldsymbol{S}\boldsymbol{U}^T(\boldsymbol{U}(\boldsymbol{S}^2 + \sigma_e^2\boldsymbol{I})^{-1}\boldsymbol{U}^T)(\boldsymbol{y} - \boldsymbol{C}\boldsymbol{\beta})$$

$$= \boldsymbol{V}\boldsymbol{S}\operatorname{diag}\left(\frac{1}{\boldsymbol{s}^2 + \sigma_e^2}\right)\boldsymbol{U}^T(\boldsymbol{y} - \boldsymbol{C}\boldsymbol{\beta})$$

$$= \boldsymbol{V}\operatorname{diag}\left(\frac{\boldsymbol{s}}{\boldsymbol{s}^2 + \sigma_e^2}\right)\boldsymbol{U}^T(\boldsymbol{y} - \boldsymbol{C}\boldsymbol{\beta}). \qquad (25)$$

$$\boldsymbol{W} = \boldsymbol{V}\boldsymbol{S}\boldsymbol{U}^T\left((\boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T)(\boldsymbol{V}\boldsymbol{S}\boldsymbol{U}^T) + \sigma_e^2\boldsymbol{I}\right)^{-1}\boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T$$

$$= \boldsymbol{V}\boldsymbol{S}\boldsymbol{U}^T\,(\boldsymbol{U}\boldsymbol{S}^2\boldsymbol{U}^T + \sigma_e^2\boldsymbol{U}\boldsymbol{U}^T)^{-1}\boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T = \boldsymbol{V}\boldsymbol{S}\boldsymbol{U}^T[\boldsymbol{U}(\boldsymbol{S}^2 + \sigma_e^2\boldsymbol{I})\boldsymbol{U}^T]^{-1}\boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T$$

$$= \boldsymbol{V}\boldsymbol{S}\boldsymbol{U}^T(\boldsymbol{U}(\boldsymbol{S}^2 + \sigma_e^2\boldsymbol{I})^{-1}\boldsymbol{U}^T)\boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T = \boldsymbol{V}\boldsymbol{S}\operatorname{diag}\left(\frac{1}{\boldsymbol{s}^2 + \sigma_e^2}\right)\boldsymbol{S}\boldsymbol{V}^T$$

$$= \boldsymbol{V}\operatorname{diag}\left(\frac{\boldsymbol{s}^2}{\boldsymbol{s}^2 + \sigma_e^2}\right)\boldsymbol{V}^T. \qquad (26)$$

We conclude that Equations 21 and 22 can be rewritten as

$$\boldsymbol{H} = \boldsymbol{V}\operatorname{diag}\left(\frac{\boldsymbol{s}}{\boldsymbol{s}^2 + \sigma_e^2}\right)\boldsymbol{U}^T(\boldsymbol{y} - \boldsymbol{C}\boldsymbol{\beta}) \qquad (27)$$

$$\boldsymbol{W} = \boldsymbol{V}\operatorname{diag}\left(\frac{\boldsymbol{s}^2}{\boldsymbol{s}^2 + \sigma_e^2}\right)\boldsymbol{V}^T. \qquad (28)$$

We now prove the claims made in the main text.

To prove that $\boldsymbol{W}$ requires less than $\boldsymbol{O}(M^2)$ storage space, we note that Equation 28 shows that $\boldsymbol{W}$ can be computed from the matrix $\boldsymbol{V}$ and the vector **s,** whose storage requirements (when using the economy SVD) are $\boldsymbol{O}(nM)$ and $\boldsymbol{O}(n)$, respectively (where n is the training sample size). To prove that $\boldsymbol{X}$ cannot be recovered from $\boldsymbol{W}$, we notice that $\boldsymbol{W}$ is independent of $\boldsymbol{U}$, indicating that $\boldsymbol{W}$ is invariant to rotations of $\varphi(\boldsymbol{X};\boldsymbol{\theta})$. Finally, to prove that $\boldsymbol{X}$ and $\boldsymbol{y}$ cannot be recovered from $\boldsymbol{H}$, we rearrange Equation 27 as follows:

$$\operatorname{diag}\left(\frac{\boldsymbol{s}^2+\sigma_e^2}{\boldsymbol{s}}\right)\boldsymbol{V}^T\boldsymbol{H} = \boldsymbol{U}^T(\boldsymbol{y}-\boldsymbol{C\beta}). \tag{29}$$

Clearly, even when $\boldsymbol{V}$ and $\boldsymbol{s}$ are known, there are an infinite combinations of orthonormal bases $\boldsymbol{U}$ and vectors $(\boldsymbol{y}-\boldsymbol{C\beta})$ satisfying Equation 29.

Permutation testing

To evaluate the statistical significance of the advantage of each method over another method, we employed a permutation test where the predicted phenotype of each individual under each method was randomly swapped between the two methods 100,000 times and the measure of interest (e.g. AUC) was re-evaluated under each permutation.

A potential concern with this test is that the selection of the optimal number of kernels cannot be evaluated in the permutation test, because the dependence structure in each permutation is different from the one in the original data, owing to the fact that predictions of two different methods are combined. To circumvent this difficulty, we first computed the optimal number of regions for each cross validation fold under each method, where optimality was defined according to the evaluated measure. For example, when AUC was measured, we computed the number of regions that maximizes the AUC in each fold. Afterwards, we associated every individual under every method with a single prediction corresponding to the optimal number of regions for her fold. The permutation test was applied using these predictions.

We additionally evaluated a different permutation test that evaluates the optimal number of kernels under each permutation, which yielded very similar results in practice (results not shown).

Estimating affection probability for binary phenotypes

As explained in the main text, estimating the affection probability of an individual under an LMM is challenging because of the need to consider the ascertainment scheme. Nevertheless, we carried out analyses of ascertained case control studies to obtain a comparison with previous works.

Direct estimation of affection probabilities that ignores the ascertainment scheme is straightforward, because LMMs compute a posterior normal distribution for the phenotype of each tested individual. Assuming that individuals are affected if their predicted phenotype is larger than zero, the affection probability is given by the probability that the normally distributed posterior phenotype is positive. Furthermore, an intercept value can be added to the LMM to maximize the likelihood of the binary phenotype (after obtaining the REML parameter estimates when treating the phenotype as quantitative), and this was done in the experiments.

Real Data preprocessing

In the Mice data set, we followed the preprocessing procedure described in (Speed and Balding 2014). Namely, Single nucleotide polymorphisms (SNPs) were excluded if they had a minor allele frequency <0.01, Hardy Weinberg equilibrium P-value < 10^{-4}, or a missingness rate >1%. Phenotypes were selected for the analysis if measurements were available for at least 1,300 mice, the coefficient of kurtosis was smaller than six, and the phenotype was not binary. Each analysis used sex as a covariate. Age at the experiment time was also used as a covariate when this data was available. Mice with a missing value for a certain phenotypes were excluded from the analysis of this phenotype. In the cross-validation procedure, mice in the same cage were placed in the same fold to prevent leakage. All variants were standardized to have zero mean and unit variance. In all experiments, AMB and MKLMM used a division of the genome into regions of approximately 75kb, using LDAK (Speed and Balding 2014).

In the WTCCC1 data sets, we performed stringent quality control preprocessing to avoid genotyping artifacts from biasing the results (Golan and Rosset 2014). SNPs were excluded if they had minor allele frequency <5%, missingness rates >1%, a significantly different missingness rate between cases and controls, or a significant deviation from Hardy Weinberg equilibrium among the controls group. Controls consisted of individuals from the national blood service control group. The second controls group of C58 birth cohort was excluded from the main experiments to address the concern that the non-linear methods may exploit subtle population structure signals differentiating the two groups. Results for analyses with both control groups are provided in Supplemental Table S6.

Individuals were excluded from the analysis if they were in the WTCCC exclusion lists or if they had missingness rates >1%. We further excluded individuals with a normalized similarity coefficient >0.05 with at least one other individual, by greedily removing individuals according to the number of related individuals they had, until no related individuals remained. To prevent spurious results due to population structure, we projected all genotype vectors to the subspace that is orthogonal to the top 10 principal components. Sex was used as a covariate in all data sets. In all experiments, AMB and MKLMM used a division of the genome into regions of approximately 75kb, using LDAK (Speed and Balding 2014).

In the ulcerative colitis (UC) data set, controls consisted of individuals from the national blood service control group. SNPs were removed if they had >0.5% missing data, p<0.01

for allele frequency difference between two control groups, $p<0.05$ for deviation from Hardy-Weinberg equilibrium, $p<0.05$ for differential missingness between cases and controls, or minor allele frequency <1%. All genotype vectors were projected to the subspace that is orthogonal to the top 10 principal components. Variants within 5kb of the major histocompatibility complex (MHC) were excluded from the analysis, because the MHC region in this data set is strongly associated with population structure, even when excluding the top principal components (Yang et al. 2014). The genome was divided into regions of approximately 75kb, using LDAK (Speed and Balding 2014). Due to the memory requirements incurred by the large data set size, the genome-wide kernel for this analysis was fixed to be a linear kernel.

An important concern in the analysis of case control phenotypes is ascertainment-induced leakage. Leakage can be introduced to the analysis when standardizing variants, because oversampling of cases leads to an overrepresentation of risk alleles in the sample. To prevent such leakage, we computed a weighted mean and variance for each SNP according to the disease prevalence, such that controls were overrepresented to match the true phenotype prevalence (the marginal variance of each SNP can be computed via the law of total variance). We then standardized each variant by subtracting the weighted mean and dividing by the weighted standard deviation. Following (Golan and Rosset 2014), the estimated prevalence for the diseases were CD (0.1%), T1D (0.5%), BD (0.5%), RA (0.5%), T2D (3%), CAD (3.5%), HT (5%) and UC (0.3%).

Simulations procedure

The synthetic phenotype simulations were carried out as follows. We created a data set consisting of 2,801 individuals from the Wellcome trust 2 national blood service controls group and their Chromosome 1 SNPs. In each simulation, we generated a synthetic phenotype by first randomly selecting genomic regions and then generating phenotypes. Ten data sets were created for each unique combination of tested parameters. We first describe the simulations procedure, and then describe the default parameter values used in all experiments.

Genomic regions were selected by first sampling a region size for each region from a Poisson(75,000) distribution, and then randomly selecting a set of consecutive SNPs spanning the selected region size. We also considered an additional region spanning all chromosome-wide SNPs.

For each region, we generated a linear effect, and one or two non-linear effects. Each non-linear effect was either a saturating effect or a groupwise effect (described below). Afterwards, an aggregated linear effect, an aggregated saturating effect and an aggregated groupwise effect were created by summing all region-specific effects of each type (excluding the chromosome-wide region) with randomly sampled mixture weights, designed to differentiate the phenotypic variance explained by different regions. In the next step, a combined chromosome-wide effect and a combined regions effect were created by summing of the aggregated effects and the chromosome-wide effects, respectively. The final phenotype consisted of a weighted sum of the two combined effects, with predetermined mixture weights, and an iid normally distributed environmental effect that was independent of the genotypes. In all simulations, the chromosome-wide region included all three effect types, while each of the other regions included a linear effect and one of the two non-linear effect types, with an equal number of regions for each effect type. We now describe the simulation procedure for each of the three effect types in detail.

The linear effect of each region was generated by drawing effect sizes for all the SNPs in the region from a standard normal distribution, and computing the weighted sum. The value of each linear effect for each individual was given by $\Sigma_i x_i \alpha_i$, where x_i is the (normalized) value of SNP i carried by the individual, α_i is the effect size of variant i, and the index i iterates over all variants in the region.

The saturating effect of each region was generated by sampling 100 pathways. For each pathway, input effect sizes α_i were sampled from a zero-mean normal distribution with a variance of 100, and an output effect size γ was sampled from a normal distribution with a variance of 0.01. Using the same notations as before, the value of each pathway was given by $\gamma \cdot \mathrm{erf}(\Sigma_i x_i \alpha_i)$. The large variance of the input effect sizes is meant to induce non-linear dynamics, because $\mathrm{erf}(z)$ is approximately linear when z is close to zero. Finally, all pathway values were summed to generate an aggregated saturating effect.

The groupwise effect of each region was generated by randomly selecting ten SNP subsets, computing a value for each one and then summing up the values. The effect of each SNP subset consisted of the element-wise multiplication of the selected SNP vectors, multiplied by an effect size drawn from a standard zero-mean normal distribution. Formally, the groupwise effect for region r for a certain individual is given by $\sum_{j=1}^{5} S_j(X^r)\alpha_j$, where X^r is the vector of SNPs in region r carried by the individual, α_j is the effect size of group j and $S_j(X^r) = \prod_{j_i} x_{j_i}$, where the index j_i iterates over SNPs that participate in group j. The SNPs for each group were selected uniformly with replacement, and the group sizes were drawn from a Poisson(2) distribution, conditional on being larger than one.

Unless otherwise stated, in all experiments the regions consisted of two, four or six randomly selected regions with lengths drawn from a Poisson(75,000) distribution with an additional chromosome-wide region. Each region included a linear and either a groupwise or a saturating effect, with an equal number of regions having each of the effect types. The chromosome-wide region included all three effect types. When combining the contribution of each effect of each region to the aggregated effects, the effects were differentiated with random mixture weight drawn from an inverse gamma distribution (the conjugate prior of the variance of the normal distribution) with shape and scale parameters 2 and 1, respectively (to yield a mean variance of 1.0). The aggregated linear effect accounted for 25% of the combined effect, while the two non-linear effects each accounted for 37.5% of the combined effect. The combined effect itself accounted for 50% of the explained phenotypic variance, and the other 50% was drawn from an iid zero mean normal distribution. In all experiments, all variants were standardized to have a zero mean and a unit variance.

References

Dempster ER, Lerner IM. 1950. Heritability of Threshold Characters. *Genetics* **35**(2): 212-236.

Gilmour AR, Thompson R, Cullis BR. 1995. Average information REML: an efficient algorithm for variance parameter estimation in linear mixed models. *Biometrics* **51**: 1440-1450.

Golan D, Lander ES, Rosset S. 2014. Measuring missing heritability: Inferring the contribution of common variants. *Proceedings of the National Academy of Sciences of the United States of America* **111**(49): E5272-5281.

Golan D, Rosset S. 2014. Effective genetic-risk prediction using mixed models. *American journal of human genetics* **95**(4): 383-393.

Hensman J, Fusi N, Lawrence ND. 2013. Gaussian Processes for Big Data. In *Uncertainty in Artificial Intelligence*.

Hornik K. 1993. Some new results on neural network approximation. *Neural Networks* **6**(8): 1069-1072.

Le Q, Sarlos T, Smola A. 2013. Fastfood-computing hilbert space expansions in loglinear time. In *International Conference on Machine Learning*, pp. 244-252.

Lee S, van der Werf J. 2006. An efficient variance component approach implementing an average information REML suitable for combined LD and linkage mapping with a general complex pedigree. *Genetics Selection Evolution* **38**(1): 1-19.

Lippert C, Listgarten J, Liu Y, Kadie CM, Davidson RI, Heckerman D. 2011. FaST linear mixed models for genome-wide association studies. *Nature methods* **8**(10): 833-835.

Morota G, Gianola D. 2014. Kernel-based whole-genome prediction of complex traits: a review. *Front Genet* **5**: 363.

Moser G, Lee SH, Hayes BJ, Goddard ME, Wray NR, Visscher PM. 2015. Simultaneous discovery, estimation and prediction analysis of complex traits using a bayesian mixture model. *PLoS Genet* **11**(4): e1004969.

Neal RM. 1996. *Bayesian Learning for Neural Networks*. Springer New York.

Nickisch H, Rasmussen CE. 2008. Approximations for binary Gaussian process classification. *Journal of Machine Learning Research* **9**: 2035-2078.

Rasmussen CE, Nickisch H. 2010. Gaussian processes for machine learning (GPML) toolbox. *The Journal of Machine Learning Research* **11**: 3011-3015.

Rasmussen CE, Williams CKI. 2006. *Gaussian Processes for Machine Learning*. The MIT Press.

Rue H, Martino S, Chopin N. 2009. Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations. *J R Stat Soc B* **71**(2): 319-392.

Shashua A. 2009. Introduction to machine learning: Class notes 67577. *arXiv preprint arXiv:09043664*.

Snelson E, Ghahramani Z. 2007. Local and global sparse Gaussian process approximations. In *International Conference on Artificial Intelligence and Statistics*, pp. 524-531.

Speed D, Balding DJ. 2014. MultiBLUP: improved SNP-based prediction for complex traits. *Genome Res* **24**(9): 1550-1557.

Yang J, Zaitlen NA, Goddard ME, Visscher PM, Price AL. 2014. Advantages and pitfalls in the application of mixed-model association methods. *Nat Genet* **46**(2): 100-106.

Yang Z, Wilson AG, Smola AJ, Song L. 2015. A la Carte - Learning Fast Kernels. In *International Conference on Artificial Intelligence and Statistics*.

Zhou X, Carbonetto P, Stephens M. 2013. Polygenic modeling with bayesian sparse linear mixed models. *PLoS Genet* **9**(2): e1003264.

Zuk O, Hechter E, Sunyaev SR, Lander ES. 2012. The mystery of missing heritability: Genetic interactions create phantom heritability. *Proceedings of the National Academy of Sciences* **109**(4): 1193-1198.