RESEARCH   **Open Access**

Determination of nonlinear genetic architecture using compressed sensing

Chiu Man Ho[*†] and Stephen DH Hsu[†]

Abstract

Background: One of the fundamental problems of modern genomics is to extract the genetic architecture of a complex trait from a data set of individual genotypes and trait values. Establishing this important connection between genotype and phenotype is complicated by the large number of candidate genes, the potentially large number of causal loci, and the likely presence of some nonlinear interactions between different genes. Compressed Sensing methods obtain solutions to under-constrained systems of linear equations. These methods can be applied to the problem of determining the best model relating genotype to phenotype, and generally deliver better performance than simply regressing the phenotype against each genetic variant, one at a time. We introduce a Compressed Sensing method that can reconstruct nonlinear genetic models (i.e., including epistasis, or gene-gene interactions) from phenotype-genotype (GWAS) data. Our method uses L1-penalized regression applied to nonlinear functions of the sensing matrix.

Results: The computational and data resource requirements for our method are similar to those necessary for reconstruction of linear genetic models (or identification of gene-trait associations), assuming a condition of *generalized sparsity*, which limits the total number of gene-gene interactions. An example of a sparse nonlinear model is one in which a typical locus interacts with several or even many others, but only a small subset of *all possible* interactions exist. It seems plausible that most genetic architectures fall in this category. We give theoretical arguments suggesting that the method is nearly optimal in performance, and demonstrate its effectiveness on broad classes of nonlinear genetic models using simulated human genomes and the small amount of currently available real data. A phase transition (i.e., dramatic and qualitative change) in the behavior of the algorithm indicates when sufficient data is available for its successful application.

Conclusion: Our results indicate that predictive models for many complex traits, including a variety of human disease susceptibilities (e.g., with additive heritability $h^2 \sim 0.5$), can be extracted from data sets comprised of $n_\star \sim 100s$ individuals, where s is the number of distinct causal variants influencing the trait. For example, given a trait controlled by ~ 10 k loci, roughly a million individuals would be sufficient for application of the method.

Keywords: Genomics, Compressed sensing, Nonlinear interactions

Background

Realistic models relating complex traits (i.e., phenotypes) to genotypes may exhibit nonlinearity (epistasis), allowing distinct regions of DNA to interact with one another. For example, one allele can influence the effect of another, altering its magnitude or sign, even silencing the second allele entirely. For some traits, the largest component of

genetic variance is linear (additive) [1], but even in this case nonlinear interactions accounting for some smaller component of variance are expected to be present. To obtain the best possible model for prediction of phenotype from genotype, or to obtain the best possible understanding of the genetic architecture, requires the ability to extract information concerning nonlinearity from phenotype–genotype (e.g., GWAS) data. In this paper we describe a computational method for this purpose.

*Correspondence: cmho@msu.edu
[†]Equal contributors
Department of Physics and Astronomy, Michigan State University, 567 Wilson Road, 48824 East Lansing, MI, USA

Our method makes use of compressive sensing (CS) [2–5], a framework originally developed for recovering sparse signals acquired from a linear sensor. The application of CS to genomic prediction (using linear models) and GWAS has been described in an earlier paper by one of the authors [6]. Before describing the new application, we first summarize results from [6].

Compressed sensing allows efficient solution of underdetermined linear systems:

$$y = Ax + \epsilon, \tag{1}$$

(ϵ is a noise term) using a form of penalized regression. L1 penalization, or LASSO, involves minimization of an objective function over candidate vectors $\hat{x}$:

$$O = ||y - A\hat{x}||_{L2} + \lambda ||\hat{x}||_{L1}, \tag{2}$$

where the penalization parameter is determined by the noise variance (see Analysis section for more detail). Because O is a convex function it is easy to minimize. Recent theorems [2–5] provide performance guarantees, and show that the $\hat{x}$ that minimizes O is overwhelmingly likely to be the sparsest solution to (1). In the context of genomics, y is the phenotype, A is a matrix of genotypes (in subsequent notation we will refer to it as g), x a vector of effect sizes, and the noise is due to nonlinear gene-gene interactions and the effect of the environment.

Let p be the number of variables (i.e., dimensionality of x, or number of genetic loci), s the sparsity (number of variables or loci with nonzero effect on the phenotype; i.e., nonzero entries in x) and n the number of measurements of the phenotype (i.e., dimensionality of y or the number of individuals in the sample). Then A is an $n \times p$ dimensional matrix. Traditional statistical thinking suggests that $n > p$ is required to fully reconstruct the solution x (i.e., reconstruct the effect sizes of each of the loci). But recent theorems in compressed sensing [2–6] show that $n > Cs \log p$ (for constant C defined over a class of matrices A) is sufficient if the matrix A has the right properties (is a good compressed sensor). These theorems guarantee that the performance of a compressed sensor is nearly optimal – within an overall constant of what is possible if an oracle were to reveal in advance which s loci out of p have nonzero effect. In fact, one expects a phase transition in the behavior of the method as n crosses a critical threshold $n_\star$ given by the inequality. In the good phase ($n > n_\star$), full recovery of x is possible.

In [6], it is shown that

a. Matrices of human SNP genotypes are good compressed sensors and are in the universality class of (i.e., have the same phase diagram as) random (Gaussian) matrices. The phase diagram is a function of sparsity s and sample size n rescaled by

dimensionality p. Given this result, simulations can be used to *predict* the sample size threshold for future genomic analyses.

b. In applications with real data the phase transition can be detected from the behavior of the algorithm as the amount of data n is varied. (For example, in the low noise case the mean p-value of selected, or nonzero, components of x exhibits a sharp jump at $n_\star$.) A priori knowledge of s is not required; in fact one deduces the value of s this way.

c. For heritability $h^2 = 0.5$ and $p \sim 10^6$ SNPs, the value of $C \log p \sim 30$. For example, a trait which is controlled by $s = 10k$ loci would require a sample size of $n \sim 300k$ individuals to determine the (linear) genetic architecture (i.e., to determine the full support, or subspace of nonzero effects, of x).

Our algorithm for dealing with nonlinear models is described in more detail in the Methods section below. It exploits the fact that although a genetic model $G(g)$ with epistasis depends nonlinearly on g, it only depends linearly on the interaction parameters X which specify the interaction coefficients (i.e., z and Z in Eq. (7)). Briefly, the method proceeds in two Steps:

Step 1. Run CS on (y, g) data, using linear model (8). Determine support of x: subset defined by s loci of nonzero effect.

Step 2. Compute $G(g)$ over this subspace. Run CS on $y = G(g) \cdot X$ model to extract nonzero components of X. These can be translated back into the linear and nonlinear effects of the original model (i.e., nonzero components of z and Z).

In the following section we show that in many cases Steps 1 and 2 lead to very good reconstruction of the original model (7) given enough data n. A number of related issues are discussed:

a. When can nonlinear effects hide causal loci from linear regression (Step 1)? In cases of this sort the locus in question would not be discovered by GWAS using linear methods.

b. Both matrices g and $G(g)$ seem to be well-conditioned CS matrices. The expected phase transitions in algorithm performance are observed for both Steps.

c. For a given partition of variance between linear (L), nonlinear (NL) and IID error ϵ, how much data $n_\star$ is required before complete selection of causal variants occurs (i.e., crossing of the phase boundary for algorithm performance)? Typically, if Step 1 is successful then with the same amount of data Step 2 will also succeed.

Relation to earlier work

Here we give a brief discussion of earlier work, with the goal of clarifying what is new and distinctive about our technique. Reviews of earlier methods aimed at detecting gene-gene interactions can be found in [7–10].

Theoretical results concerning LASSO can be found in, e.g., [11, 12], as well as in the well-known work by Candes, Tao, Donoho and collaborators [2–5]. The main point, as discussed in [6], is that despite the beautiful results in this literature there is no theorem that can be specifically applied to matrices formed of genomes, or of nonlinear functions of genomes, because of nontrivial structure: correlations between the genomes of individuals in a population. This structure implies that the matrices of interest deviate at least slightly from random matrices, and have only empirically-determined properties. Therefore, all of the rigorous theorems *are merely guides to our intuition or expectations*: empirical results from simulations are necessary to proceed further. As noted in [6], the best one can do is to test the phase transition properties of real genomic matrices and determine whether they are in the same category ("universality class") as random matrices. This being the case (as verified in [6]), one can then use the phase diagram to *predict* how much data n is required to determine the optimal genomic model for a given sparsity s, dimensionality p, and composition of variance (L, NL, IID error).

The key result is that once the phase boundary is crossed to the favorable part of the phase diagram (i.e., sufficient data is available), the support of the candidate vector $\hat{x}$ will coincide with the support of the optimal x: i.e., the solution of equation (2) in the limit of infinite data, or the sparsest solution to (1). Note that in the presence of noise the specific values of the components of $\hat{x}$ will differ from those of x; it is the *locations* of the nonzero components (support of x) that can be immediately extracted once the data threshold $n_\star$ is surpassed.

The phase transition properties of compressed sensing algorithms are under-appreciated, especially so given their practical utility. We refer the reader to pioneering work by Donoho and collaborators [13–17]. (See also discussion in [5] in end notes of Chapter 9.) Briefly, they showed that the phase behavior of CS algorithms is related to high dimensional combinatorial geometry, and that a broad class of random matrices (matrices randomly selected from a specific ensemble, such as Gaussians) *exhibit the same phase diagram* (i.e., are in the same universality class) as a function of sparsity and data size rescaled by dimensionality: ($\rho = s/n$, $\delta = n/p$). In [6] the same phase boundary was found for matrices of human SNP genotypes, verifying that genomic matrices fall into the same universality class as found in [13]. In other words, the diagram in Fig. 1, constructed in [6] from simulations with real human genotypes, coincides exactly
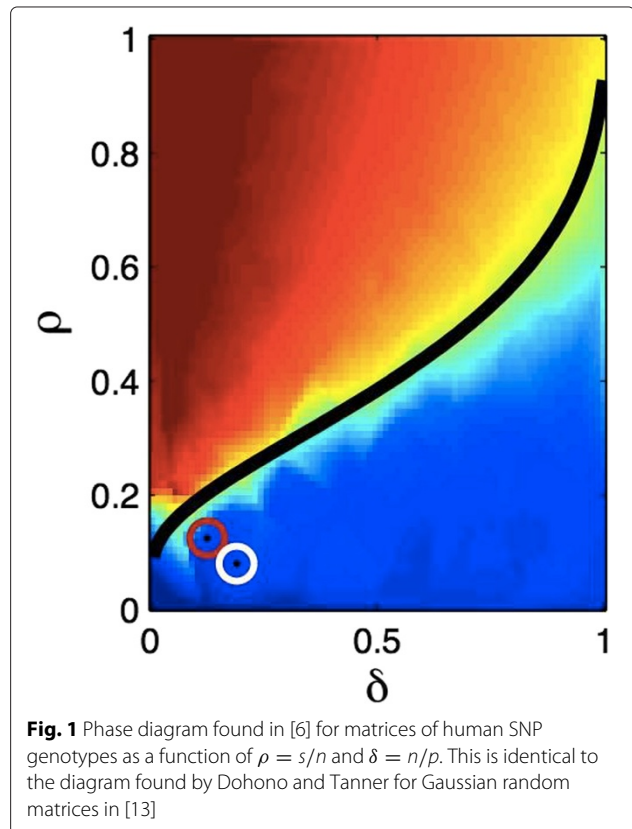

Fig. 1 Phase diagram found in [6] for matrices of human SNP genotypes as a function of $\rho = s/n$ and $\delta = n/p$. This is identical to the diagram found by Dohono and Tanner for Gaussian random matrices in [13]

with the one found in [13] for broad classes of random matrices.

This paper extends the analysis of [6] to nonlinear genomics models incorporating epistasis. We make use of the phase transition to first identify the subspace of loci with nonzero effects and then use L1-penalized regression to fit a nonlinear model defined on this subspace. We find that the nonlinear genomic matrix $G(g)$ is also a good compressed sensor – it exhibits phase transition behavior similar to that of g. In practical terms, our methods should allow the recapture of a large part of nonlinear variance in phenotype prediction.

Some alternative methods for dealing with epistasis are given in [18–23] (for an overview of several methods, see [24]). The closest proposal to the one examined in this paper is that of Devlin and collaborators [22, 23], which uses LASSO and a linear model to first narrow the subspace of interest to lower dimensionality, and then uses an interacting model to estimate nonlinear interactions. The main difference with our work is that (i) [22, 23] do not exploit the phase transition properties that allow one to determine a critical data threshold beyond which nearly ideal selection of loci has occurred, and (ii) [22, 23] do not situate their results in the theoretical framework of compressed sensing. In practical terms, as discussed below, we use signals such as the median p-value of candidate loci to

determine when the calculation is in the favorable part of the phase diagram.

The motivation of earlier work [18–23] seems to be mainly exploratory – to find specific cases of nonlinear interactions, but not to "solve" genetic architectures involving hundreds or thousands of loci. This is understandable given that only a few years ago the largest data sets available were much smaller than what is available today (e.g., hundreds of thousands of individuals, rapidly approaching a million). Our perspective is quite different: we want to establish a lower bound on the total amount of genetic variation (linear plus nonlinear) that can be recaptured for predictive purposes, given data sets that have crossed the phase boundary into the region of good recovery. As we discuss below, our main result is that for large classes of nonlinear genetic architectures one can recapture nearly all of the linear variance and half or more of the nonlinear variance using an amount of data that is somewhat beyond the phase transition for the linear case. We do not claim to have shown that our method performs better than all possible alternative methods. However, given what is known from theoretical analyses in CS, we expect our method to be close to optimal, up to logarithmic corrections and a possible improvement of the constant factor C appearing in the relation $n \sim Cs \log p$. For more detailed comparisons of earlier methods, see [24].

Data Description

Given that current data sets suitable to our analysis are extremely limited, most of our simulations are performed using synthetic genomes with the minor allele frequency (MAF) restricted to values between 0.05 and 0.5. The synthetic genomes are determined as follows: generate a random population-level MAF $\in (0.05, 0.5)$ for each locus, then populate each individual genome with 0,1,2 SNP values according to the MAF for each locus.

To assess how our method works on actual human data as compared to its performance on on synthetic genomes, we tested it on an available data set: human SNP genomes from the 1000 Genomes Project (i.e., matrices g obtained via variant calls on actual genomes from the 1000 Genomes Project). Details concerning sequencing and SNP calling for the 1000 Genomes Project can be found at: http://www.1000genomes.org/analysis.

Analysis

We study the performance of our algorithm on two specific classes of "biologically inspired" models, although we believe our results are generic for any nonlinear models with similar levels of sparsity, nonlinear variance, etc. The models described below are used to generate phenotype data y for a given set of genotypes. Our algorithm then tries to recover the model parameters. Each of the models below is a specific instance of the general class of

nonlinear models in equation (7); this class is the set of all possible models including linear and bi-linear (gene-gene interaction) effects. The model parameters α, β, γ used below can be rewritten in terms of the variables z, Z in (7).

The first category of models is the block-diagonal (BD) interaction model:

$$y^a = \sum_{i=1}^{s} \alpha_i \, g_i^a + \sum_{i=1}^{s} \beta_i \, (g_i^a)^2 + \sum_{i=1}^{s-1} \gamma_i \, g_i^a \, g_{i+1}^a + \epsilon^a. \, (3)$$

The BD models have s causal loci, each of which has (randomly determined) linear and quadratic effects on the phenotype, as well as mixed terms coupling one locus to another. In biological terms, this model describes a system in which each locus interacts with others in the same block (including itself), but not with loci outside the block.

The second category of models is the "promiscuous" (PS) interaction model:

$$y^a = \sum_{i=1}^{s} \alpha_i' g_i^a + \sum_{i=1}^{s'} \beta_i' \, (g_{s+i}^a)^2 + \sum_{i=1}^{s'/2} \gamma_i' \, g_i^a \, g_{s+i}^a + \epsilon^a. \, (4)$$

The model has s loci which have linear but no quadratic effect on the phenotype, and s' loci have quadratic but no linear effect on the phenotype. $s'/2$ of the latter type interact with counterparts of the former type. In biological terms, this model has subsets of loci which are entirely linear in effect, some which are entirely nonlinear, and interactions between these subsets.

We need to highlight here that there are no large systems of interacting loci currently understood in real biological systems; thus, one cannot be sure that any specific model (in particular, (3) or (4) above) is realistic. They are merely a way to generate datasets on which to test our method. What we do believe, based on many simulations, is that our method works well on models in which the generalized sparsity (roughly, total number of interactions) is sufficiently small (i.e., far from the maximal case in which all loci interact with all others). In the maximal limit the dimensionality of the parameter space of the resulting model is so large that simple considerations imply that it is intractable. We expect that Nature does not realize such models: they are implausible as they require essentially every gene to interact nontrivially with every other.

In both models, we fix var$(\epsilon) = 0.3$, so the total genetic variance accounted for (i.e., the broad sense heritability) is 0.7, which is in the realistic range for highly heritable complex traits such as height or cognitive ability (see, e.g., [25–27]). The total genetic variance can be divided into linear and nonlinear parts; the precise breakdown is determined by the specific parameter values in the model. We chose the probability distributions determining the coefficients so that typically half or somewhat less of the genetic variance is due to nonlinear effects (see Figs. 2, 3, 4, 5, 6, 7 and 8). For the BD model, α_i, β_i and γ_i are drawn
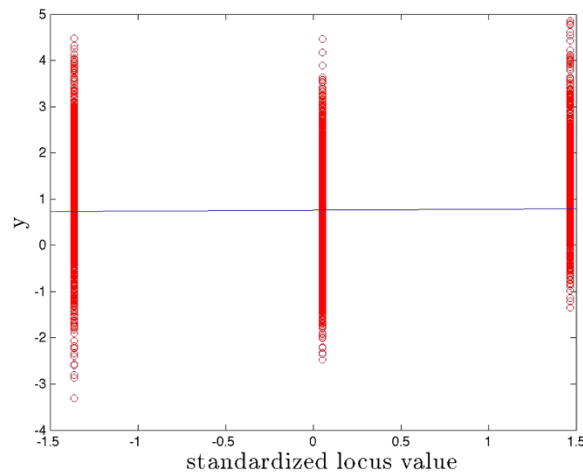
Fig. 2 Phenotype as a function of standardized locus value. The linear regression (blue line) of phenotype versus this locus value has slope close to zero. PS model with $s + s' = 5$

from normal distributions. Their means are of order unity and positive but not all the same. The standard deviation of α_i is larger than those of β_i and γ_i, but all of them are smaller than their respective means. In particular, we take their means to be $\mu(\alpha) = 1.5$, $\mu(\beta) = 1.0$ and $\mu(\gamma) = 0.5$, and their standard deviations to be $\sigma(\alpha) = 0.5$, $\sigma(\beta) = 0.2$ and $\sigma(\gamma) = 0.1$. We study the cases $s = 5, 50, 100$ with $p = 10000, 25000, 40000$ respectively. For the PS model, α'_i are drawn from $\{-1, 0, 1\}$. β'_i and γ'_i are randomly chosen from normal distributions, and are typically of order unity. (In the results shown, negative values of β'_i and γ'_i are excluded, but similar results are obtained if negative are also allowed.) We study the cases $\{s = 3, s' = 2\}, \{s = 30, s' = 20\}, \{s = 60, s' = 40\}$ with $p = 10000, 20000, 30000$ respectively.

For each of the models, we first perform Step 1 (i.e., run CS on the (y, g) data) with a value of n that is very close to p. For such a large sample size Step 1 of the algorithm always works with high precision, and determines a best-fit linear approximation (hyperplane) to the data. From this Step we can calculate the variance accounted for by linear effects, and the remaining variance which is non-linear. Parameters deduced in this manner are effectively properties of the model itself and not of the algorithm. We denote by x^* the resulting $\hat{x}$; this is an "asymptotic" linear effects vector that would result from having a very large amount of sample data.

The nonlinear variance is defined as

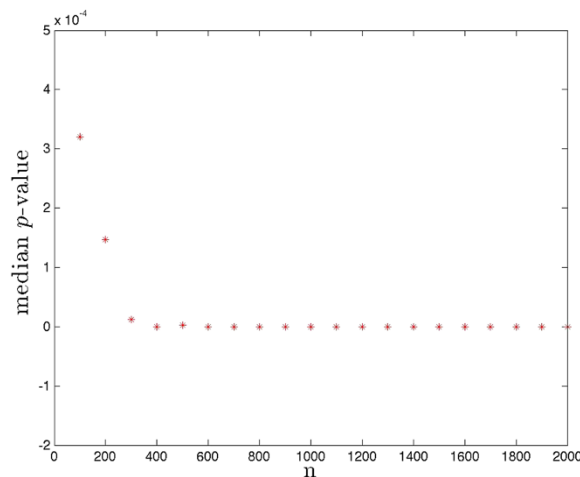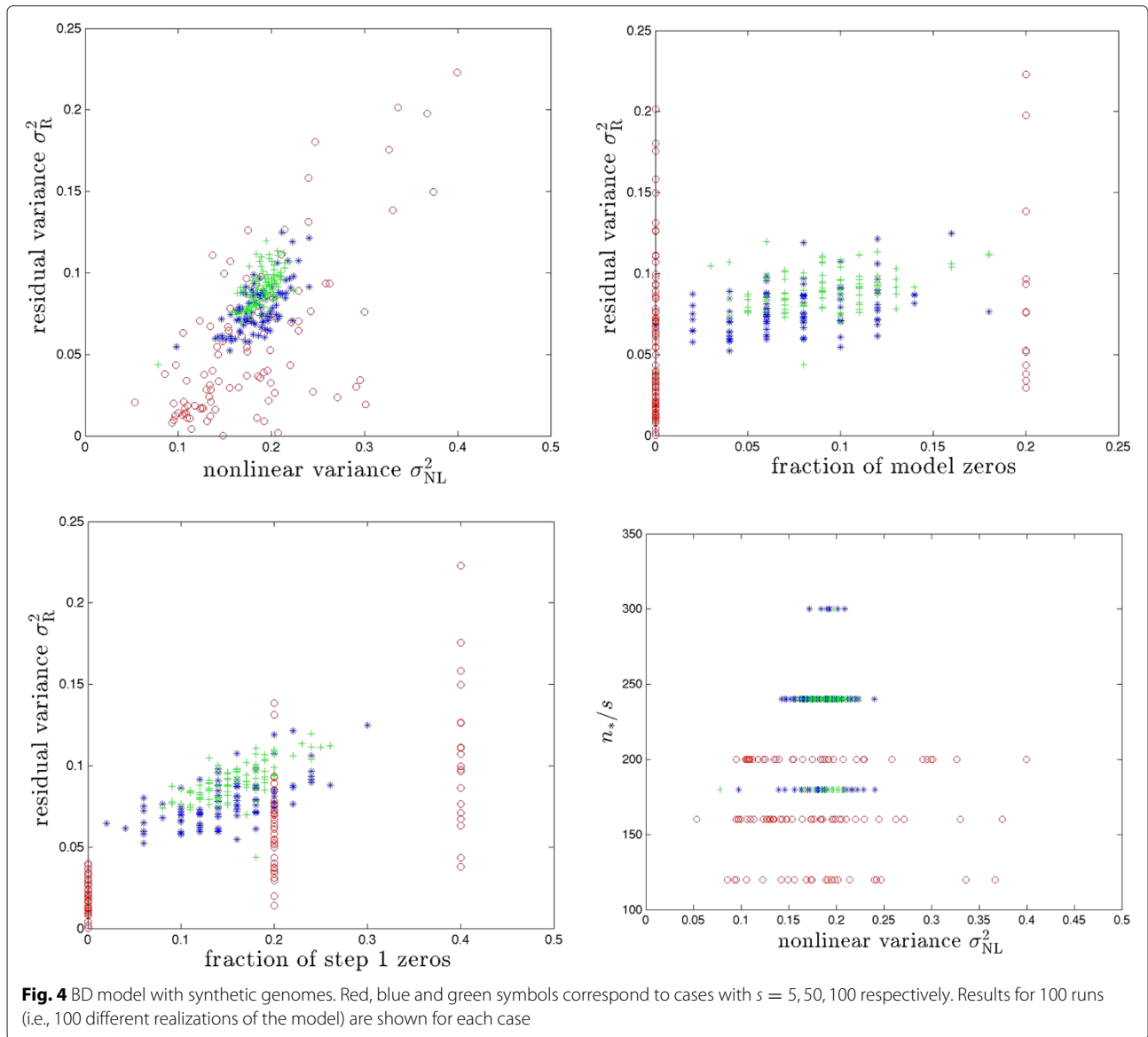$$\sigma^2_{\mathrm{NL}} \equiv \mathrm{var}(y - g\,x^* - \epsilon). \tag{5}$$

Fig. 3 The phase transition in median *p*-value as a function of sample size *n*. PS model with $s + s' = 5$
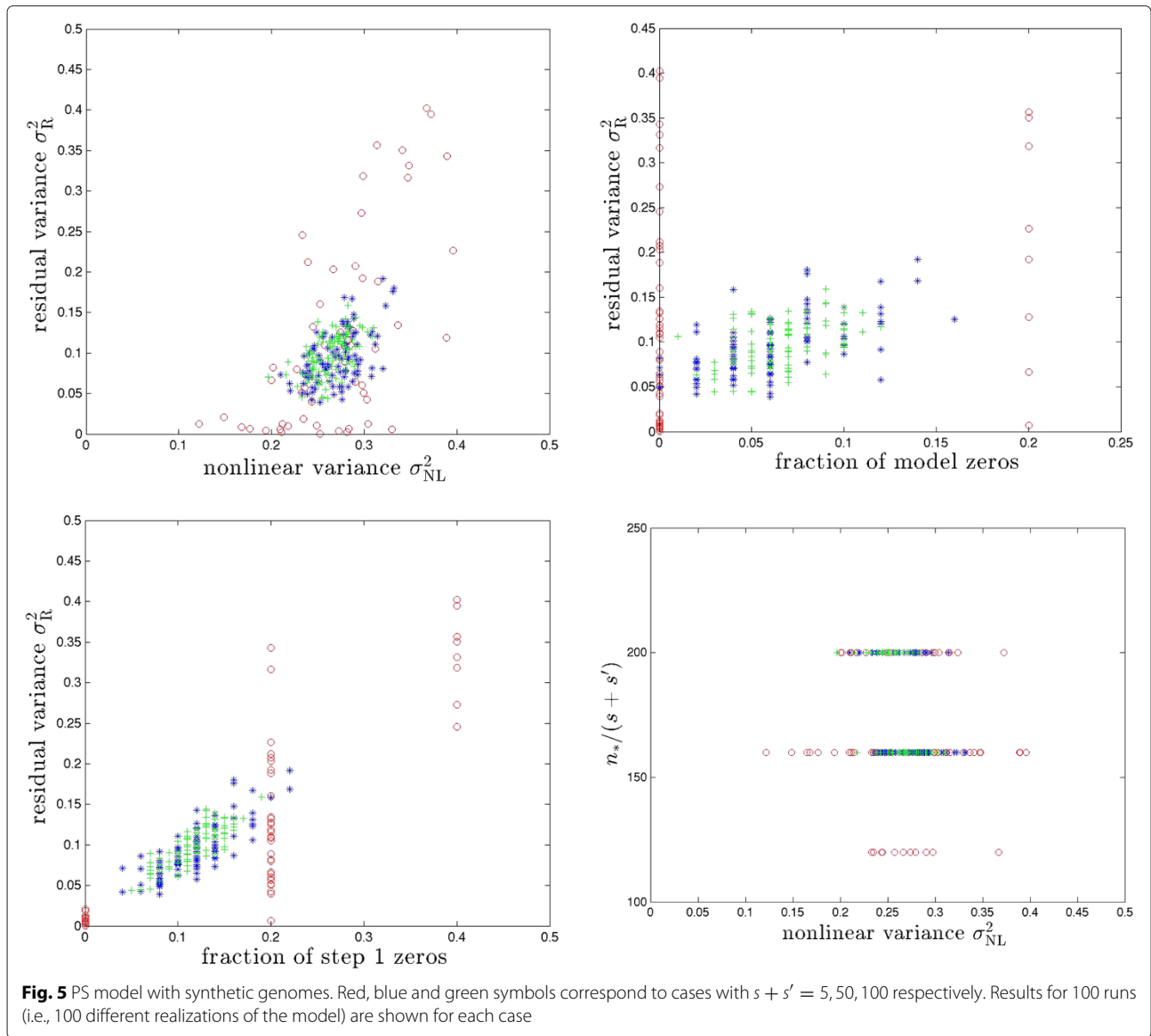
Fig. 4 BD model with synthetic genomes. Red, blue and green symbols correspond to cases with $s = 5, 50, 100$ respectively. Results for 100 runs (i.e., 100 different realizations of the model) are shown for each case

We use this quantity to estimate the penalization λ for LASSO: we set $\lambda = \sigma_{\mathrm{NL}}^2 + \mathrm{var}(\epsilon)$. In a realistic case one could set λ using an estimate of the additive heritability of the phenotype in question (e.g., obtained via twin or adoption study, or GCTA [26, 27]). This penalization may be larger than necessary; if so, the required data threshold for the phase transition might be reduced from what we observe.

From x^* we know which effects are detectable by linear CS under ideal (large sample size) conditions. In some cases, the nonlinear effects can hide a locus from detection even though at the model level (e.g., in Eqs. (3, 4)) it has a direct linear effect on the phenotype y. This happens if the best linear fit of y as a function of the locus in question has slope nearly zero (see Fig. 2). We refer to the fraction of causal loci for which this occurs as the *fraction*

of model zeros. These loci are not recoverable from either linear regression or linear CS even with large amounts of data ($n \approx p$). When this fraction is nonzero, the subspace of causal variants that is detected in Step 1 of our algorithm will differ from the actual subspace (in fact, Step 1 recovery of the causal subspace will sometimes fall short of this ideal limit, as realistic sample sizes n may be much less than p; see upper right and lower left panels in Figs. 4 and 5). This is the main cause of imperfect reconstruction of the full nonlinear model, as we discuss below.

In Step 1, we scan across increasing sample size n and compute the p-values of all genetic markers that have nonzero support (i.e., for which LASSO returns a nonzero value in $\hat{x}$) in order to detect the phase transition in CS performance. The process is terminated when the median p-value and the absolute value of its first derivative are

Fig. 5 PS model with synthetic genomes. Red, blue and green symbols correspond to cases with $s + s' = 5, 50, 100$ respectively. Results for 100 runs (i.e., 100 different realizations of the model) are shown for each case

both 10^6 times smaller than the corresponding quantities when the scanning process first starts. (The choice of 10^6 is arbitrary but worked well in our simulations – the purpose is merely to detect the region of sample size where the algorithm is working well.) This terminal sample size is defined to be n_*. The typical behavior of median p-value against n is illustrated in Fig. 3. The median p-value undergoes a phase transition, dropping to small values. n_* as defined above is typically about (2-3) times as large as the sample size at which this first occurs.
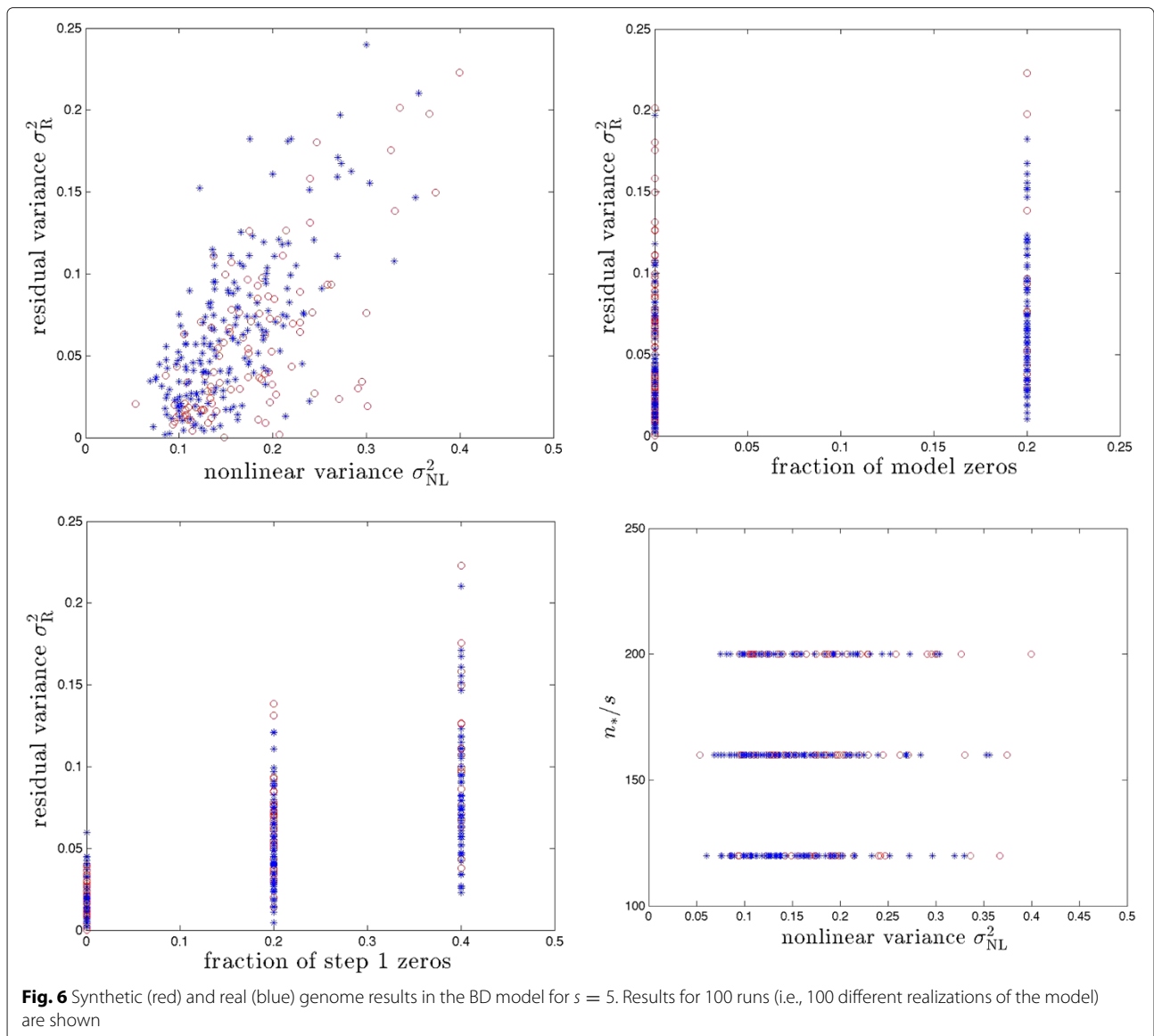
In Table 1 we display the distribution of false positives found by Step 1. That is, loci to which the algorithm assigns a nonzero effect size when in fact the original model has effect size zero. When false positives are present they cause the subspace explored in Step 2 of the

algorithm to have higher dimension than necessary. However, they do not necessarily lead to actual false positives in the final nonlinear model produced by Step 2. As is clear from Table 1, in the large majority of cases the number of false positives is small compared to the number of true positives.

For the BD and PS models, we calculate n_*/s and $n_*/(s+s')$ which are plotted against σ_{NL}^2 in Figs. 4 and 5. Next, we run Step 2 over the causal subspace determined by Step 1. Running CS gives us $\hat{X}$ (as defined in (9)) and then we compute the residual variance:

$$\sigma_{\mathrm{R}}^2 \equiv \mathrm{var}(y - G\hat{X} - \epsilon). \tag{6}$$

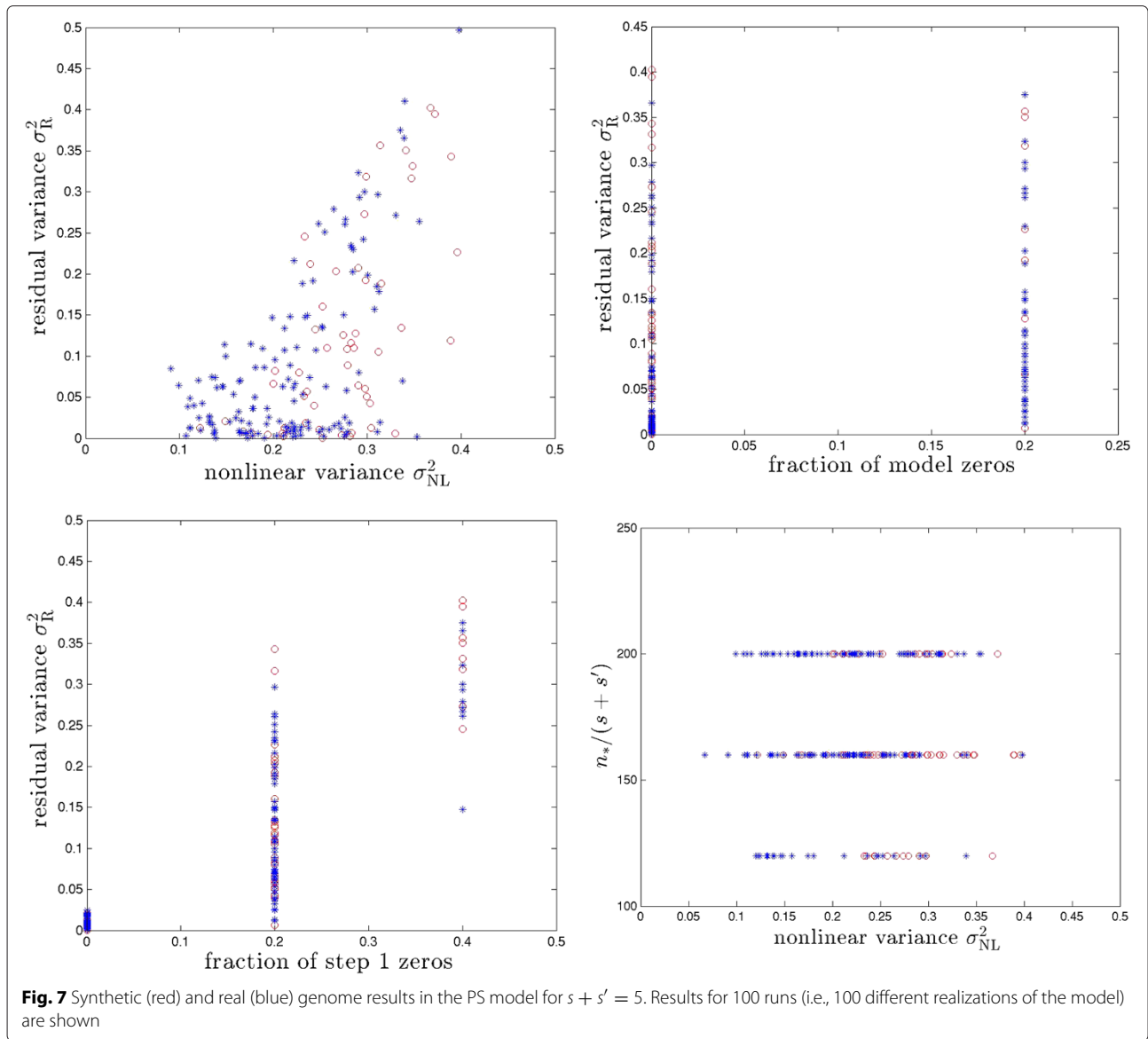(Recall that $\hat{X}$ means the candidate vector for the solution X, as was the case for $\hat{x}$ and x.)

Fig. 6 Synthetic (red) and real (blue) genome results in the BD model for $s = 5$. Results for 100 runs (i.e., 100 different realizations of the model) are shown

In Fig. 4, we display σ_R^2 against σ_{NL}^2 and fractions of zeros in both of x^* and $\hat{X}$ for the BD model. We also plot n_*/s against σ_{NL}^2. For the PS model, we display the analogous results in Fig. 5.

Results obtained using synthetic genomes are similar to those obtained from real SNP genomes, as we discuss below. For our purposes, we randomly selected p SNPs from each individual to form the rows of our sensing matrix. In Fig. 6 and 7, we compare results on synthetic and real genomes for both BD and PS models. Due to the limited sample size of ~ 1000 real genomes to which we have access, we limit ourselves to the cases $s = 5$ and $\{s = 3, s' = 2\}$. In our analysis we could detect no qualitative difference in performance between real and synthetic g. However, the correlation between columns of

real g matrices is larger than for synthetic (purely random) matrices. To compensate for this, we ran our simulations with slightly larger (e.g., 1.5 or 2 times larger) penalization λ in the case of real genomes. Our results suggest, as one might expect from theory, that the moderate correlations between SNPs found in real genomes do not alter the universality class of the compressed sensor g, indicating that our method is likely to perform similarly on real data as it does on our test data. Future analyses, however, will be required to substantiate this once the expected large number of additional real data is available.

Discussion

It is a common belief in genomics that nonlinear interactions (epistasis) in complex traits make the task of

Fig. 7 Synthetic (red) and real (blue) genome results in the PS model for $s + s' = 5$. Results for 100 runs (i.e., 100 different realizations of the model) are shown

reconstructing genetic models extremely difficult, if not impossible. In fact, it is often suggested that overcoming nonlinearity will require much larger data sets and significantly more computing power. Our results show that in broad classes of plausibly realistic models, this is not the case. Simulations using the limited amount of real data available to us supports the likelihood that this method will also work with non-simulated data. Tests on upcoming real data, however, are definitely required.

We find that the proposed method can recover a significant fraction of the predictive power (equivalently, variance) associated with nonlinear effects. The upper left panels of Figs. 4 and 5 show that we typically recover half or more of the nonlinear variance. To take a specific example, for $\sigma_{NL}^2 \sim 0.25$ over a third of the total genetic variance

$$h_{\text{broad sense}}^2 \equiv H^2 = 1 - \text{var}(\epsilon) = 0.7$$

is due to nonlinear effects. (Note total variance of y is defined to be unity.) Step 2 of our method recovers all but $\sigma_R^2 \sim 0.1$ of the total genetic variance, using the same amount of data as in the linear Step 1. The fraction of variance which is not recovered by our method is largely due to the causal variants that are not detected by Step 1 of the algorithm – i.e., the fraction of zeros. These variants would also escape detection by linear regression or essentially any other linear method using the same amount of sample data. We have also calculated false positive rates for Step 1 in our simulations and the results are shown in Table 1. Typically, the number of false positives is much smaller than the number of true positives. In other words, the dimensionality of the subspace explored in Step 2
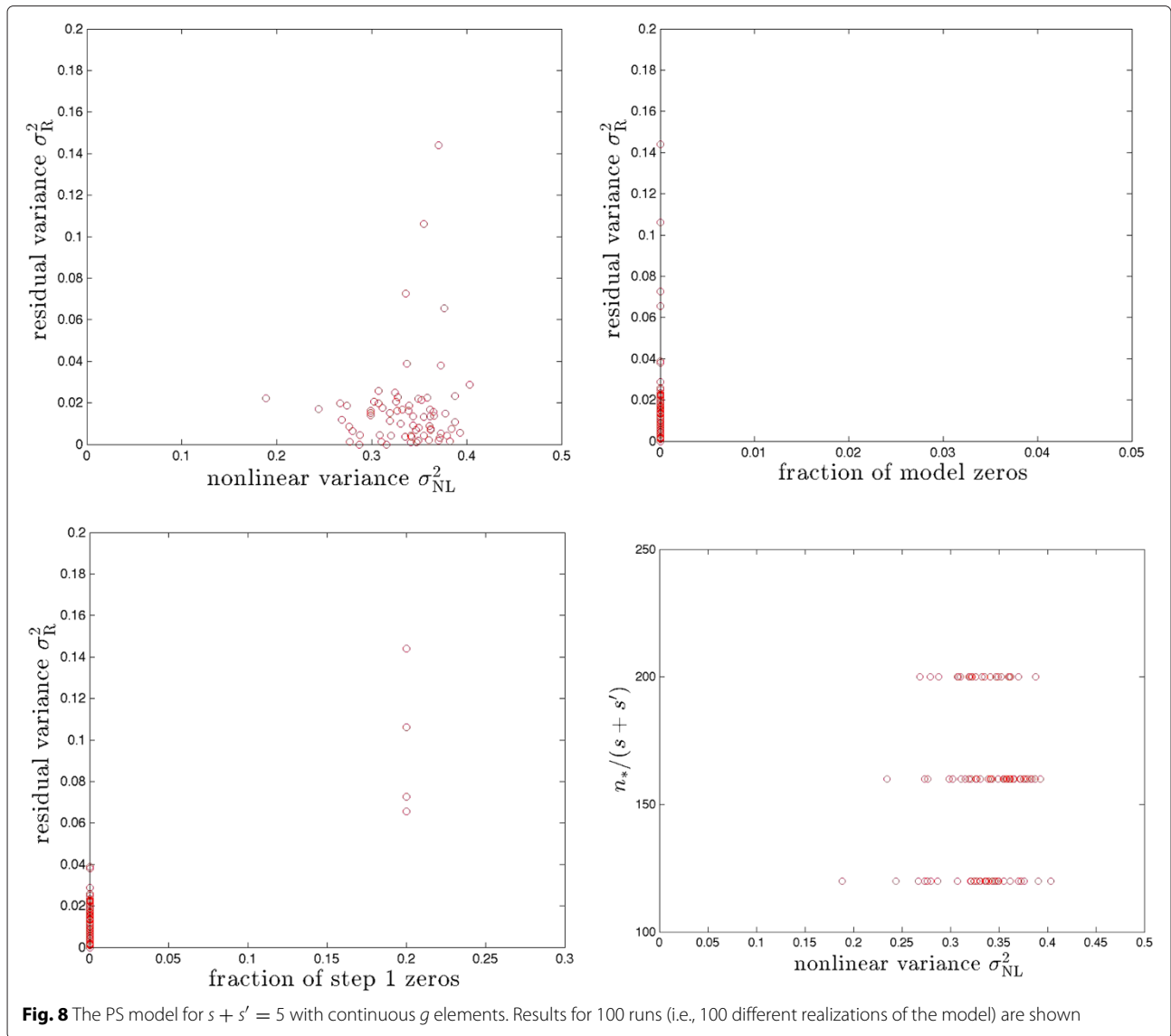
Fig. 8 The PS model for $s + s' = 5$ with continuous g elements. Results for 100 runs (i.e., 100 different realizations of the model) are shown

Table 1 Distribution of number of false positives in our simulations. These are loci which Step 1 incorrectly identifies as affecting the phenotype

	FP = 0	FP = 1	FP = 2	FP = 3	4 < FP < 7
BD, synthetic, $s = 5$	0.38	0.43	0.17	0.02	0
BD, synthetic, $s = 50$	0.39	0.37	0.19	0.03	0.02
BD, synthetic, $s = 100$	0.35	0.40	0.18	0.06	0.01
PS, synthetic, $s = 3$, $s' = 2$	0.38	0.44	0.14	0.04	0
PS, synthetic, $s = 30$, $s' = 20$	0.24	0.36	0.27	0.09	0.04
PS, synthetic, $s = 60$, $s' = 40$	0.42	0.29	0.19	0.06	0.04
BD, real data, $s = 5$	0.74	0.18	0.05	0	0.03
PS, real data, $s = 3$, $s' = 2$	0.69	0.13	0.08	0.04	0.06
Continuous, PS, synthetic, $s = 3$, $s' = 2$	0.29	0.35	0.23	0.09	0.04

The first column gives the probability that no false positives are found, the second column gives the probability that one false positive is found, etc

is nearly optimal in most cases. We are unaware of any method that can recover more of the predictive power than ours using similar sample size and computational resources.

Performance would be even better if sample sizes larger than n_* (which was somewhat arbitrarily defined) were available. Typically, $n_* \sim 100 \times$ sparsity, where sparsity s is the number of loci identified by Step 1 (i.e., dimensionality of the identified causal subspace). A lower value of n_*/s might be sufficient if we were to tune the penalization parameter λ more carefully. In a realistic setting, one can continue to improve the best fit nonlinear model as more data becomes available, eventually recovering almost all of the genetic variance.

Finally, we note that our method can be applied to problems in which the entries in g are continuous rather than discrete. (For example, compressed sensing is often applied to image reconstruction from scattered light intensities. These intensities are continuously valued and not discrete. It seems possible that the scattering medium might introduce nonlinearities, thereby making our methods of interest.) In fact, discrete values make model zeros more likely than in the continuous case. In Fig. 8, we display the results for a PS model with $\{s = 3, s' = 2\}$ and matrix entries generated from continuous probability distributions. Recovery of nonlinear variance is generally better than in the discrete case, and the fraction of zeros is smaller.

Methods

Consider the most general model which includes gene-gene interactions (we include explicit indices for clarity; $1 \leq a \leq n$ labels individuals and $1 \leq i, j \leq p$ label genomic loci)

$$y^a = \sum_i g_i^a z_i \; + \; \sum_{ij} g_i^a Z_{ij} g_j^a \; + \; \epsilon^a, \tag{7}$$

where g is an $n \times p$ dimensional matrix of genomes, z is a vector of linear effects, Z is a matrix of nonlinear interactions, and ϵ is a random error term. We could include higher order (i.e., gene-gene-gene) interactions if desired.

Suppose that we apply conventional CS to data generated from the model above. This is equivalent to finding the best-fit linear approximation

$$y^a \approx \sum_i g_i^a x_i. \tag{8}$$

If enough data (roughly speaking, $n >> s \log p$, where s is the sparsity of x) is available, the procedure will produce the best-fit hyperplane approximating the original data.

It seems plausible that the support of x, i.e., the subspace defined by nonzero components of x, will coincide with the subset of loci which have nonzero effect in *either z or Z* of the original model. That is, if the phenotype

is affected by a change in a particular locus in the original model (either through a linear effect z or through a nonlinear interaction in Z), then CS will assign a nonzero effect to that locus in the best-fit linear model (i.e., in x). As we noted in the Analysis section, this hypothesis is largely correct: the support of x tends to coincide with the support of (z, Z) except in some special cases where nonlinearity masks the role of a particular locus.

Is it possible to do better than the best-fit linear effects vector x? How hard is it to reconstruct both z and Z of the original nonlinear model? This is an interesting problem both for genomics (in which, even if the additive variance dominates, there is likely to be residual non-additive variance) and other nonlinear physical systems.

It is worth noting that although (7) is a nonlinear function of g – i.e., it allows for epistasis, gene-gene interactions, etc. – the phenotype y is nevertheless a *linear* function of the parameters z and Z. One could in fact re-express (7) as

$$y^a = \sum_i G_i^a(g) X_i + \epsilon^a \tag{9}$$

where X is a vector of effects (to be extracted) and G the most general nonlinear function of g over the s-dimensional subspace selected by the first application of CS resulting in (8). Working at, e.g., order g^2, X would have dimensionality $s(s-1)/2 + 2s$, enough to describe all possible linear and quadratic terms in (7).

Given the random nature of g, it is very likely that G will also be a well-conditioned CS matrix (we verify empirically that that this is the case). Potentially, the number of nonzero components of X could be $\sim s^k$ at order g^k. However, if the matrix Z has a sparse or block-diagonal structure (i.e., individual loci only interact with some limited number of other genes, not all s loci of nonzero effect; this seems more likely than the most general possible Z), then the sparsity of X is of order a constant k times s. Thus, extracting the full nonlinear model is only somewhat more difficult than the $Z = 0$ case. Indeed, the data threshold necessary to extract X scales as $\sim ks \log(s(s-1)/2 + 2s)$, which is less than $s \log p$ as long as $k \log(s(s-1)/2 + 2s) < \log p$.

The process for extracting X, which is equivalent to fitting the full nonlinear model in (7), is as follows:

Step 1. Run CS on (y, g) data, using linear model (8). Determine support of x: subset defined by s loci of nonzero effect.

Step 2. Compute $G(g)$ over this subspace. Run CS on $y = G(g) \cdot X$ model to extract nonzero components of X. These can be translated back into the linear and nonlinear effects of the original model (i.e., nonzero components of z and Z).

In the Analysis section we show that in many cases Steps 1 and 2 lead to very good reconstruction of the original model (7) given enough data *n*. A number of related issues are discussed:

a. When can nonlinear effects hide causal loci from linear regression (Step 1)? In cases of this sort the locus in question would not be discovered by GWAS using linear methods.
b. Both matrices *g* and *G(g)* seem to be well-conditioned CS matrices. The expected phase transitions in algorithm performance are observed for both Steps.
c. For a given partition of variance between linear (L), nonlinear (NL) and IID error ϵ, how much data $n_\star$ is required before complete selection of causal variants occurs (i.e., crossing of the phase boundary for algorithm performance)? Typically if Step 1 is successful then with the same amount of data Step 2 will also succeed.

LASSO optimization

The L1 penalization (e.g., LASSO) involves minimization of an objective function O over candidate vectors $\hat{x}$:

$$O = ||y - A\hat{x}||_{L2} + \lambda ||\hat{x}||_{L1}, \qquad (10)$$

where λ is the penalization parameter. Since O is convex, a local minimum is also a global minimum. The minimization is performed using pathwise coordinate descent [28, 29] — optimizing one parameter (coordinate), $\hat{x}_j$, at a time. This results in a modest computational complexity for the algorithm as a whole.

The solution of each sub-sequence involves a shrinkage operator $\mathcal{S}$ [6]:

$$\mathcal{S}(\hat{x}_j, \lambda) = \begin{cases} \hat{x}_j - \lambda, & \text{if } \hat{x}_j > 0 \text{ and } \hat{x}_j > \lambda; \\ \hat{x}_j + \lambda, & \text{if } \hat{x}_j < 0 \text{ and } |\hat{x}_j| > \lambda; \\ 0, & \text{if } |\hat{x}_j| \leq \lambda, \end{cases} \qquad (11)$$

where $j = 1, 2, \ldots, p$. The penalization parameter λ is estimated according to the following two part procedure. We first choose λ as the noise variance. Then after running CS, we obtain the nonlinear variance defined in Eq. (5). The ultimate λ used in the simulations is taken to be the sum of the noise variance and the nonlinear variance. In a realistic setting one could set λ using an estimate of the additive heritability of the phenotype in question (e.g., obtained via twin or adoption study, or GCTA [26, 27]). A smaller penalization might be sufficient, and allow the phase boundary to be reached with somewhat less data. We assume convergence of the algorithm if the fractional change in O between two consecutive sub-sequences is less than 10^{-4}. Note that in the case of real genomes we found that slightly increasing λ beyond the values described above yielded somewhat better results.

Availability and requirements

Project name: Source code from Determination of Non-linear Genetic Architecture using Compressed Sensing.
Project home page: https://github.com/gigascience/paper-ho2015.
Operating systems: Linux.
Programming language: MATLAB.
License: GPL v3.

Availability of supporting data

The test data and snapshots of the code supporting the results of this article are available in the GigaDB repository [30].

References
1. Hill W, Goddard M, Visscher P. Data and theory point to mainly additive genetic variance for complex traits. PLoS Genet. 2008;4(2):1000008.
2. Elad M. Sparse and redundant representations: from theory to applications in signal and image processing. New York, USA: Springer; 2010.
3. Candès E. Compressive sampling. In: Proceedings of the International Congress of Mathematicians. Madrid, Spain; 2006. p. 1433-1452.
4. Donoho DL. Compressed sensing. IEEE T Inform Theory. 2006;52:1289.
5. Foucart S, Rauhut H. A mathematical introduction to compressive sensing. Applied and Numerical Harmonic Analysis book series. New York, USA: Springer; 2013.
6. Vattikuti S, Lee J, Chang C, Hsu S, Chow C. Applying compressed sensing to genome-wide association studies. GigaScience. 2014;3:10–26.
7. McKinney B, et al. Machine learning for detecting gene-gene interactions. Appl Bioinformatics. 2006;5(1):77–88.
8. Yi N. Statistical analysis of genetic interactions. Genet Res. 2010;92(5-6): 443–59.
9. Park MY, Hastie T. Regularization path algorithms for detecting gene interactions. California, USA: Department of Statistics, Stanford University; 2006.
10. Tibshirani R. Regression shrinkage and selection via the lasso. J R Stat Soc Ser B. 1996;58:267–88.
11. Zhao P, Yu B. On model selection consistency of lasso. J Mach Learn Res. 2006;7:2541–63.
12. Meinhausen N, Yu B. Lasso-type recovery of sparse representations for high-dimensional data. Ann Stat. 2009;37(1):246–70.
13. Donoho DL, Tanner J. Observed universality of phase transitions in high-dimensional geometry, with implications for modern data analysis and signal processing. Phil Trans R Soc. 2009;367:4273–293.

14. Donoho DL. High-dimensional centrally symmetric polytopes with neighborliness proportional to dimension. Discrete Comput Geom. 2006;35(4):617–52.
15. Donoho DL, Tanner J. Neighborliness of randomly projected simplices in high dimensions. Proc Natl Acad Sci USA. 2005;102(27):9452–7.
16. Donoho DL, Tanner J. Sparse nonnegative solutions of underdetermined linear equations by linear programming. Proc Natl Acad Sci. 2005;102(27): 9446–51.
17. Donoho DL, Tanner J. Counting faces of randomly-projected polytopes when the projection radically lowers dimension. J Am Math Soc. 2009;22(1):1–53.
18. Manichaikul A, et al. A model selection approach for the identification of quantitative trait loci in experimental crosses, allowing epistasis. Genetics. 2009;181:1077–86.
19. Lee S, Xing E. Leveraging input and output structures for joint mapping of epistatic and marginal eqtls. Bioinformatics. 2012;28:137–46.
20. Zhang X, Huang S, Zou F, Wang W. Team: Efficient two-locus epistasis tests in human genome-wide association study. Bioinformatics. 2010;26: 217–27.
21. Wan X, et al. Boost: A fast approach to detecting gene-gene interactions in genome-wide case-control studies. Am J Hum Genet. 2010;87:325–40.
22. Devlin B, et al. Analysis of multilocus models of association. Genet Epidemiol. 2003;25:36–47.
23. Wu J, et al. Screen and clean: a tool for identifying interactions in genome-wide association studies. Genet Epidemiol. 2010;34(3):275–85.
24. Wang Y, et al. An empirical comparison of several recent epistatic interaction detection methods. Bioinformatics. 2011;27:2936–43.
25. Hsu S. On the genetic architecture of intelligence and other quantitative traits. arXiv:1408.3421. (Figures 6 and 7 display heritability from twins studies).
26. Yang J, Lee S, Goddard M, Visscher P. Gcta: A tool for genome-wide complex trait analysis. Am J Hum Genet. 2011;88(1):76–82.
27. Yang J, et al. Common snps explain a large proportion of the heritability for human height. Nat Genet. 2010;42(7):565–9.
28. Friedman J, Hastie T, Höfling H, Tibshirani R. Pathwise coordinate optimization. Ann Appl Stat. 2007;1:302–32.
29. Friedman J, Hastie T, Höfling H, Tibshirani R. Regularization paths for generalized linear models via coordinate descent. J Stat Softw. 2010;33: 1–22.
30. Ho CM, Hsu SDH. Supporting materials for determination of nonlinear genetic architecture using compressed sensing. GigaScience Database: http://dx.doi.org/10.5524/100162.