Reconstruction of Networks with Direct and Indirect Genetic Effects

Kruijer, Willem; Behrouzi, Pariya; Bustos-Korts, Daniela; Rodríguez-Álvarez, María Xosé; Mahmoudi, Seyed Mahdi; Yandell, Brian; Wit, Ernst; van Eeuwijk, Fred A

Published in:
Genetics

DOI:
10.1534/genetics.119.302949

Publication date:
2020

January 4, 2019

Reconstruction of networks with direct and indirect genetic effects

Willem Kruijer[1,*], Pariya Behrouzi[1], Daniela Bustos Korts[1], Maria Xose Rodriguez-Alvarez[2,3], Seyed Mahdi Mahmoudi[4,†], Brian Yandell[5], Ernst Wit[4,††], Fred van Eeuwijk[1]

1 Biometris, Wageningen University and Research (Wageningen, Netherlands)

2 Basque Center for Applied Mathematics (Bilbao, Spain)

3 IKERBASQUE, Basque Foundation for Science (Bilbao, Spain)

4 Johann Bernoulli Institute, University of Groningen (Groningen, Netherlands)

5 University of Wisconsin-Madison (Madison, Wisconsin 53706-1510, USA)

∗ Corresponding author (E-mail: willem.kruijer@wur.nl)

† Current address: Faculty of Mathematics, Statistics and Computer Science, Semnan University (Semnan, Iran)

†† Current address: University of Lugano (Lugano, Switzerland)

Abstract: Genetic effects and functional relationships between traits can be represented by directed graphs, as proposed by Wright in 1921. Nowadays such graphs are often estimated from empirical data, typically by applying causal inference methods to multi-trait observations and a small number of QTLs. When however individual QTLs explain little genetic variance, much of the genetic signal will be missed. To overcome this limitation, Gianola and Sorensen (2004) defined structural equation models with random genetic effects. Current causal inference methods for these models treat the genetic effects as nuisance terms, that need to be eliminated by taking residuals from an unstructured multi-trait mixed model (MTM). Fitting such MTM for large numbers of traits is however computationally and statistically challenging.

Here we propose an alternative strategy, where genetic effects are formally included in the graph. Using theoretical results, simulations and real data we show that this has several advantages: (1) the extended graph satisfies the global Markov property (2) genetic effects can be directly incorporated in algorithms like PC, allowing for many more traits (3) we can distinguish direct and indirect effects, and use more of the causal information contained in the data. For example, we can, under certain assumptions, recover the structure $G \rightarrow Y_1 \rightarrow Y_2$ if the genetic variance of Y_2 given Y_1 is found to be zero. Finally, we show that this can be achieved with much higher power if individual plant or plot data are used, extending the results of Kruijer et al (2015) for single trait analyses. We have implemented the method in the R-package pcgen, publicly available from CRAN.

1 Introduction

Structural equation models (SEM) have been proposed almost a century ago by Wright ([1]), and have been frequently used to describe causal relationships between phenotypes (for a review see [2]). SEM are called structural or functional because each variable is explicitly modeled as a function of the other variables and a noise term, and have a causal interpretation. The advantages compared to regression models are that (i) one can predict the behavior of the system when one or more of the structural equations are modified by some kind of intervention (ii) one can distinguish direct and indirect effects of one variable on another.

In some cases a specific structural model may be formulated using prior biological knowledge (see e.g. [3]), but there is an increasing number of applications (especially in system genetics) were no causal model can be specified in advance. In such cases causal inference methods ([4], [5], [6]) can be used to propose models. These methods are not a substitute for randomized experiments, but rather propose causal models that are most compatible with the observed data, which can be highly useful when having to prioritize future experiments. In genomics for example, the effect of gene-knockouts in yeast was better predicted with causal inference methods than using penalized regression [7]. Statistically, this is because of the manipulation [5] or truncated factorization theorem [4] which describes the distribution after an intervention on one or several variables, which regression methods cannot do.

Causal inference methods have also been applied to genetic data, where observations come from different genotypes. An important question is then how genotypic differences should be accounted for in the model. A popular strategy is to perform causal inference on the traits and all available markers, or QTLs found by mapping ([8], [9], [10]). When however part of the genetic variance is not explained by QTLs, genetic differences may have little added value to the reconstruction of the network. Moreover, typical model assumptions such as independence of residual errors may be violated.

An important class of models that may overcome this limitation was introduced in [11], who defined structural equation models containing random genetic effects. Causal inference for these models is however challenging, due to the correlations of the genetic effects across traits and individuals. To deal with these correlations, [12] and [13] proposed to perform causal inference after subtracting genomic predictions obtained from a multi-trait model (MTM). Similarly, [14] applied the PC-algorithm to the residuals of multi-SNP models. The difficulty with these approaches is that the MTM is limited to small numbers of traits, and that the existence of direct genetic effects cannot be tested. For example, if $Y_1 \rightarrow Y_2 \leftarrow Y_3$, with direct genetic effects on Y_1 and Y_3, the absence of direct genetic effects on Y_2 cannot be inferred from MTM residuals. Inspired by these problems we define a class of causal graphs in which direct genetic effects are part of the graph, and a single node G represents all direct genetic effects. For each trait Y_j an arrow $G \rightarrow Y_j$ is present if and only if the direct genetic effect G_j is nonzero, i.e. has positive variance.

See Figure 2 below for an example.

Although this idea is not new, ([15], [16], [17]), this work is, to the best of our knowledge, the first that formalizes it. In particular, we show that the Markov property holds for the graph extended with genetic effects (Theorem 1), and based on this develop the pcgen algorithm. pcgen stands for PC with genetic effects, and is an adaptation of the general PC-algorithm [5] (named after its inventors Peter Spirtes and Clark Glymour). Briefly, pcgen assesses the existence of a direct genetic effect on a given trait by testing whether its genetic variance is zero, conditional on various sets of other traits. For the existence of an edge between traits Y_1 and Y_2 we test whether in a bivariate MTM the residual covariance between Y_1 and Y_2 is zero, again conditional on sets of other traits. Alternatively, this test may be based on MTM residuals (as in existing approaches, who however did not test the existence of direct genetic effects). Under the usual assumptions of independent errors, recursiveness and faithfulness, we show that pcgen can recover the underlying partially directed graph (Theorem 2); this result holds for general genetic relatedness, and regardless of the correlations between the direct genetic effects.

Successful network reconstruction with pcgen requires sufficient power, in the test for direct genetic effects ($G \to Y_j$) as well as for the trait to trait relations ($Y_j \to Y_k$). Given a plant or other immortal population with observations on genetically identical replicates, this power is likely to be highest when the original observations are used, instead of genotypic means and a marker based genetic relatedness matrix (see [18], for the estimation of genetic variance). Because of this, we focus on experiments with replicates, although the pcgen algorithm and most of our results are generally applicable, to any species and relatedness matrix. Moreover, the use of the models developed here is not limited to pcgen, and in the discussion we will provide directions for further applications. Because fitting the MTM for all traits simultaneously is no longer necessary, pcgen can handle a considerably larger number of traits.

Overview of the paper

... Appendix D contains an overview of the notation.

2 Materials and methods

2.1 Genetic Structural Equation Models

2.1.1 Structural Equation Models

To introduce structural models, let us first consider a simple linear structural equation model (SEM) without genetic effects:

$$\mathbf{y_i} = \mathbf{x_i}B + \mathbf{y_i}\Lambda + \mathbf{e_i}, \tag{1}$$

where $\mathbf{y_i}$ is a $1 \times p$ vector of phenotypic values for p traits measured on the ith individual, $\mathbf{e_i}$ a vector of random errors, and Λ is a $p \times p$ matrix of structural coefficients. The $q \times p$

matrix $B = [\beta^{(1)} \cdots \beta^{(p)}]$ contains intercepts and trait specific fixed effects of (exogenous) covariates, whose values are contained in the $1 \times q$ vector $\mathbf{x_i}$.

Defining $n \times p$ matrices $\mathbf{Y} = [\mathbf{Y_1} \cdots \mathbf{Y_p}]$ with rows $\mathbf{y_i}$ and $\mathbf{E} = [\mathbf{E_1} \cdots \mathbf{E_p}]$ with rows $\mathbf{e_i}$, and a $n \times q$ design matrix $\mathbf{X}$, we can write

$$\mathbf{Y} = \mathbf{X}B + \mathbf{Y}\Lambda + \mathbf{E}. \tag{2}$$

Λ has zeros on the diagonal and defines a directed graph $\mathcal{G}_\mathcal{Y}$ over the traits $Y_1, \ldots, Y_p$, containing the edge $Y_j \to Y_k$ if an only if the $(j,k)th$ entry of Λ is nonzero[1]. The columns in (2) correspond to p linear *structural equations*, one for each trait. These are determined by the *path coefficients*, the nonzero elements in Λ. For example, in Figure 1, if $\mathbf{X} = 1_n$ is the $n \times 1$ vector of ones and $B = [\mu_1\ \mu_2\ \mu_3]$, the third trait has values $\mathbf{Y_3} = \mu_3 1_n + \lambda_{13}\mathbf{Y_1} + 2\lambda_{23}\mathbf{Y_2} + \mathbf{E_3}$. The equality sign here can be understood as an assignment, i.e. $\mathbf{Y_3}$ is determined by the values of $\mathbf{Y_1}$ and $\mathbf{Y_2}$ (its *parents* in the graph $\mathcal{G}_\mathcal{Y}$) and an error. If the directed graph does not contain any cycle (i.e. a directed path from a trait to itself), it is a *DAG* (directed acyclic graph), and the SEM is said to be *recursive*. In the notation we will distinguish the nodes $Y_1, \ldots, Y_p$ in the graph $\mathcal{G}_\mathcal{Y}$ (normal type), and the random vectors $\mathbf{Y_1}, \ldots, \mathbf{Y_p}$ that these nodes represent (bold face).

Assuming that the error vectors $\mathbf{e_i}$ are independent and follow a multivariate normal $N(0, \Sigma_E)$ distribution, it follows from (1) that

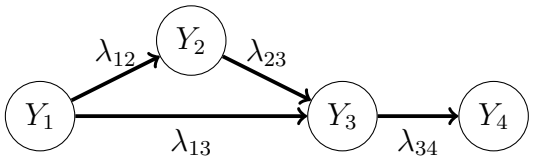
$$\begin{aligned} \mathbf{y_i} = \mathbf{x_i}B(I-\Lambda)^{-1} + \mathbf{e_i}(I-\Lambda)^{-1} &= \mathbf{x_i}B\Gamma + \mathbf{e_i}\Gamma \\ &\sim N(\mathbf{x_i}B\Gamma, \Gamma^t\Sigma_E\Gamma) = N(\mathbf{x_i}B\Gamma, \Sigma), \end{aligned} \tag{3}$$

i.e., the covariance of the $\mathbf{y_i}$'s is determined by Σ_E and $\Gamma = (I-\Lambda)^{-1}$. [2] Given sufficiently strong assumptions on Λ and Σ_E, their non-zero elements are identifiable. In Figure 1 for instance, assuming Σ_E is diagonal, it is possible to estimate the 8 parameters (λ_{12}, λ_{13}, λ_{23} and λ_{34}, and the variances of the 4 error variables) based on the sample covariance matrix $\hat{\Sigma}$ (which has 4 diagonal and 6 unique off-diagonal elements). For a more general discussion, see [19], [20] and [21].

An important property of SEM is that the effects of *interventions* can be predicted, which are changes in one or more of the structural equations. For example, suppose that in Figure 1, Y_1, Y_2 and Y_3 are the expression levels of 3 genes, and Y_4 is plant height. Then after forcing Y_2 to be zero (e.g. by knocking out the gene), the total effect of Y_1 on Y_4 changes from $(\lambda_{13}\lambda_{34} + \lambda_{12}\lambda_{23}\lambda_{34})$ to $\lambda_{13}\lambda_{34}$. More generally, the new joint distribution of $\mathbf{Y_1}, \ldots, \mathbf{Y_p}$ after intervention can be obtained from the manipulation or truncated factorization theorem ([4]), *without* needing observations from the new distribution. [16] discussed the consequences of interventions for genomic selection.

[1] We put this path coefficient in the (j,k)th entry (instead of the (k,j)th), and in (2) we post-multiply with Λ. This has the advantage that the data-matrix $\mathbf{Y}$ has the usual dimensions $n \times p$, and that the covariance of $vec(\mathbf{Y})$ has terms of the form $(\Gamma^t\Sigma_G\Gamma) \otimes K$, instead of $K \otimes (\Gamma^t\Sigma_G\Gamma)$; see e.g. equation (8)

[2] since we *post*-multiply $\mathbf{e_i}$ with $\Gamma = (I-\Lambda)^{-1}$, the covariance is $\Gamma^t\Sigma_E\Gamma$ and not $\Gamma\Sigma_E\Gamma^t$



$$\Lambda = \begin{pmatrix} 0 & \lambda_{12} & \lambda_{13} & 0 \\ 0 & 0 & \lambda_{23} & 0 \\ 0 & 0 & 0 & \lambda_{34} \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Figure 1. An example of a SEM.

2.1.2 GSEM: Structural Equation Models with genetic effects

[11] extended model (1) with random genetic effects $\mathbf{g_i}$: for individuals $i = 1, \ldots, n$, it is then assumed that

$$\mathbf{y_i} = \mathbf{x_i}B + \mathbf{y_i}\Lambda + \mathbf{g_i} + \mathbf{e_i}, \tag{4}$$

where again $\mathbf{y_i}$ is a $1 \times p$ vector of phenotypes, Λ contains the structural coefficients, and $\mathbf{e_i} \sim N(0, \Sigma_E)$ are vectors of random errors. We will refer to model (4) as a linear GSEM (genetic structural equation model), or simply GSEM. While the genetic effects introduce relatedness between individuals, there is no form of social interaction (as in e.g. [22], [23]).

The $1 \times p$ vectors $\mathbf{g_i}$ contain the direct genetic effects for individuals $i = 1, \ldots, n$. Each ${\mathbf{g_i}}^t$ follows a $N(0, \Sigma_G)$ distribution, where Σ_G is a $p \times p$ matrix of genetic variances and covariances. The vectors $\mathbf{g_i}$ are independent of the $\mathbf{e_i}$'s, but not independent among themselves. Defining a $n \times p$ matrix $\mathbf{G} = [\mathbf{G_1} \cdots \mathbf{G_p}]$ with rows $\mathbf{g_i}$ and columns $\mathbf{G_j}$ $(j = 1, \ldots, p)$, we can extend (2) as follows:

$$\mathbf{Y} = [\mathbf{Y_1} \cdots \mathbf{Y_p}] = \mathbf{X}B + \mathbf{Y}\Lambda + \mathbf{G} + \mathbf{E}. \tag{5}$$

Each vector $\mathbf{G_j}$ is the vector of direct genetic effects on the jth trait. We make the following assumptions about the GSEM defined in (5):

1. *all traits are measured on each individual:* the rows $\mathbf{y_i}$ of $\mathbf{Y}$ may be either observations on individual organisms or genotypic means of a number of replicates (plants, or plots in a field trial), but the original observations should always come from the same experiment. In addition, the residual errors originate from biological variation.

2. *recursiveness:* the graph $\mathcal{G}_\mathcal{Y}$ defined by Λ is a DAG.

3. *causal sufficiency:* the covariance matrix Σ_E of the error vectors $\mathbf{e_i}$ is diagonal, i.e. there are no latent variables. This means that all nonzero (non-genetic) correlations between traits must be the consequence of causal relations between the traits. In the discussion we describe how this assumption may be relaxed. We also assume the diagonal elements of Σ_E to be strictly positive.

4. *Genetic relatedness among individuals:* $\mathbf{G}$ is independent from $\mathbf{E}$, and has a matrix-variate normal distribution with row-covariance K and column covariance Σ_G, where

K is a $n \times n$ relatedness matrix, which we describe in more detail in section 2.1.5. Equivalent with this, the $np \times 1$ vector $vec(\mathbf{G}) = (\mathbf{G_1}^t, \ldots, \mathbf{G_p}^t)^t$ is multivariate normal with covariance $\Sigma_G \otimes K$, where vec denotes the operation of creating a column vector by stacking the columns of a matrix. Consequently, each $\mathbf{G_j}$ is multivariate normal with covariance $\sigma^2_{G,j}K$, where the variances $\sigma^2_{G,j}$ form the diagonal of Σ_G. Using the same notation, we can write that $\mathbf{E}$ is matrix-variate normal with row-covariance I_n and column covariance Σ_E, and that $vec(\mathbf{E}) \sim N(0, \Sigma_E \otimes I_n)$.

5. *No collinear genetic effects:* the diagonal elements of Σ_G do not need to be strictly positive, but for all nonzero elements, the corresponding correlation is not 1 or -1.

Assumptions 1-4 were also made in related work on structural models with random genetic effects ([12], [13]), and 1-3 are commonly made for structural models without such effects. Assumption 1 is implicit in model (5) itself, as it is assumed that the structural equations propagate all errors to traits further down in the graph. Network reconstruction without this assumption would rely completely on the genetic effects, requiring Σ_G to be diagonal, which is a rather unrealistic assumption (see the discussion, section 4.1). Assumption 1 does not require traits to be measured at the same time. In particular, it is possible to include the same trait measured at different timepoints, which of course puts contraints on the causality. Such contraints can in principle be incorporated in our model, just as other biological contraints (see e.g. [24]), although we will not explore this here.

2.1.3 Graphical representation of GSEM: extending $\mathcal{G}_\mathcal{Y}$ with genetic effects

Contrary to previous work, we will explicitly take into account the possibility that there are no direct genetic effects on some of the traits. In this case, the corresponding rows and columns in Σ_G are zero. We let $D \subseteq \{1, \ldots, p\}$ denote the index set of the traits with direct genetic effects, and write $\Sigma_G[D, D]$ for the sub-matrix with rows and columns restricted to D. From assumption 3 above it follows that $\Sigma_G[D, D]$ is non-singular, i.e. there can be no perfect correlations between direct genetic effects.

We graphically represent model (5) by a graph $\mathcal{G}$ with nodes $Y_1, \ldots, Y_p$ and a node G, which represent respectively $\mathbf{Y_1}, \ldots, \mathbf{Y_p}$ and the matrix $\mathbf{G} = [\mathbf{G_1} \cdots \mathbf{G_p}]$. $\mathcal{G}$ contains an edge $Y_j \to Y_k$ if the (j,k)th entry of Λ is nonzero, and an edge $G \to Y_j$ if $\mathbf{G_j}$ is nonzero with probability one, i.e., if $\sigma^2_{G,j} > 0$. See Figure 2 for an example. In words, $\mathcal{G}$ is defined as the original graph $\mathcal{G}_\mathcal{Y}$ over the traits (defined by Λ), extended with arrows $G \to Y_j$ for all traits with a direct genetic effect, i.e. for all $j \in D$. Consequently, our main objective of reconstructing trait-to-trait relationships and direct genetic effects translates as reconstructing $\mathcal{G}$.

As for the $\mathbf{Y_j}$'s, we distinguish the node in the graph G (normal type) and the random matrix $\mathbf{G}$ it represents (bold face). $\mathbf{G}$ is represented by a *single* node G, instead of multiple nodes $G_1, \ldots, G_p$. This choice is related to our assumption that K is the same for all traits; see Appendix G.1 for a motivating example. The orientation of any edge

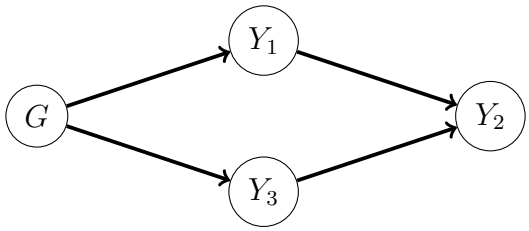


Figure 2. An example of a graph $\mathcal{G}$ representing a genetic structural equation model (GSEM). There is no direct genetic effect on Y_2, and therefore no edge $G \to Y_2$.

between G and Y_j is restricted to $G \to Y_j$, because G arises from a randomized treatment, and also because the opposite orientation would be biologically nonsensical. Because of our assumption that $\mathcal{G}_\mathcal{Y}$ is a DAG, it follows that $\mathcal{G}$ is a DAG as well, as a cycle would require at least one edge pointing into G.

We emphasize that $\mathcal{G}$ is just a mathematical object and not a complete visualization of all model terms and their distribution, as is common in the SEM-literature. In particular, $\mathcal{G}$ does not contain nodes for the residual errors, path coefficients, or information about the off-diagonal elements of Σ_G. While Σ_G is usually not entirely identifiable ([11]), we will see that $\mathcal{G}$ is identifiable in terms of its skeleton (the undirected graph obtained when removing the arrowheads) and some of the orientations.

2.1.4 Direct and indirect genetic effects

Taking the term $\mathbf{y_i}\Lambda$ from the right- to the left-hand side in equation (4), and assuming that the inverse $\Gamma = (I - \Lambda)^{-1}$ exists, it follows that

$$\mathbf{y_i} = \mathbf{x_i}B\Gamma + \mathbf{g_i}\Gamma + \mathbf{e_i}\Gamma = \mathbf{x_i}B\Gamma + \mathbf{u_i} + \mathbf{e_i}\Gamma \sim N(\mathbf{x_i}(B\Gamma), \Gamma^t\Sigma_G\Gamma + \Gamma^t\Sigma_E\Gamma), \qquad (6)$$

where $\mathbf{u_i} = \mathbf{g_i}\Gamma$ is the *total* genetic effect. Hence, as pointed out by [16], the genetic variance of a trait is not only driven by its direct genetic effect ($\mathbf{g_i}$), but also by direct genetic effects on traits affecting it, i.e. its parents in the graph $\mathcal{G}_\mathcal{Y}$. The *indirect* genetic effect is the difference $\mathbf{u_i} - \mathbf{g_i}$.

Similarly, we can distinguish the contribution of direct and indirect genetic effects to the genetic covariance. The (j, k)th element of $\Gamma^t\Sigma_G\Gamma$ in (6) is the *total* genetic covariance between $\mathbf{Y_j}$ and $\mathbf{Y_k}$. This is what is usually meant with genetic covariance. When necessary, we distinguish this from the covariance between the direct genetic effects $\mathbf{G_j}$ and $\mathbf{G_k}$, determined by $\Sigma_G[j, k]$. Indeed $\Sigma_G[j, k]$ affects the total genetic covariance, but the latter is also driven by causal relationships between traits, as defined by $\Gamma = (I-\Lambda)^{-1}$. Regarding the diagonal of $\Gamma^t\Sigma_G\Gamma$, we note that traits without a direct genetic effect may still have positive genetic variance.

2.1.5 Genetic relatedness

The genetic relatedness matrix K introduced in assumption 4 determines the covariance between the rows of $\mathbf{G}$. We will assume that K is one of the following types:

- $K = ZZ^t$, Z being the $n \times m$ incidence matrix assigning $n = mr$ plants (or plots) to m genotypes, in a balanced design with r replicates for each genotype. This K is obtained when each genotype has an independent effect, as in the classical estimation of broad-sense heritability. Since no marker-information is included, the model cannot be used for genomic prediction, but we will see that for the reconstruction of $\mathcal{G}$ (using the training genotypes) it has considerable computational and statistical advantages.

- K is estimated from a dense set of markers, assuming additive infinitesimal effects. To keep the notation simple we will still use $\sigma^2_{G,j}$ for the diagonal elements of Σ_G, instead of $\sigma^2_{A,j}$. This type of relatedness matrix is used when there is only a single individual per genotype, or when only genotypic means are available.

In both cases K has dimension $n \times n$. The balance required in the first case is necessary in Theorems 5 and 6 below, but is not a general requirement for our models, or for the pcgen algorithm.

2.1.6 The joint distribution implied by the GSEM

The sum $\mathbf{G} + \mathbf{E}$ does in general not have a matrix-variate normal distribution, but from our assumption 4 it still follows that $vec(\mathbf{G} + \mathbf{E})$ is multivariate normal with covariance $\Sigma_G \otimes K + \Sigma_E \otimes I_n$. We can therefore rewrite equation (5) as

$$\mathbf{Y} = \mathbf{X}B\Gamma + \mathbf{G}\Gamma + \mathbf{E}\Gamma = \mathbf{X}B\Gamma + \mathbf{U} + \mathbf{E}\Gamma, \tag{7}$$

and equation (6) generalizes to

$$vec(\mathbf{Y}) \sim N\left(vec(\mathbf{X}(B\Gamma)), (\Gamma^t\Sigma_G\Gamma) \otimes K + (\Gamma^t\Sigma_E\Gamma) \otimes I_n\right). \tag{8}$$

As pointed out in [11], [12] and [13], (8) can be re-written as

$$vec(\mathbf{Y}) \sim N\left(vec(\mathbf{X}\tilde{B}), V_G \otimes K + V_E \otimes I_n\right), \tag{9}$$

where $V_G = \Gamma^t\Sigma_G\Gamma$ and $V_E = \Gamma^t\Sigma_E\Gamma$, and $\tilde{B}$ is the matrix of fixed effects transformed by Γ. This is a common model for multi-trait GWAS and genomic prediction (see among others [15], [25], [26]).

Using the results of [5] (p. 371), it turns out that Γ can be written directly in terms of sums of products of path coefficients (see Appendix E.2). Consequently, there is no need

to invert $(I - \Lambda)$, although it still holds that $\Gamma = (I - \Lambda)^{-1}$, provided the inverse exists. Defining γ_j as the jth column of Γ, we can express the jth trait as

$$\mathbf{Y_j} = \mathbf{X}B\gamma_j + \mathbf{G}\gamma_j + \mathbf{E}\gamma_j = \mathbf{X}B\gamma_j + \mathbf{U_j} + \mathbf{E}\gamma_j, \tag{10}$$

i.e. equation (7), restricted to the jth column.

2.1.7 Causal inference without genetic effects

So far we have assumed that $\mathcal{G}$ in known: given sufficient restrictions on Λ, Σ_G and Σ_E, it may then be possible to estimate these matrices ([11]). In this work however, we aim to reconstruct an unknown $\mathcal{G}$, based on observations from a GSEM of the form (5). We will do this with the pcgen algorithm introduced in section 2.2, but first briefly review the necessary concepts. Appendix E.1 contains a more detailed introduction.

Suppose for the moment we have observations generated by an acyclic SEM without latent variables, and without genetic effects. From the pioneering work of Judea Pearl and others in the 1980s it is known that we can recover the skeleton of the DAG and some of the orientations, i.e. those given by the v-structures. A v-structure is any triple of nodes Y_j, Y_k, Y_l such that $Y_j \to Y_k \leftarrow Y_l$, without any edge between Y_j and Y_l. All DAGs with the same skeleton and v-structures form an *equivalence class*, which can be represented by a completed partially directed acyclic graph (CPDAG). DAGs from the same equivalence class cannot be distinguished from observational data, at least not under the assumptions we make here. For reconstruction of the CPDAG, constraint-based and score-based methods have been developed (for an overview, see [24]).

Here we focus on constraint-based methods, which rely on the equivalence of conditional independence (a property of the distribution) and directed separation (d-separation; a property of the graph). An important property is that an edge $Y_j - Y_k$ is missing in the skeleton of the DAG if and only if Y_j and Y_k are d-separated by at least one (possibly empty) set of nodes Y_S. Such Y_S is called a *separating set* for Y_j and Y_k. Given the equivalence of d-separation and conditional independence, this means that we can infer the presence of the edge $Y_j - Y_k$ in the skeleton by testing $\mathbf{Y_j} \perp\!\!\!\perp \mathbf{Y_k} | \mathbf{Y_S}$ for all $\mathbf{Y_S}$. The PC- and related algorithms therefore start with a fully connected undirected graph, and remove the edge $Y_j - Y_k$ whenever $\mathbf{Y_j}$ and $\mathbf{Y_k}$ are found to be conditionally independent for some $\mathbf{Y_S}$. While the first constraint-based algorithms such as IC [27] exhaustively tested all possible subsets, the PC-algorithm ([5]) can often greatly reduce the number of subsets to be considered.

Although structural equations are often assumed to be linear and the noise Gaussian, these assumptions are not essential for the equivalence of conditional independence and d-separation. For example, the pcalg package [28] contains a conditional independence test for binary data. However, structural equations with additional random effects are rarely considered, and many causal inference algorithms therefore assume independent observations.

2.1.8 Existing approaches for estimating $\mathcal{G}_y$, given genetic effects

To deal with the dependence introduced by the genetic effects, [12] and [13] proposed to predict the total genetic effects (i.e., the terms $\mathbf{u_i} = \mathbf{g_i}\Gamma$ in (6)), and perform causal inference on the residuals. These methods are flexible in the sense that any genomic prediction method can be used, and combined with any causal inference method.

A disadvantage however is that the presence of direct genetic effects cannot be tested. Suppose for example $G \to Y_1 \to Y_2 \to Y_3$, and we subtract the total genetic effects. Then given only the residuals, we can never know if part of the genetic variance of Y_2 was due to a direct effect $G \to Y_2$. To use the causal information contained in the genetic effects, [13] estimated 'genomic networks', based on the predictions themselves. These however seem to require additional assumptions not required for the residual networks (in particular, diagonal Σ_G), and it seems difficult to relate edges in such a network to direct genetic effects (Appendix G.2). In summary, residual and genomic networks only estimate the subgraph $\mathcal{G}_\mathcal{Y}$ of trait to trait relations, instead of the complete graph $\mathcal{G}$.

Another disadvantage is that the MTM (9) can only be fitted for a handful of traits [25], for statistical as well as computational reasons. For example, [29] showed that for general Gaussian covariance models, (residual) ML-estimation behaves like a convex optimization problem only when $n \gtrsim 14p$. Similar problems are likely to occur for Bayesian approaches. The main problem with fitting the MTM to data from GSEM model (8) is that one cannot exploit the possible sparseness of $\mathcal{G}$. Even for sparse graphs with few direct genetic effects, the matrices $\Gamma^t\Sigma_G\Gamma$ and $\Gamma^t\Sigma_E\Gamma$ may still be dense, requiring a total of $p(p+1)$ parameters. To overcome these limitations, we consider the presence or absence of direct genetic effects to be part of the causal structure, and develop pcgen, a causal inference approach directly on $\mathcal{G}$.

2.2 The pcgen algorithm

The main idea behind pcgen is that the PC-algorithm is applicable to any system in which d-separation and conditional independence are equivalent, and where conditional independence can be tested. We first describe the algorithm and propose independence tests; the equivalence is addressed in section 3.4. If we define $\mathbf{Y_{p+1}} := \mathbf{G}$ and temporarily rename the node G as Y_{p+1}, pcgen is essentially the PC-algorithm applied to $Y_1, \ldots, Y_{p+1}$:

1. **skeleton-stage.** Start with the fully connected undirected graph over $\{Y_1, \ldots, Y_{p+1}\}$, and an empty list of separation sets. Then test the conditional independence between all pairs $\mathbf{Y_j}$ and $\mathbf{Y_k}$, given subsets of other variables $\mathbf{Y_S}$. Whenever a p-value is larger than the pre-specified significance threshold α, update the skeleton by removing the edge $Y_j - Y_k$, and add Y_S to the list of separation sets for Y_j and Y_k. This is done for conditioning sets of increasing size, starting with the empty set ($S = \emptyset$; marginal independence between $\mathbf{Y_j}$ and $\mathbf{Y_k}$). Only consider S that, in the current skeleton, are adjacent to Y_j or Y_k.

2. **orientation-stage.** Apply the orientation rules given in Appendix A (R1-R3 in Algorithm 1) to the skeleton and separating sets found in the skeleton-stage. For example, if the skeleton is $Y_1 - Y_2 - Y_3$ and $\{Y_2\}$ is *not* a separating set for Y_1 and Y_3, the skeleton is oriented $Y_1 \to Y_2 \leftarrow Y_3$; otherwise, none of the two edges can be oriented.

To get pcgen, we only need to make a few refinements to these steps. First, in the skeleton stage we need to specify *how* to test conditional independence statements. Clearly, independence between two traits requires a different test than independence between a trait and $\mathbf{G}$ (i.e., $\mathbf{Y_{p+1}}$), in particular because the latter is not directly observed. Second, we need to modify the orientation rules, in order to avoid edges pointing into G. The usual rules give the correct orientations given perfect conditional independence information, but statistical errors in the tests may lead to edges of the form $Y_j \to G$. Third, statistical errors can also make the output of pc(gen) order-dependent. We therefore adopt the PC-stable algorithm of [30], who proposed to perform all operations in skeleton- and orientation-stage list-wise (details given in Appendix A).

In summary, pcgen is the PC-stable algorithm with: (1) specific conditional independence tests (described in sections 2.2.1-2.2.3 below), and (2) modified orientation rules, in order to avoid edges pointing into G (Appendix A.2). As in the original PC-algorithm, the number of type-I and type-II errors occurring in the tests is determined by the choice of the significance threshold α, which is discussed in sections 2.5 and 4.3.

2.2.1 Skeleton stage: conditional independence tests

Writing again $\mathbf{G}$ for $\mathbf{Y_{p+1}}$, we can distinguish the following types of conditional independence statements in the skeleton stage:

$$\mathbf{Y_j} \perp\!\!\!\perp \mathbf{Y_k} | (\mathbf{G}, \mathbf{Y_S}), \tag{A}$$

$$\mathbf{Y_j} \perp\!\!\!\perp \mathbf{G} | \mathbf{Y_S}, \tag{B}$$

$$\mathbf{Y_j} \perp\!\!\!\perp \mathbf{Y_k} | \mathbf{Y_S}, \tag{C}$$

where $j, k \in \{1, \ldots, p\}$ $(j \neq k)$ and $S \subseteq \{1, \ldots, p\} \setminus \{j, k\}$ (or $S \subseteq \{1, \ldots, p\} \setminus \{j\}$ in statement (A)). In words, (B) means that the trait $\mathbf{Y_j}$ is independent of all genetic effects ($\mathbf{G}$), conditional on the traits $\mathbf{Y_m}$ $(m \in S)$. If S is the empty set, this is understood as marginal independence of $\mathbf{Y_j}$ and $\mathbf{G}$. Similarly, (A) and (C) express conditional independence of traits $\mathbf{Y_j}$ and $\mathbf{Y_k}$ given $\mathbf{G}$ and $\mathbf{Y_S}$, or given $\mathbf{Y_S}$ alone.

We now propose statistical tests for statements (A) and (B), which rely on the linearity of our GSEM (model (5)). Statement (C) can be tested using standard partial correlations and Fisher's z-transform. However, as we show in appendix F, this test is redundant, since for any set Y_S that d-separates Y_j and Y_k, the set $Y_S \cup \{G\}$ will also d-separate them. We therefore skip any test for $\mathbf{Y_j} \perp\!\!\!\perp \mathbf{Y_k} | \mathbf{Y_S}$, and instead test the corresponding statement with $\mathbf{G}$, i.e., $\mathbf{Y_j} \perp\!\!\!\perp \mathbf{Y_k} | \mathbf{Y_S}, \mathbf{G}$.

2.2.2 Testing $\mathbf{Y_j} \perp\!\!\!\perp \mathbf{Y_k}|(\mathbf{G}, \mathbf{Y_S})$

For statement (A) we consider two different tests:

- The residual covariance (RC) test. The RC-test is based on the bivariate distribution of $(\mathbf{Y_j}, \mathbf{Y_k})$ conditional on the observed $\mathbf{Y_S} = \tilde{y}_S$, which can be written as

$$\begin{pmatrix} \mathbf{Y_j} \\ \mathbf{Y_k} \end{pmatrix} |\mathbf{Y_S} = \tilde{y}_S \sim N\left(\begin{pmatrix} \mu_{j|S} \\ \mu_{k|S} \end{pmatrix}, \Sigma_{jk|S} \right). \tag{11}$$

  Expressions for $\mu_{j|S}$, $\mu_{k|S}$ and $\Sigma_{jk|S}$ can be derived from equation (8), and are given in appendix E.6. For testing (A) it is assumed that $\mu_{j|S}$ and $\mu_{k|S}$ are of the form

$$\mathbf{X}B\gamma_j + \tilde{y}_S\beta_S^{(j)}, \quad \mathbf{X}B\gamma_k + \tilde{y}_S\beta_S^{(k)}, \tag{12}$$

  where $\mathbf{X}B\gamma_j$ is the marginal mean of $\mathbf{Y_j}$ (see (10)), and $\beta_S^{(j)}$ and $\beta_S^{(k)}$ are $|S| \times 1$ vectors of regression coefficients. The covariance in (11) is assumed to be of the form

$$\Sigma_{jk|S} = V_G(jk|S) \otimes K + V_E(jk|S) \otimes I_n, \tag{13}$$

  for some 2×2 matrices $V_G(jk|S)$ and $V_E(jk|S)$. Given these assumptions, we test whether the off-diagonal element in $V_E(jk|S)$ is zero, using the likelihood ratio test (LRT) described in Appendix A.3. In words, the conditional distribution is a bivariate MTM, in which we test the residual covariance[3].

  The underlying idea is that a nonzero residual covariance must be the consequence of an edge $Y_j \to Y_k$ or $Y_k \to Y_j$, because of the assumed normality and causal sufficiency. On the other hand, a nonzero *genetic* covariance may also be due to covariance between direct genetic effects on these variables, or due to a genetic effect on a common ancestor. The RC-test therefore compares the full bivariate mixed model with the submodel with diagonal $V_E(jk|S)$, while accounting for all genetic (co)variance using $V_G(jk|S)$.

- The RG-test (*R*esiduals of *G*BLUP), which is based on the residuals of MTM (9). Solving the mixed model equations, we obtain the BLUP $\mathbf{U}^*$ of the total genetic effects $\mathbf{U} = \mathbf{G}\Gamma$, the BLUE $\tilde{B}^*$ of the fixed effects, and residuals $\mathbf{Y} - \mathbf{U}^* - \mathbf{X}\tilde{B}^*$. Next, we test statement (A) using the partial correlation between the residuals of $\mathbf{Y_j}$ and $\mathbf{Y_k}$, given those of $\mathbf{Y_S}$, and assess significance with Fisher's z-transform. This is essentially the test used by [12] and [13], who instead took a fully Bayesian approach to predict $\mathbf{U}$. Here we stick to the GBLUP, and consider two variants: (i) the one described above, based on the multivariate GBLUP, and (ii) based on univariate GBLUPs, i.e. all predictions $\mathbf{U}_\mathbf{j}^*$ are obtained from single trait mixed models. In both cases, the idea is that when $\mathbf{U}^*$ is close enough to $\mathbf{U}$, it follows

[3] alternatively, we could test the residual correlation

from equations (7)-(8) that $vec(\mathbf{Y}-\mathbf{U}^*)$ approximately has covariance $(\Gamma^t\Sigma_E\Gamma)\otimes I_n$, i.e. that of independent samples, without any genetic relatedness.

Both tests rely on approximations: they do not directly test conditional independence statement (A), but a related statement. In some cases the approximation is exact, and the related statement equivalent. In other cases it is not entirely equivalent, which may introduce an additional source of error (on top of the statistical errors that come with the test itself). In the RG-test the prediction error $(\mathbf{U}^* - \mathbf{U})$ determines the quality of the approximation. The RC-test relies on assumptions (12)-(13) about the conditional mean and covariance of $(\mathbf{Y_j}, \mathbf{Y_k})$ given $\mathbf{Y_S} = \tilde{y}_S$. We discuss the appropriateness of these approximations in section 3.4.3.

Model (13) (with $S = \emptyset$) can also provide an estimate of the total genetic covariance, i.e., the off-diagonal element of $V_G(jk|S)$. In Appendix A.4 we describe a test for zero genetic covariance, which can be useful for data exploration, but has no role in pcgen. A similar test for the total genetic correlation can be obtained using the delta method ([26]).

2.2.3 Testing $\mathbf{Y_j} \perp\!\!\!\perp \mathbf{G}|\mathbf{Y_S}$

Our test for statement (B) is based on the intuition that $\mathbf{Y_j}$ is independent of $\mathbf{G} = [\mathbf{G_1} \cdots \mathbf{G_p}]$ given $\mathbf{Y_S}$, whenever there is no direct genetic effect on $\mathbf{Y_j}$ (i.e. $\mathbf{G_j} = \mathbf{0}$), and all directed paths from G to Y_j are blocked by the set $\{Y_m : m \in S\}$. In particular, if S is the empty set, there should not be any directed path from G to Y_j. Because directed paths from G to Y_j will generally introduce some genetic variance in $\mathbf{Y_j}$, the idea is to test whether there is significant genetic variance in the conditional distribution of $\mathbf{Y_j}$ given $\mathbf{Y_S} = \tilde{y}_S$. This is done as follows:

- When $K = ZZ^t$, we use the classical F-test in a 1-way ANOVA, with $\mathbf{X}$ and $\tilde{y}_S$ as covariates. Technically, this is an ANCOVA (analysis of covariance), where the treatment factor genotype is tested conditional on the covariates being in the model.
- For other K one can use the LRT, as in the RC-test for (A). The asymptotic distribution under the null-hypothesis is a mixture of a point mass at zero and a chi-square.

In both cases, it is assumed that the mean of the conditional distribution of $\mathbf{Y_j}$ given $\mathbf{Y_S} = \tilde{y}_S$ is of the form (12), and the covariance of the form

$$\Sigma_{j|S} = \sigma^2_G(j|S)K + \sigma^2_E(j|S)I_n, \tag{14}$$

for some variance components $\sigma^2_G(j|S)$ and $\sigma^2_E(j|S)$. Again, the covariance assumption holds exactly when $K = ZZ^t$ (Theorem 6); otherwise it is an approximation. As in the RC-test for statement (A), (12) is an approximation as well, and the assumed linearity

is essential. Suppose for example that $\mathbf{Y_2} := \mathbf{Y_1}^2$ and $\mathbf{Y_1} := \mathbf{G_1} \sim N(0, ZZ^t)$, where for the sake of the argument we assume absence of residual errors. Then the factor genotype will generally be significant in the ANCOVA with Y_1 as covariate. For example there could be 2 replicates of 3 genotypes, with genetic effects $(-1, -1, 0, 0, 1, 1)$, then clearly there is some unexplained genetic variance when regressing $\mathbf{Y_2} = (1, 1, 0, 0, 1, 1)^t$ on $\mathbf{Y_1}$.

2.2.4 pcRes: reconstructing only trait-to-trait relationships

Testing only conditional independencies of the form (A), one can reconstruct the graph $\mathcal{G}_\mathcal{Y}$ of trait-to-trait relations. Moreover, if this is done with the RG-test, the algorithm is very similar to the residual approaches of [12] and [13]. Staying within the context of the PC-algorithm and using residuals from GBLUP, we will call this pcRes. Similar to pcgen with the RG-test, the performance of pcRes strongly depends on the prediction error of the GBLUP.

2.3 Causal inference based on genotypic means

In section 2.1.5 we assumed the genetic relatedness to be either $K = ZZ^t$ (given observations on replicates), or a marker-based relatedness matrix (given a single observation on each genotype). However, in many applications both a marker-based relatedness matrix and replicates are available. Suppose we have r replicates of m genotypes in a completely randomized design, with a $m \times m$ relatedness matrix A. [4] Reconstruction of $\mathcal{G}$ with pcgen or $\mathcal{G}_\mathcal{Y}$ with pcRes is then possible using:

1. both A and the replicates. The distribution of the data is assumed to be

$$vec(\mathbf{Y}) \sim N\left(vec(\mathbf{X}(B\Gamma)), (\Gamma^t\Sigma_G\Gamma) \otimes (ZAZ^t) + (\Gamma^t\Sigma_E\Gamma) \otimes I_n\right), \tag{15}$$

   i.e. equation (8) with $K = ZAZ^t$.

2. only the replicates. The true distribution of the data could be as in (15), but ignoring the information in A, we assume that

$$vec(\mathbf{Y}) \sim N\left(vec(\mathbf{X}(B\Gamma)), (\Gamma^t\Sigma_G\Gamma) \otimes (ZZ^t) + (\Gamma^t\Sigma_E\Gamma) \otimes I_n\right). \tag{16}$$

3. genotypic means and A. Assuming that the original plant or plot data are distributed as in (15), the distribution of the $m \times p$ matrix $\bar{\mathbf{Y}}$ of genotypic means is such that

$$vec(\bar{\mathbf{Y}}) \sim N\left(0, (\Gamma^t\Sigma_G\Gamma) \otimes A + \frac{1}{r}(\Gamma^t\Sigma_E\Gamma) \otimes I_m\right), \tag{17}$$

   where it is assumed that the fixed effects have been accounted for in the estimation of genotypic means.

[4] We use the letter A to avoid confusion with the general $n \times n$ matrix K, and also because A typically models additive effects.

In our simulations (section 3.1 below) we compare these cases for pcRes. A similar comparison for pcgen is left for future work; here we will focus on the second case, using only replicates and assuming independent genetic effects. This is motivated by the good results for pcRes for this case, and by theoretical arguments (see Theorems 5-6 below, and the discussion (4.2)). However, we stress that in all simulations, the data are simulated as in (15), i.e. with additive genetic effects, inducing population structure.

A second issue in pcRes (independent of the choice of K) are the residuals, which can be obtained from single- or multi-trait models. In the latter case, we fit the multi-trait model (9), with K equal to ZAZ^t, ZZ^t or A. In the univariate case, we do the same, assuming the covariance model $\sigma_G^2 K + \sigma_E^2 I$ (where I is of dimension $n \times n$ or $m \times m$, depending on K).

2.4 Software, and overview of algorithms

We now give an overview of pcgen, pcRes, and variations on these methods. Most of these are available in our R-package pcgen, freely available at `https://cran.r-project.org/web/packages/p`
pcgen is built on the pcalg package ([28], [31]), in which we modified the orientation rules and the default conditional independence test. Tables 5 and 6 in Appendix D provide a brief description, with the abbreviations and the required R-commands.

pcgen can be run with either the RC- or the RG-test (pcgen-RC versus pcgen-RG). In case of the RG-test, either univariate or multivariate GBLUP could be used (pcgen-RG-uni versus pcgen-RG-multi). pcRes can be based on either univariate or multivariate GBLUP (pcRes-uni versus pcRes-multi). In either case, one can use a relatedness matrix only (postscript -K, for 'kinship'), only replicates (-R), or both replicates and a relatedness matrix (-RK). Finally, one could reconstruct $\mathcal{G}_{\mathcal{Y}}$ using the GBLUP itself (pc-GBLUP), similar to the approach of [13].

2.5 Assessing uncertainty

The PC-algorithm is asymptotically correct, in the sense that the underlying CPDAG is recovered if conditional independence can be tested without error [5]. In Theorem 2 below we provide a related consistency result for pcgen. In practice however, type-I or type-II errors are likely to occur, leading to incorrect edges in the graph. Depending on the significance level α used in each test, there may be more type-I errors (large α) or rather more type-II errors (small α). However, reliable control of the (expected) false positive rate or total number of false positives remains challenging; see the discussion, section 4.3. We will therefore just consider the p-values as they are, and analyze a given dataset for various significance thresholds. A rough indication of confidence for each remaining edge is given by the largest p-value found across all conditioning sets for which the edge was tested.

2.6 Extensions of pcgen

We conclude the materials and methods section with the following extensions of pcgen:

- **The RC-test with prior screening (pcgen-screening):** The RC-test for statement (A) is computationally efficient, in the sense that only a bivariate MTM needs to be fitted, instead of for all traits. At the same time however, once residuals are available, the RG-test is much faster than the RC-test, as the former is based on partial correlations that can be computed recursively (see e.g [32]). An appealing strategy is therefore to run PC-stable on univariate residuals (i.e., a form of pcRes), and use the resulting skeleton as the starting point for pcgen with the RC-test. The advantage, at least for sparse graphs, is that the number of conducted RC-tests is greatly reduced. In the pcgen-package this is implemented in the pcgenFast function. The skeleton based on univariate residuals typically contains somewhat more false edges, but these may be removed later on with the RC-test.

- **Inclusion of QTLs:** apart from the random genetic effects, the GSEM considered here could include fixed effect QTLs as well. Each QTL is represented by a single node, which like the random effect node G is always a root node. No edges are allowed between QTLs or between a QTL and G; moreover every edge between a QTL and a trait is oriented towards the trait. Since the QTLs may further improve the orientation of the graph, the pcgen package provides an experimental implementation of this, although a full investigation of the added value is left for future work.

- **Comparing pcgen output with genetic variance estimates:** for every trait $\mathbf{Y_j}$ having positive genetic variance, there should be either a direct genetic effect $G \to Y_j$ or a partially directed path from G to Y_j (with possibly undirected edges, but all directed edges pointing towards Y_j). However, because of statistical errors it may happen that neither of the two exist in the CPDAG obtained from pcgen, while at the same time the genetic variance (considered in the independence test $\mathbf{Y_j} \perp\!\!\!\perp \mathbf{G}$) is significant. In the pcgen-package, such conflicts can be detected using the checkG function. If conflicts occur, we conclude that there was insufficient evidence to remove the direct genetic effect from the graph, and re-run pcgen, skipping all tests $\mathbf{Y_j} \perp\!\!\!\perp \mathbf{G}|\mathbf{Y_S}$ for all $\mathbf{Y_j}$ that produced conflicts in the first run. This forces the algorithm to keep the edge $G \to Y_j$.

3 Results

3.1 Simulations, with randomly drawn graphs

3.1.1 Simulation scheme

To compare the different algorithms we simulated random GSEMs, by combining a randomly drawn DAG over the traits ($\mathcal{G}_\mathcal{Y}$) with random path coefficients, closely following the simulation scheme of [32]. Traits were simulated for 200 genotypes which (for each simulated dataset) were randomly drawn from an existing population of 254 maize hybrids [33]. Two replicates of each genotype were simulated. For each simulated dataset we randomly sampled the set D (defining the edges $G \to Y_j$) and the covariance matrix Σ_G. Given an additive relatedness matrix A based on 50k SNPs, genetic effects were simulated as in (15), i.e. such that $vec(\mathbf{G}) \sim \Sigma_G \otimes (ZAZ^t)$. Appendix B.1 provides further details, such as the magnitude of genetic (co)variances. We focus here on the comparison of

- pcgen-RC: pcgen with the RC-test, using replicates
- pcgen-RG-uni: pcgen with the RG-test, based on univariate GBLUP, and using replicates
- pcRes-uni-R: pcRes based on univariate GBLUP, using replicates
- pcRes-multi-K: pcRes based on multivariate GBLUP, using genotypic means and the relatedness matrix A that was used to simulate the data
- pcgen-screening: pcgen with the RC-test, starting with the skeleton obtained from pcRes-uni-R (see section 2.6)

Results for the other algorithms are given in Appendix B.3. In all simulations the significance threshold was $\alpha = 0.01$. The effect of sample size and the trade-off between power and false positives as function of α was already investigated by [32] for the standard PC-algorithm, and is likely to be similar for pcgen.

We separately evaluated the reconstruction of $\mathcal{G}_\mathcal{Y}$ and the edges $G \to Y_j$, as the latter is only possible with pcgen. To assess the difference between estimated and true skeleton of $\mathcal{G}_\mathcal{Y}$, we considered the true positive rate (TPR), the true discovery rate (TDR) and the false positive rate (FPR). Additionally we used the Structural Hamming Distance (SHD), which also takes into account the orientation of the edges. Appendix B.2 provides definitions of these criteria. Reconstruction of $G \to Y_j$ is only assessed in terms of TPR, TDR and FPR, as these edges can have only one orientation.

3.1.2 Results

We first performed simulations with $p = 4$ traits (scenario 1), with each potential edge between traits occurring in the true graph with probability $p_t = 1/3$. Hence, for any

given trait, the expected number of adjacent traits was $(p-1)p_t = 1$. The edges $G \to Y_j$ were included in the true graph with probability $p_g = 1/2$. In a related set of simulations (scenario 2), p_t was increased to 0.5, giving denser graphs. In both scenarios, pcgen reconstructed the edges $G \to Y_j$ with little error, the average TPR being above 0.95 and the TDR above 0.99. The FPR for these type of edges remained below 0.05. For the trait-to-trait relations (i.e. reconstruction of $\mathcal{G}_\mathcal{Y}$), the TPR, FPR and TDR all increased in scenario 2. Hence, for denser graphs, more of the true edges were found, at the expense of a somewhat higher number of false edges.

pcRes with univariate residuals from replicates (pcRes-uni-R) outperformed pcRes-multi-K, despite the fact that only the latter used the actual relatedness matrix (from which the genetic effects were simulated). Consequently, the information contained in the replicates appears much more important than the precise form of the relatedness matrix, or unbiased estimation of genetic correlations. The performance of pcRes strongly depends on the prediction error of the GBLUP (see sections 2.2.2 and 2.2.4), and, in line with the results of [18], this error appeared lowest when using the replicates. pcRes-uni-R performed almost as well as the 3 pcgen approaches; motivated by the additional computational advantage, this is the only variant of pcRes that we consider in the remainder. For the same reasons, we only considered the RG-test in pcgen in the univarate version, based on replicates (pcgen-RG-uni). The latter appeared to give more false positives than pcgen with the RC-test (pcgen-RC), although differences were small. A good compromise between computational speed and performance appears to be achieved with pcgen-screening, starting pcgen-RC with the skeleton found by pcgen-RG-uni. This will therefore be our default choice for analyzing real data below.

pcRes-uni-R and pcgen-RG-uni had identical performance in terms of TPR, TDR and FPR, as they use exactly the same tests for the trait-trait relations. However, pcgen-RG-uni (as other pcgen approaches) had the advantages that (i) also the edges $G \to Y_j$ could be inferred and (ii) due to these genetic effects, the orientation of the edges between traits was improved, as shown by a strongly reduced SHD.

To assess performance in higher dimensions, we simulated data sets with $p = 20$ traits, $p_g = 0.3$ and $p_t = 0.1$ (scenario 3) and with $p = 100$, $p_g = 0.1$ and $p_t = 0.01$ (scenario 4). Both scenarios consider sparse graphs; denser graphs can be analyzed as well, but, for p larger than 20-30, require several hours or even days, unless the size of the conditioning sets is restricted or pcgen would be parallelized. Here we limited the size of conditioning sets to 3 (scenario 3) and 2 (scenario 4). As in the first two scenarios, pcgen led to a strong reduction in SHD, and reliable reconstruction of the direct genetic effects.

	$\mathcal{G}_{\mathcal{Y}}$				$G \to Y_j$		
	TPR	FPR	TDR	SHD	TPR	FPR	TDR
Scenario 1 ($p = 4$)	($p_t = 1/3$)				($p_g = 0.5$)		
pcgen-RC	0.607	0.012	0.960	0.568	0.986	0.020	0.994
pcgen-RG-uni	0.607	0.036	0.906	0.670	0.984	0.022	0.994
pcgen-screening	0.603	0.008	0.982	0.548	0.986	0.020	0.994
pcRes-uni-R	0.607	0.036	0.906	1.362			
pcRes-multi-K	0.465	0.432	0.388	3.212			
Scenario 2 ($p = 4$)	($p_t = 0.5$)				($p_g = 0.5$)		
pcgen-RC	0.814	0.042	0.980	1.236	0.966	0.041	0.993
pcgen-RG-uni	0.820	0.074	0.951	1.320	0.965	0.043	0.992
pcgen-screening	0.805	0.034	0.990	1.238	0.966	0.042	0.992
pcRes-uni-R	0.820	0.074	0.951	2.274			
pcRes-multi-K	0.666	0.466	0.603	3.778			
Scenario 3 ($p = 20$)	($p_t = 0.1$)				($p_g = 0.3$)		
pcgen-RG-uni	0.911	0.004	0.961	7.020	0.968	0.018	0.991
pcgen-screening	0.895	0.002	0.985	6.800	0.969	0.018	0.991
pcRes-uni-R	0.912	0.004	0.961	9.866			
Scenario 4 ($p = 100$)	($p_t = 0.01$)				($p_g = 0.1$)		
pcgen-RG-uni	0.959	0.001	0.942	27.298	0.976	0.022	0.943
pcRes-uni-R	0.962	0.001	0.940	38.404			

Table 1. Performance of pcgen and residuals-based approaches, averaged over 500 simulated data-sets per scenario. For scenarios 1 and 2, performance of additional residual approaches is given in Appendix B.3.

3.2 Simulations, using a crop-growth model

We also simulated data using the popular crop growth model APSIM [34,35]. Compared to the preceding simulations this represents a more challenging (and probably more realistic) scenario, as several of the underlying assumptions are violated. In particular, the data-generating process introduces nonlinearities and latent variables. We simulated 12 traits for an existing wheat population of 199 genotypes, with 3 replicates each. The traits include 7 primary traits, 4 secondary traits and yield. Appendix B.4 provides trait acronyms (Table 3) and further details. Traits were simulated by running a discrete dynamic model from the beginning ($t = 0$) to the end of the growing season ($t = T$). Motivated by the fact that some trait measurements are destructive, observations are only taken at $t = T$. Figure 3a shows the summary graph, defining the causal effects from one time-step to the next ([24]). It does not directly describe the joint distribution of the traits at $t = T$ (obtained by marginalizing over $t = 0, \ldots, T-1$, and typically represented by an ancestral graph [36]), but for simplicity we nevertheless investigate to

what extent we can reconstruct the summary graph. There are direct genetic effects on all of the primary traits, which have heritability 0.9. The genetic effects originate from a large number of trait-specific QTLs, with randomly drawn effect sizes. There are no direct genetic effects on the secondary traits and yield.

Figure 3b shows the pcgen estimate for a single simulated data-set, with pcgen-RC and prior screening (pcgen-screening), and $\alpha = 0.01$. Similar results (not shown) were obtained for other data-sets, smaller α or without screening. Compared to the simulations above, it turned out to be much harder to detect the absence of direct genetic effects: in the pcgen reconstruction, all 12 traits had such effects (highest p-value: $1.7 \cdot 10^{-4}$). Because of the large degrees of freedom for genotype and generally small residual variances, most of the F-statistics for genotype are highly significant, but often orders of magnitude smaller than the F-ratios for the conditioning traits. For example, in the test for a direct genetic effect on flowering time (FT) conditional on SV and TFI, F-ratios for SV, TFI and genotype were respectively 3469.95, 68744.14 and 5.71. As in our example in section 2.2.3, another reason for the significant genetic effects may be the nonlinearity of the underlying model.

The reconstructed trait-to-trait relations were mostly correct, except for the missing edge $GN \to Y$, and one incorrect orientation ($Y \to GW$). pcRes made the same errors (Figure 3c), with an additional false arrow ($MGS - SP$). The standard PC-stable algorithm applied to all traits and QTLs lead to many more errors (Figure 3d), for example the false edge between GW and RUE, the missing edge $TFI \to FT$ and the incorrect orientations $BM \to RUE$ and $Y \to BM$. These errors appeared to be the consequence of small effect QTLs not being detected; consequently part of the genetic variance was not taken into account.

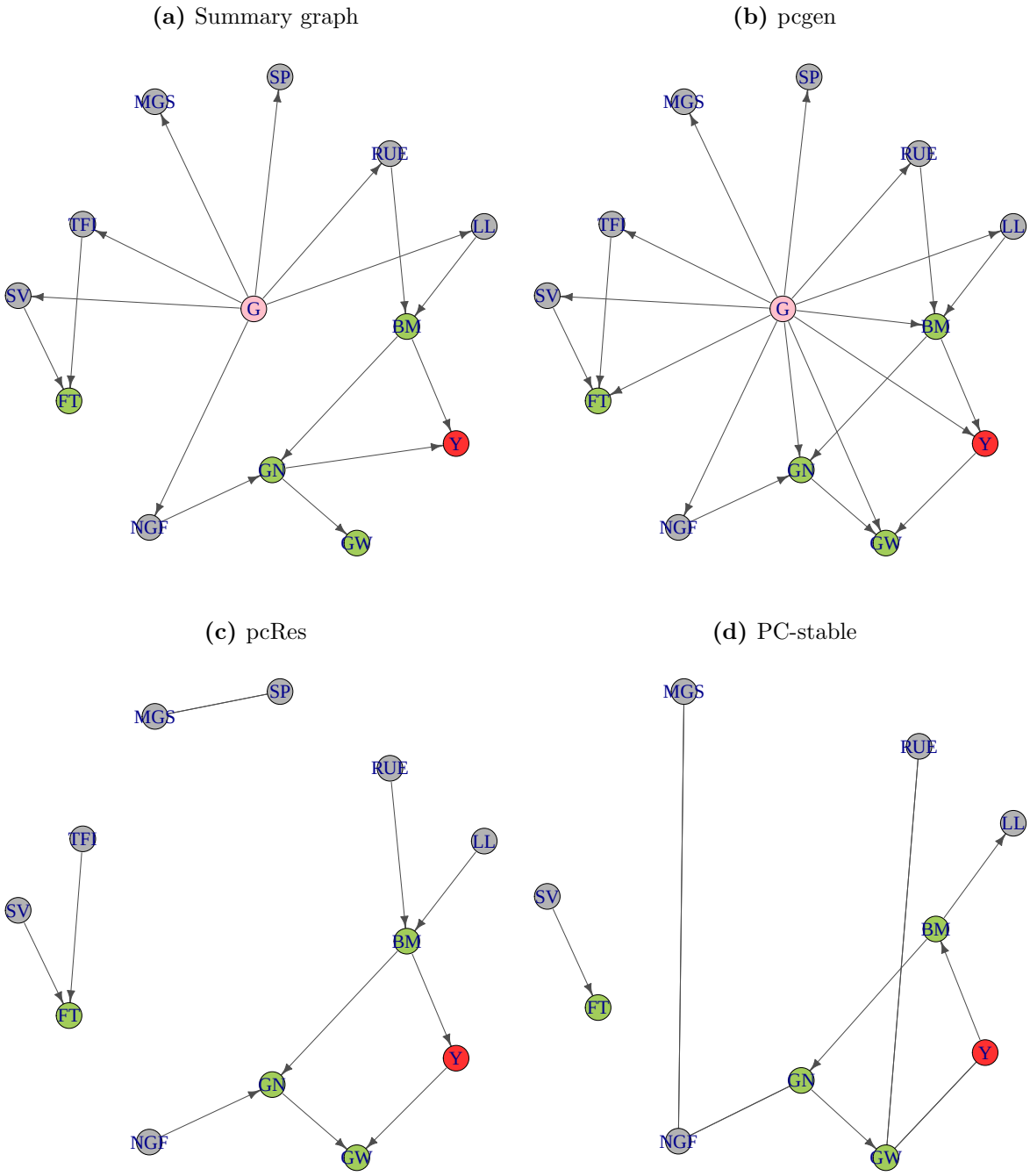


Figure 3. Networks of 12 APSIM traits in the dry environment Emerald 1993, including 7 primary traits (grey), 4 secondary traits (green) and yield (red). Trait acronoyms are given in supplementary Table 3 (Appendix B.4)

3.3 Real data

3.3.1 Maize

We now use pcgen-screening to analyze the field trials reported by [33], involving 254 hybrids of maize (*Zea mays*). We consider a subset of 6 trials, conducted in 2013 under two different watering regimes (rain-fed (-R) and irrigated (-I)) and at three locations: Karlsruhe (Germany), Nérac (France) and Graneros (Chile). Most of the trials included 3 height related traits (tassel height (TH), ear height (EH) and (total) plant height (PH)), 2 flowering time related traits (silking (S) and anthesis (A)), and 3 yield related traits (grain number (GN), grain weight (GW) and yield itself (Y)). All traits were measured independently, i.e. no trait measurement was derived from other traits.

Each trial was laid out as an alpha-lattice design, with two replicates for the irrigated trials and three for the rain-fed trials. Spatial trends and (in)complete block effects were estimated using the mixed model of [37] (R-package SpATS), and subtracted from the original data; pcgen was then applied to the detrended data, assuming a completely randomized design. Residuals from SpATS appeared approximately Gaussian, and no further transformation was applied. For the yield related traits, prior biological knowledge (Y being the product of GS and GN) might suggest a log-transform, but we assume that such biologiocal knowledge is not available to the algorithm.

Figure 4 shows the estimated networks for $\alpha = 0.01$. Networks obtained with $\alpha = 0.001$ (Appendix H) were mostly identical, except for 1 or 2 edges missing in the Karlsruhe-I, Nérac-R and Graneros-I trials. In all 6 trials there was a clear clustering of traits according to their biological nature (heigth, flowering and yield related). Edges across these groups were only found for the 3 rain-fed trials (comment on the $S - PH$ and $S - TH$ edges in Karlsruhe and Nérac, and the $PH - Y$ edge in Graneros). Except for the Graneros trials, direct genetic effects were present for all traits, which might again be the consequence of nonlinear relationships. As in the APSIM simulations, the test for direct genetic effects typically produced highly significant results, but often with small genetic effects. In the Karlsruhe-I trial for example, the ANCOVA for yield (Y) conditional on GN and GS gave an F-ratio for genotype of only 2.71, compared to 55364.24 and 8036.53 for respectively GN and GS. In two of the six trials pcgen correctly identified $GN \rightarrow Y \leftarrow GS$, i.e. the expected relations between yield and its components. In two other ones the edges $GN - Y - GS$ could not be oriented because of the additional (and unexpected) edge $GN - GS$. In the remaining two trials (Graneros-R and Nérac-I) we found $GS \rightarrow GN \leftarrow Y$. While biologically improbable, this simply represents the outcomes of the tests for the given data. In particular, the test for $\mathbf{Y_j} \perp\!\!\!\perp \mathbf{Y_k} | (\mathbf{G}, \mathbf{Y_S})$ is highly significant when $\mathbf{Y_j}$, $\mathbf{Y_k}$ and $\mathbf{Y_S}$ are respectively yield, grain size and grain number, but not significant when dropping grain number from the conditioning set ($p = 0.056$, for Nérac-I).

The trials also illustrate the distinction made in section 2.1.4, between the total genetic covariance and the covariance among direct genetic effects (as defined by Σ_G). For most

pairs of traits, the former was highly significant, with a typical (total) genetic correlation of $\rho_g = 0.3 - 0.8$ (for better interpretability we report correlations, although we test for zero genetic covariance, see Appendix A.4). Only for pairs involving GS, values were often negative, and not always significant. In the Karlsruhe-I trial for example, we found $\rho_g = -0.11$ for GS and EH, and $\rho_g = -0.41$ for GS and S. In both cases, the two traits are d-separated in the graph, but only for S (silking) the genetic covariance is significant ($p = 4.89 \cdot 10^{-9}$). While this may provide information about Σ_G, we recall that usually the latter is not entirely identifiable. For example, GN and Y always had a high genetic correlation, but without restrictions on the path coefficients one cannot estimate the corresponding element of Σ_G.

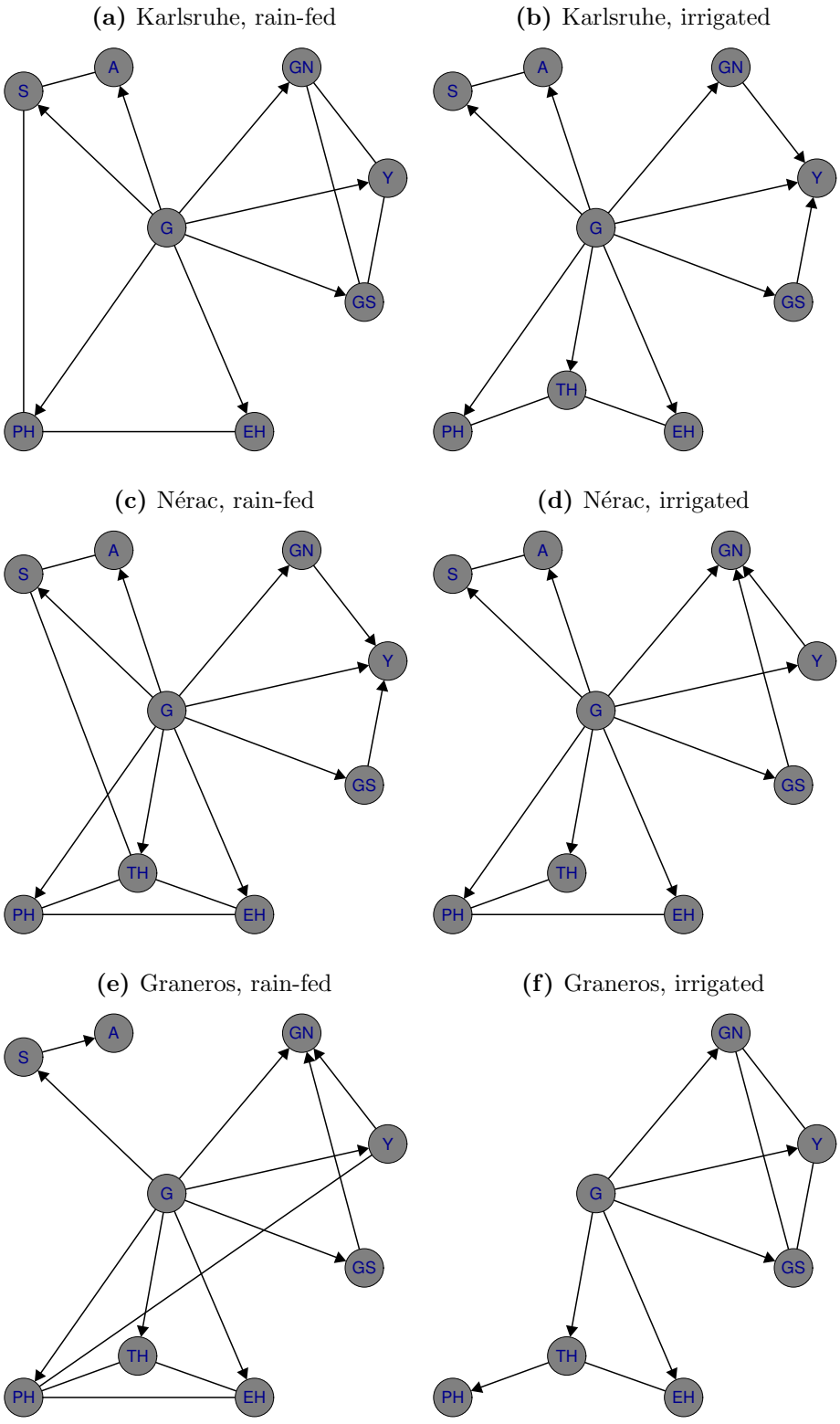


Figure 4. Estimated networks for six of the DROPS field trials in 2013, with $\alpha = 0.01$. Rows correspond to locations (Karlsruhe, Nérac, Graneros), columns to treatments (rain-fed, irrigated).

3.3.2 Rice

Using again pcgen-screening, we analyzed 34 traits measured on 274 *indica* genotypes of rice (*Oryza sativa*) under water-deficit, reported by [38]. Trait acronyms are given in Supplementary Table 4 in Appendix C. 3 replicates of each genotype were phenotyped in a randomized complete block design, and block was included as a covariate in all conditional independence tests, which were performed at significance level $\alpha = 0.01$. A first run of pcgen produced many inconsistencies in the genetic effects, i.e. traits with significantly positive heritability but without a partially directed path coming from the genotype node. We therefore applied the correction described at the end of section 2.6, adding edges $G \to Y_j$ for all traits Y_j with this inconsistency, and then re-ran pcgen. The final reconstruction is given in Figure 5, where traits are grouped into 4 shoot morphological traits (blue), 2 physiological traits (rose), 16 root morphological traits (green), 6 root anatomical traits (gray) and 6 dry matter traits (orange).

Even after correcting the inconsistencies, there were 10 traits without a direct genetic effect; for 7 of these there was a partially directed path starting from genotype, and first passing through $RL1015$ (root length, per plant, restricted to roots with diameter between 10 and 15 mm). Relations among traits mostly clustered according to their biological categories, except for the effect of TW (total weight) on CWT (cumulative water transpiration), and RW and RS (root and stem weight) appearing in a cluster of root morphology traits. In particular, there are no links from root morphological traits to biomass-related (physiological?) traits, which might be expected under water-deficit (cite ...?). Apart from errors in the test, this may be a consequence of the experimental setup, where the use of pots might have restricted root growth. For the anatomical traits, the lack of connections with trait categories could be the result of high measurement error, violating our assumption that residual errors are driven by biological variation.

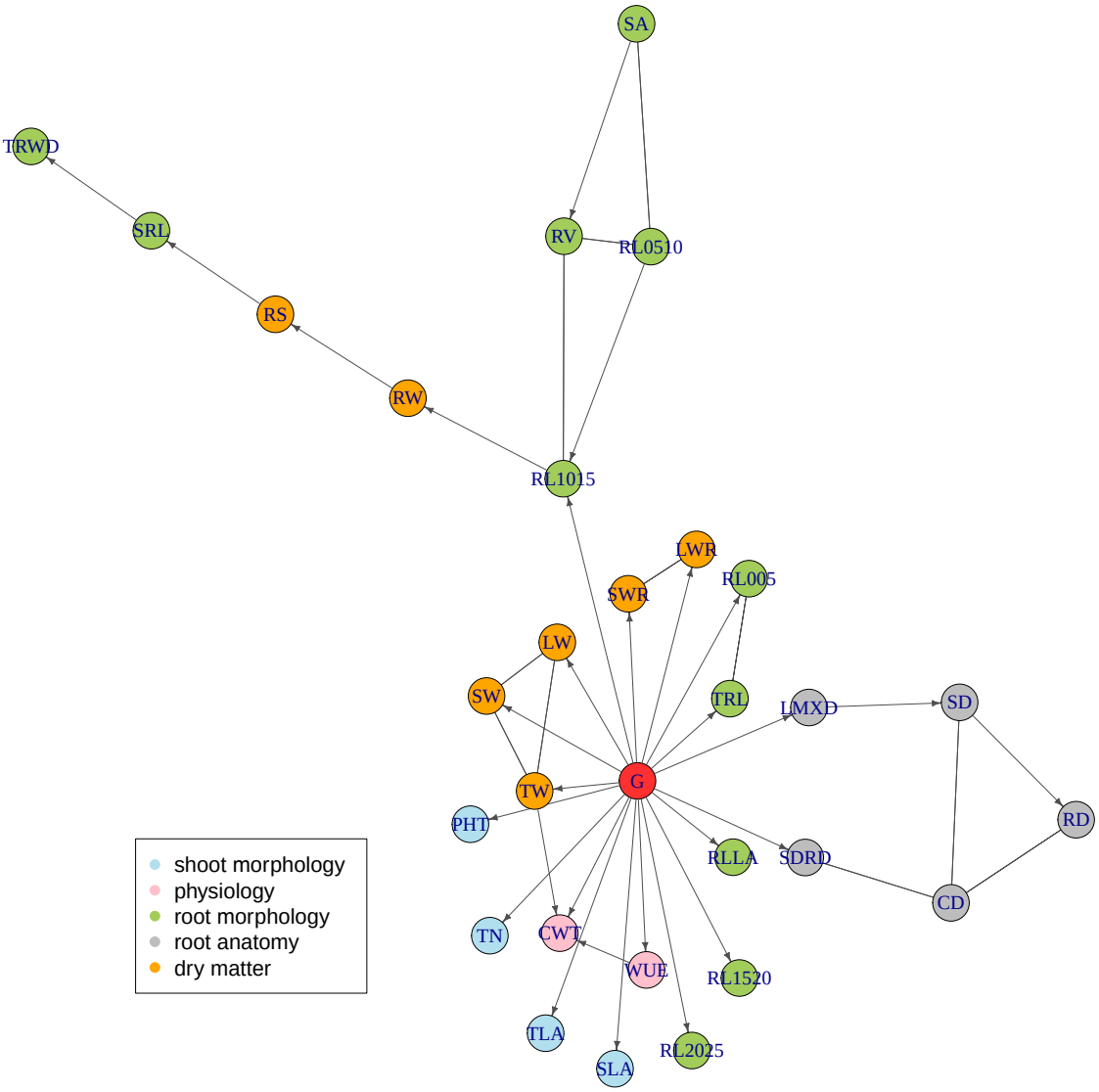


Figure 5. pcgen-reconstruction for the rice-data from [38], with $\alpha = 0.01$.

3.4 Statistical properties of pcgen

We now investigate a number of statistical issues: the assumptions required for asymptotic consistency of pcgen (section 3.4.1), the assumptions required for faithfulness (section 3.4.2), and properties of the conditional independence tests (section 3.4.3). Proofs of Theorems 1-6 are given in Appendix E.

3.4.1 Consistency

Asymptotic consistency holds if, for increasing sample size, the probability of finding the correct network converges to 1. Correct in this context means that we recover the class of partially directed graphs (CPDAG) that contains the underlying DAG. Consistency of the PC-algorithm was shown by [5] (for low dimensions) and [32] (for high dimensions). These authors distinguished consistency of the oracle version of PC, where conditional independence information is available without error, and the sample version, where conditional independence is obtained from statistical tests. For pcgen we will focus on the oracle version and consistency of the skeleton, leaving the sample version and the correctness of the orientations for future work.

As for the standard PC-algorithm, consistency of pcgen requires the equivalence between conditional independence and d-separation in the graph. Part of this is the Markov property, which states that d-separation of two nodes in the graph given a set of other nodes implies conditional independence of the corresponding random variables. The converse (conditional independence implying d-separation) is known as faithfulness. The following result provides the Markov property for SEM with genetic effects. The proof (Appendix E.8) is a straightforward adaptation of Pearl's proof for general SEM ([4]).

Theorem 1 *Suppose we have a GSEM as defined by equation* (5)*, with a graph* $\mathcal{G}$ *as defined in section 2.1.3, and satisfying assumptions 1-4 given in section 2.1.2. Then the global Markov condition holds for* $\mathcal{G}$ *and the joint distribution of* $\mathbf{G}, \mathbf{Y_1}, \ldots, \mathbf{Y_p}$*. In particular, d-separation of* Y_j *and* G *given* Y_S *implies* $\mathbf{Y_j} \perp\!\!\!\perp \mathbf{G} | \{\mathbf{Y_S}\}$*, and d-separation of* Y_j *and* Y_k *given* $\{Y_S, G\}$ *implies* $\mathbf{Y_j} \perp\!\!\!\perp \mathbf{Y_k} | \{\mathbf{Y_S}, \mathbf{G}\}$*, for all traits* $\mathbf{Y_j}$ *and* $\mathbf{Y_k}$ *and subsets* $\mathbf{Y_S}$*.*

If we now assume faithfulness, the preceding result directly gives the equivalence between conditional independence and d-separation. This in turn implies that pcgen will recover the correct skeleton, at least if conditional independence can be tested without error:

Theorem 2 *Let* $dsep(\mathcal{G})$ *denote d-separation in the graph* $\mathcal{G}$*. Suppose we have a GSEM as in Theorem 1, and we make the additional assumptions of faithfulness:*

$$\mathbf{Y_j} \perp\!\!\!\perp \mathbf{Y_k} | \{\mathbf{Y_S}, \mathbf{G}\} \implies Y_j \; dsep(\mathcal{G}) \; Y_k | \{Y_S, G\} \tag{18}$$

$$\mathbf{Y_j} \perp\!\!\!\perp \mathbf{G} | \{\mathbf{Y_S}\} \implies Y_j \; dsep(\mathcal{G}) \; G | \{Y_S\}, \tag{19}$$

for all traits $\mathbf{Y_j}$ *and* $\mathbf{Y_k}$ *and subsets* $\mathbf{Y_S}$*. Then the oracle version of pcgen gives the correct skeleton.*

3.4.2 Faithfulness

For our first faithfulness condition (equation (18)), it suffices to have faithfulness for the network without genetic effects. A necessary (but not sufficient) condition for this is that contributions from different paths do not cancel out (Appendix E.4).

Theorem 3 *Let $P_{Y|G\Gamma}$ denote the joint distribution of $\mathbf{Y_1}, \ldots, \mathbf{Y_p}$ conditional on $\mathbf{U} = \mathbf{G}\Gamma$, the matrix of total genetic effects. Then $\mathbf{Y_j} \perp\!\!\!\perp \mathbf{G}|\{\mathbf{Y_S}\}$ is equivalent with $\mathbf{Y_j} \perp\!\!\!\perp_{P_{Y|G\Gamma}} \mathbf{Y_k}$, and Y_j dsep$(\mathcal{G})$ $G|\{Y_S\}$ is equivalent with Y_j dsep$(\mathcal{G}_\mathcal{Y})$ $Y_k|Y_S$. Consequently,* (18) *holds if*

$$\mathbf{Y_j} \perp\!\!\!\perp_{P_{Y|G\Gamma}} \mathbf{Y_k}|\{\mathbf{Y_S}\} \implies Y_j \; dsep(\mathcal{G}_\mathcal{Y}) \; Y_k|Y_S. \tag{20}$$

The second faithfulness statement (equation (19)) involves d-separation of Y_j and G, and requries that the genetic effects are not collinear. If for example we have $\mathbf{Y_3} = \mathbf{Y_1} + \mathbf{Y_2} + \mathbf{E_3}$, with $\mathbf{Y_1} = \mathbf{G_1} + \mathbf{E_1}$, $\mathbf{Y_2} = \mathbf{G_2} + \mathbf{E_2}$, and $\mathbf{G_2} = -\mathbf{G_1} = \mathbf{G}$, it follows that $\mathbf{Y_3} = \mathbf{E_1} + \mathbf{E_2} + \mathbf{E_3}$. Consequently, because $\mathbf{G_3} = (0, \ldots, 0)^t$, we find that $\mathbf{Y_3}$ and $\mathbf{G} = [\mathbf{G_1}\ \mathbf{G_2}\ \mathbf{G_3}]$ are marginally independent, but in the graph $\mathcal{G}$, the nodes Y_j and G are not d-separated by the empty set, as there are directed paths $G \to Y_2 \to Y_3$ and $G \to Y_1 \to Y_3$. Conversely, if $\mathbf{G_1}$ and $\mathbf{G_2}$ are not perfectly correlated this violation of faithfulness cannot occur. The following theorem shows that marginal independence always implies d-separation. We conjecture (but could not prove) that (19) also holds for non-empty conditioning sets.

Theorem 4 *Suppose we have a GSEM sastifying Assumptions 1-5, and faithfulness for the network without genetic effects, given by* (20)*. Then* (19) *holds for $S = \emptyset$, i.e., marginal independence implies d-separation of Y_j and G_j.*

3.4.3 Properties of the tests

Theorem 2 provides consistency of the oracle version of pcgen, where conditional independence information is available without error. Proving consistency of the sample version is challenging for two reasons. First, the conditional independence tests often rely on approximations (as described in section 2.2.2). In those cases, the tests are based on misspecified models, and instead of $\mathbf{Y_j} \perp\!\!\!\perp \mathbf{Y_k}|\{\mathbf{Y_S}, \mathbf{G}\}$ or $\mathbf{Y_j} \perp\!\!\!\perp \mathbf{G}|\{\mathbf{Y_S}\}$, we test a somewhat different statement. Consequently, a necessary condition for consistency is that the approximation error converges to zero. Second, even without approximation errors, the probability on type-I and type-II errors still needs to converges to zero, with increasing sample size. This is well known for the PC-algorithm with independent Gaussian data ([32]), but more difficult to establish in the presence of genetic effects.

Here we address the first issue, leaving the second for future work. We will focus on

the RC-test [5] for $\mathbf{Y_j} \perp\!\!\!\perp \mathbf{Y_k}|\{\mathbf{Y_S}, \mathbf{G}\}$ and the test for $\mathbf{Y_j} \perp\!\!\!\perp \mathbf{G}|\{\mathbf{Y_S}\}$, assuming $K = ZZ^t$. These tests rely on the conditional distributions $\mathbf{Y_j}|\mathbf{Y_S}$ and $(\mathbf{Y_j}, \mathbf{Y_k})|\mathbf{Y_S}$, for which we made several assumptions regarding their means and covariances. It turns out that the covariances assumptions seem to hold, at least for $K = ZZ^t$. However, the assumption on the conditional means is often violated, also for $K = ZZ^t$. We will now first summarize our results for the covariance, then compute the conditional mean in two examples, and conclude with a discussion on how a better approximation of the conditional mean may be obtained.

Conditional covariance

Theorem 5 *When $K = ZZ^t$, the distribution of $(\mathbf{Y_j}, \mathbf{Y_k})|\mathbf{Y_S}$ has covariance of the form given by* (13)*, i.e., that of a bivariate MTM. Moreover, under faithfulness condition* (18)*, the residual covariance in the MTM is zero if and only if $\mathbf{Y_j} \perp\!\!\!\perp \mathbf{Y_k}|\{\mathbf{Y_S}, \mathbf{G}\}$.*

The idea behind our test for $\mathbf{Y_j} \perp\!\!\!\perp \mathbf{G}|\mathbf{Y_S}$ was that the conditional distribution of $\mathbf{Y_j}$ given $\mathbf{Y_S}$ is of the form $\sigma^2_G(j|S)K + \sigma^2_E(j|S)I_n$, and that $\sigma^2_G(j|S) = 0$ if and only if $\mathbf{Y_j} \perp\!\!\!\perp \mathbf{G}|\{\mathbf{Y_S}\}$. While the former statement always holds, we could prove the latter only for the empty conditioning set. This is because faithfulness is required, which we also established only for $S = \emptyset$ (see Theorem 4)

Theorem 6 *Suppose we have a GSEM as described in Theorem 1, with $K = ZZ^t$ and $\Sigma_G[D, D]$ of full rank (Assumption 5). Then the distribution of $\mathbf{Y_j}|\mathbf{Y_S}$ is of the form $\sigma^2_G(j|S)K + \sigma^2_E(j|S)I_n$, for any conditioning set S. Moreover, under faithfulness condition* (19)*, $\sigma^2_G(j) = \sigma^2_G(j|\emptyset)$ is zero if and only if $\mathbf{Y_j} \perp\!\!\!\perp \mathbf{G}$.*

Conditional mean

In addition to the covariance assumptions, our tests for $\mathbf{Y_j} \perp\!\!\!\perp \mathbf{Y_k}|\{\mathbf{Y_S}, \mathbf{G}\}$ and $\mathbf{Y_j} \perp\!\!\!\perp \mathbf{G}|\{\mathbf{Y_S}\}$ relied on the assumption that the conditional mean for the jth trait was $\mathbf{X}B\gamma_j + \tilde{y}_S\beta^{(j)}_S$, i.e., the sum of the fixed effects on $\mathbf{Y_j}$ and a linear regression on the traits in the conditioning set (see equation (12)). This seems correct in case Y_S contains only 1 trait, which, in the true graph, is the only parent of Y_j. In other situations however, the regression is only an approximation, as shown in the following example.

Suppose that $\mathbf{Y_1} = \mathbf{G_1} + \mathbf{E_1}$ and $\mathbf{Y_2} = \lambda\mathbf{Y_1} + \mathbf{E_2}$, with independent vectors $\mathbf{G_1} \sim N(0, \sigma^2_{G,1}K)$, $\mathbf{E_1} \sim N(\sigma^2_{E,1}I_n)$ and $\mathbf{E_2} \sim N(\sigma^2_{E,2}I_n)$. Then the graph $\mathcal{G}$ is given by

[5] The RG-test for $\mathbf{Y_j} \perp\!\!\!\perp \mathbf{Y_k}|\{\mathbf{Y_S}, \mathbf{G}\}$ requires that the GBLUP $\mathbf{U}^*$ is close to the true matrix of genetic effects ($\mathbf{U}$). Apart from the difficulty of obtaining good estimates of genetic and residual covariances, the quality of this approximation can be easily assessed using expressions for the prediction error variance (see e.g. [39]).

$G \to Y_1 \to Y_2$. There is no edge $G \to Y_2$, although this is not essential for the example. The distributions are given by

$$\begin{aligned} \mathbf{Y_1} &\sim N(0, \Sigma_1) = N\left(0, \sigma^2_{G,1}K + \sigma^2_{E,1}I_n\right), \\ \mathbf{Y_2} &\sim N(0, \Sigma_2) = N\left(0, \lambda^2\sigma^2_{G,1}K + (\lambda^2\sigma^2_{E,1} + \sigma^2_{E,2})I_n\right), \\ Cov(\mathbf{Y_1}, \mathbf{Y_2}) &= \Sigma_{12} = \lambda(\sigma^2_{G,1}K + \sigma^2_{E,1}I_n) = \lambda\Sigma_1. \end{aligned}$$

The conditional mean of $\mathbf{Y_2}$ given $\mathbf{Y_1} = y_1$ is $\mu_{2|1} = \Sigma_{12}\Sigma_1^{-1}y_1 = \lambda y_1$. As expected given the graph, the conditional mean is a simple linear regression on Y_1. However, the conditional mean of $\mathbf{Y_1}$ given $\mathbf{Y_2} = y_2$ equals

$$\mu_{1|2} = \lambda\Sigma_1\left(\lambda^2\Sigma_1 + \sigma^2_E(2)I_n\right)^{-1}y_2,$$

which is a linear transformation, but not a multiple of y_2. Consequently, our models for $\mathbf{Y_j}|\mathbf{Y_S}$ and $(\mathbf{Y_j}, \mathbf{Y_k})|\mathbf{Y_S}$ are sometimes misspecified in terms of the mean, although still correct in terms of covariance, provided $K = ZZ^t$ (Theorems 5 and 6). In these cases, (RE)ML estimates will minimize the Kullback-Leibler divergence with the true distribution, giving the right estimates of genetic variance (or residual covariance), but not necessarily the correct p-values in hypotheses testing.

The conditional mean: improving the approximation

More generally, the conditional mean is a function of the genetic and residual covariances among $\mathbf{Y_j}$ and $\mathbf{Y_S}$. In Appendix E.6 (equation (28)) we derive that $\mathbf{Y_j}|\mathbf{Y_S} = \tilde{y}_S$ has mean $\mu_{j|S} = \mathbf{X}B\gamma_j + \Sigma_{j,S}\Sigma_S^{-1}vec(\tilde{y}_S - \mathbf{X}B\Gamma_S)$. Defining $\eta_{j|S} = 0$ for $S = \emptyset$, we can write $\mu_{j|S} = \mathbf{X}B\gamma_j + \eta_{j|S}$. Consequently, our approximation of the conditional mean effectively models $\eta_{j|S}$ as a linear regression on $\tilde{y}_S$.

This approximation could probably be improved if instead we could obtain good estimates of $\hat{\Sigma}_{j,S}$ and $\hat{\Sigma}_S^{-1}$, and set $\hat{\eta}_{j|S} = \hat{\Sigma}_{j,S}\hat{\Sigma}_S^{-1}vec(\tilde{y}_S - \mathbf{X}\hat{B}\Gamma_S)$. Such estimates however require fitting a MTM for $|S| + 1$ traits, which for large S is statistically and computationally challenging, unless pairwise or other approximations are applied ([40], [41]). Moreover, it seems unclear how this $\hat{\eta}_{j|S}$ should be incorporated in the tests.

4 Discussion

Causal inference for data with random genetic effects is challenging because of the covariance between these effects, and because the usual assumption of independent observations is violated. To address these problems we proposed a model where random genetic effects are part of the causal graph, rather than a nuisance factor that first needs to be eliminated. The resulting distributions and graphs were shown to satisfy the Markov property. This lead us to develop the pcgen algorithm, which tests conditional independence between traits in the presence of genetic effects, and also conditional independence between

a trait and the genetic effects. We showed that the presence of a direct genetic effect can in principle be tested, just like the direct effect of a QTL can be tested. This is of course relative to the observed traits, i.e. for any effect $G \to Y_j$ there may always be an unmeasured trait Z such that $G \to Z \to Y_j$.

In our simulations pcgen outperformed existing approaches. Part of this improvement is due to phenotypic information on replicates, reducing the number of errors in the tests. Another part is due to the improved orientation, which is a consequence of the additional edges $G \to Y_j$. Compared to previous algorithms, pcgen also appeared computationally more efficient: depending on the choice of independence tests and the sparseness of the network, it can analyze around 20-100 traits on a single core, and many more if we limit the maximum size of the conditioning sets, or would parallelize the conditional independence tests.

To a considerable extent this efficiency is due to the use of univariate GBLUP, in the RG-test or the RC-test with prior screening. However, even without using univariate GBLUP, and with direct genetic effects on all traits, the RC-test still has an advantage over existing approaches: by incorporating the genetic effects in the PC-algorithm, we do not need to fit a MTM for all traits simultaneously, but only bivariate models. Our approach also makes genetic network reconstruction feasible with just two traits, and in absence of QTLs, or even no genotypic data at all.

As any causal inference method, pcgen only suggests causal models that are in some sense compatible with the data, and cannot validate the existence of a functional relationship, which is only possible through additional experiments. Because of the required assumptions, the identifiability issues and the uncertainty in the estimated networks, it may be better to speak of algorithms for causal *exploration* than causal *discovery*. At the same time, analysis of one trait given another (e.g. yield given flowering time) is a common and natural thing to do ([15]). From that perspective, pcgen could be seen simply as a tool that performs such analyses systematically, compares them and visualizes the results. pcgen results for different significance levels could then be reported alongside other 'descriptive' statistics like heritability estimates and genetic correlations, suggesting functional hypotheses interesting for future research.

4.1 Data from different experiments

We assumed traits to be measured on the same individuals in the same experiment, with residuals errors arising from biological variation (Assumption 1 in section 2.1.2). In certain applications this assumption can indeed be restrictive, but it seems necessary for any constraint-based causal inference approach. Suppose traits were measured in different experiments, or residual errors would mostly come from measurement errors. Then our model (5) would be replaced by $\mathbf{Y} = \mathbf{X}B + \mathbf{Y_G}\Lambda + \mathbf{G} + \mathbf{E}$, where $\mathbf{Y_G} = \mathbf{X}B + \mathbf{Y}\Lambda + \mathbf{G}$ are the trait values without errors. At first sight this may appear attractive, as appropriate design of the experiments will ensure independent errors, and assumption

3 will be guarantueed. However, the residual covariance of the observed traits will be diagonal as well, instead of the matrix $\Gamma^t \Sigma_E \Gamma$ obtained under assumption 1 (see equations (3), (6) and (8)). However, the residual covariances contained in the latter matrix turned out to be essential for network reconstruction (e.g. Theorem 5). Without assumption 1 we would therefore need to rely completely on the genetic effects. This in turn would require Σ_G to be diagonal, which seems even more restrictive. A relevant alternative approach here is that of invariance causal prediction [42], which infers causal effects that are consistent across several experiments, but still requires all traits to be measured in each experiment.

4.2 Genetically identical replicates and marker-based relatedness matrices

In principle pcgen allows for any type of genetic relatedness. We have however focused on the case of independent genetic effects, for the following reasons:

- Higher power: estimates of (total) genetic variance based on replicates are typically more accurate than marker-based estimates based on genotypic means ([18], [43]). Similary, our simulations suggest that replicates give better tests for (zero) genetic variance and residual covariance.

- Theoretical: when $K = ZZ^t$, the conditional independence statement considered in the RC-test is completely equivalent with $\mathbf{Y_j} \perp\!\!\!\perp \mathbf{Y_k} | \{\mathbf{Y_S}, \mathbf{G}\}$ (Theorem 5), while for other K it is only an approximation. Another argument is the robustness under misspecification: (univariate) broad-sense heritability estimates capture any type of genetic effect, while a model assuming only additive effects can produce strongly biased heritability estimates, if the actual genetic effects are for example partly epistatic. This can be formally shown by computing the Kullback-Leibler divergence between the true distribution and the model under consideration [44]. It seems plausible that this robustness extends to the multivariate models considered here, in particular when direct genetic effects are driven by different sets of QTLs, leading to trait-specific relatedness matrices.

- Computational: the test for $\mathbf{Y_j} \perp\!\!\!\perp \mathbf{G} | \mathbf{Y_S}$ can be based on standard ANOVA, which is many times faster than the LRT for a mixed model. Also the tests for $\mathbf{Y_j} \perp\!\!\!\perp \mathbf{Y_k} | \{\mathbf{Y_S}, \mathbf{G}\}$ are faster when $K = ZZ^t$.

The contributions of different types of genetic effects could in theory be incorporated in the network by introducing multiple genetic nodes, and conditional independence tests based on models that can distinguish these effects. This seems however difficult in practice due to the computational requirements and lack of statistical power.

Finally, we have not investigated the performance of pcgen for unbalanced designs, but it seems likely that small unbalancedness has only a minor effect. A more fundamental challenge seems to be the presence of incomplete blocks or spatial trends ([37], [45]).

4.3 Assessing uncertainty

If one mistakenly rejects the null-hypothesis of conditional independence (type-I error), pcgen leaves the corresponding edge, although it may still be removed at a later stage, with a different conditioning set. If the null-hypothesis is mistakenly accepted (type-II error), a true edge is removed, and will not be recovered. Moreover, it may affect the remaining tests, since d-separation of Y_j and Y_k is only tested given conditioning sets contained in $adj(Y_j)$ or $adj(Y_k)$, where the adjacency sets are defined relative to the current skeleton. This is correct in the oracle version, where some tests may indeed be skipped, but in the sample version pc(gen) may mistakenly skip an essential independence test. See [30] for examples.

Consequently, assessing uncertainty for constraint-based algorithms is difficult, and cannot be achieved by just applying some multiple testing correction to the p-values. To obtain bounds on the expected number of false edges in the skeleton, several authors have used stability selection [46], [47] or other sample-splitting techniques [13], but these are typically over- conservative and require an additional exchangeability assumption ([48], [49]). Alternatively, uncertainty may be assessed using Bayesian methods, which are however computationally very demanding and outside the scope of this work. Moreover, despite the recent progress in Bayesian asymptotics (...), there do not seem to be results yet regarding the correct coverage of posteriors in these models.

4.4 Scope for improving genomic prediction

pcgen can select traits with direct genetic effects, which are the most relevant in genomic selection. More generally, the usefulness of structural models for genomic selection depends on whether there are particular interventions of interest ([16], [17]). Informally speaking, an intervention is an external manipulation that forces some of the traits to have a particular distribution. For example, with a so-called hard intervention on the jth trait, $\mathbf{Y_j}$ is forced to a constant level c, e.g. $c = 0$, when $\mathbf{Y_j}$ is the expression of a gene that is knocked out. The manipulation or truncated factorization theorem [4], [5] can then predict the joint distribution of the system *after* the intervention:

$$p_{\mathbf{Y_j}:=c}(\mathbf{G}, \mathbf{Y_{-j}}) = p(\mathbf{G}) \prod_{j' \neq j} p(\mathbf{Y_{j'}} | pa(\mathbf{Y_{j'}}), \mathbf{G_{j'}}). \tag{21}$$

This is generally different from the distribution

$$p(\mathbf{G}, \mathbf{Y_{-j}} | \mathbf{Y_j} = c), \tag{22}$$

obtained from conditioning on $\mathbf{Y_j} = c$, *prior* to the intervention (see e.g. [24]). In other words, *conditioning* is not the same as *doing* (intervening). An exception occurs however when we intervene on a root node, in which case (21) and (22) are the same.

In absence of interventions on the traits, we can think of genomic prediction in terms of an intervention on the node genotype. Because the latter is a root node by definition, standard genomic prediction methods can in principle have optimal performance ([16]). More specifically, genomic prediction usually involves a regression of a target trait on a number of markers, having either fixed or random effects. In either case, it is only the total effect of genotype on the target trait that matters, not through which other traits this effect passes.

(Near) optimal prediction accuracy however requires that the regression model contains the true distribution (or a good approximation), and a sufficiently accurate estimate of this distribution. We therefore believe that structural models may sometimes be an appealing alternative, especially if the underlying model is highly nonlinear, or when prior physiological knowledge can be incorporated. The extent to which this can really improve accuracy remains to be investigated.

4.5 Open questions and extensions

Although we have shown the Markov property for our model and studied consistency of pcgen, there are a number of open questions left for future work. First, the behavior of our conditional independence tests is not completely understood, and it may be possible to construct better tests, especially for nonlinear structural models. The recent work of [50] seems particularly relevant here. The RC-test proposed here for the conditional independence (A) is exact for $K = ZZ^t$ and certain conditioning sets, but in other cases is misspecified.

A second issue is the consistency of the orientations: while we have shown pcgen's consistency in reconstructing the skeleton, we did not address this for the final CPDAG. This is well known for the PC-algorithm without genetic effects ([5], [32]), but more difficult to establish here, as the class of CPDAGs needs to be restricted to those without errors pointing to G. More generally, orientation constraints seem to be of interest for trait-to-trait relationships as well, e.g. one may require that, *if there is an edge*, the expression of a gene can only affect a metabolite and not the other way round. To the best of our knowledge, current methodology and theory has only considered the forced absence/presence of an edge, leaving the orientation to the algorithm[6]. A final question for future work is whether Theorems 4 and 6 hold for general conditioning sets.

Apart from these open questions, we believe that the idea of explicitly modeling direct genetic effects can be applied more generally. A first generalization would be to replace

[6]The pcalg-package [28] has the addBgKnowledge option to add orientations ('background knowledge') in the estimated CPDAG. This is however only done *after* running PC or a related algorithm, and is only allowed if compatible with the CPDAG.

the PC-algorithm with other constraint-based algorithms, in particular FCI and RFCI ([5], [51]). These have the advantage that the causal sufficiency assumption (no latent variables) can be dropped or considerably weakened. Finally, the presence or absence of direct genetic effects may also be incorporated in invariant causal prediction ([42]), or in Bayesian approaches for genetic network reconstruction.

4.6 Author contributions

WK developed the pcgen algorithm. PB developed the pcgen package, based on code written by WK and the EM-algorithm contributed by MR. WK wrote the paper, with input from FvE, DBK, EW, BY, PB and MR. DBK simulated data with APSIM. PB visualized the estimated networks for the rice, maize and APSIM data. WK proved Theorem 1-2 and WK, EW and MM proved Theorems 3-6.

4.7 Acknowledgements

WK was funded by the Learning from Nature project of the Dutch Technology Foundation (STW), which is part of the Netherlands Organisation for Scientific Research (NWO). We thank Niteen Kadam for providing the rice data, and Xinyou Yin for useful discussions on the interpretation of the resulting networks. Emilie Millet and François Tardieu are acknowledged for their interpretation of the maize data.

A pcgen: implementation details

A.1 The PC-stable algorithm

We first state the PC-stable algorithm of [30], which forms the basis of pcgen. As the original PC-algorithm, PC-stable has a skeleton and orientation stage; the former is described separately as Algorithm 2 below.

Instead of updating the skeleton directly after each conditional independence test, PC-stable only updates the skeleton list-wise, after doing all tests with Y_S of a given size $|S| = s$. More specifically, lines 7-9 in Algorithm 2 make an inventory of the current adjacency sets, which determines which tests of a given size s are to be conducted. In the original PC-algorithm, the skeleton (and hence the adjacency sets) were updated after each individual test, introducing an undesirable order-dependence. Since edges are not directly removed after finding conditional independence, multiple separation sets may be found for a given pair of variables. These may lead to conflicts in the orientation, for example when there are conflicting v-structures $Y_j \to Y_k \leftarrow Y_l$ and $Y_k \to Y_l \leftarrow Y_m$. Whenever possible these conflicts are resolved using the majority rule (line 6 of Algorithm 1). Unresolved conflicts are represented with an undirected edge[7]. In some cases this may lead to partially directed graphs that are not a CPDAG, but a considerable advantage of PC-stable is that it can be parallelized. Finally, also the orientation rules R1-R3 in Algorithm 1 are applied listwise.

[7]Alternatively, conflicts can be represented with a bi-directed edge ($\leftrightarrow$), but the arrowheads do not have a causal interpretation

Algorithm 1 The PC-stable algorithm (taken from [30], and adapted to our notation)

1: **INPUT:** A set of variables V, and an ordering $order(V)$. Conditional independence information among all variables in V (perfect information in the oracle version of the algorithm; estimated from data in the sample version)
2: **OUTPUT:** a graph $\mathcal{G}$ (in the oracle version $\mathcal{G}$ is always a CPDAG; in the sample version it may be only a PDAG, due to conflicts)
3: For all Y_j, Y_k, find the skeleton $\mathcal{C}$ and collections $Sep(Y_j, Y_k)$ of separating sets. To this end, we use PC-skeleton (Algorithm 2 given below).
4: Determine which unshielded triples in the skeleton $\mathcal{C}$ are unambiguous, and orient them using the separating sets:
5: **for all** pairs of nonadjacent variables Y_j, Y_k with common neighbour Y_l (such that (Y_j, Y_l, Y_k) is unambiguous) **do**
6: **if** Y_l is contained in less than half of the separating sets in $Sep(Y_j, Y_k)$ **then**
7: Replace $Y_j - Y_l - Y_k$ in $\mathcal{C}$ by $Y_j \to Y_l \leftarrow Y_k$
8: **end if**
9: **end for**
10: In the resulting PDAG, try to orient as many undirected edges as possible by repeated application of the following rules:
11: **R1** Orient $Y_k - Y_l$ into $Y_k \to Y_l$ whenever there is an arrow $Y_j \to Y_k$ such that Y_j and Y_l are nonadjacent (otherwise a new v-structure is created).
12: **R2** Orient $Y_j - Y_k$ into $Y_j \to Y_k$ whenever there is a chain $Y_j \to Y_l \to Y_k$ (otherwise a directed cycle is created).
13: **R3** Orient $Y_j - Y_k$ into $Y_j \to Y_k$ whenever there are two chains $Y_j - Y_l \to Y_k$ and $Y_j - Y_l \to Y_k$ such that Y_l and Y_l are nonadjacent (otherwise a new v-structure or a directed cycle is created).

Algorithm 2 PC-skeleton (taken from [30], and adapted to our notation)

1: **INPUT:** A set of variables (nodes) V, and an ordering $order(V)$. Conditional independence information among all variables in V (perfect information in the oracle version of the algorithm; estimated from data in the sample version)
2: **OUTPUT:** Estimated skeleton $\mathcal{C}$, collections $Sep(Y_j, Y_k)$ of separating sets (only needed when directing the skeleton afterwards)
3: Form the complete undirected graph $\tilde{\mathcal{C}}$ on the set of variables V.
4: Let $s = -1$; $\mathcal{C} = \tilde{\mathcal{C}}$
5: **repeat**
6: Let $s = s + 1$
7: **for all** variables Y_j in $\mathcal{C}$ **do**
8: Let $a(Y_j) = adj(\mathcal{C}, Y_j)$
9: **end for**
10: **repeat**
11: Select a (new) ordered pair of variables (Y_j, Y_k) that are adjacent in $\mathcal{C}$ and satisfy $|a(Y_j) \setminus \{Y_k\}| \geq s$, using $order(V)$
12: **repeat**
13: Choose a (new) set $S \subseteq a(Y_j) \setminus \{Y_k\}$ with $|S| = s$, using $order(V)$;
14: **if** Y_j and Y_k are conditionally independent given S **then**
15: Delete the edge $Y_j - Y_k$ from $\mathcal{C}$
16: Save S in $Sep(Y_j, Y_k)$ and $Sep(Y_k, Y_j)$
17: **end if**
18: **until** edge $Y_j - Y_k$ is deleted from $\mathcal{C}$, or all $S \subseteq a(Y_j) \setminus \{Y_k\}$ with $|S| = s$ have been considered
19: **until** all ordered pairs of adjacent variables (Y_j, Y_k) in $\mathcal{C}$ with $|a(Y_j) \setminus \{Y_k\}| \geq s$ have been tested for conditional independence
20: **until** all pairs of adjacent variables (Y_j, Y_k) in $\mathcal{C}$ satisfy $|a(Y_j) \setminus \{Y_k\}| < s$.

A.2 Modified orientation rules

pcgen follows the usual orientation rules of the PC-stable algorithm (lines 11-13 in Algorithm 1), except for the following modifications, which are required to avoid arrows pointing into G. Note that Algorithm 1 is written in generic notation with nodes $Y_1, \ldots, Y_p$; in pcgen we have $Y_1, \ldots, Y_{p+1}$, corresponding to $G, Y_1, \ldots, Y_p$.

- in line 5 in Algorithm 1, we skip those triples where Y_k turns out to be G
- after line 9, we orient all remaining undirected edges $G - Y_j$ as $G \to Y_j$.

These changes appear to be necessary, as edges $G \leftarrow Y_j$ cannot be avoided with the `fixedEdges` argument in the pc-function of the pcalg-package ([28]), where one can only

enforce the presence of an edge in the skeleton, but not its orientation. pcalg also has the addBgKnowledge option to add orientations ('background knowledge') in the estimated CPDAG, but this is only done *after* running the PC-algorithm, and is only allowed if compatible with the CPDAG. Here we intend to always enforce the orientation $G \to Y_j$, and include it in the causal inference algorithm.

A.3 The RC-test for $\mathbf{Y_j} \perp\!\!\!\perp \mathbf{Y_k}|\mathbf{G}, \mathbf{Y_S}$

In the RC-test for $\mathbf{Y_j} \perp\!\!\!\perp \mathbf{Y_k}|\mathbf{G}, \mathbf{Y_S}$, the null-hypothesis is that the residual covariance is zero, where the residual covariance is the off-diagonal element of $V_G(jk|S)$, in equations (11) and (13) in the main text (section 2.2.2). The residual likelihood ratio test (RLRT) statistic for this hypothesis is defined as twice the difference in residual log-likelihood between the full and reduced model. As the null-hypothesis is not on the boundary of the parameter space, the distribution of the RLRT is approximately chi-square with 1 degree of freedom. REML-estimates for the full and reduced model are obtained using an EM-algorithm [52]. To improve efficiency, we use the following computational shortcuts:

1. For the reduced model, we compute starting values for the genetic (co) variances and residual variances, using multivariate analysis of variance (MANOVA), and then fit the model using the EM-algorithm described before.

2. For the full model, we take as starting values the estimates found for the reduced model. At each iteration of the EM-algorithm, we compute a preliminary RLRT p-value on the basis of the current restricted log-likelihood and the restricted log-likelihood of the reduced model, and stop the EM-algorithm if the p-value is below the significance threshold. This is possible because the EM-algorithm always increases the likelihood at every iteration, while pcgen only requires an accept/reject decision. Of course, if one wants to know the exact p-value (e.g. to obtain a rough indication of the strength of the causal relationships), the EM-algorithm needs to be run until convergence.

3. We set a maximum number of 50 EM-iterations for the full model and 5 for the reduced model. Given the good starting values this is often sufficient, but occasionally EM would otherwise take very many iterations until convergence. Usually, in these cases, the RLRT statistic obtained with an unrestricted number of EM-iterations is not significant, meaning that stopping EM earlier rarely affects the outcome of the RLRT.

A.4 Testing the genetic covariance

Assuming that the joint distribution of $vec([\mathbf{Y_j}\,\mathbf{Y_k}])$ has covariance $V_G(jk) \otimes K + V_E(jk) \otimes I_n$ (i.e. (13) in the main text, with empty conditioning set), it is sometimes of interest

to test for absence of genetic covariance, i.e., whether the off-diagonal element of $V_G(jk)$ is zero. Although not part of pcgen, this can be tested similar to the RC-test described above.

Again we define a LRT statistic as twice the difference in residual log-likelihood between the full and reduced model, where the latter is restricted to have diagonal $V_G(jk)$. As before the distribution of this LRT is approximately chi-square with 1 degree of freedom, and REML-estimates for the full and reduced model are obtained with the same EM-algorithm used earlier. In the pcgen-package this is implemented in the gencorTest function.

B Simulations

B.1 Simulation setup

We simulate data from model (5) using the following steps, closely resembling the simulations of [32]:

- Given the required number of traits (p) and sparseness of the graph (defined by the parameter p_t below), we first generate the $p \times p$ matrix Λ (see equation (5)), which determines the structural relations among the traits. Λ is simulated using the `randomDAG` function from the R-package `pcalg` ([28]), where edges (i.e. the nonzero elements of Λ) occur with probability $p_t \in (0, 1)$ (for details see the pcalg-documentation and [32], p. 621). The expected number of neighbors of each node is then $p_t(p-1)$. The values of the nonzero coefficients are drawn independently from the uniform distribution on $[0.5, 1]$ and then given a random sign.

- The DAG defined by Λ is now extended with a genetic node G. For a proportion of $p_g \in (0, 1)$ of the traits, we add an edge $G \to Y_j$. The subset of traits D for which there is a direct genetic effect then contains $p \cdot p_g$ traits. These are always chosen to be the traits of highest topological order (in the initial DAG defined by Λ). For example, if $p = 4$ and $p_g = 0.5$, and the initial DAG is $Y_1 \to Y_2 \to Y_3 \to Y_4$, D will consist of Y_1 and Y_2.

- Next, the corresponding genetic variances and covariances in Σ_G are simulated as follows. The genetic variances are drawn independently from a uniform distribution on $[1, 2]$ and random covariances are introduced through random eigenvectors as in [53], using the `genPositiveDefMat` function from the R-package `clusterGeneration`.

- Given the relatedness matrix (K) and the required numbers of genotypes (n) and replicates (r), the direct genetic effects ($\mathbf{G}$) are drawn from the matrix-variate normal distribution with column covariance ZKZ^t and row-covariance Σ_G.

- Similarly, the residual effects ($\mathbf{E}$) are drawn from the matrix-variate normal distribution with column covariance I_{nr} and row-covariance I_p. Although the residual variance is 1 for all traits, the heritability of the traits still varies, as the variances of the direct genetic effects are between 1 and 2. Traits without a direct genetic effect typically have heritability below 0.5.

- Given the matrices $\mathbf{G}$, $\mathbf{E}$ and Λ obtained in the previous steps, $\mathbf{Y}$ is computed using (5). This is done recursively, following the topological ordering of the DAG ([32], p. 621).

B.2 Performance criteria

Following again [32] we compare the true (simulated) CPDAGs and the estimated CPDAGs. Instead of considering the complete CPDAG, we evaluate separately the subgraph defined by Λ (relations among traits) and the direct genetic effects (the subgraph with edges $G_j \to Y_j$, whenever $j \in D$). For both subgraphs we consider the following criteria, again following [32]:

- True Positive Rate (TPR): the number of correct edges (in the estimated skeleton) divided by the total number of true edges (in the true skeleton).
- False Positive Rate (FPR): the number of incorrect edges (in the estimated skeleton) divided by the total number of true gaps (in the true skeleton).
- True Discovery Rate (TDR): the number of edges in the estimated graph that are correct (i.e. exist in the true skeleton) divided by the total number of edges in the estimated graph.
- Structural Hamming Distance (SHD): the number of edge deletions, additions and flips required to transform the estimated CPDAG into the true (simulated) CPDAG. See also [54].

All criteria were computed using functions from the pcalg-package (TPR and FPR using the `compareGraphs` function, and the SHD using the function `shd`).

B.3 Simulation results for alternative methods

B.4 Simulation with APSIM

We used the crop-growth model APSIM to simulate 12 traits ($\mathbf{Y_1}, \ldots, \mathbf{Y_{12}}$) for an existing population of 199 wheat genotypes, characterized with 3,035 SNPs with minor allele frequency larger than 0.05 (details in [55]). APSIM simulations were carried out in Emerald, during 1993, which corresponds to a severe drought environment. Simulation settings were the same as in [56] and genotype-specific parameters had the ranges specified in [55]. The wheat panel was assumed to segregate for 7 of the APSIM parameters, which we refer to as the primary traits. These are the only ones for which there are direct genetic effects. The 7 primary traits ($\mathbf{Y_1}, \ldots, \mathbf{Y_7}$) are a subset of those used in [55], and were chosen because they have an important impact on grain yield, as shown by global sensitivity analysis [56]. Apart from the primary traits there are 4 intermediate traits, each of them depending on some of the primary traits, and sometimes some of the other intermediate traits. The final trait is yield, which depends on 3 of the intermediate traits. Acronyms for all 12 traits are given in Table 3. For each genotype three replicates were simulated.

The direct genetic effects on the 7 primary traits were simulated as the sum of 300 additive QTL-effects. Different samples of 300 SNPs were used for each trait, and each

	$\mathcal{G}_{\mathcal{Y}}$				$G \to Y_j$		
	TPR	FPR	TDR	SHD	TPR	FPR	TDR
Scenario 1 ($p = 4$)	($p_t = 1/3$)				($p_g = 0.5$)		
pcgen-RC	0.607	0.012	0.960	0.568	0.986	0.020	0.994
pcgen-RG-uni	0.607	0.036	0.906	0.670	0.984	0.022	0.994
pcgen-screening	0.603	0.008	0.982	0.548	0.986	0.020	0.994
pcRes-multi-R	0.605	0.161	0.626	2.052			
pcRes-uni-R	0.607	0.036	0.906	1.362			
pcRes-multi-RK	0.570	0.192	0.624	2.060			
pcRes-uni-RK	0.608	0.029	0.920	1.330			
pcRes-multi-K	0.465	0.432	0.388	3.212			
pcRes-uni-K	0.595	0.088	0.747	1.668			
pc-GBLUP	0.536	0.490	0.375	3.472			
Scenario 2 ($p = 4$)	($p_t = 0.5$)				($p_g = 0.5$)		
pcgen-RC	0.814	0.042	0.980	1.236	0.966	0.041	0.993
pcgen-RG-uni	0.820	0.074	0.951	1.320	0.965	0.043	0.992
pcgen-screening	0.805	0.034	0.990	1.238	0.966	0.042	0.992
pcRes-multi-R	0.824	0.149	0.835	2.640			
pcRes-uni-R	0.820	0.074	0.951	2.274			
pcRes-multi-RK	0.786	0.242	0.787	2.892			
pcRes-uni-RK	0.821	0.063	0.957	2.230			
pcRes-multi-K	0.666	0.466	0.603	3.778			
pcRes-uni-K	0.786	0.098	0.893	2.522			
pc-GBLUP	0.752	0.560	0.586	4.124			

Table 2. Performance of pcgen and residuals-based approaches, averaged over 500 simulated data-sets, for scenarios 1 and 2.

effect was sampled from a trait-specific Gamma distribution. The shape and rate of this distribution were obtained by fitting a Gamma distribution to empirical additive effects estimated in a GWAS analysis of real phenotypes observed for this population in the Australian wheat belt. For the phenology-related traits SV and SP we set $k = 0.7$ and $b = 13.6$, and $k = 1.3$ and $b = 13.6$ for the other primary traits. We then added Gaussian noise, to get a heritability of 0.9 for all primary traits.

The secondary traits ($\mathbf{Y_8}, \ldots, \mathbf{Y_{11}}$) and yield ($\mathbf{Y_{12}}$) were simulated by running a dynamic model from time zero to time T, the time-point at which all traits are observed:

$$\mathbf{Y_j}(T) = \int_0^T f_z(pa(\mathbf{Y_j}(t))dt, \tag{23}$$

where $pa(\mathbf{Y_j})(t)$ are the values of the 'parent traits' at time-point t, and z represents a set of fixed parameters, specific for the environment under consideration. The form of f_z

trait name / category	acronym
Primary traits	
Radiation use efficiency	RUE
Number of grains per gram of stem at flowering	NGF
Maximum grain size	MGS
Lower limit for water uptake	LL
Sensitivity to photoperiod	SP
Sensitivity to vernalization	SV
Thermal time required to reach floral initiation	TFI
Secondary traits and yield	
flowering time	FT
grain weight	GW
grain number	GN
biomass	BM
yield	Y

Table 3. Acronyms for 12 traits simulated using APSIM.

is detailed in .... The sets of parental traits stay the same over time, and therefore define the summary graph given in Figure 3 (main text).

C Rice trait acronyms and networks for different significance thresholds

Traits	Trait	acronym	Unit
(A) Shoot morphological traits			
	Plant height	PHT	cm
	Tiller number	TN	$plant^{-1}$
	Total leaf area	TLA	$m^2\ plant^{-1}$
	Specific leaf area	SLA	$m^2\ g^{-1}$
(B) Physiological traits			
	Cumulative water transpiration	CWT	$kg\ plant^{-1}$
	Water use efficiency	WUE	$g\ kg^{-1}$
(C) Root anatomical traits			
	Total root length	TRL	$m\ plant^{-1}$
	Root length (RL) with diameter class		
	RL_0-0.5	RL005	$m\ plant^{-1}$
	RL_0.5-1.0	RL0510	$m\ plant^{-1}$
	RL_1.0-1.5	RL1015	$m\ plant^{-1}$
	RL_1.5-2.0	RL1520	$m\ plant^{-1}$
	RL_2.0-2.5	RL2025	$m\ plant^{-1}$
	RL_2.5-3.0	RL2530	$m\ plant^{-1}$
	RL_3.0-3.5	RL3035	$m\ plant^{-1}$
	RL_3.5	RL35	$m\ plant^{-1}$
	Maximum root length	MRL	cm
	Surface area	SA	$cm^2\ plant^{-1}$
	Root volume	RV	$cm^3\ plant^{-1}$
	Average root thickness	ART	mm
	Specific root length	SRL	m g^{-1}
	Total root weight density	TRWD	$g\ cm^{-3}$
	Root length per unit leaf area	RLLA	$m\ m^{-2}$
(D) Root anatomical traits			
	Root diameter	RD	μm
	Cortex diameter	CD	μm
	Stele diameter	SD	μm
	Late metaxylem	LMXD	μm
	Late metaxylem	LMXN	μm
	Stele diameter in proportion of root diameter	SD:RD	%
(E) Dry matter traits			
	Leaf weight	LW	$g\ plant^{-1}$
	Stem weight	SW	$g\ plant^{-1}$
	Root weight	RW	$g\ plant^{-1}$
	Total weight	TW	$g\ plant^{-1}$
	Root:shoot ratio	RS	–
	Leaf weight ratio	LWR	–
	Stem weight ratio	SWR	–

Table 4. Trait acronyms used in Figure 5; table taken from [38]. Leaf weight (LW) was removed prior to our analysis, as it appeared to be an exact linear combination of 3 of the other traits (TW - RW - SW, i.e. total weight minus root weight minus shoot weight).

D Overview of notation, acronyms and commands in pcgen

Symbol	Meaning / Category	Introduced in section
	Other (exgenous) fixed effects in SEM	
B		
X		
	Path coefficients in linear (G)SEM	
Λ		
Γ		
γ_j		
	Graphs	
$\mathcal{G}$		
G		
Y_j		
$\mathcal{G}_{\mathcal{Y}}$		
d-separation		
$dsep(\mathcal{G}$		
	Genetic effects	
$\mathbf{G}$		
$\mathbf{g_i}$		
$\mathbf{G_j}$		
$\mathbf{U}$		
$\mathbf{u_i}$		
$\mathbf{U_j}$		
	Residual effects	
$\mathbf{E}$		
$\mathbf{e_i}$		
$\mathbf{E_j}$		
	Traits	
	Genetic relatedness	

Σ_G, $\sigma^2_{G,j}$, Σ_E, n, m, r, p, q, $\mu_{j|S}$

GBLUP, RC-test, MTM, SEM, GSEM, DAG, CPDAG

Abbreviation	Description	Target
pcgen-RC	pcgen with the RC-test	$\mathcal{G}$
pcgen-RG-uni	pcgen with the RG-test, based on univariate GBLUP	$\mathcal{G}$
pcgen-RG-multi	pcgen with the RG-test, based on multivariate GBLUP	$\mathcal{G}$
pcgen-screening	pcgen with the RC-test, starting with the skeleton obtained from pcRes-uni-R (see section 2.6)	$\mathcal{G}$
pcRes-uni-R	pcRes based on univariate GBLUP, using replicates	$\mathcal{G}_{\mathcal{Y}}$
pcRes-uni-RK	pcRes based on univariate GBLUP, using replicates + GRM	$\mathcal{G}_{\mathcal{Y}}$
pcRes-uni-K	pcRes based on univariate GBLUP, using GRM	$\mathcal{G}_{\mathcal{Y}}$
pcRes-multi-R	pcRes based on multivariate GBLUP, using replicates	$\mathcal{G}_{\mathcal{Y}}$
pcRes-multi-RK	pcRes based on multivariate GBLUP, using replicates + GRM	$\mathcal{G}_{\mathcal{Y}}$
pcRes-multi-K	pcRes based on multivariate GBLUP, using GRM	$\mathcal{G}_{\mathcal{Y}}$
pc-GBLUP	pc(stable) applied to multivariate GBLUP, similar to [13]	$\mathcal{G}_{\mathcal{Y}}$

Table 5. Overview of the algorithms available in the pcgen package, for reconstructing either $\mathcal{G}$ (the complete graph) or $\mathcal{G}_{\mathcal{Y}}$ (the subgraph of trait-to-trait relations). The required commands in the pcgen-package are given in Table 6 in Appendix D.

E Faithfulness, conditional distributions and proofs of Theorems 1-6

E.1 Overview of graph theoretic definitions

The following definitions can be found in a large number of books and articles on graph theory, graphical models and causal inference; see for example [57], [4], [5] and [32].

- Given different nodes Y_j and Y_k, a *path* from Y_j to Y_k is a sequence of edges connecting Y_j and Y_k. When all edges are directed and pointing towards Y_k, we have a *directed path*. An *undirected path* or *non-directed path* is a path that is not directed.

- A *cycle* is a path from Y_j to Y_k with an additional edge between Y_j and Y_k. A *directed cycle* is a directed path from Y_j to Y_k together with a directed edge $k \to j$.

- A *directed acyclic graph* (DAG) is a directed graph without cycles. When a graph underlying a SEM is a DAG, the SEM is said to be recursive.

- $pa(j)$ is the set of nodes Y_k for which is a directed edge $k \to j$; in this case Y_j is a *child* of Y_k and Y_k is a *parent* of Y_j. The nodes Y_j and Y_k are *adjacent* if there is an edge between them.

- If in a DAG $\mathcal{G}$ there is a directed path from Y_j to Y_k, Y_j is an ancestor of Y_k, and Y_k is a descendant of Y_j.

Abbreviation	Commands in pcgen-package
pcgen-RC	pcgen(d,..., use.res=F)
pcgen-RG-uni	C ← cor(getResiduals(d,..., cov.method='uni')) pcgen(d,..., use.res=T, res.cor = C) equivalently: pcgenFast(..., use.res=T)
pcgen-RG-multi	C ← cor(getResiduals(d,..., cov.method='us')) pcgen(d,..., use.res=T, res.cor = C)
pcgen-screening	pcgenFast(d,..., use.res=F)
pcRes-uni-R	pcRes(d,..., cov.method='uni', use.GBLUP=F)
pcRes-uni-RK	pcRes(d,..., cov.method='uni', use.GBLUP=F, K=A)
pcRes-uni-K	pcRes(m,..., cov.method='uni', use.GBLUP=F, K=A)
pcRes-multi-R	pcRes(d,..., cov.method='us', use.GBLUP=F)
pcRes-multi-RK	pcRes(d,..., cov.method='us', use.GBLUP=F, K=A)
pcRes-multi-K	pcRes(m,..., cov.method='us', use.GBLUP=F, K=A)
pc-GBLUP	pcRes(m,..., cov.method='us', use.GBLUP=T, K=A)

Table 6. R-commands needed to run the different algorithms, with the package pcgen. T stands for TRUE, F for FALSE. The first argument is the required phenotypic data-frame (suffStat = d (replicates) or suffStat = m (genotypic means)). The dots represent generic arguments (e.g. alpha and m.max, which define the significance threshold and the maximum size of the conditioning sets). cov.method determines whether univariate (uni) or multivariate (us) GBLUP is to be used ('us' stands for unstructured, as opposed to e.g. factor analytic models, which have not yet been implemented). All algorithms involving GBLUP use the residuals (use.GBLUP = F), except the genomic network similar to [13] (pc-GBLUP, with use.GBLUP = T). Finally, A is a genetic relatedness matrix, which can be included by putting K=A; otherwise the default is used (K = NULL, in which case replicates are required).

- If, for a given path, two directed edges point into the same node, the latter is a *collider*. For example, given the DAG $A \to C \leftarrow B$, C is a collider on the (only) path between A and B. In all other cases ($A \leftarrow C \to B$, $A \to C \to B$ and $A \leftarrow C \leftarrow B$), C is a *non-collider*. Several different paths can pass through a node, and being a (non-)collider is always relative to the path.

- In a DAG, a *v-structure* or *immorality* is a collection of three nodes (say A, B and C), such that there are directed edges $A \to B$ and $C \to B$ but no edge between A and C. In this case B is an *unshielded* collider; otherwise it is a *shielded* collider. Similarly, in an undirected graph, A, B and C form an *unshielded triple* if there are edges $A - B$ and $C - B$ but no edge $A - C$.

- The *skeleton* of a (partially) directed graph is the undirected graph obtained after removing all arrowheads.

- Given a directed graph $\mathcal{G}$, two nodes A and B, and a (possibly empty) subset of nodes S not containing A and B, a path between A and B is *blocked* by S if at least one of the following two conditions holds: (i) there exists a collider on the path which is not in S, and also none of its descendants are in S. (ii) there exists a non-collider on the path that is in S.

- Nodes A and B are *d-separated* by a set S if S blocks all paths from A to B.

- Given disjoint sets U, V and S (U and V should be non-empty), U and V are *d-separated* by S if S blocks all paths from Y_j to Y_k, for all nodes $j \in U$ and $k \in V$.

- Two DAGs are *equivalent* if they have the same skeleton and the same v-structures.

- An equivalence class of DAGs is a set containing all DAGs that are equivalent to one another. Any DAG in the class can be used to represent the class; however an equivalence class can also be represented by a *completed partially directed acyclic graph* (CPDAG). Following the formulation of [32], a *partially directed acyclic graph* (PDAG) is 'a graph where some edges are directed and some are undirected and one cannot trace a cycle by following the direction of directed edges and any direction for undirected edges'. A PDAG is a CPDAG if (a) every directed edge in the PDAG exists in all DAGs in the equivalence class it represents (b) for every undirected edge $j - k$ in the PDAG, the equivalence class contains at least one DAG with $j \to k$ and at least one with with $k \to j$. Chickering [58] showed that CPDAGs represent equivalence classes uniquely. For example, given a skeleton $A - B - C$, there is one equivalence class containing the three DAGs $A \to B \to C$, $C \to B \to A$ and $A \leftarrow B \to C$, and one equivalence class with only one DAG ($A \to B \leftarrow C$).

E.2 The matrix Γ expressed as a function of path coefficients

Let $\mathcal{G}_\mathcal{Y}$ denote the DAG over the nodes $Y_1, \ldots, Y_p$, with edges defined by Λ. For each $j \in \{1, \ldots, p\}$, let U_j denote the union of the set $\{Y_j\}$ and the set of root traits (i.e. those without parents in $\mathcal{G}_\mathcal{Y}$) for which there is a directed path towards Y_j. For all $j, k \in \{1, \ldots, p\}$, let Π_{jk} denote the set of all directed paths from Y_j to Y_k. For $k = j$, Π_{jj} contains only the empty path from Y_j to itself. For any directed path π from Y_j to Y_k, let $L(\pi)$ denote the product of the corresponding path coefficients as given by Λ; for the empty path we define $L(\pi) = 1$.

Using these definitions, we can decompose the variance of a trait into contributions from different ancestors, as well as its own error variance. To this end, we follow [5] and define the $p \times 1$ column vector γ_j with elements ($l = 1, \ldots, p$)

$$\begin{aligned} \gamma_{j,l} &= \sum_{\pi \in \Pi_{lj}} L(\pi) \quad \text{if } Y_l \in U_j \\ &= 0 \qquad\quad \text{otherwise.} \end{aligned} \tag{24}$$

E.3 The covariance between $\mathbf{Y_j}$ and $\mathbf{Y_k}$ as function of path coefficients

Since $\mathbf{Y_j} = \mathbf{X}B\gamma_j + \mathbf{G}\gamma_j + \mathbf{E}\gamma_j$ (equation (10) in the main text), the covariance between the $n \times 1$ vectors $\mathbf{Y_j}$ and $\mathbf{Y_k}$ can be written in terms of γ_j and γ_k:

$$Cov(\mathbf{Y_j}, \mathbf{Y_k}^t) = E[(\mathbf{Y_j} - \mathbf{X}B\gamma_j)(\mathbf{Y_k} - \mathbf{X}B\gamma_k)^t] = (\gamma_j^t \Sigma_G \gamma_k)K + (\gamma_j^t \Sigma_E \gamma_k) I_n, \tag{25}$$

for all $j, k \in \{1, \ldots, p\}$. Consequently, we can express the genetic and residual covariance between traits in terms of quadratic forms, involving Σ_G, Σ_E and the path coefficients.

As a special case of (25), it follows that without random genetic effects,

$$Cov(\mathbf{Y}[i, j], \mathbf{Y}[i, k]) = \gamma_j^t \Sigma_E \gamma_k$$

is the covariance between the jth and kth trait, for each individual i. See also [5] (Lemma 3.1.6), or [59] (Appendix 2). Using standard expressions for multivariate Gaussian distributions, this implies that

$$Cov(\mathbf{Y}[i, j], \mathbf{Y}[i, k] \mid \mathbf{Y}[i, S]) = \gamma_j^t \Sigma_E \gamma_k - (\gamma_j^t \Sigma_E \Gamma_S)(\Gamma_S^t \Sigma_E \Gamma_S)^{-1}(\Gamma_S^t \Sigma_E \gamma_k). \tag{26}$$

E.4 The path coefficients condition

It is well known that faithfulness is violated when contributions from different paths cancel out. For example, in the SEM defined by $Y_1 \to Y_2$, $Y_1 \to Y_3$ and $Y_2 \to Y_3$, with respective path coefficients 1, 1 and -1, Y_1 and Y_3 are marginally independent but not d-separated. Conversely, when faithfulness holds, we know that such cancellations cannot occur, and that the sum in (24) is never zero, i.e. $\gamma_{j,l} = 0$ only for $Y_l \notin U_j$. We will refer to this as the **path coefficients condition**.

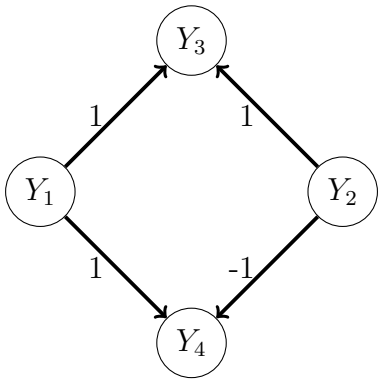


Figure 6. An example of a SEM where faithfulness does not hold, because the contributions to the covariance from the treks $Y_3 \leftarrow Y_1 \rightarrow Y_4$ and $Y_3 \leftarrow Y_2 \rightarrow Y_4$ cancel out. If Y_1 and Y_2 are Gaussian with equal (error) variances, it follows that $Cov(Y_3, Y_4) = Cov(Y_1 + Y_2 + E_3, Y_1 - Y_2 + E_4) = Cov(Y_1 + Y_2, Y_1 - Y_2) = 0$. Consequently, Y_3 and Y_4 are marginally independent, but not d-separated by the empty set.

E.5 The path coefficients condition and faithfulness

The path coefficients condition is a necessary but not a sufficient condition for faithfulness. First, faithfulness can also be violated when contributions from different paths cancel out when summing over a subset of all directed paths; see Example 2.10 in [60]. Second, it is not only the contributions of directed paths that should not cancel out, but also those of treks. A trek between Y_j and Y_k is any path between these nodes without a collider ([5]). Every trek consists of 2 directed paths, starting at the *source* of the trek, and going towards Y_k and Y_k. One of these can be the empty path; hence each directed path is also a trek. Figure 6 provides an example where contributions from different treks cancel out, leading to non-faithfulness.

Another necessary condition for faithfulness is that all error variances are strictly positive. Figure 7 provides an example of non-faithfulness due to a zero error variance. An extended version of the path coefficients condition (involving sums over subset of treks) together with strictly positive error variances may be sufficient for faithfulness, but we could not find such a result in the literature. However, from (26) it follows that for Gaussian linear SEM, faithfulness is equivalent with

$$\gamma_j^t \Sigma_E \gamma_k - (\gamma_j^t \Sigma_E \Gamma_S)(\Gamma_S^t \Sigma_E \Gamma_S)^{-1}(\Gamma_S^t \Sigma_E \gamma_k) = 0 \implies Y_j \text{ and } Y_k \text{ d-separated by } Y_S. \quad (27)$$

E.6 Conditional means and covariances

Using the notation $[, S]$ to select the columns corresponding to S, and $[S_1, S_2]$ to select both rows and columns, it follows from (8) that $\mathbf{Y_j}|\mathbf{Y_S} = \tilde{y}_S$ is multivariate normal with

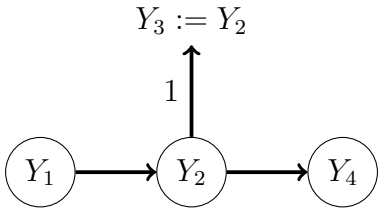


Figure 7. An example of a SEM where faithfulness does not hold, because the variance of the error E_2 is zero. The random variables Y_4 and Y_1 are conditionally independent given Y_3, but in the graph, the nodes Y_4 and Y_1 are not d-separated by Y_3.

mean and covariance

$$\begin{aligned} \mu_{j|S} &= (\mathbf{X}B\Gamma)[,j] + \Sigma_{j,S}\Sigma_S^{-1} vec(\tilde{y}_S - (\mathbf{X}B\Gamma)[,S]) \\ &= \mathbf{X}B\gamma_j + \Sigma_{j,S}\Sigma_S^{-1} vec(\tilde{y}_S - \mathbf{X}B\Gamma_S), \end{aligned} \tag{28}$$

$$\Sigma_{j|S} = \Sigma_j - \Sigma_{j,S}\Sigma_S^{-1}\Sigma_{j,S}^t, \tag{29}$$

where

$$\begin{aligned} \Sigma_{j,S} &= (\Gamma^t\Sigma_G\Gamma)[j,S] \otimes K + (\Gamma^t\Sigma_E\Gamma)[j,S] \otimes I_n & (30)\\ &= (\gamma_j^t\Sigma_G\Gamma_S) \otimes K + (\gamma_j^t\Sigma_E\Gamma_S) \otimes I_n, & (31)\\ \Sigma_S &= (\Gamma_S^t\Sigma_G\Gamma_S) \otimes K + (\Gamma_S^t\Sigma_E\Gamma_S) \otimes I_n, & (32)\\ \Sigma_j &= (\Gamma^t\Sigma_G\Gamma)[j,j]K + (\Gamma^t\Sigma_E\Gamma)[j,j]I_n = (\gamma_j^t\Sigma_G\gamma_j)K + (\gamma_j^t\Sigma_E\gamma_j)I_n. & (33) \end{aligned}$$

The matrices Σ_j, Σ_S and $\Sigma_{j,S}$ are the variance-covariance matrix of respectively $vec(\mathbf{Y_j}) = \mathbf{Y_j}$ and $vec(\mathbf{Y_S})$, and the covariance between $\mathbf{Y_j}$ and $vec(\mathbf{Y_S})$.

From equation (8) we also obtain the conditional distribution

$$\begin{aligned} vec([\mathbf{Y_j}\ \mathbf{Y_k}]) \mid \mathbf{Y_S} = \tilde{y}_S &\sim N\left(\begin{pmatrix} \mu_{j|S} \\ \mu_{k|S} \end{pmatrix}, \Sigma_{jk|S}\right) \\ &= N\left(\begin{pmatrix} \mu_{j|S} \\ \mu_{k|S} \end{pmatrix}, \Sigma_{jk} - \Sigma_{jk,S}\Sigma_S^{-1}\Sigma_{jk,S}^t\right), \end{aligned} \tag{34}$$

where $\mu_{j|S}$ and $\mu_{k|S}$ are as in equation (28), and Σ_{jk} is the $2n \times 2n$ block matrix with diagonal blocks Σ_j and Σ_k (defined as in (33)), and off-diagonal blocks $(\gamma_j^t\Sigma_G\gamma_k)K + (\gamma_j^t\Sigma_E\gamma_k)I_n$. Similarly, given the $p \times 2$ matrix Γ_{jk} with columns γ_j and γ_k, it follows that

$$\Sigma_{jk,S} = (\Gamma_{jk}^t\Sigma_G\Gamma_S) \otimes K + (\Gamma_{jk}^t\Sigma_E\Gamma_S) \otimes I_n$$

is the $2n \times |S|n$ covariance between $vec([\mathbf{Y_j}\ \mathbf{Y_k}])$ and $vec(\mathbf{Y_S})$.

E.7 Covariance structure of the conditional distributions

When $K = ZZ^t$ is block-diagonal, with $r \times r$ blocks of ones on the diagonal, then for any nonnegative constants c and d,

$$(cI_n + dK)^{-1} = c^{-1}I_n - \frac{1}{c^2(1/d + r/c)}K.$$

Hence, the inverse of $(cI_n + dK)$ is again a linear combination of I_n and K. This follows from the Woodbury identity ([61] and [62])

$$(A + CBC^t)^{-1} = A^{-1} - A^{-1}C(B^{-1} + C^tA^{-1}C)^{-1}C^tA^{-1}, \tag{35}$$

with $A = cI_n$, $B = dI_m$ and $C = Z$. In addition we have $Z^tZ = rI_m$, and therefore $K^2 = rK$. Consequently, any product of matrices of the form $(cI_n + dK)$ or their inverse is a linear combination of I_n and K.

From this it follows that when $K = ZZ^t$, $\Sigma_{j|S}$ in (29) is of the form $\sigma^2_G(j|S)K + \sigma^2_E(j|S)I_n$. Similarly, it follows that $\Sigma_{jk|S}$ in (11) is of the form

$$V_G(jk|S) \otimes K + V_E(jk|S) \otimes I_n,$$

for some 2×2 matrices $V_G(jk|S)$ and $V_E(jk|S)$.

E.8 Proof of Theorem 1

Pearl ([4], p. 51) showed that under quite general assumptions, structural equation models satisfy the global Markov property, which means that d-separation in the graph implies conditional independence. It turns out that in our case, the required assumption of independent errors applies to the p error variables and not to G. The intuition behind this is that G is not just an additional error node, but part of the causal graph, and we can always distinguish between residual (co)variance and genetic (co)variance. We now give the proof of Theorem 1, which only requires minor modifications of the proof given by Pearl for the case without the genetic effects.

Let $\mathcal{G}_E$ denote the extended graph, obtained by adding the error variables, i.e. for traits $j = 1, \ldots, p$ we add the node E_j and an edge $E_j \to Y_j$. We first show that the local Markov property holds for $\mathcal{G}_E$, i.e. for any variable $Z \in \{G, Y_1, \ldots, Y_p, E_1, \ldots, E_p\}$, Z is conditionally independent of its non-descendants given its parents. This is obvious for $Z \in \{G, E_1, \ldots, E_p\}$; we now consider Y_j. In $\mathcal{G}_E$, the set of parents of Y_j is $pa(Y_j) \cup \{E_j\}$, where $pa(Y_j)$ contains G if $j \in D$. By construction, Y_j is entirely determined by $pa(Y_j) \cup \{E_j\}$, and constant conditional on these variables. Consequently, given $pa(Y_j) \cup \{E_j\}$, it is independent of any E_k $(k \neq j)$, and of any Y_k that it is a non-descendant of Y_j (Note that if $G \notin pa(Y_j)$, Y_j is indeed conditionally independent of any non-descendant; if $G \in pa(Y_j)$, G cannot be the non-descendant because it is already in the conditioning

set). Therefore the local Markov property holds for $\mathcal{G}_E$. By Lemma E.1 below, we find that also the Markov factorization property holds for $\mathcal{G}_E$, since for any distribution having a density it is equivalent with the local and global Markov properties. Given the Markov factorization property for $\mathcal{G}_E$ and the fact that $f(e_1, \ldots, e_p) = \prod_{j=1}^{p} f_j(e_j)$, we can just integrate out the e_j, and obtain the Markov factorization property for $\mathcal{G}$.

Markov properties

The following lemma is taken from [57] (p. 51), and reformulated with somewhat less general conditions, which however suffice for our purpose.

Lemma E.1 *Let P be the joint distribution of random variables $(Y_1, \ldots, Y_p)$, having a density f, and let $\mathcal{G}$ be a DAG on these variables. The following properties are equivalent:*

- *The Markov factorization property: given the parents pa_j of each x_j, the joint density (f) can be decomposed as*

$$f(y_1, \ldots, y_p) = \prod_{j=1}^{p} f_j(y_j | pa_j),$$

  where the f_j are the conditional densities.

- *The local Markov property: any variable is conditionally independent of its non-descendants, given its parents.*

- *The global Markov property: for all disjoint sets $U, V, S \subset \{Y_1, \ldots, Y_p\}$, d-separation of U and V by S in the graph $\mathcal{G}$ implies conditional independence of U and V given S. In contrast to U and V, the conditioning set S may be empty here. A definition of d-separation is given in S1.*

E.9 Proof of Theorems 3 and 5

We first prove Theorem 3, by showing the equivalence of the left- and right hand sides of (18) and (20). The d-separation statements on the right hand sides are equivalent, as G can never be a (descendant of a) collider. Also the left hand sides ($\mathbf{Y_j} \perp\!\!\!\perp \mathbf{Y_k} | \{\mathbf{Y_S}, \mathbf{G}\}$ and $\mathbf{Y_j} \perp\!\!\!\perp_{P_{Y|G\Gamma}} \mathbf{Y_k} | \{\mathbf{Y_S}\}$) are equivalent, since

$$p_{Y|G\Gamma}(y_j, y_k | y_S) = p(y_j, y_k | y_S, G\Gamma) = p(y_j | y_S, G\Gamma) p(y_k | y_S, G\Gamma) = p_{Y|G\Gamma}(y_j | y_S) p_{Y|G}(y_k | y_S).$$

For Theorem 5 we make the additional assumption that $K = ZZ^t$, $Z = I_m \otimes (1, \ldots, 1)^t$ being the $mr \times m$ design matrix for r replicates of m genotypes in a balanced design (with $mr = n$). The first part of Theorem 5 then follows from the results in Appendix E.7. For the second part, first recall the equivalence of $\mathbf{Y_j} \perp\!\!\!\perp \mathbf{Y_k} | \{\mathbf{Y_S}, \mathbf{G}\}$ and $\mathbf{Y_j} \perp\!\!\!\perp_{P_{Y|G\Gamma}}$

$\mathbf{Y_k}|\{\mathbf{Y_S}\}$. Because of the Gaussianity and the assumed faithfulness, the latter conditional independence is equivalent with

$$\gamma_j^t \Sigma_E \gamma_k - (\gamma_j^t \Sigma_E \Gamma_S)(\Gamma_S^t \Sigma_E \Gamma_S)^{-1}(\Gamma_S^t \Sigma_E \gamma_k) = 0, \tag{36}$$

where we used (26).

Next we consider the conditional distribution of $vec([\mathbf{Y_j}\ \mathbf{Y_k}])|\mathbf{Y_S} = \tilde{y}_S$ given in (11), whose covariance is the $2n \times 2n$ block matrix $\Sigma_{jk} - \Sigma_{jk,S}\Sigma_S^{-1}\Sigma_{jk,S}^t$. All $n \times n$ blocks are a linear combinations of K and I_n, and it suffices to show that the coefficient of I_n in the off-diagonal blocks is zero if and only if (36) holds. We recall from (32) that

$$\Sigma_S = (\Gamma^t \Sigma_G \Gamma)[S,S] \otimes K + (\Gamma^t \Sigma_E \Gamma)[S,S] \otimes I_n = (\Gamma_S^t \Sigma_G \Gamma_S) \otimes K + (\Gamma_S^t \Sigma_E \Gamma_S) \otimes I_n.$$

Using the Woodbury identity (equation (35)) with $A = V_E \otimes I_n$, $B = V_G \otimes I_m$ and $C = I_p \otimes Z$, it follows that for any positive (semi) definite $p \times p$ matrices V_G and V_E, we have

$$(V_G \otimes K + V_E \otimes I_n)^{-1} = (V_E^{-1} \otimes I_n) - \left(V_E^{-1}(V_G^{-1} r V_E^{-1})^{-1} V_E^{-1}\right) \otimes K. \tag{37}$$

Setting $V_G = \Gamma_S^t \Sigma_G \Gamma_S$, $V_E = \Gamma_S^t \Sigma_E \Gamma_S$ and $A = V_E^{-1}(V_G^{-1} r V_E^{-1})^{-1} V_E^{-1}$, it follows that

$$\Sigma_S^{-1} = (\Gamma_S^t \Sigma_E \Gamma_S)^{-1} \otimes I_n - A \otimes K, \tag{38}$$

Combining this with the expressions for Σ_{jk} and $\Sigma_{jk,S}$ given in section E.6, we find that $\Sigma_{jk} - \Sigma_{jk,S}\Sigma_S^{-1}\Sigma_{jk,S}^t$ has off-diagonal blocks

$$\begin{aligned}
&(\gamma_j^t \Sigma_G \gamma_k) \otimes K + (\gamma_j^t \Sigma_E \gamma_k) \otimes I_n \\
&- ((\gamma_j^t \Sigma_E \Gamma_S) \otimes I_n + B_j \otimes K)((\Gamma_S^t \Sigma_E \Gamma_S)^{-1} \otimes I_n - A \otimes K)(\Gamma_S^t \Sigma_E \gamma_k \otimes I_n + B_k^t \otimes K),
\end{aligned}$$

for $B_j = \gamma_j \Sigma_G \Gamma_S$ and $B_k = \gamma_k \Sigma_G \Gamma_S$.

Finally, working out the products in the last display (using that $K^2 = rK$), we find that all terms involving a kronecker product with I_n correspond exactly to the left-hand-side of (36). Consequently, the residual covariance in the distribution $(\mathbf{Y_j}, \mathbf{Y_k})|\mathbf{Y_S} = \tilde{y}_S$ is zero if and only if $\mathbf{Y_j} \perp\!\!\!\perp \mathbf{Y_k}|\{\mathbf{Y_S}, \mathbf{G}\}$.

E.10 Proof of Theorem 4

To obtain faithfulness for $S = \emptyset$, we need to prove that $\mathbf{Y_j} \perp\!\!\!\perp \mathbf{G}$ implies d-separation of Y_j and G in the graph $\mathcal{G}$. Because the conditioning set is empty, it suffices to show that there are no directed paths from G to Y_j, where we can assume that $j \notin D$ (otherwise $\mathbf{G_j}$ would be nonzero, and because of the non-collinearity, $\mathbf{Y_j}$ and $\mathbf{G}$ would not be independent).

Because of the assumed Gaussianity, the independence of $\mathbf{Y_j}$ and $\mathbf{G}$ implies that

$$Cov(\mathbf{Y_j}^t, \mathbf{G}) = Cov(\gamma_j^t \mathbf{G}^t, \mathbf{G}) = \text{trace}(K)\ \gamma_j^t \Sigma_G = (0, \ldots, 0), \tag{39}$$

where we used that $vec(\mathbf{G}) \sim N(0, \Sigma_G \otimes K)$, and therefore $E(\mathbf{G}[i,j]\mathbf{G}[i,k]) = \Sigma_G[j,k]K[i,i]$, for all $i \in \{1,\ldots,n\}$ and $j,k \in \{1,\ldots,p\}$. Since trace(K) is strictly positive and the submatrix $\Sigma_G[D,D]$ has full rank, equation (39) implies that $\gamma_{j,l} = 0$ for all $l \in D$.

Finally, we use that the assumed faithfulness implies the path coefficients condition (see sections E.2-E.5). Consequently, it follows from $\gamma_{j,l} = 0$ that there is no directed path from Y_l to Y_j. Since this is the case for all $l \in D$, there can neither be a directed path from G to Y_j.

E.11 Proof of Theorem 6

Assuming $K = ZZ^t$, the first part of theorem follows from the results in Appendix E.7. For the second part, we use that $\mathbf{Y_j}$ has genetic variance $\sigma_j^2(G) = \gamma_j^t \Sigma_G \gamma_j$ (see equation (25)). Because for traits without a direct genetic effect, rows and columns in Σ_G are zero, we can rewrite this as $\gamma_j^t[D]\Sigma_G[D,D]\gamma_j[D]$. Hence, $\sigma_j^2(G) = 0$ is equivalent with $\gamma_{j,l} = 0$ for all $l \in D$, where we used that $\Sigma_G[D,D]$ is of full rank. Using the arguments from the proof of Theorem 4 and the assumed faithfulness, it follows that this is equivalent with independence of $\mathbf{Y_j}$ and $\mathbf{G}$.

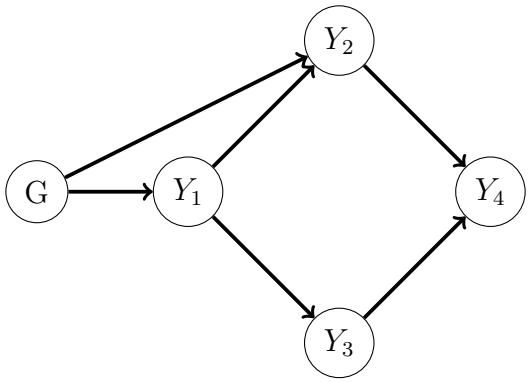


Figure 8. A genetic DAG with independent genetic effects on Y_1 and Y_2, and a direct effect $Y_1 \to Y_2$

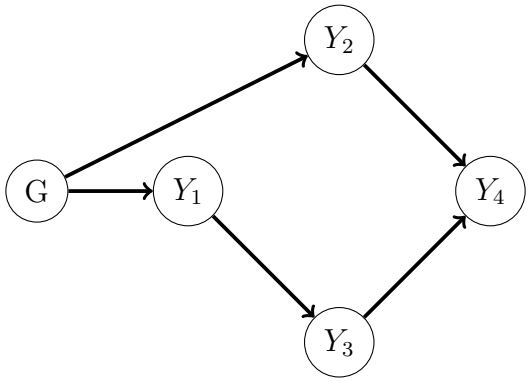


Figure 9. A typical output of PC-gen (without the modifications in section F.1), based on observations of Y_1, Y_2, Y_3, Y_4 generated by the genetic DAG of Figure 8, for 400 genotypes and 2 replicates. The edge between Y_1 and Y_2 is missing because the test for conditional independence between Y_1 and Y_2 given Y_4 has too little power.

F Skipping independence tests that do not involve G

F.1 Motivation for skipping the test for $\mathbf{Y_j \perp\!\!\!\perp Y_k | \{Y_S\}}$

Although PC-gen can be shown to be consistent, its finite sample performance can be improved if we skip some of the tests in the skeleton-stage. Differences between the population and sample version of PC-skeleton can occur everywhere in the graph, but are most likely for conditioning sets not containing G. This is illustrated in the example in Figure 8, in which there are genetic effects on traits Y_j and Y_k, as well as a direct effect of Y_j on Y_k.

Then given a large number of observations of Y_1, Y_2, Y_3, Y_4 and assuming faithfulness, we will recover the true skeleton. However, with a small or moderate sample size, the test

for conditional independence of Y_1 and Y_2 given Y_4 has very little power. The test for conditional independence of Y_1 and Y_2 given $\{Y_4, G\}$ is a lot more powerful here; however the standard PC-skeleton algorithm (and neither PC-stable) does not perform this test anymore after the null-hypothesis of conditional independence of Y_1 and Y_2 given Y_4 alone has been accepted. Therefore, a typical output of PC-gen looks like Figure 9.

In order to make PC-gen more powerful we therefore propose to perform only those conditional independence tests where G is contained in the conditioning set, at least when both variables whose conditional independence is tested have positive genetic variance[8]. Leaving out all conditional independence tests where G is not in the conditioning set still gives a valid algorithm, in the sense that in the population version of PC-skeleton, no false positives are obtained. Intuitively this is obvious, as G is a root node, and everything can be done conditionally on G. We formally show this in appendix F.

F.2 A characterization of the skeleton

If a distribution P is faithful with respect to a DAG $\mathcal{G}$, we have the following result for the skeleton of $\mathcal{G}$:

$$\begin{aligned} &\text{there is an edge between nodes } A \text{ and } B \text{ in the skeleton of a DAG } \mathcal{G} \\ \iff \quad &\forall S \subseteq V \setminus \{A, B\}, A \text{ and } B \text{ are conditionally dependent given } S. \end{aligned} \tag{40}$$

This was shown in [5] (Theorem 3.4); here we adopt the formulation of [32] (p.616). It is important to note that in general the skeleton is not equal to the so-called conditional independence graph (CIG), which is the undirected graph associated with the inverse covariance or precision matrix. The latter is characterized by an equivalence statement similar to (40), but with on the right-hand side only $S = V \setminus \{A, B\}$. Hence, if data are generated by a DAG $\mathcal{G}$ and we assume faithfulness, the skeleton of $\mathcal{G}$ is typically a subgraph of the CIG. In case $A \to B \leftarrow C$ for example, the CIG also contains an edge $A - C$ (because $S = \emptyset$ d-separates A and C, but $S = B$ does not, B being a collider on the path $A \to B \leftarrow C$).

F.3 Skipping the test for $\mathbf{Y_j \perp\!\!\!\perp Y_k | \{Y_S\}}$ does not affect pcgen (oracle)

In view of (40), our modification is correct in the sense that the population version of PC-skeleton still recovers the true skeleton. This correctness follows from the facts that

[8] In the true graph, we can partition $V = \{Y_1, \ldots, Y_p\}$ in a set V^G containing all variables having positive genetic variance and a set $(V^G)^c$ with variables without genetic variance. A variable Y_k is in V^G when there exists at least one directed path $G \to \ldots \to Y_k$. Any edge between $Y_j \in V^G$ and $Y_k \in (V^G)^c$ must be directed $Y_k \to Y_j$. When estimating the true graph (CPDAG) from data, we start PC-skeleton by testing marginal independence between each trait and G, i.e. testing genetic variance. Based on these tests we obtain estimates V^G and $(V^G)^c$, used in the remainder.

1. the PC-skeleton algorithm starts with the complete undirected graph, and then tests conditional independencies, removing edges when a conditionally independence relation is found. When in fact there *is* an edge $Y_j - Y_k$ in the true genetic DAG, then removing some of these tests clearly still produces the correct result (and in the sample version even with higher probability, which is the motivation of doing this...)

2. if there *is no* edge $Y_j - Y_k$ and a set S not containing G is blocking a path between Y_j and Y_k, then $\{G\} \cup S$ is also blocking it (since G can never be a collider, and neither be a descendant of any node). In other words, it can not happen that a set S not containing G is blocking all paths between Y_j and Y_k, and that the addition of G would 'unblock' one of these paths. Consequently, it can not happen that such a set S is the *only* set separating Y_j and Y_k, and that we would miss the only opportunity to remove the edge between Y_j and Y_k.

G Miscelleneous

G.1 Representing G by a single node: a motivating example

The main reason for representing $\mathbf{G_1}, \ldots, \mathbf{G_p}$ with a single node (instead of $G_1, \ldots, G_p$, sometimes used in the literature) is that the relatedness matrix K is the same for all traits. If for example $G_1 \to Y_1 \to Y_2 \leftarrow Y_3 \leftarrow G_3$ (as in Figure 2), it follows from (8) that the marginal distribution of $\mathbf{Y_2}$ has covariance $c_1 K + c_2 I_n$, for some nonnegative constants c_1, c_2. However, based on this distribution alone, we cannot distinguish the contributions of $\mathbf{G_1}$ and $\mathbf{G_3}$. This differs from the scenario $QTL_A \to Y_1 \to Y_2 \leftarrow Y_3 \leftarrow QTL_B$, where the total (fixed) effects of QTLs A and B on Y_2 can be estimated from the marginal distribution of Y_2. If we condition on $\mathbf{Y_1}$ and $\mathbf{Y_3}$, $\mathbf{Y_2}$ becomes independent of $\mathbf{G_1}$ and $\mathbf{G_3}$. In fact it is also independent of $\mathbf{G_2}$, since the latter is zero. Consequently, given $\mathbf{G_3}$, $\mathbf{Y_2}$ is independent of $\mathbf{G} = [\mathbf{G_1 G_2 G_3}]$, which illustrates that the conditional independencies correspond to a property of the graph $\mathcal{G}$, with $[\mathbf{G_1}\ \mathbf{G_2}\ \mathbf{G_3}]$ represented by a single node G.

G.2 The limitations of genomic networks

[13] recently proposed to estimate a directed network based on the predicted genetic effects themselves, rather than the residuals. Compared to residual-based estimation or pcgen, this however seems to require stronger assumptions.

As an example, consider the graph $Y_1 \to Y_3 \leftarrow Y_2$, with direct genetic effects on all 3 traits, i.e. $rank(\Sigma_G) = 3$. For the sake of the argument, assume also that the total genetic effects can be predicted without error, i.e. we can observe the matrix $\mathbf{U} := \mathbf{G}\Gamma$ (see equation (7); because Y_1 and Y_2 are root nodes in $\mathcal{G}_\mathcal{Y}$, it turns out that $\mathbf{U_1} := \mathbf{G_1}$ and $\mathbf{U_2} := \mathbf{G_2}$).

In order not to get an incorrect edge between Y_1 and Y_2 the PC-algorithm must find that these traits are marginally independent. Using residuals, we indeed find that $\mathbf{Y_1} - \mathbf{U_1}$ and $\mathbf{Y_2} - \mathbf{U_2}$ are independent. However, $\mathbf{U_1}$ and $\mathbf{U_2}$ are only independent if Σ_G is diagonal, which is a rather strong and restrictive additional assumption.

The only advantage of genomic networks over the residual-based ones is that they may infer some of the edges $G \to Y_j$. This is however only indirectly, through comparison with the trait-to-trait network estimated by the residuals-based network, and requires additional testing if the $\mathbf{U_j}$ are zero (i.e. the tests for $\mathbf{Y_j} \perp\!\!\!\perp \mathbf{G}|\emptyset$ in pcgen, which were not considered in [13]). Even then, the edges $G \to Y_j$ can be inferred only partially. In the above example, if we conclude that $\mathbf{U_j} = 0$ for some j, this clearly excludes $G \to Y_j$. But if $\mathbf{U_1}, \mathbf{U_2}, \mathbf{U_3}$ are all nonzero, we can only conclude (by comparing with the already known trait-to-trait network $Y_1 \to Y_3 \leftarrow Y_2$) that $G \to Y_1$ and $G \to Y_2$. But it is impossible to make any inference about $G \to Y_2$, which pcgen can in principle do.

H Maize data: reconstructions with $\alpha = 0.001$

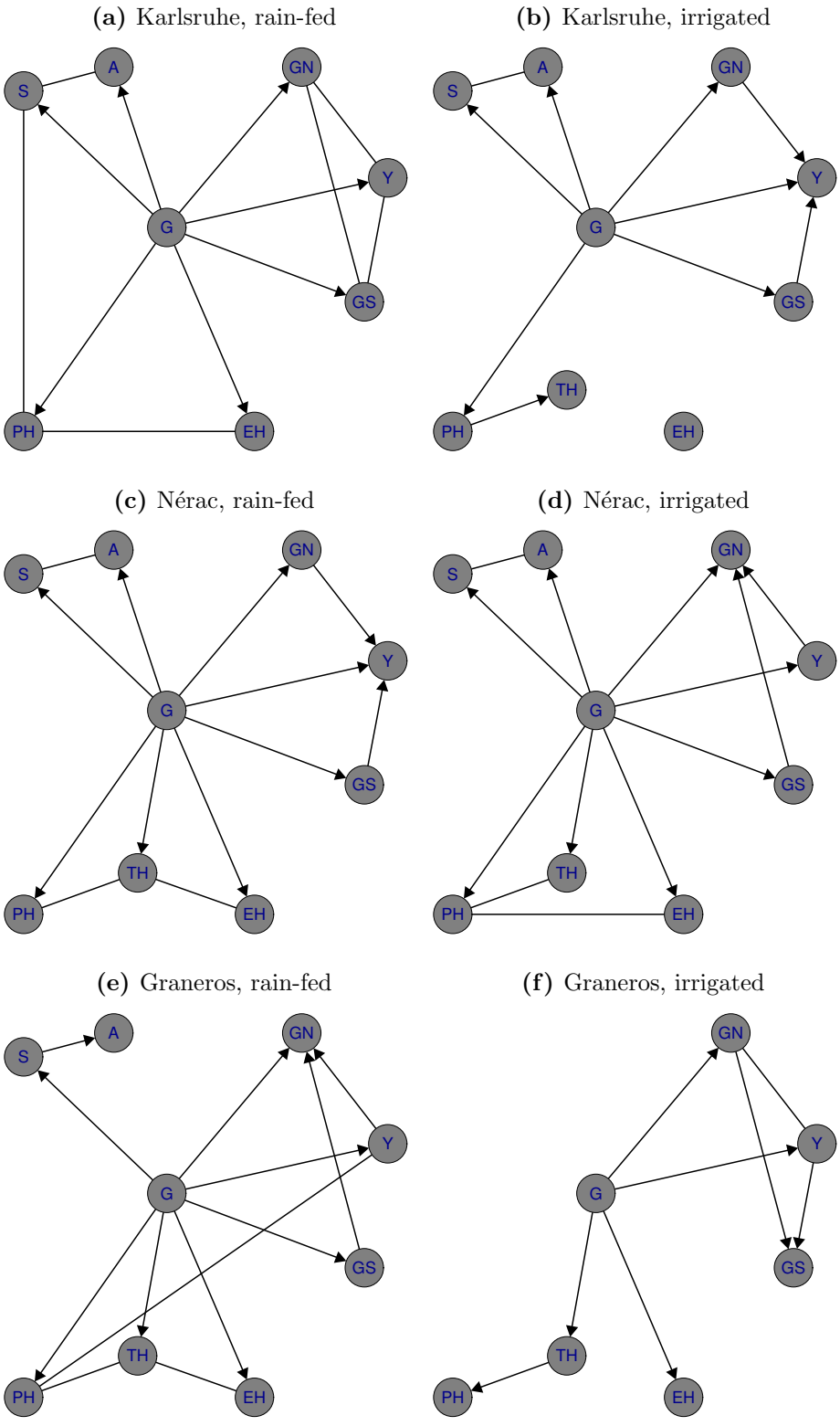


Figure 10. Estimated networks for six of the DROPS field trials in 2013, with $\alpha = 0.001$. Rows correspond to locations (Karlsruhe, Nérac, Graneros), columns to treatments (rain-fed, irrigated).

References

1. Wright S (1921) Correlation and causation. Journal of Agricultural Research : 557-585.

2. Wu XL, Heringstad B, Gianola D (2010) Bayesian structural equation models for inferring relationships between phenotypes: a review of methodology, identifiability, and applications. Journal of Animal Breeding and Genetics 127: 3–15.

3. Onogi A, Ideta O, Yoshioka T, Ebana K, Yamasaki M, et al. (2016) Uncovering a nuisance influence of a phenological trait of plants using a nonlinear structural equation: Application to days to heading and culm length in asian cultivated rice. PLoS ONE 11: 1-17.

4. Pearl J (2000) Causality. Causality, by Judea Pearl, pp.˜400.˜ISBN 0521773628.˜Cambridge, UK: Cambridge University Press, March 2000. URL http://adsabs.harvard.edu/cgi-bin/nph-bib_query?bibcode=2000caus.book.....P.

5. Spirtes P, Glymour C, Scheines R (2001) Causation, Prediction, and Search, Second Edition (Adaptive Computation and Machine Learning). A Bradford Book, second edition edition. URL http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/0262194406.

6. Maathuis MH, Nandy P (2016) A review of some recent advances in causal inference. Handbook of Big Data : 387.

7. Maathuis MH, Colombo D, Kalisch M, Bühlmann P (2010) Predicting causal effects in large-scale systems from observational data. Nature Methods 7: 247–248.

8. Chaibub Neto E, Ferrara CT, Attie AD, Yandell BS (2008) Inferring causal phenotype networks from segregating populations. Genetics 179: 1089–1100.

9. Chaibub Neto E, Broman AT, Keller MP, Attie AD, Zhang B, et al. (2013) Modeling causality for pairs of phenotypes in system genetics. Genetics .

10. Scutari M, Howell P, Balding DJ, Mackay I (2014) Multiple quantitative trait analysis using bayesian networks. Genetics 198: 129–137.

11. Gianola D, Sorensen D (2004) Quantitative genetic models for describing simultaneous and recursive relationships between phenotypes. Genetics 167: 1407–1424.

12. Valente BD, Rosa GJM, de los Campos G, Gianola D, Silva MA (2010) Searching for recursive causal structures in multivariate quantitative genetics mixed models. Genetics 185: 633–644.

13. Töpner K, Rosa GJM, Gianola D, Schön CC (2017) Bayesian networks illustrate genomic and residual trait connections in maize (zea mays l.). G3: Genes, Genomes, Genetics 7: 2779–2789.

14. Gao B, Cui Y (2015) Learning directed acyclic graphical structures with genetical genomics data. Bioinformatics : btv513.

15. Stephens M (2013) A unified framework for association analysis with multiple related phenotypes. PLoS ONE 8: e65245.

16. Valente BD, Rosa GJM, Gianola D, Wu XL, Weigel K (2013) Is structural equation modeling advantageous for the genetic improvement of multiple traits? Genetics 194: 561–572.

17. Valente BD, Morota G, Peñagaricano F, Gianola D, Weigel K, et al. (2015) The causal meaning of genomic predictors and how it affects construction and comparison of genome-enabled selection models. Genetics 200: 483–494.

18. Kruijer W, Boer MP, Malosetti M, Flood PJ, Engel B, et al. (2015) Marker-based estimation of heritability in immortal populations. Genetics 199: 379-398.

19. Drton M, Foygel R, Sullivant S (2011) Global identifiability of linear structural equation models. Ann Statist 39: 865–886.

20. Shipley B (2016) Cause and correlation in biology: A user's guide to path analysis, structural equations and causal inference. Cambridge, USA: Cambridge University Press, 2 edition.

21. Bollen KA (1989) Structural Equations with Latent Variables. Wiley-Interscience, 1 edition. URL `http://www.worldcat.org/isbn/0471011711`.

22. Moore AJ, Brodie III ED, Wolf JB (1997) Interacting phenotypes and the evolutionary process: I. direct and indirect genetic effects of social interactions. Evolution 51: 1352–1362.

23. Bijma P (2014) The quantitative genetics of indirect genetic effects: a selective review of modelling issues. Heredity 112: 61.

24. Peters J, Janzing D, Schölkopf B (2017) Elements of causal inference: foundations and learning algorithms. MIT press.

25. Zhou X, Stephens M (2014) Efficient multivariate linear mixed model algorithms for genome-wide association studies. Nat Meth 11: 407–409.

26. Korte A, Vilhjalmsson BJ, Segura V, Platt A, Long Q, et al. (2012) A mixed-model approach for genome-wide association studies of correlated traits in structured populations. Nat Genet 44: 1066–1071.

27. Pearl J (1989) Probabilistic reasoning in intelligent systems: networks of plausible inference. San Mateo etc.: Morgan Kaufmann Publishers, xiv + 552 pp.

28. Kalisch M, Mächler M, Colombo D, Maathuis MH, Bühlmann P (2012) Causal inference using graphical models with the R package pcalg. Journal of Statistical Software 47: 1–26.

29. Zwiernik P, Uhler C, Richards D (2014). Maximum likelihood estimation for linear gaussian covariance models. `arXiv:1408.5604`.

30. Colombo D, Maathuis MH (2014) Order-independent constraint-based causal structure learning. Journal of Machine Learning Research 15: 3741–3782.

31. Hauser A, Bühlmann P (2012) Characterization and greedy learning of interventional Markov equivalence classes of directed acyclic graphs. Journal of Machine Learning Research 13: 2409–2464.

32. Kalisch M, Bühlmann P (2007) Estimating high-dimensional directed acyclic graphs with the PC-algorithm. J Mach Learn Res 8: 613–636.

33. Millet E, Welcker C, Kruijer W, Negro S, Nicolas S, et al. (2016) Genome-wide analysis of yield in europe: allelic effects as functions of drought and heat scenarios. Plant Physiology : pp–00621.

34. Keating BA, Carberry PS, Hammer GL, Probert ME, Robertson MJ, et al. (2003) An overview of APSIM, a model designed for farming systems simulation. Eur J Agron 18: 267–288.

35. Holzworth DP, Huth NI, DeVoil PG, Zurcher EJ, Herrmann NI, et al. (2014) Apsim - evolution towards a new generation of agricultural systems simulation. Env Model Soft .

36. Richardson T, Spirtes P (2002) Ancestral graph markov models. Annals of Statistics : 962–1030.

37. Rodríguez-Álvarez MX, Boer MP, van Eeuwijk FA, Eilers PH (2017) Correcting for spatial heterogeneity in plant breeding experiments with p-splines. Spatial Statistics 23: 52 - 71.

38. Kadam N, Tamilselvan A, Lawas LMF, Quinones C, Bahuguna R, et al. (2017) Genetic control of plasticity in root morphology and anatomy of rice in response to water-deficit. Plant Physiology .

39. Henderson CR (1975) Best Linear Unbiased Estimation and Prediction under a Selection Model. Biometrics 31.

40. Furlotte NA, Eskin E (2015) Efficient multiple-trait association and estimation of genetic correlation using the matrix-variate linear mixed model. Genetics 200: 59–68.

41. Joo JWJ, Kang EY, Org E, Furlotte N, Parks B, et al. (2016) Efficient and accurate multiple-phenotype regression method for high dimensional data considering population structure. Genetics 204: 1379–1390.

42. Peters J, Bühlmann P, Meinshausen N (2016) Causal inference by using invariant prediction: identification and confidence intervals. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 78: 947–1012.

43. Visscher PM, Goddard ME (2014) A general unified framework to assess the sampling variance of heritability estimates using pedigree or marker-based relationships. Genetics .

44. Kruijer W (2016) Misspecification in mixed-model based association analysis. Genetics 202: 363-366.

45. Flaxman SR, Neill DB, Smola AJ (2015) Gaussian processes for independence tests with non-iid data in causal inference. Provisional acceptance at ACM Transactions on Intelligent Systems and Technology (TIST), 2015b URL http://wwwsethrfcom/files/gp-dependpdf .

46. Meinshausen N, Bühlmann P (2010) Stability selection. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 72: 417–473.

47. Stekhoven DJ, Moraes I, Sveinbjörnsson G, Hennig L, Maathuis MH, et al. (2012) Causal stability ranking. Bioinformatics 28: 2819-2823.

48. Bühlmann P, Rütimann P, Kalisch M (2011) Controlling false positive selections in high-dimensional regression and causal inference. Statistical Methods in Medical Research .

49. Meinshausen N, Hauser A, Mooij JM, Peters J, Versteeg P, et al. (2016) Methods for causal inference from gene perturbation experiments and validation. Proceedings of the National Academy of Sciences 113: 7361–7368.

50. Pfister N, Bühlmann P, Schölkopf B, Peters J (2018) Kernel-based tests for joint independence. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 80: 5-31.

51. Colombo D, Maathuis MH, Kalisch M, Richardson TS (2012) Learning high-dimensional directed acyclic graphs with latent and selection variables. Ann Statist 40: 294–321.

52. Dempster AP, Laird NM, Rubin DB (1977) Maximum likelihood from incomplete data via the em algorithm. Journal of the royal statistical society Series B (methodological) : 1–38.

53. Joe H (2006) Generating random correlation matrices based on partial correlations. Journal of Multivariate Analysis 97: 2177–2189.

54. Tsamardinos I, Brown LE, Aliferis CF (2006) The max-min hill-climbing bayesian network structure learning algorithm. Machine learning 65: 31–78.

55. Bustos-Korts D, Malosetti M, Chapman SC, Chenu K, Boer M, et al. (2017) A protocol combining statistical and crop growth modelling to evaluate phenotyping strategies useful for selection under different drought patterns. Ph.D. thesis, Wageningen University, Wageningen. doi:10.18174/421321.

56. Casadebaig P, Zheng B, Chapman S, Huth N, Faivre R, et al. (2016) Assessment of the potential impacts of plant traits across environments by combining global sensitivity analysis and dynamic modeling in wheat. PLoS One .

57. Lauritzen SL (1996) Graphical Models. Oxford Statistical Science Series. New York, USA: Oxford University Press. URL `http://www.worldcat.org/isbn/0198522193`.

58. Chickering DM (2002) Learning equivalence classes of bayesian-network structures. J Mach Learn Res 2: 445–498.

59. Lynch M, Walsh B (1998) Genetics and Analysis of Quantitative Traits. Sinauer Associates, 1 edition. URL `http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/0878934812`.

60. Peters J (2012) Restricted Structural Equation Models for Causal Inference. Ph.D. thesis, ETH Zurich and MPI for Intelligent Systems. `http://dx.doi.org/10.3929/ethz-a-007597940`.

61. Petersen KB, Pedersen MS, et al. (2008) The matrix cookbook. Technical University of Denmark 7: 510.

62. Golub GH, Van Loan CF (2012) Matrix computations, volume 3. JHU Press.