Reconstruction of networks with direct and indirect genetic effects

Willem Kruijer*, Pariya Behrouzi*, Daniela Bustos Korts*, María Xosé Rodríguez-Álvarez§,‡, Seyed Mahdi Mahmoudi†, Brian Yandell, Ernst Wit*** and Fred van Eeuwijk***

*Biometris, Wageningen University and Research (Wageningen, Netherlands), §BCAM - Basque Center for Applied Mathematics (Bilbao, Spain), ‡IKERBASQUE, Basque Foundation for Science (Bilbao, Spain), †Faculty of Mathematics, Statistics and Computer Science, Semnan University (Semnan, Iran), **University of Wisconsin-Madison (Madison, Wisconsin 53706-1510, USA), ***University of Lugano (Lugano, Switzerland)

ABSTRACT Functional relationships between phenotypic traits can be represented by structural equation models (SEMs), as proposed by Sewall Wright in 1921. Extended with genetic effects, SEMs can distinguish between direct and indirect genetic contributions to phenotypic traits. For example, given a causal effect of trait Y_1 on Y_2, the genetic variance of Y_2 partially depends on that of Y_1.

Often however the structure is unknown, and causal inference methods are used, which reconstruct a causal model that is in some sense most compatible with the data. Current causal inference methods usually consider genetic effects as nuisance terms, that need to be eliminated by taking residuals from an unstructured multi-trait mixed model (MTM). Unfortunately, fitting such MTM for large numbers of traits is computationally and statistically challenging, and it is impossible to infer the presence or absence of direct genetic effects.

Here we propose an alternative strategy, where genetic effects are formally included in the graph. Using simulations, real data and statistical results we show that this has important advantages: (1) genetic effects can be directly incorporated in causal inference, leading to the PCgen algorithm, which can handle many more traits than current approaches; (2) we can test the existence of direct genetic effects, and also improve the orientation of edges between traits. For example, we can, under certain assumptions, recover the structure $G \rightarrow Y_1 \rightarrow Y_2$ if the genetic variance of trait Y_2 given Y_1 is found to be zero. Finally, we show that network reconstruction is much more accurate if individual plant or plot data are used, extending existing results for single trait analyses. We have implemented the PCgen-algorithm in the R-package pcgen.

KEYWORDS Structural Equation Models, multivariate mixed models, causal inference

Introduction

To attain higher genetic gains, modern plant and animal breeders increasingly scale up their programs via the implementation of genomic prediction technologies (Cooper *et al.* 2014). Most of the genomic prediction applications are based on linear mixed- or Bayesian models that predict the phenotype for the target trait (yield) as a function of a multivariate distribution for SNP effects. In these models, the physiological mechanisms and traits that modulate the genotypic response to the environment over time are modeled implicitly via the SNP effects on the target trait (Zhou and Stephens 2014; Calus and Veerkamp 2011). The availability of high throughput phenotyping technologies has enabled breeders to characterize additional traits and monitor growth and development during the season. This opens new opportunities in breeding strategies, in which better-adapted genotypes result from combining loci that regulate complementary physiological mechanisms. This kind of breeding strategy is called physiological breeding (Reynolds and Langridge 2016). In physiological breeding, prediction accuracy for the target trait benefits from modeling the target trait and its underlying traits simultaneously (van Eeuwijk *et al.* 2018) because of a larger

Manuscript compiled: Monday 15th July, 2019
[3]Corresponding author: W. Kruijer (E-mail: willem.kruijer@wur.nl)

power to estimate its underlying effects (Stephens 2013). The challenge of seeking complementary traits is that many observed traits will be highly correlated to each other. Therefore, is hard to disentangle which are the causal mechanisms modulating the genotypic response over time that could potentially lead to and a larger prediction accuracy and to a transgressive segregation for the target trait. Structural equation models (SEMs), proposed by Wright (Wright 1921), and extended with random genetic effects in (Gianola and Sorensen 2004), are a promising approach to deal with this problem. In SEMs each trait is modelled explicitly as a function of the other traits and a noise term. Therefore, SEMs are a useful tool to identify which are the key traits that could be selection targets, or incorporated in multi-trait genomic prediction models to improve the prediction accuracy for the target trait (van Eeuwijk *et al.* 2018).

A first advantage of SEMs, compared to regression models, is that one can predict the behavior of the system when one or more of the structural equations are modified by some kind of intervention. Figure 1 shows an illustration of an intervention. For example, a question that could be interesting to plant breeders is; which would be the contribution of a trait (say, radiation use efficiency) to yield, if flowering time is fixated for all genotypes at a particular value?

Second, SEMs make possible to distinguish between direct and indirect effects of one trait on another, and similarly between direct and indirect genetic effects. For example, let plant height (trait $\mathbf{Y_1}$) be modelled as $\mathbf{Y_1} = \mathbf{G_1} + \mathbf{E_1}$, i.e., as the sum of a random genetic and residual effect, where all terms are $n \times 1$ vectors, containing the values for a population of n individuals. Suppose plant height has a linear effect on yield ($\mathbf{Y_2}$), with additional random effects $\mathbf{G_2}$ and $\mathbf{E_2}$:

$$\mathbf{Y_2} = \lambda \mathbf{Y_1} + \mathbf{G_2} + \mathbf{E_2} = (\lambda \mathbf{G_1} + \mathbf{G_2}) + \lambda \mathbf{E_1} + \mathbf{E_2},$$

where λ is the path (or structural) coefficient associated with the effect of $\mathbf{Y_1}$ on $\mathbf{Y_2}$. On the one hand, we have the *direct* genetic effects $\mathbf{G_1}$ and $\mathbf{G_2}$; on the other hand, we have the *total* (or marginal) genetic effects $\mathbf{U_1} = \mathbf{G_1}$ and $\mathbf{U_2} = \lambda \mathbf{G_1} + \mathbf{G_2}$, the indirect effects being $\mathbf{U_1} - \mathbf{G_1} = 0$ and $\mathbf{U_2} - \mathbf{G_2} = \lambda \mathbf{G_1}$. Similarly, we can distinguish between the 2×2 matrices Σ_G, containing the (co)variances of $\mathbf{G_1}$ and $\mathbf{G_2}$, and V_G, with the (co)variances of $\mathbf{U_1}$ and $\mathbf{U_2}$. The latter is the matrix of genetic (co)variances, appearing in the usual MTM for $\mathbf{Y_1}$ and $\mathbf{Y_2}$; here it is a function of Σ_G and λ.

Knowledge of the direct genetic effects is in principle of great interest to breeders (Valente *et al.* 2013, 2015). However, routine use of these models is currently difficult for two reasons. First, for a given SEM, not all parameters may be identifiable, i.e., because of overparameterization different sets of parameter values can lead to the same model, making estimation infeasible. Gianola and Sorensen (2004) provided criteria for identifiability and suggested putting constraints on some of the parameters, although automatic generation of interpretable and meaningful constraints remains difficult, especially in high-dimensional settings.

A second (and more fundamental) obstacle for the use of SEMs with genetic effects is that the underlying structure is often unknown. In such cases causal inference methods (Pearl 2009; Spirtes *et al.* 2001; Maathuis and Nandy 2016) can be used, that reconstruct causal models that are in some sense most compatible with the observed data. Most causal inference methods however require independent samples, and cannot account for genetic relatedness. For this reason, genotypic differences are most often modelled using a small number of QTLs with fixed effects (Chaibub Neto *et al.* 2008; Neto *et al.* 2013; Scutari *et al.* 2014), but when part of the genetic variance is not explained by QTLs (missing heritability), the use of random genetic effects seems invitable. Only a few works have studied reconstruction in the presence of such effects. Valente *et al.* (2010) and Töpner *et al.* (2017) proposed to perform causal inference after subtracting genomic predictions obtained from an MTM. Similarly, Gao and Cui (2015) applied the PC-algorithm (Spirtes *et al.* 2001) 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 the causal graph among three traits is $Y_1 \rightarrow Y_2 \leftarrow Y_3$, and there are direct genetic effects on Y_1 and Y_3, then the absence of a direct genetic effect on Y_2 cannot be inferred from MTM residuals.

Inspired by these problems we define a framework in which direct genetic effects are part of the causal 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 on Y_j is nonzero, i.e., if the jth diagonal element of Σ_G is positive. See Figure 3 below for an example. Although our causal interpretation of genetic effects is not new (Stephens 2013; Valente *et al.* 2013, 2015), this work appears to be the first that formalizes it. In particular, we show that the Markov property holds for the graph extended with genetic effects (Theorem 1 below). Informally speaking, this means that there is a one-to-one correspondence between edges in the causal graph and conditional dependencies in the distribution of the traits and genetic effects. This means that edges (either between two traits, or between a trait and G) can in principle be inferred from multi-trait data. Consequently, while some of the covariances between direct genetic effects (contained in Σ_G) may still be unidentifiable, we *can* identify which rows and columns in Σ_G are zero.

Based on the Markov-property we propose the PCgen algorithm. PCgen stands for PC with genetic effects, and is an adaptation of the general PC-algorithm (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. Under the usual assumptions of independent errors, recursiveness and faithfulness, we show that PCgen can recover the underlying partially directed graph (Theorem 2). Because fitting an MTM for all traits simultaneously is no longer necessary, PCgen can handle a considerably larger number of traits.

While our approach is generally applicable to any species and relatedness matrix, our implementation of PCgen is currently limited to the specific (but important) case of populations where observations on genetically identical replicates are available, assuming independent genetic effects (i.e. as in the classical estimation of broad-sense heritability). This is partly for pragmatic reasons (e.g., the lower computational requirements), and partly for statistical reasons. In particular, successful reconstruction requires sufficient power in the tests for direct genetic effects ($G \rightarrow Y_j$) and those for the between traits relations ($Y_j \rightarrow Y_k$). Given the availability of 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 (GRM), modelling additive effects (Kruijer *et al.* 2015). Although mixed

models with *both* replicates and a GRM may further increase power, the increase is often modest, and statistical inference can become biased under model misspecification (e.g., when the GRM models additive effects, and the true architecture is partly epistatic; see Kruijer (2016)). By contrast, using only replicates, unbiased estimation of broad-sense heritability is always possible, regardless of the population structure and genetic architecture. The downside is that the contributions of different types of genetic effects cannot be distinguished. On the positive side, PCgen appears to be the first algorithm that can infer the presence of direct genetic effects based on phenotypic data alone.

Our approach is related to that of Stephens (2013), who inferred the sets of traits being directly and indirectly affected by a given locus, assuming unrelated individuals and using only summary statistics. Here we instead consider sums of individual locus effects, for possibly related individuals. Moreover, PCgen also aims to reconstruct the causal structure *within* the sets of traits with direct and indirect genetic effects, and can deal with larger numbers of traits.

The paper is organized as follows. After introducing SEMs with genetic effects, we define their graphical structure, and from this perspective review existing approaches. We then describe the general form of the PCgen-algorithm for estimation of the graphical structure, followed by various proposals for the required conditional independence tests. Next we test PCgen performance in data simulated with a statistical as well as a crop-growth model, and analyze a maize and a rice data-set. Finally, we state several results regarding PCgen's statistical properties. Table S1 provides an overview of the notation, and Appendix A.1 contains the necessary graph-theoretic definitions. Figure 2 provides a graphical summary of our theory and methodology.

Materials and methods

Structural Equation Models

To introduce structural models, we first consider a simple linear 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 $\mathbf{x_i}$.

To write (1) in matrix-form, we define the $n \times q$ design matrix $\mathbf{X}$ with rows $\mathbf{x_i}$. Similarly we define $n \times p$ matrices $\mathbf{Y} = [\mathbf{Y_1} \cdots \mathbf{Y_p}]$ and $\mathbf{E} = [\mathbf{E_1} \cdots \mathbf{E_p}]$, with rows $\mathbf{y_i}$ and $\mathbf{e_i}$, and columns $\mathbf{Y_j}$ and $\mathbf{E_j}$. We can then 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. 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 $\mathbf{Y_3} = \mu_3 1_n + \lambda_{13}\mathbf{Y_1} + \lambda_{23}\mathbf{Y_2} + \mathbf{E_3}$. 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

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

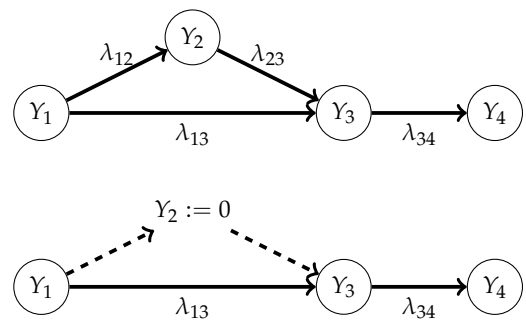

Figure 1. An example of a linear SEM. The SEM can be represented by a graph (middle), which is defined by the nonzero elements of Λ, the matrix containing the path coefficients (top). The total effect of Y_1 on Y_4 can be obtained by summing the contributions of the directed paths $Y_1 \to Y_3 \to Y_4$ and $Y_1 \to Y_2 \to Y_3 \to Y_4$, where each contribution is the product of the corresponding path coefficients. After the intervention $Y_2 := 0$ (bottom), the effect changes from $(\lambda_{13}\lambda_{34} + \lambda_{12}\lambda_{23}\lambda_{34})$ to $\lambda_{13}\lambda_{34}$.

does not contain any cycle (i.e., a directed path from a trait to itself), it is a directed acyclic graph (*DAG*), and the SEM is said to be *recursive*. In the notation we will distinguish between 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).

As mentioned above, SEMs can be used to predict the effects of *interventions*, which mathematically correspond to changes in the structural equations. For example, suppose that in Figure 1, Y_1, Y_2 and Y_3 are the expression levels of three 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 $\lambda_{13}\lambda_{34}$. Another example is given in File S7.3. More generally, the new joint distribution of $\mathbf{Y_1}, \ldots, \mathbf{Y_p}$ after an intervention can be obtained from the manipulation or truncated factorization theorem (Pearl 2009), *without* observations from the new distribution. For the consequences for genomic prediction, see Valente *et al.* (2013) and the Discussion below.

GSEM: Structural Equation Models with genetic effects

Gianola and Sorensen (2004) extended model (1) with random genetic effects $\mathbf{g_i}$: for individuals $i = 1, \ldots, n$, it is then assumed that

$$\mathbf{y_i} = \mathbf{x_i}B + \mathbf{y_i}\Lambda + \mathbf{g_i} + \mathbf{e_i}, \tag{3}$$

where the $1 \times p$ vectors $\mathbf{g_i}$ contain the direct genetic effects for individuals $i = 1, \ldots, n$. We will refer to model (3) 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 Moore *et al.* (1997), and Bijma (2014)). 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:

$$\mathbf{Y} = [\mathbf{Y_1} \cdots \mathbf{Y_p}] = \mathbf{X}B + \mathbf{Y}\Lambda + \mathbf{G} + \mathbf{E}. \quad (4)$$

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 (4):

1. *all traits are measured in the same experiment:* the rows $\mathbf{y_i}$ of $\mathbf{Y}$ may be either observations at plot or plant level or genotypic means across plots or plants, but the observations should always come from the same experiment. In addition, the residual errors originate from biological variation, i.e., measurement errors are negligible.

2. *recursiveness:* the graph $\mathcal{G}_\mathcal{Y}$ defined by Λ is a DAG. Consequently, there should be no feedback loops.

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 non-zero.

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 below (section 'Genetic relatedness'). Equivalent to 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 should not be 1 or -1.

Assumptions 1-4 were also made in related work on structural models with random genetic effects (Valente *et al.* 2010; Töpner *et al.* 2017), and 1-3 are commonly made for structural models without such effects. Assumption 1 is implicit in the GSEM model (4) 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 'Data from different experiments'). 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., Peters *et al.* (2017)), although we will not explore this here.

As in related work (Valente *et al.* 2010; Töpner *et al.* 2017) as well as in much of the literature on multi-trait genomic prediction and GWAS (see e.g., Zhou and Stephens (2014) and Calus and Veerkamp (2011)), the relatedness matrix K is the same for all traits. This may not hold if traits have very different genetic architectures, but seems a good approximation if most of the underlying QTLs are small. Large QTLs may be added as additional fixed effects, as discussed below.

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

In contrast 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. Following the notation of Stephens (2013) (in the context of individual loci), we use $D \subseteq \{1, \ldots, p\}$ to 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 5 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 (4) 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 3 for an example. In words, $\mathcal{G}$ is defined as the original graph $\mathcal{G}_\mathcal{Y}$ over the traits, extended with the node G and arrows $G \to Y_j$ for 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 between the node G in the graph (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 File S7.1 for a motivating example. The orientation of any edge between G and Y_j is restricted to $G \to Y_j$, 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 (Gianola and Sorensen 2004), 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. The skeleton is generally not equal to the conditional independence graph, which is the undirected graph associated with the inverse covariance or precision matrix (Spirtes *et al.* 2001; Kalisch and Bühlmann 2007). See File S6.2 for an example.

Direct and indirect genetic effects

As pointed out by various authors (Gianola and Sorensen 2004; Valente *et al.* 2010, 2013; Töpner *et al.* 2017), 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 $\mathcal{G}_\mathcal{Y}$. Assuming that the inverse $\Gamma = (I - \Lambda)^{-1}$ exists

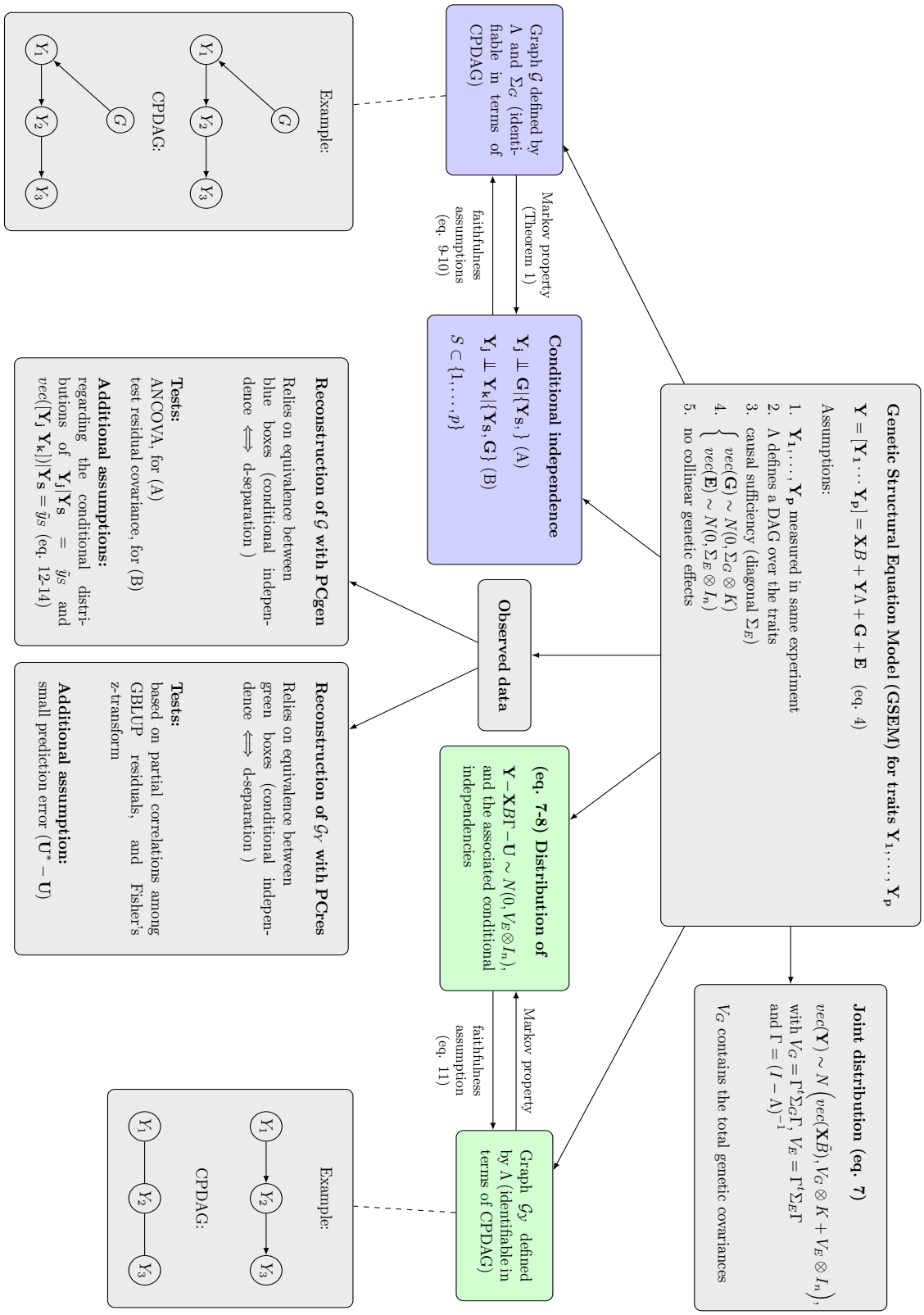

Figure 2. Graphical summary of the theory and methodology. The Markov property on the right (green; for the residuals) is well known from the literature, while the Markov property on the left (blue, for the conditional distributions of $\mathbf{Y_1}, \ldots, \mathbf{Y_p}, \mathbf{G}$) is established in Theorem 1. Table S1 contains an overview of the notation, and Appendix A.1 provides the necessary graph-theoretic definitions.

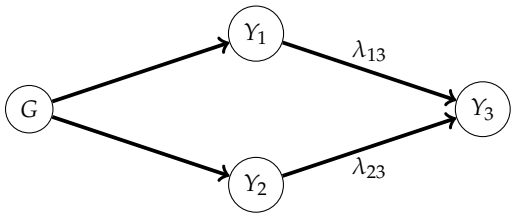

Figure 3. An example of a graph $\mathcal{G}$ representing a genetic structural equation model (GSEM), with path-coefficients λ_{13} and λ_{23}. There is no direct genetic effect on Y_3, and therefore no edge $G \to Y_3$.

[3], it follows from equation (3) that

$$\begin{aligned}\mathbf{y_i} &= \mathbf{x_i}B\Gamma + \mathbf{g_i}\Gamma + \mathbf{e_i}\Gamma = \mathbf{x_i}B\Gamma + \mathbf{u_i} + \mathbf{e_i}\Gamma \\ &\sim N(\mathbf{x_i}(B\Gamma), \Gamma^t\Sigma_G\Gamma + \Gamma^t\Sigma_E\Gamma) = N(\mathbf{x_i}(B\Gamma), V_G + V_E),\end{aligned} \tag{5}$$

where the $1 \times p$ vector $\mathbf{u_i} = \mathbf{g_i}\Gamma$ is the *total* genetic effect for the ith individual. The $n \times 1$ vector $\mathbf{U_j} = \mathbf{G}\gamma_j$ contains the total genetic effects for the jth trait, where γ_j is defined as the jth column of Γ. The vector of *indirect* genetic effects is the difference $\mathbf{U_j} - \mathbf{G_j}$. In Figure 3 for example [4], $\mathbf{G_3} = (0, \ldots, 0)^t$ and $\mathbf{U_3} = \lambda_{13}\mathbf{G_1} + \lambda_{23}\mathbf{G_2}$.

Likewise, we can distinguish between the contribution of direct and indirect genetic effects to the genetic covariance. The (j,k)th element of $V_G = \Gamma^t\Sigma_G\Gamma$ in (5) is the *total* genetic covariance between $\mathbf{Y_j}$ and $\mathbf{Y_k}$. This is what is usually meant with genetic covariance. Most often, this is different from the covariance between the direct genetic effects $\mathbf{G_j}$ and $\mathbf{G_k}$, given 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 $\Gamma = (I - \Lambda)^{-1}$. If no such relations exist, then Λ contains only zeros, and $V_G = \Sigma_G$. In general however these matrices are different, and depending on the structure of the graph and the path coeffecients, the correlation $\Sigma_G[j,k]/\sqrt{\Sigma_G[j,j]\Sigma_G[k,k]}$ may be much larger than $V_G[j,k]/\sqrt{V_G[j,j]V_G[k,k]}$, or vice-versa. For example, given direct effects $\mathbf{G_1}$ and $\mathbf{G_2}$ with equal variance and correlation 0.9, and an effect $Y_1 \to Y_2$ of size -1, the total genetic correlation is -0.22. Regarding the diagonal of V_G, we note that traits without a direct genetic effect may still have positive genetic variance.

Genetic relatedness

The genetic relatedness matrix K introduced in assumption 4 determines the covariance between the rows of $\mathbf{G}$. In principle our approach allows for any type of GRM, but for simplicity we focus on the following types. In all cases, K has dimension $n \times n$.

- $K = ZZ^t$, 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 (or repeatability). Since no marker-information is included, the model cannot be directly 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.
- given only a single individual per genotype (or genotypic means) we assume $K = A$, A being a $(n \times n)$ GRM estimated from a dense set of markers, assuming additive infinitesimal effects.
- given both r replicates of m genotypes and a GRM A of dimension $m \times m$, we assume that $K = ZAZ^t$. In absence of non-additive effects, this covariance structure uses all available information. However, for computational reasons it is usually easier to work with either the replicates or with genotypic means and the GRM A. We further explore this issue in the simulations below and in the Discussion.

The balance required when $K = ZZ^t$ is necessary in Theorems 5 and 6 below, but is not a general requirement for our models, nor for the PCgen algorithm.

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 (4) as

$$\mathbf{Y} = \mathbf{X}B\Gamma + \mathbf{G}\Gamma + \mathbf{E}\Gamma = \mathbf{X}B\Gamma + \mathbf{U} + \mathbf{E}\Gamma, \tag{6}$$

where $\mathbf{U} = \mathbf{G}\Gamma$ is the $n \times p$ matrix of total genetic effects, with columns $\mathbf{U_j}$. Equation (5) now generalizes to

$$vec(\mathbf{Y}) \sim N\left(vec(\mathbf{X}\tilde{B}), V_G \otimes K + V_E \otimes I_n\right), \tag{7}$$

where $\tilde{B} = B\Gamma$ is the matrix of fixed effects transformed by Γ. This is a common model for multi-trait GWAS and genomic prediction (see among others Stephens (2013), Zhou and Stephens (2014), Korte *et al.* (2012)). In those applications, however, V_G and V_E are usually arbitrary covariance matrices, whereas here they are modelled as functions of Σ_G, Σ_E and Γ.

Using the results of Spirtes *et al.* (2001) (p. 371), it turns out that Γ can be written directly in terms of sums of products of path coefficients (see Appendix A.2). Consequently, there is no need to invert $(I - \Lambda)$, although it still holds that $\Gamma = (I - \Lambda)^{-1}$, provided the inverse exists. Recalling that γ_j is the jth column of Γ, we can express the jth trait as

$$\mathbf{Y_j} = \mathbf{X}B\gamma_j + \mathbf{G}\gamma_j + \mathbf{E}\gamma_j = \mathbf{X}B\gamma_j + \mathbf{U_j} + \mathbf{E}\gamma_j, \tag{8}$$

which is equation (6) restricted to the jth column. Similarly, for any non-empty index-set $S \subset \{1, \ldots, p\}$, the $n \times |S|$ matrix $\mathbf{Y_S}$ of traits in S (i.e., $\mathbf{Y}$ with columns restricted to S) equals $\mathbf{X}B\Gamma_S + \mathbf{G}\Gamma_S + \mathbf{E}\Gamma_S$, where Γ_S is the $p \times |S|$ matrix with columns γ_j ($j \in S$). Equation (24) (Appendix A.6) provides an expression for the covariance of $vec(\mathbf{Y_S})$. For the corresponding nodes in the graph, we write $Y_S = \{Y_m : m \in S\}$.

Causal inference without genetic effects

So far we have assumed that $\mathcal{G}$ is known: given sufficient restrictions on Λ, Σ_G and Σ_E, it may then be possible to estimate these matrices (Gianola and Sorensen 2004). In this work however, we aim to reconstruct an unknown $\mathcal{G}$, based on observations from a GSEM of the form (4). We will do this with the PCgen algorithm introduced below, but first briefly review the necessary concepts, as well existing methods. Appendix A.1 contains a more detailed introduction.

Suppose for the moment we have observations generated by an acyclic SEM without latent variables, and without genetic

[3] We will see below that Γ can be computed from sums of products of path coefficients, without inverting Λ. Note that in the residual and genetic covariances in equation (5), we get Γ^t in front instead of Γ, as we *post*-multiply with Γ.

[4] In this particular example, $\Gamma = (I - \Lambda)^{-1} = (I + \Lambda)$.

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 using observational data, at least not under the assumptions we make here. For the reconstruction of the CPDAG, constraint-based and score-based methods have been developed (for an overview, see Peters *et al.* (2017)).

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 (Pearl 2014) exhaustively tested all possible subsets, the PC-algorithm (Spirtes *et al.* 2001) can often greatly reduce the number of subsets to be considered. Although this is not essential for the equivalence of d-separation and conditional independence, most constraint-based algorithms assume that observations be indendently and identically distributed, and structural equations with additional random effects are usually not considered.

Existing approaches for estimating $\mathcal{G}_\mathcal{Y}$, given genetic effects

To deal with the dependence introduced by the genetic effects, Valente *et al.* (2010) and Töpner *et al.* (2017) proposed to predict the total genetic effects (i.e., the term $\mathbf{U}$ 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 that $G \to Y_1 \to Y_2 \to Y_3$, 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 $G \to Y_2$. Another disadvantage is that fewer of the between-trait edges can be oriented. Technically this is because in the CPDAG (showing which orientations can be recovered from data), typically more edges are undirected; see Appendix A.1 for more details. In the preceding example, the CPDAG associated with $\mathcal{G}$ is $G \to Y_1 \to Y_2 \to Y_3$, i.e., all orientations can be recovered (see also Figure 2). By contrast, the CPDAG associated with $\mathcal{G}_\mathcal{Y}$ is $Y_1 - Y_2 - Y_3$, and we only know that the orientation is *not* $Y_1 \to Y_2 \leftarrow Y_3$.

To use the causal information associated with the genetic effects, Töpner *et al.* (2017) estimated 'genomic networks', based on the predictions themselves. These however seem to require additional assumptions, which are not required for the residual networks (in particular, diagonal Σ_G). Moreover, it seems difficult to relate edges in such a network to direct genetic effects (see the section 'Data from different experiments' in the Discussion, and File S7.2). In summary, residual and genomic networks only estimate the (CPDAG associated with the) subgraph $\mathcal{G}_\mathcal{Y}$ of trait to trait relations, instead of the complete graph $\mathcal{G}$.

Another disadvantage is that without specific models putting restrictions on V_G and V_E, the MTM (7) can only be fitted for a handful of traits (Zhou and Stephens 2014) for statistical as well as computational reasons. For example, Zwiernik *et al.* (2014) 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 problem with fitting the MTM to data from the GSEM model (4) is that one cannot exploit the possible sparseness of $\mathcal{G}$. Even for sparse graphs with few direct genetic effects, the matrices $V_G = \Gamma^t \Sigma_G \Gamma$ and $V_E = \Gamma^t \Sigma_E \Gamma$ may still be dense, requiring a total of $p(p+1)$ parameters. To overcome these limitations, we explicitly 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 $\mathcal{G}$.

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 the various independence tests; the equivalence is addressed in Theorem 1 below. 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 File S1 (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.

In order to obtain PCgen, we 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 ($\mathbf{Y_j}$) 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 when 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 Colombo and Maathuis (2014), who proposed to perform all operations in skeleton- and orientation-stage list-wise (details given in File S1). Apart from eliminating the order-dependence, this has the advantage that all conditional independence tests of a given size $|S| = s$ can be performed in parallel.

In summary, PCgen is the PC-stable algorithm with: (1) specific conditional independence tests (described shortly below) and (2) modified orientation rules, in order to avoid edges pointing into G (File S1.2). 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 section 'Assessing uncertainty' below and the Discussion.

Skeleton stage: conditional independence tests We can distinguish between the following types of conditional independence statements in the skeleton stage:

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

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

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

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, (A) 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, (B) 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 as well as some additional assumptions, which we discuss in more detail below (section 'Statistical properties of PCgen'). Statement (C) can be tested using standard partial correlations and Fisher's z-transform. However, as we show in File S6, 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 including $\mathbf{G}$, i.e., $\mathbf{Y_j} \perp\!\!\!\perp \mathbf{Y_k} | \{\mathbf{Y_S}, \mathbf{G}\}$.

Testing $\mathbf{Y_j} \perp\!\!\!\perp \mathbf{G} | \mathbf{Y_S}$ Our test for statement (A) 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 $Y_S = \{Y_m : m \in S\}$. 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 a likelihood ratio test (LRT). 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 conditional distribution of $\mathbf{Y_j}$ given $\mathbf{Y_S} = \tilde{y}_S$ is that of a single trait mixed model, the mean being a linear regression over the conditioning traits. This assumption is made mathematically precise below in equations (12) and (14).

Testing $\mathbf{Y_j} \perp\!\!\!\perp \mathbf{Y_k} | \{\mathbf{G}, \mathbf{Y_S}\}$ For statement (B) we mostly use the residual covariance (RC) test, which is based on the conditional distribution of $vec([\mathbf{Y_j}\ \mathbf{Y_k}])$ given the observed $\mathbf{Y_S} = \tilde{y}_S$. It is assumed that this distribution is that of a bivariate MTM, again with the mean being a linear regression over the conditioning traits; see equations (13) and (14) below. Assuming the bivariate MTM, we test whether the residual covariance[5] is zero, using the LRT described in File S1.3. 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 residual covariance, while accounting for all genetic (co)variances. The RC-test is not to be confused with a test for zero *genetic* covariance. The latter is often useful for data exploration, but has no role in PCgen. In File S1.4 we describe a LRT test, which is implemented in our software.

An alternative to the RC-test is the RG-test (*R*esiduals of *G*BLUP). Fitting the MTM (7), we obtain the BLUP $\mathbf{U}^*$ of the total genetic effects $\mathbf{U} = \mathbf{G}\Gamma$, and the BLUE $\tilde{B}^*$ of the fixed effects. We then test the significance of partial correlations among residuals, i.e. the columns of $\mathbf{Y} - \mathbf{U}^* - \mathbf{X}\tilde{B}^*$. When $\mathbf{U}^*$ is close enough to $\mathbf{U}$, it follows from (6)-(7) that the covariance of $vec(\mathbf{Y} - \mathbf{U}^*)$ is approximately $(\Gamma^t \Sigma_E \Gamma) \otimes I_n$, i.e., that of independent samples, without any genetic relatedness. This approach is very similar to the work of Valente *et al.* (2010) and Töpner *et al.* (2017), who instead took a fully Bayesian approach to predict $\mathbf{U}$. In either case, the performance of the RG-test critically depends on the prediction error $(\mathbf{U}^* - \mathbf{U})$. As mentioned before, fitting an MTM is usually challenging for more than 5-10 traits; we therefore also consider residuals of single trait models.

PCres: reconstructing only trait-to-trait relationships Testing only conditional independencies of the form (B), 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 Valente *et al.* (2010) and Töpner *et al.* (2017). Staying within the context of the PC-algorithm and using residuals from GBLUP, we will call this PCres. As for the RG-test in PCgen, PCres can be based on residuals of either single or multi-trait mixed models.

Software

In our R-package `pcgen` we implemented PCgen for the case $K = ZZ^t$. PCres is implemented for $K = ZZ^t$, $K = A$ as well as $K = ZAZ^t$. Moreover, PCres can be based on either residuals of the full MTM (7) (only for small numbers of traits), or from univariate models (the default). Tables 2 and 3 in File S2 provide a complete overview of the options, with the required R-commands. The package is freely available at `https://cran.r-project.org/web/packages/PCgen/index.html`. `pcgen` is built on the pcalg package (Kalisch *et al.* 2012; Hauser and Bühlmann 2012), in which we modified the orientation rules and the default conditional independence test.

Assessing uncertainty

The PC-algorithm is asymptotically correct, in the sense that the underlying CPDAG is recovered if conditional independence

[5] alternatively, we could test the residual correlation

can be tested without error (Spirtes *et al.* 2001). In Theorem 2 below we provide a similar 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). Reliable control of the (expected) false positive rate or total number of false positives remains challenging; see the Discussion ('Assessing uncertainty'). We will therefore just consider the p-values as they are, and analyze a given dataset for various significance thresholds. Following Kalisch and Bühlmann (2007) and Kalisch *et al.* (2012), we report, for each remaining edge, the largest p-value found across all conditioning sets for which the edge was tested.

Extensions of PCgen

File S3 describes several extensions of PCgen, which are partly implemented in our software. Among others, the causal graph $\mathcal{G}$ and PCgen could be extended with fixed effect QTLs, and PCgen can be speed up by starting with a skeleton obtained from PCres ('prior screening'). As in the pcalg-package, it is possible to restrict the maximum size of the conditioning sets, also to speed up computation.

Results

Simulations with randomly drawn graphs

To compare the different algorithms we simulated random GSEMs, by randomly sampling the sets D (defining the traits with direct genetic effects) and the covariance matrices Σ_G, combined with randomly drawn DAGs over the traits ($\mathcal{G}_{\mathcal{Y}}$). Traits were simulated for an existing population of 254 maize hybrids (Millet *et al.* 2016). Two replicates of each genotype were simulated. Given the 254×254 additive relatedness matrix A based on 50k SNPs, genetic effects were simulated such that $vec(\mathbf{G}) \sim \Sigma_G \otimes (ZAZ^t)$ (i.e. the vector of genetic effects for all the replicates, and all traits). File S4.1 provides further details, such as the magnitude of genetic (co)variances. We focus here on the comparison of

- PCgen based on the replicates, assuming $K = ZZ^t$ (i.e. ignoring A).
- PCres (replicates): PCres based on residuals from univariate GBLUP, again using only the replicates.
- PCres (means): PCres based on residuals from multivariate GBLUP, using genotypic means and the relatedness matrix A that was used to simulate the data.

Table S2 provides results for variations on these algorithms. 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 Kalisch and Bühlmann (2007) 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. File S4.2 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.

Simulation 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 0.97 and FPR around 0.03 (Table 1). In the first scenario, about a third of the actual edges between traits was not detected with PCgen (TDR ≈ 0.65, i.e., the proportion of true edges that was discovered). At the same time the number of false edges was very low, which is also reflected in high TPR values (the proportion of edges in the reconstruction that is true). In scenario 2, the TPR, FPR and TDR all increased. Hence, for denser graphs, more of the true edges were found, at the expense of a somewhat higher number of false edges.

PCres (replicates) outperformed PCres (means), in spite of the use of univariate GBLUP, and ignoring the actual relatedness matrix. Hence, 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, and, in line with the results of Kruijer *et al.* (2015), this error appeared lowest when using the replicates. The use of both the replicates and the marker-based GRM (i.e., assuming $K = ZAZ^t$, as the data were generated), further improved performance, but only slightly (Table S1, 'PCres-uni-RA'). Unsurprisingly, the MTM required for PCres (means) was computationally more demanding, and often not estimably for more than 4 traits. Motivated by this computational advantage and the statistical advantages mentioned in the Discussion, all analyses in the remainder will only consider PCgen and PCres based on replicates.

For the trait-to-trait relations, PCgen and PCres (replicates) had very similar performance in terms of TPR, TDR and FPR. However, PCgen substantially improved the orientation of these edges, as shown by the reduced SHD. This is not because the algorithm as such is better, but rather because more of the edges can be oriented from conditional independence tests, which in turn is a consequence of the addition of the edges $G \to Y_j$ to the graph over the traits (see again the example in Figure 2).

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 (Table 1).

Simulations using a crop-growth model

We also simulated data using the popular crop growth model APSIM-wheat (Keating *et al.* 2003; Holzworth *et al.* 2014). Compared to the preceding simulations this represents a more challenging 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 three replicates

	$\mathcal{G}_{\mathcal{Y}}$				$G \to Y_j$		
	TPR	FPR	TDR	SHD	TPR	FPR	TDR
Scenario 1 ($p = 4$)	($p_t = 1/3$)				($p_g = 0.5$)		
PCgen	0.647	0.006	0.981	0.442	0.982	0.026	0.995
PCres (replicates)	0.650	0.039	0.908	1.410			
PCres (means)	0.521	0.390	0.438	3.174			
Scenario 2 ($p = 4$)	($p_t = 0.5$)				($p_g = 0.5$)		
PCgen	0.804	0.033	0.986	1.246	0.976	0.031	0.995
PCres (replicates)	0.819	0.073	0.939	2.320			
PCres (means)	0.672	0.364	0.659	3.628			
Scenario 3 ($p = 20$)	($p_t = 0.1$)				($p_g = 0.3$)		
PCgen	0.895	0.002	0.985	6.806	0.969	0.018	0.991
PCres (replicates)	0.911	0.004	0.961	9.874			
Scenario 4 ($p = 100$)	($p_t = 0.01$)				($p_g = 0.1$)		
PCgen	0.959	0.001	0.942	27.288	0.976	0.022	0.943
PCres (replicates)	0.962	0.001	0.940	38.410			

Table 1. Performance of PCgen and residuals-based approaches, averaged over 500 simulated data-sets per scenario. Standard errors for the TPR, FPR and TDR were between 0.001 and 0.015. Standard errors for the SHD were around 0.06 (scenarios 1 and 2), 0.18 (scenario 3) and 0.28 (scenario 4). For the performance of other variants of PCgen and PCres in scenarios 1 and 2, see Table S2. In scenario 4, we used PCgen with the RG-test (PCgen-RG-uni); in the other scenarios we used the RC-test, with prior screening (PCgen-RC-screening). All acronyms are explained in Table 2 in File S2. PCres (replicates) and PCres (means) refer to PCres-uni-R and PC-multi-A.

each. The traits include seven primary traits, four secondary traits and yield (Y). File S5 provides further details, and trait acronyms are given in Table S3. 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 S1 (A) shows the summary graph [6], defining the causal effects from one time-step to the next (Peters *et al.* 2017). There are direct genetic effects on all of the primary traits, which have heritability 0.9. The genetic effects originate from 300 trait-specific QTLs, with randomly drawn effect sizes. There are no direct genetic effects on the secondary traits and yield.

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 (Figure S1 (B); highest p-value: $1.7 \cdot 10^{-4}$). These false positives seemed to be a consequence of the nonlinearities in the data-generating distribution, which are not accounted for in our tests. 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 S1 (C)), with an additional false arrow ($MGS - SP$). The standard PC-stable algorithm applied to all traits and QTLs led to many more errors (Figure S1 (D)), such as the false edge between GW and RUE, the missing edge $TFI \to FT$ and some incorrect orientations. These errors occurred because many conditional independencies between a trait and a QTL were not detected, leading to removal of the corresponding edges. This in turn led to problems in the remaining tests, where part of the genetic variance was not taken into account. We emphasize that all 300 QTLs were available to the PC-algorithm, and no other markers were provided. Hence, the poor performance in this case is really a consequence of the small effects, rather than the difficulty of QTL *detection*.

[6] This 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 (Richardson *et al.* 2002)), but for simplicity we nevertheless investigate to which extent we can reconstruct the summary graph, with observations taken at $t = T$.

Two case-studies

We now use PCgen to analyze real data from four field trials and one experiment in a phenotyping platform. In all network reconstructions we used a significance threshold of $\alpha = 0.01$. Reconstructions with $\alpha = 0.001$ are shown in Figures S2 and S4. Table S6 and Figure S5 contain p-values for the remaining edges. In all data-sets we removed traits that were derived as sums or ratios of other traits, rather than being directly measured. In particular, the maize data do not contain grain number, which was defined as the ratio of yield over grain weight. We return to this issue in the Discussion. The maize and rice data used below can be accessed at respectively https://doi.org/10.15454/IASSTN and https://doi.org/10.6084/m9.figshare.7964357.v1.

Maize First we analyze the field trials described in Millet *et al.* (2016) and Millet *et al.* (2019), involving 254 hybrids of maize (*Zea mays*). We consider a subset of four trials, representing four (out of a total of five) different environmental scenarios described in Millet *et al.* (2016). See Table S5 for an overview. The scenarios were derived from physiological knowledge, crop-growth

models and environmental sensors in the fields. Scenarios were defined as a combination of well-watered or water-deficient conditions (WW versus WD) and temperature. The latter was classified as "Cool" (average maximum and night temperature below respectively 33 and 20 degrees Celcius), "Hot" (above 33 and 20 degrees) or "Hot(days)" (max. temperature above 33, night temperature below 20). Most trials included 7 traits:

- three height traits, i.e., tassel height (TH), ear height (EH) and plant height (PH); the latter is missing in the Ner12R trial.
- two flowering time traits: anthesis (A) and silking (Sk), which are respectively male and female flowering.
- two yield related traits: grain weight (GW) and yield (Y).

Table S4 provides an overview of trait acronyms. Each trial was laid out as an alpha-lattice design, with either two or three replicates. Spatial trends and (in)complete block effects were estimated using the mixed model of Rodríguez-Álvarez *et al.* (2017) (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.

All traits have direct genetic effects, and traits mostly cluster according to their biological category (heigth, flowering and yield related), especially in the WW scenarios (Figure 4; panels A and B). In the Ner13W and Ner12R trials (B and C) there are edges between yield and respectively tassel- and plant-height, but these (conditional) dependencies are weak and disappear in the reconstruction with $\alpha = 0.001$ (Figure S2). Much stronger are the edges between yield and one or both of the flowering traits, in the water-deficit trials (C and D); the corresponding conditional independence tests gave highly significant p-values for all of the considered conditioning sets (Table S6). By contrast, in the trial without heat or drought stress (Kar12W), the $Y - Sk$ and $Y - A$ edges were already removed in the test conditioning only on the genetic effects; Figure S3 provides an illustration. The relation between yield and delay in silking in maize is well known (see e.g., Borrás *et al.* (2007) and Araus *et al.* (2012)). In the most stressed environment (Bol12R), there is an additional edge between plant height and silking. This may relate to the fact that the timing of anthesis determines the number of phytomeres (number of internodes and leaves) that a plant will generate, which in turns affects plant height (McMaster *et al.* 2005). The strong correlation between anthesis (A) and silking (Sk) may explain the presence of the edge $PH - Sk$ (rather than $PH - A$).

Finally, apart from the Bol12R trial there is never an edge between Y and GW, which seems due to the choice of the genetic material (giving little variation in grain weight) and the design of the trials (targeting stress around flowering time, rather than the grain filling period). See Millet *et al.* (2016) for further details. For all trials, the structure of the graphs is such that none of the between-trait edges can be oriented (technically, this is due to a lack of v-structures). However, for some of these edges, physiological knowledge clearly suggests a certain orientation, in particular for $Sk - Y$ and $GW - Y$.

The trials also illustrate the difference between the total genetic covariance (V_G) and the covariance among direct genetic effects, as defined by Σ_G. For most pairs of traits, the total genetic correlation[7] was between $\rho_g = 0.3 - 0.9$ (Table S7). The (total) genetic correlation between yield and silking was strongly negative in both WD trials (-0.44 and -0.61), and in the Bol12R trial also for yield and anthesis (-0.43). In all trials, genetic correlation with grain weight (GW) was negative for most traits, but not always significant. In the Kar12W trial for example, we found $\rho_g = -0.010$ for GW and PH, and $\rho_g = -0.435$ for GW and Sk (silking). In both cases, the two traits are d-separated in the graph (conditioning on $\{G\}$), but only for Sk the genetic covariance is significant ($p = 1.31 \cdot 10^{-9}$). While this may provide information about Σ_G, we recall that most often, the latter is not entirely identifiable.

As we have seen in the examples following equation (5), the existence of an edge between two traits in the graph does not necessarily imply a strong genetic correlation. In other words, having a shared genetic basis is not the same thing as the presence of a causal effect, found after conditioning on the genetic effects and other traits. In the Ner12R trial, for example, there is no edge between yield and grain weight, but a significant genetic correlation, while in the Bol12R trial it is the other way round.

Rice Next we analyze 25 traits measured on 274 *indica* genotypes of rice (*Oryza sativa*) under water-deficit, reported by Kadam *et al.* (2017). Three replicates of each genotype were phenotyped in a randomized complete block design, and block was included as a covariate in all conditional independence tests. Tests were restricted to conditioning sets of at most four traits. A first run of PCgen produced several inconsistencies in the genetic effects, i.e., traits with significantly positive heritability but without a partially directed path coming from the node G. We therefore applied the correction described in File S3, adding edges $G \to Y_j$ for all traits with this inconsistency, and then re-ran PCgen. The final reconstruction is given in Figure 5, where traits are grouped into three shoot morphological traits (blue), one physiological trait (rose), 13 root morphological traits (green), five root anatomical traits (gray) and three dry matter traits (orange).

After correcting the inconsistencies, there were nine traits without a direct genetic effect. Five of these (MRL, ART, RL2530, RL3035 and RL35) were completely isolated in the graph, without edges connecting to any other trait. All of these traits are related to either root length, or to the length of thicker roots, which contribute to drought adaptation under field conditions (Uga *et al.* 2013). However, as the experiment was done in pots, roots were constrained in their exploration range and therefore genotypic differences in root length would not translate into differential access to water and biomass (Poorter *et al.* 2012). Four other traits (TRWD, RW, RL0510 and RV) had at least one adjacent trait in the graph, but no direct genetic effect. At a lower significance level ($\alpha = 0.001$, Figure S4), direct genetic effects disappeared also for cumulative water transpiration (CWT), and for three root anatomical traits (RD, CD and SD). For RV (root volume), a direct genetic effect was only present with $\alpha = 0.001$, which was an artefact of the way the initial consistencies were resolved.

Traits related to root surface area (SA), root volume (RV) and roots with small diameter class (RL005, RL1015) had direct genetic effects and were connected amongst each other. As expected, traits related to root volume and area influenced root weight and total root weight density (RW, TRWD). In the reconstruction with $\alpha = 0.001$, cumulative water transpired (CWT) was affected by stem and leaf weight (SW, LW) and by RL0510, in agreement with physiological knowledge that water transpiration is influenced by water demand (related to the above-ground

[7] For better interpretability we report correlations, although we test for zero genetic covariance (see File S1.4).

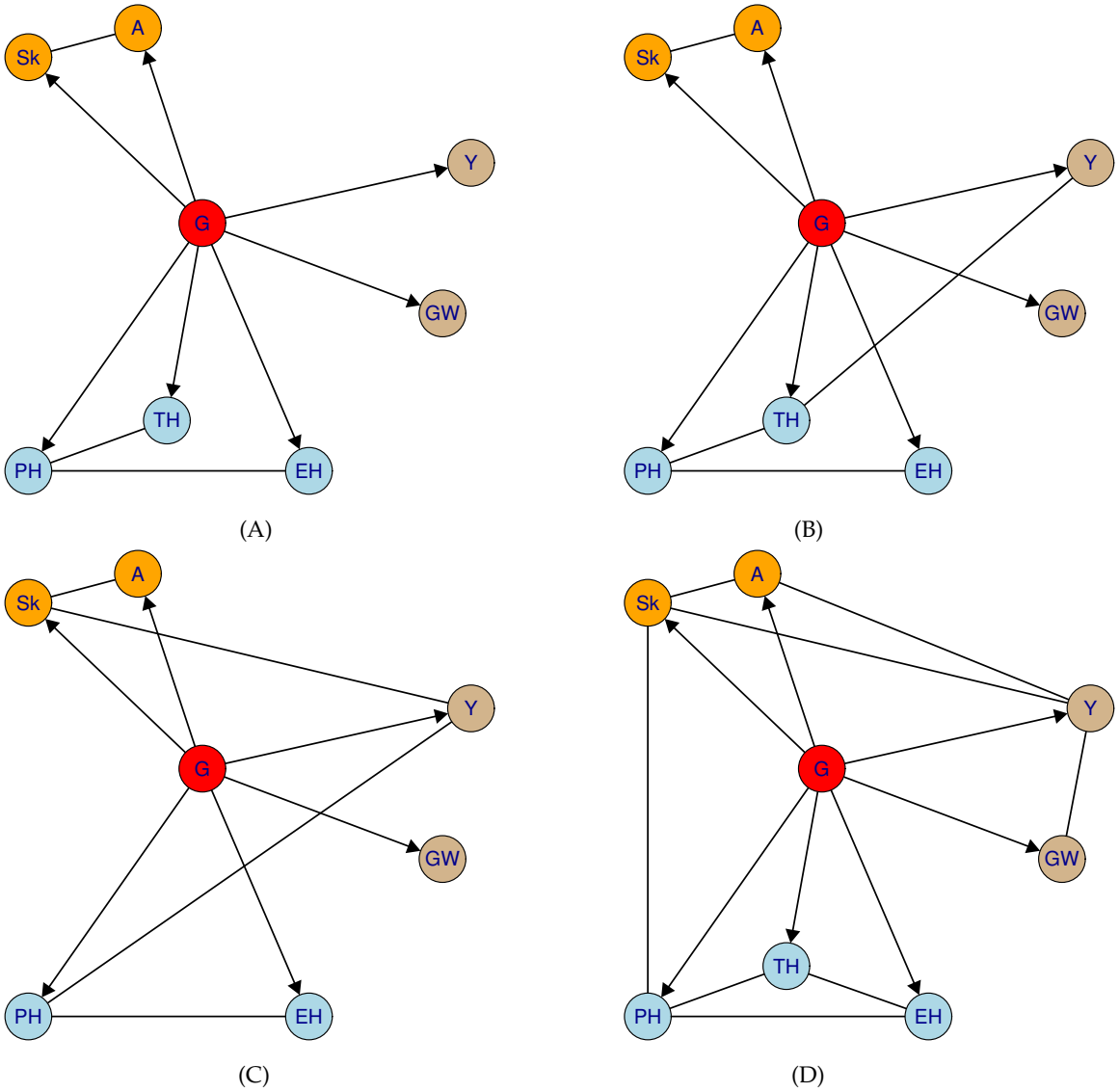

Figure 4. Estimated networks with $\alpha = 0.01$, for four of the DROPS field trials. Trait categories are flowering (orange), height (blue) and yield (brown). Each trial represents a different environmental scenario, being well-watered (WW) or water-deficit (WD), and different temperatures (see text). (A) Kar12W, (WW, Cool), (B) Ner13W (WW, Hot), (C) Ner12R (WD, Hot (days)), (D) Bol12R (WD, Hot). Trait acronyms are given in Table S4.

biomass) and water supply (related to the roots' water uptake capacity). The corresponding edges were also present in the recontruction with $\alpha = 0.01$, where however they could not be oriented because of the denser network (in particular, the presence of $G \to CWT$). Root anatomical traits (LMXD, SD, CD and RD) appeared as a separate module, not related to the plant water dynamics, suggesting that root anatomy had a smaller impact on water uptake, compared to root biomass.

Statistical properties of PCgen

We now investigate a number of statistical issues: the assumptions required for asymptotic consistency of PCgen, the assumptions required for faithfulness, and properties of the conditional independence tests. Readers primarily interested in the application of PCgen could skip this section and continue with the Discussion. Proofs of Theorems 1-6 are given in Appendix A.

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 Spirtes *et al.* (2001) (for low dimensions) and Kalisch and Bühlmann (2007) (for high dimensions). These authors distinguished between 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 A.8) is a straightforward adaptation of Pearl's proof for general SEMs (Pearl 2009).

Theorem 1 *Suppose we have a GSEM as defined by equation (4), with a graph $\mathcal{G}$ as defined in the Materials and Methods section, and satisfying assumptions 1-4. 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:

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:*

$$\mathbf{Y_j} \perp\!\!\!\perp \mathbf{Y_k} | \{\mathbf{Y_S}, \mathbf{G}\} \implies Y_j \, dsep(\mathcal{G}) \, Y_k | \{Y_S, G\} \tag{9}$$

$$\mathbf{Y_j} \perp\!\!\!\perp \mathbf{G} | \{\mathbf{Y_S}\} \implies Y_j \, dsep(\mathcal{G}) \, G | \{Y_S\}, \tag{10}$$

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.

Faithfulness For our first faithfulness condition (9), it suffices to have faithfulness for the graph without genetic effects. A necessary (but not sufficient) condition for this is that contributions from different paths do not cancel out (Appendix A.4):

Theorem 3 *Let $P_{\mathbf{Y}|\mathbf{U}}$ 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 $\mathbf{Y_j} \perp\!\!\!\perp \mathbf{Y_k} | \{\mathbf{Y_S}, \mathbf{G}\}$ is equivalent with $\mathbf{Y_j} \perp\!\!\!\perp_{P_{\mathbf{Y}|\mathbf{U}}} \mathbf{Y_k}$, and $Y_j \, dsep(\mathcal{G}) \, Y_k | \{Y_S, G\}$ is equivalent with $Y_j \, dsep(\mathcal{G}_\mathcal{Y}) \, Y_k | Y_S$. Consequently, (9) holds if*

$$\mathbf{Y_j} \perp\!\!\!\perp_{P_{\mathbf{Y}|\mathbf{U}}} \mathbf{Y_k} | \{\mathbf{Y_S}\} \implies Y_j \, dsep(\mathcal{G}_\mathcal{Y}) \, Y_k | Y_S. \tag{11}$$

Consequently, we can rephrase (9) in terms of a faithfulness assumption for the analogous SEM without genetic effects.

The second faithfulness statement (equation (10)) 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 (10) also holds for non-empty conditioning sets.

Theorem 4 *Suppose we have a GSEM sastifying Assumptions 1-5, and faithfulness for the graph without genetic effects, given by (11). Then (10) holds for $S = \emptyset$, i.e., marginal independence implies d-separation of Y_j and G.*

Hence, faithfulness involving $\mathbf{Y_j}$ and $\mathbf{G}$ requires (at least) absence of collinearities between genetic effects, as well as faithfulness for the corresponding SEM without genetic effects.

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 assumptions made for our conditional independence tests may not always hold, introducing approximation errors. Second, even without these errors, the probabilities of type-I and type-II errors still need to converge to zero with increasing sample size. This is well known for the PC-algorithm with independent Gaussian data (Kalisch and Bühlmann 2007), but more difficult to establish in the presence of genetic effects. Here we address the first issue, leaving the second for future work.

Our tests [8] for conditional independence statements (A) and (B) (i.e. $\mathbf{Y_j} \perp\!\!\!\perp \mathbf{G} | \{\mathbf{Y_S}\}$ and $\mathbf{Y_j} \perp\!\!\!\perp \mathbf{Y_k} | \{\mathbf{Y_S}, \mathbf{G}\}$) rely on the conditional distributions of respectively $\mathbf{Y_j}$ and $vec([\mathbf{Y_j}, \mathbf{Y_k}])$, given the observed $\mathbf{Y_S}$:

$$\mathbf{Y_j} | vec(\mathbf{Y_S}) = vec(\tilde{y}_S) \sim N(\mu_{j|S}, \Sigma_{j|S})$$

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

[8] For statement (B) we will focus on the RC-test. The RG-test 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., Henderson (1975)).

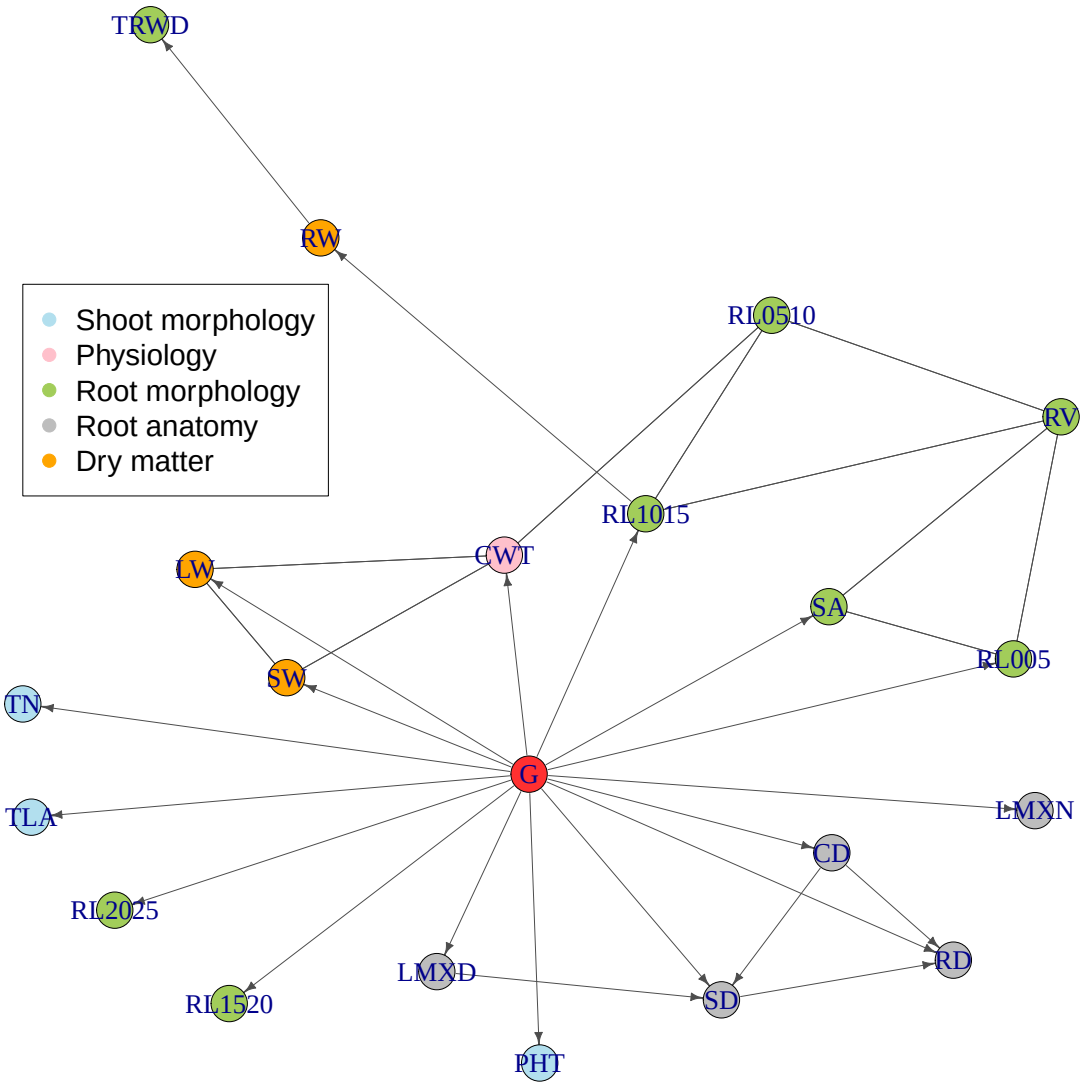

Figure 5. PCgen-reconstruction for the rice-data from Kadam *et al.* (2017), with $\alpha = 0.01$. Five traits (MRL, ART, RL2530, RL3035 and RL35) are not shown, as they were completely isolated in the graph, without any connections to other traits or G.

The normality of these distributions directly follows from the assumed normality of the genetic and residual effects. We made the following assumptions about the form of their covariance and mean:

- The covariance matrix $\Sigma_{j|S}$ is that of a single trait mixed model with the same relatedness matrix K assumed in the GSEM, i.e.,

$$\Sigma_{j|S} = \sigma^2_G(j|S)K + \sigma^2_E(j|S)I_n, \tag{12}$$

  for some variance components $\sigma^2_G(j|S)$ and $\sigma^2_E(j|S)$.
- The covariance matrix $\Sigma_{jk|S}$ is that of a bivariate MTM, again with the same K assumed in the GSEM:

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

  for some 2×2 matrices $V_G(jk|S)$ and $V_E(jk|S)$.
- The conditional means $\mu_{j|S}$ and $\mu_{k|S}$ are linear regressions over the conditioning traits:

$$\mathbf{X}B\gamma_j + \tilde{y}_S\beta^{(j)}_S, \quad \mathbf{X}B\gamma_k + \tilde{y}_S\beta^{(k)}_S, \tag{14}$$

  where $\mathbf{X}B\gamma_j$ is the marginal mean of $\mathbf{Y_j}$ (see (8)), and $\beta^{(j)}_S$ and $\beta^{(k)}_S$ are $|S| \times 1$ vectors of regression coefficients.

In the following theorems we show that when $K = ZZ^t$, assumptions (12) and (13) always hold, i.e. they directly follow from our GSEM model.

Theorem 5 *When $K = ZZ^t$, the distribution of $vec([\mathbf{Y_j}, \mathbf{Y_k}])|vec(\mathbf{Y_S})$ has covariance of the form given by (13), i.e., that of a bivariate MTM. Moreover, under faithfulness condition (9), the residual covariance in the MTM is zero if and only if $\mathbf{Y_j} \perp\!\!\!\perp \mathbf{Y_k}|\{\mathbf{Y_S}, \mathbf{G}\}$.*

Theorem 6 *Suppose we have a GSEM as described in Theorem 1, with $K = ZZ^t$. Then the covariance 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, assuming the faithfulness condition (10) and $\Sigma_G[D, D]$ of full rank (Assumption 5), $\sigma^2_G(j) = \sigma^2_G(j|\emptyset)$ is zero if and only if $\mathbf{Y_j} \perp\!\!\!\perp \mathbf{G}$.*

Apart from the covariance structure, these theorems address the correctness of our tests. In particular, Theorem 5 shows that the residual covariance in the distribution of $\{\mathbf{Y_j}, \mathbf{Y_k}\}|\mathbf{Y_S}$ is indeed the right quantity to test statement (B). Similarly, the genetic variance in the conditional distribution of $\mathbf{Y_j}|\mathbf{Y_S}$ is the relevant thing for testing (A). This appears to be true for any conditioning set S, although we could prove it only for the empty conditioning set, because faithfulness is required (which we also established only for $S = \emptyset$; see Theorem 4).

The situation is different for assumption (12), regarding the conditional means: even when $K = ZZ^t$, it holds for certain conditioning sets and not for others. We illustrate this with 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 $G \to Y_1 \to Y_2$. 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. In summary, 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). Despite the approximation error occurring sometimes for the conditional means, our tests still seem to perform reasonable, as shown in the simulations above.

Finally, we briefly discuss how the approximation could be improved. In general, the conditional mean is a function of the genetic and residual covariances between $\mathbf{Y_j}$ and $\mathbf{Y_S}$. In Appendix A.6 (equation (21)) 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 models $\eta_{j|S}$ as a linear regression on $\tilde{y}_S$. This approximation could probably be improved if 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 an MTM for $|S| + 1$ traits, which for large S is statistically and computationally challenging, unless pairwise or other approximations are applied (Furlotte and Eskin 2015; Joo *et al.* 2016). Moreover, it seems unclear how $\hat{\eta}_{j|S}$ should be incorporated in the tests.

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 have 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 led us to develop the PCgen algorithm, which tests conditional independence between traits in the presence of genetic effects, and also conditional independence between traits and genetic effects. We showed that the presence of a direct genetic effect can in principle be tested, just like the direct (fixed) 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 the linear simulations as well as in the rice data, our tests could indeed identify the absence of many direct genetic effects. By contrast, in the APSIM simulations and maize data all traits had such effects. In the latter case, this could be for biological reasons, i.e., the genetic variance of each trait might really be 'unique' to some degree. However, the APSIM results showed that nonlinearities could increase the false positive rate in the edges $G \to Y_j$, which may be avoided in future versions with better conditional independence tests. Such tests might also allow for non-Gaussian data.

In our simulations PCgen outperformed existing approaches, regarding the reconstruction of between trait relations. 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 to be computationally more efficient: depending on the choice of independence tests and the sparseness of the network, it can analyze around 10-50 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.

As for the original PC-algorithm, PCgen is most efficient for sparse graphs, i.e. when each trait is connected to only a few other traits, and when there are few direct genetic effects. But even if this is not the case PCgen still has an advantage over existing approaches: by incorporating the genetic effects in the PC-algorithm, we do not need to fit an 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 conditional on other traits (e.g., yield given plant height) is a common and natural thing to do (Stephens 2013). 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.

Dealing with derived traits

We analyzed the maize and rice data-sets using the traits as they were measured, without adding 'derived' traits defined by ratios, sums or differences of the original traits. Because such derived traits are not measured themselves there is no error associated with them, apart from 'copies' of errors in the original trait. For example, if there is much variation in leaf weight but little in the total weight of a plant, the derived trait 'leaf weight ratio' will be essentially a copy of the original leaf weight trait. This can violate our assumption of faithfulness and lead to errors in the reconstruction; see Figure 7 in Appendix A for an example. Sometimes, derived traits are biologically highly relevant. It may then be desirable to include them in the analysis, and omit some of the original traits. Alternatively, derived traits may be added after running PCgen. For example, we may extend the reconstructed graph with the node $Y_3 := Y_1 + Y_2$ and edges $Y_1 \rightarrow Y_3$ and $Y_2 \rightarrow Y_3$, provided that biologically this makes sense.

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 certain applications this assumption can indeed be restrictive, but seems to be inevitable. Suppose traits were measured in different experiments, or residual errors would mostly come from measurement errors. Then there would be no propagation of residual errors, and the reconstruction would rely completely on how the *genetic* effects are propagated through the network. The GSEM model (4) would need to be replaced by $\mathbf{Y} = \mathbf{X}B + \mathbf{U} + \mathbf{E}$ and $\mathbf{U} = \mathbf{U}\Lambda + \tilde{\mathbf{U}}$, where $\tilde{\mathbf{U}}$ in a sense models direct genetic effects, reminiscent of the genomic networks of Töpner *et al.* (2017). However, if data are actually generated by equation (4), these can only partially reconstruct the direct genetic effects, even without any type-I or type-II error in the tests (see File S7.2). Biologically, the genomic networks have a different interpretation: for example, we would assume that *the genetic component* in high blood-pressure causes some cardiovascular disease, rather than high blood-pressure itself. The alternative model implies that the observed traits have diagonal residual covariance, instead of the matrix $\Gamma^t \Sigma_E \Gamma$ obtained under assumption 1 (see equation (5)). However, the latter matrix turned out to be essential for network reconstruction (see e.g. Theorems 5-6 above). 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 appears rather unrealistic.

A relevant alternative approach here is that of invariance causal prediction (Peters *et al.* 2016), which infers causal effects that are consistent across several experiments, but still requires all traits to be measured in each experiment (as well as low genotype-by-environment interaction).

Replicates versus means

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:

- Performance under model misspecification: different types of genetic effects could in theory be represented by introducing multiple genetic nodes, with conditional independence tests that can distinguish between these effects. But this seems difficult in practice due to the computational requirements and lack of statistical power (Uhler *et al.* 2013; Blair *et al.* 2012; Kruijer 2016). For this reason it seems, previous work on network reconstruction used genotypic means and an additive GRM. For the analysis of a single trait however, Kruijer (2016) showed that broad-sense heritability estimates (obtained with $K = ZZ^t$) 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. It seems plausible that this robustness extends to the multivariate models considered here, for example when direct genetic effects are driven by different sets of QTLs, leading to trait-specific relatedness matrices.
- Higher power: estimates of (total) genetic variance based on replicates are typically more accurate than marker-based estimates based on genotypic means (Kruijer *et al.* 2015; Visscher and Goddard 2015), and the use of replicates is therefore also likely to improve hypothesis testing. For the reconstruction of trait-to-trait relations with PCres, our simulations indeed suggest that replicates give more power. Mixed models with *both* replicates and a GRM might further increase power if the true architecture is really additive (Kruijer *et al.* (2015)), but also these models lead to biased inference if the actual architecture is different (Kruijer (2016)).
- 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 not, and an alternative test might be required.

Apart from these statistical issues, there is also a computational advantage: the test for $\mathbf{Y_j} \perp\!\!\!\perp \mathbf{G} | \mathbf{Y_S}$ can be based on standard ANCOVA, 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 $K = ZZ^t$.

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 (Rodríguez-Álvarez *et al.* 2017; Flaxman *et al.* 2015).

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 not rejected (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, but in the sample version of PC(gen), $adj(Y_j)$ or $adj(Y_k)$ may become smaller than the corresponding adjacency sets in the true graph, and the algorithm may therefore not perform an essential independence test. See Colombo and Maathuis (2014) 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 (Meinshausen and Bühlmann 2010; Stekhoven *et al.* 2012) or other sample-splitting techniques (Töpner *et al.* 2017), but these are often too stringent and require an additional exchangeability assumption (Bühlmann *et al.* 2013; Meinshausen *et al.* 2016). Moreover, these approaches do not provide a level of confidence for *specific* edges. For example, an edge present in 60% of all subsamples may appear to be present in the true graph with a probability of 0.6, but in fact there is no justification for such a statement.

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 (Ghosal and van der Vaart 2017), there do not seem to be results yet regarding the correct coverage of posteriors in these models.

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 is an interest in some kind of intervention (Valente *et al.* 2013, 2015). 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 (Pearl 2009; Spirtes *et al.* 2001) can then predict the joint distribution of the system *after* the intervention:

$$p_{\mathbf{Y_j}:=c}(\mathbf{G}, \mathbf{Y_{-j}}) = p(\mathbf{G}) \prod_{j' \neq j} p(\mathbf{Y_{j'}} | pa(\mathbf{Y_{j'}}), \mathbf{G_{j'}}), \qquad (15)$$

where $pa(\mathbf{Y_{j'}})$ is the set of parents of $Y_{j'}$. This is generally different from the distribution

$$p(\mathbf{G}, \mathbf{Y_{-j}} | \mathbf{Y_j} = c), \qquad (16)$$

obtained from conditioning on $\mathbf{Y_j} = c$, *prior* to the intervention (see e.g., Peters *et al.* (2017)). In other words, *conditioning* is not the same as *doing* (intervening). See File S7.3 for an example. An exception occurs however when we intervene on a root node, in which case (15) and (16) are the same (see Peters *et al.* (2017), p. 110).

In absence of interventions on the traits, we can think of genomic prediction in terms of an intervention on the node G. Because the latter is a root node by definition, standard genomic prediction methods can in principle have optimal performance (Valente *et al.* 2013). 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.

Finally, a technical obstacle for applying our networks to genomic prediction is the identifiability issue mentioned in the introduction. PCgen (and the PC-algorithm in general) typically outputs a *partially* directed graph, several DAGs being compatible with this graph. This is particularly problematic for edges between the traits in D (the traits with direct genetic efffects). For the traits with only indirect genetic effects, it is in principle possible to estimate how much of the genetic variance originates from a partiular trait in D, the result being independent of the chosen DAG. This would first require estimates of the (total) genetic covariance among traits in D, obtained either by fitting an MTM, or by an approximation (as in Furlotte and Eskin (2015)).

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, it may be possible to construct better tests, especially for nonlinear structural models and non-Gaussian error distributions. The recent work of Pfister *et al.* (2018) seems particularly relevant here. 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 (Spirtes *et al.* 2001; Kalisch and Bühlmann 2007), 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[9]. 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

[9] The pcalg-package (Kalisch *et al.* 2012) 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.

generally. In particular, we hope that the ideas developed here provide a first step towards the more amibitious goal of modeling multiple traits through time, simultanesouly for many environments. A first generalization would be to replace the PC-algorithm with other constraint-based algorithms, in particular FCI and RFCI (Spirtes *et al.* 2001; Colombo *et al.* 2012). These have the advantage that the causal sufficiency assumption (no latent variables) can be dropped or considerably weakened. The presence or absence of direct genetic effects could also be incorporated in (empirical) Bayesian approaches for genetic network reconstruction, or in invariant causal prediction (Peters *et al.* 2016). It might also be possible to extend the approach of Stephens (2013), and focus only on the detection of traits with direct genetic effects. Finally, SEM with genetic effects might be useful as covariance models in multi-trait GWAS, as an alternative to unstructured (Zhou and Stephens 2014) or low-rank (Millet *et al.* 2016) models.

Author contributions

WK developed the PCgen algorithm. PB developed the package `pcgen`, 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, and analyzed the rice data. 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.

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). MX was funded by project MTM2017-82379-R (AEI/FEDER, UE), by the Basque Government through the BERC 2018-2021 program and by the Spanish Ministry of Science, Innovation and Universities (BCAM Severo Ochoa accreditation SEV-2017-0718). 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 providing and interpretating the maize data.

Literature Cited

Araus, J. L., M. D. Serret, and G. Edmeades, 2012 Phenotyping maize for adaptation to drought. Frontiers in physiology **3**: 305.

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

Blair, R. H., D. J. Kliebenstein, and G. A. Churchill, 2012 What can causal networks tell us about metabolic pathways? PLoS Comput Biol **8**: e1002458.

Borrás, L., M. E. Westgate, J. P. Astini, and L. Echarte, 2007 Coupling time to silking with plant growth rate in maize. Field Crops Research **102**: 73–85.

Bühlmann, P., P. Rütimann, and M. Kalisch, 2013 Controlling false positive selections in high-dimensional regression and causal inference. Statistical methods in medical research **22**: 466–492.

Bustos-Korts, D., 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.

Calus, M. P. and R. F. Veerkamp, 2011 Accuracy of multi-trait genomic selection using different methods. Genetics Selection Evolution **43**: 26.

Casadebaig, P., B. Zheng, S. Chapman, N. Huth, R. Faivre, *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 .

Chaibub Neto, E., C. T. Ferrara, A. D. Attie, and B. S. Yandell, 2008 Inferring causal phenotype networks from segregating populations. Genetics **179**: 1089–1100.

Chickering, D. M., 2002 Learning equivalence classes of bayesian-network structures. J. Mach. Learn. Res. **2**: 445–498.

Colombo, D. and M. H. Maathuis, 2014 Order-independent constraint-based causal structure learning. Journal of Machine Learning Research **15**: 3741–3782.

Colombo, D., M. H. Maathuis, M. Kalisch, and T. S. Richardson, 2012 Learning high-dimensional directed acyclic graphs with latent and selection variables. Ann. Statist. **40**: 294–321.

Cooper, M., C. D. Messina, D. Podlich, L. R. Totir, A. Baumgarten, *et al.*, 2014 Predicting the future of plant breeding: complementing empirical evaluation with genetic prediction. Crop Pasture Sci. **65**: 311–336.

Dempster, A. P., N. M. Laird, and D. B. Rubin, 1977 Maximum likelihood from incomplete data via the em algorithm. Journal of the royal statistical society. Series B (methodological) pp. 1–38.

Flaxman, S. R., D. B. Neill, and A. J. Smola, 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://www.sethrf.com/files/gp-depend.pdf .

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

Gao, B. and Y. Cui, 2015 Learning directed acyclic graphical structures with genetical genomics data. Bioinformatics **31**: 3953–3960.

Ghosal, S. and A. van der Vaart, 2017 *Fundamentals of Nonparametric Bayesian Inference*. Cambridge Series in Statistical and Probabilistic Mathematics, Cambridge University Press.

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

Golub, G. H. and C. F. Van Loan, 2012 *Matrix computations*, volume 3. JHU Press.

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

Henderson, C. R., 1975 Best linear unbiased estimation and prediction under a selection model. Biometrics pp. 423–447.

Holzworth, D. P., N. I. Huth, E. J. Zurcher, N. I. Herrmann, G. McLean, *et al.*, 2014 Apsim–evolution towards a new generation of agricultural systems simulation. Environmental Modelling & Software **62**: 327–350.

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

Joo, J. W. J., E. Y. Kang, E. Org, N. Furlotte, B. Parks, *et al.*, 2016 Efficient and accurate multiple-phenotype regression method for high dimensional data considering population structure. Genetics **204**: 1379–1390.

Kadam, N., A. Tamilselvan, L. M. F. Lawas, C. Quinones, R. Bahuguna, *et al.*, 2017 Genetic control of plasticity in root morphology and anatomy of rice in response to water-deficit. Plant Physiology .

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

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

Keating, B. A., P. S. Carberry, G. L. Hammer, M. E. Probert, M. J. Robertson, *et al.*, 2003 An overview of APSIM, a model designed for farming systems simulation. Eur J Agron **18**: 267–288.

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

Kruijer, W., 2016 Misspecification in mixed-model based association analysis. Genetics **202**: 363–366.

Kruijer, W., M. P. Boer, M. Malosetti, P. J. Flood, B. Engel, *et al.*, 2015 Marker-based estimation of heritability in immortal populations. Genetics **199**: 379–398.

Lauritzen, S. L., 1996 *Graphical Models*. Oxford Statistical Science Series, Oxford University Press, New York, USA.

Lynch, M. and B. Walsh, 1998 *Genetics and Analysis of Quantitative Traits*. Sinauer Associates, first edition.

Maathuis, M. H., 2014 Graphical models and causality: Directed acyclic graphs (dags) and conditional (in)dependence.

Maathuis, M. H. and P. Nandy, 2016 A review of some recent advances in causal inference. Handbook of Big Data p. 387.

McMaster, G. S., W. Wilhelm, and A. Frank, 2005 Developmental sequences for simulating crop phenology for water-limiting conditions. Australian Journal of Agricultural Research **56**: 1277–1288.

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

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

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

Millet, E. J., W. Kruijer, A. Coupel-Ledru, S. Alvarez Prado, L. Cabrera-Bosquet, *et al.*, 2019 Genomic prediction of maize yield across european environmental conditions. Nature Genetics .

Moore, A. J., E. D. Brodie III, and J. B. Wolf, 1997 Interacting phenotypes and the evolutionary process: I. direct and indirect genetic effects of social interactions. Evolution **51**: 1352–1362.

Neto, E. C., A. T. Broman, M. P. Keller, A. D. Attie, B. Zhang, *et al.*, 2013 Modeling causality for pairs of phenotypes in system genetics. Genetics pp. genetics–112.

Pearl, J., 2009 *Causality*. Cambridge university press.

Pearl, J., 2014 *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Elsevier.

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.

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

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

Petersen, K. B., M. S. Pedersen, *et al.*, 2008 The matrix cookbook. Technical University of Denmark **7**: 510.

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

Poorter, H., J. Bühler, D. van Dusschoten, J. Climent, and J. A. Postma, 2012 Pot size matters: a meta-analysis of the effects of rooting volume on plant growth. Functional Plant Biology **39**: 839–850.

Reynolds, M. and P. Langridge, 2016 Physiological breeding. Curr. Opin. Plant Biol. **31**: 162–171.

Richardson, T., P. Spirtes, *et al.*, 2002 Ancestral graph markov models. The Annals of Statistics **30**: 962–1030.

Rodríguez-Álvarez, M. X., M. P. Boer, F. A. van Eeuwijk, and P. H. Eilers, 2017 Correcting for spatial heterogeneity in plant breeding experiments with p-splines. Spatial Statistics **23**: 52 – 71.

Scutari, M., P. Howell, D. J. Balding, and I. Mackay, 2014 Multiple quantitative trait analysis using bayesian networks. Genetics **198**: 129–137.

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

Spirtes, P., C. Glymour, and R. Scheines, 2001 *Causation, Prediction, and Search, Second Edition (Adaptive Computation and Machine Learning)*. A Bradford Book, second edition edition.

Stekhoven, D. J., I. Moraes, G. Sveinbjörnsson, L. Hennig, M. H. Maathuis, *et al.*, 2012 Causal stability ranking. Bioinformatics **28**: 2819–2823.

Stephens, M., 2013 A unified framework for association analysis with multiple related phenotypes. PLoS ONE **8**: e65245.

Töpner, K., G. J. M. Rosa, D. Gianola, and C.-C. Schön, 2017 Bayesian networks illustrate genomic and residual trait connections in maize (zea mays l.). G3: Genes, Genomes, Genetics **7**: 2779–2789.

Tsamardinos, I., L. E. Brown, and C. F. Aliferis, 2006 The max-min hill-climbing bayesian network structure learning algorithm. Machine learning **65**: 31–78.

Uga, Y., K. Sugimoto, S. Ogawa, J. Rane, M. Ishitani, *et al.*, 2013 Control of root system architecture by DEEPER ROOTING 1 increases rice yield under drought conditions. Nat. Publ. Gr. **45**: 1097–1102.

Uhler, C., G. Raskutti, P. Bühlmann, B. Yu, *et al.*, 2013 Geometry of the faithfulness assumption in causal inference. The Annals of Statistics **41**: 436–463.

Valente, B. D., G. Morota, F. Peñagaricano, D. Gianola, K. Weigel, *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.

Valente, B. D., G. J. M. Rosa, G. de los Campos, D. Gianola, and M. A. Silva, 2010 Searching for recursive causal structures in multivariate quantitative genetics mixed models. Genetics **185**: 633–644.

Valente, B. D., G. J. M. Rosa, D. Gianola, X.-L. Wu, and K. Weigel, 2013 Is structural equation modeling advantageous for the

genetic improvement of multiple traits? Genetics **194**: 561–572.

van Eeuwijk, F. A., D. Bustos-Korts, E. J. Millet, M. P. Boer, W. Kruijer, *et al.*, 2018 Modelling strategies for assessing and increasing the effectiveness of new phenotyping techniques in plant breeding. Plant Science .

Visscher, P. M. and M. E. Goddard, 2015 A general unified framework to assess the sampling variance of heritability estimates using pedigree or marker-based relationships. Genetics **199**: 223–232.

Wright, S., 1921 Correlation and causation. Journal of Agricultural Research pp. 557–585.

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

Zwiernik, P., C. Uhler, and D. Richards, 2014 Maximum likelihood estimation for linear gaussian covariance models.

Appendix A. Faithfulness, conditional distributions and proofs of Theorems 1-6

A.1. Overview of graph theoretic definitions

Definitions of for example d-separation and CPDAGs can be found in many books and articles on graphical models and causal inference; see for example Lauritzen (1996); Pearl (2009); Spirtes *et al.* (2001); Kalisch and Bühlmann (2007). The following summary was inspired by Shipley (2016) and Maathuis (2014).

- 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 the same node, we have a *directed path*. A path that is not directed is an *undirected* or *non-directed* path.
- 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 $Y_k \to Y_j$.
- A *directed acyclic graph* (DAG) is a directed graph without any directed cycle. When a graph underlying a SEM is a DAG, the SEM is said to be *recursive*.
- $pa(Y_j)$ is the set of nodes Y_k for which there is a directed edge $Y_k \to Y_j$; 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 $Y_k \to Y_j$, $Y_j \to Y_k$ or $Y_k - Y_j$.
- If in a DAG $\mathcal{G}$ there is a directed path from Y_j to Y_k, then Y_k is a *descendant* of Y_j, and Y_j is an *ancestor* of Y_k.
- 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. If there *is* an edge between A and C, 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 $Y_j \in U$ and $Y_k \in V$.
- 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. 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 ($A \to B \leftarrow C$). Any DAG in the class can be used to represent the class. But instead of picking an arbitrary DAG, an equivalence class can also be represented by a *completed partially directed acyclic graph* (CPDAG). 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' (Kalisch and Bühlmann 2007). 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 $A - B$ in the PDAG, the equivalence class contains at least one DAG with $A \to B$ and at least one with $B \to A$. Chickering (2002) showed that CPDAGs represent equivalence classes uniquely.

A.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 V_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 Spirtes *et al.* (2001) and define the $p \times 1$ column vector γ_j with elements ($l = 1, \ldots, p$)

$$\begin{aligned} \gamma_{j,l} &= \sum_{\pi \in \Pi_{lj}} L(\pi) \quad \text{if } Y_l \in V_j \\ &= 0 \qquad\qquad \text{otherwise.} \end{aligned} \tag{17}$$

A.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 (8) 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:

$$Cov(\mathbf{Y_j}, \mathbf{Y_k}^t) = E[(\mathbf{Y_j} - \mathbf{X}B\gamma_j)(\mathbf{Y_k} - \mathbf{X}B\gamma_k)^t] = (\gamma_j^t \Sigma_G \gamma_k)K + (\gamma_j^t \Sigma_E \gamma_k) I_n, \tag{18}$$

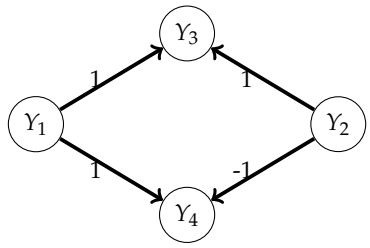

Figure 6. An example of a SEM where faithfulness does not hold, because the contributions to the covariance from the treks $Y_3 \leftarrow Y_1 \rightarrow Y_4$ and $Y_3 \leftarrow Y_2 \rightarrow Y_4$ cancel out. If Y_1 and Y_2 are Gaussian with equal error variances, it follows that for every individual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

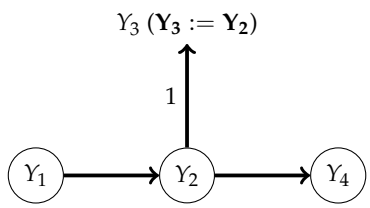

Figure 7. An example of a SEM where faithfulness does not hold, because the variance of the error variables in E_2 is zero. The random vectors $\mathbf{Y_1}$ and $\mathbf{Y_4}$ are conditionally independent given $\mathbf{Y_3}$, but in the graph, the nodes Y_1 and Y_4 are not d-separated by Y_3.

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 (18), 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 Spirtes *et al.* (2001) (Lemma 3.1.6), or Lynch and Walsh (1998) (Appendix 2). Using standard expressions for multivariate Gaussian distributions, this implies that

$$Cov(\mathbf{Y}[i,j], \mathbf{Y}[i,k] \mid \mathbf{Y}[i,S]) = \gamma_j^t \Sigma_E \gamma_k - (\gamma_j^t \Sigma_E \Gamma_S)(\Gamma_S^t \Sigma_E \Gamma_S)^{-1}(\Gamma_S^t \Sigma_E \gamma_k). \tag{19}$$

A.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 $Y_1 \rightarrow Y_2$, $Y_1 \rightarrow Y_3$ and $Y_2 \rightarrow Y_3$, 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 (17) is never zero, i.e., $\gamma_{j,l} = 0$ only for $Y_l \notin V_j$. We will refer to this as the **path coefficients condition**.

A.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 Peters (2012). 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 (Spirtes *et al.* 2001). 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 (19) it follows that for Gaussian linear SEM, faithfulness is equivalent with

$$\gamma_j^t \Sigma_E \gamma_k - (\gamma_j^t \Sigma_E \Gamma_S)(\Gamma_S^t \Sigma_E \Gamma_S)^{-1}(\Gamma_S^t \Sigma_E \gamma_k) = 0 \implies Y_j \text{ and } Y_k \text{ d-separated by } Y_S. \tag{20}$$

A.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 (7) that $\mathbf{Y_j}|\mathbf{Y_S} = \tilde{y}_S$ is multivariate normal with mean and covariance

$$\mu_{j|S} = (\mathbf{X}B\Gamma)[,j] + \Sigma_{j,S}\Sigma_S^{-1}vec(\tilde{y}_S - (\mathbf{X}B\Gamma)[,S]) = \mathbf{X}B\gamma_j + \Sigma_{j,S}\Sigma_S^{-1}vec(\tilde{y}_S - \mathbf{X}B\Gamma_S), \tag{21}$$

$$\Sigma_{j|S} = \Sigma_j - \Sigma_{j,S}\Sigma_S^{-1}\Sigma_{j,S}^t, \tag{22}$$

where

$$\Sigma_{j,S} = (\Gamma^t\Sigma_G\Gamma)[j,S] \otimes K + (\Gamma^t\Sigma_E\Gamma)[j,S] \otimes I_n = (\gamma_j^t\Sigma_G\Gamma_S) \otimes K + (\gamma_j^t\Sigma_E\Gamma_S) \otimes I_n, \tag{23}$$

$$\Sigma_S = (\Gamma_S^t\Sigma_G\Gamma_S) \otimes K + (\Gamma_S^t\Sigma_E\Gamma_S) \otimes I_n, \tag{24}$$

$$\Sigma_j = (\Gamma^t\Sigma_G\Gamma)[j,j]K + (\Gamma^t\Sigma_E\Gamma)[j,j]I_n = (\gamma_j^t\Sigma_G\gamma_j)K + (\gamma_j^t\Sigma_E\gamma_j)I_n. \tag{25}$$

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 (7) in the main text we also obtain the conditional distribution

$$vec([\mathbf{Y_j}\,\mathbf{Y_k}]) \mid \mathbf{Y_S} = \tilde{y}_S \sim N\left(\begin{pmatrix}\mu_{j|S}\\ \mu_{k|S}\end{pmatrix}, \Sigma_{jk|S}\right) = N\left(\begin{pmatrix}\mu_{j|S}\\ \mu_{k|S}\end{pmatrix}, \Sigma_{jk} - \Sigma_{jk,S}\Sigma_S^{-1}\Sigma_{jk,S}^t\right), \tag{26}$$

where $\mu_{j|S}$ and $\mu_{k|S}$ are as in equation (21), and Σ_{jk} is the $2n \times 2n$ block matrix with diagonal blocks Σ_j and Σ_k (defined as in (25)), 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 $vec(\mathbf{Y_S})$.

A.7. Covariance structure of the conditional distributions

When $K = ZZ^t$ is block-diagonal, with m blocks of ones of dimension $r \times r$ on the diagonal, then for any positive 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 (Petersen *et al.* 2008; Golub and Van Loan 2012)

$$(A + CBC^t)^{-1} = A^{-1} - A^{-1}C(B^{-1} + C^tA^{-1}C)^{-1}C^tA^{-1}, \tag{27}$$

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 (22) is of the form $\sigma_G^2(j|S)K + \sigma_E^2(j|S)I_n$. Similarly, it follows that $\Sigma_{jk|S}$ (equation () in the main text) 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 $V_E(jk|S)$.

A.8. Proofs of Theorems 1 and 2

Pearl (2009) (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 graph, obtained by extending $\mathcal{G}$ with 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 the lemma 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 $\mathcal{G}$. This concludes the proof of Theorem 1.

Given the result of Theorem 1 and the assumed faithulness, Theorem 2 now follows directly from the consistency for the general PC-algorithm (Spirtes *et al.* 2001); see the first part of the proof of their Theorem 5.1 (p. 407).

Markov properties

The following lemma is taken from Lauritzen (1996) (p. 51), and reformulated with somewhat less general conditions, which however suffice for our purpose.

Lemma. Let P be the joint distribution of random variables $(Y_1, \ldots, Y_p)$, having a density f, and let $\mathcal{H}$ 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 $\mathcal{H}$ 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 Appendix A.1.

A.9. Proof of Theorems 3 and 5

We first prove Theorem 3, by showing the equivalence of the left- and right-hand sides of equations (9) and (11) in the main text. 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 $\mathbf{Y_j} \perp\!\!\!\perp_{P_{\mathbf{Y}|\mathbf{U}}} \mathbf{Y_k} | \{\mathbf{Y_S}\}$) are equivalent, since
$$p_{\mathbf{Y}|\mathbf{U}}(y_j, y_k | y_S) = p(y_j, y_k | y_S, G\Gamma) = p(y_j | y_S, G\Gamma) p(y_k | y_S, G\Gamma) = p_{\mathbf{Y}|\mathbf{U}}(y_j | y_S) p_{\mathbf{Y}|\mathbf{U}}(y_k | y_S).$$

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 A.7. For the second part, we first recall the equivalence of $\mathbf{Y_j} \perp\!\!\!\perp \mathbf{Y_k} | \{\mathbf{Y_S}, \mathbf{G}\}$ and $\mathbf{Y_j} \perp\!\!\!\perp_{P_{\mathbf{Y}|\mathbf{U}}} \mathbf{Y_k} | \{\mathbf{Y_S}\}$. Because of the Gaussianity and the assumed faithfulness, the latter conditional independence is equivalent with
$$\gamma_j^t \Sigma_E \gamma_k - (\gamma_j^t \Sigma_E \Gamma_S)(\Gamma_S^t \Sigma_E \Gamma_S)^{-1} (\Gamma_S^t \Sigma_E \gamma_k) = 0, \tag{28}$$
where we used (19).

Next we consider the conditional distribution of $vec([\mathbf{Y_j}\ \mathbf{Y_k}]) | \mathbf{Y_S} = \tilde{y}_S$ given in () (main text), 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 (28) holds. We recall from (24) 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 (27)) 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-definite $p \times p$ matrices V_G and V_E, we have
$$(V_G \otimes K + V_E \otimes I_n)^{-1} = (V_E^{-1} \otimes I_n) - \left(V_E^{-1} (V_G^{-1} r V_E^{-1})^{-1} V_E^{-1} \right) \otimes K. \tag{29}$$
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
$$\Sigma_S^{-1} = (\Gamma_S^t \Sigma_E \Gamma_S)^{-1} \otimes I_n - A \otimes K. \tag{30}$$
Combining this with the expressions for Σ_{jk} and $\Sigma_{jk,S}$ given in Appendix A.6, we find that $\Sigma_{jk} - \Sigma_{jk,S} \Sigma_S^{-1} \Sigma_{jk,S}^t$ has off-diagonal blocks
$$(\gamma_j^t \Sigma_G \gamma_k) \otimes K + (\gamma_j^t \Sigma_E \gamma_k) \otimes I_n - ((\gamma_j^t \Sigma_E \Gamma_S) \otimes I_n + B_j \otimes K)((\Gamma_S^t \Sigma_E \Gamma_S)^{-1} \otimes I_n - A \otimes K)(\Gamma_S^t \Sigma_E \gamma_k \otimes I_n + B_k^t \otimes K),$$
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 $K^2 = (ZZ^t)^2 rK$), we find that all terms involving a kronecker product with I_n correspond exactly to the left-hand side of (28). 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 $\mathbf{Y_j} \perp\!\!\!\perp \mathbf{Y_k} | \{\mathbf{Y_S}, \mathbf{G}\}$.

A.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
$$Cov(\mathbf{Y_j}^t, \mathbf{G}) = Cov(\gamma_j^t \mathbf{G}^t, \mathbf{G}) = \text{trace}(K)\, \gamma_j^t \Sigma_G = (0, \ldots, 0), \tag{31}$$
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 (31) implies that $\gamma_{j,l} = 0$ for all $l \in D$. Finally, we use that the assumed faithfulness implies the path coefficients condition (see Appendices A.2-A.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.

A.11. Proof of Theorem 6

Assuming $K = ZZ^t$, the first part of theorem follows from the results in Appendix A.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 (18)). 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 (which is a consequence of assumption 5). 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 $\mathbf{G}$.

Supplementary file 1. PCgen: implementation details

1.1. The PC-stable algorithm

We first state the PC-stable algorithm of Colombo and Maathuis (2014), 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 (as the original PC-algorithm), PC-stable only updates the skeleton list-wise, after doing all tests with Y_S of a given size $|S| = s$. More specifically, lines 7-8 in Algorithm 2 make an inventory of the current adjacency sets, which determines which tests of a given size s are to be conducted. Also the orientation rules R1-R3 in Algorithm 1 are applied listwise.

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[10]. In some cases this may lead to partially directed graphs that are not a CPDAG, but the advantages are that PC-stable is order-independent and easy to parallelize.

Algorithm 1. The PC-stable algorithm (following the description of Colombo and Maathuis (2014), 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 $\mathcal{H}$ (in the oracle version $\mathcal{H}$ 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: In the resulting PDAG, try to orient as many undirected edges as possible by repeated application of the following rules:
9: **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).
10: **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).
11: **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 (following the description of Colombo and Maathuis (2014), 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: **repeat**
10: 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 $order(V)$
11: **repeat**
12: Choose a (new) set $S \subseteq a(Y_j) \setminus \{Y_k\}$ with $|S| = s$, using $order(V)$;
13: **if** Y_j and Y_k are conditionally independent given S **then**
14: Delete the edge $Y_j - Y_k$ from $\mathcal{C}$
15: Save S in $Sep(Y_j, Y_k)$ and $Sep(Y_k, Y_j)$
16: **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
17: **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
18: **until** all pairs of adjacent variables (Y_j, Y_k) in $\mathcal{C}$ satisfy $|a(Y_j) \setminus \{Y_k\}| < s$.

1.2. Modified orientation rules

PCgen follows the usual orientation rules of the PC-stable algorithm (lines 9-11 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

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

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 7, 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 (Kalisch *et al.* 2012), 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.

1.3. 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 () and (13) in the main text. 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. Residual Maximum Likelihood (REML) estimates for the full and reduced model are obtained using an EM-algorithm Dempster *et al.* (1977). 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.

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 exact p-values are required 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 usually 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.

1.4. 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 often 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. Alternatively, we could test whether the total genetic *correlation* is zero, using the delta method as in Korte *et al.* (2012).

Again we define a RLRT 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 RLRT 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 function `gencorTest`.

Supplementary file 2. Overview of algorithms

PCgen can be run with either the RC- or the RG-test. Moreover, in PCgen as well as PCres one can use observations on replicates (if available), a GRM (typically with means), or both a GRM and replicates. Together this leads to a large number of options, which we could not all implement. Tables 2 and 3 below give an overview of the algorithms implemented in our pcgen package. We now give a brief overview, with an explanation of the acronyms used. Note that for the sake of readibility we used 'PCgen', 'PCres(means)' and 'PCres(replicates)' in the main text, to refer to respectively PCgen-screening, PCres-multi-A and PCres-uni-R; see the tables below.

PCgen-RC and PCgen-RG refer to PCgen with respectively the RC- or the RG-test for conditional independencies of the form (B) (the test for (A) is always the same). In case of the RG-test, either univariate or multivariate GBLUP can be used (PCgen-RG-uni versus PCgen-RG-multi). Alternatively, we can run PCgen-RC starting with a skeleton determined by PCgen-RG-uni, as described in File S3; we will refer to this variant as PCgen-RC-screening. All currently implemented variations of PCgen require data on replicates, as we did not implement the likelihood ratio test for $\mathbf{Y_j} \perp\!\!\!\perp \mathbf{G} | \mathbf{Y_S}$ (described in the main text).

PCres can be based on either univariate or multivariate GBLUP (PCres-uni versus PCres-multi). In either case, one can use an (additive) relatedness matrix (-A), replicates (-R), or both replicates and the relatedness matrix (-RA). If only a relatedness matrix is used, genotypic means are required. Alternatively, one could reconstruct $\mathcal{G}_\mathcal{Y}$ using the GBLUP itself (PC-GBLUP) itself, similar to the approach of Töpner *et al.* (2017).

Finally, note that if we have r replicates of m genotypes in a completely randomized design, with an $m \times m$ relatedness matrix A, the distribution of the data is

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

i.e., equation (7) (main text) with $K = ZAZ^t$. Then the distribution of the $m \times p$ matrix $\tilde{\mathbf{Y}}$ of genotypic means is then given by

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

Table 2. 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 3 below. The abbreviations 'PCgen', 'PCres(means)' and 'PCres(replicates)' used in the main text refer to respectively PCgen-screening, PCres-multi-A and PCres-uni-R. All PCres approaches are based on the residuals of GBLUP, while PC-GBLUP uses the GBLUP itself.

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]**	PCgen with the RC-test, starting with the skeleton obtained from PCres-uni-R	$\mathcal{G}$
PCres-uni-R **[PCres(replicates)]**	PCres based on univariate GBLUP, using replicates	$\mathcal{G}_{\mathcal{Y}}$
PCres-uni-RA	PCres based on univariate GBLUP, using replicates and a genetic relatedness matrix (GRM)	$\mathcal{G}_{\mathcal{Y}}$
PCres-uni-A	PCres based on univariate GBLUP, using a GRM	$\mathcal{G}_{\mathcal{Y}}$
PCres-multi-R	PCres based on multivariate GBLUP, using replicates	$\mathcal{G}_{\mathcal{Y}}$
PCres-multi-RA	PCres based on multivariate GBLUP, using replicates and a GRM	$\mathcal{G}_{\mathcal{Y}}$
PCres-multi-A **[PCres(means)]**	PCres based on multivariate GBLUP, using a GRM	$\mathcal{G}_{\mathcal{Y}}$
PC-GBLUP	PC(stable) applied to multivariate GBLUP, similar to Töpner *et al.* (2017)	$\mathcal{G}_{\mathcal{Y}}$

where it is assumed that the fixed effects have been accounted for in the estimation of genotypic means. Consequently, the means are distributed according to the same GSEM, but with r times fewer observations, and residual (co)variances that are r times as small.

Supplementary file 3. Extensions of PCgen

We considered the following extensions of PCgen, most of which are implemented in the **pcgen**-package:

- **Prior screening (PCgen-screening):** The RC-test for statement (B) is computationally efficient, in the sense that only a bivariate MTM needs to be fitted, instead of an MTM 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 Kalisch and Bühlmann (2007)). While residuals from the full MTM (for all traits) are difficult to obtain, residuals from single-trait GBLUP can be computed very easily. An appealing strategy, therefore, is 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 GSEMs considered here could include fixed effect QTLs as well. Each QTL is represented by a single node, which like the node G is always a root node. No edges are allowed between QTLs or between a QTL and G, and every edge between a QTL and a trait is oriented towards the trait. The QTLs may further improve the orientation of the graph, but require different types of conditional independence tests, which 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 significantly different from zero. 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 $G \to Y_j$.
- **Restricting the maximum size of S:** As in the original PC-algorithm, the conditioning sets S considered in PCgen can be restricted to a pre-specified maximum size. If the only set that d-separates two traits Y_j and Y_k is bigger than this size, the reconstructed graph will contain the false edge $Y_j - Y_k$ (unless a statistical error occurred in one of the earlier tests, with smaller S); hence the graph will be somewhat denser than without the restriction. Nevertheless, setting a maximum size may have substantial computational and statistical advantages. This is especially the case when testing $\mathbf{Y_j} \perp\!\!\!\perp \mathbf{G}|\mathbf{Y_S}$, and many traits have a direct genetic effect, making the adjacency set of the node G very large. Without any restriction, the number of tests would then increase

Table 3. R-commands for the different algorithms, as implemented in the package `pcgen`. 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. All algorithms involving GBLUP use the residuals (use.GBLUP = F), except the genomic network similar to Töpner *et al.* (2017) (PC-GBLUP, with use.GBLUP = TRUE). 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). The abbreviations 'PCgen', 'PCres(means)' and 'PCres(replicates)' used in the main text refer to respectively PCgen-screening, PCres-multi-A and PCres-uni-R.

Abbreviation	Commands in `pcgen` package
PCgen-RC	`pcgen(d,..., use.res = FALSE)`
PCgen-RG-uni	`C <- cor(getResiduals(d,..., cov.method = "uni"))`
	`pcgen(d,..., use.res = TRUE, res.cor = C)`
	equivalently: `pcgenFast(..., use.res = TRUE)`
PCgen-RG-multi	`C <- cor(getResiduals(d,..., cov.method = "us"))`
	`pcgen(d,..., use.res = TRUE, res.cor = C)`
PCgen-screening **[PCgen]**	`pcgenFast(d,..., use.res = FALSE)`
PCres-uni-R **[PCres(replicates)]**	`pcRes(d,..., cov.method = "uni", use.GBLUP = FALSE)`
PCres-uni-RA	`pcRes(d,..., cov.method = "uni", use.GBLUP = FALSE, K = A)`
PCres-uni-A	`pcRes(m,..., cov.method = "uni", use.GBLUP = FALSE, K = A)`
PCres-multi-R	`pcRes(d,..., cov.method = "us", use.GBLUP = FALSE)`
PCres-multi-RA	`pcRes(d,..., cov.method = "us", use.GBLUP = FALSE, K = A)`
PCres-multi-A **[PCres(means)]**	`pcRes(m,..., cov.method = "us", use.GBLUP = FALSE, K = A)`
PC-GBLUP	`pcRes(m,..., cov.method = "us", use.GBLUP = TRUE, K = A)`

exponentially with the number of traits. This problem is less severe for PCres; for large data-sets we therefore propose to use the PCgen-screening approach described above, with no or little restriction in the screening step. We will do this below for the larger simulated data sets and the rice data.

Supplementary file 4. Simulations

4.1. Simulation setup

We simulate data from model (4) (main text) using the following steps, closely resembling the simulations of Kalisch and Bühlmann (2007):

- 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, which determines the structural relations among the traits. Λ is simulated using the `randomDAG` function from the R-package `pcalg` (Kalisch *et al.* 2012), where edges (i.e., the nonzero elements of Λ) occur with probability $p_t \in (0,1)$ (for details see the pcalg-documentation and Kalisch and Bühlmann (2007), 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 we simulate the corresponding genetic variances and covariances in Σ_G. The genetic variances are drawn independently from a uniform distribution on $[1, 2]$ and random covariances are introduced through random eigenvectors as in Joe (2006), 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 (4) in the main text. This is done recursively, following the topological ordering of the DAG (Kalisch and Bühlmann (2007), p. 621).

4.2. Performance criteria

As in Kalisch and Bühlmann (2007) 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 Kalisch and Bühlmann (2007):

- 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 CPDAG. See also Tsamardinos *et al.* (2006).

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

Supplementary file 5. Simulation with APSIM

We used the crop-growth model APSIM-wheat 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 Bustos-Korts (2017)). APSIM simulations were carried out in Emerald, during 1993, which corresponds to a severe drought environment. Simulation settings were the same as in Casadebaig *et al.* (2016) and genotype-specific parameters had the ranges specified in Bustos-Korts (2017). The wheat panel was assumed to segregate for seven 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 seven primary traits ($\mathbf{Y_1}, \ldots, \mathbf{Y_7}$) are a subset of those used in Bustos-Korts (2017), and were chosen because they have an important impact on grain yield, as shown by global sensitivity analysis (Casadebaig *et al.* 2016). Apart from the primary traits there are four secondary traits, each of them depending on some of the primary traits, and sometimes some of the other secondary traits. The final trait is yield, which depends on three of the secondary traits. Acronyms for all 12 traits are given in Table S3. 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 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:

$$\mathbf{Y_j}(T) = \int_0^T f_z(pa(\mathbf{Y_j}(t))dt, \tag{32}$$

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 is detailed in Bustos-Korts (2017). The sets of parental traits stay the same over time, and therefore define the summary graph given in Figure S1 (main text).

Supplementary file 6. Skipping conditional independence tests that do not involve G

6.1. Skipping tests for statements of the form $\mathbf{Y_j} \perp\!\!\!\perp \mathbf{Y_k} | \{\mathbf{Y_S}\}$: a motivation

In Theorem 2 (main text) we provided conditions for consistency of the oracle version of PCgen. In our experience, the finite sample performance of PCgen can be improved if we skip some of the tests in the skeleton-stage. Differences between the oracle and sample version of PCgen can occur everywhere in the graph, but seem to occur most often for conditioning sets not containing G. This is illustrated in the example in Figure 8, in which there are genetic effects on traits Y_1 and Y_2, as well as a direct effect of Y_1 on Y_2.

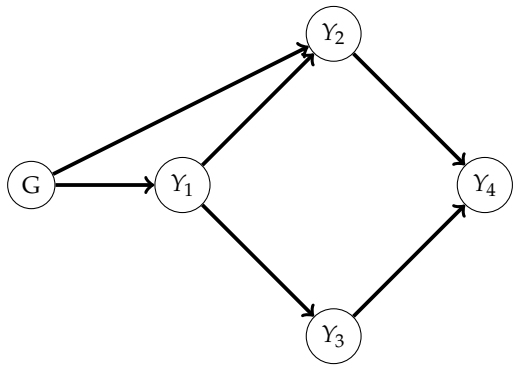

Figure 8. A GSEM with independent genetic effects on Y_1 and Y_2, and a direct effect $Y_1 \to Y_2$.

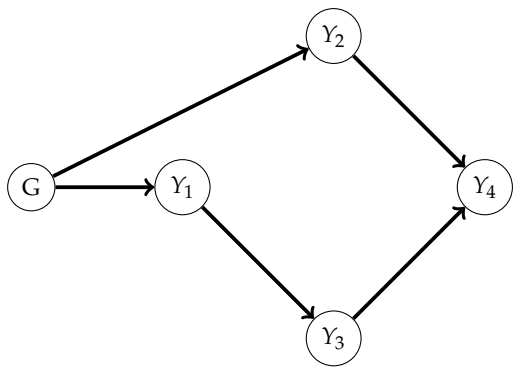

Figure 9. Typical output of PCgen, without skipping tests for $\mathbf{Y_j} \perp\!\!\!\perp \mathbf{Y_k} | \{\mathbf{Y_S}\}$, and based on observations of $(\mathbf{Y_1}, \mathbf{Y_2}, \mathbf{Y_3}, \mathbf{Y_4})$ generated by the model of Figure 8, for 400 genotypes and 2 replicates. The edge between Y_1 and Y_2 is missing because the test for conditional independence of $\mathbf{Y_1}$ and $\mathbf{Y_2}$ given $\mathbf{Y_4}$ has too little power.

Then given a large number of observations and assuming faithfulness, PCgen will recover the true skeleton. However, with a small or moderate sample size, the test for conditional independence of $\mathbf{Y_1}$ and $\mathbf{Y_2}$ given $\mathbf{Y_4}$ has very little power. The test for conditional independence of $\mathbf{Y_1}$ and $\mathbf{Y_2}$ given $\{\mathbf{Y_4}, \mathbf{G}\}$ is a lot more powerful here. However, this test is not performed anymore, after the null-hypothesis of conditional independence of $\mathbf{Y_1}$ and $\mathbf{Y_2}$ given $\mathbf{Y_4}$ has been accepted. Therefore, a typical output of PCgen looks like Figure 9.

In order to make PCgen more powerful we therefore propose to perform only those conditional independence tests where $\mathbf{G}$ is contained in the conditioning set, at least when both variables whose conditional independence is tested have positive genetic variance. This still gives a valid algorithm, in the sense that the oracle version of PCgen gives identical output. Intuitively this is obvious, as G is a root node, and everything can be done conditionally on $\mathbf{G}$. We formally show this below in section 6.3, for which we need the following well known characterization of the skeleton.

6.2. A characterization of the skeleton

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

$$\begin{aligned} &\text{there is an edge between nodes } A \text{ and } B \text{ in the skeleton of a DAG } \mathcal{H} \\ \iff \quad &\forall S \subseteq V \setminus \{A, B\}, A \text{ and } B \text{ are conditionally dependent given } S. \end{aligned} \tag{33}$$

This was shown in Spirtes *et al.* (2001) (Theorem 3.4); here we adopt the formulation of Kalisch and Bühlmann (2007) (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 (33), but with on the right-hand side only $S = V \setminus \{A, B\}$. Hence, if data are generated by a DAG $\mathcal{H}$ and we assume faithfulness, the skeleton of $\mathcal{H}$ 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 $A \to B \leftarrow C$).

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

In view of (33), our modification is correct in the sense that the oracle version of PCgen still recovers the true skeleton. This correctness follows from the facts that

1. like the PC-skeleton algorithm, PCgen starts with the complete undirected graph, and then tests conditional independencies, removing edges when a conditional independence relation is found. When in fact there *is* an edge $Y_j - Y_k$ in the true graph $\mathcal{G}$, then of course skipping some of these tests still produces the correct result (and in the sample version of PCgen even with higher probability).

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 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.

Supplementary file 7. Miscelleneous

7.1. 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 G (instead of multiple nodes $G_1, \ldots, G_p$, sometimes used in the literature) is that the relatedness matrix K is the same for all traits. If for example $G_1 \to Y_1 \to Y_3 \leftarrow Y_2 \leftarrow G_2$ (as in Figure 3 in the main text), it follows from (7) (also in the main text) that the marginal distribution of $\mathbf{Y_3}$ 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 $\mathbf{G_2}$. This differs from the scenario $QTL_A \to Y_1 \to Y_3 \leftarrow Y_2 \leftarrow QTL_B$, where the total (fixed) effects of QTLs A and B on Y_3 can be estimated from the marginal distribution of Y_3. If we condition on $\mathbf{Y_1}$ and $\mathbf{Y_2}$, $\mathbf{Y_3}$ becomes independent of $\mathbf{G_1}$ and $\mathbf{G_2}$. In fact it is also independent of $\mathbf{G_3}$, since the latter is zero. Consequently, given $\mathbf{Y_1}$ and $\mathbf{Y_2}$, $\mathbf{Y_3}$ is independent of $\mathbf{G} = [\mathbf{G_1}\ \mathbf{G_2}\ \mathbf{G_3}]$, 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.

7.2. Genomic networks

Töpner *et al.* (2017) recently proposed to estimate a directed network based on the predicted genetic effects themselves, rather than the residuals. As mentioned in the Discussion (section 'Data from different experiments'), the implicit assumption is that the GSEM model (equation (4) in the main text) is replaced by a model defined by $\mathbf{Y} = \mathbf{X}B + \mathbf{U} + \mathbf{E}$ and $\mathbf{U} = \mathbf{U}\Lambda + \tilde{\mathbf{U}}$. This is however problematic if in fact the multi-trait data follow equation (4). For example, consider again the graph from Figure 3 in the main text ($Y_1 \to Y_3 \leftarrow Y_2$), with direct genetic effects on Y_1 and Y_2. 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 (6) in the main text). Because Y_1 and Y_2 are root nodes in $\mathcal{G}_y$, $\mathbf{U_1} = \mathbf{G_1}$ and $\mathbf{U_2} = \mathbf{G_2}$. We can make the following observations:

- To get the correct skeleton, the PC-algorithm must find that $\mathbf{Y_1}$ and $\mathbf{Y_2}$ 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.
- We may infer some of the edges $G \to Y_j$, but not all of them. Moreover, this can be done only indirectly, through comparison with the trait-to-trait network estimated by the residuals-based approach, and requires additional testing if the $\mathbf{U_j}$ are zero (similar to the tests for $\mathbf{Y_j} \perp\!\!\!\perp \mathbf{G} | \emptyset$ in PCgen). In particular, if $\mathbf{U_j} = 0$ for some j, this clearly excludes $G \to Y_j$. But in our example we find that $\mathbf{U_1}, \mathbf{U_2}, \mathbf{U_3}$ are all nonzero, and 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 $G \to Y_2$. It is impossible to make any inference about $G \to Y_3$, which PCgen can in principle do.

7.3. Intervening versus conditioning

To see the difference between intervening and conditioning, see the SEM in Figure 10, where there are no fixed effects, all path-coefficients are 1, and all error variables are standard normal and independent. To keep the algebra simple, we did not include genetic effects, but a similar difference between intervening and conditioning occurs when for example there is a genetic effect $\mathbf{G_1} \sim N(0, K)$ on the first trait. Prior to the intervention $\mathbf{Y_2} := (1, \ldots, 1)^t$, the joint distribution of $(\mathbf{Y_1}[i], \mathbf{Y_2}[i], \mathbf{Y_3}[i])^t$ (for each individual i) is multivariate normal with mean zero and covariance Σ given by

$$\Lambda = \begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}, \quad \Gamma = (I - \Lambda)^{-1} = \begin{pmatrix} 1 & 1 & 2 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}, \quad \Sigma = \Gamma^t \Sigma_E \Gamma = \Gamma^t \Gamma = \begin{pmatrix} 1 & 1 & 2 \\ 1 & 2 & 3 \\ 2 & 3 & 6 \end{pmatrix}.$$

From well-known results for the multivariate normal distribution, it follows that the conditional distribution of $\mathbf{Y_3}[i]$ given $\mathbf{Y_2}[i] = 1$ is normal with mean and variance both equal to 2. The conditional mean being larger than 1 may seem counterintuitive, but can be understood by noting that the information $\mathbf{Y_2}[i] = 1$ suggests a larger than exptected value for $\mathbf{Y_1}[i]$, which in turn gives an additional (expected) effect on $\mathbf{Y_3}[i]$. In case we intervene and $\mathbf{Y_2}[i]$ is forced to have a value 1, there is no effect $Y_1 \to Y_2$, and the distribution of $\mathbf{Y_3}[i]$ is normal with variance 2 and mean 1.

Figure 10. An example of a linear SEM, before and after the intervention $Y_2 := 1$.

Table S1. Overview of notation. Appendix A.1 provides additional graph-theoretic terminology.

Symbol	Meaning / Category
The observed traits	
$\mathbf{Y}$	$n \times p$ matrix of observations, for n individuals and p traits
$\mathbf{Y_j}$	$n \times 1$ vector of observations on the j trait (jth column of $\mathbf{Y}$)
$\mathbf{Y_S}$	$n \times \lvert S \rvert$ matrix of observations, obtained by restricting $\mathbf{Y}$ to the columns in a subset $S \subset \{1, \ldots, p\}$
Path coefficients in linear (G)SEM	
Λ	$p \times p$ matrix of path coefficients
Γ	inverse of $(I - \Lambda)$. See also equation (8), and equation (17) in Appendix A.2
γ_j	jth column of Γ
Γ_S	Γ with columns restricted to a subset $S \subset \{1, \ldots, p\}$
Other (exogenous) fixed effects in linear (G)SEM	
B	$q \times p$ matrix of fixed effects of exogenous covariates, including an intercept
$\mathbf{X}$	$n \times q$ design matrix for these covariates
Genetic effects	
$\mathbf{G}$	$n \times p$ matrix of direct genetic effects
$\mathbf{g_i}$	$1 \times p$ vector of direct genetic effects for individual i (ith row of $\mathbf{G}$)
$\mathbf{G_j}$	$n \times 1$ vector of direct genetic effects for the jth trait (jth column of $\mathbf{G}$)
$\mathbf{U}$	$n \times p$ matrix of total genetic effects; equals $\mathbf{G}\Gamma$
$\mathbf{u_i}$	$1 \times p$ vector of total genetic effects for individual i (ith row of $\mathbf{U}$)
$\mathbf{U_j}$	$n \times 1$ vector of total genetic effects for the jth trait (jth column of $\mathbf{U}$)
Residual effects	
$\mathbf{E}$	$n \times p$ matrix of residual effects
$\mathbf{e_i}$	$1 \times p$ vector of residual effects for individual i (ith row of $\mathbf{E}$)
$\mathbf{E_j}$	$n \times 1$ vector of residual effects for for the jth trait (jth column of $\mathbf{E}$)
Covariance matrices and (conditional) means	
Σ_G	$p \times p$ covariance matrix for the direct genetic effects
D	subset of $\{1, \ldots, p\}$, indicating nonzero diagonal elements in Σ_G
$\sigma^2_{G,j}$	The jth diagonal element of Σ_G
Σ_E	$p \times p$ covariance matrix for the residual errors
V_G	$p \times p$ covariance matrix of the total genetic effects; equals $\Gamma^t \Sigma_G \Gamma$
V_E	$p \times p$ covariance matrix of the transformed residuals ($\mathbf{E}\Gamma$); equals $\Gamma^t \Sigma_E \Gamma$
$\mu_{j\|S}, \Sigma_{j\|S}$	mean and covariance of $\mathbf{Y_j}$ given $\mathbf{Y_S}$ -
Graphs	
Y_j	the node representing the random vector $\mathbf{Y_j}$ (i.e., the jth trait)
Y_S	the set of nodes $\{Y_m : m \in S\}$
$\mathcal{G}_\mathcal{Y}$	graph over the traits $Y_1, \ldots, Y_p$, defined by Λ
G	The node representing the matrix of direct genetic effects $\mathbf{G}$
$\mathcal{G}$	$\mathcal{G}_\mathcal{Y}$, extended with the node G and edges $G \to Y_j$ defined by D

	$\mathcal{G}_{\mathcal{Y}}$				$G \to Y_j$		
	TPR	FPR	TDR	SHD	TPR	FPR	TDR
Scenario 1 ($p = 4$)	($p_t = 1/3$)				($p_g = 0.5$)		
PCgen-RC	0.652	0.009	0.964	0.454	0.982	0.026	0.995
PCgen-RG-uni	0.650	0.039	0.908	0.584	0.981	0.027	0.995
PCgen-RC-screening	0.647	0.006	0.981	0.442	0.982	0.026	0.995
PCres-multi-R	0.657	0.116	0.719	1.908			
PCres-uni-R	0.650	0.039	0.908	1.410			
PCres-multi-RA	0.658	0.088	0.779	1.728			
PCres-uni-RA	0.653	0.026	0.932	1.374			
PCres-multi-A	0.521	0.390	0.438	3.174			
PCres-uni-A	0.640	0.083	0.758	1.760			
PC-GBLUP	0.635	0.661	0.384	4.284			
Scenario 2 ($p = 4$)	($p_t = 0.5$)				($p_g = 0.5$)		
PCgen-RC	0.818	0.035	0.980	1.238	0.976	0.031	0.995
PCgen-RG-uni	0.819	0.073	0.939	1.358	0.974	0.032	0.996
PCgen-RC-screening	0.804	0.033	0.986	1.246	0.976	0.031	0.995
PCres-multi-R	0.834	0.132	0.862	2.560			
PCres-uni-R	0.819	0.073	0.939	2.320			
PCres-multi-RA	0.831	0.098	0.884	2.430			
PCres-uni-RA	0.821	0.058	0.947	2.266			
PCres-multi-A	0.672	0.364	0.659	3.628			
PCres-uni-A	0.782	0.124	0.863	2.688			
PC-GBLUP	0.793	0.709	0.563	4.634			

Table S2. Performance of PCgen and residuals-based approaches, averaged over 500 simulated data-sets, for scenarios 1 and 2. For ease of presentation we used in the main text the abbreviations 'PCgen', 'PCres(means)' and 'PCres(replicates)', to refer to respectively PCgen-screening, PCres-multi-A and PCres-uni-R. See also Tables 2 and 3 in File S2.

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 S3. Acronyms for 12 traits simulated using APSIM.

trait name / category	acronym
Flowering	
anthesis	A
silking	Sk
Height	
ear height	EH
plant height	PH
tassel height	TH
Yield related	
grain weight	GW
yield	Y

Table S4. Maize data: trait acronyms

trial acronym	location	drought	temperature	year
Kar12W	Karlsruhe (Germany)	WW	Cool	2012
Ner13W	Nérac (France)	WW	Hot	2013
Ner12R	Nérac (France)	WD	Hot (days)	2012
Bol12R	Bologna (Italy)	WD	Hot	2012

Table S5. Maize data: overview of the trials. Details on the environmental classification are given in the main text, and in Millet *et al.* (2016).

Kar12W	G	PH	TH	EH	A	Sk	Y	GW
G	1.00E+00	1.75E-08	1.62E-04	4.06E-10	4.83E-09	3.98E-18	3.76E-14	3.44E-26
PH	1.75E-08	1.00E+00	0.00E+00	2.17E-04	6.27E-02	3.06E-01	1.81E-01	3.66E-01
TH	1.62E-04	0.00E+00	1.00E+00	3.60E-01	3.10E-02	1.08E-02	1.65E-01	1.00E-01
EH	4.06E-10	2.17E-04	3.60E-01	1.00E+00	1.25E-01	2.07E-01	2.58E-01	1.09E-01
A	4.83E-09	6.27E-02	3.10E-02	1.25E-01	1.00E+00	0.00E+00	3.01E-01	9.68E-01
Sk	3.98E-18	3.06E-01	1.08E-02	2.07E-01	0.00E+00	1.00E+00	5.26E-01	1.08E-02
Y	3.76E-14	1.81E-01	1.65E-01	2.58E-01	3.01E-01	5.26E-01	1.00E+00	2.22E-01
GW	3.44E-26	3.66E-01	1.00E-01	1.09E-01	9.68E-01	1.08E-02	2.22E-01	1.00E+00
Ner13W	G	PH	TH	EH	A	Sk	Y	GW
G	1.00E+00	1.79E-07	4.49E-10	8.16E-13	5.80E-15	1.21E-34	3.81E-34	2.83E-41
PH	1.79E-07	1.00E+00	0.00E+00	3.71E-03	5.53E-01	4.79E-02	7.80E-01	4.06E-01
TH	4.49E-10	0.00E+00	1.00E+00	1.41E-02	5.74E-01	4.21E-02	4.05E-03	8.43E-01
EH	8.16E-13	3.71E-03	1.41E-02	1.00E+00	3.70E-01	1.17E-02	2.65E-02	5.91E-01
A	5.80E-15	5.53E-01	5.74E-01	3.70E-01	1.00E+00	1.60E-12	3.20E-01	1.94E-02
Sk	1.21E-34	4.79E-02	4.21E-02	1.17E-02	1.60E-12	1.00E+00	3.59E-01	1.87E-01
Y	3.81E-34	7.80E-01	4.05E-03	2.65E-02	3.20E-01	3.59E-01	1.00E+00	3.62E-02
GW	2.83E-41	4.06E-01	8.43E-01	5.91E-01	1.94E-02	1.87E-01	3.62E-02	1.00E+00
Ner12R	G	PH		EH	A	Sk	Y	GW
G	1.00E+00	8.68E-14		1.47E-25	1.39E-68	1.16E-50	5.26E-62	7.03E-59
PH	8.68E-14	1.00E+00		4.65E-13	1.70E-02	7.46E-01	8.86E-03	5.00E-01
EH	1.47E-25	4.65E-13		1.00E+00	4.53E-01	3.52E-01	1.60E-01	1.66E-02
A	1.39E-68	1.70E-02		4.53E-01	1.00E+00	6.56E-03	2.25E-01	3.43E-01
Sk	1.16E-50	7.46E-01		3.52E-01	6.56E-03	1.00E+00	3.55E-09	2.59E-02
Y	5.26E-62	8.86E-03		1.60E-01	2.25E-01	3.55E-09	1.00E+00	1.79E-02
GW	7.03E-59	5.00E-01		1.66E-02	3.43E-01	2.59E-02	1.79E-02	1.00E+00
Bol12R	G	PH	TH	EH	A	Sk	Y	GW
G	1.00E+00	8.58E-11	2.92E-11	1.80E-51	4.40E-25	5.92E-37	4.55E-32	3.17E-61
PH	8.58E-11	1.00E+00	0.00E+00	5.09E-07	7.89E-02	3.25E-03	1.32E-01	9.86E-01
TH	2.92E-11	0.00E+00	1.00E+00	1.52E-05	1.23E-01	3.36E-01	1.53E-02	9.71E-01
EH	1.80E-51	5.09E-07	1.52E-05	1.00E+00	1.78E-01	2.29E-01	7.36E-02	2.71E-01
A	4.40E-25	7.89E-02	1.23E-01	1.78E-01	1.00E+00	1.47E-10	2.54E-03	9.30E-02
Sk	5.92E-37	3.25E-03	3.36E-01	2.29E-01	1.47E-10	1.00E+00	3.99E-09	2.99E-02
Y	4.55E-32	1.32E-01	1.53E-02	7.36E-02	2.54E-03	3.99E-09	1.00E+00	2.65E-04
GW	3.17E-61	9.86E-01	9.71E-01	2.71E-01	9.30E-02	2.99E-02	2.65E-04	1.00E+00

Table S6. PCgen-reconstruction for the 4 maize trials, with $\alpha = 0.01$: p-values. For each pair of nodes, the table contains the highest p-value observed across all conditional independence tests performed for that pair.

Kar12W	PH	TH	EH	A	Sk	Y	GW
PH	NA (NA)	0.91 (0)	0.59 (1.1e-11)	0.53 (2.6e-12)	0.47 (2.1e-10)	0.32 (0.00035)	-0.099 (0.23)
TH	0.91 (0)	NA (NA)	0.61 (4.9e-11)	0.49 (4.7e-10)	0.48 (7.5e-10)	0.39 (4.2e-05)	-0.15 (0.074)
EH	0.59 (1.1e-11)	0.61 (4.9e-11)	NA (NA)	0.56 (3.2e-12)	0.51 (1.2e-10)	0.16 (0.11)	-0.19 (0.029)
A	0.53 (2.6e-12)	0.49 (4.7e-10)	0.56 (3.2e-12)	NA (NA)	0.96 (0)	0.31 (0.00019)	-0.4 (6.2e-08)
Sk	0.47 (2.1e-10)	0.48 (7.5e-10)	0.51 (1.2e-10)	0.96 (0)	NA (NA)	0.28 (0.00076)	-0.44 (1.3e-09)
Y	0.32 (0.00035)	0.39 (4.2e-05)	0.16 (0.11)	0.31 (0.00019)	0.28 (0.00076)	NA (NA)	0.13 (0.15)
GW	-0.099 (0.23)	-0.15 (0.074)	-0.19 (0.029)	-0.4 (6.2e-08)	-0.44 (1.3e-09)	0.13 (0.15)	NA (NA)
Ner13W	PH	TH	EH	A	Sk	Y	GW
PH	NA (NA)	0.92 (0)	0.7 (1.1e-16)	0.59 (1.3e-15)	0.49 (5.9e-11)	0.48 (1.3e-09)	0.0095 (0.9)
TH	0.92 (0)	NA (NA)	0.67 (0)	0.66 (0)	0.57 (5.6e-16)	0.51 (8.4e-12)	0.02 (0.8)
EH	0.7 (1.1e-16)	0.67 (0)	NA (NA)	0.66 (0)	0.58 (1.4e-15)	0.31 (0.00012)	-0.042 (0.61)
A	0.59 (1.3e-15)	0.66 (0)	0.66 (0)	NA (NA)	0.94 (0)	0.52 (1.9e-14)	-0.11 (0.11)
Sk	0.49 (5.9e-11)	0.57 (5.6e-16)	0.58 (1.4e-15)	0.94 (0)	NA (NA)	0.37 (1.1e-07)	-0.24 (0.00062)
Y	0.48 (1.3e-09)	0.51 (8.4e-12)	0.31 (0.00012)	0.52 (1.9e-14)	0.37 (1.1e-07)	NA (NA)	0.15 (0.043)
GW	0.0095 (0.9)	0.02 (0.8)	-0.042 (0.61)	-0.11 (0.11)	-0.24 (0.00062)	0.15 (0.043)	NA (NA)
Ner12R	PH		EH	A	Sk	Y	GW
PH	NA (NA)		0.29 (0.0034)	0.039 (0.65)	-0.09 (0.31)	0.26 (0.0037)	0.16 (0.083)
EH	0.29 (0.0034)		NA (NA)	0.57 (2.7e-15)	0.21 (0.0058)	0.21 (0.0055)	-0.21 (0.0088)
A	0.039 (0.65)		0.57 (2.7e-15)	NA (NA)	0.68 (0)	0.12 (0.081)	-0.16 (0.026)
Sk	-0.09 (0.31)		0.21 (0.0058)	0.68 (0)	NA (NA)	-0.44 (2.2e-11)	-0.23 (0.00097)
Y	0.26 (0.0037)		0.21 (0.0055)	0.12 (0.081)	-0.44 (2.2e-11)	NA (NA)	0.38 (7.9e-08)
GW	0.16 (0.083)		-0.21 (0.0088)	-0.16 (0.026)	-0.23 (0.00097)	0.38 (7.9e-08)	NA (NA)
Bol12R	PH	TH	EH	A	Sk	Y	GW
PH	NA (NA)	0.89 (0)	0.66 (0)	0.42 (3.4e-08)	0.38 (5.9e-07)	-0.046 (0.58)	0.02 (0.8)
TH	0.89 (0)	NA (NA)	0.53 (1.4e-12)	0.37 (1.7e-06)	0.34 (9.2e-06)	-0.051 (0.54)	0.12 (0.13)
EH	0.66 (0)	0.53 (1.4e-12)	NA (NA)	0.44 (1.9e-10)	0.36 (4.8e-07)	-0.025 (0.74)	-0.12 (0.11)
A	0.42 (3.4e-08)	0.37 (1.7e-06)	0.44 (1.9e-10)	NA (NA)	0.9 (0)	-0.43 (3.3e-09)	-0.11 (0.15)
Sk	0.38 (5.9e-07)	0.34 (9.2e-06)	0.36 (4.8e-07)	0.9 (0)	NA (NA)	-0.61 (0)	-0.1 (0.16)
Y	-0.046 (0.58)	-0.051 (0.54)	-0.025 (0.74)	-0.43 (3.3e-09)	-0.61 (0)	NA (NA)	0.0068 (0.93)
GW	0.02 (0.8)	0.12 (0.13)	-0.12 (0.11)	-0.11 (0.15)	-0.1 (0.16)	0.0068 (0.93)	NA (NA)

Table S7. Estimates of total genetic correlation in the four maize trials, with significance between parentheses.

Traits	Trait	acronym	Unit
(A) Shoot morphological traits			
	Plant height	PHT	cm
	Tiller number	TN	$plant^{-1}$
	Total leaf area	TLA	$m^2\ plant^{-1}$
(B) Physiological traits			
	Cumulative water transpiration	CWT	$kg\ plant^{-1}$
(C) Root anatomical traits			
	Maximum root length	MRL	cm
	Surface area	SA	$cm^2\ plant^{-1}$
	Root volume	RV	$cm^3\ plant^{-1}$
	Average root thickness	ART	mm
	Total root weight density	TRWD	$g\ cm^{-3}$
	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	$m\ plant^{-1}$
(D) Root anatomical traits			
	Root diameter	RD	μm
	Cortex diameter	CD	μm
	Stele diameter	SD	μm
	Late metaxylem diameter	LMXD	μm
	Late metaxylem number	LMXN	μm
(E) Dry matter traits			
	Leaf weight	LW	$g\ plant^{-1}$
	Stem weight	SW	$g\ plant^{-1}$
	Root weight	RW	$g\ plant^{-1}$

Table S8. Rice data: trait acronyms, following the description of Kadam *et al.* (2017), with derived traits removed.

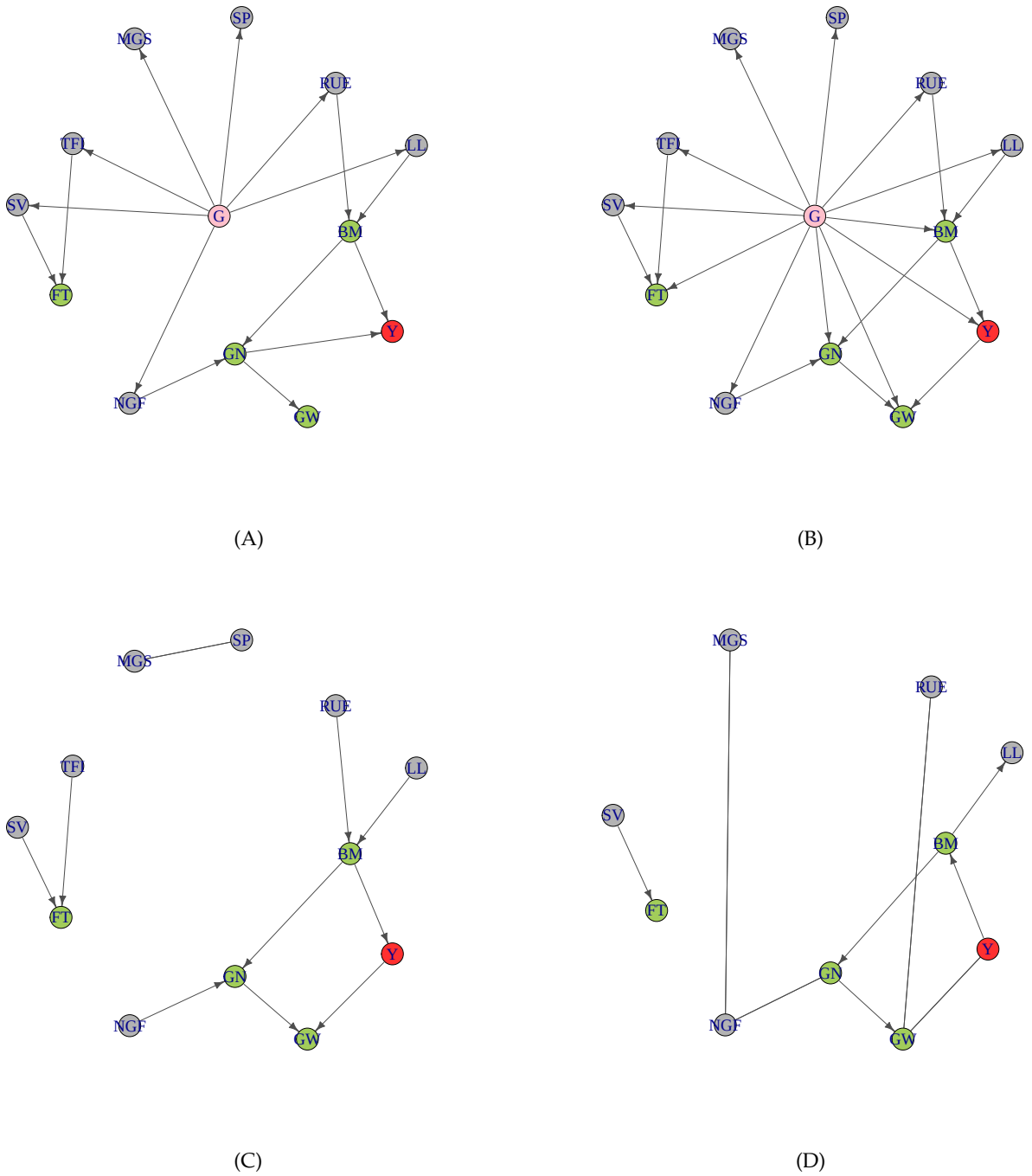

Figure S1. Networks of 12 APSIM traits in the dry environment Emerald 1993 (severe drought), including seven primary traits (grey), four secondary traits (green) and yield (red). (A) Summary graph, (B) PCgen, (C) pcRes and (D) PC-stable, using the 300 simulated QTLs. Trait acronyms are given in Table S3.

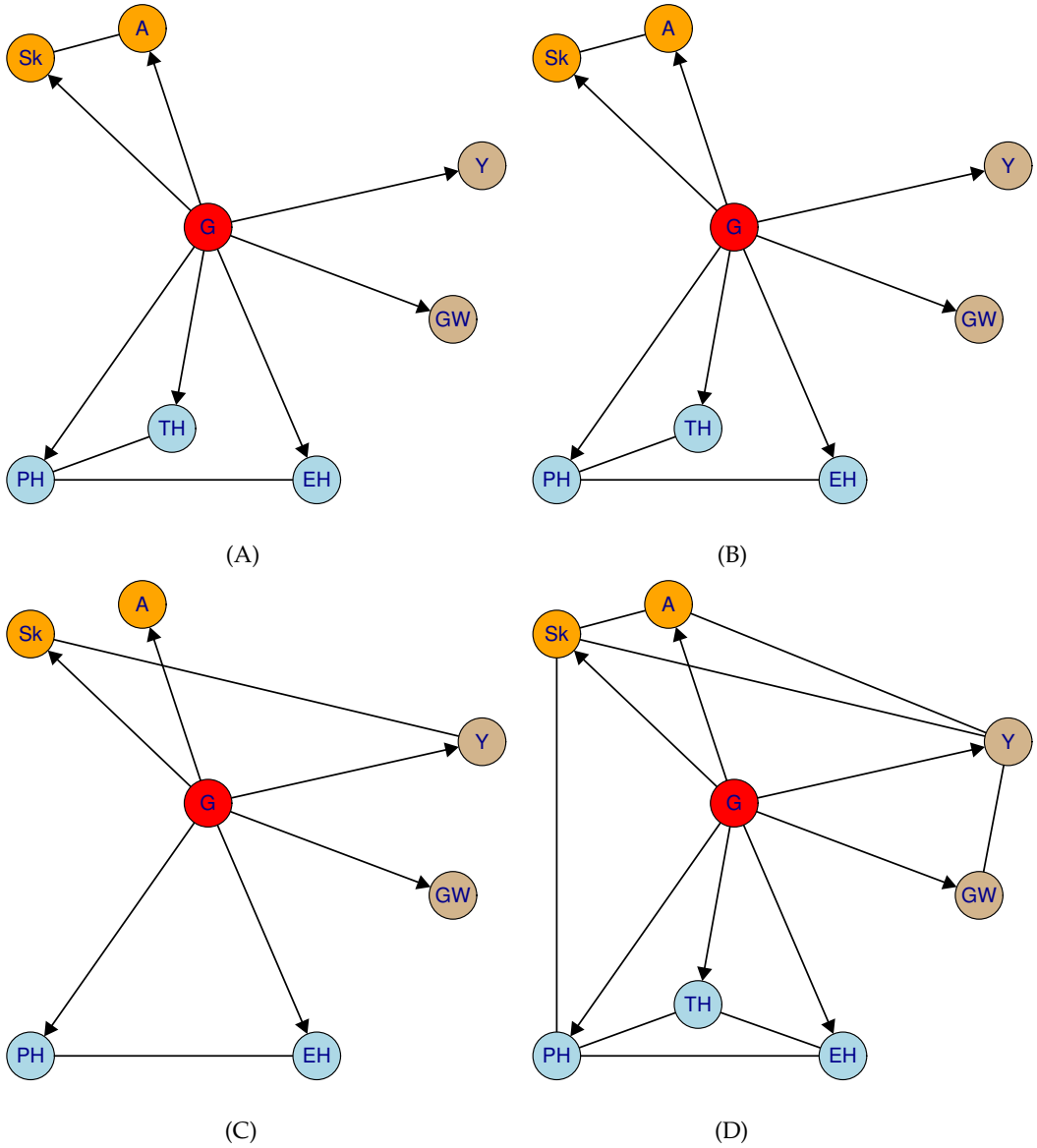

Figure S2. Estimated networks with $\alpha = 0.001$, for four of the DROPS field trials: (A) Kar12W, (WW, Cool), (B) Ner13W (WW, Hot), (C) Ner12R (WD, Hot (days)) and (D) Bol12R (WD, Hot). Trait categories are flowering (orange), height (blue) and yield (brown). Each trial represents a different environmental scenario, being well-watered (WW) or water-deficit (WD), and different temperatures (see text). Trait acronyms are explained in Table S4.

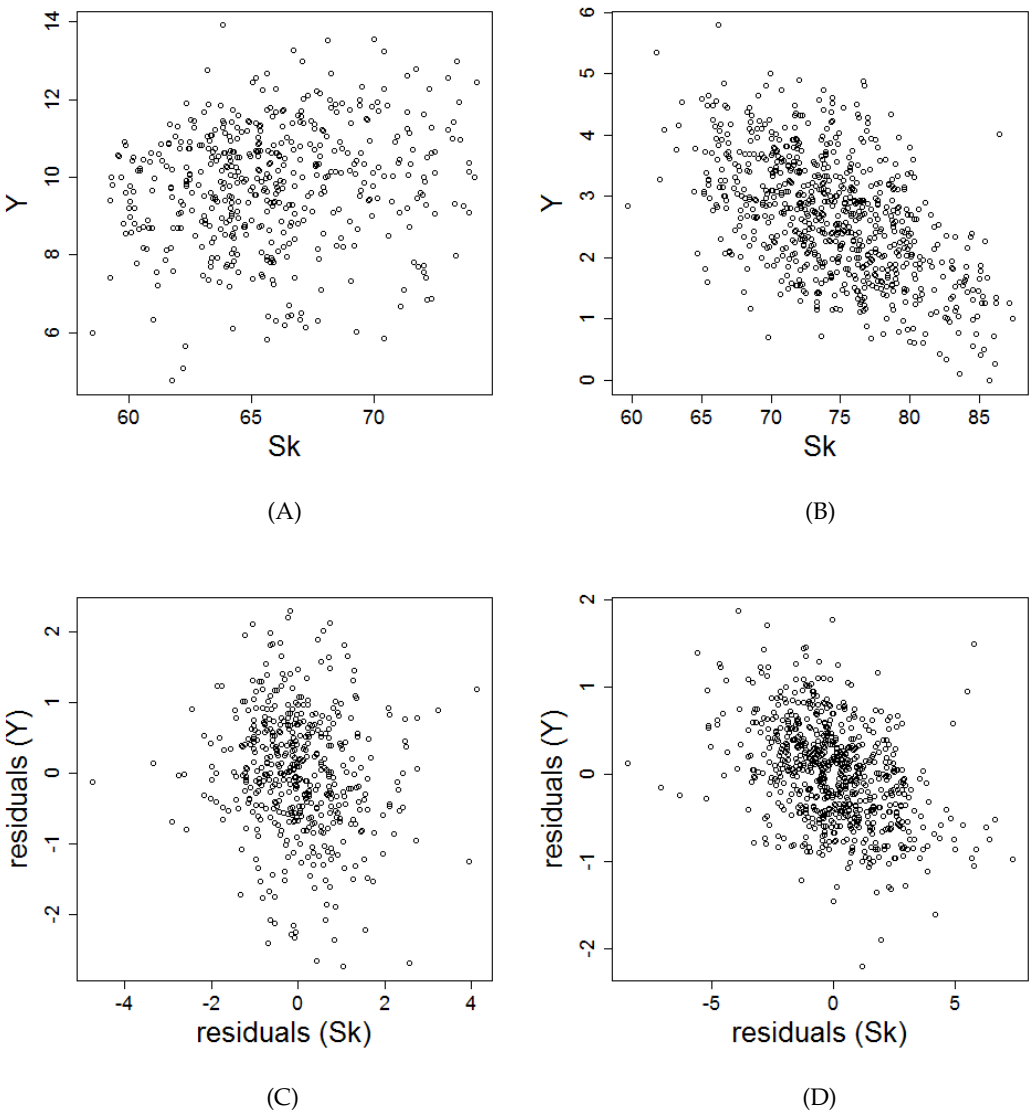

Figure S3. Yield (*Y*) versus silking (*Sk*), in a trial without heat and drought stress (Kar12W, (A)), and in a trial with severe stress (Bol12R, (B)). Panels (C) and (D) show the corresponding residuals from a bivariate multi-trait model.

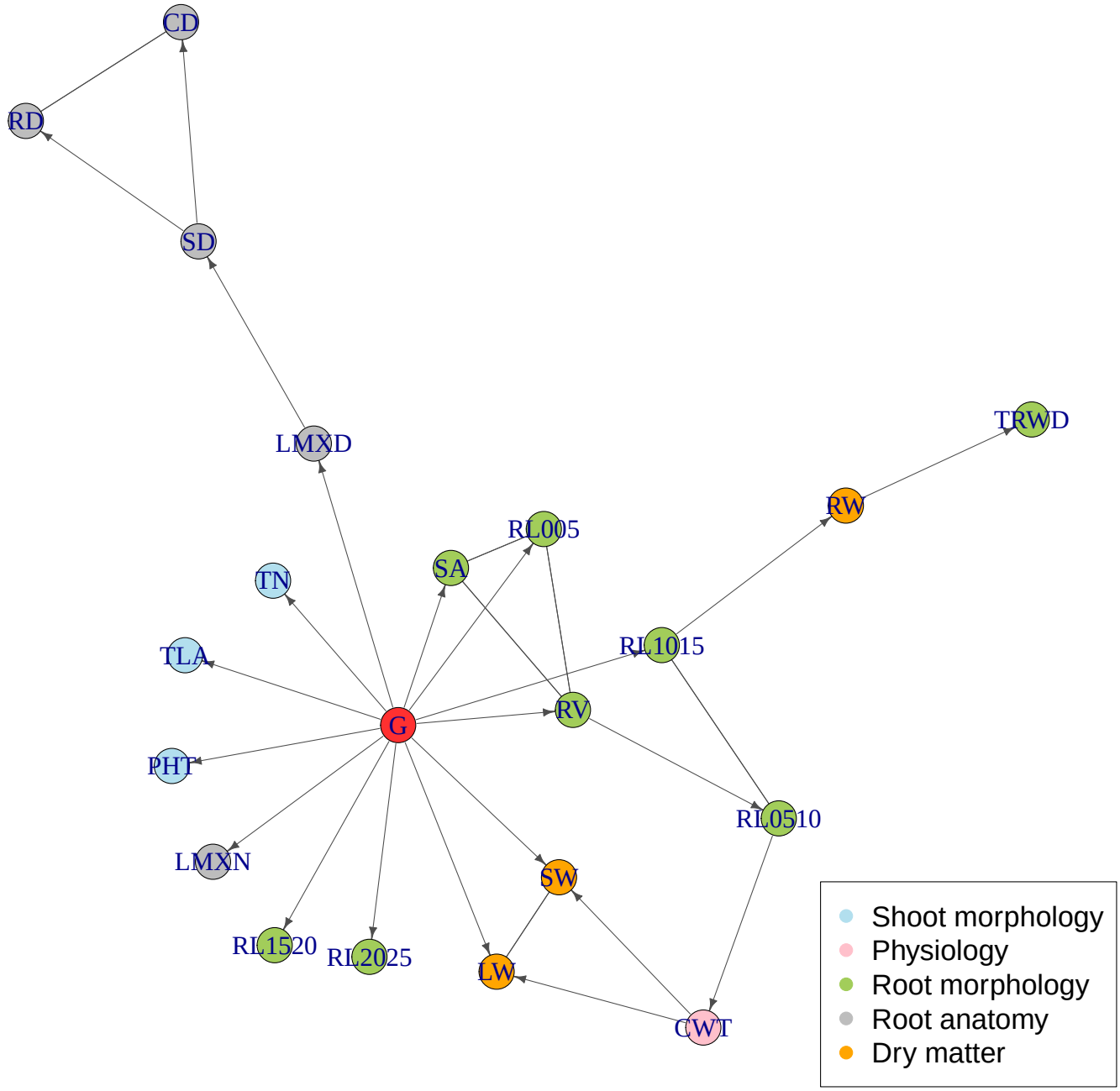

Figure S4. PCgen-reconstruction for the rice-data from Kadam *et al.* (2017), with $\alpha = 0.001$. Five traits (MRL, ART, RL2530, RL3035 and RL35) are not shown, as they were completely isolated in the graph, without any connections to other traits or G.

	G	PHT	TN	TLA	CWT	RL005	RL0510	RL1015	RL1520	RL2025	RL2530	RL3035	RL35	MRL	SA	RV	ART	TRWD	RD	CD	SD	LMXD	LMXN	LW	SW	RW
RW	0.02	1	1	1	0.23	0.67	0.23	0	1	1	1	1	1	0.03	0.11	0.01	0.28	0	0.01	0.38	0.23	0.04	0.01	0.01	0.79	1
SW	−Inf	1	1	1	0	0.7	0.95	0.4	1	1	1	1	1	0.3	0.55	0.07	0.38	0.12	0.08	0.02	0.05	0.13	0.08	0	1	0.79
LW	−Inf	1	1	1	0	0.45	0.55	0.06	1	1	1	1	1	0.96	0.01	0.03	0.91	0.23	0.78	0.48	0.03	0.02	0.02	1	0	0.01
LMXN	−Inf	1	1	1	0.01	0.38	0.13	0.04	1	1	1	1	1	0.37	0.16	0.05	0.35	0.02	0.02	0.03	0.02	0.12	1	0.02	0.08	0.01
LMXD	−Inf	1	1	1	0.01	0.71	0.1	0.03	1	1	1	1	1	0.1	0.37	0.17	0.98	0.01	0.14	0.02	0	1	0.12	0.02	0.13	0.04
SD	−Inf	1	1	1	0.01	0.17	0.19	0.26	1	1	1	1	1	0.04	0.08	0.04	0.89	0.78	0	0	1	0	0.02	0.03	0.05	0.23
CD	−Inf	1	1	1	0.03	0.9	0.11	0.02	1	1	1	1	1	0.13	0.5	0.11	0.31	0.94	0.01	1	0	0.02	0.03	0.48	0.02	0.38
RD	−Inf	1	1	1	0.02	0.01	0.35	0.08	1	1	1	1	1	0.12	0.75	0.23	0.31	0.32	1	0.01	0	0.14	0.02	0.78	0.08	0.01
TRWD	0.02	1	1	1	0.02	0.71	0.3	0.67	1	1	1	1	1	0.13	0.05	0.75	0.09	1	0.32	0.94	0.78	0.01	0.02	0.23	0.12	0
ART	0.37	1	1	1	0.8	0.5	0.68	0.14	1	1	1	1	1	0.61	0.99	0.32	1	0.09	0.31	0.31	0.89	0.98	0.35	0.91	0.38	0.28
RV	0.02	1	1	1	0.13	0.01	0	0.01	1	1	1	1	1	0.07	0.01	1	0.32	0.75	0.23	0.11	0.04	0.17	0.05	0.03	0.07	0.01
SA	−Inf	1	1	1	0.12	0.01	0.01	0.2	1	1	1	1	1	0.14	1	0.01	0.99	0.05	0.75	0.5	0.08	0.37	0.16	0.01	0.55	0.11
MRL	0.08	1	1	1	0.36	0.05	0.76	0.39	1	1	1	1	1	1	0.14	0.07	0.61	0.13	0.12	0.13	0.04	0.1	0.37	0.96	0.3	0.03
RL35	0.27	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
RL3035	0.23	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
RL2530	0.21	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
RL2025	−Inf	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
RL1520	−Inf	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
RL1015	−Inf	1	1	1	0.87	0.15	0	1	1	1	1	1	1	0.39	0.2	0.01	0.14	0.67	0.08	0.02	0.26	0.03	0.04	0.06	0.4	0
RL0510	0.05	1	1	1	0	0.14	1	0	1	1	1	1	1	0.76	0.01	0	0.68	0.3	0.35	0.11	0.19	0.1	0.13	0.55	0.95	0.23
RL005	−Inf	1	1	1	0.15	1	0.14	0.15	1	1	1	1	1	0.05	0.01	0.01	0.5	0.71	0.01	0.9	0.17	0.71	0.38	0.45	0.7	0.67
CWT	−Inf	1	1	1	1	0.15	0	0.87	1	1	1	1	1	0.36	0.12	0.13	0.8	0.02	0.02	0.03	0.01	0.01	0.01	0	0	0.23
TLA	−Inf	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
TN	−Inf	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
PHT	−Inf	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
G	1	−Inf	−Inf	−Inf	−Inf	−Inf	0.05	−Inf	−Inf	−Inf	0.21	0.23	0.27	0.08	−Inf	0.02	0.37	0.02	−Inf	−Inf	−Inf	−Inf	−Inf	−Inf	−Inf	0.02

Figure S5. PCgen-reconstruction for the rice-data from Kadam *et al.* (2017), with $\alpha = 0.01$: p-values. For each pair of nodes, the figure displays the highest p-value observed across all conditional independence tests performed for that pair.