Learning Gene Networks Underlying Clinical Phenotypes Using SNP Perturbations

Calvin McCarter,[1] Judie Howrylak,[2] Seyoung Kim[3*]

1 Machine Learning Department, Carnegie Mellon University, calvinm@cmu.edu

2 Pulmonary, Allergy and Critical Care Division, Penn State Milton S. Hershey Medical Center, jhowrylak@pennstatehealth.psu.edu

3 Computational Biology Department, Carnegie Mellon University, sssykim@cs.cmu.edu

* Corresponding author

Abstract

Recent technologies are generating an abundance of genome sequence data and molecular and clinical phenotype data, providing an opportunity to understand the genetic architecture and molecular mechanisms underlying diseases. Previous approaches have largely focused on the co-localization of single-nucleotide polymorphisms (SNPs) associated with clinical and expression traits, each identified from genome-wide association studies and expression quantitative trait locus (eQTL) mapping, and thus have provided only limited capabilities for uncovering the molecular mechanisms behind the SNPs influencing clinical phenotypes. Here we aim to extract rich information on the functional role of trait-perturbing SNPs that goes far beyond this simple co-localization. We introduce a computational framework called Perturb-Net for learning the gene network that modulates the influence of SNPs on phenotypes, using SNPs as naturally occurring perturbation of a biological system. Perturb-Net uses a probabilistic graphical model to directly model both the cascade of perturbation from SNPs to the gene network to the phenotype network and the network at each layer of molecular and clinical phenotypes. Perturb-Net learns the entire model by solving a single optimization problem with an extremely fast algorithm that can analyze human genome-wide data within a few hours. In our analysis of asthma data, for a locus that was previously implicated in asthma susceptibility but for which little is known about the molecular mechanism underlying the association, Perturb-Net revealed the gene network modules that mediate the influence of the SNP on asthma phenotypes. Many genes in this network module were well supported in the literature as asthma-related.

Introduction

One of the key questions in biology is how genetic variation perturbs gene regulatory systems to influence disease susceptibility or other phenotypes in a population. Recent advances in technologies have allowed researchers to obtain genome sequence data along with phenotype data at different levels of biological systems, such as gene expression,[1] proteome,[2] metabolome,[3] and various clinical phenotype data. Combining genome sequence data with various types of molecular and clinical phenotype data in a computational analysis has the potential to reveal the complex molecular mechanisms controlled by different genetic loci that underlie diseases and other phenotypes.

To study gene regulatory systems, many previous works have considered the naturally-occurring perturbation of gene expression by genetic variants such as single nucleotide polymorphisms (SNPs), captured in expression and genotype data collected from a population. Compared to experimental perturbation methods such as gene knockdown[4] and genome editing techniques,[5] SNP perturbation for functional genomics studies has an advantage of being more cost effective, being easily applicable to humans, and being potentially more meaningful subtle perturbations because they exist in nature.[6] However, it comes with the computational challenge of having to isolate the perturbation effect of each individual genetic variant, when a large number of genetic variants are perturbing the gene network simultaneously. Several computational methods have been proposed to address this challenge. Sparse conditional Gaussian graphical models (sCGGMs) have been introduced for simultaneously identifying the gene network and expression quantitative trait loci (eQTLs) from population SNP and expression data.[7,8,9] Many other works have relied on statistically less powerful approaches of identifying eQTLs first and then incorporating the eQTLs in the network learning procedure.[10,11,12]

However, there have been relatively few works on modeling how a gene network perturbed by SNPs mediates the SNP perturbation of phenotypes. Most of the existing methods did not directly address this problem and thus, provided only limited capabilities for uncovering the molecular mechanisms behind the SNP perturbation of clinical phenotypes. Many of the previous approaches were concerned with identifying simply the co-localization of eQTLs and trait-associated SNPs,[13,14,15] each of which were identified in a separate eQTL mapping[1,10,16,17] and a genome-wide association study.[18,19] These methods did not provide a description of the regulatory roles of the trait-associated SNPs beyond their co-localization with eQTLs. The genome-transcriptome-phenome structured association method[20] focused only on identifying eQTLs and trait-associated SNPs, and was concerned with neither learning a gene network nor uncovering its role in modulating SNP effects on phenotypes. A predictive network model for diseases that involves Bayesian networks for gene regulatory networks have been proposed,[21] but this approach relied on an elaborate pipeline of analysis to identify disease-related gene modules and genetic variants that could potentially lead to loss of statistical power.

Here, our goal is to extract rich information on the functional role of trait-perturbing SNPs that goes far beyond the simple co-localization with eQTLs, which was the focus of many of the previous studies.[13,14,15] Towards this goal, we introduce a computational framework called Perturb-Net for directly modeling and learning the gene network that modulates the influence of SNPs on phenotypes, using SNPs as naturally occurring perturbation of a biological system. Perturb-Net builds on the key idea in the previous work on sCGGMs[7,8] for learning a gene network using SNP perturbations, and models the cascade of a gene network and a phenotype netowrk under SNP perturbations as a cascade of sCGGMs, called a sparse Gaussian chain graph model (Figure 1A). Our probabilistic graphical model framework naturally leads to a set of inference algorithms for inferring a detailed description of how different parts of the gene network mediate the influence of SNPs on phenotypes, given the model estimated from population genotype, expression, and phenotype data (Figure 1B). The Perturb-Net model and inference procedures together provide a powerful tool for studying the gene regulatory mechanisms whose perturbations by SNPs lead to diseases.

We present a statistically powerful and extremely efficient algorithm for learning the Perturb-Net model. The Perturb-Net learning algorithm is statistically powerful, since it estimates the entire model by solving a single optimization problem with minimal loss of statistical power and with a guarantee in finding the optimal solution due to the convexity of the optimization problem. The Perturb-Net learning algorithm is also computationally efficient and can analyze human genome-wide data with 500,000 SNPs, 11,000 gene expression levels, and several dozens of phenotype data within a few hours. The performance of the Perturb-Net learning algorithm directly depends on that of sCGGM optimization, since it uses the sCGGM learning algorithm as a key module. The previous state-of-the-art method[22] had limited scalability due to expensive computation time and large memory requirement, requiring more than 4 hours for only 10,000 SNPs and running out of memory for 40,000 SNPs. We present a new learning algorithm Fast-sCGGM and its extension Mega-sCGGM with orders-of-magnitude speed-up in computation time that runs on a single machine without running out of memory and that is parallelizable. Our new sCGGM learning algorithms allow Perturb-Net to be applied to human genome-wide data.

We demonstrate Perturb-Net on the data collected for participants in the Childhood Asthma Management Program (CAMP).[23,24,25] Perturb-Net revealed the asthma gene network and how different parts of this gene network mediate the SNP perturbations of phenotypes. Furthermore, for a locus that was previously implicated in asthma susceptibility but for which little has been known about the molecular mechanism underlying the association, Perturb-Net revealed the gene network modules that mediate the influence of the SNP on asthma phenotypes. Many genes in this network module were well supported in the literature as asthma-related, suggesting our framework can reveal the molecular mechanisms underlying the SNP perturbations of phenotypes.

Material and Methods

We describe the model and learning/inference algorithms for Perturb-Net for learning the gene network under SNP perturbations that underlies clinical phenotypes.

Perturb-Net model

Let $\mathbf{x} \in \{0, 1, 2\}^p$ denote minor allele frequencies at p loci of an individual, $\mathbf{y} \in \mathbb{R}^q$ expression levels for q genes, and $\mathbf{z} \in \mathbb{R}^r$ measurements for r phenotypes. Then, Perturb-Net models the cascaded influence of SNPs on a gene network and a phenotype network as a Gaussian chain graph model (Figure 1A), which is a factorized conditional probability distribution defined as follows:

$$p(\mathbf{y}, \mathbf{z}|\mathbf{x}) = p(\mathbf{y}|\mathbf{x})p(\mathbf{z}|\mathbf{y}). \tag{1}$$

Each probability factor above is modeled as a conditional Gaussian graphical model (CGGM):[7,8,22]

$$p(\mathbf{y}|\mathbf{x}) = \exp(-\frac{1}{2}\mathbf{y}^T\mathbf{\Lambda_y}\mathbf{y} - \mathbf{x}^T\mathbf{\Theta_{xy}}\mathbf{y})/Z_1(\mathbf{x}), \tag{2}$$

$$p(\mathbf{z}|\mathbf{y}) = \exp(-\frac{1}{2}\mathbf{z}^T\mathbf{\Lambda_z}\mathbf{z} - \mathbf{y}^T\mathbf{\Theta_{yz}}\mathbf{z})/Z_2(\mathbf{y}). \tag{3}$$

The first probability factor in Eq. (2) models the gene network perturbed by SNPs, representing the gene network as a $q \times q$ positive definite matrix $\mathbf{\Lambda_y}$ and the SNP perturbation of this network as $\mathbf{\Theta_{xy}} \in \mathbb{R}^{p \times q}$. The second probability factor in Eq. (3) models the phenotype network $\mathbf{\Lambda_z}$, a $r \times r$ positive definite matrix, and the perturbation of this network by gene expression levels $\mathbf{\Theta_{yz}} \in \mathbb{R}^{q \times r}$. The $Z_1(\mathbf{x})$ and $Z_2(\mathbf{y})$ in Eqs. (2) and (3) are the constants for ensuring that each CGGM is a proper probability distribution that integrates to one. Our model in Eq. (1) defines a probability distribution over the graph shown in Figure 1A. Thus, a non-zero value in the (i, j)th element of the network parameters, $[\mathbf{\Lambda_y}]_{i,j}$ of $\mathbf{\Lambda_y}$ and $[\mathbf{\Lambda_z}]_{i,j}$ of $\mathbf{\Lambda_z}$, corresponds to presence of an edge between the ith and jth expression or clinical phenotypes. Similarly, a non-zero value in the (i, j)th element of the perturbation parameters, $[\mathbf{\Theta_{xy}}]_{i,j}$ of $\mathbf{\Theta_{xy}}$ and $[\mathbf{\Theta_{yz}}]_{i,j}$ of $\mathbf{\Theta_{yz}}$, indicates an edge between the ith perturbant and the jth expression or clinical phenotype.

This Gaussian chain graph model corresponds to the continuous counterpart of the chain graph model obtained by threading conditional random fields (CRFs) for discrete random variables. CRFs and the chain graph models built from CRFs have been hugely popular in other application areas of statistical machine learning such as text modeling and image analysis for modeling multiple correlated output features influenced by input features.[26,27,28] Here, we explore the use of a chain graph model constructed with sCGGMs, corresponding to Gaussian CRFs, and develop an extremely fast learning algorithm that runs on human data within a few hours and a set of inference algorithms for dissecting the gene regulatory mechanisms that govern the influence of SNPs on phenotypes.

Perturb-Net learning algorithms

We present an extremely efficient algorithm for obtaining a sparse estimate of the model parameters with few edges in the graph. The Perturb-Net learning algorithm minimizes the negative log-likelihood of data with L_1 regularization,[29] which is a convex optimization problem with a guarantee in finding the optimal solution. Given genotype data $\mathbf{X} \in \mathbb{R}^{n \times p}$ for n samples and p SNPs, expression data $\mathbf{Y} \in \mathbb{R}^{n \times q}$ for q genes, and phenotype data $\mathbf{Z} \in \mathbb{R}^{n \times r}$ for r phenotypes for the same n samples, we estimate a sparse Gaussian chain graph model in Eq. (1) by minimizing the negative log-likelihood of data along with sparsity-inducing L_1 penalty:

$$\min_{\mathbf{\Lambda_y}\succ 0,\mathbf{\Theta_{xy}},\mathbf{\Lambda_z}\succ 0,\mathbf{\Theta_{yz}}} f(\mathbf{\Lambda_y},\mathbf{\Theta_{xy}}) + f(\mathbf{\Lambda_z},\mathbf{\Theta_{yz}}), \tag{4}$$

where

$$\begin{aligned} f(\mathbf{\Lambda_y},\mathbf{\Theta_{xy}}) &= -\log|\mathbf{\Lambda_y}| + \mathrm{tr}(\mathbf{S_{yy}}\mathbf{\Lambda_y} + 2\mathbf{S_{xy}}^T\mathbf{\Theta_{xy}} + \mathbf{\Lambda_y}^{-1}\mathbf{\Theta}_{\mathbf{xy}}^T\mathbf{S_{xx}}\mathbf{\Theta_{xy}}) \\ &\quad + \lambda_{\mathbf{\Lambda_y}}\|\mathbf{\Lambda_y}\|_1 + \lambda_{\mathbf{\Theta_{xy}}}\|\mathbf{\Theta_{xy}}\|_1 \\ f(\mathbf{\Lambda_z},\mathbf{\Theta_{yz}}) &= -\log|\mathbf{\Lambda_z}| + \mathrm{tr}(\mathbf{S_{zz}}\mathbf{\Lambda_z} + 2\mathbf{S_{yz}}^T\mathbf{\Theta_{yz}} + \mathbf{\Lambda_z}^{-1}\mathbf{\Theta}_{\mathbf{yz}}^T\mathbf{S_{yy}}\mathbf{\Theta_{yz}}) \\ &\quad + \lambda_{\mathbf{\Lambda_z}}\|\mathbf{\Lambda_z}\|_1 + \lambda_{\mathbf{\Theta_{yz}}}\|\mathbf{\Theta_{yz}}\|_1, \end{aligned}$$

given data covariance matrices $\mathbf{S_{xx}} = \frac{1}{n}\mathbf{X}^T\mathbf{X}$, $\mathbf{S_{xy}} = \frac{1}{n}\mathbf{X}^T\mathbf{Y}$, $\mathbf{S_{yy}} = \frac{1}{n}\mathbf{Y}^T\mathbf{Y}$, $\mathbf{S_{yz}} = \frac{1}{n}\mathbf{Y}^T\mathbf{Z}$, and $\mathbf{S_{zz}} = \frac{1}{n}\mathbf{Z}^T\mathbf{Z}$, and $\|\cdot\|_1$ for the non-smooth elementwise L_1 penalty. The regularization parameters $\lambda_{\mathbf{\Lambda_y}}, \lambda_{\mathbf{\Theta_{xy}}}, \lambda_{\mathbf{\Lambda_z}}, \lambda_{\mathbf{\Theta_{yz}}} > 0$ are chosen to maximize the Bayesian information criterion (BIC). We do not penalize the diagonal entries of $\mathbf{\Lambda_y}$ and $\mathbf{\Lambda_z}$, following the common practice for sparse inverse covariance estimation.

The above optimization problem decouples into two subproblems, each containing one of two disjoint sets of parameters $\{\mathbf{\Lambda_y}, \mathbf{\Theta_{xy}}\}$ and $\{\mathbf{\Lambda_z}, \mathbf{\Theta_{yz}}\}$, each of which can be solved with an sCGGM optimization algorithm. Since our learning algorithm uses an sCGGM learning method as a key module, we developed sCGGM learning algorithms called Fast-sCGGM for reducing computation time and Mega-sCGGM for futher improving Fast-sCGGM to remove the memory constraint, both of which are parallelizable over multiple cores of a machine. Fast-sCGGM improves the computation time of the previous method by alternately optimizing the network parameter ($\mathbf{\Lambda_y}$ and $\mathbf{\Lambda_z}$) and the perturbation parameters ($\mathbf{\Theta_{xy}}$ and $\mathbf{\Theta_{yz}}$), where each of the alternate optimization can be efficiently solved using the fast Lasso optimization technque as the key subroutine (see Appendix A for detail).

While Fast-sCGGM improves the computation time of the previous method, it is limited by the memory size required to store large $q \times q$ or $p \times q$ matrices during the iterative optimization. A naive approach to reduce the memory footprint would be to recompute portions of these matrices on demand for each coordinate update, which would be very expensive. Hence, we combine Fast-sCGGM with block coordinate descent and introduce the Mega-sCGGM algorithm to scale up the optimization to very large problems on a machine with limited memory. During the iterative optimization, we update blocks of the large matrices so that within each block, the computation of the large matrices can be cached and re-used. These blocks are determined automatically in each iteration by exploiting the sparse stucture (see Appendix A for detail).

We introduce a modification of our learning algorithm for semi-supervised learning, to handle the situation where expression data are available only for a subset of individuals because of the difficulty of obtaining tissue samples. This modification corresponds to an expectation maximization (EM) algorithm[30] that imputes the missing expression levels in the E-step and performs our Fast-sCGGM or Mega-sCGGM optimization in the M-step. For semi-supervised learning, given a dataset $\mathcal{D} = \{\mathcal{D}_o, \mathcal{D}_h\}$, where $\mathcal{D}_o = \{\mathbf{X}_o, \mathbf{Y}_o, \mathbf{Z}_o\}$ for the fully-observed data and $\mathcal{D}_h = \{\mathbf{X}_h, \mathbf{Z}_h\}$ for the samples with missing gene-expression levels, we adopt an EM algorithm[30] that iteratively maximizes the expected log-likelihood of data:

$$\mathcal{L}(\mathcal{D}_o; \mathbf{\Theta}) + \mathrm{E}\big[\mathcal{L}(\mathcal{D}_h, \mathbf{Y}_h; \mathbf{\Theta})\big],$$

combined with L_1-regularization, where $\mathcal{L}(\mathcal{D}_o; \mathbf{\Theta})$ and $\mathcal{L}(\mathcal{D}_h; \mathbf{\Theta})$ are the log-likelihood of data $\mathcal{D}_o$ and $\mathcal{D}_h$ with respect to the model in Eq. (1) and the expectation is taken with respect to:

$$p(\mathbf{y}|\mathbf{z}, \mathbf{x}) = N(\mu_{\mathbf{y}|\mathbf{x},\mathbf{z}}, \mathbf{\Sigma}_{\mathbf{y}|\mathbf{x},\mathbf{z}}), \tag{5}$$
$$\mu_{\mathbf{y}|\mathbf{x},\mathbf{z}} = -\mathbf{\Sigma}_{\mathbf{y}|\mathbf{x},\mathbf{z}}(\mathbf{\Theta}_{\mathbf{yz}}\mathbf{z} + \mathbf{\Theta}_{\mathbf{xy}}^T\mathbf{x}) \quad \text{and} \quad \mathbf{\Sigma}_{\mathbf{y}|\mathbf{x},\mathbf{z}} = (\mathbf{\Lambda}_{\mathbf{y}} + \mathbf{\Theta}_{\mathbf{yz}}\mathbf{\Lambda}_{\mathbf{z}}\mathbf{\Theta}_{\mathbf{yz}}^T)^{-1}.$$

A naive implementation of this EM algorithm leads to an algorithm that requires expensive computation time and large storage of dense matrices that exceeds the computer memory. To make the EM algorithm efficient in terms of both time and memory, we embed the expensive E-step computation within the M-step, using a low-rank representation of dense matrices (see Appendix B). This implementation produces the same estimate as the original EM algorithm.

Perturb-Net inference procedures

While the sparse Gaussian chain graph model explicitly represents pair-wise dependencies among variables as edges in the graph, there are other dependencies that are only implicitly represented in the model but can be revealed by performing an inference on the estimated probabilistic graphical model. Here, we provide an overview of the inferred dependencies, all of which involve simple matrix operations.

The following two inference methods directly follow from the inference method for an sCGGM (Figure 1A),[7,8] which infers the indirect perturbation effects that arise from the direct perturbation effects propagating to other parts of the network.

- **Indirect SNP perturbation effects on gene expression levels: $\mathbf{B_{xy}} = -\mathbf{\Theta_{xy}}\mathbf{\Lambda_y}^{-1}$,** where $[\mathbf{B_{xy}}]_{i,j}$ represents the indirect perturbation effect of SNP i on the expression level of gene j (blue dashed arrow in Figure 1A). This can be seen by deriving the marginal distribution from the sCGGM component model $p(\mathbf{y}|\mathbf{x})$ as follows:

$$p(\mathbf{y}|\mathbf{x}) = N(\mathbf{B}_{\mathbf{xy}}^T\mathbf{x}, \mathbf{\Lambda_y}^{-1}). \tag{6}$$

  From Eq. (6), the marginal distribution for the expression level $[\mathbf{y}]_i$ of gene i can be obtained as $p([\mathbf{y}]_i|\mathbf{x}) = N\big([[\mathbf{B_{xy}}]_{:,i}]^T\mathbf{x}, [\mathbf{\Lambda_y}^{-1}]_{i,i}\big)$. While $[\mathbf{\Theta_{xy}}]_{i,j}$ represents the direct perturbation effect of SNP i on the expression of gene j, $[\mathbf{B_{xy}}]_{i,j}$ represents the overall perturbation effect that aggregates all indirect influence of this SNP on gene j through other genes. When SNP i does not influence the expression of gene j directly but exerts influence on gene j through other genes connected to gene j in the network $\mathbf{\Lambda_y}$, we have $[\mathbf{\Theta_{xy}}]_{i,j} = 0$ but $[\mathbf{B_{xy}}]_{i,j} \neq 0$.

- **Indirect effects of gene expression levels on clinical phenotypes: $\mathbf{B_{yz}} = -\mathbf{\Theta_{yz}}\mathbf{\Lambda_z}^{-1}$,** where $[\mathbf{B_{yz}}]_{i,j}$ represents the indirect influence of the expression level of gene i on phenotype j (red dashed arrow in Figure 1A). Similarly as above, this can be seen by deriving the marginal distribution from the sCGGM component model $p(\mathbf{z}|\mathbf{y})$, as follows:

$$p(\mathbf{z}|\mathbf{y}) = N(\mathbf{B}_{\mathbf{yz}}^T\mathbf{y}, {\mathbf{\Lambda}_{\mathbf{z}}}^{-1}).$$

Then, the marginal distribution for $[\mathbf{z}]_i$ of phenotype i can be obtained as $p([\mathbf{z}]_i|\mathbf{y})$ $= N\big([[\mathbf{B}_{\mathbf{yz}}]_{:,i}]^T\mathbf{y}, [{\mathbf{\Lambda}_{\mathbf{z}}}^{-1}]_{i,i}\big)$. While $[\mathbf{\Theta}_{\mathbf{yz}}]_{i,j}$ represents the direct influence of gene i on phenotype j, $[\mathbf{B}_{\mathbf{yz}}]_{i,j}$ represents the overall influence that aggregates all indirect influence of this expression level on phenotype j through other phenotypes.

The sparse Gaussian chain graph model provides the following additional inference procedures for extracting the information on whether SNP perturbation effects on the gene network reach the phenotypes and how different genes or subnetworks of the gene network mediate SNP effects on phenotypes (Figure 1B).

- **SNP effects on clinical phenotypes:** $\mathbf{B}_{\mathbf{xz}} = \mathbf{B}_{\mathbf{xy}}\mathbf{B}_{\mathbf{yz}}$, where $[\mathbf{B}_{\mathbf{xz}}]_{i,j}$ represents the overall influence of SNP i on phenotype j mediated by gene expression levels in gene network $\mathbf{\Lambda}_{\mathbf{y}}$ (purple dashed arrow in Figure 1B). The effects of SNPs on phenotypes are not directly modeled in our model but can be inferred by deriving the marginal distribution $p(\mathbf{z}|\mathbf{x})$ as follows:

$$p(\mathbf{z}|\mathbf{x}) = N(\mathbf{B}_{\mathbf{xz}}^T\mathbf{x}, {\mathbf{\Lambda}_{\mathbf{z}}}^{-1} + {\mathbf{\Lambda}_{\mathbf{z}}}^{-1}\mathbf{\Theta}_{\mathbf{yz}}^T{\mathbf{\Lambda}_{\mathbf{y}}}^{-1}\mathbf{\Theta}_{\mathbf{yz}}{\mathbf{\Lambda}_{\mathbf{z}}}^{-1}).$$

  The marginal distribution for the phenotype $[\mathbf{z}]_i$ of phenotype i given $\mathbf{x}$ can be obtained as $p([\mathbf{z}]_i|\mathbf{x}) = N\big([[\mathbf{B}_{\mathbf{xz}}]_{:,i}]^T\mathbf{x}, [{\mathbf{\Lambda}_{\mathbf{z}}}^{-1} + {\mathbf{\Lambda}_{\mathbf{z}}}^{-1}\mathbf{\Theta}_{\mathbf{yz}}^T{\mathbf{\Lambda}_{\mathbf{y}}}^{-1}\mathbf{\Theta}_{\mathbf{yz}}{\mathbf{\Lambda}_{\mathbf{z}}}^{-1}]_{i,i}\big)$, where each element $[\mathbf{B}_{\mathbf{xz}}]_{i,j}$ represents the overall influence of SNP i on phenotype j mediated by the gene network in $\mathbf{\Lambda}_{\mathbf{y}}$ and other phenotypes connected to phenotype j in $\mathbf{\Lambda}_{\mathbf{z}}$.

- **SNP effects on clinical phenotypes mediated by a gene module:** The overall SNP effects on phenotypes in $\mathbf{B}_{\mathbf{xz}}$ above can be decomposed into the SNP effects on phenotypes mediated by each gene module. Let M be a gene module that consists of a subset of the q genes whose expression levels were modeled in $\mathbf{\Lambda}_{\mathbf{y}}$ (yellow and orange gene modules in Figure

1B). Then, the effects of SNPs on phenotypes mediated by the genes in module M can be obtained as follows:

$$\mathbf{B}_{\mathbf{xz}}^{M} = \sum_{k \in M} [\mathbf{B}_{\mathbf{xy}}]_{:,k} [\mathbf{B}_{\mathbf{yz}}]_{k,:},$$

where $[\mathbf{B}_{\mathbf{xy}}]_{:,a}$ represents the ath row of $\mathbf{B}_{\mathbf{xy}}$ and $[\mathbf{B}_{\mathbf{yz}}]_{b,:}$ represents the bth column of $\mathbf{B}_{\mathbf{yz}}$. In the above equation, $[\mathbf{B}_{\mathbf{xz}}^{M}]_{i,j}$ quantifies the effect of SNP i on phenotype j through the expression levels of genes in module M. If $M_1, \ldots, M_s$ are disjoint subsets of q genes, where $\cup_{m=1,\ldots,s} M_m$ is the full set of q genes, we have the following decomposition:

$$\mathbf{B}_{\mathbf{xz}} = \sum_{m=1}^{s} \mathbf{B}_{\mathbf{xz}}^{M_m}.$$

- **Inferred dependencies among genes after seeing phenotype data:** $\mathbf{\Lambda}_{\mathbf{y}|\mathbf{x},\mathbf{z}} = \mathbf{\Lambda}_{\mathbf{y}} + \mathbf{\Theta}_{\mathbf{yz}} \mathbf{\Lambda}_{\mathbf{z}}{}^{-1} \mathbf{\Theta}_{\mathbf{yz}}^{T}$ represents gene network $\mathbf{\Lambda}_{\mathbf{y}}$ augmented with the component $\mathbf{\Theta}_{\mathbf{yz}} \mathbf{\Lambda}_{\mathbf{z}}{}^{-1} \mathbf{\Theta}_{\mathbf{yz}}^{T}$ introduced through dependencies in phenotype network $\mathbf{\Lambda}_{\mathbf{z}}$ (blue dashed edge in Figure 1B). In this augmented network, additional edges are introduced between two genes if their expression levels influence the same trait or if they both affect traits that are connected in the phenotype network $\mathbf{\Lambda}_{\mathbf{z}}$. The posterior gene network $\mathbf{\Lambda}_{\mathbf{y}|\mathbf{x},\mathbf{z}}$, which contains the dependencies among expression levels after taking into account phenotype data, can be obtained by inferring the posterior distribution given phenotypes from the estimated Gaussian chain graph model as follows:

$$p(\mathbf{y}|\mathbf{x},\mathbf{z}) = N\Big(-\big(\mathbf{z}^T \mathbf{\Theta}_{\mathbf{yz}}^{T} + \mathbf{x}^T \mathbf{\Theta}_{\mathbf{xy}}\big) \mathbf{\Lambda}_{\mathbf{y}|\mathbf{x},\mathbf{z}}^{-1},\ \mathbf{\Lambda}_{\mathbf{y}|\mathbf{x},\mathbf{z}}^{-1} \Big),$$

where

$$\mathbf{\Lambda}_{\mathbf{y}|\mathbf{x},\mathbf{z}} = \mathbf{\Lambda}_{\mathbf{y}} + \mathbf{\Theta}_{\mathbf{yz}} \mathbf{\Lambda}_{\mathbf{z}}{}^{-1} \mathbf{\Theta}_{\mathbf{yz}}^{T}.$$

The inferred network $\mathbf{\Lambda}_{\mathbf{y}|\mathbf{x},\mathbf{z}}$ can also be seen by inferring from the estimated model the joint distribution

$$p(\mathbf{z},\mathbf{y}|\mathbf{x}) = N\Big(-\boldsymbol{\Lambda}^{-1}_{(\mathbf{z},\mathbf{y})}\boldsymbol{\Theta}^{T}_{(\mathbf{yz},\mathbf{xy})}\mathbf{x}, \boldsymbol{\Lambda}^{-1}_{(\mathbf{z},\mathbf{y})}\Big),$$

where $\boldsymbol{\Theta}_{(\mathbf{yz},\mathbf{xy})} = (\mathbf{0}_{p\times r}, \boldsymbol{\Theta}_{\mathbf{xy}})$ with $p \times r$ matrix of 0's and $\boldsymbol{\Lambda}_{(\mathbf{z},\mathbf{y})} = \begin{pmatrix} \boldsymbol{\Lambda}_{\mathbf{z}} & \boldsymbol{\Theta}^{T}_{\mathbf{yz}} \\ \boldsymbol{\Theta}_{\mathbf{yz}} & \boldsymbol{\Lambda}_{\mathbf{y}|\mathbf{x},\mathbf{z}} \end{pmatrix}$. This joint distribution is an alternative representation of the same Gaussian chain graph model in Eq. (1) and corresponds to another sCGGM over $\mathbf{y}$ and $\mathbf{z}$ conditional on $\mathbf{x}$. This process of introducing the additional dependencies via $\boldsymbol{\Theta}_{\mathbf{yz}}\boldsymbol{\Lambda}_{\mathbf{z}}{}^{-1}\boldsymbol{\Theta}^{T}_{\mathbf{yz}}$ in this new sCGGM, which is equivalent to the original chain graph model, is also known as moralization in the probabilistic graphical model literature.[27]

Prediction tasks

We use the estimated Perturb-Net model and the results of probabilistic inference on this model to make predictions on previously unseen patients. From each of the two component sCGGMs in our model, we make the following predictions:

- $\hat{\mathbf{y}}_{\text{new}}|\mathbf{x}_{\text{new}} = \mathbf{B}^{T}_{\mathbf{xy}}\mathbf{x}_{\text{new}}$ for predicting the expression levels $\hat{\mathbf{y}}_{\text{new}}$ given the genotypes $\mathbf{x}_{\text{new}}$ of a new patient
- $\hat{\mathbf{z}}_{\text{new}}|\mathbf{y}_{\text{new}} = \mathbf{B}^{T}_{\mathbf{yz}}\mathbf{y}_{\text{new}}$ for predicting the phenotypes $\hat{\mathbf{z}}_{\text{new}}$ given the expression levels $\mathbf{y}_{\text{new}}$ of a new patient

From the full sparse Gaussian chain graph model, we make the following predictions:

- $\hat{\mathbf{z}}_{\text{new}}|\mathbf{x}_{\text{new}} = \mathbf{B}^{T}_{\mathbf{xz}}\mathbf{x}_{\text{new}}$ for predicting the phenotypes $\hat{\mathbf{z}}_{\text{new}}$ given the genotypes $\mathbf{x}_{\text{new}}$ of a new patient
- $\hat{\mathbf{y}}_{\text{new}}|\mathbf{x}_{\text{new}}, \mathbf{z}_{\text{new}} = -\Big(\mathbf{z}^{T}_{\text{new}}\boldsymbol{\Theta}^{T}_{\mathbf{yz}} + \mathbf{x}^{T}_{\text{new}}\boldsymbol{\Theta}_{\mathbf{xy}}\Big)\boldsymbol{\Lambda}^{-1}_{\mathbf{y}|\mathbf{x},\mathbf{z}}$ for predicting the gene expression levels $\hat{\mathbf{y}}_{\text{new}}$ given the genotypes $\mathbf{x}_{\text{new}}$ and the phenotypes $\mathbf{x}_{\text{new}}$ of a new patient

Lasso for comparison with our algorithms

We compare the performance of our method with that of Lasso,[29,31] a popular statistical method based on linear regression models for studying the associations among SNPs, expression measurements, and phenotypes. We begin by setting up a two-layer multivariate regression model for genotypes $\mathbf{x} \in \{0, 1, 2\}^p$, expression measurements $\mathbf{y} \in \mathbb{R}^q$, and phenotypes $\mathbf{z} \in \mathbb{R}^r$ as follows:

$$\mathbf{y} = \mathbf{A}_{\mathbf{xy}}^T \mathbf{x} + \epsilon_{\mathbf{y}}, \quad \epsilon_{\mathbf{y}} \sim \mathcal{N}(\mathbf{0}_q, \mathbf{\Omega}_{\mathbf{y}}),$$
$$\mathbf{z} = \mathbf{A}_{\mathbf{yz}}^T \mathbf{y} + \epsilon_{\mathbf{z}}, \quad \epsilon_{\mathbf{z}} \sim \mathcal{N}(\mathbf{0}_r, \mathbf{\Omega}_{\mathbf{z}}),$$

where $\mathbf{A}_{\mathbf{xy}} \in \mathbb{R}^{p \times q}$ and $\mathbf{A}_{\mathbf{yz}} \in \mathbb{R}^{q \times r}$ are regression coefficients, $\epsilon_{\mathbf{y}} \in \mathbb{R}^q$ and $\epsilon_{\mathbf{z}} \in \mathbb{R}^r$ are noise distributed with zero means and diagonal covariances $\mathbf{\Omega}_{\mathbf{y}} = \text{diag}(\sigma^2_{\mathbf{y}_1}, \ldots, \sigma^2_{\mathbf{y}_q})$ and $\mathbf{\Omega}_{\mathbf{z}} = \text{diag}(\sigma^2_{\mathbf{z}_1}, \ldots, \sigma^2_{\mathbf{z}_q})$.

Given genotype data $\mathbf{X} \in \{0, 1, 2\}^{n \times p}$ for n samples and p SNPs, expression data $\mathbf{Y} \in \mathbb{R}^{n \times q}$ for q genes, and phenotype data $\mathbf{Z} \in \mathbb{R}^{n \times r}$ for r phenotypes, we obtain a Lasso estimate of the regression coefficients by minimizing L_1-regularized negative log-likelihood as follows:

$$\min_{\mathbf{A}_{\mathbf{xy}}} \frac{1}{n} \text{tr}\left(\left(\mathbf{Y} - \mathbf{X}^T \mathbf{A}_{\mathbf{xy}}\right) \left(\mathbf{Y} - \mathbf{X}^T \mathbf{A}_{\mathbf{xy}}\right)^T \right) + \gamma_1 ||\mathbf{A}_{\mathbf{xy}}||_1,$$
$$\min_{\mathbf{A}_{\mathbf{yz}}} \frac{1}{n} \text{tr}\left(\left(\mathbf{Z} - \mathbf{Y}^T \mathbf{A}_{\mathbf{yz}}\right) \left(\mathbf{Z} - \mathbf{Y}^T \mathbf{A}_{\mathbf{yz}}\right) \right)^T + \gamma_2 ||\mathbf{A}_{\mathbf{yz}}||_1.$$

Using the Lasso estimate of the regression coefficients $\mathbf{A}_{\mathbf{xy}}$ and $\mathbf{A}_{\mathbf{yz}}$, we compute predictions for this model analogously to our sparse Gaussian chain graph model.

- $\hat{\mathbf{y}}_{\text{new}} | \mathbf{x}_{\text{new}} = \mathbf{A}_{\mathbf{xy}}^T \mathbf{x}_{\text{new}}$
- $\hat{\mathbf{z}}_{\text{new}} | \mathbf{y}_{\text{new}} = \mathbf{A}_{\mathbf{yz}}^T \mathbf{y}_{\text{new}}$
- $\hat{\mathbf{z}}_{\text{new}} | \mathbf{x}_{\text{new}} = \mathbf{A}_{\mathbf{xz}}^T \mathbf{z}_{\text{new}}$, where $\mathbf{A}_{\mathbf{xz}} = \mathbf{A}_{\mathbf{xy}} \mathbf{A}_{\mathbf{yz}}$.
- $\hat{\mathbf{y}}_{\text{new}} | \mathbf{x}_{\text{new}}, \mathbf{z}_{\text{new}} = \left(\mathbf{z}_{\text{new}}^T \mathbf{\Omega}_{\mathbf{z}} \mathbf{A}_{\mathbf{yz}}^T + \mathbf{x}_{\text{new}}^T \mathbf{A}_{\mathbf{xy}} \mathbf{\Omega}_{\mathbf{z}}\right) \mathbf{\Omega}_{\mathbf{y}|\mathbf{x},\mathbf{z}}$, where $\mathbf{\Omega}_{\mathbf{y}|\mathbf{x},\mathbf{z}} = \left([\mathbf{\Omega}_{\mathbf{y}}]^{-1} + \mathbf{A}_{\mathbf{yz}} [\mathbf{\Omega}_{\mathbf{z}}]^{-1} \mathbf{A}_{\mathbf{yz}}{}^T\right)^{-1}$. For this prediction task, we estimate the variances as

follows:[32]

$$\sigma^2_{\mathbf{y}_i} = \frac{1}{n - s_{\mathbf{y}_i}} \left([\mathbf{Y}]_{:,i} - \mathbf{X}^T[\mathbf{A}_{\mathbf{xy}}]_{:,i}\right)^T \left([\mathbf{Y}]_{:,i} - \mathbf{X}^T[\mathbf{A}_{\mathbf{xy}}]_{:,i}\right), \quad \text{for } i = 1, \dots, q,$$
$$\sigma^2_{\mathbf{z}_i} = \frac{1}{n - s_{\mathbf{z}_i}} \left([\mathbf{Z}]_{:,i} - \mathbf{Y}^T[\mathbf{A}_{\mathbf{yz}}]_{:,i}\right)^T \left([\mathbf{Z}]_{:,i} - \mathbf{Y}^T[\mathbf{A}_{\mathbf{yz}}]_{:,i}\right) \quad \text{for } i = 1, \dots, r,$$

where $s_{\mathbf{y}_i}$ and $s_{\mathbf{z}_i}$ are the numbers of non-zero entries in $[\mathbf{A}_{\mathbf{xy}}]_{:,i}$ and $[\mathbf{A}_{\mathbf{yz}}]_{:,i}$ respectively.

Preparation of asthma dataset

We applied our method to a dataset comprising genotype, gene expression, and clinical phenotype data, collected from asthma patients participating in CAMP study.[23,24,25] We used 174 non-Hispanic Caucasian subjects for whom both genotype and clinical phenotype data were available. For a subset of 140 individuals, gene expression data from primary peripheral blood CD4+ lymphocytes were also available. After removing SNPs with minor allele frequency less than 0.1 and those with missing reference SNP ids, we obtained 495,597 SNPs for autosomal chromosomes. We then imputed missing genotypes using fastPHASE.[33] Given expression levels for 22,184 mRNA transcripts profiled with Illumina HumanRef8 v2 BeadChip arrays,[25] we removed transcript levels with expression variance less than 0.01, which resulted in a set of 11,598 expression levels to be used in our analysis. Then, we converted the expression values to their z-scores. The clinical phenotype data comprised 35 phenotypes (Table S1), including 25 features related to lung function and 10 features collected via blood testing. The clinical phenotypes were converted to their z-scores within each phenotype so that all phenotypes have equal variance. We then imputed missing values using low-rank matrix completion.[34]

Comparison of the computation time of different algorithms

In order to compare the computation of different algorithms, we used the following software and hardware setup. For Lasso, we used the implementation in GLMNET[35] with a backend written in Fortran. For Newton coordinate descent, which is the previous state-of-the-art approach for optimizing sCGGMs, we took the implementation written in C++ provided by the authors[22] and sped up this implementation with the Eigen matrix library, by employing low-rank matrix representations and using sparse matrix multiplications. For all methods, the code was compiled and run with OpenMP multi-threading enabled on the same machines with 20Gb of memory and 16 cores. We used the same regularization parameters for our method and the previous method for sCGGM optimization, so the resulting solutions were identical with the same sparsity levels. For Lasso, we chose the regularization parameters so that the L_1-norm of the regression matrix roughly matched that of our inferred indirect SNP effects.

Results

Comparison of the scalability of Mega-sCGGM and other methods

We assess the scalability of Mega-sCGGM and other previous algorithms on the expression measurements of 11,598 genes and the genotypes of 495,597 SNPs for 140 subjects from the CAMP data. We estimated sCGGMs, using both our new method and the previous state-of-the-art method based on the Newton coordinate descent method.[22] Since the sCGGM optimization problem is convex with a single globally optimal solution, both our and previous methods obtain the same parameter estimates, although the computation time differs between the two methods. We also obtained the computation time of Lasso implemented in GLMNET,[29,35] the well-known computationally efficient algorithm for learning a simple but less powerful regression model. Although the sparse multivariate regression with covariance estimation[36] has also provided a methodology that could be used for learning a gene network influenced by SNPs, this approach has been found to take days to learn a model from a small dataset of only 1,000 SNPs and 500 gene expression levels,[8] so we did not include it in our experiment. All of the optimization methods were run on the same hardware setup with comparable software implementations.

In our comparison of different methods, our algorithm significantly outperformed the previous state-of-the-art method for learning an sCGGM in terms of both computation time and memory requirement and scaled similarly to Lasso (Figure 2). In comparison of our method with Lasso on datasets with 40,056 SNPs from chromosome 1, 21,757 SNPs for chromosomes 1 through 6, and

495,597 SNPs from all autosomal chromosomes and all expression measurements, our method was not substantially slower than Lasso, even though our method learns a more expressive model than Lasso. The previous sCGGM optimization algorithm ran out of memory even on the smallest dataset above with SNPs only from chromosome 1, so we compared the two algorithms on a much smaller dataset with 1,000 and 10,000 SNPs. On 10,000 SNPs, the previous algorithm for sCGGM required more than four hours, whereas in less than four hours, our algorithm was able to run on all 495,597 SNPs.

Analysis of asthma data

We now fit a sparse Gaussian chain graph model to the genotype, expression, clinical phenotype data gathered from participants in the Childhood Asthma Management Program (CAMP).[23,24,25] After preprocessing the data, we applied our method to the data from 140 subjects for whom all data were available for 495,597 SNPs on 22 autosomal chromosomes, 11,598 gene expression levels, and 35 phenotypes (Table S1) and 34 additional subjects for whom data were available only for genotypes and phenotypes but not for expression levels. Below we perform a detailed analysis of the estimated model.

Overview of the Perturb-Net model

We first examined the overall estimated model for the module structures in the phenotype and gene networks (Figure 3). To see the structure in the phenotype network $\mathbf{\Lambda_z}$, we reordered the nodes of the network by applying hierarchical clustering to each set of the lung function and blood test phenotypes. This revealed the dense connectivities within the two known groups of phenotypes and the two sub-clusters within the group of lung function phenotypes (Figure 3A).

The gene network $\mathbf{\Lambda_y}$ also showed a clear module structure (Figure 3B). To find the module structure in the network, we identified the genes that are connected to at least one other gene in the network $\mathbf{\Lambda_y}$ and partitioned the network over those genes into 20 subnetworks with roughly equal number of nodes, using the network clustering algorithm METIS.[37] Out of 11,598 genes, 6,102 genes were connected to at least one other gene in the network. For the rest of our analysis, we focus on the network and modules over the 6,102 genes, since these genes are likely to form modules for pathways with a functional impact on asthma phenotypes. Modules 1-15 were densely connected clusters of co-expressed genes, suggesting those modules are likely to consist of a functionally coherent set of genes, whereas modules 16-20 had relatively fewer edge connections within each cluster.

Next, we considered the effects of the gene modules on the lung and blood phenotypes in $\mathbf{\Theta_{yz}}$ and the SNP perturbations of the gene modules in $\mathbf{\Theta_{xy}}$. Modules 1-12 had relatively small effects on the phenotypes despite their dense connectivities, whereas modules 13-20 appeared to have stronger effects on both groups of phenotypes (Figure 3C). The SNP effects on the modules in $\mathbf{\Theta_{xy}}$ for the top 1000 eQTL hotspots, determined by overall SNP effects on all genes ($\sum_j |[\mathbf{\Theta_{xy}}]_{i,j}|$ for each SNP i), showed that many of these hotspots perturb the expression of genes in the same module in the gene network (Figure 3D). Given these observations from the visual inspection of $\mathbf{\Theta_{yz}}$ and $\mathbf{\Theta_{xy}}$, we summarized $\mathbf{\Theta_{yz}}$ and $\mathbf{\Theta_{xy}}$ at module level and compared the module-level summaries across modules. To quantify the module-level influence of expression levels on each group of phenotypes, from the direct influence $\mathbf{\Theta_{yz}}$ and indirect influence $\mathbf{B_{yz}}$ we computed the overall effect sizes of all genes in the given gene module on all phenotypes in each of the lung and blood phenotype groups ($\sum_{i\in M,j\in K} |[\mathbf{\Theta_{yz}}]_{i,j}|$ and $\sum_{i\in M,j\in K} |[\mathbf{B_{yz}}]_{i,j}|$ for each gene module M and phenotype group K). Similarly, from $\mathbf{\Theta_{xy}}$ and $\mathbf{B_{xy}}$ we computed the overall SNP effect sizes on all genes in the given module ($\sum_{i,j\in M} |[\mathbf{\Theta_{xy}}]_{i,j}|$ and $\sum_{i,j\in M} |[\mathbf{B_{xy}}]_{i,j}|$ for each SNP i and module M).

Among the 20 gene modules, modules 13-20 overall had stronger influence on both lung and blood phenotypes than the other gene modules (Figures 4A and 4B), although SNP perturbations were found across all gene modules without any preference to those modules with stronger influence on phenotypes (Figure 4C). For modules 13-20, the overall effect sizes on the lung phenotypes ranged between 0.8 and 7.5 for direct and indirect influence with an exception of module 14, whereas for modules 1-12, the overall effect sizes were less than 0.8 (Figure 4A). Modules 13-20 also had strong effects on the blood phenotypes (Figure 4B), although module 13 had substantially stronger effect on the blood phenotypes than on the lung phenotypes. On the other hand, the overall SNP effects were similar across all gene modules for both the direct and indirect SNP effects (Figure 4B). The overall indirect SNP effects were larger for some modules (e.g., module 14), but this was largely because of the substantially stronger edge connectivities in that module, which led to stronger propagation of the direct SNP perturbation effects.

Gene modules that influence phenotypes are enriched for immune genes

To determine the functional role of the gene modules, we performed gene ontology (GO) gene set enrichment analysis.[38,39] For each module, we performed a Fisher's exact test to find the significantly enriched GO categories in biological processes (p-value < 0.05 after Bonferroni correction for multiple testing), using the GO database with annotations for 21,002 genes.

Among all 20 modules, modules 13-15 had a statistically significant enrichment of GO terms related to immune system function, which also corresponded to the most significant enrichments across all modules (Table 1). Even though modules 16-20 did not have any significant enrichment of asthma-related GO categories, many of the genes in these modules were connected to genes in modules 13-15 in the posterior gene network $\mathbf{\Lambda}_{\mathbf{y}|\mathbf{x},\mathbf{z}}$ (Figure 5), and thus this subset of genes in modules 16-20 may be also involved with immune system function. To see if this is indeed the case, we obtained the significantly enriched GO categories in the 374 genes in modules 16-20 that are connected to modules 13-15 in the posterior network $\mathbf{\Lambda}_{\mathbf{y}|\mathbf{x},\mathbf{z}}$ (p-value < 0.05 after Bonferroni correction). This set of genes was significantly enriched for several GO categories related to immune system processes, including cellular response to stress (p-value $= 2.90 \times 10^{-2}$ with overlap of 35 genes out of 1599 genes in the category), regulation of defense response to virus (p-value $= 4.36 \times 10^{-2}$ with overlap of 6 genes out of 71 genes in the category), and regulation of immune effector process (p-value $= 4.42 \times 10^{-2}$ with overlap of 14 genes out of 409 genes in the category).

Thus, all of the modules that influence phenotypes, modules 13-20, showed enrichments in immune-related genes, with significant enrichment for modules 13-15 and weaker but still significant enrichment for modules 16-20. Since asthma is an immune disorder, the enrichment of immune-related genes in the trait-perturbing modules provides evidence that these modules are likely to play an important role in asthma patients.

SNPs perturbing asthma phenotypes overlap with SNPs perturbing immune modules

The SNP perturbation of the gene modules above (Figure 4C) may or may not result in a change in phenotypes. To see if the SNPs perturbing each gene module have an impact on the lung and blood phenotypes, we compared the top module-specific eQTLs in $\mathbf{\Theta_{xy}}$ with the SNPs with the strongest effects on the lung or blood phenotypes in $\mathbf{B_{xz}}$ inferred from our sparse Gaussian chain graph model. The SNPs with the strongest effects on the lung (or blood) phenotypes were determined based on the sum over the SNP effects on all lung (or blood) phenotypes in $|\mathbf{B_{xz}}|$. Similarly, the top module-specific eQTLs were determined based on the sum over the SNP effect sizes on all expression levels in each module in $|\mathbf{\Theta_{xy}}|$. We obtained the overlap between the SNPs perturbing the phenotype network and the SNPs perturbing the gene network, considering the top 100 and 200 module-specific eQTLs and top 200 SNPs perturbing each phenotype group (the cutoff for top 200 SNPs shown as the magenta line at SNP effect size 0.013 for lung traits in Figure S1A and at SNP effect size 0.0037 for blood traits in Figure S1B). Using Fisher's exact test, we also assessed the significance of these overlaps within the set of SNPs with non-zero effects in $\mathbf{\Theta_{xy}}$.

In our comparison, only a subset of the eQTLs influenced phenotypes, but the eQTLs perturbing the immune modules, modules 13-20, were more likely to perturb the phenotypes than the eQTLs for the other modules (Figures 6A and 6B). Among the top 100 module-specific eQTLs, only a fraction of those SNPs overlapped with top 200 SNPs perturbing phenotypes (ranging from 0% to 18% of eQTLs across modules for an overlap with SNPs perturbing lung phenotypes and ranging from 2% to 20% of eQTLs across modules for an overlap with SNPs perturbing blood phenotypes). These fractions increased as we considered more eQTLs as in top 200 and all module-specific eQTLs. This matches with the observations from previous studies that not all of the eQTLs affect higher-level phenotypes[21] and that trait-associated SNPs are likely to be eQTLs.[40] However, in our analysis, the eQTLs for immune-related modules, modules 13-20, tended to have larger overlaps than the other modules (Figures 6A and 6B). Furthermore, we found these overlaps are statistically significant for all of the immune modules, modules 13-20, but not for all of the other modules, and the most statistically significant overlaps were from the immune modules (Figures 6C and 6D). This suggests that eQTLs that perturb the modules that influence phenotypes are more likely to perturb phenotypes than eQTLs that perturb other gene modules.

The immune modules mediate SNP perturbation of phenotypes

To understand the molecular mechanisms that underlie the SNPs perturbing phenotypes beyond the simple overlap of SNPs perturbing the phenotype network and SNPs perturbing the gene network, we used the Perturb-Net inference procedure to obtain the decomposition of the SNP effects on phenotypes $\mathbf{B_{xz}}$ into the component SNP effects on phenotypes $\mathbf{B}_{\mathbf{xz}}^{M_1}, \ldots, \mathbf{B}_{\mathbf{xz}}^{M_{20}}$ mediated by each of the 20 gene modules. We examined this decomposition for the 50 SNPs with the strongest effects on each group of lung and blood phenotypes (the cutoff for top 50 SNPs is shown as the green line at SNP effect size 0.04 for the lung phenotypes in Figure S1A and at SNP effect size 0.011 on blood phenotypes in Figure S1B).

For each set of 50 SNPs with the strongest perturbation effects on lung or blood phenotypes, nearly all of their effects on phenotypes were mediated by modules 12 through 20. The decomposition of the SNP effects on lung phenotypes (Figure 7A) into the 20 components (Figure 7B) shows that only the components for modules 12 through 20 contain non-zero SNP effects on the lung phenotypes, except for module 6, which mediates the effects of SNP rs1008932. We further summarized the component SNP effects by summing across all lung phenotypes for each SNP ($\sum_{j \in \text{Lung}} |[\mathbf{B}_{\mathbf{xz}}^{M}]_{i,j}|$ for module M and SNP i; Figure 7C). In Figure 7C, for 45 out of the 50 SNPs the SNP effect on the lung phenotypes is mediated by a single module from modules 12-20. For the other 5 SNPs, although their effects on phenotypes were mediated by two or three modules, the module with the strongest mediator effect had effect size at least 5 times as large as the other modules. Although only 20 SNPs overlapped between the two sets of top 50 SNPs for lung and blood phenotypes, the SNP effects on blood phenotypes were also mediated by modules 12-20 (Figure 8). This indicates that modules 12-20 can potentially explain the molecular mechanisms behind the SNP perturbations of asthma phenotypes.

Module 13 explains the molecular mechanism of the previously known association between SNP rs63340 and asthma susceptibility

We performed an in-depth analysis of module 13, its influence on asthma phenotypes, and its perturbation by SNP rs63340, one of the SNPs with the strongest effects on this module and also on phenotypes (Figure 9). SNP rs63340 ranked third for its effect on module 13 and 41st for its effect on phenotypes. The genome region 16q21, where SNP rs63340 is located, has been previously found to be linked to asthma and atopy in several previous genome-wide screenings,[41,42,43,44,45] though the mechanism behind this association has not been fully elucidated. Our model indicated that this locus directly perturbs the expression levels of *NRP1, DCANP1, EPHB1, NLRP7* and *GZMB*. Several of these genes have been previously linked to asthma. *NRP1* is known to be a part of one of important mediators involved in the pathogenesis of asthma.[46] A promoter nucleotide variant in *DCANP1* was previously associated with serum IgE levels among asthmatics.[47] *EPHB1* has been previously linked to lung function traits in asthma.[48] Our model found *EPHA2, KCNA5, NRP1*, and *CLEC4C* as key mediator genes, determined by the row of $\mathbf{B}_{\mathbf{xz}}^{m}$ for SNP rs63340 and for gene m in module 13 summed across all phenotypes, that mediate the effects of SNP rs63340 on asthma phenotypes. Among these genes, *KCNA5* has been known to be connected to pulmonary vasoconstriction[49,50] and a SNP near *KCNA5* was significantly associated with asthma.[51] In addition, *CLEC4C* has been known to be involved in immune response.[52,53] Thus, the results from Perturb-Net are well supported by the previous findings in the literature and provide insights into the gene network underlying the previously reported association between the locus and asthma phenotypes.

Comparison with other methods

We compared our method with the two-layer Lasso both qualitatively by visual inspection of the estimated parameters and quantitatively by assessing the predictive power of different methods.

Comparison of the estimated models We compared the results from our approach and the two-layer Lasso by visually inspecting the estimated SNP effects on gene modules and the estimated gene module effects on phenotypes. For the top 50 SNPs perturbing the lung phenotypes (Figure 7), we examined the overall SNP effects on each gene module based on $\mathbf{\Theta_{xy}}$ and $\mathbf{B_{xy}}$ from our model and $\mathbf{A_{xy}}$ from the two-layer Lasso ($\sum_{j \in M} |[\mathbf{\Theta_{xy}}]_{i,j}|$, $\sum_{j \in M} |[\mathbf{B_{xy}}]_{i,j}|$, and $\sum_{j \in M} |[\mathbf{A_{xy}}]_{i,j}|$ for SNP i and module M). To see how each gene module influences phenotypes, we computed the magnitudes of overall gene module effects on each phenotype from $\mathbf{\Theta_{yz}}$ and $\mathbf{B_{yz}}$ in our model and $\mathbf{A_{yz}}$ in the two-layer Lasso ($\sum_{j \in M} |[\mathbf{\Theta_{yz}}]_{j,k}|$, $\sum_{j \in M} |[\mathbf{B_{yz}}]_{j,k}|$, and $\sum_{j \in M} |[\mathbf{A_{yz}}]_{j,k}|$ for module M and phenotype k).

Unlike the Perturb-Net model, the two-layer Lasso does not model direct and indirect perturbation effects separately but attempts to capture both types of effects in a single set of parameters. Thus, the perturbation effects captured by the two-layer Lasso appeared to be a compromise between the direct and indirect perturbation effects captured by Perturb-Net (Figure 10). However, the SNP effects appeared to be similar across $\mathbf{\Theta_{xy}}$, $\mathbf{B_{xy}}$, and $\mathbf{A_{xy}}$ in the module-level summaries (Figures 10A-10C), because the direct SNP perturbation effects tended to propagate to other genes only within each module, but not to genes in other modules. On the other hand, the module effects on phenotypes showed a distinct pattern across $\mathbf{\Theta_{yz}}$, $\mathbf{B_{yz}}$, and $\mathbf{A_{yz}}$ (Figures 10D-10F), because in our model, the direct influence of gene expression levels on a phenotype induces the indirect influence on other correlated phenotypes, whereas the Lasso parameter tries to capture both types of information in a single parameter.

Comparison of prediction accuracy We assess the ability to make predictions about new asthma patients based on the estimated Perturb-Net model and compare the results with those from the two-layer Lasso. We split the data into train and test sets and obtained the prediction accuracy using the test set after training a model on the train set. We set aside 25 samples as a test set and used the remaining 115 fully-observed samples and 34 partially observed samples to train a sparse Gaussian chain graph model with our semi-supervised learning method. We also trained a model, using only the 115 fully observed samples with the supervised learning method, and compared the results from the two-layer Lasso, also trained from the fully observed samples. Given the estimated models, we performed prediction tasks and obtained the prediction error as the squared difference between the observed and predicted values averaged across samples in test set.

The Perturb-Net model estimated from all data had the smallest prediction error for all of the prediction tasks (Table 2). In particular, our model with semi-supervised learning performed better than our model with supervised learning, demonstrating that leveraging partially observed data can help learn a model with greater predictive power. For supervised learning, our model outperformed Lasso. This demonstrates that taking into account the network structure in expression levels and clinical phenotypes increases the performance on prediction tasks.

Discussion

We introduced a statistical framework called Perturb-Net for learning a gene network underlying phenotypes using SNP perturbations and for identifying SNPs that perturb this network, given population genotype, expression, and phenotype data. Compared to many of the previous methods that focused on the co-localization of eQTLs and genetic association signals for phenotypes,[13,14,15] using multi-stage methods,[10,11,12] our approach combines all available data in a single statistical analysis and directly models the multiple layers of a biological system with a cascade of influence from SNPs to expression levels to phenotypes, while modeling each layer as a network. Our probabilistic graphical model framework allows to model eQTLs with or without an impact on phenotypes for an investigation of co-localization of SNPs perturbing expression levels and SNPs perturbing phenotypes and to extract rich information on the molecular mechanisms that explains the influence of SNPs on phenotypes. We developed fast learning algorithms called Fast-sCGGM and Mega-sCGGM for learning sCGGM components of the Perturb-Net model, which serve as the key subroutine of our Perturb-Net learning method, to enable analysis of human genome scale data within a few hours.

Our results from applying Perturb-Net to asthma data confirmed the observations from the previous studies, including GWAS, eQTL mapping, and gene network modeling.[11,40] Our results confirmed the finding from previous studies on combining the results of GWAS and eQTL mapping[40] that there is a partial overlap between SNPs perturbing expression levels and SNPs perturbing phenotypes. In addition, this overlap was more significant for eQTLs that perturb trait-associated modules than eQTLs that perturb other parts of the gene network, as was previously reported.[21]

The analysis of the asthma data with Perturb-Net provided new insights. Perturb-Net was able to systematically reveal the gene network that lies between the SNPs and phenotypes and to uncover how different parts of this gene network modulate the SNP effects on phenotypes in a statistically principled manner. Often, there are genetic loci that have been previously known to be linked to the disease susceptibility, though little is known about the underlying molecular mechanism. In such cases, the Perturb-Net analysis of asthma data demonstrated the potential to reveal the molecular pathway that are perturbed by previously known trait-associated loci.

Perturb-Net provides a flexible tool that can be extended in several different ways in a straightforward manner. Because the sparse Gaussian chain graph model in Perturb-Net uses sCGGMs as building blocks, the sCGGM component models can be threaded in different ways to form sparse Gaussian chain graph models with different structures. For example, if expression data from multiple tissue types are available for a patient cohort along with genome sequence and phenotype data, a sparse Gaussian chain graph model can be set up with multiple component sCGGMs, each corresponding to the gene network under SNP perturbation in each tissue type, linked to another sCGGM for modeling expression levels influencing phenotypes. Models like this can reveal SNPs that perturb phenotypes through different tissue types and through different modules in each tissue-specific gene network. Another possible extension is to thread more than two component sCGGMs within a sparse Gaussian chain graph model to model more than two layers in a biological system, including epigenomes, metabolomes, and proteomes.

Appendix A: Fast-sCGGM and Mega-sCGGM for efficiently learning sCGGMs

We introduce our scalable learning algorithms for sCGGM, since the learning algorithm for sparse Gaussian chain graph models in Eq. (1) uses the sCGGM learning algorithm as a key module. Assume an sCGGM[7,8] for gene expression levels $\mathbf{y} \in \mathbb{R}^q$ for q genes and genotype data $\mathbf{x} \in \{0, 1, 2\}^p$

for minor allele frequencies at p loci be given as follows:

$$p(\mathbf{y}|\mathbf{x}) = \exp(-\mathbf{y}^T\mathbf{\Lambda}\mathbf{y} - 2\mathbf{x}^T\mathbf{\Theta}\mathbf{y})/Z(\mathbf{x}), \tag{7}$$

where $\mathbf{\Lambda}$ is a $q \times q$ matrix representing a gene network, $\mathbf{\Theta}$ is a $p \times q$ matrix modeling SNPs influencing the expression levels of genes in the network, and $Z(\mathbf{x}) = (2\pi)^{q/2}|\mathbf{\Lambda}|^{-1}\exp(\mathbf{x}^T\mathbf{\Theta}\mathbf{\Lambda}^{-1}\mathbf{\Theta}^T\mathbf{x})$ is the constant to ensure the probability distribution integrates to 1. Then, given genotype data $\mathbf{X} \in \mathbb{R}^{n\times p}$ for n samples and p SNPs, each element taking a value from $\{0, 1, 2\}$ for the number of minor alleles at the locus, and expression data $\mathbf{Y} \in \mathbb{R}^{n\times q}$ for q genes for the same samples, a parameter estimate of the sCGGM in Eq. (7) can be obtained by minimizing L_1-regularized negative log-likelihood:

$$\min_{\mathbf{\Lambda}\succ 0,\mathbf{\Theta}} \quad f(\mathbf{\Lambda}, \mathbf{\Theta}) = g(\mathbf{\Lambda}, \mathbf{\Theta}) + h(\mathbf{\Lambda}, \mathbf{\Theta}), \tag{8}$$

where $g(\mathbf{\Lambda}, \mathbf{\Theta}){=}{-}\log|\mathbf{\Lambda}|{+}\mathrm{tr}(\mathbf{S_{yy}}\mathbf{\Lambda}{+}2\mathbf{S_{xy}}^T\mathbf{\Theta}{+}\mathbf{\Lambda}^{-1}\mathbf{\Theta}^T\mathbf{S_{xx}}\mathbf{\Theta})$ is the smooth negative log-likelihood, given data covariance matrices $\mathbf{S_{xx}} = \frac{1}{n}\mathbf{X}^T\mathbf{X}, \mathbf{S_{xy}} = \frac{1}{n}\mathbf{X}^T\mathbf{Y}, \mathbf{S_{yy}} = \frac{1}{n}\mathbf{Y}^T\mathbf{Y}$, and $h(\mathbf{\Lambda}, \mathbf{\Theta}){=}\lambda_{\mathbf{\Lambda}}\|\mathbf{\Lambda}\|_1 + \lambda_{\mathbf{\Theta}}\|\mathbf{\Theta}\|_1$ for the non-smooth elementwise L_1 penalty. $\lambda_{\mathbf{\Lambda}}, \lambda_{\mathbf{\Theta}} > 0$ are regularization parameters.

Below, we introduce Fast-sCGGM for learning an sCGGM that substantially reduces computation time by orders of magnitude compared to the previous state-of-the-art method.[22] Then, we describe Mega-sCGGM, a modification of Fast-sCGGM, that performs block-wise computation to learn a model from large human genome-wide data on a machine with limited memory.

Fast-sCGGM for improving computation time

Fast-sCGGM uses an alternate Newton coordinate descent method that alternately updates $\mathbf{\Lambda}$ and $\mathbf{\Theta}$, optimizing Eq. (8) over $\mathbf{\Lambda}$ given $\mathbf{\Theta}$ and vice versa until convergence. Our approach is based on the key observation that with $\mathbf{\Lambda}$ fixed, the problem of solving Eq. (8) over $\mathbf{\Theta}$ becomes simply the well-known Lasso optimization, which can be solved efficiently using a coordinate descent method.[54]

On the other hand, optimizing Eq. (8) for $\mathbf{\Lambda}$ given $\mathbf{\Theta}$ requires forming a quadratic approximation to find a generalized Newton direction and performing line search to find the step size. However, this computation is significantly simpler than performing the same type of computation on both $\mathbf{\Lambda}$ and $\mathbf{\Theta}$ jointly as in the previous approach.[22] Our algorithm iterates between the following two steps until convergence:

- **Coordinate descent optimization for $\mathbf{\Theta}$ given $\mathbf{\Lambda}$:** With $\mathbf{\Lambda}$ fixed, the optimization problem in Eq. (8) becomes

$$\underset{\mathbf{\Theta}}{\operatorname{argmin}}\ g_{\mathbf{\Lambda}}(\mathbf{\Theta}) + \lambda_{\mathbf{\Theta}}\|\mathbf{\Theta}\|_1, \tag{9}$$

where $g_{\mathbf{\Lambda}}(\mathbf{\Theta}) = \operatorname{tr}(2{\mathbf{S_{xy}}}^T\mathbf{\Theta} + \mathbf{\Lambda}^{-1}\mathbf{\Theta}^T\mathbf{S_{xx}}\mathbf{\Theta})$. Since $g_{\mathbf{\Lambda}}(\mathbf{\Theta})$ is a quadratic function, Eq. (9) corresponds to the Lasso problem and the coordinate descent method can be used to solve this efficiently.

- **Coordinate descent optimization for $\mathbf{\Lambda}$ given $\mathbf{\Theta}$:** Given fixed $\mathbf{\Theta}$, the problem in Eq. (8) becomes

$$\underset{\mathbf{\Lambda}\succ 0}{\operatorname{argmin}}\, g_{\mathbf{\Theta}}(\mathbf{\Lambda}) + \lambda_{\mathbf{\Lambda}}\|\mathbf{\Lambda}\|_1, \tag{10}$$

where $g_{\mathbf{\Theta}}(\mathbf{\Lambda}) = -\log|\mathbf{\Lambda}| + \operatorname{tr}(\mathbf{S_{yy}}\mathbf{\Lambda} + \mathbf{\Lambda}^{-1}\mathbf{\Theta}^T\mathbf{S_{xx}}\mathbf{\Theta})$. In order to solve this, we first find a generalized Newton direction that minimizes the L_1-regularized quadratic approximation $\bar{g}_{\mathbf{\Lambda},\mathbf{\Theta}}(\Delta_{\mathbf{\Lambda}})$ of $g_{\mathbf{\Theta}}(\mathbf{\Lambda})$:

$$\mathbf{D}_{\mathbf{\Lambda}} = \underset{\Delta_{\mathbf{\Lambda}}}{\operatorname{argmin}}\, \bar{g}_{\mathbf{\Lambda},\mathbf{\Theta}}(\Delta_{\mathbf{\Lambda}}) + \lambda_{\mathbf{\Lambda}}\|\mathbf{\Lambda} + \Delta_{\mathbf{\Lambda}}\|_1, \tag{11}$$

where $\bar{g}_{\mathbf{\Lambda},\mathbf{\Theta}}(\Delta_{\mathbf{\Lambda}})$ is obtained from a second-order Taylor expansion and is given as

$$\bar{g}_{\mathbf{\Lambda},\mathbf{\Theta}}(\Delta_{\mathbf{\Lambda}}) = \operatorname{vec}(\nabla_{\mathbf{\Lambda}} g(\mathbf{\Lambda},\mathbf{\Theta}))^T \operatorname{vec}(\Delta_{\mathbf{\Lambda}}) + \frac{1}{2}\operatorname{vec}(\Delta_{\mathbf{\Lambda}})^T \nabla^2_{\mathbf{\Lambda}} g(\mathbf{\Lambda},\mathbf{\Theta}) \operatorname{vec}(\Delta_{\mathbf{\Lambda}}).$$

In the above equation, $\nabla_{\mathbf{\Lambda}} g(\mathbf{\Lambda}, \mathbf{\Theta}) = \mathbf{S}_{\mathbf{yy}} - \mathbf{\Sigma} - \mathbf{\Psi}$ and $\nabla^2_{\mathbf{\Lambda}} g(\mathbf{\Lambda}, \mathbf{\Theta}) = \mathbf{\Sigma} \otimes (\mathbf{\Sigma} + 2\mathbf{\Psi})$, where $\mathbf{\Sigma} = \mathbf{\Lambda}^{-1}$ and $\mathbf{\Psi} = \mathbf{\Sigma}\mathbf{\Theta}^T \mathbf{S}_{\mathbf{xx}} \mathbf{\Theta}\mathbf{\Sigma}$, are the components of the gradient and Hessian matrices corresponding to $\mathbf{\Lambda}$. The problem in Eq. (11) is again equivalent to the Lasso problem, which can be solved efficiently via coordinate descent. Given the Newton direction for $\mathbf{\Lambda}$, we update $\mathbf{\Lambda} \leftarrow \mathbf{\Lambda} + \alpha \mathbf{D}_{\mathbf{\Lambda}}$, where step size $0 < \alpha \leq 1$ ensures sufficient decrease in Eq. (8) and positive definiteness of $\mathbf{\Lambda}$. The α is obtained by line search on the objective in Eq. (10).

In order to further reduce computation time, we adopt the following strategies that have been previously used for sparse Gaussian graphical model and sCGGM optimizations.[22,55] First, to improve the efficiency of coordinate descent for the Lasso problem in Eqs. (9) and (11), we restrict the updates to an active set of variables given as:

$$\begin{aligned} \mathcal{S}_{\mathbf{\Lambda}} &= \{(\Delta_{\mathbf{\Lambda}})_{ij} : |(\nabla_{\mathbf{\Lambda}} g(\mathbf{\Lambda}, \mathbf{\Theta}))_{ij}| > \lambda_{\mathbf{\Lambda}} \vee \mathbf{\Lambda}_{ij} \neq 0\} \\ \mathcal{S}_{\mathbf{\Theta}} &= \{(\Delta_{\mathbf{\Theta}})_{ij} : |(\nabla_{\mathbf{\Theta}} g(\mathbf{\Lambda}, \mathbf{\Theta}))_{ij}| > \lambda_{\mathbf{\Theta}} \vee \mathbf{\Theta}_{ij} \neq 0\}. \end{aligned}$$

Because the active set sizes $m_{\mathbf{\Lambda}} = |\mathcal{S}_{\mathbf{\Lambda}}|, m_{\mathbf{\Theta}} = |\mathcal{S}_{\mathbf{\Theta}}|$ approach the number of non-zero entries in the sparse solutions for $\mathbf{\Lambda}$ and $\mathbf{\Theta}$ over iterations, this strategy yields a substantial speedup. Second, to further improve the efficiency of coordinate descent, we store intermediate results for the large matrix products that need to be computed repeatedly. We compute and store $\mathbf{U} := \Delta_{\mathbf{\Lambda}}\Sigma$ and $\mathbf{V} := \Delta_{\mathbf{\Theta}}\Sigma$ at the beginning of the optimization. Then, after a coordinate descent update to $(\Delta_{\mathbf{\Lambda}})_{ij}$, row i and j of $\mathbf{U}$ are updated. Similarly, after an update to $(\Delta_{\mathbf{\Theta}})_{ij}$, row i of $\mathbf{V}$ is updated. Finally, in each iteration of Fast-sCGGM, we warm-start $\mathbf{\Lambda}$ and $\mathbf{\Theta}$ from the results of the previous iteration and make a single pass over the active set. This ensures decrease in the objective in Eq. (8), while reducing the overall computation time in practice. The pseudocode for Fast-sCGGM is provided in Algorithm 1.

Mega-sCGGM for removing memory requirement

Fast-sCGGM as described above is still limited by the space required to store large matrices during coordinate descent computation. Solving Eq. (11) for updating $\mathbf{\Lambda}$ requires precomputing and storing

Algorithm 1: Fast-sCGGM

```
input  : Inputs $\mathbf{X} \in \mathbb{R}^{n \times p}$ and $\mathbf{Y} \in \mathbb{R}^{n \times q}$; regularization parameters $\lambda_{\mathbf{\Lambda}}, \lambda_{\mathbf{\Theta}}$
output : Parameters $\mathbf{\Lambda}, \mathbf{\Theta}$
Initialize $\mathbf{\Theta} \leftarrow 0, \mathbf{\Lambda} \leftarrow I_q$
for $t = 0, 1, \ldots$ do
    Determine active sets $\mathcal{S}_{\mathbf{\Lambda}}, \mathcal{S}_{\mathbf{\Theta}}$
    Solve via coordinate descent: $D_{\mathbf{\Lambda}} = \arg\min_{\Delta_{\mathbf{\Lambda}}, \Delta_{\overline{\mathcal{S}_{\mathbf{\Lambda}}}}=0} \bar{g}_{\mathbf{\Lambda},\mathbf{\Theta}}(\mathbf{\Lambda} + \Delta_{\mathbf{\Lambda}}, \mathbf{\Theta}) + h(\mathbf{\Lambda} + \Delta_{\mathbf{\Lambda}}, \mathbf{\Theta})$
    Update $\mathbf{\Lambda} = \mathbf{\Lambda} + \alpha D_{\mathbf{\Lambda}}$, where step size $\alpha$ is found with line search
    Solve via coordinate descent: $\mathbf{\Theta} = \mathrm{argmin}_{\mathbf{\Theta}_{\mathcal{S}_{\mathbf{\Theta}}}} \; g_{\mathbf{\Lambda}}(\mathbf{\Theta}) + \lambda_{\mathbf{\Theta}} \|\mathbf{\Theta}\|_1$
```

$q \times q$ matrices, $\mathbf{\Sigma}$ and $\mathbf{\Psi} = \mathbf{\Sigma}\mathbf{\Theta}^T\mathbf{S}_{\mathbf{xx}}\mathbf{\Theta}\mathbf{\Sigma}$, whereas solving Eq. (9) for updating $\mathbf{\Theta}$ requires $\mathbf{\Sigma}$ and a $p \times p$ matrix $\mathbf{S}_{\mathbf{xx}}$. A naive approach to reduce the memory footprint would be to recompute portions of these matrices on demand for each coordinate update, which would be very expensive.

Here, we describe Mega-sCGGM that combines the alternating Newton coordinate descent algorithm in Fast-sCGGM with block coordinate descent to scale up the optimization to very large problems on a machine with limited memory. During coordinate descent optimization, we update blocks of $\mathbf{\Lambda}$ and $\mathbf{\Theta}$ so that within each block, the computation of the large matrices can be cached and re-used, where these blocks are determined automatically by exploiting the sparse stucture. For $\mathbf{\Lambda}$, we extend the block coordinate descent approach in BigQUIC[56] developed for sparse Gaussian graphical models to take into account the conditioning variables in CGGMs. For $\mathbf{\Theta}$, we describe a new approach for block coordinate descent update. Our algorithm can, in principle, be applied to problems of any size on a machine with limited memory and converges to the same optimal solution as our alternating Newton coordinate descent method.

Blockwise optimization for $\mathbf{\Lambda}$ A coordinate-descent update of $[\Delta_{\mathbf{\Lambda}}]_{i,j}$ requires the ith and jth columns of $\mathbf{\Sigma}$ and $\mathbf{\Psi}$. If these columns are in memory, they can be re-used. Otherwise, it is a cache miss and we should compute them on demand as follows. We obtain $[\mathbf{\Sigma}]_{:,i}$ by solving linear system $\mathbf{\Lambda}[\mathbf{\Sigma}]_{:,i} = \mathbf{e}_i$, where $\mathbf{e}_i$ is a vector of q 0's except for 1 in the ith element, with conjugate gradient method. Then, $[\mathbf{\Psi}]_{:,i}$ can be obtained from $\mathbf{R}^T[\mathbf{R}]_{:,i}$, where $\mathbf{R} = \mathbf{X}\mathbf{\Theta}\mathbf{\Sigma}$.

In order to reduce cache misses, we perform block coordinate descent, where within each block, the columns of $\mathbf{\Sigma}$ are cached and re-used. Suppose we partition $\mathcal{N} = \{1, \ldots, q\}$ into $k_{\mathbf{\Lambda}}$ blocks,

Algorithm 2: Mega-sCGGM

```
input  : $\mathbf{X} \in \mathbb{R}^{n\times p}$ and $\mathbf{Y} \in \mathbb{R}^{n\times q}$; regularization parameters $\lambda_{\mathbf{\Lambda}}, \lambda_{\mathbf{\Theta}}$
output : Parameters $\mathbf{\Lambda}, \mathbf{\Theta}$
Initialize $\mathbf{\Theta} \leftarrow 0, \mathbf{\Lambda} \leftarrow I_q$
for $t = 0, 1, \dots$ do
    Determine active sets $\mathcal{S}_{\mathbf{\Lambda}}, \mathcal{S}_{\mathbf{\Theta}}$
    Partition columns of $\mathbf{\Lambda}$ into $k_{\mathbf{\Lambda}}$ blocks                ▷ Minimize over $\mathbf{\Lambda}$
    Initialize $\Delta_{\mathbf{\Lambda}} \leftarrow 0$
    for $z = 1$ to $k_{\mathbf{\Lambda}}$ do
        Compute $[\mathbf{\Sigma}]_{:,C_z}$, $[\mathbf{U}]_{:,C_z}$, and $[\mathbf{\Psi}]_{:,C_z}$
        for $r = 1$ to $k_{\mathbf{\Lambda}}$ do
            if $z \neq r$ then
                Identify columns $B_{zr} \subset C_r$ with active elements in $\mathbf{\Lambda}$
                Compute $[\mathbf{\Sigma}]_{:,B_{zr}}$, $[\mathbf{U}]_{:,B_{zr}}$, and $[\mathbf{\Psi}]_{:,B_{zr}}$
            Update all active $[\Delta_{\mathbf{\Lambda}}]_{i,j}$ in $(C_z, C_r)$
    Update $\mathbf{\Lambda} \leftarrow \mathbf{\Lambda} + \alpha\Delta_{\mathbf{\Lambda}}$, where step size $\alpha$ is found by line search.
    Partition columns of $\mathbf{\Theta}$ into $k_{\mathbf{\Theta}}$ blocks                ▷ Minimize over $\mathbf{\Theta}$
    for $r = 1$ to $k_{\mathbf{\Theta}}$ do
        Compute $[\mathbf{\Sigma}]_{:,C_r}$, and initialize $\mathbf{V} \leftarrow \mathbf{\Theta}[\mathbf{\Sigma}]_{:,C_r}$
        for row $i \in \{1, \dots, p\}$ if $I_\phi(\mathcal{S}_{(i,C_r)})$ do
            Compute $[\mathbf{S_{xx}}]_{i,j}$ for non-empty columns $j$ in $V_{C_r}$
            Update all active $[\mathbf{\Theta}]_{i,j}$ in $(i, C_r)$
```

$C_1, \dots, C_{k_{\mathbf{\Lambda}}}$. We apply this partitioning to the rows and columns of $\Delta_{\mathbf{\Lambda}}$ to obtain $k_{\mathbf{\Lambda}} \times k_{\mathbf{\Lambda}}$ blocks. We perform coordinate-descent updates in each block, updating all elements in the active set within that block. Let $[\mathbf{A}]_{:,C_r}$ denote a matrix containing columns of $\mathbf{A}$ that correspond to the subset C_r. In order to perform coordinate-descent updates on (C_r, C_z) block of $\Delta_{\mathbf{\Lambda}}$, we need $[\mathbf{\Sigma}]_{:,C_r}$, $[\mathbf{\Sigma}]_{:,C_z}$, $[\mathbf{\Psi}]_{:,C_r}$, and $[\mathbf{\Psi}]_{:,C_r}$. Thus, we pick the smallest possible $k_{\mathbf{\Lambda}}$ such that we can store $2q/k_{\mathbf{\Lambda}}$ columns of $\mathbf{\Sigma}$ and $\mathbf{\Psi}$ in memory. When updating the variables within block (C_z, C_r) of $\Delta_{\mathbf{\Lambda}}$, there are no cache misses once $[\mathbf{\Sigma}]_{:,C_z}$, $[\mathbf{\Sigma}]_{:,C_z}$, $[\mathbf{\Psi}]_{:,C_z}$, and $[\mathbf{\Psi}]_{:,C_r}$ are computed and stored. After updating each $[\Delta_{\mathbf{\Lambda}}]_{i,j}$ to $[\Delta_{\mathbf{\Lambda}}]_{i,j} + \mu$, we maintain $[\mathbf{U}]_{:,C_z}$ and $[\mathbf{U}]_{:,C_r}$ by

$[\mathbf{U}]_{i,t} \leftarrow [\mathbf{U}]_{i,t} + \mu[\mathbf{\Sigma}]_{j,t}, [\mathbf{U}]_{j,t} \leftarrow [\mathbf{U}]_{j,t} + \mu[\mathbf{\Sigma}]_{i,t}, \forall t \in \{C_z \cup C_r\}$.

To go through all blocks, we update blocks $(C_z, C_1), \dots, (C_z, C_k)$ for each $z \in \{1, \dots, k_{\mathbf{\Lambda}}\}$. Since all of these blocks share $[\mathbf{\Sigma}]_{:,C_z}$ and $[\mathbf{\Psi}]_{:,C_z}$, we precompute and store them in memory. When updating an off-diagonal block $(C_z, C_r), z \neq r$, we compute $[\mathbf{\Sigma}]_{:,C_r}$ and $[\mathbf{\Psi}]_{:,C_r}$. Overall, each block of $\mathbf{\Sigma}$ and $\mathbf{\Psi}$ will be computed $k_{\mathbf{\Lambda}}$ times.

In typical real-world problems, the graph structure of $\mathbf{\Lambda}$ will exhibit clustering, with an approximately block diagonal structure. We exploit this structure by choosing a partition $\{C_1, \ldots, C_{k_{\mathbf{\Lambda}}}\}$ that reduces cache misses. Within diagonal blocks (C_r, C_r)'s, once $[\mathbf{\Sigma}]_{:,C_r}$ and $[\mathbf{\Psi}]_{:,C_r}$ are computed, there are no cache misses. For off-diagonal blocks (C_r, C_z)'s, $r \neq z$, we have a cache miss only if some variable in $\{[\Delta]_{i,j} | i \in C_r, j \in C_z\}$ lies in the active set. We minimize the active set in off-diagonal blocks via clustering, following the strategy for sparse Gaussian graphical model estimation in BigQUIC[56] and using the METIS[37] graph clustering library.

Although the worst-case scenario is to compute $\mathbf{\Sigma}$ and $\mathbf{\Psi}$ $k_{\mathbf{\Lambda}}$ times to update all elements of $\Delta_{\mathbf{\Lambda}}$, in practice, graph clustering dramatically reduces this additional cost of block-wise optimization. In the best case, if the active set for $\mathbf{\Lambda}$ is perfectly block-diagonal and graph clustering identifies this block diagonal structure, we need to compute $\mathbf{\Sigma}$ and $\mathbf{\Psi}$ only once to update all the blocks. A depiction of our blockwise optimization scheme is given in Figure S2.

Blockwise Optimization for $\mathbf{\Theta}$ The coordinate descent update of $[\mathbf{\Theta}]_{i,j}$ requires $[\mathbf{S_{xx}}]_{:,i}$ and $[\mathbf{\Sigma}]_{:,j}$ to compute $[\mathbf{S_{xx}}]_{:,i}^T[\mathbf{V}]_{:,j}$, where $[\mathbf{V}]_{:,j} = \mathbf{\Theta}[\mathbf{\Sigma}]_{:,j}$. If $[\mathbf{S_{xx}}]_{:,i}$ and $[\mathbf{\Sigma}]_{:,j}$ are not already in the memory, it is a cache miss. Computing $[\mathbf{S_{xx}}]_{:,i}$ takes $O(np)$, which is expensive if we have many cache misses.

We propose a block coordinate descent approach for solving Eq. (9) that groups these computations to reduce cache misses. Given a partition $\{1, \ldots, q\}$ into $k_{\mathbf{\Theta}}$ subsets, $C_1, \ldots, C_{k_{\mathbf{\Theta}}}$, we divide $\mathbf{\Theta}$ into $p \times k_{\mathbf{\Theta}}$ blocks, where each block comprises a portion of a row of $\mathbf{\Theta}$. We denote each block (i, C_r), where $i \in \{1, \ldots, p\}$. Since updating block (i, C_r) requires $[\mathbf{S_{xx}}]_{:,i}$ and $[\mathbf{\Sigma}]_{:,C_r}$, we pick smallest possible $k_{\mathbf{\Theta}}$ such that we can store $q/k_{\mathbf{\Theta}}$ columns of $\mathbf{\Sigma}$ in memory. While performing coordinate descent updates within block (i, C_r) of $\mathbf{\Theta}$, there are no cache misses, once $[\mathbf{S_{xx}}]_{:,i}$ and $[\mathbf{\Sigma}]_{:,C_r}$ are in memory. After updating each $[\mathbf{\Theta}]_{i,j}$ to $[\mathbf{\Theta}]_{i,j} + \mu$, we update $[\mathbf{V}]_{:,C_r}$ by $[\mathbf{V}]_{i,t} \leftarrow [\mathbf{V}]_{i,t} + \mu[\mathbf{\Sigma}]_{j,t}, \forall t \in C_r$.

In order to sweep through all blocks, each time we select a $q \in \{1, \ldots, k_{\mathbf{\Theta}}\}$ and update blocks $(1, C_r), \ldots, (p, C_r)$. Since all of these p blocks with the same C_r share the computation of $[\mathbf{\Sigma}]_{:,C_r}$, we compute and store $[\mathbf{\Sigma}]_{:,C_r}$ in memory. Within each block, the computation of $[\mathbf{S_{xx}}]_{:,i}$ is shared, so we precompute and store it in memory, before updating this block. The full matrix of $\mathbf{\Sigma}$ will be computed once while sweeping through the full $\mathbf{\Theta}$, whereas $\mathbf{S_{xx}}$ will be computed $k_{\mathbf{\Theta}}$ times.

We further reduce cache misses for $[\mathbf{S_{xx}}]_{:,i}$ by strategically selecting partition $C_1, \ldots, C_{k_{\mathbf{\Theta}}}$, based on the observation that if the active set is empty in block (i, C_r), we can skip this block and forgo computing $[\mathbf{S_{xx}}]_{:,i}$. We therefore choose a partition where the active set variables are clustered into as few blocks as possible. Formally, we want to minimize $\sum_{i,q} |I_\phi(\mathcal{S}_{(i,C_r)})|$, where $I_\phi(\mathcal{S}_{(i,C_r)})$ is an indicator function that outputs 1 if the active set $\mathcal{S}_{(i,C_r)}$ within block (i, C_r) is not empty and 0 otherwise. We therefore perform graph clustering over the graph $G = (V, E)$ defined from the active set in $\mathbf{\Theta}$, where $V = \{1, \ldots, q\}$ with one node for each column of $\mathbf{\Theta}$, and $E = \{(j,k) | [\mathbf{\Theta}]_{i,j} \in \mathcal{S}_{\mathbf{\Theta}}, [\mathbf{\Theta}]_{i,k} \in \mathcal{S}_{\mathbf{\Theta}} \text{ for } i = 1, \ldots, p\}$, connecting two nodes j and k with an edge if both $[\mathbf{\Theta}]_{i,j}$ and $[\mathbf{\Theta}]_{i,k}$ are in the active set. This edge set corresponds to the non-zero elements of $\mathbf{\Theta}^T\mathbf{\Theta}$, so the graph can be computed quickly in $O(m_{\mathbf{\Theta}} q)$.

We also exploit row-wise sparsity in $\mathbf{\Theta}$ to reduce the cost of each cache miss. Every empty row in $\mathbf{\Theta}$ corresponds to an empty row in $\mathbf{V} = \mathbf{\Theta\Sigma}$. Because we only need elements in $[\mathbf{S_{xx}}]_{:,i}$ for the dot product $[\mathbf{S_{xx}}]_{:,i}^T[\mathbf{V}]_{:,j}$, we skip computing the kth element of $[\mathbf{S_{xx}}]_{:,i}$ if the kth row of $\mathbf{\Theta}$ is all zeros. Our blockwise optimization scheme for $\mathbf{\Theta}$ is depicted in Figure S3.

Parallelization in Fast-sCGGM and Mega-sCGGM

We parallelize some of the expensive computations in Fast-sCGGM and Mega-sCGGM on multi-core machines. For both methods, we parallelize all matrix-matrix and matrix-vector multiplications. In addition, we parallelize the computation of columns of $\mathbf{\Sigma}$ and $\mathbf{\Psi}$ in Fast-sCGGM and the computation of multiple columns of $\mathbf{\Sigma}$ and $\mathbf{\Psi}$ within each block in Mega-sCGGM. In Mega-sCGGM, we parallelize the computation of each row of $\mathbf{S_{xx}}$ whenever it is recomputed.

Appendix B: Efficient implementation of EM algorithm for semi-supervised learning for Perturb-Net model

The standard EM algorithm iterates between an M-step for finding the parameter estimate maximizing the expected log-likelihood (or minimizing the negative log-likelihood) and an E-step for finding the expected sufficient statistics based on the posterior probability distribution in Eq. (5). The M-step is carried out by using our Mega-sCGGM algorithm. In E-step, a naive inversion of $\mathbf{\Lambda_y} + \mathbf{\Theta_{yz}} \mathbf{\Lambda_z}^{-1} \mathbf{\Theta}_{\mathbf{yz}}^T$ to obtain $\mathbf{\Sigma_{y|x,z}}$ is expensive and storage of this dense matrix may exceed computer memory for large gene expression datasets. We reduce the time cost and avoid memory limit in E-step, assuming that the number of phenotypes r is relatively small compared to the number of genes (i.e., $r << q$), which is typical for most studies. Instead of explicitly performing the E-step, we embed the E-step within the M-step, such that the E-step results are represented implicitly to fit in memory and computed explicitly on-demand as needed in the M-step. Specifically, instead of performing the full E-step, we implicitly represent $\mathbf{\Lambda_y} + \mathbf{\Theta_{yz}} \mathbf{\Lambda_z}^{-1} \mathbf{\Theta}_{\mathbf{yz}}^T$ as $\mathbf{\Lambda_y} + \mathbf{K}\mathbf{K}^T$, using low-rank component $\mathbf{K} = \mathbf{\Theta_{yz}} \mathbf{L}_{\mathbf{z}}^T$ and the sparse Cholesky factorization of trait network $\mathbf{L_z}\mathbf{L}_{\mathbf{z}}^T = \mathbf{\Lambda_{zz}}$. Then, during M-step, we invert $\mathbf{\Lambda_y} + \mathbf{K}\mathbf{K}^T$, one column at a time as needed, using the conjugate gradient method. This modified EM algorithm is equivalent to the original EM algorithm that iterates between an M-step and an E-step, producing the same estimate.

Supplemental Data

Supplemental Data include 3 figures and 1 table.

Acknowledgements

SYK, CM were supported by NSF CAREER Award No. MCB-1149885 (nsf.gov) and PA CURE (www.health.pa.gov). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Declaration of Interests

The authors declare no competing interests.

Web Resources

Perturb-Net software will be available at `https://github.com/sssykim/Perturb-Net`.

References

1. GTEx Consortium (2015). The genotype-tissue expression (GTEx) pilot analysis: multi-tissue gene regulation in humans. Science *348*, 648–660.

2. Altelaar, A. M., Munoz, J., and Heck, A. J. (2013). Next-generation proteomics: towards an integrative view of proteome dynamics. Nature Reviews Genetics *14*, 35.

3. Shulaev, V. (2006). Metabolomics technology and bioinformatics. Briefings in bioinformatics *7*, 128–139.

4. Cusanovich, D. A., Pavlovic, B., Pritchard, J. K., and Gilad, Y. (2014). The functional consequences of variation in transcription factor binding. PLoS Genet *10*, e1004226.

5. Shalem, O., Sanjana, N. E., and Zhang, F. (2015). High-throughput functional genomics using CRISPR–Cas9. Nature Reviews Genetics *16*, 299.

6. Jansen, R. C. (2003). Studying complex biological systems using multifactorial perturbation. Nature Reviews Genetics *4*, 145.

7. Sohn, K.-A. and Kim, S. (2012). Joint estimation of structured sparsity and output structure in multiple-output regression via inverse-covariance regularization. In Proceedings of the 15th International Conference on Artificial Intelligence and Statistics (AISTATS) pages 1081–1089, JMLR W&CP.

8. Zhang, L. and Kim, S. (2014). Learning gene networks under SNP perturbations using eQTL datasets. PLoS computational biology *10*, e1003420.

9. Frot, B., Jostins, L., and McVean, G. (2018). Graphical model selection for Gaussian conditional random fields in the presence of latent variables. Journal of the American Statistical Association *accepted*.

10. Emilsson, V., Thorleifsson, G., Zhang, B., Leonardson, A. S., Zink, F., Zhu, J., Carlson, S., Helgason, A., Walters, G. B., Gunnarsdottir, S. *et al.* (2008). Genetics of gene expression and its effect on disease. Nature *452*, 423–428.

11. Schadt, E. E. (2009). Molecular networks as sensors and drivers of common human diseases. Nature *461*, 218–223.

12. Schadt, E. E., Molony, C., Chudin, E., Hao, K., Yang, X., Lum, P. Y., Kasarskis, A., Zhang, B., Wang, S., Suver, C. *et al.* (2008). Mapping the genetic architecture of gene expression in human liver. PLoS biology *6*, e107.

13. Giambartolomei, C., Vukcevic, D., Schadt, E. E., Franke, L., Hingorani, A. D., Wallace, C., and Plagnol, V. (2014). Bayesian test for colocalisation between pairs of genetic association studies using summary statistics. PLoS genetics *10*, e1004383.

14. He, X., Fuller, C. K., Song, Y., Meng, Q., Zhang, B., Yang, X., and Li, H. (2013). Sherlock: detecting gene-disease associations by matching patterns of expression QTL and GWAS. The American Journal of Human Genetics *92*, 667–680.

15. Gamazon, E. R., Wheeler, H. E., Shah, K. P., Mozaffari, S. V., Aquino-Michaels, K., Carroll, R. J., Eyler, A. E., Denny, J. C., Nicolae, D. L., Cox, N. J. *et al.* (2015). A gene-based association method for mapping traits using reference transcriptome data. Nature genetics *47*, 1091.

16. Greenawalt, D. M., Dobrin, R., Chudin, E., Hatoum, I. J., Suver, C., Beaulaurier, J., Zhang,

B., Castro, V., Zhu, J., Sieberts, S. K. *et al.* (2011). A survey of the genetics of stomach, liver, and adipose gene expression from a morbidly obese cohort. Genome Research *21*, 1008–1016.

17. Gibson, G., Powell, J. E., and Marigorta, U. M. (2015). Expression quantitative trait locus analysis for translational medicine. Genome Medicine *7*, 60.

18. Moffatt, M. F., Gut, I. G., Demenais, F., Strachan, D. P., Bouzigon, E., Heath, S., Von Mutius, E., Farrall, M., Lathrop, M., and Cookson, W. O. (2010). A large-scale, consortium-based genomewide association study of asthma. New England Journal of Medicine *363*, 1211–1221.

19. Wellcome Trust Case Control Consortium (2007). Genome-wide association study of 14,000 cases of seven common diseases and 3,000 shared controls. Nature *447*, 661.

20. Curtis, R. E., Yin, J., Kinnaird, P., and Xing, E. P. (2012). Finding genome-transcriptome-phenome association with structured association mapping and visualization in genamap. In Biocomputing 2012 pages 327–338. World Scientific.

21. Peters, L. A., Perrigoue, J., Mortha, A., Iuga, A., Song, W.-m., Neiman, E. M., Llewellyn, S. R., Di Narzo, A., Kidd, B. A., Telesco, S. E. *et al.* (2017). A functional genomics predictive network model identifies regulators of inflammatory bowel disease. Nature genetics *49*, 1437.

22. Wytock, M. and Kolter, J. (2013). Sparse Gaussian conditional random fields: algorithms, theory, and application to energy forecasting. In Proceedings of the 30th International Conference on Machine Learning volume 28 JMLR W&CP.

23. Group, C. A. M. P. R. (1999). The childhood asthma management program (CAMP): design, rationale, and methods. Controlled clinical trials *20*, 91–120.

24. Group, C. A. M. P. R. (2000). Long-term effects of budesonide or nedocromil in children with asthma. New England Journal of Medicine *343*, 1054–1063.

25. Murphy, A., Chu, J.-H., Xu, M., Carey, V. J., Lazarus, R., Liu, A., Szefler, S. J., Strunk, R., DeMuth, K., Castro, M. *et al.* (2010). Mapping of numerous disease-associated expression

polymorphisms in primary peripheral blood cd4+ lymphocytes. Human molecular genetics *19*, 4745–4757.

26. Lafferty, J., McCallum, A., and Pereira, F. C. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In Proceedings of the 18th International Conference on Machine Learning volume 951 pages 282–289,.

27. Koller, D. and Friedman, N. (2009). *Probabilistic graphical models: principles and techniques.* MIT press.

28. Zhang, L., Zeng, Z., and Ji, Q. (2011). Probabilistic image modeling with an extended chain graph for human activity recognition and image segmentation. IEEE Transactions on Image Processing *20*, 2401–2413.

29. Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society. Series B (Methodological) *58*, 267–288.

30. Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. Journal of the royal statistical society. Series B (methodological) *39*, 1–38.

31. Wu, T. T., Chen, Y. F., Hastie, T., Sobel, E., and Lange, K. (2009). Genome-wide association analysis by lasso penalized logistic regression. Bioinformatics *25*, 714–721.

32. Reid, S., Tibshirani, R., and Friedman, J. (2016). A study of error variance estimation in lasso regression. Statistica Sinica *26*, 35–67.

33. Scheet, P. and Stephens, M. (2006). A fast and flexible statistical model for large-scale population genotype data: applications to inferring missing genotypes and haplotypic phase. The American Journal of Human Genetics *78*, 629–644.

34. Chen, M., Lin, Z., Ma, Y., and Wu, L. (2009). The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. Technical report Coordinated Science Laboratory, University of Illinois at Urbana-Champaign.

35. Qian, J., Hastie, T., Friedman, J., Tibshirani, R., and Simon, N. (2013). Glmnet for matlab. Accessed: Nov *13*, 2017.

36. Rothman, A. J., Levina, E., and Zhu, J. (2010). Sparse multivariate regression with covariance estimation. Journal of Computational and Graphical Statistics *19*, 947–962.

37. Karypis, G. and Kumar, V. (1995). Metis-unstructured graph partitioning and sparse matrix ordering system, version 2.0. Technical report University of Minnesota, Department of Computer Science and Engineering, Army HPC Research Center, Minneapolis, MN.

38. Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., Davis, A. P., Dolinski, K., Dwight, S. S., Eppig, J. T. *et al.* (2000). Gene ontology: tool for the unification of biology. Nature Genetics *25*, 25.

39. Gene Ontology Consortium (2004). The gene ontology (GO) database and informatics resource. Nucleic Acids Research *32*, D258–D261.

40. Nicolae, D. L., Gamazon, E., Zhang, W., Duan, S., Dolan, M. E., and Cox, N. J. (2010). Trait-associated snps are more likely to be eQTLs: annotation to enhance discovery from GWAS. PLoS genetics *6*, e1000888.

41. Wills-Karp, M. and Ewart, S. L. (2004). Time to draw breath: asthma-susceptibility genes are identified. Nature Reviews Genetics *5*, 376.

42. Ober, C., Tsalenko, A., Parry, R., and Cox, N. J. (2000). A second-generation genomewide screen for asthma-susceptibility alleles in a founder population. The American Journal of Human Genetics *67*, 1154–1162.

43. Haagerup, A., Bjerke, T., Schiøtz, P. O., Binderup, H., Dahl, R., and Kruse, T. (2002). Asthma and atopy–a total genome scan for susceptibility genes. Allergy *57*, 680–686.

44. Kurz, T., Altmueller, J., Strauch, K., Rüschendorf, F., Heinzmann, A., Moffatt, M., Cookson, W., Inacio, F., Nürnberg, P., Stassen, H. *et al.* (2005). A genome-wide screen on the genetics

of atopy in a multiethnic European population reveals a major atopy locus on chromosome 3q21. 3. Allergy *60*, 192–199.

45. Al-Shobaili, H. A., Ahmed, A. A., Alnomair, N., Alobead, Z. A., and Rasheed, Z. (2016). Molecular genetic of atopic dermatitis: an update. International journal of health sciences *10*, 96.

46. Shim, E.-J., Chun, E., Kang, H.-R., Cho, S.-H., Min, K.-U., and Park, H.-W. (2013). Expression of semaphorin 3a and neuropilin 1 in asthma. Journal of Korean medical science *28*, 1435–1442.

47. Kim, Y., Park, C., Shin, H., Choi, J., Cheong, H., Park, B., Choi, Y., Jang, A., Park, S., Lee, Y. *et al.* (2007). A promoter nucleotide variant of the dendritic cell-specific DCNP1 associates with serum IgE levels specific for dust mite allergens among the korean asthmatics. Genes and immunity *8*, 369.

48. Liao, S.-Y., Lin, X., and Christiani, D. C. (2014). Genome-wide association and network analysis of lung function in the Framingham heart study. Genetic epidemiology *38*, 572–578.

49. Platoshyn, O., Brevnova, E. E., Burg, E. D., Yu, Y., Remillard, C. V., and Yuan, J. X.-J. (2006). Acute hypoxia selectively inhibits kcna5 channels in pulmonary artery smooth muscle cells. American Journal of Physiology-Cell Physiology *290*, C907–C916.

50. Archer, S. L., Wu, X.-C., Thébaud, B., Nsair, A., Bonnet, S., Tyrrell, B., McMurtry, M. S., Hashimoto, K., Harry, G., and Michelakis, E. D. (2004). Preferential expression and function of voltage-gated, o2-sensitive k+ channels in resistance pulmonary arteries explains regional heterogeneity in hypoxic pulmonary vasoconstriction: ionic diversity in smooth muscle cells. Circulation research *95*, 308–318.

51. Tantisira, K. G., Damask, A., Szefler, S. J., Schuemann, B., Markezich, A., Su, J., Klanderman, B., Sylvia, J., Wu, R., Martinez, F. *et al.* (2012). Genome-wide association identifies the T gene as a novel asthma pharmacogenetic locus. American journal of respiratory and critical care medicine *185*, 1286–1291.

52. Chappell, C. P., Giltiay, N. V., Draves, K. E., Chen, C., Hayden-Ledbetter, M. S., Shlomchik, M. J., Kaplan, D. H., and Clark, E. A. (2014). Targeting antigens through blood dendritic cell antigen 2 on plasmacytoid dendritic cells promotes immunologic tolerance. The Journal of Immunology *192*, 5789–5801.

53. Riboldi, E., Daniele, R., Parola, C., Inforzato, A., Arnold, P. L., Bosisio, D., Fremont, D. H., Bastone, A., Colonna, M., and Sozzani, S. (2011). Human c-type lectin domain family 4, member c (clec4c/bdca-2/cd303) is a receptor for asialo-galactosyl-oligosaccharides. Journal of Biological Chemistry *286*, 35329–35333.

54. Friedman, J., Hastie, T., Höfling, H., Tibshirani, R. *et al.* (2007). Pathwise coordinate optimization. The Annals of Applied Statistics *1*, 302–332.

55. Hsieh, C.-J., Dhillon, I. S., Ravikumar, P. K., and Sustik, M. A. (2011). Sparse inverse covariance matrix estimation using quadratic approximation. In Advances in Neural Information Processing Systems 24 pages 2330–2338.

56. Hsieh, C.-J., Sustik, M. A., Dhillon, I. S., Ravikumar, P. K., and Poldrack, R. (2013). BIG & QUIC: Sparse inverse covariance estimation for a million variables. In Advances in Neural Information Processing Systems 26 pages 3165–3173.

Table 1. GO categories enriched in gene modules in the estimated asthma gene network

	Size	Biological Process Pathway	P-value	Overlap*
1	314	Cellular macromolecule metabolic process	2.94×10^{-12}	159 / 7006
2	297	Cellular nitrogen compound metabolic process	1.64×10^{-4}	112 / 5164
3	314	Nucleobase-containing compound metabolic process	5.62×10^{-13}	123 / 4538
4	314	Organelle organization	2.02×10^{-4}	79 / 3167
5	314	Nucleobase-containing compound metabolic process	4.58×10^{-4}	106 / 4538
6	314	Unclassified	NA	NA
7	314	Cellular localization	1.80×10^{-4}	60 / 2287
8	314	Cellular metabolic process	1.73×10^{-4}	178 / 9003
9	314	Cellular metabolic process	8.39×10^{-6}	185 / 9003
10	314	Macromolecule metabolic process	8.73×10^{-5}	159 / 7749
11	314	Heterocycle metabolic process	7.63×10^{-3}	103 / 4715
12	298	Translation	1.65×10^{-9}	27 / 383
13*	297	Immune system process	3.66×10^{-12}	83 / 2552
		Response to stimulus	1.70×10^{-9}	162 / 8009
		Response to stress	8.43×10^{-9}	90 / 3333
14*	314	Immune response	3.96×10^{-38}	106 / 1673
		Leukocyte activation involved in immune response	9.04×10^{-31}	62 / 607
		Granulocyte activation	1.24×10^{-26}	53 / 495
15*	313	Cell activation in immune response	4.95×10^{-32}	65 / 611
		Myeloid leukocyte activation	5.28×10^{-32}	63 / 566
		Immune system process	1.01×10^{-31}	124 / 2552
16	290	Cellular process	3.73×10^{-3}	132 / 15013
17	290	Regulation of macromolecule metabolic process	1.58×10^{-3}	74 / 6142
18	282	Organonitrogen compound metabolic process	1.97×10^{-2}	60 / 5523
19	285	Cell cycle	1.50×10^{-8}	37 / 1355
20	295	Cellular component organization or biogenesis	4.34×10^{-3}	66 / 5525

* The number of genes in the overlap / the total number of genes in the GO category

Table 2. Prediction errors of different methods on asthma test set

Prediction task	Lasso	Our Model	Our Model with Semi-supervised Learning
$\mathbf{y}\vert\mathbf{x}$	0.76494	0.75322	**0.75318**
$\mathbf{z}\vert\mathbf{y}$	1.03486	0.97068	**0.89317**
$\mathbf{y}\vert\mathbf{x},\mathbf{z}$	0.78161	0.75346	**0.75324**
$\mathbf{z}\vert\mathbf{x}$	0.85785	0.85795	**0.85709**

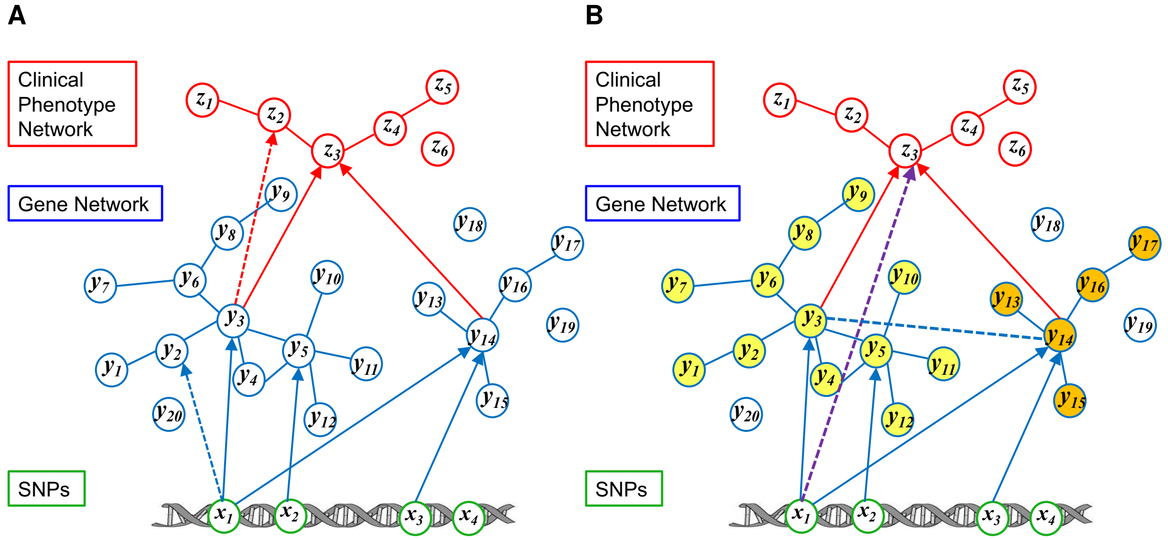

Figure 1. Illustration of the Perturb-Net approach. (A) Perturb-Net uses a sparse Gaussian chain graph model with a cascade of two sCGGMs, one for a gene network influenced by SNPs (blue solid edges and nodes) and the other for a clinical trait network influenced by gene expression levels (red solid edges and nodes). The sCGGM inference procedures can be used to infer hidden interactions in each of the two component sCGGMs, such as the indirect effect of SNP x_1 on expression level y_2 through expression level y_3 (blue dashed arrow) and the indirect effect of expression level y_3 on phenotype z_2 through phenotype z_3 (red dashed arrow). (B) The inference procedures of sparse Gaussian chain graph models are used to infer the information on how the gene network mediates SNP effects on phenotypes. Examples of such inferred interactions are shown for the perturbation effect of SNP x_1 on phenotype z_3 (purple dashed arrow), which can be decomposed into two components mediated by each of the two gene modules (yellow and orange nodes), and the inferred dependencies between expression level y_3 and expression level y_{14} (blue dashed line) induced by phenotype z_3 in the posterior gene network, after seeing the clinical phenotypes.

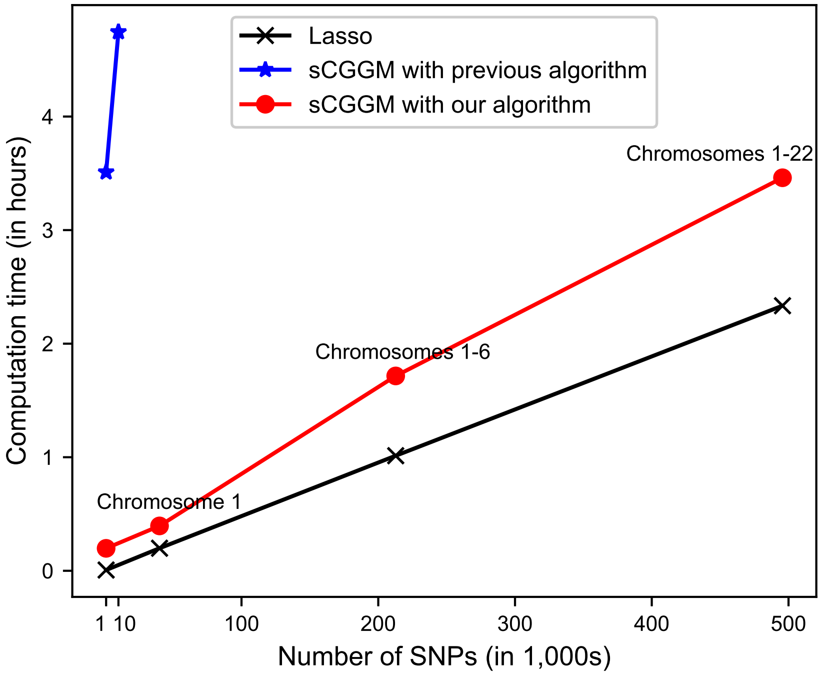

Figure 2. Comparison of computation time of different methods. The computation time of our Mega-sCGGM is compared with that of previous learning algorithm for sCGGMs and Lasso. We applied all methods to all expression data and genotype data from chromosome 1, chromosomes 1-6, chromosomes 1-16, and chromosomes 1-22. The previous algorithm for sCGGMs ran out of memory at chromosome 1, so we obtained its computation time with much smaller datasets with 1,000 and 10,000 SNPs.

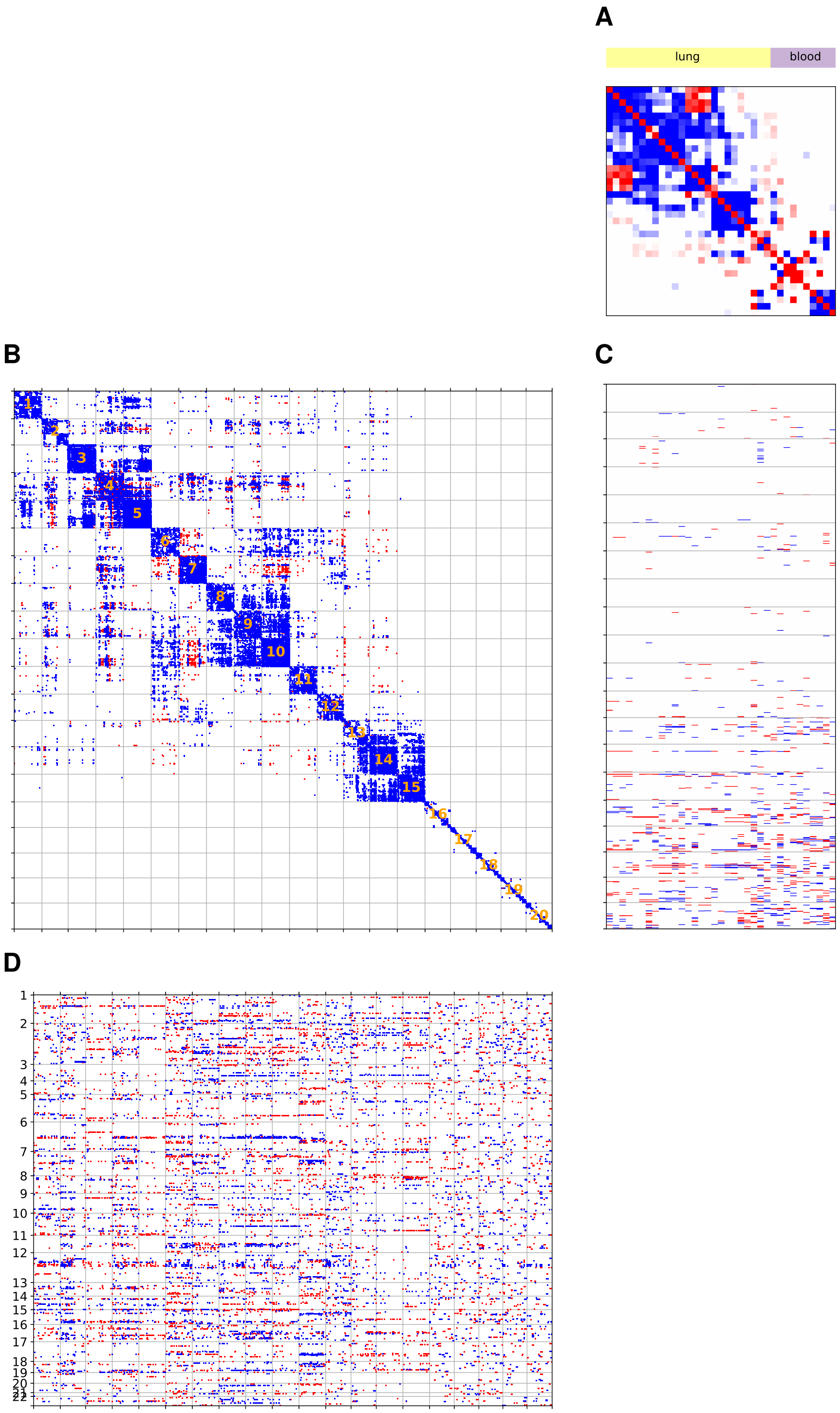

Figure 3. The Perturb-Net model estimated from asthma data. The parameters of the sparse Gaussian chain graph model estimated from the asthma data are shown. (A) Asthma phenotype network $\mathbf{\Lambda_z}$. The phenotypes were ordered by hierarchical clustering applied to within each of the two groups of phenotypes, lung function traits (yellow) and blood test traits (purple). (B) Gene network $\mathbf{\Lambda_y}$. The gene network is annotated with 20 modules obtained from applying a network clustering algorithm METIS[37] to $\mathbf{\Lambda_y}$. (C) The influence of gene expression levels on phenotypes $\mathbf{\Theta_{yz}}$. (D) SNP perturbation of gene expression levels $\mathbf{\Theta_{xy}}$ for the top 1,000 eQTL hotspots, ordered by genomic location and labeled by chromosomes. In each panel, non-zero elements of the estimated parameters are shown as blue for positive interactions and red for negative interactions.

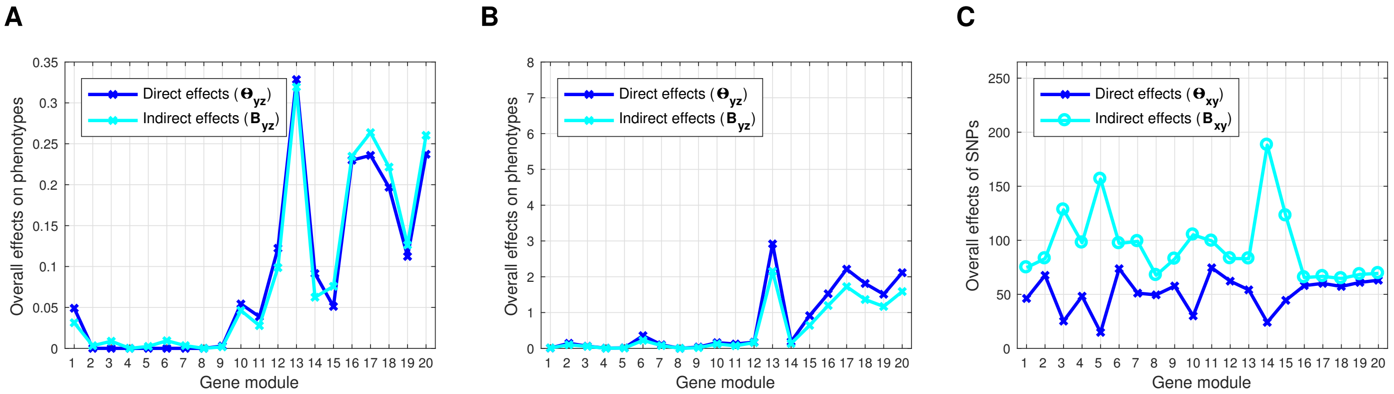

Figure 4. SNP effects on gene modules and gene-module effects on phenotypes. Given the estimated $\mathbf{\Theta_{yz}}$ and inferred $\mathbf{B_{yz}}$ for gene-expression effects on phenotypes from the Perturb-Net model, we show the gene-module effects on each group of phenotypes for (A) lung and (B) blood, computed as the sum of absolute effect sizes across all genes within the module and across all phenotypes in the group. (C) Given the estimated $\mathbf{\Theta_{xy}}$ and inferred $\mathbf{B_{xy}}$ for SNP effects on gene network, we show the SNP effects on each gene module, summarized as the sum of absolute effect sizes across all SNPs and all genes within the module.

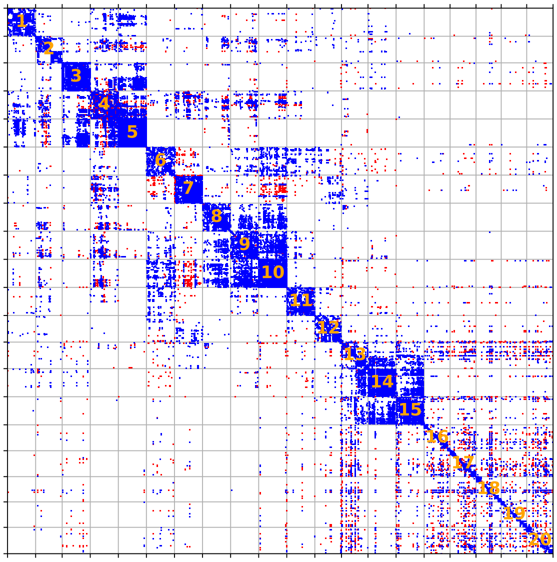

Figure 5. Asthma posterior gene network. The posterior gene network $\mathbf{\Lambda}_{\mathbf{y}|\mathbf{x},\mathbf{z}}$ after taking into account the phenotype data.

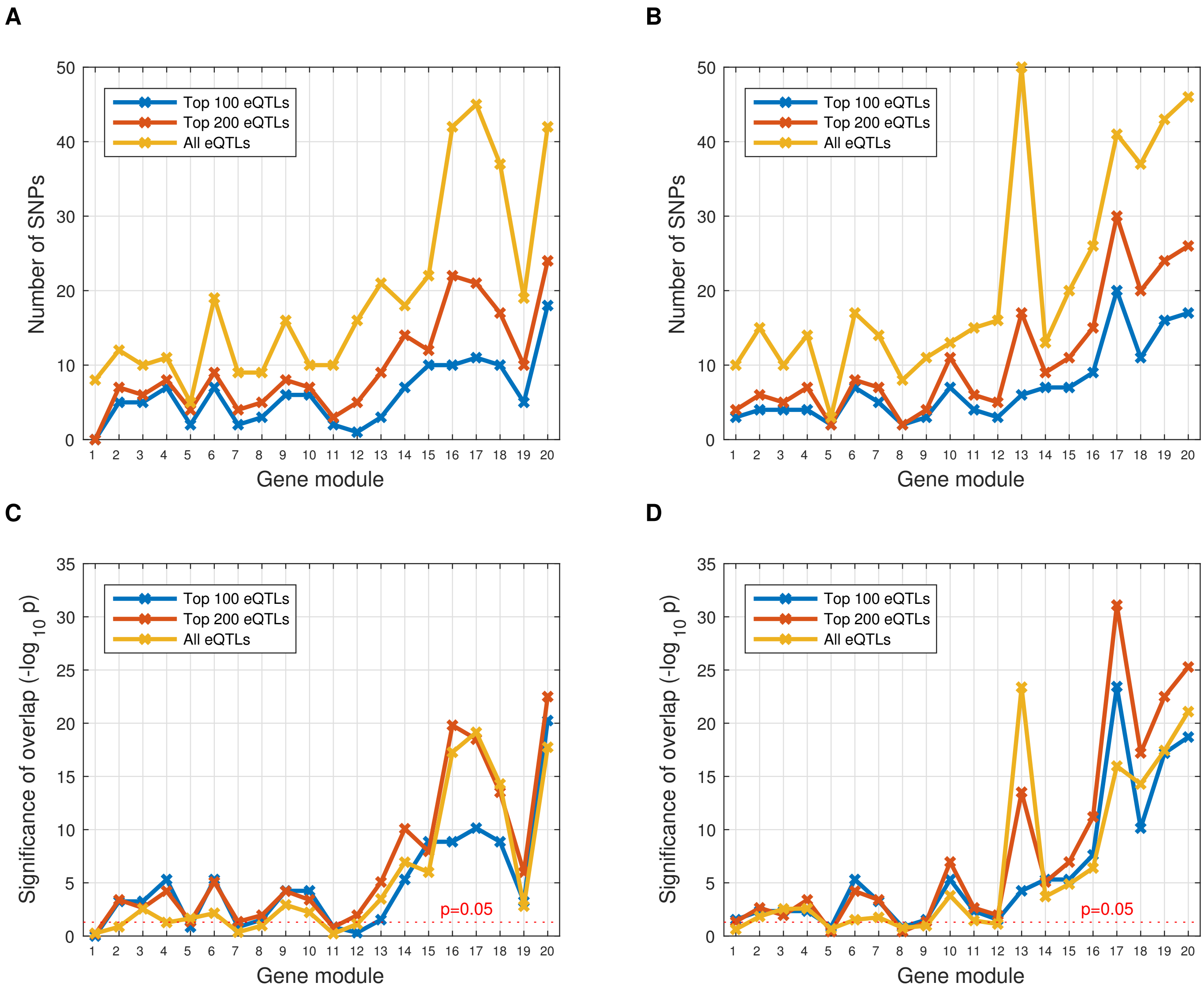

Figure 6. Overlap between SNPs perturbing phenotype network and SNPs perturbing gene network. For each gene module and each group of lung and blood phenotypes, we found the overlap between the top 200 SNPs perturbing the phenotype subnetwork and each of the top 100, 200, and all eQTLs perturbing the gene module. The number of SNPs in the overlap is shown for (A) lung phenotypes and (B) blood phenotypes. Statistical significance of the overlap is shown for (C) lung phenotypes and (D) blood phenotypes.

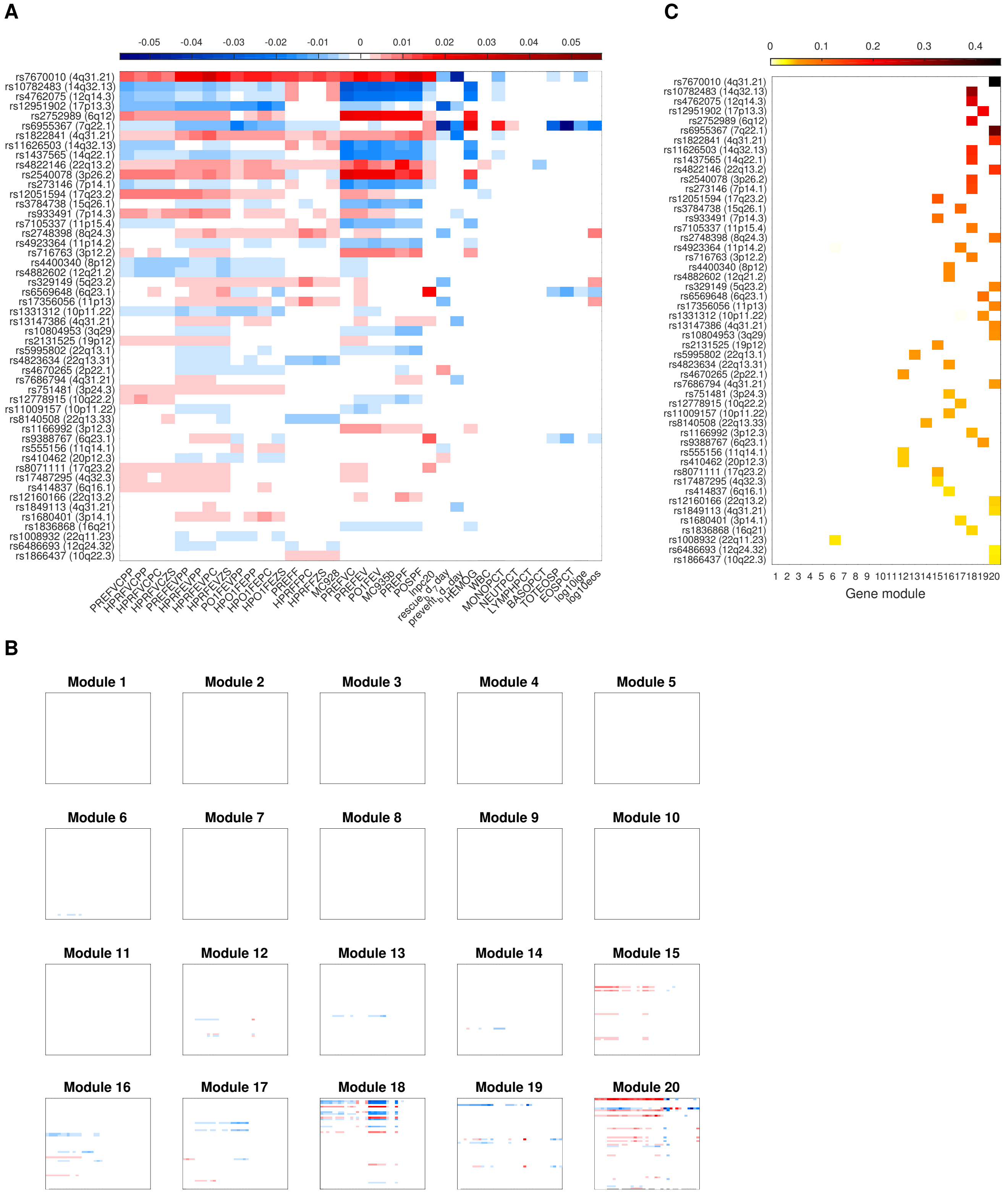

Figure 7. Top 50 SNPs perturbing lung phenotypes and their perturbations effects on phenotypes mediated by gene modules. For the top 50 SNPs perturbing lung phenotypes, we show (A) their effect sizes on phenotypes $\mathbf{B_{xz}}$ and (B) the decomposition of $\mathbf{B_{xz}}$ into component effects $\mathbf{B}_{\mathbf{xz}}^{M_1}, \ldots, \mathbf{B}_{\mathbf{xz}}^{M_{20}}$ mediated by each of the 20 gene modules. The sum over all component effects in Panel (B) is equal to the overall effects in Panel (A). (C) We summarize each component SNP effect $\mathbf{B}_{\mathbf{xz}}^{m}$ for module m in Panel (B) as a row-wise sum of $\mathbf{B}_{\mathbf{xz}}^{m}$, shown as the mth column in the figure. The SNPs are ordered according to their overall effect sizes on the lung phenotypes.

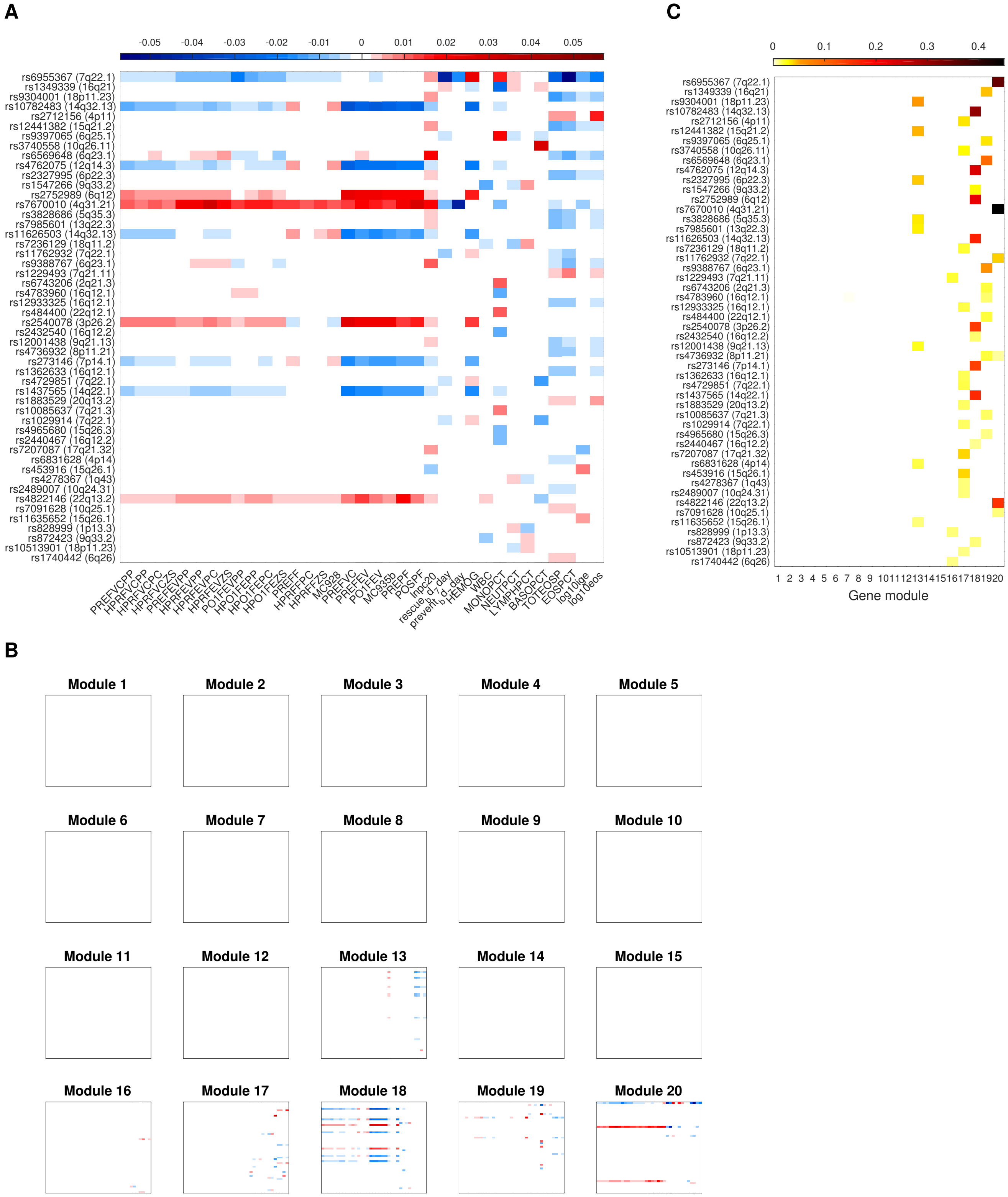

Figure 8. Top 50 SNPs perturbing blood phenotypes and their perturbations effects on phenotypes mediated by gene modules. For the top 50 SNPs perturbing blood phenotypes, we show (A) their effect sizes on phenotypes $\mathbf{B_{xz}}$ and (B) the decomposition of $\mathbf{B_{xz}}$ into component effects $\mathbf{B}^{M_1}_{\mathbf{xz}}, \ldots, \mathbf{B}^{M_{20}}_{\mathbf{xz}}$ mediated by each of the 20 gene modules. The sum over all component effects in Panel (B) is equal to the overall effects in Panel (A). (C) We summarize each component SNP effect $\mathbf{B}^{m}_{\mathbf{xz}}$ for module m in Panel (B) as a row-wise sum of $\mathbf{B}^{m}_{\mathbf{xz}}$, shown as the mth column in the figure. The SNPs are ordered according to their overall effect sizes on blood phenotypes.

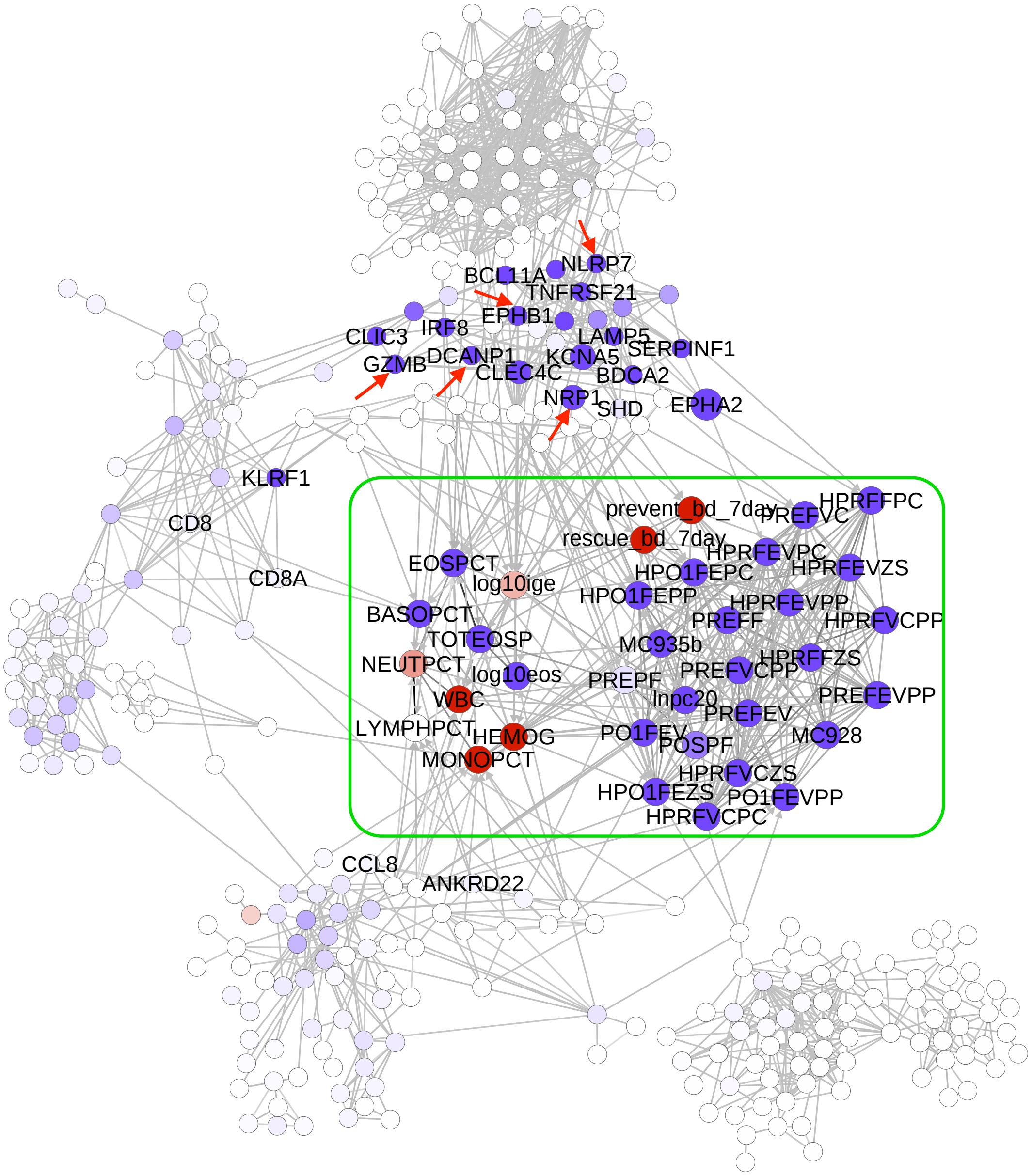

Figure 9. The gene network for module 13, its influence on asthma phenotypes, and its perturbation by SNP rs63340. The asthma phenotype network $\mathbf{\Lambda_z}$ in our estimated model is shown in the green box and the gene network $\mathbf{\Lambda_y}$ for module 13 is shown outside of the green box. The edges across the two networks correspond to direct influence of expression levels on phenotypes $\mathbf{\Theta_{yz}}$. The five genes (*NRP1, DCANP1, EPHB1, NLRP7, and GZMB*) whose expression is directly perturbed by SNP rs63340 with effect size > 0.05 in $\mathbf{\Theta_{xy}}$ are labeled with arrows, colored red to indicate positive eQTL effects. Node colors depict the indirect effects of this eQTL on gene expression levels $\mathbf{B_{xy}}$ and phenotypes $\mathbf{B_{xz}}$, with red for up-regulation and blue for down-regulation. Node size of genes depicts the component of the eQTL effects on phenotypes mediated by the given gene, the row of $\mathbf{B^{\mathit{m}}_{xz}}$ for SNP rs63340 and for gene m in module 13 summed across all phenotypes.

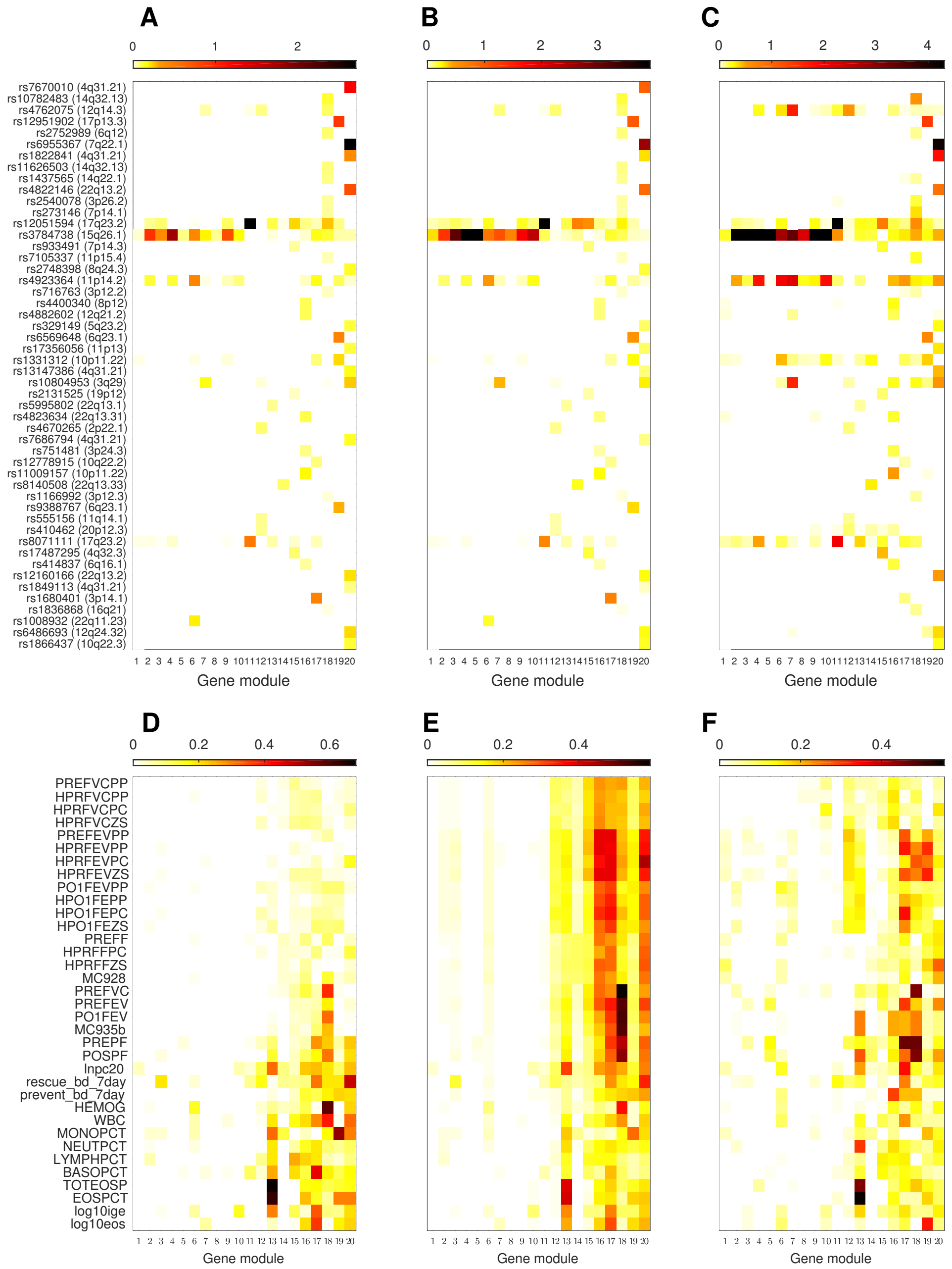

Figure 10. Comparison of different methods for learning the cascaded influence of SNPs to gene modules to phenotypes. For the top 50 SNPs perturbing lung phenotypes (Figure 7), the effects of these SNPs on each of the gene modules are shown for (A) $\mathbf{\Theta_{yz}}$ from our model, (B) $\mathbf{B_{yz}}$ inferred from our model, and (C) $\mathbf{A_{yz}}$ from the two-layer Lasso. The effects of the expression levels in each gene module on lung phenotypes are shown for (D) $\mathbf{\Theta_{yz}}$ from our model, (E) $\mathbf{B_{yz}}$ inferred from our model, and (F) $\mathbf{A_{yz}}$ from the two-layer Lasso. The effect sizes in each model parameter matrix above were summed across all genes within each module after taking absolute values.