SpiderLearner: An ensemble approach to Gaussian graphical model estimation

Katherine H. Shutta[1¤*], Laura B. Balzer[1], Denise M. Scholtens[2], Raji Balasubramanian[1]

1 Department of Biostatistics and Epidemiology, University of Massachusetts - Amherst, Amherst, MA, USA

2 Department of Preventive Medicine, Division of Biostatistics, Northwestern University Feinberg School of Medicine, Chicago, IL, USA

¤Current Address: Department of Biostatistics and Epidemiology, University of Massachusetts - Amherst, Amherst, MA, USA

* kshutta@umass.edu

Abstract

Multivariate biological data are often modeled using networks in which nodes represent a biological variable (e.g., genes) and edges represent associations (e.g., coexpression). A Gaussian graphical model (GGM), or partial correlation network, is an undirected graphical model in which a weighted edge between two nodes represents the magnitude of their partial correlation, and the absence of an edge indicates zero partial correlation. A GGM provides a roadmap of direct dependencies between variables, providing a valuable systems-level perspective. Many methods exist for estimating GGMs; estimated GGMs are typically highly sensitive to choice of method, posing an outstanding statistical challenge. We address this challenge by developing SpiderLearner, a tool that combines a range of candidate GGM estimation methods to construct an ensemble estimate as a weighted average of results from each candidate. In simulation studies, SpiderLearner performs better than or comparably to the best of the candidate methods. We apply SpiderLearner to estimate a GGM for gene expression in a publicly available dataset of 260 ovarian cancer patients. Using the community structure of the GGM, we develop a network-based risk score which we validate in six independent datasets. The risk score requires only seven genes, each of which has important biological function. Our method is flexible, extensible, and has demonstrated potential to identify *de novo* biomarkers for complex diseases. An open-source implementation of our method is available at `https://github.com/katehoffshutta/SpiderLearner`.

Introduction

Gaussian graphical models (GGMs) are a modeling framework for network-based analyses of multivariate data. This framework begins with the assumption that data are sampled from a multivariate normal distribution; under this

assumption, a GGM is defined as a graph in which nodes correspond to variables and weighted edges correspond to the magnitude of the partial correlation between them (1). In this framework, the absence of an edge between nodes corresponds to zero partial correlation, i.e., conditional independence between the variables, given the other variables in the network.

The weighted adjacency matrix of a GGM consists of the partial correlations between nodes. Let $\mathbf{X} \sim N_p(\mathbf{0}, \Sigma)$ be a centered p-dimensional multivariate normal random variable, and let $\mathbf{X}_{-i,-j}$ represent $\mathbf{X}$ with the i^{th} and j^{th} variables removed. The partial correlation between X_i and X_j is defined as:

$$\rho_{X_i,X_j|\mathbf{X}_{-i,-j}} = \frac{Cov[(X_i, X_j|\mathbf{X}_{-i,-j})]}{\sqrt{Var[(X_i|\mathbf{X}_{-i,-j}]}\sqrt{Var[(X_j|\mathbf{X}_{-i,-j})]}} \tag{1}$$

Under the assumption of multivariate normality, a particularly useful relationship holds between the precision matrix $\Theta = \Sigma^{-1}$ and the partial correlation (2). Let θ_{ij} represent the i, j^{th} element of Θ; it can be shown that

$$\rho_{X_i,X_j|X_{-i,-j}} = -\frac{\theta_{ij}}{\sqrt{\theta_{ii}\theta_{jj}}} \tag{2}$$

Equation 2 shows that estimating a GGM is equivalent to estimating Θ. In the case where the sample size n is much larger than the number of predictors p, a maximum likelihood estimate of Θ can be found simply by inverting the sample covariance. When n is close to or less than p, this inverse is undefined or numerically unstable, meaning this method cannot be used. The usual approach in this setting is the graphical lasso, which estimates a sparse precision matrix by optimizing the penalized likelihood function (3; 4; 5)

$$\ell(\Theta) = \log|\Theta| - tr(S\Theta) - \lambda||\Theta||_1 \tag{3}$$

where S is the sample covariance matrix of the observed data and $\lambda > 0$ is a non-negative tuning parameter, with higher values of λ leading to sparser estimates of Θ. Several existing open-source software resources implement various versions and extensions of the graphical lasso, including methods for selecting the tuning parameter λ. For example, the `glasso` R package implements the original algorithm developed in 2008 by Friedman et. al. as augmented by computational advances developed in 2011 by Witten et. al. (3; 6). The `huge` R package incorporates the graphical lasso algorithm and additionally provides options for a tuning-insensitive method called `tiger` published by Liu et al. in 2017 (7),(8). The `bootnet` R package includes a broad range of different network estimation methods, seven of which are for GGM estimation, in a framework for bootstrap estimation of network accuracy (9).

There is clearly no shortage of options for a researcher who is interested in estimating a GGM; this is both a blessing and a curse. Estimating a GGM using these packages requires the researcher to make several decisions with regard to data preprocessing, tuning parameter selection, choice of scoring criteria for model selection, and selection of hyperparameters for these scoring criteria. The final estimated GGM may be highly sensitive to these choice, making it difficult to compare GGMs across studies and assess reproducibility (10; 11; 12). Because it is impossible to know

a priori which approach is best for a given problem, researcher bias toward use of a particular "favorite method" can have a large impact on the estimation and interpretation of a GGM, consequently affecting the scientific conclusions inferred.

Ensemble methods are a broad class of statistical approaches which follow the general principle of combining several different candidate models to generate a single ensemble model (13; 14). One such method is the Super Learner approach of van der Laan et al (15). Super Learner uses an internal cross-validation scheme to fit a convex combination of candidate algorithms ("learners") that minimizes a user-defined loss function. This convex combination is the Super Learner ensemble model. Large-sample properties of the Super Learner are established by comparison to the expected loss (i.e., risk) of an oracle model, which is the best model among all possible convex combinations given the true, unknown, data generating process. Under mild conditions on the loss function and the set of candidate learners, the expected difference between the risk of the Super Learner ensemble model and the risk of the oracle model converges to zero as the sample size goes to infinity(15).

Here, we develop SpiderLearner, a network estimation tool which applies the Super Learner approach to the problem of fitting a GGM by optimizing a likelihood-based loss function through the use of cross-validation. Our approach improves GGM estimation by circumventing the complicated decision-making burden described above. The SpiderLearner considers a library of candidate GGM estimation methods and constructs the optimal convex combination of their results, eliminating the need for the researcher to make arbitrary decisions in the estimation process. Through simulation studies, we demonstrate that the SpiderLearner achieves equal or better performance than each of the candidate approaches according to several criteria (out-of-sample likelihood, bias, mean squared error (MSE), matrix correlation).

Previous work has shown that the use of network models in prediction problems for disease outcomes is a promising area of work (e.g.,(16; 17; 18)). We connect this previous work to ours by presenting an illustrative application of the SpiderLearner to develop a robust risk score for ovarian cancer prognosis based on a publicly-available gene expression dataset (19; 20).

The SpiderLearner improves the applicability of GGMs as a network modeling framework, creating a more robust methodology by using data-driven ensemble learning to eliminate the need for researchers to choose an estimation approach. Our risk score application demonstrates the potential of the SpiderLearner to discover meaningful biological insights in complex multivariate data.

Materials and methods

SpiderLearner model formulation

The foundations for a Super Learner-type method are (i) specifying a library of candidate algorithms, (ii) specifying a loss function, and (iii) implementing a cross-validation scheme to determine the optimal convex combination of the candidates (21). We introduce the foundations of our method similarly, but focus first on (ii) and (iii); we address (i) when describing our simulation study design.

To develop the loss function for the SpiderLearner, we begin by supposing that we have a library of M different candidate methods and have applied each candidate to obtain M estimated GGMs for a given input dataset. Our next goal is to estimate a weighted combination of these M estimates that may provide an even better fit than each method does alone, in the spirit of the Super Learner approach(15). We consider first a basic test-train setting, in which we assume the availability of two independent datasets X_{train} and X_{test}, drawn at random from the same population. Let $\hat{\Theta}_1^{(train)}, \ldots, \hat{\Theta}_M^{(train)}$ denote the estimated precision matrices from the application of the M different methods to X_{train}. We seek a convex combination

$$\Theta_{SL} = \alpha_1 \hat{\Theta}_1^{(train)} + \alpha_2 \hat{\Theta}_2^{(train)} + \cdots + \alpha_M \hat{\Theta}_M^{(train)}; \sum_{m=1}^{M} \alpha_m = 1; \alpha_m \geq 0 \tag{4}$$

that minimizes the negative log-likelihood in the independent dataset X_{test}:

$$-\ell(\Theta_{SL}) = -\log(Pr(X_{test}|\Theta_{SL})) \tag{5}$$

Let $X_{test}^{(i)}$ denote the data vector for the i^{th} observation in the test dataset. In the GGM setting, where $X_{test}^{(i)}|\Theta \sim MVN(0, \Theta), i = 1, \ldots, n$, the log likelihood can be expressed as:

$$\ell(\Theta_{SL}) \propto -\frac{n}{2}\log(|\Theta_{SL}|) - \frac{1}{2}\sum_{i=1}^{n}(X_{test}^{(i)})^T \Theta_{SL} X_{test}^{(i)} \tag{6}$$

We negate the log likelihood and incorporate our definition of Θ_{SL} from Equation 4 to develop a loss function in terms of the coefficients $\boldsymbol{\alpha} = \alpha_1, \ldots, \alpha_M$:

$$\begin{aligned} Q(\boldsymbol{\alpha}) &= \frac{n}{2}\log\left(|\alpha_1 \hat{\Theta}_1^{(train)} + \cdots + \alpha_M \hat{\Theta}_M^{(train)}|\right) && (7) \\ &+ \frac{1}{2}\sum_{i=1}^{n}(X_{test}^{(i)})^T \left(\alpha_1 \hat{\Theta}_1^{(train)} + \cdots + \alpha_M \hat{\Theta}_M^{(train)}\right) X_{test}^{(i)} && (8) \end{aligned}$$

This loss function 12 is then minimized, subject to the constraints of the convex combination:

$$\hat{\boldsymbol{\alpha}} = \text{argmin}_{\boldsymbol{\alpha}:\sum_{m=1}^{M}\alpha_m=1;\alpha_m\geq 0}\{Q(\boldsymbol{\alpha})\} \tag{9}$$

Standard constrained optimization algorithms can such as those implemented in the the `solnp` function from the R package `Rsolnp` (22) can be used to find the coefficients α that solve 9 on the test dataset. Once these coefficients have been found we complete the process by running the original M candidate methods again using the full dataset $X = X_{train} \cup X_{test}$, obtaining estimates $\hat{\Theta}_1 \ldots, \hat{\Theta}_M$. We then use these estimates to construct the SpiderLearner estimate of the precision matrix as:

$$\hat{\Theta}_{SL} = \sum_{m=1}^{M} \hat{\alpha}_m \hat{\Theta}_m \tag{10}$$

The train-test approach is limited in that the estimate of the out-of-sample loss will tend to be (i) an overestimate due to the relatively small size of the training dataset, and (ii) suffer from high variability due to the sensitivity of the approach to the characteristics of the training and test datasets (23). To overcome these limitations, we extend our approach from the simple train-test setting described above to K-fold cross-validation, where $K > 2$. K-fold cross-validation has the advantage of permitting the user to navigate the bias-variance tradeoff in the estimation of out-of-sample loss (23).

Briefly, we begin the K-fold cross-validation by partitioning the data X into K folds of approximately equal size $\sim n/K$. We next repeat the above process of determining the precision matrix estimator $\hat{\Theta}_{SL}$ K times; each time, data from the k^{th} fold is withheld ($k = 1, \ldots, K$) as the test set while the remaining $(K-1)$ of the folds serve as the training set.

To provide further detail, we first introduce some notation. Let X_k be the k^{th} fold of the dataset X, and let X_{-k} be the remainder of the dataset X with the k^{th} fold withheld. Let $\hat{\Theta}_m^{(-k)}$ be the precision matrix estimate for method m trained on X_{-k}, and let $X_k^{(i)}$ be the i^{th} observation in fold K. We define $\Theta_{SL}^{(-k)}$, which is a function of $\boldsymbol{\alpha}$, as:

$$\Theta_{SL}^{(-k)} = \alpha_1 \hat{\Theta}_1^{(-k)} + \cdots + \alpha_M \hat{\Theta}_M^{(-k)} \tag{11}$$

Next, we define $Q_k(\boldsymbol{\alpha})$ to be the loss of the estimate $\Theta_{SL}^{(-k)}$ evaluated on the withheld data X_k:

$$Q_k(\boldsymbol{\alpha}) = \frac{n}{2} \log\left(|\Theta_{SL}^{(-k)}|\right) + \frac{1}{2} \sum_{i=1}^{n} (X_k^{(i)})^T \Theta_{SL}^{(-k)} X_k^{(i)} \tag{12}$$

Let $\bar{Q}(\boldsymbol{\alpha})$ be the average loss aross K folds:

$$\bar{Q}(\boldsymbol{\alpha}) = \frac{1}{K} \sum_{k=1}^{K} Q_k(\boldsymbol{\alpha}) \tag{13}$$

Let n_k be the number of observations in the k^{th} fold. Then the K-fold cross-validated coefficient estimator of $\hat{\boldsymbol{\alpha}}$ is:

$$\hat{\boldsymbol{\alpha}} = \text{argmin}_{\boldsymbol{\alpha}:\sum_{m=1}^{M}\alpha_m=1;\alpha_m\geq 0}\left\{\frac{1}{K}\sum_{k=1}^{K}\left(-\frac{n_k}{2}\log(|\Theta_{SL}^{(-k)}|) - \frac{1}{2}\sum_{k=1}^{n_k}(X_k^{(i)})^T\Theta_{SL}^{(-k)}X_k^{(i)}\right)\right\} \tag{14}$$

Finally, let $\hat{\Theta}_m$ be the precision matrix estimate from method m using the full dataset. We then define the K-fold cross-validated SpiderLearner estimator as:

$$\hat{\Theta}_{SL} = \sum_{m=1}^{M}\hat{\alpha}_m\hat{\Theta}_m \tag{15}$$

A diagram of this workflow for $M = 4$ estimation methods and $K = 5$ cross-validation folds is shown in Figure 1. The choice of K may depend on a variety of factors including sample size and number of predictors (i.e., dimensionality of the problem); in practice, $K = 5$ and $K = 10$ have demonstrated generally good balance in the bias-variance trade off(23). We discuss the choice of K further in the Results section.

Large-sample properties

The large-sample properties of Super Learner derived by (15) require a bounded loss function. Our loss function $\bar{Q}(\alpha)$ (Equation 12) is not bounded; therefore, it is not clear if the large-sample oracle results of (15) apply with the log likelihood-based loss function evaluated on multivariate normal data. (24) note that oracle results also hold for certain types of unbounded loss functions as described in (25); however, it is not straightforward to formally show that $\bar{Q}(\alpha)$ meets the necessary criteria (see S1 Appendix in the Supplement for details). We note this as an area for future work, while observing that in practice the log likelihood is often used as a loss function for Super Learner estimation (e.g., (26; 27; 28)), and the log likelihood loss is provided as part of the standard implementation of SuperLearner (29).

We additionally explored a transformation that permits the application of the oracle results of (15). Specifically, $\bar{Q}(\alpha)$ can be transformed into a bounded loss function $\bar{Q}'(\alpha)$ by applying the inverse logit function:

$$\bar{Q}'(\alpha) = \frac{\exp(\bar{Q}(\alpha))}{1+\exp(\bar{Q}(\alpha))} \tag{16}$$

In practice, this transformation can be sensitive to the scale of $\bar{Q}(\alpha)$ and quickly become numerically equal to one for a broad range of α. We observed good behavior by scaling $\bar{Q}(\alpha)$ to the sample size n and number of predictors p:

$$\bar{Q}'(\alpha) = \frac{\exp(\frac{1}{np}\bar{Q}(\alpha))}{1+\exp(\frac{1}{np}\bar{Q}(\alpha))} \tag{17}$$

However, we suspect that further issues with the numerical stability of this transformation are possible and encourage diagnostics such as testing the value of $\bar{Q}'(\alpha)$ for different values of α when using this transformation in practice.

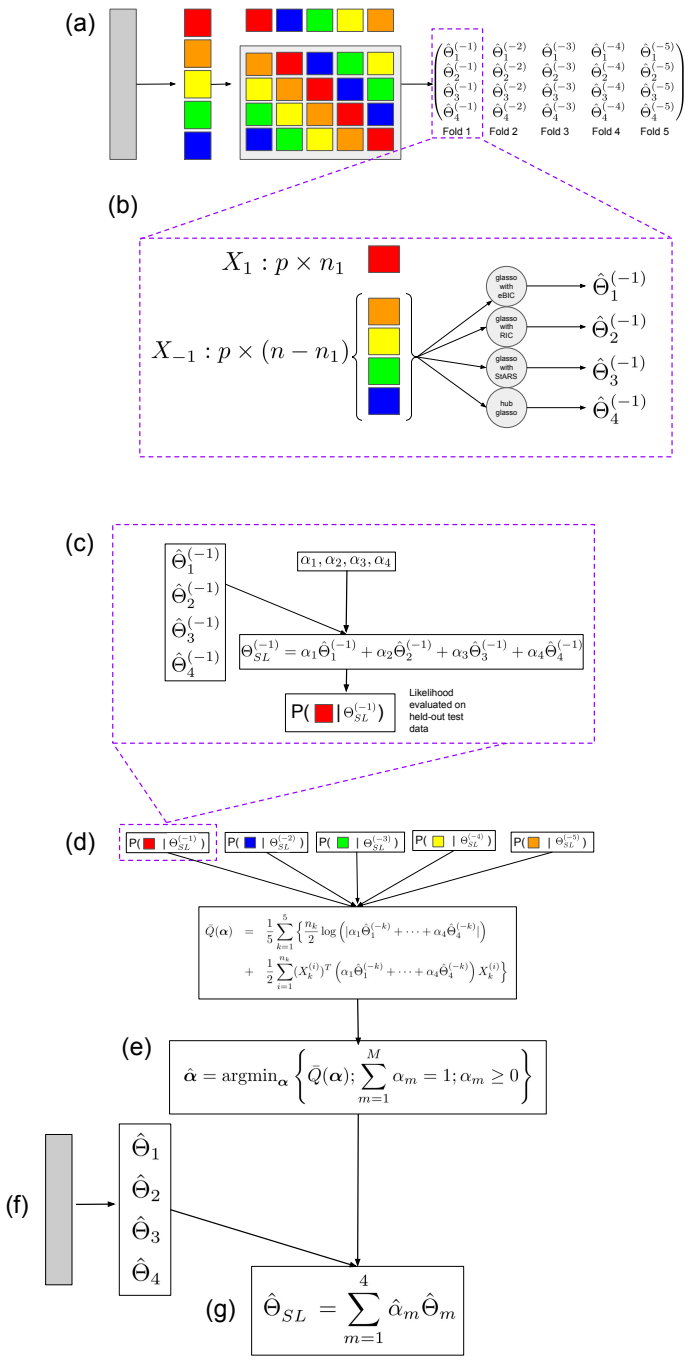


Figure 1: (a) Data are partitioned into five folds. Each fold is left out from the model fitting process in turn. (b) Every candidate model is fit on the training data in each fold. This generates an $(M = 4) \times (K = 5)$ array of estimated matrices. (c) For each held-out dataset k and coefficient set $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_4)$, the estimator $\hat{\Theta}^{(-k)}$ is calculated from the estimates obtained in (b). The likelihood of the estimator given the held-out data is then calculated. The process is repeated across all $K = 5$ folds and averaged to yield our loss function. (d) The loss function is minimized to yield the optimal coefficients $\hat{\boldsymbol{\alpha}}$, subject to the constraints of the convex combination. (e) The $M = 4$ methods are used to fit $\hat{\Theta}_1, \ldots, \hat{\Theta}_4$ on the whole dataset. (f) The final SpiderLearner estimator $\hat{\Theta}_{SL}$ is calculated as the convex combination of the coefficients selected in (d) with the models fit in (e).

Equation 12 is used as the loss function in the simulation and application sections below, while the original loss function (Equation 12) and the bounded loss function (Equation 17) are both provided as options in the SpiderLearner implementation. A comparison of performance of the original and bounded loss functions can be found in Supplementary Figure S1.

Simulation study

To assess the performance of the SpiderLearner algorithm, we conducted several simulation studies with varying sample sizes and numbers of predictors (Table 1a). In all simulations, a variety of network topologies and densities were considered (Table 1b). The simulation workflow (Figure 2) consisted of (i) designing gold-standard networks corresponding to each topology and density, (ii) assigning edge weights to the network based on an observed distribution of partial correlations from a real biological dataset and converting the associated weighted adjacency matrices to valid precision matrices , (iii) sampling multivariate normal data based on the precision matrices from (ii), (iv) using various methods, including our proposed ensemble method, to estimate the original network from the sampled data, and (v) comparing the estimated network to the original gold standard used to generate the data.

Simulation	n	p	q	$.9 * n/q$
A	10,000	50	1275	7.06
B	1,600	50	1275	1.13
C	100	50	1275	0.07
D	60	100	5050	0.01

(a) Simulation study dimensionality. n represents the sample size; p, the number of predictors in the network; q, the number of parameters that need to be estimated in the model; $.9 * n/q$: the sample size-to-parameter ratio in each training set in the 10-fold cross-validation.

Topology	Density	`igraph` Function	Simulated Density (A,B,C)	Simulated Density (D)
Random	Low	`sample_gnp`	0.053	0.061
Random	High	`sample_gnp`	0.219	0.194
Small World	Low	`sample_smallworld`	0.082	0.061
Small World	High	`sample_smallworld`	0.204	0.202
Scale-Free	Low	`sample_pa`	0.079	0.059
Scale-Free	High	`sample_pa`	0.192	0.191
Hub-and-Spoke	Low	`sample_pa`	0.079	0.059
Hub-and-Spoke	High	`sample_pa`	0.192	0.191

(b) Gold-standard networks were constructed using a variety of functions from the `igraph` package. Graph density is a function of the parameters used in each function as well as the number of predictors in the graph, and cannot be exactly specified. Parameters used in this study were chosen to achieve approximately 6 percent dense graphs in the low-density cases and 20 percent dense graphs in the high-density cases.

Table 1: Details of the simulation study designed to test the robustness of the SpiderLearner algorithm to differences in dimensionality and topology.

We explored four different network topologies in our simulations: random, small world, scale-free, and hub-and-spoke. Each topology has unique characteristics that may be relevant for biological data.

Random graph In an Erdös-Renyí/Gilbert random graph on p nodes, it is assumed that each of the $\binom{p}{2}$ possible edges is equally likely to exist, according to some fixed probability π (30; 31). A random graph can be constructed by sampling each edge independently from a $Bernoulli(\pi)$ distribution (31). The degree distribution of a random graph is approximately a Poisson distribution (32).

Small world graph A small world graph is characterized by a type of community structure that is absent in the random graph and which results in a shorter average path length (33). A small world graph on p nodes can be

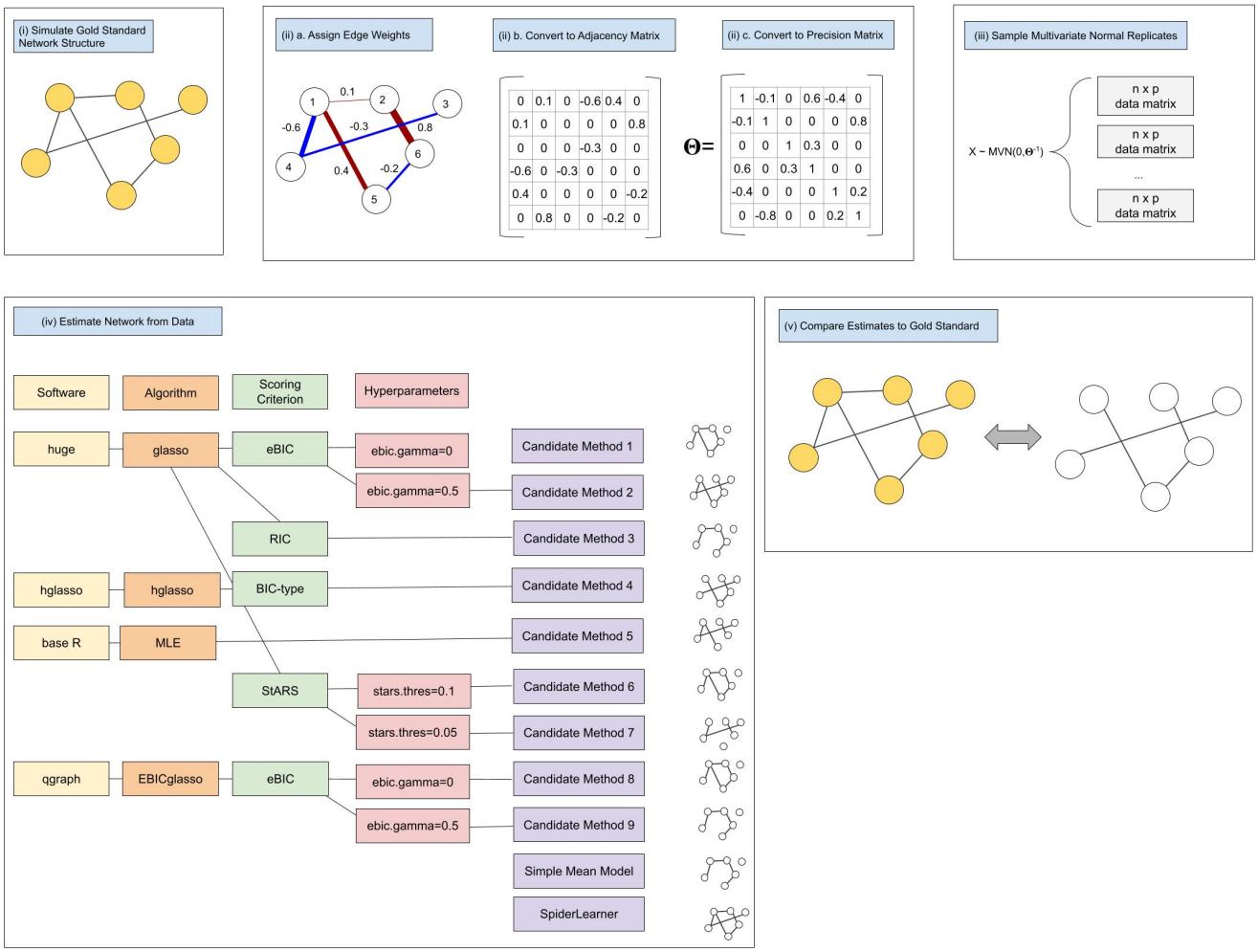


Figure 2: Simulation study workflow. In (i) we design gold-standard networks. In (ii), we assign edge weights to the gold standards by sampling from the distribution of partial correlations observed in the CATHGEN dataset and convert the corresponding adjacency matrices to precision matrices. In (iii), we sample multivariate normal data based on the precision matrices from (ii). In (iv), we estimate the networks from the sampled data. In (v), we compare the estimated network to the gold standard.

simulated by beginning with a circular lattice in which each node is connected to k of its neighbors. From this point, the graph is "rewired" by going around the lattice and reconnecting each edge to a different node at random with some fixed rewiring probability π. These random disruptions of the lattice structure create shortcuts across the graph, leading to the lower average path length for the graph as a whole.

Scale-free graph A scale-free graph is characterized by a power law degree distribution that can be simulated by considering an empty graph on p nodes and adding edges k-at-a-time following a growth and preferential attachment model, in which the probability that a particular node gets another edge added to it is proportional to how many edges it already has (34; 32). A log-log plot of the degree distribution (i.e., log frequency vs. log degree) in a scale-free graph is approximately a straight line; thus the model for generating this graph is referred to as linear preferential attachment.

Hub-and-spoke graph A hub-and-spoke graph arises in a similar way as the scale-free graph does, but the probability that an edge is added to a particular node is proportional to the k^{th} power of the degree of that node, for some $k > 1$ (superlinear preferential attachment) (32). This graph is characterized by hub nodes with very high degree and non-hub nodes with very low degree.

We explore these topologies along with graphs of different edge densities, where the edge density of a graph is defined as the number of edges divided by $\binom{p}{2}$, the number of possible edges on p nodes. For each of the four topologies, we simulated networks with two different density levels (low density: approximately 6 percent dense, and high density: approximately 20 percent dense). A visualization of these gold-standard networks shown in Figure 3.

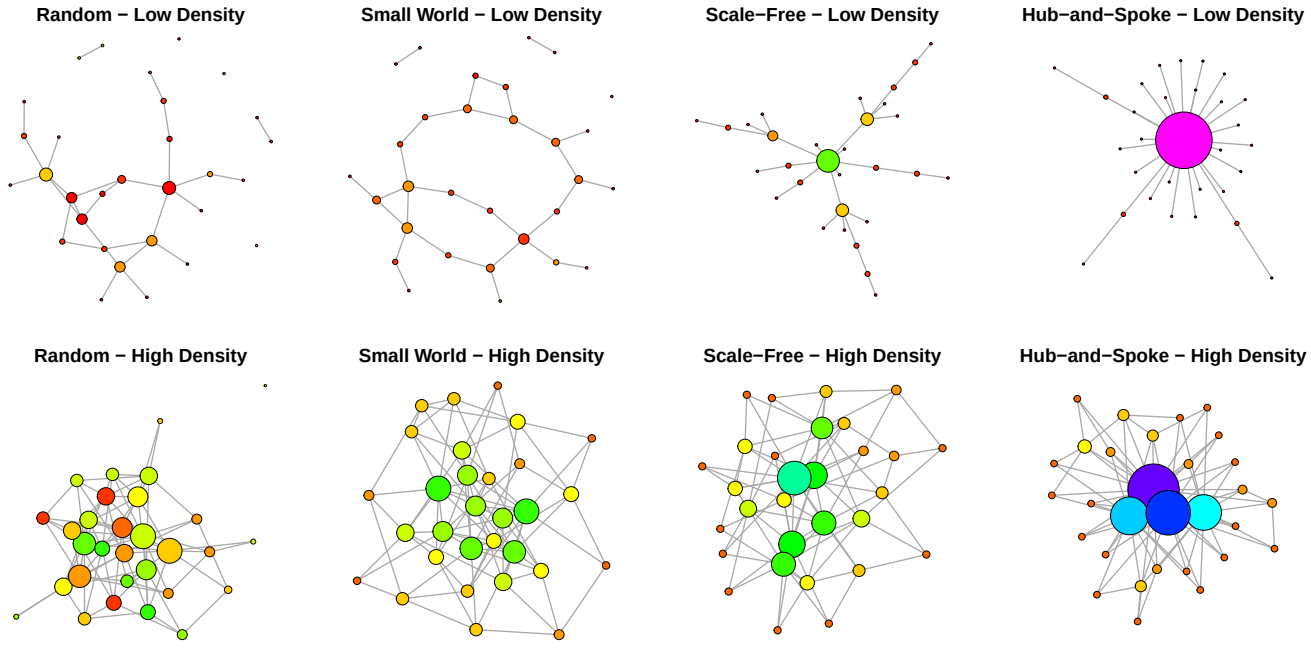


Figure 3: Sample graph topologies simulated using the `igraph` R package. Node size and color indicate degree; larger nodes have higher degree.

Designing gold-standard networks

The `igraph` package in R was used to simulate gold-standard networks (35). For random networks, the `sample_gnp` function was used (36). For small world networks, the `sample_smallworld` function was used (33). For scale-free and hub-and-spoke networks, the `sample_pa` function was used (Table 1b).

In an effort to create a realistic edge weight distribution from a biological distribution, we used metabolomics data from the CATHeterization GENetics (CATHGEN) biorepository as a starting point (37). The CATHGEN biorepository consists of data from a prospectively-collected clinical study of $\sim 10,000$ participants undergoing cardiac catheterization with scheduled annual followup at Duke University Hospital; further details of the study population have previously been published in (37). Measurements of 407 metabolites were available for 136 of these participants, including 68 participants with incident coronary artery disease (CAD) and an equal number of participants without CAD during follow up; further description of this metabolomics study can be found in (38). We used the graphical lasso with the eBIC scoring criterion (hyperparameter $\gamma = 0$) to estimate a GGM for this dataset. The resulting distribution of the nonzero partial correlations was skewed right with several high outliers (Supplementary Figure S2). We used the histogram of the edge weights as a discrete probability distribution from which to sample edge weights for our simulated networks (bin size 0.01, range -0.32 to 0.76).

To use the weighted adjacency matrix to sample from the multivariate normal distribution, we begin by obtaining a valid precision matrix using Equation 2. For simplicity, we assume $\theta_{ii} = \theta_{jj} = 1$, which gives the (i, j) entry of the precision matrix as:

$$\theta_{ij} = -\rho_{ij|X_{-i,-j}} \tag{18}$$

We use this relationship to determine the full precision matrix Θ. Although there is no guarantee that a matrix generated with this approach will be positive definite, we observed positive definite matrices for most of the simulations in this paper. In cases where matrices were not positive definite, we performed a "boosting" step involving adding a small multiple of the magnitude of the minimum eigenvalue to the diagonal of the matrix. For a $p \times p$ precision matrix $\mathbf{A}$ with minimum eigenvalue λ^A_{min}, this correction is:

$$\mathbf{A}' = \mathbf{A} + 1.01 * |\lambda^A_{min}| * \mathbf{I}_p \tag{19}$$

where $\mathbf{I}_p$ is the $p \times p$ identity matrix. This is similar to the approach taken by Tan et. al. in (39).

Sampling, estimation, dimensionality, and candidate methods

To sample network data from the gold-standard networks, we inverted each estimated precision matrix Θ to find the corresponding covariance matrix Σ,then simulated a sample of size n by drawing $X_1, \ldots, X_n \sim MVN(\mathbf{0}, \Sigma)$. Finally, we estimated precision matrices from this sample in three ways: (i) by applying candidate methods individually (ii) by

using a simple mean ensemble model in which each candidate is weighted equally, and (iii) by using the SpiderLearner (Figure 2).

The dimension of a GGM is typically described in terms of the number of samples n and the number of predictors p included in the model. Importantly, and in contrast to many regression approaches, p is not the number of parameters in the model: the precision matrix corresponding to the GGM has $q = p * (p-1)/2 + p$ unique entries that need to be estimated. Because of the quadratic relationship between the number of predictors and the number of parameters to be estimated, dimensionality becomes a major factor in estimation even if a GGM does not include very many predictors. Dimensionalities simulated in this study are shown in Table 1a.

Nine different candidate methods were considered for input to the ensemble algorithm (Figure 2). Candidate Methods 1,2,3,6, and 7 use the `huge` and `huge.select` functions from the R package `huge` with the `glasso` method, which corresponds to the original graphical lasso (7; 3; 6). The difference between these methods is the choice of scoring criterion used in the `huge.select` function to select the tuning parameter (λ in Equation 3). The first criterion is the extended Bayesian information criterion (eBIC), which optimizes a BIC-type quantity tuned by a hyperparameter γ, where $\gamma = 0$ corresponds to a standard BIC measure and $\gamma = 0.5$ is a typical default value for graphical modeling (12; 40). Candidate Methods 1 and 2 apply this criterion with $\gamma = 0$ and $\gamma = 0.5$, respectively. Candidate Method 3 applies a criterion called the rotation information criterion (RIC), which is based on a permutation strategy that generates a null distribution for comparison (12; 7). Candidate Methods 6 and 7 use a criterion called the stability approach to regularization selection (StARS), which is a sub-sampling based approach (12; 41). One of several hyperparameters that can be selected using StARS is `stars.thres`, which relates to the amount of variability that is tolerated across the subsamples (41). Candidate 6 applies the StARS criterion with `stars.thres` = 0.05 and Candidate Method 7 applies it it with `stars.thres` = 0.1 (the default). Candidate Method 4 is the hub graphical lasso, which is an extension of the original graphical lasso that can effectively model hub structures in networks and is implemented in the `hglasso` R package (39). Candidate Method 5 is the MLE, i.e., inverse of the sample covariance as computed with the `cov` function in base R. Candidate Methods 8 and 9 are similar to Candidate Methods 1 and 2; they also use the original graphical lasso along with an eBIC scoring criterion, but are implemented in the `qgraph` R package (42; 43). A difference between the `qgraph` implementation and the `huge` implementation is in the default range of tuning parameters λ considered. Let λ^* be the smallest value of λ that creates an empty graph; `huge` uses a logarithmic sequence of ten candidate λ values between $0.1\lambda^*$ and λ^*, while `qgraph` uses a larger logarithmic sequence of length 100 between $0.01\lambda^*$ and λ^* (7; 43).

We use the following shorthand for these nine methods in the remainder of this paper:

- Candidate Method 1: glasso-ebic-0
- Candidate Method 2: glasso-ebic-0.5
- Candidate Method 3: glasso-ric
- Candidate Method 4: hglasso

- Candidate Method 5: MLE
- Candidate Method 6: glasso-stars-0.05
- Candidate Method 7: glasso-stars-0.1
- Candidate Method 8: qgraph-ebic-0
- Candidate Method 9: qgraph-ebic-0.5

While Candidate Method 5 is typically well-defined in Simulations A-C (barring multicollinearity), it is not in Simulation D, where $n < p$. Therefore, Simulation D excludes Candidate Method 5.

Assessing estimation performance

Once a precision matrix is estimated, we compare it to the original, data-generating, gold-standard precision matrix in order to assess performance. We begin by introducing notation that will be helpful in defining our performance metrics. Let $\hat{\Theta}$ be an estimate of the true $p \times p$ precision matrix Θ, and let $\hat{\theta}_{ij}$ and θ_{ij} represent the corresponding elements of each. We define the **error matrix** Δ as $\hat{\Theta} - \Theta$, and refer to its i, j^{th} element as:

$$\delta_{ij} = \hat{\theta}_{ij} - \theta_{ij} \tag{20}$$

Note that, although the true precision matrix is symmetric, the estimated matrix may not be: a notable example of possible asymmetry is in the graphical lasso algorithm (44). Therefore, we consider every element of the error matrix Δ rather than just upper or lower triangular components when assessing estimation performance.

One area of interest is to assess error in the estimated edge weights in the GGM. Because these edge weights follow directly from the estimated precision matrix, we begin by focusing our efforts on quantifying error in the precision matrix itself. The first metric we use is the based on the size of Δ as assessed by the Frobenius norm:

$$||\Delta||_F = \sqrt{\sum_{i=1}^{p}\sum_{j=1}^{p}\delta_{ij}^2} \tag{21}$$

To obtain a quantity that can be compared across topologies, we scale $||\Delta||_F$ by the Frobenius norm of the true precision matrix, $||\Theta||_F$, defining the **relative Frobenius norm (RFN)** as

$$RFN = \frac{||\Delta||_F}{||\Theta||_F} \tag{22}$$

We are interested in the generalizability of the SpiderLearner to independent datasets. For this purpose, we assessed the **out-of-sample log likelihood** of each estimated precision matrix on a new, independent sample of the same size generated from the same gold-standard precision matrix.

Next, we considered bias and MSE for each matrix entry. For R simulated replicates of multivariate normal data sampled according to the same gold-standard precision matrix, the **replicate-estimated bias** of element $\hat{\theta}_{ij}$ is given by:

$$\text{Bias}_{ij} = \frac{1}{R}\sum_{r=1}^{R}\delta_{ij}^{(r)} \tag{23}$$

Similarly, the **replicate-estimated mean squared error** of an element δ_{ij} is given by:

$$\text{MSE}_{ij} = \frac{1}{R}\sum_{r=1}^{R}(\delta_{ij}^{(r)})^2 \tag{24}$$

Because most of the candidate models in the ensemble learner are shrinkage methods, we expect that MSE will vary based on the size of each element. Moreover, some methods penalize the diagonal of the precision matrix while others do not. We therefore summarized performance by investigating this element-wise bias and MSE in six categories: (i) the zero elements of the gold-standard matrix, (ii-iv) small, medium, and large entries, corresponding to the bottom quartile, middle 50%, and top quartile of the off-diagonal non-zero matrix elements of the gold-standard matrix, and (v) the diagonal elements of the gold-standard matrix.

In some cases, it may be of interest to assess the performance of a method at estimating the edge set of the GGM, rather than its edge weights. In this case, GGM estimation is essentially a classification problem, where each possible edge (i, j) is classified as either included in the network or excluded from the network. For this purpose, it is necessary to classify edges in some way. In Simulations A-C, where the MLE is included as a candidate learner, it is likely that the estimated GGM will be completely connected as the MLE will not contain entries that are exactly zero. For these three simulations, we therefore construct a sparser graph by thresholding based on the significance of the Fisher-transformed partial correlation coefficient (see Supplement for details). We control the FWER at level $\alpha = 0.05$ with a Bonferroni correction (45).

In Simulation D, the Fisher-transformed partial correlation is not well-defined (see Supplement). We therefore classify any non-zero partial correlation coefficient as an edge. Because the MLE is not included in Simulation D, this classification yields a graph that is not completely connected. This methodology is described in detail in the Supplement.

As an additional measure, we calculated the **matrix RV coefficient**, an analogue of a correlation coefficient, between estimated and gold-standard matrices, as implemented in the R package `MatrixCorrelation` (46; 47).

Results

We conducted 100 iterations for each of the eight network topologies in each of Simulations A-D. Results for Simulation A and Simulation D are presented here; results for Simulation B and Simulation C are presented in the Supplement. To provide some perspective of computation times, a table of runtimes for various values of n and p is available in Supplementary Table S1.

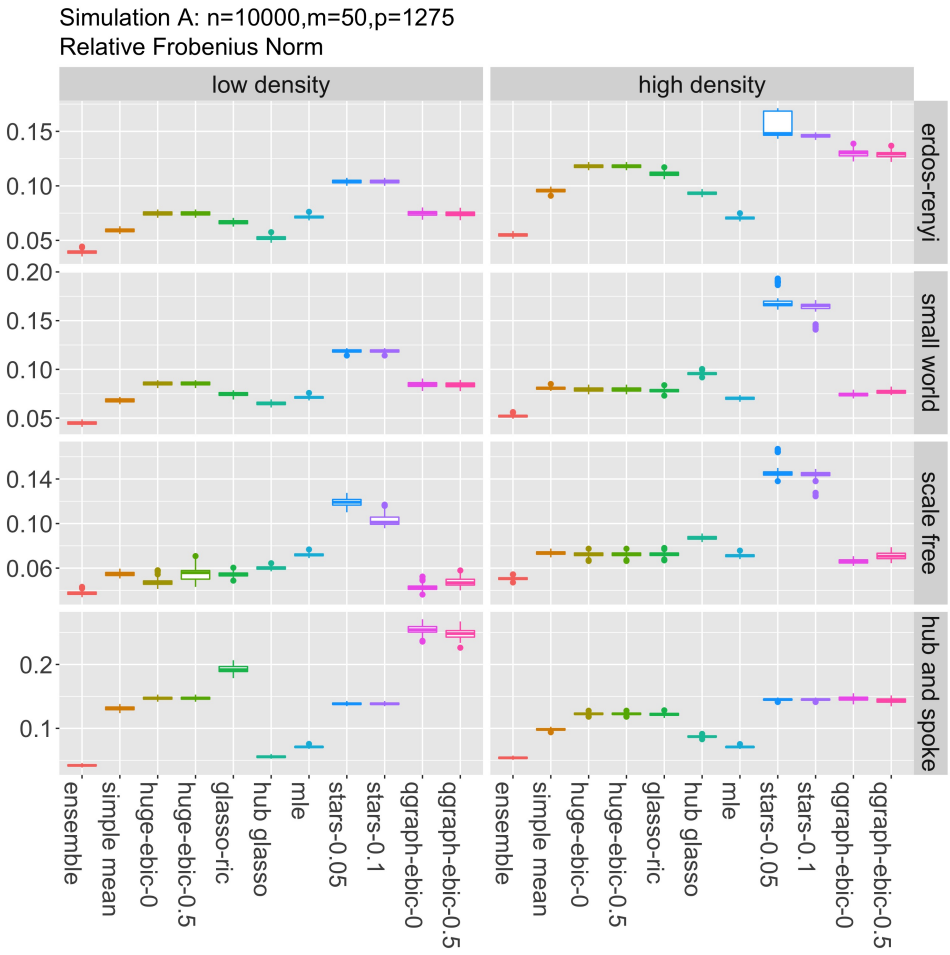
Simulation A: Sample size >> number of features, parameters estimated ($n >> p, q$)

Ensemble weights for Simulation A are shown in Table 2a. The SpiderLearner algorithm selected at least three different methods to have nonzero weights for each topology, demonstrating that combining multiple candidate algorithms is indeed important from a likelihood-based loss perspective. For every topology, `qgraph-ebic-0` and the inverse sample covariance (i.e., MLE) were included in the combination, although the weights varied broadly by topology, with `qgraph-ebic-0` weights ranging from 0.03 for the low-density hub-and-spoke topology to 0.42 for the low-density scale-free topology and the MLE weights ranging from 0.28 for the low-density random graph topology to 0.59 for the high-density random graph topology. The `hub glasso` was included in seven out of eight topologies (excluding the low-density scale-free topology), again with broadly varying weights (0.10-0.63). The `glasso-ebic-0` was selected for minor contributions in the low-density scale-free case(0.33) and the high-density scale-free case (0.08). The `glasso-ric`,`glasso-stars-0.05` and `glasso-stars-0.1` methods were weighted zero for all topologies.

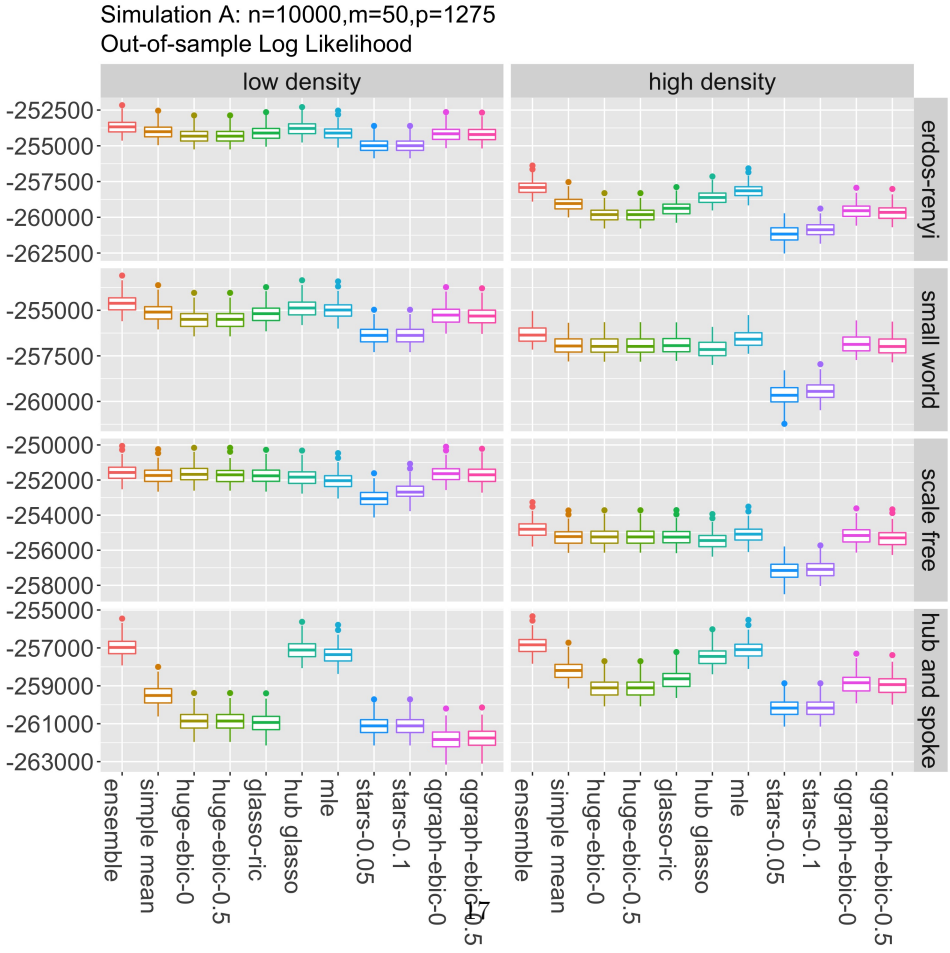
Results for the relative Frobenius norm of the error matrix are shown in Figure 4a. It can be easily seen that the performance of each method varied highly according to this metric, emphasizing the importance of our approach. The SpiderLearner performed better than the individual candidates and better than the simple mean ensemble model across all settings considered. The performance as assessed by out-of-sample log likelihood can be seen in Figure 4b. Again, the SpiderLearner performed well; we note that variability of the out-of-sample log likelihood was not as high as that of the RFN.

The element-wise bias and MSE for the five entry categories (zero, small, medium, large, and diagonal) are shown in Supplementary Figure S4. For the SpiderLearner as well as all the candidate methods, bias varied by entry category. The MLE showed the smallest bias in each case, which is logical given that it was the only non-shrinkage method employed. The SpiderLearner performed better than or comparably to the remainder of the algorithms in terms of the magnitude of bias, while having the added benefit of smaller variability of bias for most elements. The exception was for the true zero elements, in which the SpiderLearner incorporated the MLE and had a higher variability accordingly. A similar pattern was observed for MSE.

Sensitivity and specificity of each method in Simulation A are shown in Supplementary Table S2 and S3, respectively. The SpiderLearner was more sensitive than some candidate methods and less sensitive than others; it was more sensitive than the simple mean in every case. It had perfect specificity (as did most candidate methods), selecting no false positives. This observation may be due to the conservative nature of our threshold, which is based on a Bonferroni



(a) Relative Frobenius norm, Simulation A.



	glasso - ebic - 0	glasso - ebic - 0.5	glasso - ric	hglasso	mle	glasso - stars - 0.05	glasso - stars - 0.1	qgraph - ebic - 0	qgraph - ebic - 0.5
Erdos-Renyi Low	0	0	0	0.57	0.28	0	0	0.15	0
Erdos-Renyi High	0	0	0	0.33	0.59	0	0	0.08	0
Small World Low	0	0	0	0.46	0.39	0	0	0.15	0
Small World High	0	0	0	0.19	0.55	0	0	0.26	0
Scale Free Low	0.33	0	0	0	0.25	0	0	0.42	0
Scale Free High	0.08	0.08	0	0.1	0.51	0	0	0.23	0
Hub-and-Spoke Low	0	0	0	0.63	0.34	0	0	0.03	0
Hub-and-Spoke High	0	0	0	0.36	0.57	0	0	0.06	0

(a) Simulation A

	glasso - ebic - 0	glasso - ebic - 0.5	glasso - ric	hglasso	glasso - stars - 0.05	glasso - stars - 0.1	qgraph - ebic - 0	qgraph - ebic - 0.5
Erdos-Renyi Low	0	0	0	0.02	0.07	0.21	0.68	0.01
Erdos-Renyi High	0	0	0	0.06	0.09	0.61	0.23	0
Small World Low	0.01	0.01	0.01	0.02	0.03	0.07	0.37	0.48
Small World High	0	0	0	0.08	0.21	0.61	0.09	0
Scale Free Low	0	0	0	0.03	0.05	0.23	0.65	0.03
Scale Free High	0	0	0	0.09	0.14	0.76	0.01	0
Hub-and-Spoke Low	0	0	0	0.02	0.04	0.21	0.57	0.15
Hub-and-Spoke High	0	0	0	0.1	0.15	0.74	0.01	0

(b) Simulation D

Table 2: Average weight for each method as selected by SpiderLearner in N=100 simulations.

correction (see Supplement).

Simulation D: Sample size $<$ number of features $<<$ number of parameters estimated ($n < p << q$)

Ensemble weights for Simulation D are shown in Table 2b. The SpiderLearner algorithm selected at least four of the candidate methods in every case. Interestingly, the `glasso-ebic-0`, `glasso-ebic-0.5`, and `glasso-ric` contributed the least to the ensemble in Simulation D after being important players in Simulation A. On the other hand, `glasso-stars-0.05` and `glasso-stars-0.1` both contributed to the ensemble in all eight cases in Simulation D, but they were not selected in any case in Simulation A. The hub graphical lasso was a highly-weighted candidate in most cases in Simulation A, but had very low weights in Simulation D. These observations are further evidence of the important of considering multiple methods when estimating a GGM: the performance (in the log-likelihood sense) of estimates from different methods varies broadly based on the characteristics of the true underlying network.

Results for the RFN in Simulation D are shown in Figure 4c. We generally saw the SpiderLearner performing comparably to the `qgraph-ebic-0` candidate method and `qgraph-ebic-0.5`, two methods which were typically highly weighted in the ensemble (Table 2b). The out-of-sample log likelihood performance is shown in Figure 4d. The SpiderLearner again performed well when compared to the remainder of the methods. The hub graphical lasso had notably lower out-of-sample log likelihood than the other candidates, suggesting overfitting in this setting.

Bias and RMSE for Simulation D are shown in Supplementary Figure S10. In Simulation D, the hub graphical lasso clearly outperformed all the other methods in terms of bias, while suffering a large MSE (i.e., high variance). The SpiderLearner was able to detect this tradeoff and avoid excessive variance by assigning a low weight to the hub graphical lasso. Aside from the hub graphical lasso, bias and MSE were comparable across the SpiderLearner, the simple mean, and the remaining candidate methods for the zero, small, medium, and large entries. Bias differed for the diagonal entries as some of the candidate methods applied a shrinkage penalty to the diagonal, while some did not.

In Simulation D, candidate methods either had a moderate sensitivity or a very low sensitivity; the SpiderLearner fell into the moderate category, with sensitivity around 0.6. (Supplementary Table S10). Many methods with low sensitivity selected empty graphs, possibly due to the small sample size relative to the number of predictors in this simulation setting. Specificity was similarly bimodal, with the SpiderLearner, the simple mean, and the hub graphical lasso having a specificity around 0.45, while other methods had a specificity of near 1 (Supplementary Table S11).

The MLE as the estimator for the precision matrix

In simulation settings A,B, and C, the sample size n is larger than the number of predictors p in the model, meaning that the sample covariance matrix is non-singular, except in the case of multicollinearity. The sample covariance matrix is the MLE for the population covariance matrix, and because inversion of a non-singular matrix is a continuous function, the inverted sample covariance matrix is the MLE for the population precision matrix (48; 49). Notably, that the likelihood-based SpiderLearner model selects models other than the MLE, and that other individual regularized algorithms perform better than the MLE according to the relative Frobenius norm, matrix RV coefficient, and out-of-sample likelihood. We hypothesized that this phenomenon was related to the sparsity of the underlying network. To investigate, we ran the SpiderLearner algorithm on an Erdös-Renyí random graph with a variety of densities (0.05, 0.1, 0.25, 0.5, 0.75, 1) with 30 iterations for each density. As hypothesized, the weight of the MLE in the ensemble model increases with the density of the graph, as shown in Figure 5. These results suggest that even though the ensemble loss function does not incorporate a shrinkage penalty, it is still advantageous from the likelihood-based perspective to shrink estimates of small precision matrix entries to zero in the case where the population precision matrix is sparse. The takeaway is that shrinkage methods can improve out-of-sample performance even in low-dimensional cases, which is consistent with results observed in the original LASSO publication (50).

Choice of K

A practical question in this methodology is how to select K in the K-fold cross-validation. Higher values of K give more training data, meaning the estimates of the candidate precision matrices $\Theta_1, \ldots, \Theta_M$ are more accurate and more precise; however, less data are available to estimate the out-of-sample log likelihood on the left-out test data, meaning estimates of $\boldsymbol{\alpha}$ will suffer higher bias and variance. Lower values of K give less training data and more out-of-sample data, but it is not immediately clear that this causes the reverse problem: quality of estimates of $\boldsymbol{\alpha}$ depend both on the amount of out-of-sample data as well as the quality of the estimates of $\Theta_1, \ldots, \Theta_M$.

It is apparent that there is a complex "Θ-α tradeoff" underlying our method, making it challenging to recommend a particular choice of K without further investigation. For these reasons, we conducted a simple simulation study to assess the impact of the use of different values of K. We used the high-density Erdös-Renyí random graph topology as a gold standard network, generated 100 samples of size $n = 150$ on $p = 50$ predictors ($q = 1275$ parameters to be estimated), and ran the SpiderLearner algorithm for $K \in \{2, 5, 10, 15, 20, 30\}$. We then calculated (i) the element-wise standard deviation of the estimated precision matrix $\hat{\Theta}_{SL}$, (ii) the element-wise bias of $\hat{\Theta}_{SL}$, and (iii) the variability of the selected coefficients $\hat{\alpha}_1, \ldots, \hat{\alpha}_M$ for each value of K. Because the library includes shrinkage methods, we calculated summary measures for (i) and (ii) in four categories of the true matrix: zeros, bottom 10 percent of non-zero entries, middle 80 percent of non-zero entries, and top 10 percent of non-zero entries. Because the true matrix is symmetric, we only assessed diagonal and lower triangular elements.

Supplementary Figure S11 shows that the element-wise standard deviation of $\hat{\Theta}_{SL}$ increases slightly with increases

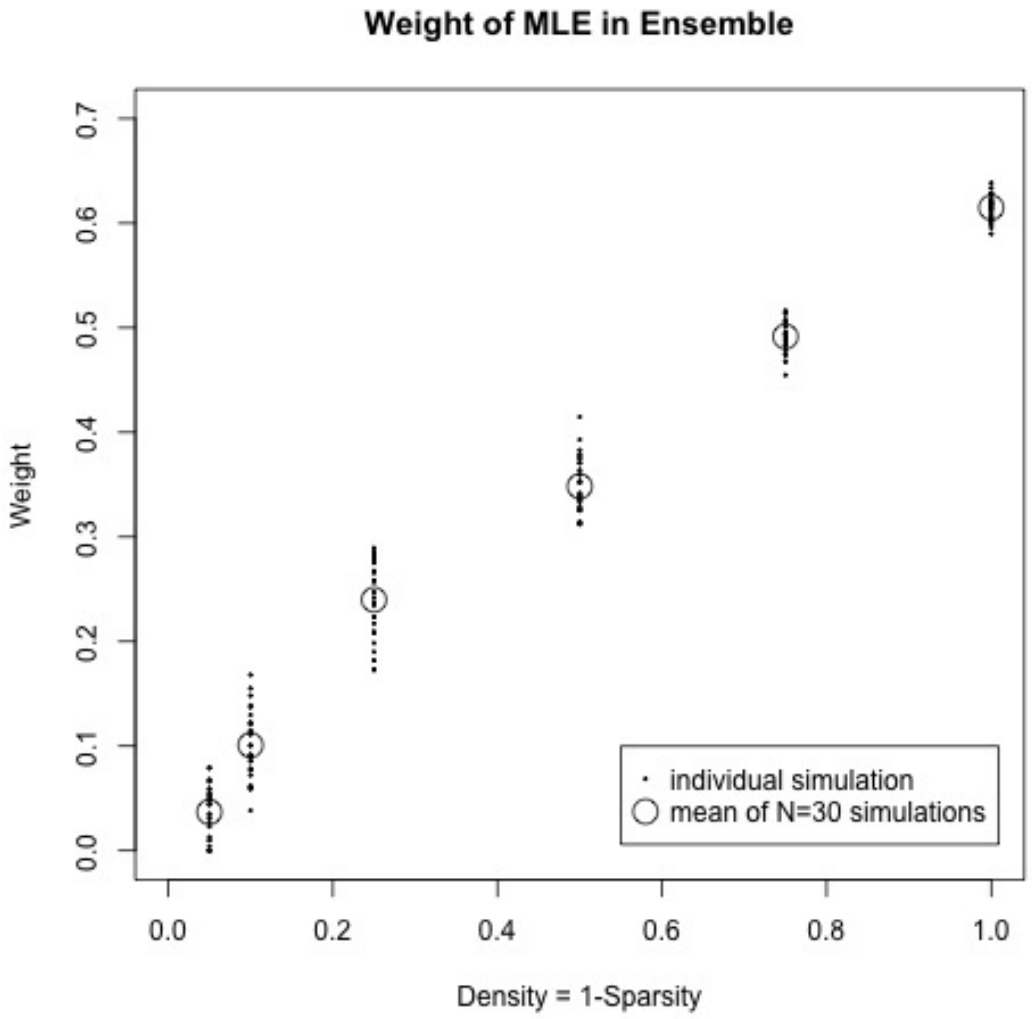


Figure 5: Using a random graph topology with six different densities (0.05, 0.1, 0.25, 0.5, 0.75, and 1), we explored the relationship of the weight of the MLE in the ensemble model with the graph density. In sparse graphs, the MLE is not weighted heavily by the algorithm; as density increases, the MLE begins to dominate the contribution to the convex combination.

in K, but that most of the change happens when moving from $K = 2$ to $K = 5$ and from $K = 5$ to $K = 10$. Supplementary Figure S12 shows that entry-wise bias decreases substantially as K increases for medium and large entries; although it slightly increases as K increases for zero entries and for small entries, the magnitude of these increases is small compared to the decrease in bias for the medium and large entries. Supplementary Figure S13 shows that the variability of the weights $\hat{\alpha}_1, \ldots, \hat{\alpha}_9$ is similar across all considered values of $K \geq 5$. These results lead us to suggest that $K = 10$ is a good choice, with $K = 5$ as an option for large datasets if computing time is a limitation.

Selection of candidate GGM library

Existing Super Learner literature suggests that a broad and varied library of candidate learners is beneficial, and that overfitting or highly variable estimates are typically not observed consequences of a large library size, although cross-validating the Super Learner itself is recommended as a best practice (15; 24; 51). Practical limitations to library size include computation time and interpretability. We investigated the sensitivity of model results to the library size and content in 100 simulated datasets of size $n = 1000, p = 50$ using the high-density scale-free graph topology. We used three different libraries: (i) a small baseline library consisting of the hub graphical lasso and the MLE, chosen due to the ability of the hub graphical lasso to model the scale-free topology and the generally favorable properties of the MLE, (ii) a medium library consisting of the small library along with the `huge-ebic-glasso` method with $\gamma = 0$ and $\gamma = 0.5$, and (iii) a large library consisting of the nine methods used in Simulations A-C. Our results indicate that while the large library provides the best fit, the results from the medium and small library do not differ substantially as a whole in this case (Supplementary Figure S14). In addition to this simulation setting, we further explored the sensitivity of the estimated network to the library selection in a real data example, in which we observed that some libraries yield similar results while others differ (Supplementary Table S12, Supplementary Figure S16).

Application: Ovarian cancer risk modeling

Datasets

To demonstrate the application of our method to real data, we used sixteen ovarian cancer gene expression datasets from the Curated Ovarian Cancer collection of Ganzfried et. al. (20). One of the datasets was used to train a SpiderLearner model and develop a network-based risk score from the resulting network; the other fifteen were used as independent validation datasets to evaluate the risk score performance (Figure 6). The training dataset consists of 260 late-stage ovarian cancer patients with gene expression data for 20106 genes, obtained via microarray experiments by Yoshihara et al. (19) ("Yoshihara dataset"). Characteristics of the Yoshihara dataset have been previously described in (19), where it is referred to as Japanese data set A. Briefly, the study conducted by Yoshihara et al. included participants with advanced stage high-grade serous ovarian cancer who underwent debulking surgery followed by chemotherapy, with followup for up to ten years. Yoshihara et al. assessed overall survival was assessed as the time from the primary surgery to death due to ovarian cancer. 131 of the 260 patients were living at the end of the study; we treated these patients' outcomes as right-censored. Basic characteristics of all sixteen datasets are shown in Table 3. For details regarding the 15 validation datasets, we refer the reader to the original publications, also shown in Table 3. In the application below, we used the nine candidate GGM estimation methods described in Figure 2 as the SpiderLearner library, with $K = 10$-fold cross-validation. All data are publicly available through the R package `curatedOvarianData` (via Bioconductor), and code to reproduce the application workflow is available at `https://github.com/katehoffshutta/SpiderLearnerWorkflow`.

Workflow and results

In (19), Yoshihara et al. present a 126-gene signature of high-risk ovarian cancer based on overall survival, defined as time from primary surgery to death or loss-to-followup. To investigate the relationships between the genes in this signature, we used SpiderLearner to estimate a GGM for this study setting. We extracted 116 of the genes presented in (19) from an example dataset in the `curatedOvarianData` R package. The weights selected by SpiderLearner were 0.69 for `hglasso`, 0.13 for `huge-ebic-0`, 0.12 for `qgraph-ebic-0`, 0.05 for the MLE, and zero for the remainder of the candidate algorithms.

Community detection is a useful way to identify clusters in graphical models. We applied the `cluster_walktrap` community detection algorithm as implemented in the `igraph` R package to detect communities in the SpiderLearner-estimated GGM as well as the GGMs estimated by the nine candidate algorithms and the simple mean (65; 35). The `cluster_walktrap` algorithm requires the choice of a step size for the random walk. For each network, we selected the step size between 1 and 10 that maximized the overall modularity of the network. Estimates for the nine candidate methods are shown in Figure 7a. The community structure varies notably across methods, further motivating the use of our ensemble method.

To derive biological insight from the detected communities, we developed a network-based risk score utilizing

Dataset	Platform ID	N	Age: Mean(SD) *missing*	Tumor Stage (% < 4) *missing*	Summary Stage (% Late) *missing*	Summary Grade High (%) *missing*	Validated in KM Models	Validated in CoxPH Models
GSE32062.GPL6480 (19)	hgug4112a	260	-	78	100	50	-	-
E.MTAB.386 (52)	illuminaHumanv2	129	60.71(14.24)	85	99	1	no	yes
GSE13876 (53)	OperonHumanV3	157	57.95(12.39)	-	100	54 *13*	yes	yes
GSE14764 (54)	hgu133a	80	-	98	89	68	yes	yes
GSE17260 (55)	hgug4112a	110	-	80	100	39	no	no
GSE18520 (56)	hgu133plus2	63	-	100 *10*	84 *10*	84 *10*	no	no
GSE19829.GPL570 (57)	hgu133plus2	28	-	-	-	-	yes	no
GSE19829.GPL8300 (57)	hgu95av2	42	-	-	-	-	no	no
GSE26712 (58)	hgu133a	195	61.54(11.86) *13*	80 *13*	95 *10*	95 *10*	no	no
GSE30009 (59)	NA	103	62.45(11.14)	80	100	89 *2*	no	no
GSE30161 (60)	hgu133plus2	58	62.57(10.61)	91	100	57 *4*	no	yes
GSE32063 (19)	hgug4112a	40	-	78	100	42	no	no
GSE9891 (61)	hgu133plus2	285	59.62(10.59) *3*	92 *3*	84 *3*	57 *6*	yes	yes
PMID17290060 (62)	hgu133a	117	-	85 *1*	98 *1*	49 *3*	yes	no
PMID19318476 (63)	hgu133a	42	61.46(10.61) *1*	76 *1*	93 *1*	57 *1*	yes	no
TCGA (64)	hthgu133a	578	59.7(11.56) *10*	85 *15*	90 *15*	83 *23*	no	no

Table 3: Basic characteristics and references for the 16 ovarian cancer datasets used in the SpiderLearner application.

topological characteristics of the estimated GGM to identify a set of genes with which to predict overall survival. We closely followed the approach of (19) in developing their ovarian cancer prognostic index. (19) began by using a penalized Cox proportional hazards (Cox PH) model to obtain regression coefficients for each of the 126 genes. Next, the authors calculated a prognostic index as follows:

$$\text{Prognostic score} = \sum_{i=1}^{126} \beta_i X_i \tag{25}$$

where β_i was the regression coefficient for gene i in the penalized Cox PH model and X_i was its centered and standardized gene expression value. Finally, (19) determined the optimal threshold value of their prognostic index by (i) assigning patients to a high-risk or low-risk group based on a proposed threshold, (ii) calculating the p-value of a log-rank test for difference in overall survival between the high-risk and low-risk group, and (iii) repeating this process for a number of thresholds and finding a threshold that minimized the p-value in (ii).

The workflow that we applied is in the same spirit, and is shown in Figure 6a. Rather than using all 126 genes to produce the score, we aimed to find more a more parsimonious score by leveraging the network structure of the Yoshihara dataset to select a subset of relevant genes. For each candidate approach, the simple mean, and the SpiderLearner, we began by identifying the gene in each community with the highest hub score by applying the `hub_score` function of the `igraph` R package to the adjacency matrix of the estimated GGM (35; 66). Hub scores reflect how influential nodes are based on the eigendecomposition of the weighted adjacency matrix of a graph, with a higher hub score corresponding to more influence (66). Earlier work in bipartite networks of SNPs and genes has demonstrated that hubs within communities are enriched for disease-associated SNPs (67). We hypothesized that local hubs in GGM communities might have similar functional leverage and therefore be useful predictors of ovarian cancer outcomes.

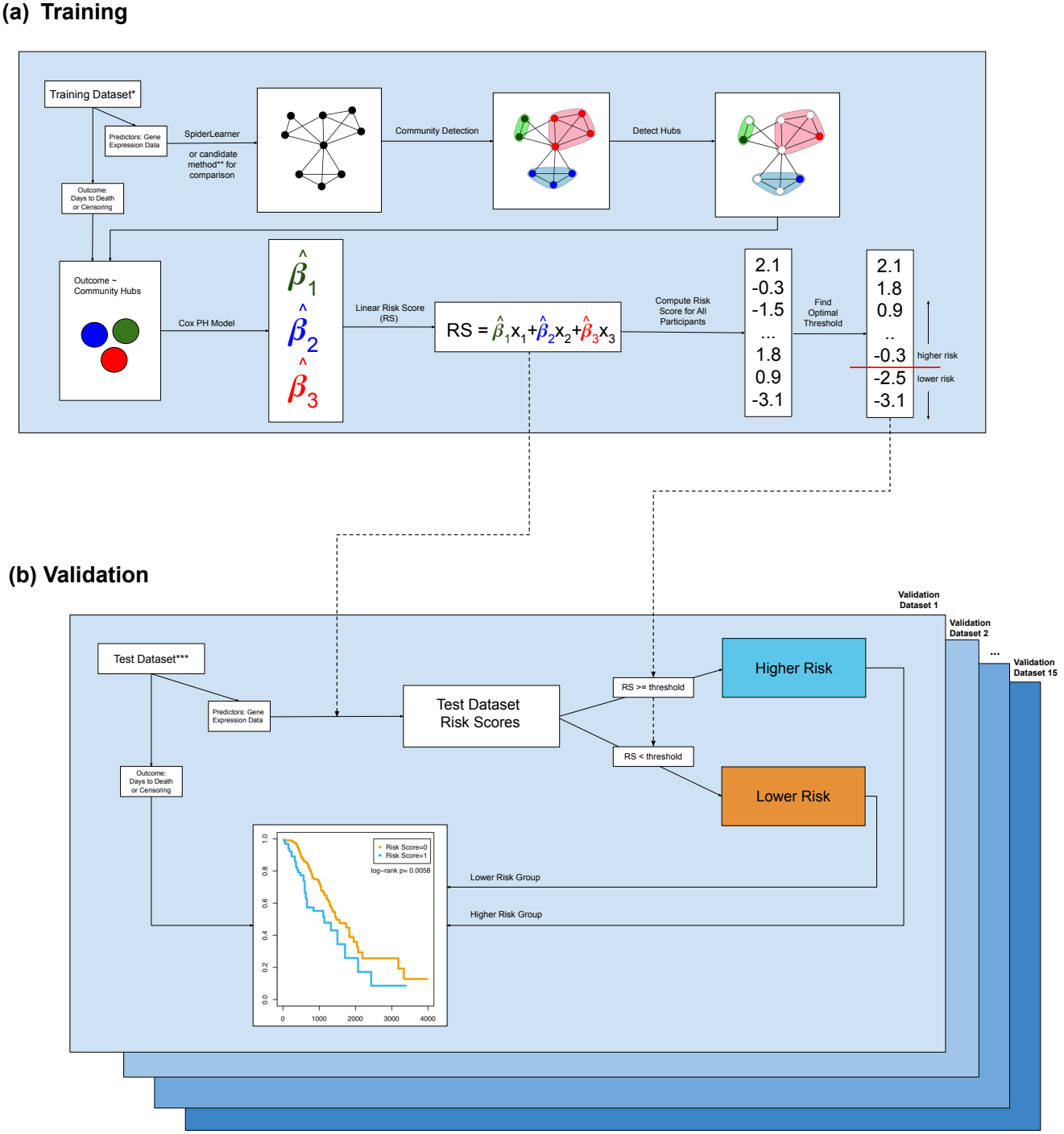
To develop the risk score, we fit Cox PH models regressing days to death on the local hubs in each network using the `survival` package in R (68). For GGM estimation method m with local hubs $x_1, \ldots, x_p$, we denote the corresponding Cox PH model coefficients as $\beta_1^{(m)}, \ldots, \beta_p^{(m)}$. Following the development of the prognostic index in (19), the risk score for patient i according to method m was calculated as

$$S_i^{(m)} = \sum_{j=1}^{p} \beta_j^{(m)} x_{ij} \tag{26}$$

where x_{ij} is the centered, standardized expression level of gene j for person i.

We next mapped the score in Equation 26 to a binary indicator of high risk or low risk by establishing a threshold point. As in (19), we selected an optimal threshold for this score by testing a grid of threshold values and selecting the value attaining largest separation between the estimated Kaplan-Meier survival curves of the high-risk and low-risk groups in the Yoshihara dataset, as measured by the lowest p-value according to a log-rank test of the difference. For the ensemble method, this threshold was 0.461 (log-rank test $p = 3.7 * 10^{-10}$).

The framework for validation of the risk score is shown in Figure 6b. Each of the ten total risk scores and associated



***Training dataset**
Yoshihara et al. (2012), N=260

****Candidate methods for comparison:**
All of the methods included in the library plus a simple mean ensemble where each candidate is equally weighted

*****Test datasets**
15 independent datasets from the Curated Ovarian Data R package of Ganzfried et al.

Figure 6: Workflow for training and validating the network-based risk score.

Method	N genes	N validated datasets/total	median validation p-value
ensemble	7	6/14	0.08
glasso - ebic - 0	8	1/14	0.37
glasso - ebic - 0.5	116	2/15	0.42
glasso - ric	14	5/15	0.15
hglasso	11	3/14	0.2
MLE	6	0/1	0.9
glasso - stars - 0.05	42	4/15	0.29
glasso - stars - 0.1	19	6/15	0.22
qgraph - ebic - 0	10	3/15	0.11
qgraph - ebic - 0.5	42	4/15	0.29

(a) Results of risk score validation on 15 independent datasets.

Dataset	p	Sample Size	Low Risk	High Risk	Median Survival	Median Survival (Low Risk)	Median Survival (High Risk)	Censoring (%)
GSE13876	0.015	157	119	38	750	900	480	28
GSE14764*	0.028	80	59	21	1650	-	1200	74
GSE19829.GPL570	0.03	28	20	8	1440	1440	960	39
GSE9891**	0.006	285	216	69	1440	1470	1140	59
PMID17290060	0.001	117	93	24	1920	2340	690	43
PMID19318476	0.019	42	33	9	1020	1050	720	48

(b) Survival characteristics of validated datasets.

Table 4: Results from assessing risk score in validation datasets demonstrate that the ensemble network produced the most robust risk score from the training dataset. Median survival (in days to death) is estimated from the Kaplan-Meier curves for the six datasets in which the seven-gene risk score validated. *In the GSE14764 study, more than half of the low-risk participants survived past the end of the study, so the median survival was undefined in this group. The time of last observation in the low-risk group was 1590 days. ** Vital status missing for 3 participants in GSE9891

thresholds for distinguishing low- vs. high-risk participants was evaluated in 15 independent ovarian cancer datasets comprised of data available from (20) that contained information on the outcome (days to death) as well as vital status. A risk score was considered to validate if the log-rank p-value between the Kaplan-Meier estimates for the high-risk and low-risk groups was less than 0.05. Figure 7c shows the distribution of the validation p-values across the 15 datasets for the risk score developed from the ensemble network as well as from the nine candidate networks. In some cases, the optimal threshold score determined from the training dataset was such that there were insufficient samples above and below the threshold to perform a log-rank test in the validation dataset; these cases are omitted from the boxplot (MLE: N=14 of 15 studies, `ensemble`: N=1, `hglasso`: N=1, `glasso-ebic-0`: N=1). Table 4a shows further detail about the validation. Notably, in the case of `glasso-ebic-0.5`, an empty network was selected. Consequently, every gene formed its own community and all 116 genes were required to construct the risk score. We can thus use the `glasso-ebic-0.5` case to benchmark the ensemble method vs. a naive approach in which the network structure is not leveraged to construct the risk score. Figure 7c shows that the ensemble approach provides a considerable gain.

We note that the SpiderLearner risk score model has two important advantages over the risk score developed by each candidate method. First, it is a much more parsimonious model, requiring only seven predictors to develop a

Dataset	Unadjusted HR	p	Adjusted HR	p	Covariates Available
E.MTAB.386	1.53 (1.09,2.13)	0.01	1.49 (1.06, 2.1)	0.02	Age, Tumor Stage < 4
GSE13876	1.58 (1.16,2.15)	0	1.67(1.2,2.31)	0	Age, Summary Grade
GSE14764	2.3 (1.17,4.52)	0.02	2.26 (1.17,4.39)	0.02	Summary Grade, Summary Stage
GSE17260	1.35 (0.76,2.4)	0.31	1.34 (0.74,2.41)	0.33	Summary Grade, Tumor Stage < 4
GSE18520	1.46 (0.93,2.28)	0.1	-	-	None
GSE19829.GPL570	1.95 (0.85,4.46)	0.11	-	-	None
GSE19829.GPL8300	1.21 (0.6,2.47)	0.59	-	-	None
GSE26712	1.18 (0.9,1.55)	0.23	1.04 (0.78, 1.39)	0.78	Age, Tumor Stage < 4
GSE30009	1.22 (0.21,7.04)	0.82	2.51 (0.39,16.06)	0.33	Age, Summary Grade, Tumor Stage < 4
GSE30161	1.89 (1.12,3.19)	0.02	2.03 (1.15, 3.57)	0.01	Age, Summary Grade, Tumor Stage < 4
GSE32063	0.75 (0.35,1.63)	0.47	0.85 (0.38,1.91)	0.69	Summary Grade, Tumor Stage < 4
GSE9891	1.75 (1.26,2.41)	0	1.86(1.33,2.6)	0	Age, Summary Grade, Summary Stage
PMID17290060	1.43 (0.94,2.19)	0.09	1.32 (0.86,2.05)	0.21	Summary Grade, Tumor Stage < 4
PMID19318476	1.9 (0.9,4.03)	0.09	1.9 (0.81, 4.46)	0.14	Age, Summary Grade, Tumor Stage < 4
TCGA	1.15 (0.95,1.38)	0.15	1.17 (0.97,1.41)	0.1	Age, Summary Grade, Summary Stage

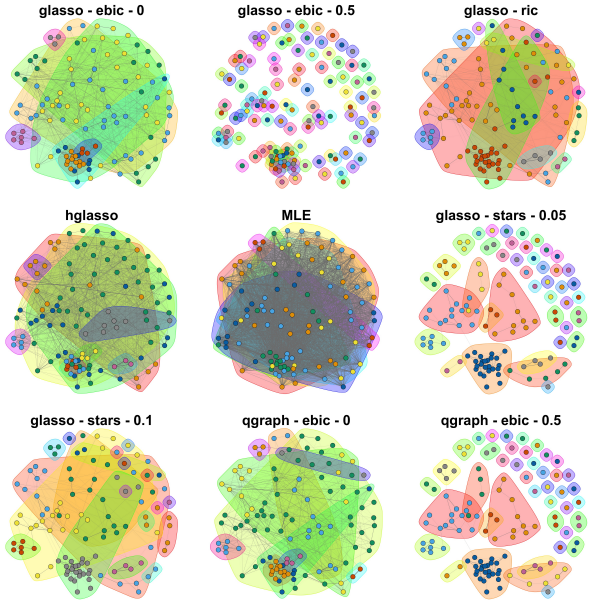
Table 5: Estimated unadjusted and adjusted hazard ratios and 95% confidence intervals for the SpiderLearner risk score. Confidence intervals represent the exponentiated the endpoints of a Wald-type 95% confidence interval for the log hazard ratio.

risk score that validated in 6 of the 15 validation datasets, whereas the only other method achieving this performance required 42 genes to do so. Second, it has the lowest median validation p-value across the 15 validation datasets, and that median approaches nominal significance (SpiderLearner median validation $p = 0.08$). This result indicates better risk prediction even among those datasets in which the risk score did not validate according to the $p < 0.05$ criterion. In an effort to steer clear of overvaluing p-value thresholds, we emphasize that these overall, non-thresholded, results also reinforce the robustness of the SpiderLearner risk-score. Kaplan-Meier plots for the SpiderLearner-based risk score on the six datasets in which it validated are shown in Figure 7d. Similar plots for all 15 datasets are available in Supplementary Figure S17. Median survival times in the low-risk and high-risk group differ substantially (Table 4b), suggesting our method is capable of producing findings with clinical relevance.

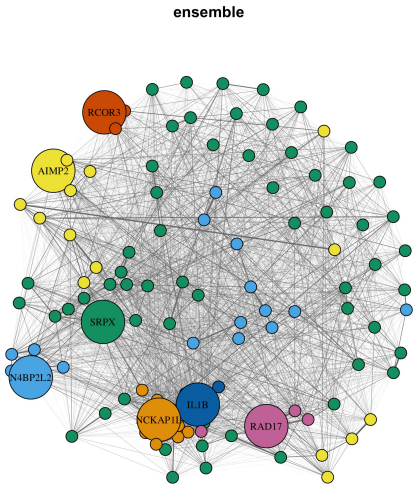
In order to explore the influence of available clinical covariates on the relationship between our risk score and survival, we performed an additional analysis involving Cox PH models. We first estimated the unadjusted hazard ratio of the risk score with a model including only the risk score as a covariate. Next, we estimated the adjusted hazard ratio, adjusting for the following covariates where available: age at time of pathological diagnosis, summary grade (low, high), summary stage (early, late), and tumor stage (< 4, 4). The unadjusted hazard ratio for the risk score was significant in five of the 15 validation sets; in these five cases, the adjusted hazard ratio was also significant (Table 5, Supplementary Figure S18). These results indicate that our risk score provides additional prognostic information above and beyond that contained in these clinical characteristics. Further, the median p-value of the hazard ratio of the SpiderLearner risk score was comparable to that of the best candidate methods, even though it used fewer predictors to generate the score (Supplementary Figure S19).

Inspection of Table 3 shows no substantial differences in age, tumor stage, summary grade, or summary stage between the datasets in which the risk score validated and those in which it did not. We note that 5 of the 6 validated datasets in which the risk score validated used the hgu133a (Affymetrix Human Genome U133) platform

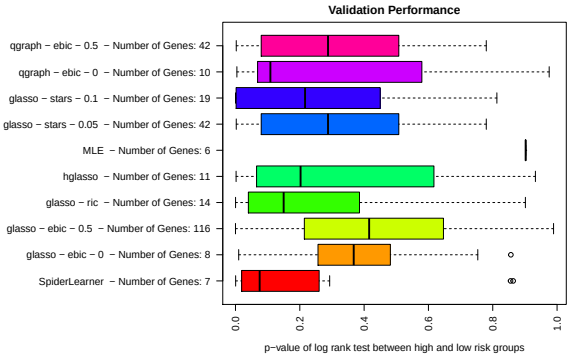
or the hgu133plus2 (Affymetrix Human Genome U133 Plus 2.0) platform, while 8 of the 15 used one or the other of these. The association between platform and validated status was not statistically significant (Fisher's exact test, $p = 0.12$).



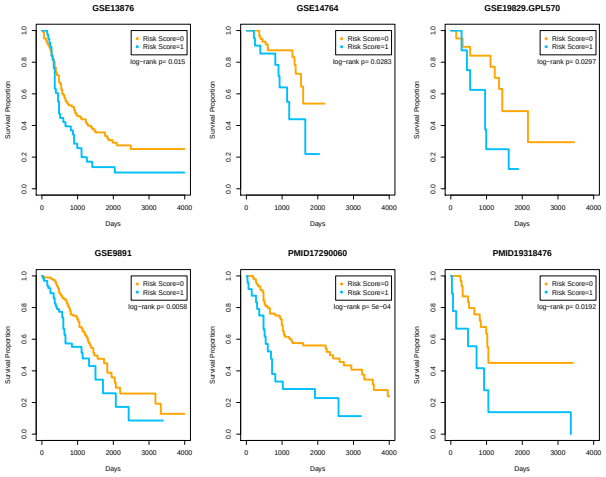
(a) Candidate GGMs and communities.



(b) SpiderLearner GGM and communities.



(c) Performance of risk score on independent data.



(d) Survival differences for validated risk score.

Figure 7: (a) Estimated GGMs and communities for 116 of the genes present in the Yoshihara 126-gene signature for each of the candidate methods in the SpiderLearner library. Vertices of the same color belong to the same community. Estimated community structure varied widely by candidate method. (b) SpiderLearner-estimated network, including the seven local hubs used in the risk score. (c) Boxplots of the validation log rank test p-value testing the null hypothesis of no difference between high-risk and low-risk estimates as defined by the hub-based risk score versus the two-sided alternative. The ensemble model shows a better validation performance and uses a more parsimonious risk score model than other candidates, requiring only seven genes as predictors. (d) The SpiderLearner-based risk score includes the seven labeled genes in (b) as predictors. For the six of 15 validation datasets in which the SpiderLearner-based risk score was successfully validated, Kaplan-Meier estimates for the low-risk (risk score = 0) and high-risk (risk-score=1) groups are shown in (d) along with the p-value of the log-rank test comparing the two curves.

Discussion

In this work, we establish SpiderLearner, an ensemble method for estimating a Gaussian graphical model (GGM) from a convex combination of precision matrices estimated using a broad range of existing open-source candidate methods. In a wide variety of simulation settings, SpiderLearner consistently performed comparably to or better than each of the candidate methods according to a variety of metrics, including relative Frobenius norm of the error matrix, matrix RV coefficient, and element-wise MSE. Importantly, some of the individual candidate methods performed quite poorly; since a researcher's best option *a priori* is to simply choose one of the candidate methods at will, our ensemble method provides a considerable advantage for practical use.

New methods for GGM estimation are being continually developed and assessed. For example, Lartigue et al (2020) conduct an extensive simulation study on GGM estimation for small sample sizes and present a composite procedure that uses a likelihood criterion to select a GGM (69). Methods such as these that are specific to the particular research settings such as the small-sample case are areas for further development. An advantage of SpiderLearner is that such methods, when developed, can be included as candidate models in the ensemble library.

We demonstrated the practical utility of our approach by modeling the network-level interactions of genes belonging to a previously published 126-gene signature of high-risk ovarian cancer (19). The seven genes selected by the ensemble model for inclusion in the risk score are *AIMP2, NCKAP1L, SRPX, N4BP2L2,IL1B,RAD17,* and *RCOR3* (Figure 7b). All seven of these genes have important biological function, with experimental evidence linking their expression levels to processes such as cell proliferation and immune system function that have implications in the study of the development, progression, and treatment of cancer.

AIMP2 is an important tumor suppressor gene, and its splice variant *AIMP2-DX2* has been shown to be an effective pharmaceutical target in chemotherapy-resistant ovarian cancer (70).

Recent work demonstrated that *in vitro* overexpression of *SRPX* resulted in increased ovarian cancer cell invasion activity, while shRNA reduction of *SRPX* mRNA led to a decrease (71).

RAD17 encodes a protein that is related to checkpoint signalling in the cell cycle; RAD17 expression is oscillatory, and engineered stabilization of RAD17 resulted in disrupted checkpoint signalling and consequent diminished re-entry into the cell cycle (72).

RCOR3 encodes a protein called CoREST/REST corepressor 3 and is a paralog of RCOR1, a protein which works together with lysine-specific demethylase 1 (LSD1) in epigenetic regulation of cell fates (73). Upadhyay et al. (2014) demonstrate that RCOR3 is recruited to target genes by LSD1 along with a protein called growth factor independent 1B transcriptional repressor (GFI1B), decreasing histone demethylation and thus de-repressing target gene expression. LSD1 is known to repress tumor suppressor gene expression in oncogenesis, and it is suggested that an increase in RCOR3 expression could attenuate this contribution to oncogenesis.

N4BP2L2 encodes a protein known as (N4BP2L2 full name) or as phosphonoformate immunoassociated protein 5 (PFAAP5) (74). There is evidence that N4BP2L2 is involved in neutrophil deficiency (neutropenia), participating in

transcriptional regulation of a neutrophil production pathway (74).

IL1B encodes the interleukin-1β protein, which has been shown to be higher in serum and plasma of ovarian cancer patients relative to healthy women and has been implicated in important signaling cascades, including the p38/JNK pathway and the NF-κB pathway (75). *NCKAP1L* has recently been identified as a novel tumor micro environment-related biomarker in luminal breast cancer, but has not been previously studied extensively in relation to ovarian cancer (76). *IL1B* and *NCKAP1L* are both members of a number of interesting GO biological processes, including the regulation of phagocytosis, vascular EGFR regulation, neutrophil chemotaxis and migration, granulocyte chemotaxis, and regulation of T-cell, interleukin-6, lymphocyte, and mononuclear cell proliferation.

Recent advances have been proposed to improve the applicability and reproducibility of network estimation methods. (77) propose a Monte Carlo-based method for generating confidence intervals for network statistics, allowing a researcher to assess whether a network property such as edge presence or node centrality differs from that expected by random chance. (78) present a bootstrap-based approach which allows researchers to investigate the variability of an estimated network. (79) develop a network meta-analysis framework that permits integration of estimated networks across multiple studies. Each of these methods can be in theory be applied to GGMs, but rely on the use of an initial estimation algorithm. Consequently, results will still remain sensitive to the many choices that the researcher must make during the estimation process. Our method can thus complement the advances described above, potentially contributing to improved reproducibility and generalizability in GGM estimation.

A limitation of this approach is its time-consuming nature; for K-fold cross validation with M candidate models, the time cost of estimating the ensemble model would be about $M(K+1)$ times the cost of estimating just one candidate model (assuming all candidates take roughly the same amount of time). Moreover, the number of model parameters to be estimated by each candidate model grows quadratically with the number of predictors included in the network, meaning that the computational cost of the ensemble model can quickly become substantial for larger predictor sets. Because model fitting in each fold is independent, parallelization is a good solution to this problem when multiple cores are available. We have implemented parallel processing in the SpiderLearner code to help reduce runtime.

A second limitation lies in the rigidity of the convex combination of precision matrices. The same coefficient is applied to every element of each precision matrix in the current ensemble model formulation. A more flexible extension could address this limitation by partitioning matrices into regions determined to be similar across methods (e.g., the row and column corresponding to a hub node), fitting a convex combination within each partition, and combining these results to yield the ensemble precision matrix.

Conclusion

The past decade has shown numerous advances in GGM estimation, but the burden has still been left on the researcher to determine the specifics of the estimation process, including important aspects such as choice of method,

tuning parameter selection, scoring criteria, and hyperparameter settings. Our SpiderLearner ensemble method removes this barrier, enabling researchers to easily construct a likelihood-based optimal combination from a library of candidate methods. The parsimonious seven-gene risk score identified by our ensemble network-based approach has clear statistical relevance as demonstrated by the validation in six of 15 independent validation datasets, and biological relevance as demonstrated by existing literature on the functions of the seven genes in the SpiderLearner risk score. SpiderLearner is available as open-source R code at `https://github.com/katehoffshutta/SpiderLearner`, and code to reproduce the simulation and application workflows are available at `https://github.com/katehoffshutta/SpiderLearnerWorkflow`.

Supporting information

S1 Appendix. Asymptotics. Proof that in the univariate normal case, the negative log likelihood loss function satisfies the bounded tails condition of (25).

S1 Fig. Comparison of bounded and unbounded loss functions.

S2 Fig. Distribution of partial correlations estimated from the CATHGEN metabolomics dataset.

S1 Table. Runtimes for a range of n,p.

S2 Table. Simulation A sensitivity. Mean sensitivity of edge detection in Simulation A.

S3 Table. Simulation A specificity. Mean specificity of edge detection in Simulation A.

S3 Fig. Simulation A diagnostics. Relative Frobenius norm, Matrix RV coefficient, in-sample log likelihood, and out-of-sample log likelihood for Simulation A.

S4 Fig. Simulation A bias and MSE.

S4 Table. Simulation B weights. Average weights of each candidate method in the ensemble for all of the gold-standard networks explored in Simulation B.

S5 Table. Simulation B sensitivity. Mean sensitivity of edge detection in Simulation B.

S6 Table. Simulation B specificity. Mean specificity of edge detection in Simulation B.

S5 Fig. Simulation B diagnostics. Relative Frobenius norm, Matrix RV coefficient, in-sample log likelihood, and out-of-sample log likelihood for Simulation B.

S6 Fig. Simulation B bias and MSE.

S7 Table. Simulation C weights. Average weights of each candidate method in the ensemble for all of the gold-standard networks explored in Simulation C.

S8 Table. Simulation C sensitivity. Mean sensitivity of edge detection in Simulation C.

S9 Table. Simulation C specificity. Mean specificity of edge detection in Simulation C.

S7 Fig. Simulation C diagnostics. Relative Frobenius norm, Matrix RV coefficient, in-sample log likelihood, and out-of-sample log likelihood for Simulation C.

S8 Fig. Simulation C bias and MSE.

S10 Table. Simulation D sensitivity. Mean sensitivity of edge detection in Simulation D.

S11 Table. Simulation D specificity. Mean specificity of edge detection in Simulation D.

S9 Fig. Simulation D diagnostics. Relative Frobenius norm, Matrix RV coefficient, in-sample log likelihood, and out-of-sample log likelihood for Simulation D.

S10 Fig. Simulation D bias and MSE.

S11 Fig. Choice of K and variability. Element-wise standard error as a function of the number of folds K used to train the SpiderLearner.

S12 Fig. Choice of K and bias. Element-wise bias as a function of the number of folds K used to train the SpiderLearner.

S13 Fig. Choice of K and variability of ensemble weights. Boxplots of ensemble weights for each of the nine candidate methods as a function of the number of folds K used to train the SpiderLearner.

S14 Fig. Sensitivity of estimated model to library. Simulation setting comparing three libraries.

S15 Fig. Sensitivity of estimated model to candidate method. Example for easy-to-visualize 14-gene set.

S12 Table. Sensitivity of estimated model to library. Table of results from application setting comparing four libraries on small ovarian cancer dataset.

S16 Fig. Sensitivity of estimated model to library. Plot of estimated models in application setting comparing four libraries on small ovarian cancer dataset.

S17 Fig. Ensemble risk score performance in all 15 validation datasets. Kaplan-Meier plots of high-risk and low-risk groups with log-rank p-value.

S18 Fig. Forest plots of unadjusted and adjusted hazard ratios associated with the SpiderLearner risk score.

S19 Fig. Significance of hazard ratios associated with the network risk score for all candidate methods, the simple mean, and the SpiderLearner.

Acknowledgments

The authors gratefully acknowledge Subhajit Naskar for his contributions in designing the simulation studies that inspired this work.

The authors gratefully acknowledge the participants of the CATHGEN study and of the 16 ovarian cancer studies utilized in the biological application of this manuscript.

Author Contributions

KHS conceptualized, developed, and implemented the method. LBB consulted on theoretical foundations of the method. KHS and RB conceptualized the simulation and application. KHS conducted the simulation and application. DMS provided input on the application and interpretation. RB provided guidance and supervision for the project. KHS wrote the manuscript. LBB, DMS, and RB reviewed the manuscript and provided critical input.

Competing Interests Statement

The authors have no competing interests to declare.

Human Subjects Statement

CATHGEN dataset

The CATHGEN study was approved by the Duke Institutional Review Board and subjects provided informed consent, as described in (37).

Ovarian cancer datasets

We refer the reader to each originally published dataset (Table 3) for details regarding human subjects protections in each of these studies. Because all of these data are public, analyses performed here is IRB-exempt under Category 4 – Secondary Research Uses of Identifiable Private Information or Identifiable Biospecimens.

References

[1] C. Uhler, "Gaussian graphical models: An algebraic and geometric perspective," 2017.

[2] S. L. Lauritzen, *Graphical models*, vol. 17. Clarendon Press, 1996.

[3] J. Friedman, T. Hastie, and R. Tibshirani, "Sparse inverse covariance estimation with the graphical lasso," *Biostatistics*, vol. 9, no. 3, pp. 432–441, 2008.

[4] M. Yuan and Y. Lin, "Model selection and estimation in the gaussian graphical model," *Biometrika*, vol. 94, pp. 19–35, 2007.

[5] O. Banerjee, L. E. Ghaoui, and A. d'Aspremont, "Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data," *Journal of Machine learning research*, vol. 9, no. Mar, pp. 485–516, 2008.

[6] D. M. Witten, J. H. Friedman, and N. Simon, "New insights and faster computations for the graphical lasso," *Journal of Computational and Graphical Statistics*, vol. 20, no. 4, pp. 892–900, 2011.

[7] T. Zhao, H. Liu, K. Roeder, J. Lafferty, and L. Wasserman, "The huge package for high-dimensional undirected graph estimation in r," *Journal of Machine Learning Research*, vol. 13, no. Apr, pp. 1059–1062, 2012.

[8] H. Liu, L. Wang, *et al.*, "Tiger: A tuning-insensitive approach for optimally estimating gaussian graphical models," *Electronic Journal of Statistics*, vol. 11, no. 1, pp. 241–294, 2017.

[9] S. Epskamp, D. Borsboom, and E. I. Fried, "Estimating psychological networks and their accuracy: A tutorial paper," *Behavior Research Methods*, vol. 50, no. 1, pp. 195–212, 2018.

[10] K. Shutta, S. Naskar, K. Rexrode, D. Scholtens, and R. Balasubramanian, "Estimation of metabolomic networks with gaussian graphical models." `https://raji-lab.github.io/News/Metabolomics2019.pdf`, 2019. Metabolomics 2019 Conference, The Hague, Netherlands.

[11] K. Shutta, S. Naskar, K. Rexrode, D. Scholtens, and R. Balasubramanian, "Estimation of metabolomic networks with gaussian graphical models." `https://raji-lab.github.io/News/ENAR2020.pdf`, 2020. ENAR 2020 Conference, Online.

[12] A. C. Wysocki and M. Rhemtulla, "On penalty parameter selection for estimating network models," *Multivariate behavioral research*, pp. 1–15, 2019.

[13] D. H. Wolpert, "Stacked generalization," *Neural networks*, vol. 5, no. 2, pp. 241–259, 1992.

[14] L. Breiman, "Stacked regressions," *Machine learning*, vol. 24, no. 1, pp. 49–64, 1996.

[15] M. J. van der Laan, E. C. Polley, and A. E. Hubbard, "Super learner," *Statistical applications in genetics and molecular biology*, vol. 6, no. 1, 2007.

[16] H.-Y. Chuang, E. Lee, Y.-T. Liu, D. Lee, and T. Ideker, "Network-based classification of breast cancer metastasis," *Molecular systems biology*, vol. 3, no. 1, p. 140, 2007.

[17] C. Winter, G. Kristiansen, S. Kersting, J. Roy, D. Aust, T. Knösel, P. Rümmele, B. Jahnke, V. Hentrich, F. Rückert, *et al.*, "Google goes cancer: improving outcome prediction for cancer patients by network-based ranking of marker genes," *PLoS Comput Biol*, vol. 8, no. 5, p. e1002511, 2012.

[18] Y. Li, H. Tang, Z. Sun, A. O. Bungum, E. S. Edell, W. L. Lingle, S. M. Stoddard, M. Zhang, J. Jen, P. Yang, *et al.*, "Network-based approach identified cell cycle genes as predictor of overall survival in lung adenocarcinoma patients," *Lung Cancer*, vol. 80, no. 1, pp. 91–98, 2013.

[19] K. Yoshihara, T. Tsunoda, D. Shigemizu, H. Fujiwara, M. Hatae, H. Fujiwara, H. Masuzaki, H. Katabuchi, Y. Kawakami, A. Okamoto, *et al.*, "High-risk ovarian cancer based on 126-gene expression signature is uniquely characterized by downregulation of antigen presentation pathway," *Clinical cancer research*, vol. 18, no. 5, pp. 1374–1385, 2012.

[20] B. F. Ganzfried, M. Riester, B. Haibe-Kains, T. Risch, S. Tyekucheva, I. Jazic, X. V. Wang, M. Ahmadifar, M. J. Birrer, G. Parmigiani, *et al.*, "curatedovariandata: clinically annotated data for the ovarian cancer transcriptome," *Database*, vol. 2013, 2013.

[21] A. I. Naimi and L. B. Balzer, "Stacked generalization: an introduction to super learning," *European journal of epidemiology*, vol. 33, no. 5, pp. 459–464, 2018.

[22] Y. Ye, *Interior algorithms for linear, quadratic, and linearly constrained non-linear programming*. PhD thesis, Ph. D. thesis, Department of ESS, Stanford University, 1987.

[23] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An introduction to statistical learning*, vol. 112. Springer, 2013.

[24] E. C. Polley and M. J. van der Laan, "Super learner in prediction," 2010.

[25] A. W. van der Vaart, S. Dudoit, and M. J. van der Laan, "Oracle inequalities for multi-fold cross validation," *Statistics and Decisions*, vol. 24, no. 3, pp. 351–371, 2006.

[26] M. L. Petersen, E. LeDell, J. Schwab, V. Sarovar, R. Gross, N. Reynolds, J. E. Haberer, K. Goggin, C. Golin, J. Arnsten, *et al.*, "Super learner analysis of electronic adherence data improves viral prediction and may provide strategies for selective hiv rna monitoring," *Journal of acquired immune deficiency syndromes (1999)*, vol. 69, no. 1, p. 109, 2015.

[27] L. B. Balzer, D. V. Havlir, M. R. Kamya, G. Chamie, E. D. Charlebois, T. D. Clark, C. A. Koss, D. Kwarisiima, J. Ayieko, N. Sang, *et al.*, "Machine learning to identify persons at high-risk of human immunodeficiency virus acquisition in rural kenya and uganda," *Clinical Infectious Diseases*, vol. 71, no. 9, pp. 2326–2333, 2020.

[28] W. Zheng, L. Balzer, M. van Der Laan, M. Petersen, and S. Collaboration, "Constrained binary classification using ensemble learning: an application to cost-efficient targeted prep strategies," *Statistics in medicine*, vol. 37, no. 2, pp. 261–279, 2018.

[29] E. Polley, E. LeDell, C. Kennedy, S. Lendle, and M. van der Laan, "Package 'superlearner'," 2019.

[30] P. Erdős and A. Rényi, "On the evolution of random graphs," *Publ. Math. Inst. Hung. Acad. Sci*, vol. 5, no. 1, pp. 17–60, 1960.

[31] E. N. Gilbert, "Random graphs," *The Annals of Mathematical Statistics*, vol. 30, no. 4, pp. 1141–1144, 1959.

[32] A.-L. Barabási *et al.*, *Network science*. Cambridge university press, 2016.

[33] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world'networks," *nature*, vol. 393, no. 6684, p. 440, 1998.

[34] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *science*, vol. 286, no. 5439, pp. 509–512, 1999.

[35] G. Csardi, T. Nepusz, *et al.*, "The igraph software package for complex network research," *InterJournal, complex systems*, vol. 1695, no. 5, pp. 1–9, 2006.

[36] P. Erdős and A. Rényi, "On the evolution of random graphs," *Publ. Math. Inst. Hung. Acad. Sci*, vol. 5, no. 1, pp. 17–60, 1960.

[37] W. E. Kraus, C. B. Granger, M. H. Sketch, M. P. Donahue, G. S. Ginsburg, E. R. Hauser, C. Haynes, L. K. Newby, M. Hurdle, Z. E. Dowdy, *et al.*, "A guide for a cardiovascular genomics biorepository: the cathgen experience," *Journal of cardiovascular translational research*, vol. 8, no. 8, pp. 449–457, 2015.

[38] H. Xu, X. Gu, M. G. Tadesse, and R. Balasubramanian, "A modified random survival forests algorithm for high dimensional predictors and self-reported outcomes," *Journal of Computational and Graphical Statistics*, vol. 27, no. 4, pp. 763–772, 2018.

[39] K. Tan, P. London, K. Mohan, S. Lee, M. Fazel, and D. Witten, "Learning graphical models with hubs.," *Journal of machine learning research: JMLR*, vol. 15, pp. 3297–3331, 2014.

[40] R. Foygel and M. Drton, "Extended bayesian information criteria for gaussian graphical models," in *Advances in neural information processing systems*, pp. 604–612, 2010.

[41] H. Liu, K. Roeder, and L. Wasserman, "Stability approach to regularization selection (stars) for high dimensional graphical models," in *Advances in neural information processing systems*, pp. 1432–1440, 2010.

[42] S. Epskamp, A. O. Cramer, L. J. Waldorp, V. D. Schmittmann, D. Borsboom, *et al.*, "qgraph: Network visualizations of relationships in psychometric data," *Journal of statistical software*, vol. 48, no. 4, pp. 1–18, 2012.

[43] S. Epskamp, G. Costantini, J. Haslbeck, A. Isvoranu, A. O. Cramer, L. J. Waldorp, V. D. Schmittmann, and D. Borsboom, "Package 'qgraph'," 2020.

[44] B. T. Rolfs and B. Rajaratnam, "A note on the lack of symmetry in the graphical lasso," *Computational Statistics & Data Analysis*, vol. 57, no. 1, pp. 429–434, 2013.

[45] O. J. Dunn, "Multiple comparisons among means," *Journal of the American statistical association*, vol. 56, no. 293, pp. 52–64, 1961.

[46] P. Robert and Y. Escoufier, "A unifying tool for linear multivariate statistical methods: the rv-coefficient," *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, vol. 25, no. 3, pp. 257–265, 1976.

[47] U. G. Indahl, T. Næs, and K. H. Liland, "A similarity index for comparing coupled matrices," *Journal of Chemometrics*, vol. e3049, 2018.

[48] G. Stewart, "On the continuity of the generalized inverse," *SIAM Journal on Applied Mathematics*, vol. 17, no. 1, pp. 33–45, 1969.

[49] G. Casella and R. L. Berger, *Statistical inference*, vol. 2. Duxbury Pacific Grove, CA, 2002.

[50] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[51] E. C. Polley, S. Rose, and M. J. van der Laan, "Super learning," pp. 43–66, 2011.

[52] S. Bentink, B. Haibe-Kains, T. Risch, J.-B. Fan, M. S. Hirsch, K. Holton, R. Rubio, C. April, J. Chen, E. Wickham-Garcia, *et al.*, "Angiogenic mrna and microrna gene expression signature predicts a novel subtype of serous ovarian cancer," *PloS one*, vol. 7, no. 2, p. e30269, 2012.

[53] A. P. Crijns, R. S. Fehrmann, S. de Jong, F. Gerbens, G. J. Meersma, H. G. Klip, H. Hollema, R. M. Hofstra, G. J. te Meerman, E. G. de Vries, *et al.*, "Survival-related profile, pathways, and transcription factors in ovarian cancer," *PLoS Med*, vol. 6, no. 2, p. e1000024, 2009.

[54] C. Denkert, J. Budczies, S. Darb-Esfahani, B. Györffy, J. Sehouli, D. Könsgen, R. Zeillinger, W. Weichert, A. Noske, A.-C. Buckendahl, *et al.*, "A prognostic gene expression index in ovarian cancer—validation across different independent data sets," *The Journal of Pathology: A Journal of the Pathological Society of Great Britain and Ireland*, vol. 218, no. 2, pp. 273–280, 2009.

[55] K. Yoshihara, A. Tajima, T. Yahata, S. Kodama, H. Fujiwara, M. Suzuki, Y. Onishi, M. Hatae, K. Sueyoshi, H. Fujiwara, *et al.*, "Gene expression profile for predicting survival in advanced-stage serous ovarian cancer across two independent datasets," *PloS one*, vol. 5, no. 3, p. e9615, 2010.

[56] S. C. Mok, T. Bonome, V. Vathipadiekal, A. Bell, M. E. Johnson, D.-C. Park, K. Hao, D. K. Yip, H. Donninger, L. Ozbun, *et al.*, "A gene signature predictive for outcome in advanced ovarian cancer identifies a survival factor: microfibril-associated glycoprotein 2," *Cancer cell*, vol. 16, no. 6, pp. 521–532, 2009.

[57] P. A. Konstantinopoulos, D. Spentzos, B. Y. Karlan, T. Taniguchi, E. Fountzilas, N. Francoeur, D. A. Levine, and S. A. Cannistra, "Gene expression profile of brcaness that correlates with responsiveness to chemotherapy and with outcome in patients with epithelial ovarian cancer," *Journal of clinical oncology*, vol. 28, no. 22, p. 3555, 2010.

[58] T. Bonome, D. A. Levine, J. Shih, M. Randonovich, C. A. Pise-Masison, F. Bogomolniy, L. Ozbun, J. Brady, J. C. Barrett, J. Boyd, *et al.*, "A gene signature predicting for survival in suboptimally debulked patients with ovarian cancer," *Cancer research*, vol. 68, no. 13, pp. 5478–5486, 2008.

[59] J.-P. Gillet, A. M. Calcagno, S. Varma, B. Davidson, M. B. Elstrand, R. Ganapathi, A. A. Kamat, A. K. Sood, S. V. Ambudkar, M. V. Seiden, *et al.*, "Multidrug resistance–linked gene signature predicts overall survival of patients with primary ovarian serous carcinoma," *Clinical Cancer Research*, vol. 18, no. 11, pp. 3197–3206, 2012.

[60] J. S. Ferriss, Y. Kim, L. Duska, M. Birrer, D. A. Levine, C. Moskaluk, D. Theodorescu, and J. K. Lee, "Multi-gene expression predictors of single drug responses to adjuvant chemotherapy in ovarian carcinoma: predicting platinum resistance," *PloS one*, vol. 7, no. 2, p. e30550, 2012.

[61] R. W. Tothill, A. V. Tinker, J. George, R. Brown, S. B. Fox, S. Lade, D. S. Johnson, M. K. Trivett, D. Etemadmoghadam, B. Locandro, *et al.*, "Novel molecular subtypes of serous and endometrioid ovarian cancer linked to clinical outcome," *Clinical cancer research*, vol. 14, no. 16, pp. 5198–5208, 2008.

[62] H. K. Dressman, A. Berchuck, G. Chan, J. Zhai, A. Bild, R. Sayer, J. Cragun, J. Clarke, R. S. Whitaker, L. Li, *et al.*, "An integrated genomic-based approach to individualized treatment of patients with advanced-stage ovarian cancer," *Journal of clinical oncology*, vol. 25, no. 5, pp. 517–525, 2007.

[63] A. Berchuck, E. S. Iversen, J. Luo, J. P. Clarke, H. Horne, D. A. Levine, J. Boyd, M. A. Alonso, A. A. Secord, M. Q. Bernardini, *et al.*, "Microarray analysis of early stage serous ovarian cancers shows profiles predictive of favorable outcome," *Clinical Cancer Research*, vol. 15, no. 7, pp. 2448–2455, 2009.

[64] C. G. A. R. Network *et al.*, "Integrated genomic analyses of ovarian carcinoma," *Nature*, vol. 474, no. 7353, p. 609, 2011.

[65] P. Pons and M. Latapy, "Computing communities in large networks using random walks," in *J. Graph Algorithms Appl*, Citeseer, 2006.

[66] J. M. Kleinberg, "Authoritative sources in a hyperlinked environment," *Journal of the ACM (JACM)*, vol. 46, no. 5, pp. 604–632, 1999.

[67] J. Platig, P. J. Castaldi, D. DeMeo, and J. Quackenbush, "Bipartite community structure of eqtls," *PLoS computational biology*, vol. 12, no. 9, p. e1005033, 2016.

[68] T. M. Therneau, *A Package for Survival Analysis in R*, 2021. R package version 3.2-11.

[69] T. Lartigue, S. Bottani, S. Baron, O. Colliot, S. Durrleman, and S. Allassonnière, "Gaussian graphical model exploration and selection in high dimension low sample size setting," *arXiv*, pp. arXiv–2003, 2020.

[70] J. W. Choi, J.-W. Lee, J. K. Kim, H.-K. Jeon, J.-J. Choi, D. G. Kim, B.-G. Kim, D.-H. Nam, H. J. Kim, S. H. Yun, *et al.*, "Splicing variant of aimp2 as an effective target against chemoresistant ovarian cancer," *Journal of molecular cell biology*, vol. 4, no. 3, pp. 164–173, 2012.

[71] C. L. Liu, H. W. Pan, P. L. Torng, M. H. Fan, and T. L. Mao, "Srpx and hmcn1 regulate cancer-associated fibroblasts to promote the invasiveness of ovarian carcinoma," *Oncology reports*, vol. 42, no. 6, pp. 2706–2715, 2019.

[72] L. Zhang, C.-H. Park, J. Wu, H. Kim, W. Liu, T. Fujita, M. Balasubramani, E. M. Schreiber, X.-F. Wang, and Y. Wan, "Proteolysis of rad17 by cdh1/apc regulates checkpoint termination and recovery from genotoxic stress," *The EMBO journal*, vol. 29, no. 10, pp. 1726–1737, 2010.

[73] G. Upadhyay, A. H. Chowdhury, B. Vaidyanathan, D. Kim, and S. Saleque, "Antagonistic actions of rcor proteins regulate lsd1 activity and cellular differentiation," *Proceedings of the National Academy of Sciences*, vol. 111, no. 22, pp. 8071–8076, 2014.

[74] S. J. Salipante, M. E. Rojas, B. Korkmaz, Z. Duan, J. Wechsler, K. F. Benson, R. E. Person, H. L. Grimes, and M. S. Horwitz, "Contributions to neutropenia from pfaap5 (n4bp2l2), a novel protein mediating transcriptional repressor cooperation between gfi1 and neutrophil elastase," *Molecular and cellular biology*, vol. 29, no. 16, pp. 4394–4405, 2009.

[75] C. Rébé and F. Ghiringhelli, "Interleukin-1β and cancer," *Cancers*, vol. 12, no. 7, p. 1791, 2020.

[76] Y. Wang, M. Zhu, F. Guo, Y. Song, X. Fan, and G. Qin, "Identification of tumor microenvironment-related prognostic biomarkers in luminal breast cancer," *Frontiers in Genetics*, vol. 11, 2020.

[77] D. Steinley, M. Hoffman, M. J. Brusco, and K. J. Sher, "A method for making inferences in network analysis: Comment on forbes, wright, markon, and krueger (2017).," 2017.

[78] S. Epskamp, D. Borsboom, and E. I. Fried, "Estimating psychological networks and their accuracy: A tutorial paper," *Behavior Research Methods*, vol. 50, no. 1, pp. 195–212, 2018.

[79] S. Epskamp, A.-M. Isvoranu, and M. Cheung, "Meta-analytic gaussian network aggregation," 2020.