A Fast and Scalable Joint Estimator for Integrating Additional Knowledge in Learning Multiple Related Sparse Gaussian Graphical Models

Beilun Wang [1] **Arshdeep Sekhon** [1] **Yanjun Qi** [1]

Abstract

We consider the problem of including additional knowledge in estimating sparse Gaussian graphical models (sGGMs) from aggregated samples, arising often in bioinformatics and neuroimaging applications. Previous joint sGGM estimators either fail to use existing knowledge or cannot scale-up to many tasks (large K) under a high-dimensional (large p) situation. In this paper, we propose a novel Joint Elementary Estimator incorporating additional Knowledge (JEEK) to infer multiple related sparse Gaussian Graphical models from large-scale heterogeneous data. Using domain knowledge as weights, we design a novel hybrid norm as the minimization objective to enforce the superposition of two weighted sparsity constraints, one on the shared interactions and the other on the task-specific structural patterns. This enables JEEK to elegantly consider various forms of existing knowledge based on the domain at hand and avoid the need to design knowledge-specific optimization. JEEK is solved through a fast and entry-wise parallelizable solution that largely improves the computational efficiency of the state-of-the-art $O(p^5K^4)$ to $O(p^2K^4)$. We conduct a rigorous statistical analysis showing that JEEK achieves the same convergence rate $O(\log(Kp)/n_{tot})$ as the state-of-the-art estimators that are much harder to compute. Empirically, on multiple synthetic datasets and two real-world data, JEEK outperforms the speed of the state-of-arts significantly while achieving the same level of prediction accuracy. [1]

1. Introduction

Technology revolutions in the past decade have collected large-scale heterogeneous samples from many scientific domains. For instance, genomic technologies have delivered petabytes of molecular measurements across more than hundreds of types of cells and tissues from national projects like ENCODE (Consortium et al., 2012) and TCGA (Network et al., 2011). Neuroimaging technologies have generated petabytes of functional magnetic resonance imaging (fMRI) datasets across thousands of human subjects (shared publicly through projects like openfMRI (Poldrack et al., 2013). Given such data, understanding and quantifying variable graphs from heterogeneous samples about multiple contexts is a fundamental analysis task.

Such variable graphs can significantly simplify network-driven studies about diseases (Ideker & Krogan, 2012), can help understand the neural characteristics underlying clinical disorders (Uddin et al., 2013) and can allow for understanding genetic or neural pathways and systems. The number of contexts (denoted as K) that those applications need to consider grows extremely fast, ranging from tens (e.g., cancer types in TCGA (Network et al., 2011)) to thousands (e.g., number of subjects in openfMRI (Poldrack et al., 2013)). The number of variables (denoted as p) ranges from hundreds (e.g., number of brain regions) to tens of thousands (e.g., number of human genes).

The above data analysis problem can be formulated as jointly estimating K conditional dependency graphs $G^{(1)}, G^{(2)}, \dots, G^{(K)}$ on a single set of p variables based on heterogeneous samples accumulated from K distinct contexts. For homogeneous data samples from a given i-th context, one typical approach is the sparse Gaussian Graphical Model (sGGM) (Lauritzen, 1996; Yuan & Lin, 2007). sGGM assumes samples are independently and identically drawn from $N_p(\mu^{(i)}, \Sigma^{(i)})$, a multivariate Gaussian distribution with mean vector $\mu^{(i)}$ and covariance matrix $\Sigma^{(i)}$. The graph structure $G^{(i)}$ is encoded by the sparsity pattern of the inverse covariance matrix, also named precision matrix, $\Omega^{(i)}$. $\Omega^{(i)} := (\Sigma^{(i)})^{-1}$. $\Omega^{(i)}_{jk} = 0$ if and only if in $G^{(i)}$ an edge does not connect j-th node and k-th node (i.e., conditional independent). sGGM imposes an ℓ_1 penalty

[1] Department of Computer Science, University of Virginia, *http://www.jointnets.org/*. Correspondence to: Beilun Wang <bw4mw@virginia.edu>, Yanjun Qi <yanjun@virginia.edu>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018. Copyright 2018 by the author(s).

[1] In this updated version, we correct one equation error we had before about kw norm's dual form. Our code implementation was correct, therefore no change in our toolbox and empirical results.

on the parameter $\Omega^{(i)}$ to achieve a consistent estimation under high-dimensional situations. When handling heterogeneous data samples, rather than estimating sGGM of each condition separately, a multi-task formulation that jointly estimates K different but related sGGMs can lead to a better generalization(Caruana, 1997).

Previous studies for joint estimation of multiple sGGMs roughly fall into four categories: (Danaher et al., 2013; Mohan et al., 2013; Chiquet et al., 2011; Honorio & Samaras, 2010; Guo et al., 2011; Zhang & Wang, 2012; Zhang & Schneider, 2010; Zhu et al., 2014): (1) The first group seeks to optimize a sparsity regularized data likelihood function plus an extra penalty function $\mathcal{R}'$ to enforce structural similarity among multiple estimated networks. Joint graphical lasso (JGL) (Danaher et al., 2013) proposed an alternating direction method of multipliers (ADMM) based optimization algorithm to work with two regularization functions ($\ell_1 + \mathcal{R}'$). (2) The second category tries to recover the support of $\Omega^{(i)}$ using sparsity penalized regressions in a column by column fashion. Recently (Monti et al., 2015) proposed to learn population and subject-specific brain connectivity networks via a so-called "Mixed Neighborhood Selection" (MSN) method in this category. (3) The third type of methods seeks to minimize the joint sparsity of the target precision matrices under matrix inversion constraints. One recent study, named SIMULE (Shared and Individual parts of MULtiple graphs Explicitly) (Wang et al., 2017b), automatically infers both specific edge patterns that are unique to each context and shared interactions preserved among all the contexts (i.e. by modeling each precision matrix as $\Omega^{(i)} = \Omega_I^{(i)} + \Omega_S$) via the constrained ℓ_1 minimization. Following the CLIME estimator (Pang et al., 2014), the constrained ℓ_1 convex formulation can also be solved column by column via linear programming. However, all three categories of aforementioned estimators are difficult to scale up when the dimension p or the number of tasks K are large because they cannot avoid expensive steps like SVD (Danaher et al., 2013) for JGL, linear programming for SIMULE or running multiple iterations of p expensive penalized regressions in MNS. (4) The last category extends the so-called ”Elementary Estimator” graphical model (EE-GM) formulation (Yang et al., 2014b) to revise JGL's penalized likelihood into a constrained convex program that minimizes ($\ell_1 + \mathcal{R}'$). One proposed estimator FASJEM (Wang et al., 2017a) is solved in an entry-wise manner and group-entry-wise manner that largely outperforms the speed of its JGL counterparts. More details of the related works are in Section (5).

One significant caveat of state-of-the-art joint sGGM estimators is the fact that little attention has been paid to incorporating existing knowledge of the nodes or knowledge of the relationships among nodes in the models. In addition to the samples themselves, additional information is widely available in real-world applications. In fact, incorporating the knowledge is of great scientific interest. A prime example is when estimating the functional brain connectivity networks among brain regions based on fMRI samples, the spatial position of the regions are readily available. Neuroscientists have gathered considerable knowledge regarding the spatial and anatomical evidence underlying brain connectivity (e.g., short edges and certain anatomical regions are more likely to be connected (Watts & Strogatz, 1998)). Another important example is the problem of identifying gene-gene interactions from patients' gene expression profiles across multiple cancer types. Learning the statistical dependencies among genes from such heterogeneous datasets can help to understand how such dependencies vary from normal to abnormal and help to discover contributing markers that influence or cause the diseases. Besides the patient samples, state-of-the-art bio-databases like HPRD (Prasad et al., 2009) have collected a significant amount of information about direct physical interactions among corresponding proteins, regulatory gene pairs or signaling relationships collected from high-qualify bio-experiments.

Although being strong evidence of structural patterns we aim to discover, this type of information has rarely been considered in the joint sGGM formulation of such samples. To the authors' best knowledge, only one study named as W-SIMULE tried to extend the constrained ℓ_1 minimization in SIMULE into weighted ℓ_1 for considering spatial information of brain regions in the joint discovery of heterogeneous neural connectivity graphs (Singh et al., 2017). This method was designed just for the neuroimaging samples and has $O(p^5K^4)$ time cost, making it not scalable for large-scale settings (more details in Section 3).

This paper aims to fill this gap by adding additional knowledge most effectively into scalable and fast joint sGGM estimations. We propose a novel model, namely Joint Elementary Estimator incorporating additional Knowledge (JEEK), that presents a principled and scalable strategy to include additional knowledge when estimating multiple related sGGMs jointly. Briefly speaking, this paper makes the following contributions:

- **Novel approach:** JEEK presents a new way of integrating additional knowledge in learning multi-task sGGMs in a scalable way. (Section 3)
- **Fast optimization:** We optimize JEEK through an entry-wise and group-entry-wise manner that can dramatically improve the time complexity to $O(p^2K^4)$. (Section 3.4)
- **Convergence rate:** We theoretically prove the convergence rate of JEEK as $O(\log(Kp)/n_{tot})$. This rate shows the benefit of joint estimation and achieves the same convergence rate as the state-of-the-art that are much harder to compute. (Section 4)
- **Evaluation:** We evaluate JEEK using several synthetic

datasets and two real-world data, one from neuroscience and one from genomics. It outperforms state-of-the-art baselines significantly regarding the speed. (Section 6)

JEEK provides the flexibility of using $(K+1)$ different weight matrices representing the extra knowledge. We try to showcase a few possible designs of the weight matrices in Section 12, including (but not limited to):

- Spatial or anatomy knowledge about brain regions;
- Knowledge of known co-hub nodes or perturbed nodes;
- Known group information about nodes, such as genes belonging to the same biological pathway or cellular location;
- Using existing known edges as the knowledge, like the known protein interaction databases for discovering gene networks (a semi-supervised setting for such estimations).

We sincerely believe the scalability and flexibility provided by JEEK can make structure learning of joint sGGM feasible in many real-world tasks.

Att: Due to space limitations, we have put details of certain contents (e.g., proofs) in the appendix. Notations with "S:" as the prefix in the numbering mean the corresponding contents are in the appendix. For example, full proofs are in Section (10).

Notations: math notations we use are described in Section (8). $n_{tot} = \sum\limits_{i=1}^{K} n_i$ is the total number of data samples. The Hadamard product $\circ$ is the element-wise product between two matrices. Also to simplify the notations, we abuse the notation $\frac{1}{W}$ to represent a new matrix being generated by element wise division of each entry in W by 1.

2. Background

Sparse Gaussian graphical model (sGGM): The classic formulation of estimating sparse Gaussian Graphical model (Yuan & Lin, 2007) from a single given condition (single sGGM) is the "graphical lasso" estimator (GLasso) (Yuan & Lin, 2007; Banerjee et al., 2008). It solves the following ℓ_1 penalized maximum likelihood estimation (MLE) problem:

$$\underset{\Omega>0}{\operatorname{argmin}} - \log\det(\Omega) + <\Omega, \widehat{\Sigma}> + \lambda_n ||\Omega||_1 \tag{2.1}$$

M-Estimator with Decomposable Regularizer in High-Dimensional Situations: Recently the seminal study (Negahban et al., 2009) proposed a unified framework for high-dimensional analysis of the following general formulation: M-estimators with decomposable regularizers:

$$\underset{\theta}{\operatorname{argmin}}\, \mathcal{L}(\theta) + \lambda_n \mathcal{R}(\theta) \tag{2.2}$$

where $\mathcal{R}(\cdot)$ represents a decomposable regularization function and $\mathcal{L}(\cdot)$ represents a loss function (e.g., the negative log-likelihood function in sGGM $\mathcal{L}(\Omega) = -\log\det(\Omega) + <\Omega, \widehat{\Sigma}>$). Here $\lambda_n > 0$ is the tuning parameter.

Elementary Estimators (EE): Using the analysis framework from (Negahban et al., 2009), recent studies (Yang et al., 2014a;b;c) propose a new category of estimators named "Elementary estimator" (EE) with the following general formulation:

$$\begin{aligned} &\underset{\theta}{\operatorname{argmin}}\, \mathcal{R}(\theta) \\ &\text{subject to:} \mathcal{R}^*(\theta - \widehat{\theta}_n) \le \lambda_n \end{aligned} \tag{2.3}$$

Where $\mathcal{R}^*(\cdot)$ is the dual norm of $\mathcal{R}(\cdot)$,

$$\mathcal{R}^*(v) := \sup_{u \neq 0} \frac{<u, v>}{\mathcal{R}(u)} = \sup_{\mathcal{R}(u) \le 1} <u, v>. \tag{2.4}$$

The solution of Eq. (10.7) achieves the near optimal convergence rate as Eq. (2.2) when satisfying certain conditions. $\mathcal{R}(\cdot)$ represents a decomposable regularization function (e.g., ℓ_1-norm) and $\mathcal{R}^*(\cdot)$ is the dual norm of $\mathcal{R}(\cdot)$ (e.g., ℓ_∞-norm is the dual norm of ℓ_1-norm). λ_n is a regularization parameter.

The basic motivation of Eq. (10.7) is to build simpler and possibly fast estimators, that yet come with statistical guarantees that are nonetheless comparable to regularized MLE. $\widehat{\theta}_n$ needs to be carefully constructed, well-defined and closed-form for the purpose of simpler computations. The formulation defined by Eq. (10.7) is to ensure its solution having the desired structure defined by $\mathcal{R}(\cdot)$. For cases of high-dimensional estimation of linear regression models, $\widehat{\theta}_n$ can be the classical ridge estimator that itself is closed-form and with strong statistical convergence guarantees in high-dimensional situations.

EE-sGGM: (Yang et al., 2014b) proposed elementary estimators for graphical models (GM) of exponential families, in which $\widehat{\theta}_n$ represents so-called proxy of backward mapping for the target GM (more details in Section 11). The key idea (summarized in the upper row of Figure 2) is to investigate the vanilla MLE and where it "breaks down" for estimating a graphical model of exponential families in the case of high-dimensions (Yang et al., 2014b). Essentially the vanilla graphical model MLE can be expressed as a backward mapping that computes the model parameters from some given moments in an exponential family distribution. For instance, in the case of learning Gaussian GM (GGM) with vanilla MLE, the backward mapping is $\widehat{\Sigma}^{-1}$ that estimates Ω from the sample covariance matrix (moment) $\widehat{\Sigma}$. We introduce the details of backward mapping in Section 11.

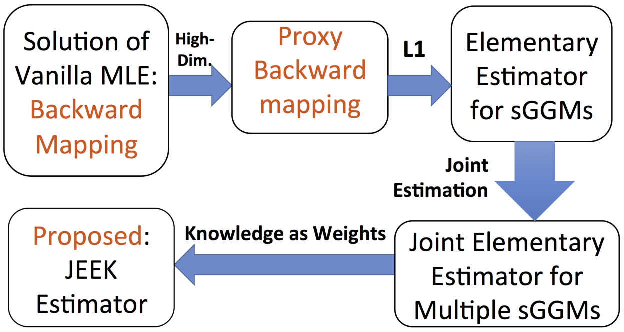

Solution of Vanilla MLE: Backward Mapping
High-Dim.
Proxy Backward mapping
L1
Elementary Estimator for sGGMs
Joint Estimation
Joint Elementary Estimator for Multiple sGGMs
Knowledge as Weights
Proposed: JEEK Estimator

Figure 1. Basic idea of JEEK.

However, even though this backward mapping has a simple closed form for GGM, the backward mapping is normally not well-defined in high-dimensional settings. When given the sample covariance $\widehat{\Sigma}$, we cannot just compute the vanilla MLE solution as $[\widehat{\Sigma}]^{-1}$ for GGM since $\widehat{\Sigma}$ is rank-deficient when $p > n$. Therefore Yang et al. (Yang et al., 2014b) used carefully constructed proxy backward maps as $\widehat{\theta}_n = [T_v(\widehat{\Sigma})]^{-1}$ that is both available in closed-form, and well-defined in high-dimensional settings for GGMs. We introduce the details of $[T_v(\widehat{\Sigma})]^{-1}$ and its statistical property in Section 11. Now Eq. (10.7) becomes the following closed-form estimator for learning sparse Gaussian graphical models (Yang et al., 2014b):

$$\begin{aligned}&\underset{\Omega}{\operatorname{argmin}}||\Omega||_{1,,\text{off}}\\&\text{subject to:}||\Omega - [T_v(\widehat{\Sigma})]^{-1}||_{\infty,\text{off}} \leq \lambda_n\end{aligned} \tag{2.5}$$

Eq. (2.5) is a special case of Eq. (10.7), in which $\mathcal{R}(\cdot)$ is the off-diagonal ℓ_1-norm and the precision matrix Ω is the θ we search for. When $\mathcal{R}(\cdot)$ is the ℓ_1-norm, the solution of Eq. (10.7) (and Eq. (2.5)) just needs to perform entry-wise thresholding operations on $\widehat{\theta}_n$ to ensure the desired sparsity structure of its final solution.

3. Proposed Method: JEEK

In applications of Gaussian graphical models, we typically have more information than just the data samples themselves. This paper aims to propose a simple, scalable and theoretically-guaranteed joint estimator for estimating multiple sGGMs with additional knowledge in large-scale situations.

3.1. A Joint EE (JEE) Formulation

We first propose to jointly estimate multiple related sGGMs from K data blocks using the following formulation:

$$\underset{\Omega^{(1)},\Omega^{(2)},\ldots,\Omega^{(K)}}{\operatorname{argmin}} \sum_{i=1}^{K} \mathcal{L}(\Omega^{(i)}) + \lambda_n \mathcal{R}(\Omega^{(1)},\Omega^{(2)},\ldots,\Omega^{(K)}) \tag{3.1}$$

where $\Omega^{(i)}$ denotes the precision matrix for i-th task. $\mathcal{L}(\Omega) = -\log\det(\Omega) + <\Omega, \widehat{\Sigma}>$ describes the negative log-likelihood function in sGGM. $\Omega^{(i)} \succ 0$ means that $\Omega^{(i)}$ needs to be a positive definite matrix. $\mathcal{R}(\cdot)$ represents a decomposable regularization function enforcing sparsity and structure assumptions (details in Section (3.2)).

For ease of notation, we denote that $\Omega^{tot} = (\Omega^{(1)},\Omega^{(2)},\ldots,\Omega^{(K)})$ and $\Sigma^{tot} = (\Sigma^{(1)},\Sigma^{(2)},\ldots,\Sigma^{(K)})$. Ω^{tot} and Σ^{tot} are both $p \times Kp$ matrices (i.e., Kp^2 parameters to estimate). Now define an inverse function as $\text{inv}(A^{tot}) := ({A^{(1)}}^{-1}, {A^{(2)}}^{-1}, \ldots, {A^{(K)}}^{-1})$, where A_{tot} is a given $p \times Kp$ matrix with the same structure as Σ_{tot}. Then we rewrite Eq. (3.1) into the following form:

$$\underset{\Omega^{tot}}{\operatorname{argmin}}\mathcal{L}(\Omega^{tot}) + \lambda_n \mathcal{R}(\Omega^{tot}) \tag{3.2}$$

Now connecting Eq. (3.2) to Eq. (2.2) and Eq. (10.7), we propose the following joint elementary estimator (JEE) for learning multiple sGGMs:

$$\begin{aligned}&\underset{\Omega^{tot}}{\operatorname{argmin}}\, \mathcal{R}(\Omega^{tot})\\&\text{subject to: } \mathcal{R}^*(\Omega^{tot} - \widehat{\Omega}^{tot}_{n_{tot}}) \leq \lambda_n\end{aligned} \tag{3.3}$$

The fundamental component in Eq. (10.7) for the single context sGGM was to use a well-defined proxy function to approximate the vanilla MLE solution (named as the backward mapping for exponential family distributions) (Yang et al., 2014b). The proposed proxy $\widehat{\theta}_n = [T_v(\widehat{\Sigma})]^{-1}$ is both well-defined under high-dimensional situations and also has a simple closed-form. Following a similar idea, when learning multiple sGGMs, we propose to use $inv(T_v(\widehat{\Sigma}^{tot}))$ for $\widehat{\Omega}^{tot}_{n_{tot}}$ and get the following joint elementary estimator:

$$\begin{aligned}&\underset{\Omega^{tot}}{\operatorname{argmin}}\mathcal{R}(\Omega^{tot})\\&\text{Subject to: } \mathcal{R}^*(\Omega^{tot} - inv(T_v(\widehat{\Sigma}^{tot}))) \leq \lambda_n\end{aligned} \tag{3.4}$$

3.2. Knowledge as Weight (KW-Norm)

The main goal of this paper is to design a principled strategy to incorporate existing knowledge (other than samples or structured assumptions) into the multi-sGGM formulation. We consider two factors in such a design:

(1) When learning multiple sGGMs jointly from real-world applications, it is often of great scientific interests to model and learn context-specific graph variations explicitly, because such variations can "fingerprint" important markers in domains like cognition (Ideker & Krogan, 2012) or pathology (Kelly et al., 2012). Therefore we design to share parameters between different contexts. Mathematically, we model $\Omega^{(i)}$ as two parts:

$$\Omega^{(i)} = \Omega_I^{(i)} + \Omega_S \tag{3.5}$$

where $\Omega_I^{(i)}$ is the individual precision matrix for context i and Ω_S is the shared precision matrix between contexts. Again, for ease of notation we denote $\Omega_I^{tot} = (\Omega_I^{(1)}, \Omega_I^{(2)}, \dots, \Omega_I^{(K)})$ and $\Omega_S^{tot} = (\Omega_S, \Omega_S, \dots, \Omega_S)$.

(2) We represent additional knowledge as positive weight matrices from $\mathbb{R}^{p \times p}$. More specifically, we represent the knowledge of the task-specific graph as weight matrix $\{W^{(i)}\}$ and W_S representing existing knowledge of the shared network. The positive matrix-based representation is a powerful and flexible strategy that can describe many possible forms of existing knowledge. In Section (12), we provide a few different designs of $\{W^{(i)}\}$ and W_S for real-world applications. In total, we have weight matrices $\{W_I^{(1)}, W_I^{(2)}, \dots, W_I^{(K)}, W_S\}$ to represent additional knowledge. To simplify notations, we denote $W_I^{tot} = (W_I^{(1)}, W^{(2)}, \dots, W_I^{(K)})$ and $W_S^{tot} = (W_S, W_S, \dots, W_S)$.

Now we propose the following knowledge as weight norm (kw-norm) combining the above two:

$$\mathcal{R}(\Omega^{tot}) = ||W_I^{tot} \circ \Omega_I^{tot}||_1 + ||W_S^{tot} \circ \Omega_S^{tot}||_1 \qquad (3.6)$$

Here the Hadamard product $\circ$ is the element-wise product between two matrices i.e. $[A \circ B]_{ij} = A_{ij}B_{ij}$.

The kw-norm(Eq. (3.6)) has the following three properties:

- (i) kw-norm is a norm function if and only if any entries in W_I^{tot} and W_S^{tot} do not equal to 0.
- (ii) If the condition in (i) holds, kw-norm is a decomposable norm.
- (iii) If the condition (i) holds, the dual norm of kw-norm is $\mathcal{R}^*(u) = \max(||\frac{1}{W_I^{tot}} \circ u||_\infty, ||\frac{1}{W_S^{tot}} \circ u||_\infty)$ [2]

Section 10.1 provides proofs of the above claims.

3.3. JEE with Knowledge (JEEK)

Plugging Eq. (3.6) to Eq. (3.4), we obtain the following formulation of JEEK for learning multiple related sGGMs from heterogeneous samples:

$$\begin{aligned} &\operatorname*{argmin}_{\Omega_I^{tot}, \Omega_S^{tot}} ||W_I^{tot} \circ \Omega_I^{tot}||_1 + ||W_S^{tot} \circ \Omega_S^{tot}|| \\ {}^{2}\ \text{Subject to: } & ||\frac{1}{W_I^{tot}} \circ (\Omega^{tot} - inv(T_v(\widehat{\Sigma}^{tot})))||_\infty \le \lambda_n \\ & ||\frac{1}{W_S^{tot}} \circ (\Omega^{tot} - inv(T_v(\widehat{\Sigma}^{tot})))||_\infty \le \lambda_n \\ & \Omega^{tot} = \Omega_S^{tot} + \Omega_I^{tot} \end{aligned} \qquad (3.7)$$

In Section 4, we theoretically prove that the statistical convergence rate of JEEK achieves the same sharp convergence rate as the state-of-the-art estimators for multi-task sGGMs. Our proofs are inspired by the unified framework of the high-dimensional statistics (Negahban et al., 2009).

3.4. Solution of JEEK:

A huge computational advantage of JEEK (Eq. (3.7)) is that it can be decomposed into $p \times p$ independent small linear programming problems. To simplify notations, we denote $\Omega_I^{(i)}{}_{j,k}$ (the $\{j,k\}$-th entry of $\Omega^{(i)}$) as a_i. Similarly we denote $\Omega_{S j,k}$ as b and $[T_v(\widehat{\Sigma}^{(i)})]^{-1}_{j,k}$ be c_i. Similarly we denote $W^{(i)}_{j,k} = w_i$ and $W^S_{j,k} = w_s$. "A group of entries" means a set of parameters $\{a_1, \dots, a_K, b\}$ for certain j, k.

In order to estimate $\{a_1, \dots, a_K, b\}$, JEEK (Eq. (3.7)) can be decomposed into the following formulation for a certain j, k :

$$\begin{aligned} &\operatorname*{argmin}_{a_i, b} \sum_i |w_i a_i| + K|w_s b| \\ {}^{2}\ \text{Subject to: } & |a_i + b - c_i| \le \lambda_n \min(w_i, w_s), \\ & i = 1, \dots, K \end{aligned} \qquad (3.8)$$

Eq. (3.8) can be easily converted into a linear programming form of Eq. (8.1) with only $K+1$ variables. The time complexity of Eq. (3.8) is $O(K^4)$. Considering JEEK has a total $p(p-1)/2$ of such subproblems to solve, the computational complexity of JEEK (Eq. (3.7)) is therefore $O(p^2K^4)$. We summarize the optimization algorithm of JEEK in Algorithm 1 (details in Section (8.2)).

4. Theoretical Analysis

KW-Norm: We presented the three properties of kw-norm in Section 3.2. The proofs of these three properties are included in Section (10.1).

Theoretical error bounds of Proxy Backward Mapping: (Yang et al., 2014b) proved that when ($p \ge n$), the proxy backward mapping $[T_v(\widehat{\Sigma})]^{-1}$ used by EE-sGGM achieves the sharp convergence rate to its truth (i.e., by proving $||T_v(\widehat{\Sigma}))^{-1} - \Sigma^{*-1}||_\infty = O(\sqrt{\frac{\log p}{n}})$). The proof was extended from the previous study (Rothman et al., 2009) that devised $T_v(\widehat{\Sigma})$ for estimating covariance matrix consistently in high-dimensional situations. See detailed proofs in Section 11.3. To derive the statistical error bound of JEEK, we need to assume that $inv(T_v(\widehat{\Sigma}^{tot}))$ are well-defined. This is ensured by assuming that the true $\Omega^{(i)^*}$ satisfy the conditions defined in Section (10.1).

[2] In our previous version, we mistakenly wrote $\mathcal{R}^*(u) = \max(||W_I^{tot} \circ u||_\infty, ||W_S^{tot} \circ u||_\infty)$. We correct relevant equations here. Our code implementation was correct, thus not change.

Algorithm 1. Joint Elementary Estimator with additional knowledge (JEEK) for Multi-task sGGMs

Input: Data sample matrix $\mathbf{X}^{(i)}$ ($i = 1$ **to** K), regularization hyperparameter λ_n, Knowledge weight matrices $\{W_I^{(i)}, W_S\}$ and **LP(.)** (a linear programming solver)

Output: $\{\Omega^{(i)}\}$ ($i = 1$ **to** K)

```
for $i = 1$ to $K$ do
    Initialize $\widehat{\Sigma}^{(i)} = \frac{1}{n_i - 1}\sum_{s=1}^{n_i}(\mathbf{X}_{s,}^{(i)} - \widehat{\mu}^{(i)})(\mathbf{X}_{s,}^{(i)} - \widehat{\mu}^{(i)})^T$
    (the sample covariance matrix of $\mathbf{X}^{(i)}$)
    Initialize $\Omega^{(i)} = \mathbf{0}_{p \times p}$
    Calculate the proxy backward mapping $[T_v(\widehat{\Sigma}^{(i)})]^{-1}$
end for
for $j = 1$ to $p$ do
    for $k = 1$ to $j$ do
        $c_i = [T_v(\widehat{\Sigma}^{(i)})]^{-1}_{j,k}$
        $w_i = W^{(i)}_{j,k}$
        $w_s = W_{S\,j,k}$
        $a_i, b = \mathbf{LP}(w_i, w_s, c_i, \lambda_n)$ where $i = 1, \dots, K$ and
        LP(.) solves Eq. (3.8)
        for $i = 1$ to $K$ do
            ${\Omega^{(i)}}_{j,k} = {\Omega^{(i)}}_{k,j} = a_i + b$
            ${\Omega_I^{(i)}}_{j,k} = a_i$
            $\Omega_{S\,j,k} = b$
        end for
    end for
end for
```

Theoretical error bounds of JEEK: We now use the high-dimensional analysis framework from (Negahban et al., 2009), three properties of kw-norm, and error bounds of backward mapping from (Rothman et al., 2009; Yang et al., 2014b) to derive the statistical convergence rates of JEEK. Detailed proofs of the following theorems are in Section 4 .

Before providing the theorem, we need to define the structural assumption, the IS-Sparsity, we assume for the parameter truth.

(IS-Sparsity): The 'true' parameter of Ω^{tot*} can be decomposed into two clear structures–$\{\Omega_I^{tot*}$ and $\Omega_S^{tot*}\}$. Ω_I^{tot*} is exactly sparse with k_i non-zero entries indexed by a support set S_I and Ω_S^{tot*} is exactly sparse with k_s non-zero entries indexed by a support set S_S. $S_I \bigcap S_S = \emptyset$. All other elements equal to 0 (in $(S_I \bigcup S_S)^c$).

Theorem 4.1. *Consider Ω^{tot} whose true parameter Ω^{tot*} satisfies the **(IS-Sparsity)** assumption. Suppose we compute the solution of Eq. (3.7) with a bounded λ_n such that $\lambda_n \geq \max(||\frac{1}{W_I^{tot}} \circ (\Omega^{tot*} - inv(T_v(\widehat{\Sigma}^{tot})))||_\infty, ||\frac{1}{W_S^{tot}} \circ (\Omega^{tot*} - inv(T_v(\widehat{\Sigma}^{tot})))||_\infty)$, then the estimated solution $\widehat{\Omega}^{tot}$ satisfies the following error bounds:*

$$||\widehat{\Omega}^{tot} - \Omega^{tot*}||_F \leq 4\sqrt{k_i + k_s}\lambda_n$$

$$\max(||\frac{1}{W_I^{tot}} \circ (\widehat{\Omega}^{tot} - \Omega^{tot*})||_\infty, ||\frac{1}{W_S^{tot}} \circ (\widehat{\Omega}^{tot} - \Omega^{tot*}||_\infty) \leq 2\lambda_n$$

$$||\frac{1}{W_I^{tot}} \circ (\widehat{\Omega}_I^{tot} - \Omega_I^{tot*})||_1 + ||\frac{1}{W_S^{tot}} \circ (\widehat{\Omega}_S^{tot} - \Omega_S^{tot*})||_1 \leq 8(k_i + k_s)\lambda_n \tag{4.1}$$

Proof. See detailed proof in Section 10.2 □

Theorem (4.1) provides a general bound for any selection of λ_n. The bound of λ_n is controlled by the distance between Ω^{tot*} and $inv(T_v(\widehat{\Sigma}^{tot}))$. We then extend Theorem (4.1) to derive the statistical convergence rate of JEEK. This gives us the following corollary:

Corollary 4.2. *Suppose the high-dimensional setting, i.e., $p > \max(n_i)$. Let $v := a\sqrt{\frac{\log(Kp)}{n_{tot}}}$. Then for $\lambda_n := \frac{8\kappa_1 a}{\kappa_2}\sqrt{\frac{\log(Kp)}{n_{tot}}}$ and $n_{tot} > c\log Kp$, with a probability of at least $1 - 2C_1\exp(-C_2 Kp\log(Kp))$, the estimated optimal solution $\widehat{\Omega}^{tot}$ has the following error bound:*

$$||\widehat{\Omega}^{tot} - \Omega^{tot*}||_F \leq \frac{16\kappa_1 a \max\limits_{j,k}({W_I^{tot}}_{j,k}, {W_S^{tot}}_{j,k})}{\kappa_2}\sqrt{\frac{(k_i + k_s)\log(Kp)}{n_{tot}}} \tag{4.2}$$

where a, c, κ_1 and κ_2 are constants.

Proof. See detailed proof in Section 10.2.2 (especially from Eq. (10.14) to Eq. (10.22)). □

Bayesian View of JEEK: In Section (9) we provide a direct Bayesian interpretation of JEEK through the perspective of hierarchical Bayesian modeling. Our hierarchical Bayesian interpretation nicely explains the assumptions we make in JEEK.

5. Connecting to Relevant Studies

JEEK is closely related to a few state-of-the-art studies summarized in Table 1. We compare the time complexity and functional properties of JEEK versus these studies.

NAK: (Bu & Lederer, 2017) For the single task sGGM, one recent study (Bu & Lederer, 2017) (following ideas from (Shimamura et al., 2007)) proposed to integrating Additional Knowledge (NAK)into estimation of graphical models through a weighted Neighbourhood selection formulation (NAK) as: $\widehat{\beta}^j = \underset{\beta, \beta_j = 0}{\text{argmin}} \frac{1}{2}||X^j - X\beta||_2^2 + ||\mathbf{r}_j \circ \beta||_1$.

Method	JEEK	W-SIMULE	JGL	FASJEM	NAK (run K times)
Time Complexity	$O(K^4p^2)$ $(\Rightarrow O(K^4)$ if parallelizing completely)	$O(K^4p^5)$	$O(T \times Kp^3)$	$O(T \times Kp^2)$	$O(Knp^3 + Kp^4)$
Additional Knowledge	YES	YES	NO	NO	YES

Table 1. Compare JEEK versus baselines. Here T is the number of iterations.

NAK is designed for estimating brain connectivity networks from homogeneous samples and incorporate distance knowledge as weight vectors. [3] In experiments, we compare JEEK to NAK (by running NAK R package K times) on multiple synthetic datasets of simulated samples about brain regions. The data simulation strategy was suggested by (Bu & Lederer, 2017). Same as the NAK (Bu & Lederer, 2017), we use the spatial distance among brain regions as additional knowledge in JEEK.

W-SIMULE: (Singh et al., 2017) Like JEEK, one recent study (Singh et al., 2017) of multi-sGGMs (following ideas from (Wang et al., 2017b)) also assumed that $\Omega^{(i)} = \Omega_I^{(i)} + \Omega_S$ and incorporated spatial distance knowledge in their convex formulation for joint discovery of heterogeneous neural connectivity graphs. This study, with name W-SIMULE (Weighted model for Shared and Individual parts of MULtiple graphs Explicitly) uses a weighted constrained ℓ_1 minimization:

$$\underset{\Omega_I^{(i)},\Omega_S}{\operatorname{argmin}} \sum_i ||W \circ \Omega_I^{(i)}||_1 + \epsilon K ||W \circ \Omega_S||_1 \tag{5.1}$$
$$\text{Subject to: } ||\Sigma^{(i)}(\Omega_I^{(i)} + \Omega_S) - I||_\infty \leq \lambda_n,\ i = 1, \dots, K$$

W-SIMULE simply includes the additional knowledge as a weight matrix W. [4]

Different from W-SIMULE, JEEK separates the knowledge of individual context and the shared using different weight matrices. While W-SIMULE also minimizes a weighted $\ell 1$ norm, its constraint optimization term is entirely different from JEEK. The formulation difference makes the optimization of JEEK much faster and more scalable than W-SIMULE (Section (6)). We have provided a complete theoretical analysis of error bounds of JEEK, while W-SIMULE provided no theoretical results. Empirically, we compare JEEK with W-SIMULE R package from (Singh et al., 2017) in the experiments.

JGL: (Danaher et al., 2013): Regularized MLE based multi-sGGMs Studies mostly follow the so called joint graphical lasso (JGL) formulation as Eq. (5.2):

$$\underset{\Omega^{(i)} \succ 0}{\operatorname{argmin}} \sum_{i=1}^{K} (-L(\Omega^{(i)}) + \lambda_n \sum_{i=1}^{K} ||\Omega^{(i)}||_1 \tag{5.2}$$
$$+ \lambda'_n \mathcal{R}'(\Omega^{(1)}, \Omega^{(2)}, \dots, \Omega^{(K)})$$

$\mathcal{R}'(\cdot)$ is the second penalty function for enforcing some structural assumption of group property among the multiple graphs. One caveat of JGL is that $\mathcal{R}'(\cdot)$ cannot model explicit additional knowledge. For instance,it can not incorporate the information of a few known hub nodes shared by the contexts. In experiments, we compare JEEK to JGL-co-hub and JGL-perturb-hub toolbox provided by (Mohan et al., 2013).

FASJEM: (Wang et al., 2017a) One very recent study extended JGL using so-called Elementary superposition-structured moment estimator formulation as Eq. (5.3):

$$\underset{\Omega_{tot}}{\operatorname{argmin}} ||\Omega_{tot}||_1 + \epsilon \mathcal{R}'(\Omega_{tot})$$
$$s.t. ||\Omega_{tot} - \text{inv}(T_v(\widehat{\Sigma}_{tot}))||_\infty \leq \lambda_n \tag{5.3}$$
$$\mathcal{R}'^*(\Omega_{tot} - \text{inv}(T_v(\widehat{\Sigma}_{tot}))) \leq \epsilon \lambda_n$$

FASJEM is much faster and more scalable than the JGL estimators. However like JGL estimators it can not model additional knowledge and its optimization needs to be carefully re-designed for different $\mathcal{R}'(\cdot)$. [5]

Both NAK and W-SIMULE only explored the formulation for estimating neural connectivity graphs using spatial information as additional knowledge. Differently our experiments (Section (6)) extend the weight-as-knowledge formulation on weights as distance, as shared hub knowledge, as perturbed hub knowledge, and as nodes' grouping information (e.g., multiple genes are known to be in the same pathway). This has largely extends the previous studies in showing the real-world adaptivity of the proposed formulation. JEEK elegantly formulates existing knowledge based on the problem at hand and avoid the need to design knowledge-specific optimization.

6. Experiments

We empirically evaluate JEEK and baselines on four types of datasets, including two groups of synthetic data, one real-

[3] Here $\widehat{\beta}^j$ indicates the sparsity of j-th column of a single $\widehat{\Omega}$. Namely, $\widehat{\beta}^j_k = 0$ if and only if $\widehat{\Omega}_{k,j} = 0$. $\mathbf{r}_j$ is a weight vector as the additional knowledge The NAK formulation can be solved by a classic Lasso solver like glmnet.

[4] It can be solved by any linear programming solver and can be column-wise paralleled. However, it is very slow when $p > 200$ due to the expensive computation cost $O(K^4p^5)$.

[5] FASJEM extends JGL into multiple independent group-entry wise optimization just like JEEK. Here $\mathcal{R}'^*(\cdot)$ is the dual norm of $\mathcal{R}'(\cdot)$. Because (Wang et al., 2017a) only designs the optimization of two cases (group,2 and group,inf), we can not use it as a baseline.

world fMRI dataset for brain connectivity estimation and one real-world genomics dataset for estimating interaction among regulatory genes (results in Section (6.2)). In order to incorporating various types of knowledge, we provide five different designs of the weight matrices in Section 12. Details of experimental setup, metrics and hyper-parameter tuning are included in Section (13.1). Baselines used in our experiments have been explained in details by Section (5). We also use JEEK with no additional knowledge (JEEK-NK) as a baseline.

JEEK is available as the R package 'jeek' in CRAN.

6.1. Experiment: Simulated Samples with Known Hubs as Knowledge

Inspired the JGL-co-hub and JGL-perturb-hub toolbox (JGL-node) provided by (Mohan et al., 2013), we empirically show JEEK's ability to model known co-hub or perturbed-hub nodes as knowledge when estimating multiple sGGMs. We generate multiple simulated Gaussian datasets through the random graph model (Rothman et al., 2008) to simulate both the co-hub and perturbed-hub graph structures (details in 14.1). We use JGL-node package, W-SIMULE and JEEK-NK as baselines for this set of experiments. The weights in $\{W_I^{tot}, W_S^{tot}\}$ are designed using the strategy proposed in Section (12).

We use AUC score (to reflect the consistency and variance of a method's performance when varying its important hyper-parameter) and computational time cost to compare JEEK with baselines. We compare all methods on many simulated cases by varying p from the set $\{100, 200, 300, 400, 500\}$ and the number of tasks K from the set $\{2, 3, 4\}$. In Figure 2 and Figure 3(a)(b), JEEK consistently achieves higher AUC-scores than the baselines JGL, JEEK-NK and W-SIMULE for all cases. JEEK is more than 10 times faster than the baselines on average. In Figure 2, for each $p > 300$ case (with $n = p/2$), W-SIMULE takes more than one month and JGL takes more than one day. Therefore we can not show them with $p > 300$.

6.2. Experiment: Gene Interaction Network from Real-World Genomics Data

Next, we apply JEEK and the baselines on one real-world biomedical data about gene expression profiles across two different cell types. We explored two different types of knowledge: (1) Known edges and (2) Known group about genes. Figure 3(c) shows that JEEK has lower time cost and recovers more interactions than baselines (higher number of matched edges to the existing bio-databases.). More results are in Appendix Section (14.2) and the design of weight matrices for this case is in Section (12).

6.3. Experiment: Simulated Data about Brain Connectivity with Distance as Knowledge

Following (Bu & Lederer, 2017), we use one known Euclidean distance between human brain regions as additional knowledge W and use it to generate multiple simulated datasets (details in Section 14.3). We compare JEEK with the baselines regarding (a) Scalability (computational time cost), and (b) effectiveness (F1-score, because NAK package does not allow AUC calculation). For each simulation case, the computation time for each estimator is the summation of a method's execution time over all values of λ_n. Figure 4(a)(b) show clearly that JEEK outperforms its baselines. JEEK has a consistently higher F1-Score and is almost 6 times faster than W-SIMULE in the high dimensional case. JEEK performs better than JEEK-NK, confirming the advantage of integrating additional distance knowledge. While NAK is fast, its F1-Score is nearly 0 and hence, not useful for multi-sGGM structure learning.

6.4. Experiment: Functional Connectivity Estimation from Real-World Brain fMRI Data

We evaluate JEEK and relevant baselines for a classification task on one real-world publicly available resting-state fMRI dataset: ABIDE(Di Martino et al., 2014). The ABIDE data aims to understand human brain connectivity and how it reflects neural disorders (Van Essen et al., 2013). ABIDE includes two groups of human subjects: autism and control, and therefore we formulate it as $K = 2$ graph estimation. We utilize the spatial distance between human brain regions as additional knowledge for estimating functional connectivity edges among brain regions. We use Linear Discriminant Analysis (LDA) for a downstream classification task aiming to assess the ability of a graph estimator to learn the differential patterns of the connectome structures. (Details of the ABIDE dataset, baselines, design of the additional knowledge W matrix, cross-validation and LDA classification method are in Section (14.4).)

Figure 4(c) compares JEEK and three baselines: JEEK-NK, W-SIMULE and W-SIMULE with no additional knowledge (W-SIMULE-NK). JEEK yields a classification accuracy of 58.62% for distinguishing the autism subjects versus the control subjects, clearly outperforming JEEK-NK and W-SIMULE-NK. JEEK is roughly 7 times faster than the W-SIMULE estimators, locating at the top left region in Figure 4(c) (higher classification accuracy and lower time cost). We also experimented with variations of the W matrix and found the classification results are fairly robust to the variations of W (Section (14.4)).

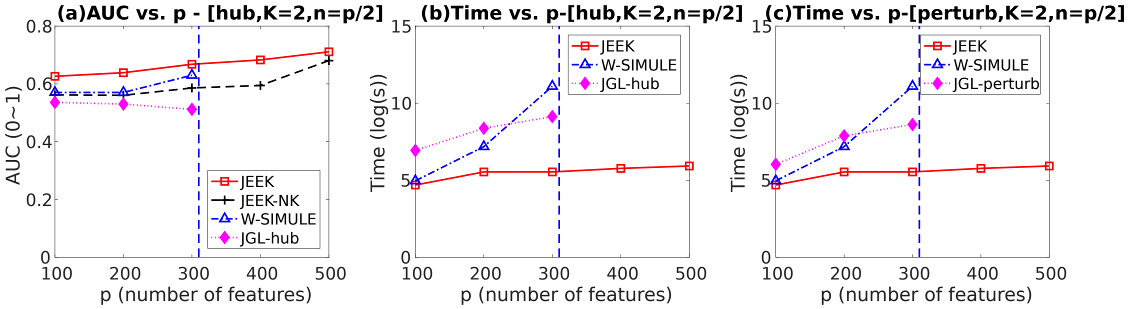


Figure 2. Performance comparison on simulation Datasets using co-Hub Knowledge: AUC vs. Time when varying number of nodes p.

7. Conclusions

We propose a novel method, JEEK, to incorporate additional knowledge in estimating multi-sGGMs. JEEK achieves the same asymptotic convergence rate as the state-of-the-art. Our experiments has showcased using weights for describing pairwise knowledge among brain regions, for shared hub knowledge, for perturbed hub knowledge, for describing group information among nodes (e.g., genes known to be in the same pathway), and for using known interaction edges as the knowledge.

Acknowledgement: *This work was supported by the National Science Foundation under NSF CAREER award No. 1453580. Any Opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect those of the National Science Foundation.*

Appendix:

8. More about Method

Notations: $X^{(i)}_{n_i \times p}$ is the data matrix for the i-th task, which includes n_i data samples being described by p different feature variables. Then $n_{tot} = \sum\limits_{i=1}^{K} n_i$ is the total number of data samples. We use notation $\Omega^{(i)}$ for the precision matrices and $\widehat{\Sigma}^{(i)}$ for the estimated covariance matrices. Given a p-dimensional vector $\mathbf{x} = (x_1, x_2, \dots, x_p)^T \in \mathbb{R}^p$, we denote the l_1-norm of x as $||\mathbf{x}||_1 = \sum\limits_i |x_i|$. $||\mathbf{x}||_\infty = \max\limits_i |x_i|$ is the l_∞-norm of $\mathbf{x}$. Similarly, for a matrix X, let $||X||_1 = \sum\limits_{i,j} |X_{i,j}|$ be the ℓ_1-norm of X and $||X||_\infty = \max\limits_{i,j} |X_{i,j}|$ be the ℓ_∞-norm of X. $||X||_F = \sqrt{\sum\limits_i \sum\limits_j X^2_{i,j}}$

8.1. More about Solving JEEK

In Eq. (3.8), let $a_i = a_i^+ - a_i^-$ and $b = b^+ - b^-$. If $a_i \geq 0$, then $a_i^+ = a_i$ and $a_i^- = 0$. If $a_i < 0$, then $a_i^+ = 0$ and $a_i^- = -a_i$. The b^+ and b^- have the similar definition. Then Eq. (3.8) can be solved by the following small linear programming problem.

$$\operatorname*{argmin}_{a_i, b} \sum_i (w_i a_i^+ + w_i a_i^-) + K w_s b^+ + K w_s b^-$$

$$\begin{aligned}
\text{Subject to: } & a_i^+ - a_i^- + b^+ - b^- \leq c_i + \lambda_n \min(w_i, w_s), \\
& a_i^+ - a_i^- + b^+ - b^- \geq c_i - \lambda_n \min(w_i, w_s), \\
& a_i^+, a_i^-, b^+, b^- \geq 0 \\
& i = 1, \dots, K
\end{aligned}$$

8.2. JEEK is Group entry-wise and parallelizing optimizable

JEEK can be easily paralleled. Essentially we just need to revise the "For loop" of step 6 and step 7 in Algorithm 1 into, for instance, "entry per machine" "entry per core". Now We prove that JEEK is group entry-wise and parallelizing optimizable. We prove that our estimator can be optimized asynchronously in a group entry-wise manner.

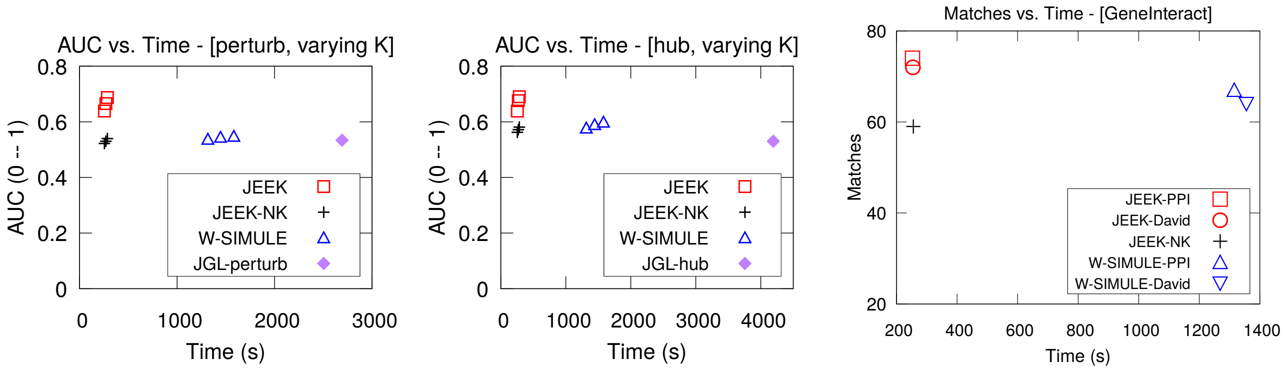
Theorem 8.1. ***(JEEK is Group entry-wise optimizable)*** *Suppose we use JEEK to infer multiple inverse of covariance matrices summarized as $\widehat{\Omega}_{tot}$. $\{[\widehat{\Omega}_I^{(i)}]_{j,k}, [\widehat{\Omega}_S]_{j,k} | i = 1, \dots, K\}$. describes a group of $K+1$ entries at (j,k) position. Varying $j \in \{1, 2, \dots, p\}$ and $k \in \{1, 2, \dots, p\}$, we have a total of $p \times p$ groups. If these groups are independently estimated by JEEK, then we have,*

$$\bigcup_{j=1}^{p} \bigcup_{k=1}^{p} \{([\widehat{\Omega}_I^{(i)}]_{j,k} + [\widehat{\Omega}_S]_{j,k}) | i = 1, \dots, K\} = \widehat{\Omega}_{tot}. \tag{8.1}$$

Proof. Eq. (3.8) are the small sub-linear programming problems on each group of entries. □

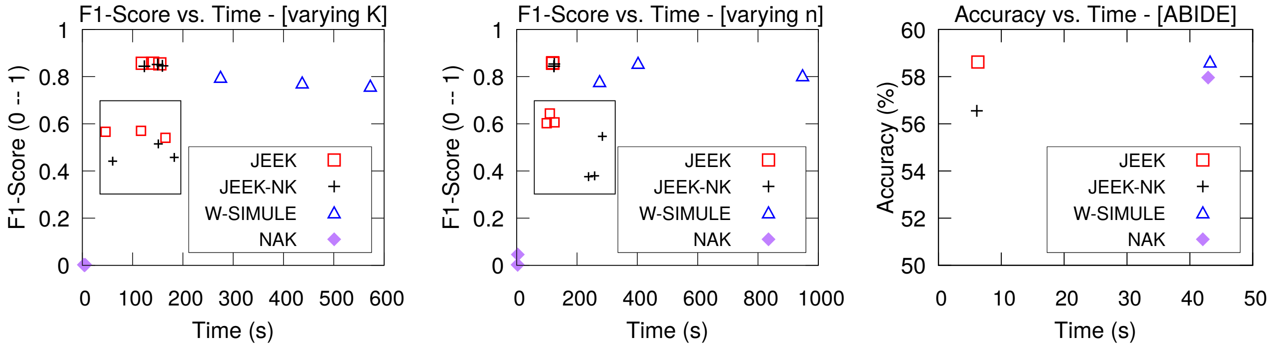
8.3. Extending JEEK with Structured Norms

We can add more flexibility into the JEEK by adding structured norms like those second normalization functions used



(a) (b) (c)

Figure 3. (a)(b) Performance comparison on simulation Datasets about hubs: AUC vs. Time when varying number of tasks K. (a) is the perturbed hub cases and (b) is for the co-hub cases. (c) Performance comparison on one real-world gene expression dataset with two cell types. Two type knowledge are used to cover one fifth of the nodes, therefore each method corresponds to two performance points.



(a) (b) (c)

Figure 4. Experimental Results on Simulated Brain Datasets and on ABIDE. (a) Performance obtained on simulated brain samples with respect to F1-score vs. computational time cost when varying the number of tasks K. (b) Performance obtained on simulated brain samples with respect to F1-score vs. computational time cost when varying the number of samples n. In both (a) and (b) the smaller box shows an enlarged view comparing JEEK and JEEK-NK points. All JEEK points are in the top left region indicating higher F1-score and lower computational cost. (c). On ABIDE, JEEK outperforms the baseline methods in both classification accuracy and running time cost. JEEK and JEEK-NK points in the top left region and JEEK points are higher in terms of y-axis positions.

in JGL. This will extend JEEK to the following formulation:

$$\begin{aligned}&\underset{\Omega_I^{tot},\Omega_S^{tot}}{\operatorname{argmin}} ||W_I^{tot} \circ \Omega_I^{tot}||_1 + ||W_S^{tot} \circ \Omega_S^{tot}|| + \epsilon\mathcal{R}'(\Omega^{tot})\\ &\text{Subject to: } ||\frac{1}{W_I^{tot}} \circ (\Omega^{tot} - inv(T_v(\widehat{\Sigma}^{tot})))||_\infty \leq \lambda_n\\ &\qquad ||\frac{1}{W_S^{tot}} \circ (\Omega^{tot} - inv(T_v(\widehat{\Sigma}^{tot})))||_\infty \leq \lambda_n\\ &\qquad \mathcal{R}^{*\prime}(\Omega^{tot}) \leq \epsilon\lambda_n\end{aligned} \tag{8.2}$$

Here, $\mathcal{R}'$ needs to consider Ω^{tot}. We propose two ways to solve Eq. (8.2). (1) The first is to use the parallelized proximal algorithm directly. However, this requires the kw-norm has a closed-form proximity, which has not been discovered. (2) In the second strategy we assume each weighted ℓ_1 norm (either $\Omega^{(i)}{}_I$ or Ω_S) in the objective of Eq. (8.2) as an indepedent regularizer. However, this increases the number of proximities we need to calculate per iteration to $K+1$. Both two solutions make the extend-JEEK algorithm less fast or less scalable. Therefore, We choose not to introduce this work in this paper.

9. Connecting to the Bayesian statistics

Our approach has a close connection to a hierarchical Bayesian model perspective. We show that the additional knowledge weight matrices are also the parameters of the prior distribution of $\Omega_I^{(i)}, \Omega_S$. In our formulationEq. (3.7), $W_I^{(i)}, W_S$ are the additional knowledge weight matrices. From a hierarchical Bayesian view, the first level of the prior is a Gaussian distribution and the second level is a Laplace distribution. In the following section, we show that

$W_I^{(i)}, W_S$ are also the parameters of Laplace distributions, which is a prior distribution of $\Omega_I^{(i)}, \Omega_S$.

Since by the definition, $\Omega_{I\ j,k}^{(i)}\Omega_{Sj,k} = 0$. There are only two possible situations:

Case I ($\Omega_{I\ j,k}^{(i)} = 0$):

$$X^{(i)}|\mu^{(i)}, \Omega^{(i)} \sim N(\mu^{(i)}, (\Omega^{(i)})^{-1}) \tag{9.1}$$

$$\Omega_{j,k}^{(i)}|\mu^{(i)}, W_{I\ j,k}^{(i)}, W_{Sj,k} = \Omega_{Sj,k}|\mu^{(i)}, W_{Sj,k} \tag{9.2}$$

$$\begin{aligned} &p(\Omega_{Sj,k}|\mu^{(i)}, W_{Sj,k}) \\ &\propto e^{-(W_{Sj,k}|\Omega_{Sj,k}|)} \end{aligned} \tag{9.3}$$

Here $\Omega_{Sj,k}|\mu^{(i)}, W_{Sj,k}$ follows a Laplace distribution with mean 0. $1/W_{Sj,k} > 0$ is the diversity parameter. The larger $W_{Sj,k}$ is, the distribution of $\Omega_{Sj,k}|\mu^{(i)}, W_{Sj,k}$ more likely concentrate on the 0. Namely, there will be the higher density for $\Omega_{Sj,k} = 0|\mu^{(i)}, W_{Sj,k}$.

Case II ($\Omega_{Sj,k} = 0$):

$$X^{(i)}|\mu^{(i)}, \Omega^{(i)} \sim N(\mu^{(i)}, (\Omega^{(i)})^{-1}) \tag{9.4}$$

$$\Omega_{j,k}^{(i)}|\mu^{(i)}, W_{I\ j,k}^{(i)}, W_{Sj,k} = \Omega_{I\ j,k}^{(i)}|\mu^{(i)}, W_{I\ j,k}^{(i)} \tag{9.5}$$

$$\begin{aligned} &p(\Omega_{I\ j,k}^{(i)}|\mu^{(i)}, W_{I\ j,k}^{(i)}) \\ &\propto e^{-(W_{I\ j,k}^{(i)}|\Omega_{I\ j,k}^{(i)}|)} \end{aligned} \tag{9.6}$$

Here $\Omega_{I\ j,k}^{(i)}|\mu^{(i)}, W_{I\ j,k}^{(i)}$ follows a Laplace distribution with mean 0. $1/W_{I\ j,k}^{(i)} > 0$ is the diversity parameter. The larger $W_{I\ j,k}^{(i)}$ is, the distribution of $\Omega_{I\ j,k}^{(i)}|\mu^{(i)}, W_{I\ j,k}^{(i)}$ more likely concentrate on the 0. Namely, there will be the higher density for $\Omega_{I\ j,k}^{(i)} = 0|\mu^{(i)}, W_{I\ j,k}^{(i)}$.

Therefore, we can combine the above two cases into the following one equation.

$$\begin{aligned} &p(\Omega_{j,k}^{(i)}|\mu^{(i)}, W_{I\ j,k}^{(i)}, W_{Sj,k}) \\ &\propto e^{-(W_{I\ j,k}^{(i)}|\Omega_{I\ j,k}^{(i)}| + W_{Sj,k}|\Omega_{Sj,k}|)} \end{aligned} \tag{9.7}$$

Our final hierarchical Bayesian formulation consists of the Eq. (9.1) and Eq. (9.7). This model is a generalization of the model considered in the seminal paper on the Bayesian lasso(Park & Casella, 2008). The parameters $W_{I\ j,k}^{(i)}, W_{Sj,k}$ in our general model are hyper-parameters that specify the shape of the prior distribution of each edges in $\Omega^{(i)}$. The negative log-posterior distribution of $\Omega^{(i)}$ is now given by:

$$\begin{aligned} &-\log(\mathbb{P}(\Omega^{(i)}|X^{(i)}, \mu^{(i)}, W_{I\ j,k}^{(i)}, W_{Sj,k})) \\ &\propto -\log(det(\Omega^{(i)^{-1}})) + <\Omega^{(i)}, \widehat{\Sigma}^{(i)}> \\ &+ \sum_{j,k}(W_{I\ j,k}^{(i)}|\Omega_{I\ j,k}^{(i)}| + W_S|\Omega_{Sj,k}|) \end{aligned} \tag{9.8}$$

Eq. (9.8) follows a weighted variation of Eq. (2.1).

10. More about Theoretical Analysis

10.1. Theorems and Proofs of three properties of kw-norm

In this sub-section, we prove the three properties of kw-norm used in Section 3.2. We then provide the convergence rate of our estimator based on these three properties.

- (i) kw-norm is a norm function if and only if any entries in W_I^{tot} and W_S^{tot} do not equal to 0.
- (ii) If the condition in (i) holds, kw-norm is a decomposable norm.
- (iii) If the condition in (i) holds, the dual norm of kw-norm is $\mathcal{R}^*(u) = \max(||W_I^{tot} \circ u||_\infty, ||W_S^{tot} \circ u||_\infty)$.

10.1.1. Norm:

First we prove the correctness of the argument that kw-norm is a norm function by the following theorem:

Theorem 10.1. *Eq. (3.6) is a norm if and only if* $\forall 1 \geq j, k \leq p, W_{I\ jk}^{(i)} \neq 0$, *and* $W_{Sj,k} \neq 0$.

This theorem gives the sufficient and necessary conditions to make kw-norm (Eq. (3.6)) a norm function.

10.1.2. Decomposable Norm:

Then we show that kw-norm is a decomposable norm within a certain subspace. Before providing the theorem, we give the structural assumption of the parameter.

(IS-Sparsity): The 'true' parameter for Ω^{tot*} (multiple GGM structures) can be decomposed into two clear structures–Ω_I^{tot*} and Ω_S^{tot*}. Ω_I^{tot*} is exactly sparse with k_i non-zero entries indexed by a support set S_I and Ω_S^{tot*} is exactly sparse with k_s non-zero entries indexed by a support set S_S. $S_I \bigcap S_S = \emptyset$. All other elements equal to 0 (in $(S_I \bigcup S_S)^c$).

Definition 10.2. *(IS-subspace)*

$$\mathcal{M}(S_I \bigcup S_S) = \{\theta_j = 0 | \forall j \notin S_I \bigcup S_S\} \tag{10.1}$$

Theorem 10.3. *Eq. (3.6) is a decomposable norm with respect to $\mathcal{M}$ and $\bar{\mathcal{M}}^\perp$*

10.1.3. Dual Norm of kw-Norm:

To obtain the final formulation Eq. (3.7) and its statistical convergence rate, we need to derive the dual norm formulation of kw-norm.

Theorem 10.4. *The dual norm of kw-norm (Eq. (3.6)) is*

$$\mathcal{R}^*(u) = \max(||\frac{1}{W_I^{tot}} \circ u||_\infty, ||\frac{1}{W_S^{tot}} \circ u||_\infty) \tag{10.2}$$

The details of the proof are as follows.

10.1.4. Proof of Theorem (10.1)

Lemma 10.5. *For kw-norm, ${W_I^{tot}}_{j,k} \neq 0$ and ${W_S^{tot}}_{j,k} \neq 0$ equals to ${W_I^{tot}}_{j,k} > 0$ and ${W_S^{tot}}_{j,k} > 0$.*

Proof. If ${W_I^{tot}}_{j,k} < 0$, then $|{W_I^{tot}}_{j,k}{\Omega_I^{tot}}_{j,k}| = |{W_I^{tot}}_{j,k}||{\Omega_I^{tot}}_{j,k}| = |-{W_I^{tot}}_{j,k}{\Omega_I^{tot}}_{j,k}|$. Notice that $-{W_I^{tot}}_{j,k} > 0$. □

Proof. To prove the kw-norm is a norm, by Lemma (11.2) the only thing we need to prove is that $f(x) = ||W \circ x||_1$ is a norm function if $W_{i,j} > 0$. 1. $f(ax) = ||aW \circ x||_1 = |a|||W \circ x||_1 = |a|f(x)$. 2. $f(x+y) = ||W \circ (x+y)||_1 = ||W \circ x + W \circ y||_1 \leq ||W \circ x||_1 + ||W \circ y||_1 = f(x) + f(y)$. 3. $f(x) \geq 0$ 4. If $f(x) = 0$, then $\sum |W_{i,j}x_{i,j}| = 0$. Since $W_{i,j} \neq 0$, $x_{i,j} = 0$. Therefore, $x = 0$. Based on the above, $f(x)$ is a norm function. Since summation of norm is still a norm function, kw-norm is a norm function. □

Furthermore, we have the following Lemma:

Lemma 10.6. *The dual norm of $f(x) = ||W \circ x||_1$ is*

$$||\frac{1}{W} \circ x||_\infty$$

.

Proof.

$$\begin{aligned} f^*(u) &= \sup_{||W \circ x||_1 \leq 1} < u, x > && (10.3) \\ &\leq \sup_{||W \circ x||_1 \leq 1} (\sum_{k=1,\dots,p} |w_k x_k|) \max_{k=1,\dots,p} |\frac{1}{w_k} u_k| && (10.4) \\ &= ||\frac{1}{W} \circ u||_\infty && (10.5) \end{aligned}$$

□

10.1.5. Proof of Theorem (10.3)

Proof. Assume $u \in \mathcal{M}$ and $v \in \bar{\mathcal{M}}^\perp$, $\mathcal{R}(u+v) = ||W_I^{tot} \circ (u_I + v_I)||_1 + ||W_S^{tot} \circ (u_S + v_S)||_1 = ||W_I^{tot} \circ u_I||_1 + ||W_S^{tot} \circ u_S||_1 + ||W_I^{tot} \circ v_I||_1 + ||W_S^{tot} \circ v_S||_1 = \mathcal{R}(u) + \mathcal{R}(v)$. Therefore, kw-norm is a decomposable norm with respect to the subspace pair $(\mathcal{M}, \bar{\mathcal{M}}^\perp)$. □

10.1.6. Proof of Theorem (10.4)

Proof. Suppose $\mathcal{R}(\theta) = \sum_{\alpha \in I} c_\alpha \mathcal{R}_\alpha(\theta_\alpha)$, where $\sum_{\alpha \in I} \theta_\alpha = \theta$. Then the dual norm $\mathcal{R}^*(\cdot)$ can be derived by the following equation.

$$\begin{aligned} \mathcal{R}^*(u) &= \sup_\theta \frac{< \theta, u >}{\theta} \\ &= \sup_{\theta_\alpha} \frac{\sum_\alpha < u, \theta_\alpha >}{\sum_\alpha c_\alpha \mathcal{R}_\alpha(\theta_\alpha)} \\ &= \sup_{\theta_\alpha} \frac{\sum_\alpha < u/c_\alpha, \theta_\alpha >}{\sum_\alpha \mathcal{R}_\alpha(\theta_\alpha)} \\ &\leq \sup_{\theta_\alpha} \frac{\sum_\alpha \mathcal{R}^*_\alpha(u/c_\alpha)\mathcal{R}(\theta_\alpha)}{\sum_\alpha \mathcal{R}_\alpha(\theta_\alpha)} \\ &\leq \max_{\alpha \in I} \mathcal{R}^*_\alpha(u)/c_\alpha. \end{aligned} \tag{10.6}$$

Therefore by Lemma (10.6), the dual norm of kw-norm is $\mathcal{R}^*(u) = \max(||W_I^{tot} \circ u||_\infty, ||W_S^{tot} \circ u||_\infty)$. □

10.2. Appendix: Proofs of Theorems about All Error Bounds of JEEK

10.2.1. Derivation of Theorem (4.1)

JEEK formulation Eq. (3.7) and EE-sGGM Eq. (2.5) are special cases of the following generic formulation:

$$\begin{aligned} &\operatorname*{argmin}_\theta \mathcal{R}(\theta) \\ &\text{subject to:} \mathcal{R}^*(\theta - \widehat{\theta}_n) \leq \lambda_n \end{aligned} \tag{10.7}$$

Where $\mathcal{R}^*(\cdot)$ is the dual norm of $\mathcal{R}(\cdot)$,

$$\mathcal{R}^*(v) := \sup_{u \neq 0} \frac{< u, v >}{\mathcal{R}(u)} = \sup_{\mathcal{R}(u) \leq 1} < u, v > . \tag{10.8}$$

Connecting Eq. (3.7) and Eq. (10.7), $\mathcal{R}()$ is the kw-norm. $\widehat{\theta}_n$ represents a close approximation of θ^*.

Following the unified framework (Negahban et al., 2009), we first decompose the parameter space into a subspace pair$(\mathcal{M}, \bar{\mathcal{M}}^\perp)$, where $\bar{\mathcal{M}}$ is the closure of $\mathcal{M}$. Here $\bar{\mathcal{M}}^\perp := \{v \in \mathbb{R}^p | < u, v >= 0, \forall u \in \bar{\mathcal{M}}\}$. $\mathcal{M}$ is the **model subspace** that typically has a much lower dimension than the original high-dimensional space. $\bar{\mathcal{M}}^{\perp}$ is the **perturbation subspace** of parameters. For further proofs, we assume the regularization function in Eq. (10.7) is **decomposable** w.r.t the subspace pair $(\mathcal{M}, \bar{\mathcal{M}}^{\perp})$.

(C1) $\mathcal{R}(u+v) = \mathcal{R}(u) + \mathcal{R}(v), \forall u \in \mathcal{M}, \forall v \in \bar{\mathcal{M}}^{\perp}$.

(Negahban et al., 2009) showed that most regularization norms are decomposable corresponding to a certain subspace pair.

Definition 10.7. ***Subspace Compatibility Constant*** *Subspace compatibility constant is defined as* $\Psi(\mathcal{M}, |\cdot|) := \sup\limits_{u \in \mathcal{M}\setminus\{0\}} \frac{\mathcal{R}(u)}{|u|}$ *which captures the relative value between the error norm* $|\cdot|$ *and the regularization function* $\mathcal{R}(\cdot)$.

For simplicity, we assume there exists a true parameter θ^* which has the exact structure w.r.t a certain subspace pair. Concretely:

(C2) $\exists$ a subspace pair $(\mathcal{M}, \bar{\mathcal{M}}^{\perp})$ such that the true parameter satisfies $\text{proj}_{\mathcal{M}^{\perp}}(\theta^*) = 0$

Then we have the following theorem.

Theorem 10.8. *Suppose the regularization function in Eq. (10.7) satisfies condition* ***(C1)****, the true parameter of Eq. (10.7) satisfies condition* ***(C2)****, and* λ_n *satisfies that* $\lambda_n \geq \mathcal{R}^*(\widehat{\theta}_n - \theta^*)$*. Then, the optimal solution* $\widehat{\theta}$ *of Eq. (10.7) satisfies:*

$$\mathcal{R}^*(\widehat{\theta} - \theta^*) \leq 2\lambda_n \tag{10.9}$$

$$||\widehat{\theta} - \theta^*||_2 \leq 4\lambda_n \Psi(\bar{\mathcal{M}}) \tag{10.10}$$

$$\mathcal{R}(\widehat{\theta} - \theta^*) \leq 8\lambda_n \Psi(\bar{\mathcal{M}})^2 \tag{10.11}$$

For the proposed JEEK model, $\mathcal{R}(\Omega^{tot}) = ||W_I^{tot} \circ \Omega_I^{tot}||_1 + ||W_S^{tot} \circ \Omega_S^{tot}||_1$. Based on the results in(Negahban et al., 2009), $\Psi(\bar{\mathcal{M}}) = \sqrt{k_i + k_s}$, where k_i and k_s are the total number of nonzero entries in Ω_I^{tot} and Ω_S^{tot}. Using $\mathcal{R}(\Omega^{tot}) = ||W_I^{tot} \circ \Omega_I^{tot}||_1 + ||W_S^{tot} \circ \Omega_S^{tot}||_1$ in Theorem (10.8), we have the following theorem (the same as Theorem (4.1)),

Theorem 10.9. *Suppose that* $\mathcal{R}(\Omega^{tot}) = ||W_I^{tot} \circ \Omega_I^{tot}||_1 + ||W_S^{tot} \circ \Omega_S^{tot}||_1$ *and the true parameter* Ω^{tot*} *satisfy the conditions* ***(C1)(C2)*** *and* $\lambda_n \geq \mathcal{R}^*(\widehat{\Omega}^{tot} - \Omega^{tot*})$*, then the optimal point* $\widehat{\Omega}^{tot}$ *of Eq. (3.7) has the following error bounds:*

$$\begin{aligned}
&\max(||W_I^{tot} \circ (\widehat{\Omega}^{tot} - \Omega^{tot*})||_\infty, ||W_S^{tot} \circ (\widehat{\Omega}^{tot} - \Omega^{tot*}||_\infty) \\
&\qquad\qquad \leq 2\lambda_n \\
&||\widehat{\Omega}^{tot} - \Omega^{tot*}||_F \leq 4\sqrt{k_i + k_s}\lambda_n \\
&||W_I^{tot} \circ (\widehat{\Omega}_I^{tot} - \Omega_I^{tot*})||_1 + ||W_S^{tot} \circ (\widehat{\Omega}_S^{tot} - \Omega_S^{tot*})||_1 \\
&\qquad\qquad \leq 8(k_i + k_s)\lambda_n
\end{aligned} \tag{10.12}$$

10.2.2. Proof of Theorem (10.8)

Proof. Let $\delta := \widehat{\theta} - \theta^*$ be the error vector that we are interested in.

$$\begin{aligned}
&\mathcal{R}^*(\widehat{\theta} - \theta^*) = \mathcal{R}^*(\widehat{\theta} - \widehat{\theta}_n + \widehat{\theta}_n - \theta^*) \\
&\leq \mathcal{R}^*(\widehat{\theta}_n - \widehat{\theta}) + \mathcal{R}^*(\widehat{\theta}_n - \theta^*) \leq 2\lambda_n
\end{aligned} \tag{10.13}$$

By the fact that $\theta^*_{\mathcal{M}^{\perp}} = 0$, and the decomposability of $\mathcal{R}$ with respect to $(\mathcal{M}, \bar{\mathcal{M}}^{\perp})$

$$\begin{aligned}
&\mathcal{R}(\theta^*) \\
&= \mathcal{R}(\theta^*) + \mathcal{R}[\Pi_{\bar{\mathcal{M}}^{\perp}}(\delta)] - \mathcal{R}[\Pi_{\bar{\mathcal{M}}^{\perp}}(\delta)] \\
&= \mathcal{R}[\theta^* + \Pi_{\bar{\mathcal{M}}^{\perp}}(\delta)] - \mathcal{R}[\Pi_{\bar{\mathcal{M}}^{\perp}}(\delta)] \\
&\leq \mathcal{R}[\theta^* + \Pi_{\bar{\mathcal{M}}^{\perp}}(\delta) + \Pi_{\bar{\mathcal{M}}}(\delta)] + \mathcal{R}[\Pi_{\bar{\mathcal{M}}}(\delta)] \\
&- \mathcal{R}[\Pi_{\bar{\mathcal{M}}^{\perp}}(\delta)] \\
&= \mathcal{R}[\theta^* + \delta] + \mathcal{R}[\Pi_{\bar{\mathcal{M}}}(\delta)] - \mathcal{R}[\Pi_{\bar{\mathcal{M}}^{\perp}}(\delta)]
\end{aligned} \tag{10.14}$$

Here, the inequality holds by the triangle inequality of norm. Since Eq. (10.7) minimizes $\mathcal{R}(\widehat{\theta})$, we have $\mathcal{R}(\theta^* + \Delta) = \mathcal{R}(\widehat{\theta}) \leq \mathcal{R}(\theta^*)$. Combining this inequality with Eq. (10.14), we have:

$$\mathcal{R}[\Pi_{\bar{\mathcal{M}}^{\perp}}(\delta)] \leq \mathcal{R}[\Pi_{\bar{\mathcal{M}}}(\delta)] \tag{10.15}$$

Moreover, by Hölder's inequality and the decomposability of $\mathcal{R}(\cdot)$, we have:

$$\begin{aligned}
&||\Delta||_2^2 = \langle \delta, \delta \rangle \leq \mathcal{R}^*(\delta)\mathcal{R}(\delta) \leq 2\lambda_n \mathcal{R}(\delta) \\
&= 2\lambda_n[\mathcal{R}(\Pi_{\bar{\mathcal{M}}}(\delta)) + \mathcal{R}(\Pi_{\bar{\mathcal{M}}^{\perp}}(\delta))] \leq 4\lambda_n \mathcal{R}(\Pi_{\bar{\mathcal{M}}}(\delta)) \\
&\leq 4\lambda_n \Psi(\bar{\mathcal{M}})||\Pi_{\bar{\mathcal{M}}}(\delta)||_2
\end{aligned} \tag{10.16}$$

where $\Psi(\bar{\mathcal{M}})$ is a simple notation for $\Psi(\bar{\mathcal{M}}, ||\cdot||_2)$.

Since the projection operator is defined in terms of $||\cdot||_2$ norm, it is non-expansive: $||\Pi_{\bar{\mathcal{M}}}(\Delta)||_2 \leq ||\Delta||_2$. Therefore, by Eq. (10.16), we have:

$$||\Pi_{\bar{\mathcal{M}}}(\delta)||_2 \leq 4\lambda_n \Psi(\bar{\mathcal{M}}), \tag{10.17}$$

and plugging it back to Eq. (10.16) yields the error bound Eq. (10.10).

Finally, Eq. (10.11) is straightforward from Eq. (10.15) and Eq. (10.17).

$$\begin{aligned}&\mathcal{R}(\delta) \le 2\mathcal{R}(\Pi_{\bar{\mathcal{M}}}(\delta))\\ &\le 2\Psi(\bar{\mathcal{M}})||\Pi_{\bar{\mathcal{M}}}(\delta)||_2 \le 8\lambda_n\Psi(\bar{\mathcal{M}})^2.\end{aligned} \tag{10.18}$$

□

10.2.3. Conditions of Proving Error Bounds of JEEK

JEEK achieves similar convergence rates as the SIMULE(Wang et al., 2017b) (W-SIMULE with no additional knowledge) and FASJEM estimator (Wang et al., 2017a). The other multiple sGGMs estimation methods have not provided such convergence rate analysis.

To derive the statistical error bound of JEEK, we need to assume that $inv(T_v(\widehat{\Sigma}^{tot}))$ are well-defined. This is ensured by assuming that the true $\Omega^{(i)^*}$ satisfy the following conditions (Yang et al., 2014b):

(C-MinInf$-\Sigma$): The true $\Omega^{(i)^*}$ Eq. (3.7) have bounded induced operator norm, i.e., $|||\Omega^{(i)^*}|||_\infty := \sup\limits_{w\neq 0\in\mathbb{R}^p} \frac{||\Sigma^{(i)^*}w||_\infty}{||w||_\infty} \le \kappa_1$.

(C-Sparse-Σ): The true covariance matrices $\Sigma^{(i)^*}$ are "approximately sparse" (following (Bickel & Levina, 2008)). For some constant $0 \le q < 1$ and $c_0(p)$, $\max\limits_i \sum\limits_{j=1}^{p} |[\Sigma^{(i)^*}]_{ij}|^q \le c_0(p)$. [6]

We additionally require $\inf\limits_{w\neq 0\in\mathbb{R}^p} \frac{||\Omega^{(i)^*}w||_\infty}{||w||_\infty} \ge \kappa_2$.

10.2.4. Proof of Corollary (4.2)

Proof. In the following proof, we re-denote the following two notations: $\Sigma_{tot} := \begin{pmatrix} \Sigma^{(1)} & 0 & \cdots & 0 \\ 0 & \Sigma^{(2)} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \Sigma^{(K)} \end{pmatrix}$

and

$$\Omega_{tot} := \begin{pmatrix} \Omega^{(1)} & 0 & \cdots & 0 \\ 0 & \Omega^{(2)} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \Omega^{(K)} \end{pmatrix}$$

The condition (C-SparseΣ) and condition (C-MinInfΣ) also hold for Ω^*_{tot} and Σ^*_{tot}. In order to utilize Theorem (10.9) for this specific case, we only need to show that $||\Omega^*_{tot} - [T_\nu(\widehat{\Sigma}_{tot})]^{-1}||_\infty \le \lambda_n$ for the setting of λ_n in the statement:

$$\begin{aligned}&||\Omega^*_{tot} - [T_\nu(\widehat{\Sigma}_{tot})]^{-1}||_\infty\\ &= ||[T_\nu(\widehat{\Sigma}_{tot})]^{-1}(T_\nu(\widehat{\Sigma}_{tot})\Omega^*_{tot} - I)||_\infty\\ &\le |||[T_\nu(\widehat{\Sigma}_{tot})w]|||_\infty||T_\nu(\widehat{\Sigma}_{tot})\Omega^*_{tot} - I||_\infty\\ &= |||[T_\nu(\widehat{\Sigma}_{tot})]^{-1}|||_\infty||\Omega^*_{tot}(T_\nu(\widehat{\Sigma}_{tot}) - \Sigma^*_{tot})||_\infty\\ &\le |||[T_\nu(\widehat{\Sigma}_{tot})]^{-1}|||_\infty|||\Omega^*_{tot}|||_\infty||T_\nu(\widehat{\Sigma}_{tot}) - \Sigma^*_{tot}||_\infty.\end{aligned} \tag{10.19}$$

We first compute the upper bound of $|||[T_\nu(\widehat{\Sigma}_{tot})]^{-1}|||_\infty$. By the selection ν in the statement, Lemma (11.2) and Lemma (11.3) hold with probability at least $1-4/p'^{\tau-2}$. Armed with Eq. (11.9), we use the triangle inequality of norm and the condition (C-SparseΣ): for any w,

$$\begin{aligned}&||T_\nu(\widehat{\Sigma}_{tot})w||_\infty = ||T_\nu(\widehat{\Sigma}_{tot})w - \Sigma w + \Sigma w||_\infty\\ &\ge ||\Sigma w||_\infty - ||(T_\nu(\widehat{\Sigma}_{tot}) - \Sigma)w||_\infty\\ &\ge \kappa_2||w||_\infty - ||(T_\nu(\widehat{\Sigma}_{tot}) - \Sigma)w||_\infty\\ &\ge (\kappa_2 - ||(T_\nu(\widehat{\Sigma}_{tot}) - \Sigma)w||_\infty)||w||_\infty\end{aligned} \tag{10.20}$$

Where the second inequality uses the condition (C-SparseΣ). Now, by Lemma (11.2) with the selection of ν, we have

$$|||T_\nu(\widehat{\Sigma}_{tot}) - \Sigma|||_\infty \le c_1(\frac{\log(Kp')}{n_{tot}})^{(1-q)/2}c_0(p) \tag{10.21}$$

where c_1 is a constant related only on τ and $\max_i \Sigma_{ii}$. Specifically, it is defined as $6.5(16(\max_i \Sigma_{ii})\sqrt{10\tau})^{1-q}$. Hence, as long as $n_{tot} > (\frac{2c_1c_0(p)}{\kappa_2})^{\frac{2}{1-q}}\log p'$ as stated, so that $|||T_\nu(\widehat{\Sigma}_{tot}) - \Sigma|||_\infty \le \frac{\kappa_2}{2}$, we can conclude that $||T_\nu(\widehat{\Sigma}_{tot})w||_\infty \ge \frac{\kappa_2}{2}||w||_\infty$, which implies $|||[T_\nu(\widehat{\Sigma}_{tot})]^{-1}|||_\infty \le \frac{2}{\kappa_2}$.

The remaining term in Eq. (10.19) is $||T_\nu(\widehat{\Sigma}_{tot}) - \Sigma^*_{tot}||_\infty$; $||T_\nu(\widehat{\Sigma}_{tot}) - \Sigma^*_{tot}||_\infty \le ||T_\nu(\widehat{\Sigma}_{tot}) - \widehat{\Sigma}_{tot}||_\infty + ||\widehat{\Sigma}_{tot} - \Sigma^*_{tot}||_\infty$. By construction of $T_\nu(\cdot)$ in (C-Thresh) and by Lemma (11.3), we can confirm that $||T_\nu(\widehat{\Sigma}_{tot}) - \widehat{\Sigma}_{tot}||_\infty$ as well as $||\widehat{\Sigma}_{tot} - \Sigma^*_{tot}||_\infty$ can be upper-bounded by ν.

Therefore,

$$\begin{aligned}&\max(||W_I^{tot} \circ (\Omega^{tot^*} - inv(T_v(\widehat{\Sigma}^{tot})))||_\infty,\\ &||W_S^{tot} \circ (\Omega^{tot^*} - inv(T_v(\widehat{\Sigma}^{tot})))||_\infty)\\ &\le O(\max\max_{j,k}({W_I^{tot}}_{j,k}, {W_S^{tot}}_{j,k})\sqrt{\frac{\log(Kp)}{n_{tot}}})\end{aligned} \tag{10.22}$$

[6] This indicates for some positive constant d, $[\Sigma^{(i)^*}]_{jj} \le d$ for all diagonal entries. Moreover, if $q = 0$, then this condition reduces to $\Sigma^{(i)^*}$.

By combining all together, we can confirm that the selection of λ_n satisfies the requirement of Theorem (10.9), which completes the proof. □

11. Appendix: More Background of Proxy Backward mapping and Theorems of T_v Being Invertible

The first row of Figure 2 summarizes the EE-sGGMs. Two important concepts:

(1) Backward Mapping: The Gaussian distribution is naturally an exponential-family distribution. Based on (Wainwright & Jordan, 2008), learning an exponential family distribution from data means to estimate its canonical parameter. For an exponential family distribution, computing the canonical parameter through vanilla graphical model MLE can be expressed as a backward mapping (the first step in Figure 2). For a Gaussian, the backward mapping is easily computable as the inverse of the sample covariance matrix. More details in Section (11.1).

(2) Proxy Backward Mapping: When being high-dimensional, we can not compute the backward mapping of Gaussian through the inverse of the sample covariance matrix. Now the key is to find a closed-form and statistically guaranteed estimator as the proxy backward mapping under high-dimensional cases. By the conclusion given by the EE-sGGM, we choose $\{(\{[T_v(\widehat{\Sigma}^{(i)})]^{-1})\}$ as the proxy backward mapping for $\{\Omega^{(i)}\}$.

$$[T_v(A)]_{ij} := \rho_v(A_{ij}) \tag{11.1}$$

where $\rho_v(\cdot)$ is chosen to be a soft-thresholding function.

11.1. More About Background: backward mapping for an exponential-family distribution:

The solution of vanilla graphical model MLE can be expressed as a backward mapping(Wainwright & Jordan, 2008) for an exponential family distribution. It estimates the model parameters (canonical parameter θ) from certain (sample) moments. We provide detailed explanations about backward mapping of exponential families, backward mapping for Gaussian special case and backward mapping for differential network of GGM in this section.

Backward mapping: Essentially the vanilla graphical model MLE can be expressed as a backward mapping that computes the model parameters corresponding to some given moments in an exponential family distribution. For instance, in the case of learning GGM with vanilla MLE, the backward mapping is $\widehat{\Sigma}^{-1}$ that estimates Ω from the sample covariance (moment) $\widehat{\Sigma}$.

Suppose a random variable $X \in \mathbb{R}^p$ follows the exponential family distribution:

$$\mathbb{P}(X;\theta) = h(X)\text{exp}\{< \theta, \phi(\theta) > -A(\theta)\} \tag{11.2}$$

Where $\theta \in \Theta \subset \mathbb{R}^d$ is the canonical parameter to be estimated and Θ denotes the parameter space. $\phi(X)$ denotes the sufficient statistics as a feature mapping function $\phi : \mathbb{R}^p \to \mathbb{R}^d$, and $A(\theta)$ is the log-partition function. We then define mean parameters v as the expectation of $\phi(X)$: $v(\theta) := \mathbb{E}[\phi(X)]$, which can be the first and second moments of the sufficient statistics $\phi(X)$ under the exponential family distribution. The set of all possible moments by the moment polytope:

$$\mathcal{M} = \{v | \exists p \text{ is a distribution s.t. } \mathbb{E}_p[\phi(X)] = v\} \tag{11.3}$$

Mostly, the graphical model inference involves the task of computing moments $v(\theta) \in \mathcal{M}$ given the canonical parameters $\theta \in \textcircled{H}$. We denote this computing as **forward mapping** :

$$\mathcal{A} : \textcircled{H} \to \mathcal{M} \tag{11.4}$$

The learning/estimation of graphical models involves the task of the reverse computing of the forward mapping, the so-called **backward mapping** (Wainwright & Jordan, 2008). We denote the interior of $\mathcal{M}$ as $\mathcal{M}^0$. **backward mapping** is defined as:

$$\mathcal{A}^* : \mathcal{M}^0 \to \textcircled{H} \tag{11.5}$$

which does not need to be unique. For the exponential family distribution,

$$\mathcal{A}^* : v(\theta) \to \theta = \nabla A^*(v(\theta)). \tag{11.6}$$

Where $A^*(v(\theta)) = \sup_{\theta \in \textcircled{H}} < \theta, v(\theta) > -A(\theta)$.

Backward Mapping: Gaussian Case If a random variable $X \in \mathbb{R}^p$ follows the Gaussian Distribution $N(\mu, \Sigma)$, then $\theta = (\Sigma^{-1}\mu, -\frac{1}{2}\Sigma^{-1})$. The sufficient statistics $\phi(X) = (X, XX^T)$, $h(x) = (2\pi)^{-\frac{k}{2}}$, and the log-partition function

$$A(\theta) = \frac{1}{2}\mu^T\Sigma^{-1}\mu + \frac{1}{2}\log(|\Sigma|) \tag{11.7}$$

When performing the inference of Gaussian Graphical Models, it is easy to estimate the mean vector $v(\theta)$, since it equals to $\mathbb{E}[X, XX^T]$.

When learning the GGM, we estimate its canonical parameter θ through vanilla MLE. Because Σ^{-1} is one entry of θ

we can use the backward mapping to estimate Σ^{-1}.

$$\begin{aligned}\theta &= (\Sigma^{-1}\mu, -\frac{1}{2}\Sigma^{-1}) = \mathcal{A}^*(v) = \nabla A^*(v) \\ &= ((\mathbb{E}_\theta[XX^T] - \mathbb{E}_\theta[X]\mathbb{E}_\theta[X]^T)^{-1}\mathbb{E}_\theta[X], \qquad (11.8) \\ &\quad -\frac{1}{2}(\mathbb{E}_\theta[XX^T] - \mathbb{E}_\theta[X]\mathbb{E}_\theta[X]^T)^{-1}).\end{aligned}$$

By plugging in Eq. (11.7) into Eq. (11.6), we get the backward mapping of Ω as $(\mathbb{E}_\theta[XX^T] - \mathbb{E}_\theta[X]\mathbb{E}_\theta[X]^T)^{-1}) = \widehat{\Sigma}^{-1}$, easily computable from the sample covariance matrix.

11.2. Theorems of T_v Being Invertible

Based on (Yang et al., 2014b) for any matrix A, the element wise operator T_v is defined as:

$$[T_v(A)]_{ij} = \begin{cases} A_{ii} + v & if\ i = j \\ sign(A_{ij})(|A_{ij}| - v) & otherwise, i \neq j \end{cases}$$

Suppose we apply this operator T_v to the sample covariance matrix $\frac{X^TX}{n}$ to obtain $T_v(\frac{X^TX}{n})$. Then, $T_v(\frac{X^TX}{n})$ under high dimensional settings will be invertible with high probability, under the following conditions:
Condition-1 (Σ-Gaussian ensemble) Each row of the design matrix $X \in \mathbb{R}^{n\times p}$ is i.i.id sampled from $N(0, \Sigma)$.
Condition-2 The covariance Σ of the Σ-Gaussian ensemble is strictly diagonally dominant: for all row i, $\delta_i := \Sigma_{ii} - \Sigma_{j\neq i} \geq \delta_{min} > 0$ where δ_{min} is a large enough constant so that $||\Sigma||\infty \leq \frac{1}{\delta_{min}}$.

This assumption guarantees that the matrix $T_v(\frac{X^TX}{n})$ is invertible, and its induced ℓ_∞ norm is well bounded. Then the following theorem holds:

Theorem 11.1. *Suppose Condition-1 and Condition-2 hold. Then for any* $v \geq 8(max_i\Sigma_{ii})\sqrt{(\frac{10\tau\log p'}{n})}$*, the matrix* $T_v(\frac{X^TX}{n})$ *is invertible with probability at least* $1-4/p'^{\tau-2}$ *for* $p' := max\{n, p\}$ *and any constant* $\tau > 2$.

Then we provide the error bound of T_v in the first lemma of Section (11.3) and use it in deriving the error bound of JEEK.

11.3. Useful lemma(s) of Error Bounds of (Proxy) Backward Mapping

Lemma 11.2. *(Theorem 1 of (Rothman et al., 2009)). Let* δ *be* $\max_{ij}||[\frac{X^TX}{n}]_{ij} - \Sigma_{ij}|$*. Suppose that* $\nu > 2\delta$*. Then, under the conditions (C-SparseΣ), and as* $\rho_v(\cdot)$ *is a soft-threshold function, we can deterministically guarantee that the spectral norm of error is bounded as follows:*

$$|||T_v(\widehat{\Sigma}) - \Sigma|||_\infty \leq 5\nu^{1-q}c_0(p) + 3\nu^{-q}c_0(p)\delta \qquad (11.9)$$

Lemma 11.3. *(Lemma 1 of (Ravikumar et al., 2011)). Let* $\mathcal{A}$ *be the event that*

$$||\frac{X^TX}{n} - \Sigma||_\infty \leq 8(\max_i \Sigma_{ii})\sqrt{\frac{10\tau\log p'}{n}} \qquad (11.10)$$

where $p' := \max(n, p)$ *and* τ *is any constant greater than 2. Suppose that the design matrix X is i.i.d. sampled from* Σ*-Gaussian ensemble with* $n \geq 40\max_i \Sigma_{ii}$*. Then, the probability of event* $\mathcal{A}$ *occurring is at least* $1 - 4/p'^{\tau-2}$.

12. Design W_S and $W_I^{(i)}$: connections with related work and real-world applications

In this section, we showcase with specific examples that our proposed model JEEK can easily incorporate edge-level (like distance) as well as node-based (like hubs or groups) knowledge for the joint estimation of multiple graphs. To this end, we introduce four different choices of W_S^{tot} and W_I^{tot} in our formulation Eq. (3.7). By simply designing different choices of W_S^{tot} and W_I^{tot}, we can express different kinds of additional knowledge explicitly without changing the optimization algorithm.

Specifically, we design W_S and $W_I^{(i)}$ for cases like:

- (1) the additional knowledge is available in the form of a $p * p$ matrix W. For instance distance matrix among brain regions in neuroscience study belongs to this type;
- (2) the existing knowledge is not in the form of matrix about nodes. We need to design W for such cases, for example the information of known hub nodes or the information of how nodes fall into groups (e.g., genes belonging to the same pathway or locations).

For the second kind, we showcase three different designs of weight matrices for representing (a) known co-Hub nodes, (b) perturbed hub nodes, and (c) node grouping information.

The design of knowledge matrices is loosely related to the different structural assumptions used by he JGL studies as ((Mohan et al., 2013), (Danaher et al., 2013)). For example, JGL can use specially designed norms like the one proposed in (Mohan et al., 2013) to push multiple graphs to have a similar set of nodes as hubs. However JGL can not model additional knowledge like a specific set of nodes are hub nodes (like we know node j is a hub node). Differently, JEEK can design $\{W_I^{(i)}, W_S\}$ for incorporating such knowledge. Essentially JEEK is complementary to JGL because they capture different type of prior information.

12.1. Case study I: Knowledge as matrix form like a distance matrix or some known edges

The first example we consider is exploiting a spatial prior to jointly estimate brain connectivity for multiple subject groups. Over time, neuroscientists have gathered considerable knowledge regarding the spatial and anatomical information underlying brain connectivity (*i.e.* short edges and certain anatomical regions are more likely to be connected (Watts & Strogatz, 1998)). Previous studies enforce these priors via a matrix of weights, W, corresponding to edges. To use our proposed model JEEK for such tasks, we can similarly choose $W = W_I^{(i)} = W_S$ in Eq. (3.7)).

12.2. Case study II: Knowledge of co-hub nodes

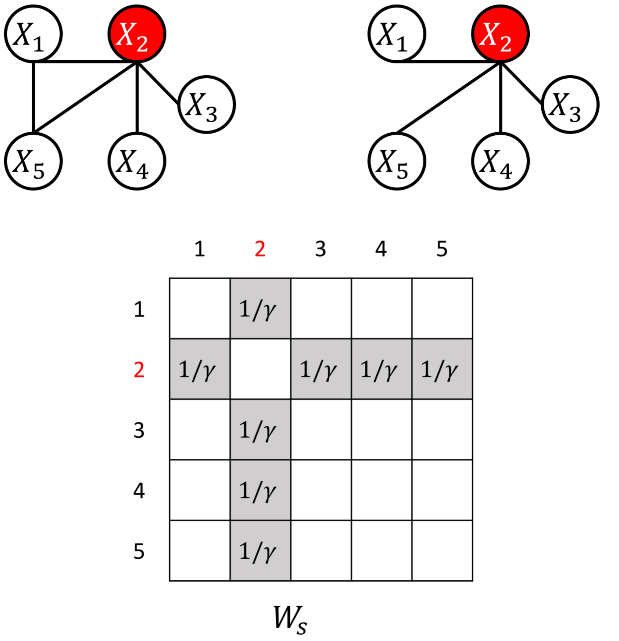
The structure assumption we consider is graphs with co-hub nodes. Namely, there exists a set of nodes $NId = \{j | j \in \{1, 2, \dots, p\}\}$ such that $\Omega_{j,k}^{(i)} \neq 0, \forall i \in \{1, 2, \dots, K\}$ and $k \in \{1, \dots, p\}$. The above sub-figure of Figure 8 is an example of the co-hub nodes.

A so-called JGL-hub (Mohan et al., 2013) estimator chooses $\mathcal{R}'(\cdot) = \sum_{i<i'} P_q(\Omega^{(i)} - \Omega^{(i')})$ in Eq. (5.2) to account for the co-hub structure assumption. Here $P_q(\Theta_1, \Theta_2, \dots, \Theta_k) = 1/2||\Theta_1, \dots, \Theta_k||_{\ell_1,\ell_q}$. Θ_i is a symmetric matrix and $||\cdot||_{\ell_1,\ell_q}$ is the notation of ℓ_1, ℓ_q-norm. JGL-hub formulation needs a complicated ADMM solution with computationally expensive SVD steps.

We design W_S and $W_I^{(i)}$ for the co-hub type knowledge in JEEK via: (1) We initialize $\{W_I^{(i)}, W_S\}$ with $\mathbf{1}_{p\times p}$; (2) $W_{S j,k} = \frac{1}{\gamma}, \forall j \in NId$ and $k \in 1, \dots, p$ where γ is a hyperparameter. Therefore, the smaller weights for the edge connecting to the node j of all the graphs enforce the co-hub structure.; (3). After this process, each entry of $\{W_I^{(i)}, W_S\}$ equals to either $\frac{1}{\gamma}$ or 1. The below sub-figure of Figure 8 is an example of the designed W_S.

12.3. Case study III: Knowledge of the perturbed hub nodes

Another structure assumption we study is graphs with perturbed nodes. Namely, there exists a set of nodes $NId = \{j | j \in \{1, 2, \dots, p\}\}$ so that there exists i, i' $\Omega_{j,k}^{(i)} \neq 0$, and $\Omega_{j,k}^{(i')} = 0, \forall k \in \{1, \dots, p\}$. The above sub-figure of Figure 9 is an example of the perturbed nodes. A so-called JGL-perturb (Mohan et al., 2013) estimator chose $\mathcal{R}'(\cdot) = \sum_{i<i'} P_q((\Omega^{(1)} - \text{diag}(\Omega^{(1)})), \dots, (\Omega^{(K)} - \text{diag}(\Omega^{(K)})))$ in Eq. (5.2). Here $P_q(\cdot)$ has the same definition as mentioned previously. This JGL-perturb formulation also needs a complicated ADMM solution with computationally expensive SVD steps.



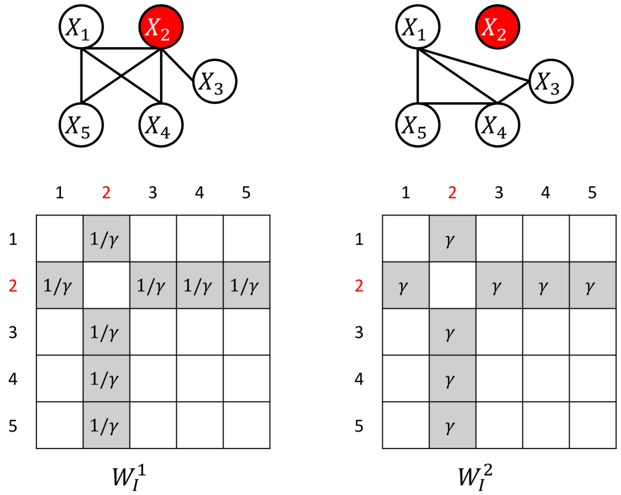
	1	2	3	4	5
1		1/γ			
2	1/γ		1/γ	1/γ	1/γ
3		1/γ			
4		1/γ			
5		1/γ			

Figure 5. co-hub. Top: An example of the co-hub node structure. Bottom: The designed W_S for the co-hub structure case (white off-diagonal entries are 1).

To design W_S and $W_I^{(i)}$ for this type of knowledge in JEEK, we use a similar strategy as the above strategy: (1) We initialize $\{W_I^{(i)}, W_S\}$ with $\mathbf{1}_{p\times p}$; we let ${W_I^{(i)}}_{j,k} = \frac{1}{\gamma}, {W_I^{(i')}}_{j,k} = \gamma, \forall j \in NId$ and $k \in 1, \dots, p$. Therefore, the different weights for the edge connecting to the node j in different $W_I^{(i)}$ enforce the node-perturbed structure. ; (3). After this process, each entry of $\{W_I^{(i)}, W_S\}$ equals to either $\frac{1}{\gamma}$,γ or 1. The below sub-figure of Figure 9 is an example of the designed $\{W_I^{(i)}\}$.

12.4. Case study IV: Knowledge of group information about nodes

To design W_S and $W_I^{(i)}$ for the group information about a set of nodes, we use a simple three-step strategy: (1) We initialize $\{W_I^{(i)}, W_S\}$ with $\mathbf{1}_{p\times p}$; (2) We let $W_{S j,k} = \frac{1}{\gamma}, \forall (j,k) \in Id$ where γ is a hyperparameter. Therefore, the smaller weights for the edge (j,k) in all the graphs favors the edges among nodes in the same group. ; (3). After this process, each entry of $\{W_I^{(i)}, W_S\}$ equals to either $\frac{1}{\gamma}$ or 1. The below sub-figure of Figure 7 is an example of the designed W_S (extra knowledge is that X_2, X_3, X_4 belong to the same group).



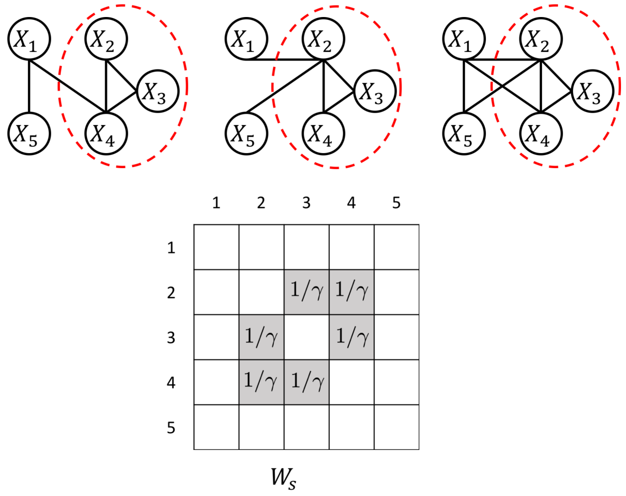
$$\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & & 1/\gamma & & & \\ 2 & 1/\gamma & & 1/\gamma & 1/\gamma & 1/\gamma \\ 3 & & 1/\gamma & & & \\ 4 & & 1/\gamma & & & \\ 5 & & 1/\gamma & & & \end{array}$$

W_I^1

$$\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & & \gamma & & & \\ 2 & \gamma & & \gamma & \gamma & \gamma \\ 3 & & \gamma & & & \\ 4 & & \gamma & & & \\ 5 & & \gamma & & & \end{array}$$

W_I^2

Figure 6. Perturb hub nodes. Top: An example of the perturbed node structure. Bottom: The designed W_I for the perturbed case. (white off-diagonal entries are 1.)



$$\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & & & & & \\ 2 & & & 1/\gamma & 1/\gamma & \\ 3 & & 1/\gamma & & 1/\gamma & \\ 4 & & 1/\gamma & 1/\gamma & & \\ 5 & & & & & \end{array}$$

W_s

Figure 7. Co-group example case. Top: An example of the co-group node structure. Bottom: The designed W_S for the case. (white off-diagonal entries are 1.)

13. More about Experimental Setup

13.1. Experimental Setup

On four types of datasets, we focus on empirically evaluating JEEK with regard to three aspects: (i) effectiveness, computational speed and scalability in brain connectivity simulation data; (ii) flexibility in incorporating different types of knowledge of known hub nodes in graphs; (iii) effectiveness and computational speed for brain connectivity estimation from real-world fMRI.

13.2. Evaluation Metrics

- AUC-score: The edge-level false positive rate (FPR) and true positive rate (TPR) are used to measure the difference between the true graphs and the predicted graphs. We obtain FPR vs. TPR curve for each method by tuning over a range of its regularization parameter. We use the area under the FPR -TPR curve (AUC-Score) to compare the predicted versus true graph. Here, FPR $= \dfrac{\text{FP}}{\text{FP} + \text{TN}}$ and TPR $= \dfrac{\text{TP}}{\text{FN} + \text{TP}}$. TP (true positive) and TN (true negative) means the number of true edges and non-edges correctly estimated by the predicted network respectively. FP (false positive) and FN (false negative) are the number of incorrectly predicted nonzero entries and zero entries respectively.
- F1-score: We first use the edge-level F1-score to compare the predicted versus true graph. Here, F1 $= \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$, where Precision $= \frac{\text{TP}}{\text{TP}+\text{FP}}$ and Recall $= \frac{\text{TP}}{\text{TP}+\text{FN}}$. The better method achieves a higher F1-score.
- Time Cost: We use the execution time (measured in seconds or log(seconds)) for a method as a measure of its scalability. To ensure a fair comparison, we try 30 different λ_n (or λ_2) and measure the total time of execution for each method. The better method uses less time[7]

Evaluations: For the first experiment on brain simulation data, we evaluate JEEK and the baseline methods on F1-score and running time cost. For the second experiment, we use AUC-score and running time cost.[8] For the third experiment, our evaluation metrics include classification accuracy, likelihood and running time cost.

- The first set of experiments evaluates the speed and scalability of our model JEEK on simulation data imitating brain connectivity. We compare both the estimation performance and computational time of JEEK with the baselines in multiple simulated datasets.
- In the second experiment, we show JEEK's ability to incorporate knowledge of known hubs in multiple graphs. We also compare the estimation performance and scalability of JEEK with the baselines in multiple simulated datasets.
- Thirdly, we evaluate the ability to import additional knowledge for enhancing graph estimation in a real world dataset. The dataset used in this experiment is a human brain fMRI dataset with two groups of subjects: autism and control. Our choice of this dataset is motivated by recent literature in neuroscience that has suggested many known weights between different regions in human brain as the additional knowledge.

[7] The machine that we use for experiments is an AMD 64-core CPU with a 256GB memory.

[8] We cannot use AUC-score for the first set of experiments as the baseline NAK only gives us the best adjacency matrix after tuning over their hyperparameters. It does not provide an option for tuning the λ_n.

13.3. Hyper-parameters:

We need to tune four hyper-parameters v, λ_n, λ_2 and γ:

- v is used for soft-thresholding in JEEK. We choose v from the set $\{0.001i|i = 1, 2, \dots, 1000\}$ and pick a value that makes $T_v(\widehat{\Sigma}^{(i)})$ invertible.
- λ_n is the main hyper-parameter that controls the sparsity of the estimated network. Based on our convergence rate analysis in Section 4, $\lambda_n \geq C\sqrt{\frac{\log Kp}{n_{tot}}}$ where $n_{tot} = Kn$ and $n = n_i$. Accordingly, we choose λ_n from a range of $\{0.01 \times \sqrt{\frac{\log Kp}{n_{tot}}} \times i|i \in \{1, 2, 3, \dots, 30\}\}$.
- λ_2 controls the regularization of the second penalty function in JGL-type estimators. We tune λ_2 from the set $\{0.01, 0.05, 0.1\}$ for all experiments and pick the one that gives the best results.
- γ is a hyperparameter used to design the $W_I^{(i)}, W_S$ (5). The value of γ intuitively indicates the confidence of the additional knowledge weights. In the second experiment, we choose $\gamma = \{2, 4, 10\}$.

14. More about Experimental Results

14.1. More Experiment: Simulate Samples with Known Hubs as Knowledge

In this set of experiments, we show empirically JEEK's ability to model knowledge of known hub nodes across multiple sGGMs and its advantages in scalability and effectiveness. We generate multiple simulated Gaussian datasets for both the co-hub and perturbed-hub graph structures.

Simulation Protocol to generate simulated datasets: We generate multiple sets of synthetic multivariate-Gaussian datasets. First, we generate random graphs following the Random Graph Model (Rothman et al., 2008). This model assumes $\Omega^{(i)} = \mathbf{B}_I^{(i)} + \mathbf{B}_S + \delta I$, where each off-diagonal entry in $\mathbf{B}^{(i)}$ is generated independently and equals 0.5 with probability $0.1i$ and 0 with probability $1 - 0.1i$. The shared part $\mathbf{B}_S$ is generated independently and equal to 0.5 with probability 0.1 and 0 with probability 0.9. δ is selected large enough to guarantee positive definiteness. We generate cohub and perturbed structure simulations, using the following data generation models:

- **Random Graphs with cohub nodes:** After we generate the random graphs using the aforementioned Random Graph Model, we randomly generate a set of nodes $NId = \{j|j \in \{1, 2, \dots, p\}\}$ as the cohub nodes among all the random graphs. The cardinal number of this set equals to $5\%p$. For each of these nodes j, we randomly select 90% edges $E_j = \{(j, k)|k \in \{1, 2, \dots, p\}\}$ to be included in the graph. Then we set $\Omega_{j,k}^{(i)} = \Omega_{k,j}^{(i)} = 0.5, \forall i \in \{1, 2, \dots, K\}$ and $(j, k) \in E_j$.
- **Random Graphs with perturbed nodes:** After we generate the random graphs using the aforementioned Random Graph Model, we randomly generate a set of nodes $NId = \{j|j \in \{1, 2, \dots, p\}\}$ as the perturbed hub nodes for the random graphs. The cardinal number of this set equals to $5\%p$. For all graphs $\{\Omega^{(i)}|\text{i is odd}\}$, for each of these nodes $j \in NId$, we randomly select 90% edges $E_j = \{(j, k)|k \in \{1, 2, \dots, p\}\}$ to be included in the graph. We set $\Omega_{j,k}^{(i)} = \Omega_{k,j}^{(i)} = 0.5, \forall$ odd $i \in \{1, 2, \dots, K\}$ and $(j, k) \in E_j$. For all graphs $\{\Omega^{(i)}|\text{i is even}\}$ and nodes $j \in NId$, we randomly select 10% edges $E_j^{'} = \{(j, k)|k \in \{1, 2, \dots, p\}\}$ to be included in the graph. We set $\Omega_{j,k}^{(i)} = \Omega_{k,j}^{(i)} = 0.5, \forall$ even $i \in \{1, 2, \dots, K\}$ and $(j, k) \in E_j^{'}$. This creates a perturbed node structure in the multiple graphs.

Experimental baselines: We employ JGL-node for cohub and perturbed hub node structure (JGL-hub and JGL-perturb respectively) and W-SIMULE as the baselines for this set of experiments. The weights in $\{W_I^{tot}, W_S^{tot}\}$ are designed by the strategy mentioned in Section 12.

Experiment Results: We assess the performance of JEEK in terms of effectiveness (AUC score) and scalability (computational time cost) through baseline comparison as follows:

(a) Effectiveness: We plot the AUC-score for a number of multiple simulated datasets generated by varying the number of features p, the number of tasks K and the number of samples n. We calculate AUC by varying λ_n. For the JGL estimator, we additionally vary λ_2 and select the best AUC (section 13.1). In Figure 8 (a) and Figure 8 (b), we plot the AUC-Score for the cohub node structure vs varying p and K, respectively. Figure 9 (a) and Figure 9 (b) plot the same for the perturbed node structure. In Figure 8 (a) and Figure 9 (a), we vary p in the set $\{100, 200, 300, 400, 500\}$ and set $K = 2$ and $n = p/2$. For $p > 300$ and $n = p/2$, W-SIMULE takes more than one month and JGL takes more than one day. Therefore we can not show their results for $p > 300$. For both the cohub and perturbed node structures, JEEK consistently achieves better AUC-score than the baseline methods as p is increased. For Figure 8(b) and Figure 9 (b), we vary K in the set $\{2, 3, 4\}$ and set $p = 200$ and $n = p/2$. JEEK consistently has a higher AUC-score than the baselines JGL and W-SIMULE as K is increased.

(b) Scalability: In Figure 8 (c) and (d), we plot the computational time cost for the cohub node structure vs the number of features p and the number of tasks K, respectively. Figure 9 (c) and (d) plot the same for the perturbed node structure. We interpolate the points of computation time of each estimator into curves. For each simulation

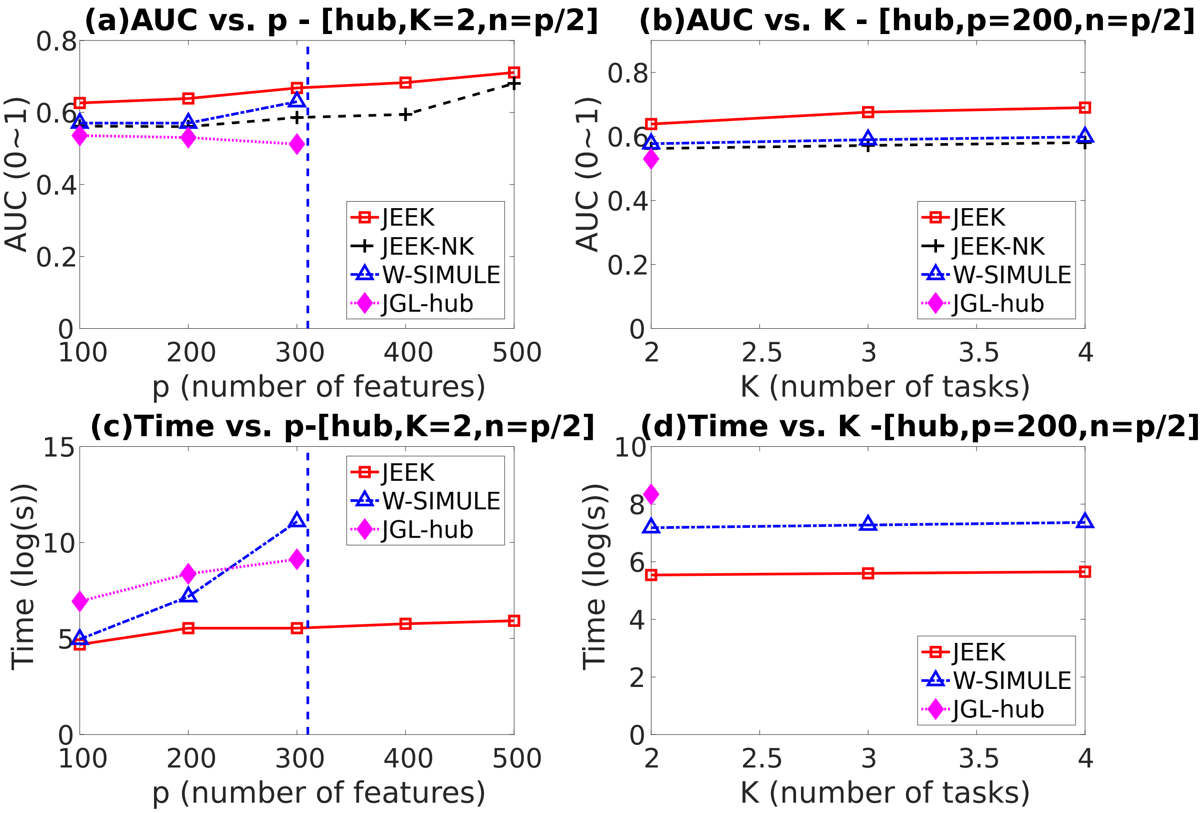


Figure 8. Cohub node structure: (a) AUC-score vs the number of features (p). (b) AUC-score vs the number of tasks (K). (c) Time cost (log(seconds)) vs the number of features (p). (d) Time cost (log(seconds)) vs the number of tasks (K). For $p > 300$ and $n = p/2$ W-SIMULE takes more than one month and JGL takes more than one day (indicated by dotted blue line). JGL package can only run for $K = 2$.

case, the computation time for each estimator is the summation of a method's execution time over all values of λ_n. In Figure 8(c) and Figure 9(c), we vary p in the set $\{100, 200, 300, 400, 500\}$ and set $K = 2$ and $n = p/2$. When $p > 300$ and $n = p/2$, W-SIMULE takes more than one month and JGL takes more than one day. Hence, we have omitted their results for $p > 300$. For both the cohub and perturbed node structures, JEEK is consistently more than 5 times faster as p is increased. In Figure 8(d) and Figure 9 (d), we vary K in the set $\{2, 3, 4\}$ and fix $p = 200$ and $n = p/2$. JEEK is 50 times faster than the baselines for all cases with $p = 200$ and as K is increased. In summary, JEEK is on an average more than 10 times faster than all the baselines.

(c) Stability of Results when varying W matrices: Additionally, to account for JEEK's explicit structure assumption, we also vary the ratio of known hub nodes to the total number of hub nodes. The known hub nodes are used to design the $\{W_I^i, W_S\}$ matrices(details in Section 5). In Figure 10(a) and (b), AUC for JEEK increases as the ratio of the number of known to total hub nodes increases. The initial increase in AUC is particularly significant as it confirms that JEEK is effective in harvesting additional knowledge for multiple sGGMs. The increase in AUC is particularly significant in the perturbed node case (Figure 10(b)). The AUC for the hub case does not have a correspondingly large increase with an increase in ratio because the total number of hub nodes are only 5% of the total nodes. In comparison, an increase in this ratio leads to a more significant increase in AUC because the perturbed node assumption has more information than the cohub node structure. We show in Figure 10(c) and (d) that the computational cost is largely unaffected by this ratio for both the cohub and perturbed node structure.

We also empirically check how the parameter r in the designed knowledge weight matrices influences the performance. In Figure 11(a) and (b), we show that the designed strategy for including additional knowledge as W is not affected by variations of γ. We vary γ in the set of $\{2, 4, 10\}$.

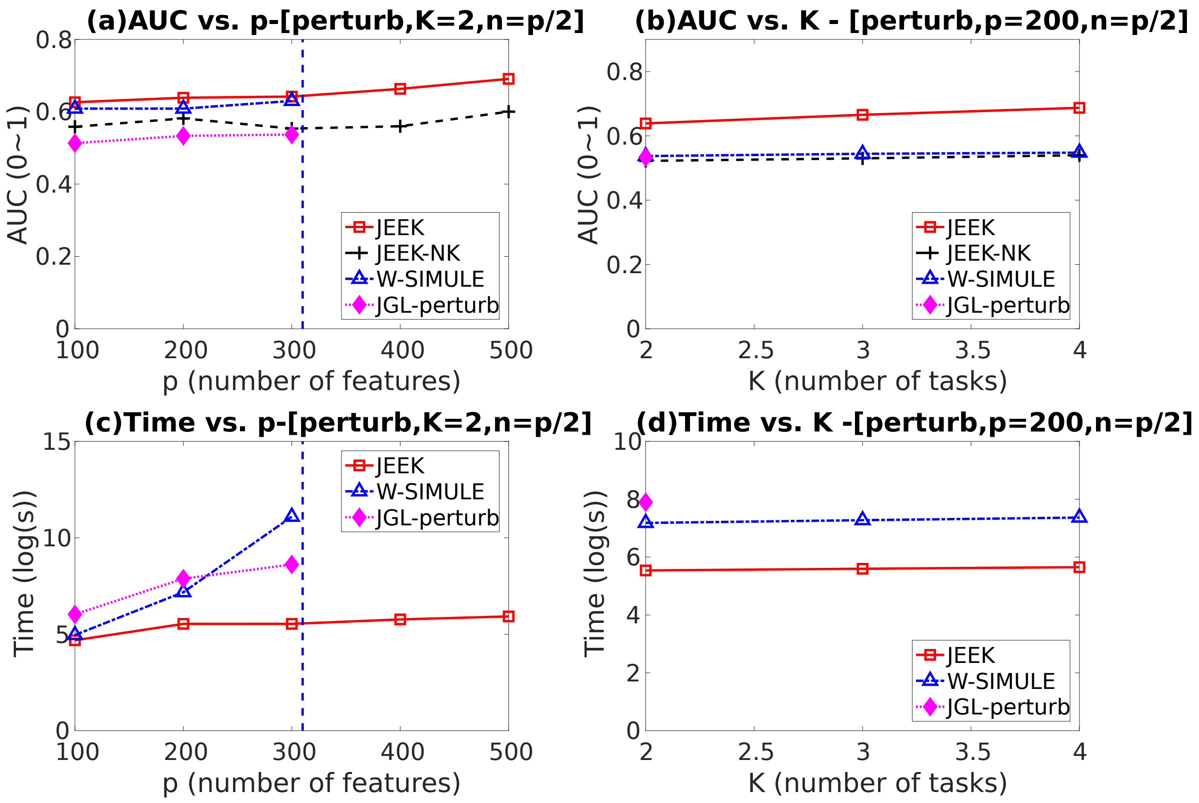


Figure 9. Perturbed node structure: (a) AUC-score vs the number of features (p). (b) AUC-score vs the number of tasks (K). (c) Time cost (log(seconds)) of JEEK and the baseline methods vs the number of features (p). (d) Time cost (log(seconds)) vs the number of tasks (K). for $p > 300$ and $n = p/2$, W-SIMULE takes more than one month and JGL takes more than one day (indicated by dotted blue line). JGL package can only run for $K = 2$.

In summary, the AUC-score(Figure 11(a),(b)) and computational time cost(Figure 11(c),(d)) remains relatively unaffected by the changes in γ for both co-hub and perturbed-hub case.

Figure 12 empirically shows the performance of our methods and baselines when varying the number of samples. We vary n in the set $\{100, 200, 400\}$ and fix $p = 200$ and $K = 2$. In Figure 12 (c) and (d), we plot the time cost vs the number of samples n for the cohub and perturbed node structures respectively. JEEK is much faster than both JGL-node (JGL-hub and JGL-perturb) and W-SIMULE for all cases. Also, the time cost of JEEK does not vary significantly as n increases. In Figure 12 (a) and (b) we also present the AUC-score vs the varying number of samples n for the cohub and perturbed node structures respectively. For both the cohub and perturbed node structure, JEEK achieves a higher AUC-score compared to W-SIMULE and JGL-node (JGL-hub and JGL-perturb) when $p > n$. The only cases in which the W-SIMULE performs better in Figure 12 (a) and (b) is the low dimensional case ($p = 200$, $n = 400$). This is as expected because JEEK is designed for high dimensional data situations.

14.2. More Experiment: Gene Interaction Network from Real-World Genomics Data

Next, we apply JEEK and the baselines on one real-world biomedical data: gene expression profiles describing many human samples across multiple cancer types aggregated by (McCall et al., 2011).

Advancements in genome-wide monitoring have resulted in enormous amounts of data across most of the common cell contexts, like multiple common cancer types (Network et al., 2011). Complex diseases such as cancer are the result of multiple genetic and epigenetic factors. Thus, recent research has shifted towards the identification of multiple genes/proteins that interact directly or indirectly in contributing to certain disease(s). Structure learning of sGGMs on such heterogeneous datasets can uncover statistical dependencies among genes and understand how such depen-

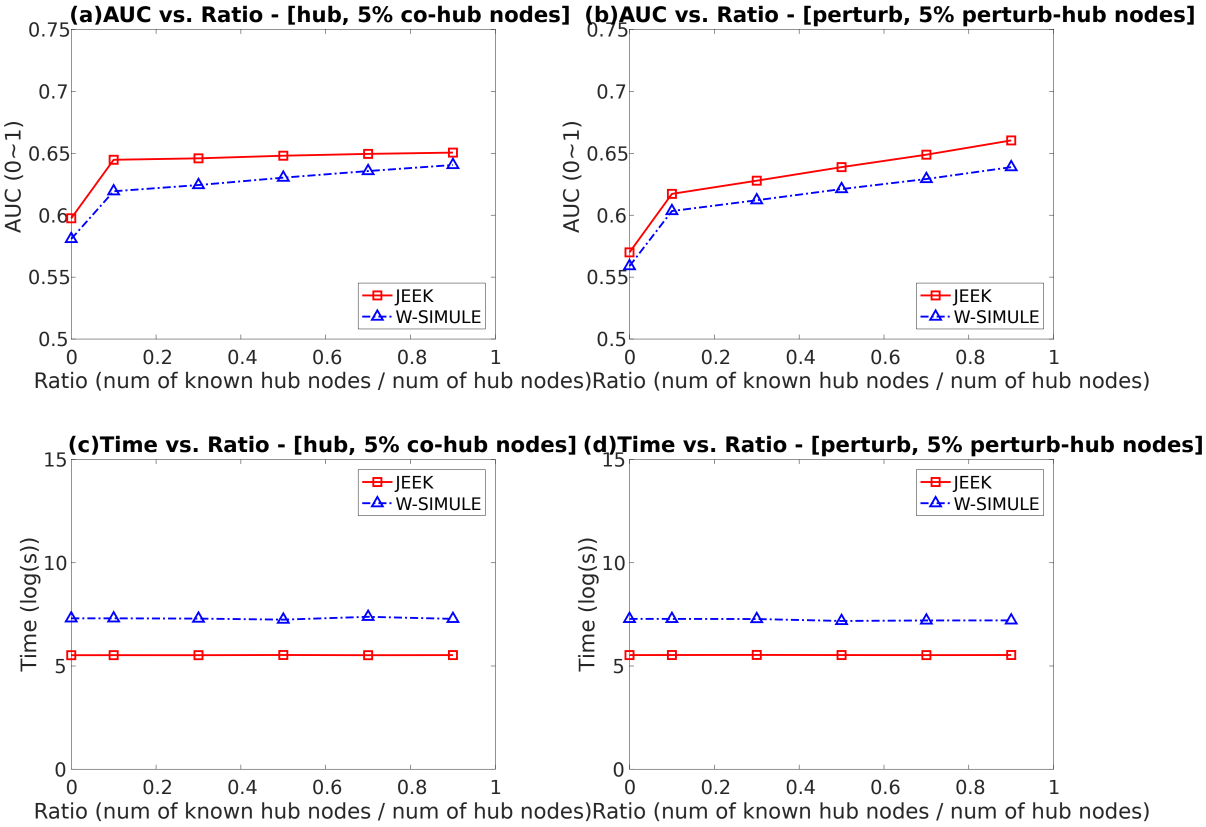


Figure 10. AUC-Score vs. ratio of number of known hub nodes to number of total hub nodes for (a) Cohub node structure (b) perturbed node structure. Computational Time Cost vs. ratio of number of known hub nodes to number of total hub nodes for (a) Cohub node structure (b) perturbed node structure.

dencies vary from normal to abnormal or across different diseases. These structural variations are highly likely to be contributing markers that influence or cause the diseases.

Two major cell contexts are selected from the human expression dataset provided by (McCall et al., 2011): leukemia cells (including 895 samples and normal blood cells (including 227 samples). Then we choose the top 1000 features from the total 12,704 features (ranked by variance) and perform graph estimation on this two-task dataset. We explore two type of knowledge in the experiments.

The first kind (DAVID) is about the known group information about nodes, such as genes belonging to the same biological pathway or cellular location. We use the popular "functional enrichment" analysis tool DAVID (Da Wei Huang & Lempicki, 2008) to get a set of group information about the 1000 genes. Multiple different types of groups are provided by DAVID and we pick the co-pathway. We only use the grouping information covering 20% of the nodes (randomly picked from 1000). The derived dependency graphs are compared by using the number of predicted edges being validated by three major existing protein/gene interaction databases (Prasad et al., 2009; Orchard et al., 2013; Stark et al., 2006) (average over both cell contexts).

The second type (PPI) is using existing known edges as the knowledge, like the known protein interaction databases for discovering gene networks (a semi-supervised setting for such estimations). We use three major existing protein/gene interaction databases (Prasad et al., 2009; Orchard et al., 2013; Stark et al., 2006). We only use the known interaction edge information covering 20% of the nodes (randomly picked from 1000). The derived dependency graphs are compared by using the number of predicted edges that are not part of the known knowledge and are being validated by three major existing protein/gene interaction databases (Prasad et al., 2009; Orchard et al., 2013; Stark et al., 2006) (average over both cell contexts).

We would like to point out that the interactions JEEK and baselines find represent statistical dependencies between

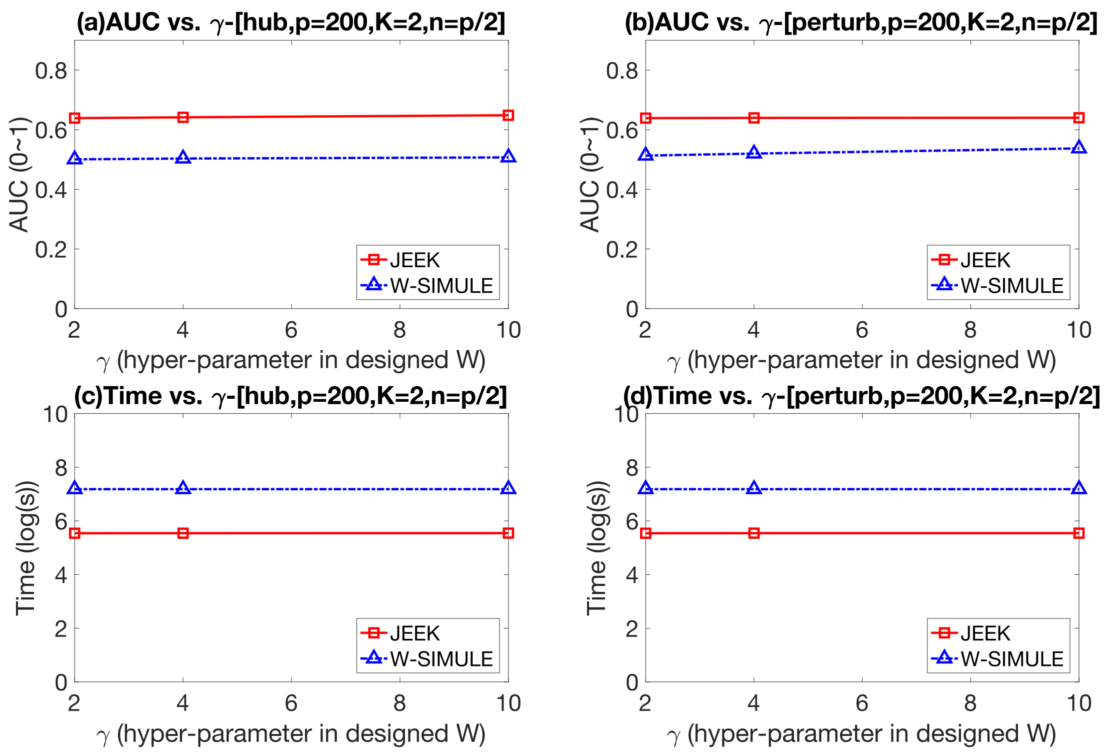


Figure 11. AUC-Score vs. γ (a) Cohub node structure for (b) perturbed node structure. Computational Time Cost vs. γ for (a) Cohub node structure (b) perturbed node structure.

genes that vary across multiple cell types. There exist many possibilities for such interactions, including like physical protein-protein interactions, regulatory gene pairs or signaling relationships. Therefore, we combine multiple existing databases for a joint validation. The numbers of matches between interactions in databases and those edges predicted by each method have been shown as the y-axis in Figure 3(c). It clearly shows that JEEK consistently outperforms two baselines.

14.3. More Experiment: Simulated Samples about Brain Connectivity with Distance as Knowledge

In this set of experiments, we confirm JEEK's ability to harvest additional knowledge using brain connectivity simulation data. Following (Bu & Lederer, 2017), we employ the known Euclidean distance between brain regions as additional knowledge W to generate simulated datasets. To generate the simulated graphs, we use $p_{j,k} = inv.logit(10 - W_{j,k}/3)$ as the probability of an edge between nodes j and k in the graphs, where $W_{j,k}$ is the Euclidean distance between regions j and k of the brain.

The generate datasets all have $p = 116$ corresponding to the number of brain regions in the distance matrix shared by (Bu & Lederer, 2017). We vary K from the set $\{2, 3, 4\}$ with $n = p/2$. The F1-scores for JEEK, JEEK-NK and W-SIMULE is the best F1-score after tuning over λ_n. The hyperparameter tuning for NAK is done by the package itself.

Simulated brain data generation model: We generate multiple sets of synthetic multivariate-Gaussian datasets. To imitate brain connectivity, we use the Euclidean distance between the brain regions as additional knowledge W where $W_{j,k}$ is the Euclidean distance between regions j and k. We fix $p = 116$ corresponding to the number of brain regions (Bu & Lederer, 2017). We generate the graph $\Omega^{(i)}$ following $\Omega^{(i)} = \mathbf{B}_I^{(i)} + \mathbf{B}_S + \delta I$, where each off-diagonal entry in $\mathbf{B}_I^{(i)}$ is generated independently and equals 0.5 with probability $p_{j,k} = inv.logit(10 - W_{j,k}/3)$ and 0 with probability $1 - p_{j,k}$ (Bu & Lederer, 2017). Similarly, the shared part $\mathbf{B}_S$ is generated independently and equal to 0.5 with probability $p_{j,k} = inv.logit(10 - W_{j,k}/3)$ and 0 with probability $1 - p_{j,k}$. δ is selected large enough to guarantee the positive definiteness. This choice ensures there are more direct connections between close regions, effectively simulating brain connectivity. For each case of simulated data generation, we generate K blocks of data samples following the

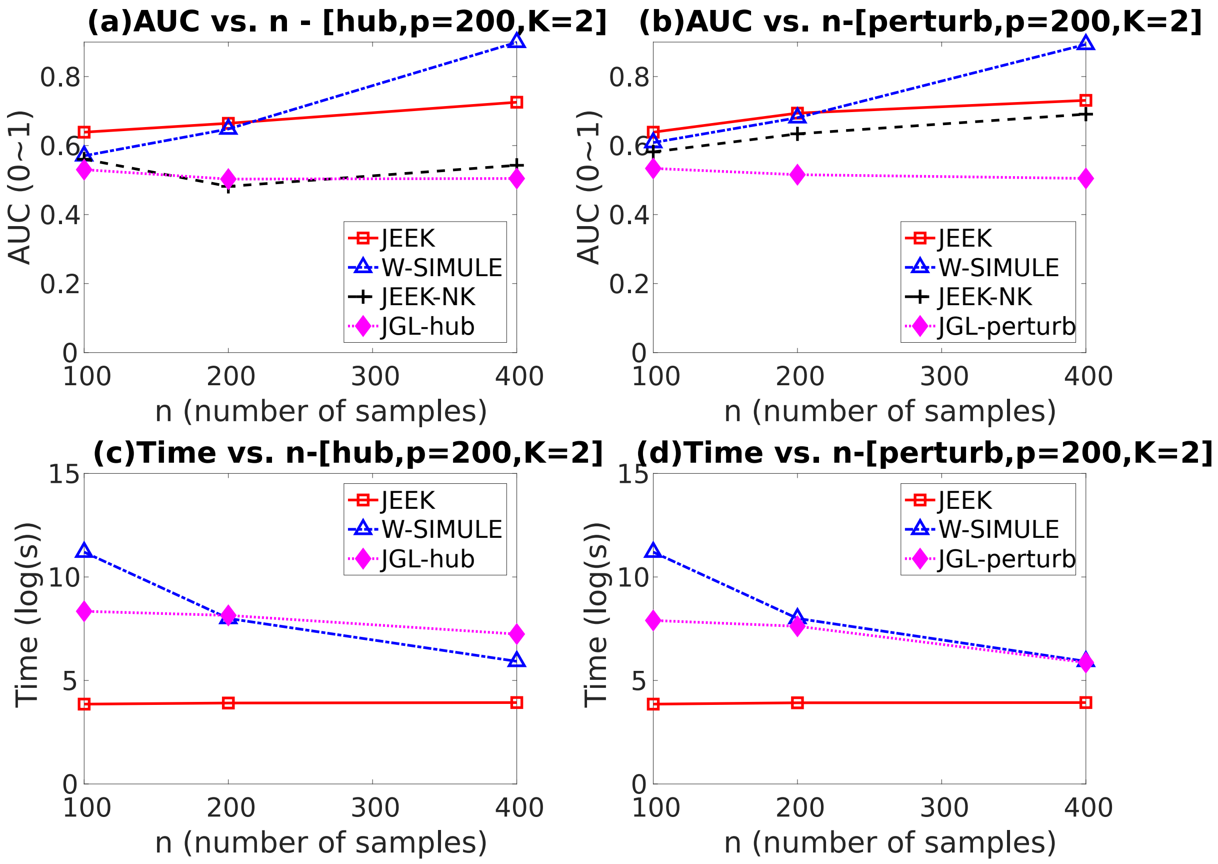


Figure 12. AUC vs. number of samples n for (a) cohub node structure (b) perturbed node structure. Computational Time Cost vs. number of samples for (c) cohub node structure and (d) perturbed node structure.

distribution $N(0, (\Omega^{(i)})^{-1})$. Details see Section 13.1.

Experimental baselines: We choose W-SIMULE, NAK and JEEK with no additional knowledge(JEEK-NK) as the baselines. (see Section 5).

Experiment Results: We compare JEEK with the baselines regarding two aspects– (a) Scalability (Computational time cost), and (b) Effectiveness (F1-score). Figure 4(a) and Figure 4(b) respectively show the F1-score vs. computational time cost with varying number of tasks K and the number of samples n. In these experiments, $p = 116$ corresponding to the number of brain regions in the distance matrix provided by (Bu & Lederer, 2017). In Figure 4(a), we vary K in the set $\{2, 3, 4\}$ with $n = p/2$. In Figure 4(b), we vary n in the set $\{p/2, p, 2p\}$ and fix $K = 2$. The F1-score plotted for JEEK, JEEK-NK and W-SIMULE is the best F1-score after tuning over λ_n. The hyperparameter tuning for NAK is done by the package itself. For each simulation case, the computation time for each estimator is the summation of a method's execution time over all values of λ_n. The points in the top left region of Figure 4 indicate higher F1-score and lower computational cost. Clearly, JEEK outperforms its baselines as all JEEK points are in the top left region of Figure 4. JEEK has a consistently higher F1-Score and is almost 6 times faster than W-SIMULE in the high dimensional case. JEEK performs better than JEEK-NK, confirming the advantage of integrating additional knowledge in graph estimation. While NAK is fast, its F1-Score is nearly 0 and hence, not useful for multi-sGGM estimation.

14.4. More Experiment: Brain Connectivity Estimation from Real-World fMRI

Experimental Baselines: We choose W-SIMULE as the the baseline in this experiment. We also compare JEEK to JEEK-NK and W-SIMULE-NK to demonstrate the need for additional knowledge in graph estimation.

ABIDE Dataset: This data is from the Autism Brain Imaging Data Exchange (ABIDE) (Di Martino et al., 2014), a publicly available resting-state fMRI dataset. The ABIDE data aims to understand human brain connectivity and how it reflects neural disorders (Van Essen et al., 2013). The data is retrieved from the Preprocessed Connectomes Project (Craddock, 2014), where preprocessing is performed using the Configurable Pipeline for the Analysis of Connectomes (CPAC) (Craddock et al., 2013) without global signal correction or band-pass filtering. After preprocessing with this pipeline, 871 individuals remain (468 diagnosed with autism). Signals for the 160 (number of features $p = 160$) regions of interest (ROIs) in the often-used Dosenbach Atlas (Dosenbach et al., 2010) are examined.

Distance as Additional Knowledge: To select the weights $\{W_I^{(i)}, W_S\}$, two separate spatial distance matrices W were derived from the Dosenbach atlas. The first, referred to as *anatomical*^{i}, gives each ROI one of 40 well-known, anatomic labels (*e.g.* "basal ganglia", "thalamus"). Weights $W_{j,k}$ take the low value i if two ROIs have the same label, and the high value $10 - i$ otherwise. The second additional knowledge matrix, referred to as *dist*^{i}, sets the weight of each edge ($W_{j,k}$) to its spatial length, in MNI space[9], raised to the power i. Then $W_I^{(i)} = W_S = W$.

Cross-validation: Classification is performed using the 3-fold cross-validation suggested by the literature (Poldrack et al., 2008)(Varoquaux et al., 2010). The subjects are randomly partitioned into three equal sets: a training set, a validation set, and a test set. Each estimator produces $\widehat{\Omega}^{(1)} - \widehat{\Omega}^{(2)}$ using the training set. Then, these differential networks are used as inputs to linear discriminant analysis (LDA), which is tuned via cross-validation on the validation set. Finally, accuracy is calculated by running LDA on the test set. This classification process aims to assess the ability of an estimator to learn the differential patterns of the connectome structures. We cannot use NAK to perform classification for this task, as NAK outputs only an adjacency matrix, which cannot be used for estimation using LDA.

Parameter variation: The results are fairly robust to variations of the W. (see Table 2). The effect of changing W seems to have a fairly small effect on the log-likelihood of the model. This is likely because both penalize picking physically long edges, which agrees with observations from neuroscience. The *dist* W effectively encourages the selection of short edges, and the *anatomical* W also has substantial spatial localization.

[9] MNI space is a coordinate system used to refer to analagous points on different brains.

References

Banerjee, O., El Ghaoui, L., and d'Aspremont, A. Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data. *The Journal of Machine Learning Research*, 9:485–516, 2008.

Bickel, P. J. and Levina, E. Covariance regularization by thresholding. *The Annals of Statistics*, pp. 2577–2604, 2008.

Bu, Y. and Lederer, J. Integrating additional knowledge into estimation of graphical models. *arXiv preprint arXiv:1704.02739*, 2017.

Caruana, R. Multitask learning. *Machine learning*, 28(1): 41–75, 1997.

Chiquet, J., Grandvalet, Y., and Ambroise, C. Inferring multiple graphical structures. *Statistics and Computing*, 21(4):537–553, 2011.

Consortium, E. P. et al. An integrated encyclopedia of dna elements in the human genome. *Nature*, 489(7414):57–74, 2012.

Craddock, C. Preprocessed connectomes project: open sharing of preprocessed neuroimaging data and derivatives. In *61st Annual Meeting*. AACAP, 2014.

Craddock, C., Sikka, S., Cheung, B., Khanuja, R., Ghosh, S., Yan, C., Li, Q., Lurie, D., Vogelstein, J., Burns, R., et al. Towards automated analysis of connectomes: The configurable pipeline for the analysis of connectomes (c-pac). *Front Neuroinform*, 42, 2013.

Da Wei Huang, B. T. S. and Lempicki, R. A. Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources. *Nature protocols*, 4(1):44–57, 2008.

Danaher, P., Wang, P., and Witten, D. M. The joint graphical lasso for inverse covariance estimation across multiple classes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 2013.

Di Martino, A., Yan, C.-G., Li, Q., Denio, E., Castellanos, F. X., Alaerts, K., Anderson, J. S., Assaf, M., Bookheimer, S. Y., Dapretto, M., et al. The autism brain imaging data exchange: towards a large-scale evaluation of the intrinsic brain architecture in autism. *Molecular psychiatry*, 19(6): 659–667, 2014.

Dosenbach, N. U., Nardos, B., Cohen, A. L., Fair, D. A., Power, J. D., Church, J. A., Nelson, S. M., Wig, G. S., Vogel, A. C., Lessov-Schlaggar, C. N., et al. Prediction of individual brain maturity using fmri. *Science*, 329(5997): 1358–1361, 2010.

Prior	Sparsity=8%		Sparsity=16%	
	Log-Likelihood	Test Accuracy	Log-Likelihood	Test Accuracy
No Additional Knowledge	-294.34	0.56	-283.27	0.55
dist	-289.12	0.53	-285.69	0.55
dist^2	-283.78	0.54	-282.92	0.54
anatomical^1	-292.42	0.56	-289.34	0.57
anatomical^2	-291.29	0.58	-285.63	0.56

Table 2. Variations of the W and multi-task component yield fairly stable results.

Guo, J., Levina, E., Michailidis, G., and Zhu, J. Joint estimation of multiple graphical models. *Biometrika*, pp. asq060, 2011.

Honorio, J. and Samaras, D. Multi-task learning of gaussian graphical models. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pp. 447–454, 2010.

Ideker, T. and Krogan, N. J. Differential network biology. *Molecular systems biology*, 8(1):565, 2012.

Kelly, C., Biswal, B. B., Craddock, R. C., Castellanos, F. X., and Milham, M. P. Characterizing variation in the functional connectome: promise and pitfalls. *Trends in cognitive sciences*, 16(3):181–188, 2012.

Lauritzen, S. L. *Graphical models*, volume 17. Clarendon Press, 1996.

McCall, M. N., Uppal, K., Jaffee, H. A., Zilliox, M. J., and Irizarry, R. A. The gene expression barcode: leveraging public data repositories to begin cataloging the human and murine transcriptomes. *Nucleic acids research*, 39 (suppl 1):D1011–D1015, 2011.

Mohan, K., London, P., Fazel, M., Lee, S.-I., and Witten, D. Node-based learning of multiple gaussian graphical models. *arXiv preprint arXiv:1303.5145*, 2013.

Monti, R. P., Anagnostopoulos, C., and Montana, G. Learning population and subject-specific brain connectivity networks via mixed neighborhood selection. *arXiv preprint arXiv:1512.01947*, 2015.

Negahban, S., Yu, B., Wainwright, M. J., and Ravikumar, P. K. A unified framework for high-dimensional analysis of m-estimators with decomposable regularizers. In *Advances in Neural Information Processing Systems*, pp. 1348–1356, 2009.

Network, C. G. A. R. et al. Integrated genomic analyses of ovarian carcinoma. *Nature*, 474(7353):609–615, 2011.

Orchard, S., Ammari, M., Aranda, B., Breuza, L., Briganti, L., Broackes-Carter, F., Campbell, N. H., Chavali, G., Chen, C., Del-Toro, N., et al. The MIntAct project IntAct as a common curation platform for 11 molecular interaction databases. *Nucleic acids research*, pp. gkt1115, 2013.

Pang, H., Liu, H., and Vanderbei, R. The fastclime package for linear programming and large-scale precision matrix estimation in r. *Journal of Machine Learning Research*, 15:489–493, 2014.

Park, T. and Casella, G. The bayesian lasso. *Journal of the American Statistical Association*, 103(482):681–686, 2008.

Poldrack, R. A., Fletcher, P. C., Henson, R. N., Worsley, K. J., Brett, M., and Nichols, T. E. Guidelines for reporting an fmri study. *Neuroimage*, 40(2):409–414, 2008.

Poldrack, R. A., Barch, D. M., Mitchell, J., Wager, T., Wagner, A. D., Devlin, J. T., Cumba, C., Koyejo, O., and Milham, M. Toward open sharing of task-based fmri data: the openfmri project. *Frontiers in neuroinformatics*, 7:12, 2013.

Prasad, T. K., Goel, R., Kandasamy, K., Keerthikumar, S., Kumar, S., Mathivanan, S., Telikicherla, D., Raju, R., Shafreen, B., Venugopal, A., et al. Human protein reference database?2009 update. *Nucleic acids research*, 37 (suppl 1):D767–D772, 2009.

Ravikumar, P., Wainwright, M. J., Raskutti, G., Yu, B., et al. High-dimensional covariance estimation by minimizing l1-penalized log-determinant divergence. *Electronic Journal of Statistics*, 5:935–980, 2011.

Rothman, A. J., Bickel, P. J., Levina, E., Zhu, J., et al. Sparse permutation invariant covariance estimation. *Electronic Journal of Statistics*, 2:494–515, 2008.

Rothman, A. J., Levina, E., and Zhu, J. Generalized thresholding of large covariance matrices. *Journal of the American Statistical Association*, 104(485):177–186, 2009.

Shimamura, T., Imoto, S., Yamaguchi, R., and Miyano, S. Weighted lasso in graphical gaussian modeling for large gene network estimation based on microarray data. In

Genome Informatics 2007: Genome Informatics Series Vol. 19, pp. 142–153. World Scientific, 2007.

Singh, C., Wang, B., and Qi, Y. A constrained, weighted-l1 minimization approach for joint discovery of heterogeneous neural connectivity graphs. *arXiv preprint arXiv:1709.04090*, 2017.

Stark, C., Breitkreutz, B.-J., Reguly, T., Boucher, L., Breitkreutz, A., and Tyers, M. Biogrid: a general repository for interaction datasets. *Nucleic acids research*, 34 (suppl_1):D535–D539, 2006.

Uddin, L. Q., Supekar, K., Lynch, C. J., Khouzam, A., Phillips, J., Feinstein, C., Ryali, S., and Menon, V. Salience network–based classification and prediction of symptom severity in children with autism. *JAMA psychiatry*, 70(8):869–879, 2013.

Van Essen, D. C., Smith, S. M., Barch, D. M., Behrens, T. E., Yacoub, E., Ugurbil, K., Consortium, W.-M. H., et al. The wu-minn human connectome project: an overview. *Neuroimage*, 80:62–79, 2013.

Varoquaux, G., Gramfort, A., Poline, J.-B., and Thirion, B. Brain covariance selection: better individual functional connectivity models using population prior. In *Advances in neural information processing systems*, pp. 2334–2342, 2010.

Wainwright, M. J. and Jordan, M. I. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1-2):1–305, 2008.

Wang, B., Gao, J., and Qi, Y. A fast and scalable joint estimator for learning multiple related sparse gaussian graphical models. In *Artificial Intelligence and Statistics*, pp. 1168–1177, 2017a.

Wang, B., Singh, R., and Qi, Y. A constrained l1 minimization approach for estimating multiple sparse gaussian or nonparanormal graphical models. *Machine Learning*, 106 (9-10):1381–1417, 2017b.

Watts, D. J. and Strogatz, S. H. Collective dynamics of 'small-world'networks. *nature*, 393(6684):440–442, 1998.

Yang, E., Lozano, A., and Ravikumar, P. Elementary estimators for high-dimensional linear regression. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pp. 388–396, 2014a.

Yang, E., Lozano, A. C., and Ravikumar, P. Elementary estimators for graphical models. In *Advances in Neural Information Processing Systems*, pp. 2159–2167, 2014b.

Yang, E., Lozano, A. C., and Ravikumar, P. Elementary estimators for sparse covariance matrices and other structured moments. In *ICML*, pp. 397–405, 2014c.

Yuan, M. and Lin, Y. Model selection and estimation in the gaussian graphical model. *Biometrika*, 94(1):19–35, 2007.

Zhang, B. and Wang, Y. Learning structural changes of gaussian graphical models in controlled experiments. *arXiv preprint arXiv:1203.3532*, 2012.

Zhang, Y. and Schneider, J. G. Learning multiple tasks with a sparse matrix-normal penalty. In *Advances in Neural Information Processing Systems*, pp. 2550–2558, 2010.

Zhu, Y., Shen, X., and Pan, W. Structural pursuit over multiple undirected graphs. *Journal of the American Statistical Association*, 109(508):1683–1696, 2014.