mathematics

MDPI

Article

Jewel: A Novel Method for Joint Estimation of Gaussian Graphical Models

Claudia Angelini [1], Daniela De Canditiis [2] and Anna Plaksienko [1,3,*]

[1] Istituto per le Applicazioni del Calcolo "Mauro Picone", CNR-Napoli, 80131 Naples, Italy; c.angelini@iac.cnr.it
[2] Istituto per le Applicazioni del Calcolo "Mauro Picone", CNR-Roma, 00185 Rome, Italy; d.decanditiis@iac.cnr.it
[3] Gran Sasso Science Institute, 67100 L'Aquila, Italy
* Correspondence: anna.plaksienko@gssi.it

Abstract: In this paper, we consider the problem of estimating multiple Gaussian Graphical Models from high-dimensional datasets. We assume that these datasets are sampled from different distributions with the same conditional independence structure, but not the same precision matrix. We propose *jewel*, a joint data estimation method that uses a node-wise penalized regression approach. In particular, *jewel* uses a group Lasso penalty to simultaneously guarantee the resulting adjacency matrix's symmetry and the graphs' joint learning. We solve the minimization problem using the group descend algorithm and propose two procedures for estimating the regularization parameter. Furthermore, we establish the estimator's consistency property. Finally, we illustrate our estimator's performance through simulated and real data examples on gene regulatory networks.

Keywords: Gaussian Graphical Model; group Lasso; joint estimation; network estimation

Citation: Angelini, C.; De Canditiis, D.; Plaksienko, A. *Jewel*: A Novel Method for Joint Estimation of Gaussian Graphical Models. *Mathematics* **2021**, *9*, 2105. https://doi.org/10.3390/math9172105

Academic Editors: Marina Alexandra Pedro Andrade and Maria Alves Teodoro

Received: 12 July 2021
Accepted: 26 August 2021
Published: 31 August 2021

1. Introduction

Network analysis is becoming a powerful tool for describing the complex systems that arise in physical, biomedical, epidemiological, and social sciences, see in [1,2]. In particular, estimating network structure and its complexity from high-dimensional data has been one of the most relevant statistical challenges of the last decade [3]. The mathematical framework to use depends on the type of relationship among the variables that the network should incorporate. For example, in the context of gene regulatory networks (GRN), traditional co-expression methods are useful to capture marginal correlation among genes without distinguishing between direct or mediated gene interactions. Instead, graphical models (GM) constitute a well-known framework for describing conditional dependency relationships between random variables in a complex system. Therefore, they are more suited to describe direct relations among genes, not mediated by the remaining genes.

In the GMs' framework, an undirected graph $G = (V, E)$ describes the joint distribution of the random vector $(X_1, \ldots, X_p)$, the individual variables being the graph's nodes and the edges reflecting the presence/absence of conditional dependency relation among them. When the system of random variables has a multivariate Gaussian distribution $(X_1, \ldots, X_p) \sim N(0, \mathbf{\Sigma})$, we refer to Gaussian Graphical Models (GGMs). In such a case, the graph estimation is equivalent to estimating the precision matrix's support, namely, the inverse of the covariance matrix associated with the multivariate Gaussian distribution.

There is extensive literature on learning a GGM, and we refer to the works in [4–7] for an overview. In brief, under high-dimensional setting and sparsity assumptions on the precision matrix, we can broadly classify the available methods into two categories: methods that estimate the entire precision matrix $\mathbf{\Omega}$ and those that estimate only its support (i.e., the edge set E). Methods in the first category usually impose a Lasso type penalty on the inverse covariance matrix entries when maximizing the log-likelihood, as the GLasso

approach proposed in [8,9]. Methods in the second category date back to the seminal paper of Meinshausen and Bühlmann [10] and use the variable selection properties of Lasso regression.

In recent years, the focus shifted from the inference of a single graph given a dataset to the inference of multiple graphs given different datasets, assuming that the graphs share some common structure. This setting encompasses cases where datasets measure similar entities (variables) under different conditions or setups. It is useful for describing various situations encountered in real applications, such as meta-analyses and heterogeneous data integration. For example, in GNR, a single dataset might represent the gene expression levels of n patients affected by a particular type of cancer, and the inference aims to understand the specific regulatory mechanisms. International projects, such as The Cancer Genome Atlas (TGCA), or large repositories, such as Gene Expression Omnibus (GEO), make available tens of thousands of gene expression samples. Therefore, it is easy to find several datasets (i.e., different studies collected using different high-throughput assays or performed in different laboratories) on the experimental condition of interest. Assuming that, despite the difference in the technologies and preprocessing, the hidden regulatory mechanisms investigated in the different studies are similar if not the same, it is worth integrating them to improve the inference. A similar type of challenge is also present in the integrative multi-omics analysis. For instance, one can measure gene expression, single nucleotide polymorphisms (SNPs), methylation levels, etc. on the same set of individuals and want to combine the information across the different omics. In both cases, it is possible to develop multitasking or joint learning approaches to gain power in the inference (see [11,12] to cite few examples).

More specifically, in multiple GGMs inference, we have $K \geq 2$ different datasets $\mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(K)}$, each drawn from a Gaussian distribution $N(0, \mathbf{\Sigma}^{(k)})$. Each dataset measures (almost) the same sets of variables in a specific class (or condition) k. The aim is to estimate the underlying GGMs under the assumption that they share some common structure across the classes.

Several methods are already available in the literature to deal with such a problem. They differ on the assumptions on how the conditional dependency structure is shared across the various distributions. The majority of these methods extend maximum likelihood (MLE) approaches to the multiple data framework. Among the most important contributions, we cite the work in [13] that penalizes the MLE by a hierarchical penalty enforcing similar sparsity patterns across classes, allowing some class-specific difference. However, as the penalty is not convex, convergence is not guaranteed. JGL [14] is another important method that proposes two alternatives. The first penalizes the MLE by a fused Lasso penalty, and the second penalizes the MLE by combining two group Lasso penalties. In this last alternative, the two penalties enforce a similar sparsity structure and some differences, respectively. Two other approaches are [15], which penalizes the MLE by an adaptive Lasso penalty whose weights are updated iteratively, and the work in [16], which penalizes the MLE by a group Lasso penalty. On the other hand, the work in [17,18] extends the regression-based approach in [10] to the multiple data setting. In particular, in [17], the authors constructed a two-step procedure. In the first step, they infer the graphical structure using a regression-based approach with a group Lasso penalty for exploiting the common structure across the classes. In the second step, they penalize the MLE subject to the first step edges' set. The authors of [18] proposed regression-based minimization problem with cooperative group Lasso penalty. The first term is the standard group Lasso penalty which enforces the same structure across classes by penalizing elements of precision matrices. The second one penalizes negative elements of those matrices to promote differences between classes. By authors' hypothesis, swap in the sign can occur only for class-specific connections, justifying such penalty.

In this paper, we propose *jewel*, a new technique for jointly estimate a GGM in the multiple dataset framework. We assume that the underlying graph structure is the same across the K classes. However, each class preserves its specific precision matrix. We extend

the regression-based method proposed in [10] to the case of multiple datasets, in the same spirit as in [18] and the first step of the procedure in [17], which mainly differ in the definition of variables' groups. More specifically, in [17,18] the groups are determined by the ij positions across the K precision matrices and in our approach, the groups are determined by both ij and ji positions across the K precision matrices. Consequently, both [17,18] do not provide symmetric estimates of precision matrices' supports and require postprocessing. Instead, *jewel* grouping strategy exploits both the common structure across the datasets and the symmetry of the precision matrices support. We consider this a valuable improvement compared to the competitors as such an approach allows to avoid the need for postprocessing.

The rest of the paper is organized as follows. In Section 2, we derive the *jewel* method and present the numerical algorithm for its solution, as well as establish the theoretical properties of the estimator in Section 2.4 and discuss the choice of the tuning parameter in Section 2.5. Code is provided in Section 2.6. Finally, in Section 3, we illustrate our method's performance in a simulation study comparing it with some other available alternatives and describe a real data application from gene expression datasets.

2. *Jewel*: Joint Node-Wise Estimation of Multiple Gaussian Graphical Models

This section describes the mathematical setup, the proposed method for the joint estimation of GGMs (*jewel*), the numerical algorithm adopted, the theoretical property of the proposed estimator and approaches of estimating the regularization parameter.

2.1. Problem Setup

Let $\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \ldots, \mathbf{X}^{(K)}$ be $K \geq 2$ datasets of dimension $n_k \times p_k$, respectively, that represent similar entities measured under K different conditions or collected in distinct classes. Each dataset $\mathbf{X}^{(k)}$ represents n_k observations $\mathbf{x}_1^{(k)}, \ldots, \mathbf{x}_{n_k}^{(k)}$, where each $\mathbf{x}_i^{(k)} = (x_{i1}^{(k)}, \ldots, x_{ip_k}^{(k)})$ is a p_k-dimensional row vector. Without loss of generality, assume that each data matrix is standardized to have columns with zero mean and variance equal to one.

The proposed model assumes that $\mathbf{x}_1^{(k)}, \ldots, \mathbf{x}_{n_k}^{(k)}$ represent independent samples from a $\mathcal{N}(0, \mathbf{\Sigma}^{(k)})$, with $\mathbf{\Sigma}^{(k)} \succ 0$. Moreover, we also assume that most of the variables are in common in all the K datasets. For example, this assumption includes datasets that measure the same set of p variables under K different conditions; however, few variables might not be observed in some datasets due to some technical failure. Under this setup, the model assumes that the variables share the same conditional independence structure across the K datasets.

Precisely, let $G_k = (V_k, E_k)$ be the undirected graph that describes the conditional independence structure of the k-th distribution $\mathcal{N}(0, \mathbf{\Sigma}^{(k)})$, i.e., $V_k = \{X_1, \ldots, X_{p_k}\}$ and $(i,j) \notin E_k \iff X_i \perp\!\!\!\perp X_j | X_{\{l, l \neq i,j\}}$ where $\{X_i, X_j\} \subseteq V_k$. Note that we use notation (i,j) to denote the undirected edge incident to vertices i and j, or equivalently j and i (if such edge exists). We use notation $\{i,j\}$ to denote the pair of vertices or variables $\{X_i, X_j\}$. We assume that there exists a common undirected graph $G = (V, E)$ with $V = V_1 \cup \cdots \cup V_K$ and $E \subseteq V \times V$ such that $(i,j) \notin E \iff X_i \perp\!\!\!\perp X_j | X_{\{l, l \neq i,j\}}$ with $\{X_i, X_j\} \subseteq V$. In other words, two variables of V are conditionally independent in all the distributions of which they are part or they are never conditionally independent in any. However, when they are not conditionally independent, the conditional correlation might be different in the different datasets to model relations that can be differently tuned depending on specific experimental condition or set-up.

Let us denote $\mathbf{\Omega}^{(k)} = (\mathbf{\Sigma}^{(k)})^{-1}$ the true precision matrix for k-th distribution. As inferring a GGM is equivalent to estimating the support of the precision matrix, estimating a GGM from multiple datasets translates into simultaneously inferring the support of all precision matrices $\mathbf{\Omega}^{(k)}$, with the constraint $\Omega_{ij}^{(k)} = \Omega_{ji}^{(k)} = 0$ for all k such that $\{X_i, X_j\} \subseteq V_k$. We emphasize here that the aim is to estimate only the structure of the common network, not the entries of precision matrices.

2.2. Jewel

jewel is inspired by the node-wise regression-based procedure proposed in [10] for inferring a GGM from a single dataset. Let us first revise this method. Let fix a single dataset, say k, and define $\mathbf{\Theta}^{(k)}$, the $p_k \times p_k$ zero diagonal matrix with extra-diagonal entries $\Theta_{ij}^{(k)} = -\Omega_{ij}^{(k)}/\Omega_{jj}^{(k)}$. As $\Omega_{jj}^{(k)} > 0$ for the positiveness of $\mathbf{\Sigma}^{(k)}$, the support of $\mathbf{\Omega}^{(k)}$ coincides with the support of $\mathbf{\Theta}^{(k)}$. In [10], the authors proposed to learn matrix $\mathbf{\Theta}^{(k)}$ instead of $\mathbf{\Omega}^{(k)}$. To this aim, they solved the following multivariate regression problem with the Lasso penalty

$$\hat{\mathbf{\Theta}}^{(k)} = \underset{\substack{\mathbf{\Theta}^{(k)} \in \mathbb{R}^{p_k \times p_k} \\ diag=0}}{\arg\min} \left\{ \frac{1}{2n_k} ||\mathbf{X}^{(k)} - \mathbf{X}^{(k)}\mathbf{\Theta}^{(k)}||_F^2 + \lambda \sum_{i \neq j} |\Theta_{ij}^{(k)}| \right\}, \tag{1}$$

where λ is a tuning regularization parameter and $||\cdot||_F$ is the Frobenius norm. Note that problem in Equation (1) is separable into p_k independent univariate regression problems where each column of matrix $\mathbf{\Theta}^{(k)}$ is obtained independently from the others. Although computationally efficient, such an approach does not exploit either guarantee the symmetry of the solution. Therefore, the authors proposed to post-process the solution to restore the symmetry, for example by the "AND" or "OR" rules. More recently, the authors of [19] proposed a modified approach where $\mathbf{\Theta}^{(k)}$ is obtained by solving the following minimization problem:

$$\begin{aligned} \hat{\mathbf{\Theta}}^{(k)} = \underset{\substack{\mathbf{\Theta}^{(k)} \in \mathbb{R}^{p_k \times p_k} \\ diag=0}}{\arg\min} \Bigg\{ & \frac{1}{2n_k} ||\mathbf{X}^{(k)} - \mathbf{X}^{(k)}\mathbf{\Theta}^{(k)}||_F^2 + \\ & \lambda\sqrt{2} \sum_{i<j=1}^{p} \sqrt{\left(\Theta_{ij}^{(k)}\right)^2 + \left(\Theta_{ji}^{(k)}\right)^2} \Bigg\}, \end{aligned} \tag{2}$$

that corresponds to a multivariate regression problem with a group Lasso penalty. In Equation (2), the number of unknown parameters, i.e., the extra-diagonal terms of $\mathbf{\Theta}^{(k)}$, are $p_k(p_k - 1)$ and are arranged into $\binom{p_k}{2}$ groups of dimension 2, each group including $\Theta_{ij}^{(k)}$ and $\Theta_{ji}^{(k)}$. As a consequence, the minimization problem in Equation (2) results in a group sparse estimate of $\hat{\mathbf{\Theta}}^{(k)}$ which exploits and hence guarantees the symmetry of the estimated support.

In this work, we further extend the approach in [19] to simultaneously learn the K zero-diagonal matrices $\mathbf{\Theta}^{(k)}$, each with $p_k(p_k - 1)$ unknowns parameters. Let $p = |V|$ be the cardinality of V and divide its elements into $\binom{p}{2}$ groups. Each group consists of pairs of variables $\Theta_{ij}^{(k)}$ and $\Theta_{ji}^{(k)}$ across all the datasets that contain them. More precisely, for all $1 \leq i < j \leq p$, we group together variables $\{\Theta_{ij}^{(k)}, \Theta_{ji}^{(k)} : \{X_i, X_j\} \subseteq V_k\}$ and denote g_{ij} the group's cardinality. Based on our hypothesis, we have that the vector of each group of variables coincides with the zero vector when the variables X_i and X_j are conditionally independent in all the datasets that contain them (i.e., when $(i,j) \notin E$), or it is different from zero when the variables X_i and X_j are conditionally dependent in all the data sets that contain them (i.e., when $(i,j) \in E$). In the latter case, each data set can modulate the strength of the dependency in a specific way because, when not equal to zero, the group elements are not forced to be equal. This is another advantage of our proposal to treat groups as symmetric pairs across all the datasets.

Therefore, in this paper we propose *jewel*, which estimates simultaneously $\hat{\boldsymbol{\Theta}}^{(1)}, \ldots, \hat{\boldsymbol{\Theta}}^{(K)}$ by solving the following minimization problem:

$$(\hat{\boldsymbol{\Theta}}^{(1)}, \ldots, \hat{\boldsymbol{\Theta}}^{(K)}) = \underset{\substack{\boldsymbol{\Theta}^{(1)} \in \mathbb{R}^{p_1 \times p_1} \\ \vdots \\ \boldsymbol{\Theta}^{(K)} \in \mathbb{R}^{p_K \times p_K}, \\ diag=0}}{\arg\min} \left\{ \frac{1}{2} \sum_{k=1}^{K} \frac{1}{n_k} ||\mathbf{X}^{(k)} - \mathbf{X}^{(k)} \boldsymbol{\Theta}^{(k)}||_F^2 + \right. \\ \left. \lambda \sum_{i<j=1}^{p} \sqrt{g_{ij}} \sqrt{\sum_{k:\{X_i, X_j\} \subseteq V_k} \left(\Theta_{ij}^{(k)}\right)^2 + \left(\Theta_{ji}^{(k)}\right)^2} \right\}. \tag{3}$$

The estimated edge set $\hat{E}$ is obtained as $\text{supp}(\hat{\boldsymbol{\Theta}}^{(1)}) = \cdots = \text{supp}(\hat{\boldsymbol{\Theta}}^{(K)})$, that are equal by construction. Indeed, the problem in Equation (3) corresponds to a multivariate regression problem with a group Lasso penalty that enforces the same support for the estimates $\hat{\boldsymbol{\Theta}}^{(k)}$ naturally exploiting and guaranteeing the symmetry of the solution.

Note that, although the minimization problem in Equation (3) can be written and solved for a number of variables p_k that is different in each dataset, the notations and the formulas simplify a lot when we assume that all variables coincide in the K datasets. Under this assumption, for all k we have $p_k = p$ and $V_k = V$, then each group $\{\Theta_{ij}^{(k)}, \Theta_{ji}^{(k)} : \{X_i, X_j\} \subseteq V_k\}$ has cardinality $g_{ij} = 2K$, and $G = (V, E)$ is the GGM associated to each and all the datasets. Consequently, while the general formulation of Equation (3) can be useful for some applications where the variables of the different datasets may partly not coincide due to missing values or other reasons, for the sake of clarity from now on, we will use the simplified formulation, referring to remarks notes about the general formulation.

We use the following notations through the rest of the paper:

- With bold capital letters we represent matrices, with bold lower case letters—vectors, which are intended as columns if not stated otherwise;
- $\boldsymbol{\theta} = \left(\Theta_{21}^{(1)}, \ldots, \Theta_{p1}^{(1)}, \ldots, \Theta_{1p}^{(K)}, \ldots, \Theta_{(p-1)p}^{(K)}\right)^T$, $\dim(\boldsymbol{\theta}) = p(p-1)K \times 1$, is the vector of the unknown coefficients;
- $\boldsymbol{\theta}_{[ij]}$ denotes the restriction of vector $\boldsymbol{\theta}$ to the variables belonging to the group $i < j$; specifically $\boldsymbol{\theta}_{[ij]} = (\Theta_{ij}^{(1)}, \Theta_{ji}^{(1)}, ..., \Theta_{ij}^{(K)}, \Theta_{ji}^{(K)})$, thus it has length $g_{ij} = 2K$;
- λ stands for $\sqrt{2K}\lambda$,
- $X_{.i}^{(k)}$ corresponds to the i-th column of matrix $\mathbf{X}^{(k)}$ and $\mathbf{X}_{.-i}^{(k)}$ corresponds to the submatrix of $\mathbf{X}^{(k)}$ without the i-th column;
- $\mathbf{y} = \left({X_{.1}^{(1)}}^T, \ldots, {X_{.p}^{(1)}}^T, \ldots, {X_{.1}^{(K)}}^T, \ldots, {X_{.p}^{(K)}}^T\right)^T$, $\dim(\mathbf{y}) = Np \times 1$, $N = \sum_{k=1}^{K} n_k$, denotes the vector concatenating the columns of all data matrices;
-

$$\mathbf{X} = \begin{pmatrix} \begin{pmatrix} \mathbf{X}_{.-1}^{(1)} & 0 & \ldots & 0 \\ 0 & \mathbf{X}_{.-2}^{(1)} & \ldots & 0 \\ \vdots & & \ddots & \\ 0 & \ldots & & \mathbf{X}_{.-p}^{(1)} \end{pmatrix} & & \\ & \ddots & \\ & & \begin{pmatrix} \mathbf{X}_{.-1}^{(K)} & 0 & \ldots & 0 \\ 0 & \mathbf{X}_{.-2}^{(K)} & \ldots & 0 \\ \vdots & & \ddots & \\ 0 & \ldots & & \mathbf{X}_{.-p}^{(K)} \end{pmatrix} \end{pmatrix}$$

denotes the block-diagonal matrix made up by the block-diagonal matrices $\mathbf{X}^{(k)}_{\cdot -j}$, $k = 1 \dots K$, $j = 1 \dots p$, $\dim(\mathbf{X}) = Np \times p(p-1)K$;

- $\mathbf{D} = blkdiag\left(\frac{1}{\sqrt{n_k}}\mathbf{I}_{n_k p}\right)_{k=1\dots K}$, $\dim(\mathbf{D}) = Np \times Np$;
- $||\mathbf{u}||_2 = ||\mathbf{u}||$ for any vector $\mathbf{u}$;
- $||\mathbf{u}||^2_{\mathbf{D}} = \mathbf{u}^T\mathbf{D}^2\mathbf{u} = ||\mathbf{D}\mathbf{u}||^2_2$, for all $\mathbf{u} \in \mathbb{R}^{Np}$.

With these notations, $\sqrt{g_{ij}} = 2K\ \forall i, j$ and λ reparametrized as $\lambda\sqrt{2K}$, the penalty in Equation (3) can be easily rewritten using vector $\boldsymbol{\theta}_{[ij]} = (\Theta^{(1)}_{ij}, \Theta^{(1)}_{ji}, \dots, \Theta^{(K)}_{ij}, \Theta^{(K)}_{ji})$ as

$$\sum_{i<j=1}^{p} \sqrt{\sum_{k=1}^{K}\left(\Theta^{(k)}_{ij}\right)^2 + \left(\Theta^{(k)}_{ji}\right)^2} = \sum_{i<j=1}^{p} ||\boldsymbol{\theta}_{[ij]}||. \tag{4}$$

Moreover, the goodness-of-fit term becomes

$$\sum_{k=1}^{K} \frac{1}{n_k} ||\mathbf{X}^{(k)} - \mathbf{X}^{(k)}\boldsymbol{\Theta}^{(k)}||^2_F = ||\mathbf{y} - \mathbf{X}\boldsymbol{\theta}||^2_{\mathbf{D}}. \tag{5}$$

Combining Equations (4) and (5), we rewrite the minimization problem in Equation (3) as follows:

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta} \in \mathbb{R}^{p(p-1)K}}{\arg\min} \underbrace{\frac{1}{2}||\mathbf{y} - \mathbf{X}\boldsymbol{\theta}||^2_{\mathbf{D}} + \lambda \sum_{i<j=1}^{p} ||\boldsymbol{\theta}_{[ij]}||}_{F(\boldsymbol{\theta})}. \tag{6}$$

This alternative formulation will be useful to present the algorithm and to study theoretical properties of *jewel*.

2.3. Numerical Algorithm

Function $F(\boldsymbol{\theta})$ in Equation (6) is convex and separable in terms of the groups. Moreover, we note that matrix $\mathbf{X}$, used in the formulation of Equation (6), satisfies the *orthogonal group hypothesis* that requires the restriction of $\mathbf{X}$ to the columns of each group to be orthogonal—indeed, $\mathbf{X}_{\cdot[ij]}$ is orthogonal by construction. Therefore, given λ, we can solve the minimization problem by applying the iterative group descent algorithm proposed in [20] which consists of updating one group of variables $i < j$ at a time freezing the other groups at their current value, cycling until convergence.

More precisely, given a starting value for vector $\boldsymbol{\theta}$, the *jewel* algorithm updates the group of variables $i < j$ minimizing function $F(\boldsymbol{\theta})$ for that group, considering the rest of the variables fixed to their current value. Consider the group $i < j$ and compute the subgradient of $F(\boldsymbol{\theta})$ with respect to the variables $\boldsymbol{\theta}_{[ij]}$. We have that the subgradient is a vector of $g_{ij} = 2K$ components defined as follows:

$$\frac{\partial F}{\partial \Theta^{(k)}_{ij}} = \begin{cases} -\frac{1}{n_k} {X^{(k)}_{\cdot i}}^T (X^{(k)}_{\cdot j} - \mathbf{X}^{(k)}\Theta^{(k)}_{\cdot j}) + \lambda \frac{\Theta^{(k)}_{ij}}{||\boldsymbol{\theta}_{[ij]}||} & \text{if} \quad ||\boldsymbol{\theta}_{[ij]}|| \neq 0 \\ -\frac{1}{n_k} {X^{(k)}_{\cdot i}}^T (X^{(k)}_{\cdot j} - \mathbf{X}^{(k)}\Theta^{(k)}_{\cdot j}) + \lambda u & \text{if} \quad ||\boldsymbol{\theta}_{[ij]}|| = 0 \end{cases} \tag{7}$$

where u is the relative entry of a vector $\mathbf{u} \in \mathbb{R}^{2K}$ with $||\mathbf{u}|| \leq 1$.

Now define vector $\mathbf{z} = (z_{ij}^{(1)}, z_{ji}^{(1)}, \ldots, z_{ij}^{(K)}, z_{ji}^{(K)})^T \in \mathbb{R}^{2K}$ with entries

$$\begin{aligned} z_{ij}^{(k)} &= -\frac{1}{n_k} X_{.i}^{(k)\,T} \left(X_{.j}^{(k)} - \sum_{m \neq i,j} X_{.m}^{(k)} \Theta_{mj}^{(k)} \right) \\ z_{ji}^{(k)} &= -\frac{1}{n_k} X_{.j}^{(k)\,T} \left(X_{.i}^{(k)} - \sum_{m \neq i,j} X_{.m}^{(k)} \Theta_{mi}^{(k)} \right). \end{aligned} \tag{8}$$

Vector $\mathbf{z}$ depends on the observed data $\mathbf{X}^{(k)}$, $k = 1 \ldots K$, and the current values of $\Theta_{ml}^{(k)}$ not involving the pairs of variables (i, j) we are seeking.

From the work in [20], we have that the minimizer of function $F(\boldsymbol{\theta})$ with respect to the variables $\boldsymbol{\theta}_{[ij]}$ is the following multivariate soft-thresholding operator

$$\begin{pmatrix} \hat{\Theta}_{ij}^{(1)} \\ \hat{\Theta}_{ji}^{(1)} \\ \vdots \\ \hat{\Theta}_{ij}^{(K)} \\ \hat{\Theta}_{ji}^{(K)} \end{pmatrix} = \left(1 - \frac{\lambda}{||\mathbf{z}||} \right)_+ \mathbf{z}. \tag{9}$$

The soft-thresholding operator $(\cdot)_+$ acts on vector $\mathbf{z}$ by shortening it towards 0 by an amount λ if its norm is greater or equal to λ and by putting it equal to zero if its norm is less than λ.

For a fixed value of λ, Equations (8) and (9) represent the updating step for each group $i < j$. The update is cyclically repeated for all groups. The entire cycle represents the generic update step of the iterative procedure. Thus, it is repeated until convergence. Precisely, we stop the procedure when the relative difference between two successive approximations of vector $\boldsymbol{\theta}$ is less than a given tolerance *tol*, or when the algorithm reaches a maximum number of iterations.

Remark 1. *The numerical algorithm can be easily extended to minimize the general model of Equation (3). When the data matrices include different number of variables p_k, then vector $\mathbf{z}$ has dimension g_{ij} that can be different for each pair $\{i, j\}$. Indeed, $\mathbf{z}$ and $\hat{\boldsymbol{\theta}}_{[ij]}$ incorporate only those k datasets for which the pairs of variables $\{i, j\}$ were observed. Equation (9) is still valid with $\lambda \cdot \sqrt{g_{ij}}$ in place of λ as each update step is done independently for each pair and $\sqrt{g_{ij}}$ is a constant that does not influence the convergence of the algorithm.*

Although *jewel*'s formal description involves large matrices and several matrix-vector products at each step, its implementation remains computationally feasible even for large datasets. Indeed, vectors $\mathbf{y}$, $\boldsymbol{\theta}$ and matrix $\mathbf{X}$ used in Equation (6) do not need to be explicitly built and the scalar products in Equation (8) can be obtained by modifying previously computed values.

Moreover, in our implementation of the group descend algorithm, we adopted the active shooting approach, as proposed in [9,21]. In this strategy, one exploits the problem's sparse nature efficiently, providing an increase in computational speed. The idea is to divide all pairs of variables into "active"— those which are not equal to zero at the current step— and "non-active"—those which are equal to zero at the current step—and update only the first ones. From a computational point of view, we define the upper triangular matrix **Active** of dimension $p \times p$, which takes trace of the current support of $\boldsymbol{\Theta}^{(k)}$, $k = 1 \ldots K$. **Active** can be initialized by setting all up extra-diagonal elements equal to 1, meaning that at the beginning of the procedure, all the groups $i < j$ are "active". Then, if the soft-thresholding operator zeroes group $\{2, 3\}$ during the iterations, the corresponding matrix element is set to zero, $Active_{23} = 0$, indicating that the group $\{2, 3\}$ is no longer

active and its status will be no more updated. At the end of the algorithm, matrix **Active** contains the estimate of the edge set E, as $(i,j) \in \hat{E} \iff Active_{ij} = 1$.

We provide the pseudocode for the algorithm we implemented for a fixed parameter λ. We also show how vector $\mathbf{z}$ can be efficiently updated using the residuals $\mathbf{R}^{(k)}$ of the linear regressions.

Note that at the end of each iteration, we evaluate the difference between two successive approximations with $\sum_k \left|\mathbf{\Theta}^{(k,t+1)} - \mathbf{\Theta}^{(k,t)}\right| / \sum_k \left|\mathbf{\Theta}^{(k,t)}\right| < tol$. In the simulation study we discover that *tol* has small influence on the final estimate of **Active**, thus we use $tol = 0.01$ to speed-up the calculations. However, as it might influence the evaluation of the residual error used to apply the BIC criterion, in Section 2.5 we set $tol = 10^{-4}$.

Remark 2. *Due to separability of the function $F(\boldsymbol{\theta})$ in Equation (6), if the graph structure is block-diagonal (i.e., the adjacency matrix encoding the graph is block-diagonal), then the minimization problem in Equation (6) can be solved independently for each block. In the case of ultrahigh-dimensional data, this strategy turns to be very computationally efficient since each block could be, in principle, solved in parallel. The work in [22] provides the conditions to split the original problem into independent subproblems.*

2.4. Theoretical Property

In this subsection, we establish the consistency property for the *jewel* estimator. Our findings are largely based on [23], where a GGM is inferred for temporal panel data. We start with formulation of the minimization problem given in Equation (6) in term of vector $\boldsymbol{\theta} = \left(\Theta_{21}^{(1)}, \ldots, \Theta_{p1}^{(1)}, \ldots, \Theta_{1p}^{(K)}, \ldots, \Theta_{(p-1)p}^{(K)}\right)^T$. Before presenting the main result in Theorem 1, let us introduce some auxiliary notations and Lemma 1, which will be useful in the proof of the theorem.

Let us denote $\boldsymbol{\theta^0}$ the true parameter vector of dimension $Kp(p-1) \times 1$. $\boldsymbol{\theta^0}$ is our unknown, and its non-zero components describe the true edge set E of the graph. $\boldsymbol{\theta}^0$ is naturally divided into $\binom{p}{2}$ groups, each consisting of the true parameters $\boldsymbol{\theta}^0_{[ij]}$ whose row/column index refer to the same pair $\{i,j\}$. Let s denote the true number of edges in E and define the sets of "active" and "non-active" groups as

$$S = \{(i,j) : i < j, \boldsymbol{\theta}^0_{[ij]} \not\equiv 0\} = \{(i,j) : i < j, (i,j) \in E\}$$
$$S^c = \{(i,j) : i < j, \boldsymbol{\theta}^0_{[ij]} \equiv 0\} = \{(i,j) : i < j, (i,j) \notin E\}$$

respectively, with $|S| = s$ and $|S^c| = \dfrac{p(p-1)}{2} - s = q$. Therefore, S contains all pairs of nodes for which there is an edge in E and S^c contains all pairs of nodes for which there is an absence of edge.

Now, referring to the linear regression problem formulation of *jewel* given in Equation (6), we define the additive Gaussian noise vector $\boldsymbol{\varepsilon}$ by the following:

$$\boldsymbol{\varepsilon} = (\underbrace{\varepsilon_1^{(1)^T}, \ldots, \varepsilon_p^{(1)^T}}_{\sim \mathcal{N}_{n_1 p}\left(0, \mathbf{\Lambda}^{(1)} \otimes \mathbf{I}_{n_1}\right)}, \ldots, \underbrace{\varepsilon_1^{(K)^T}, \ldots, \varepsilon_p^{(K)^T}}_{\sim \mathcal{N}_{n_K p}\left(0, \mathbf{\Lambda}^{(K)} \otimes \mathbf{I}_{n_K}\right)})^T, \quad \dim(\boldsymbol{\varepsilon}) = Np \times 1$$
$$\boldsymbol{\varepsilon} \sim \mathcal{N}_{Np}\left(0, blkdiag\left(\mathbf{\Lambda}^{(k)} \otimes \mathbf{I}_{n_k}\right)_{k=1\ldots K}\right),$$

with matrices

$$\mathbf{\Lambda}^{(k)} = diag\left(\frac{1}{\Omega_{11}^{(k)}}, \ldots, \frac{1}{\Omega_{pp}^{(k)}}\right).$$

Given these definitions, the data model can be rewritten as $\mathbf{y} = \mathbf{X}\boldsymbol{\theta}^0 + \boldsymbol{\varepsilon}$ and the following lemma holds:

Lemma 1 (Group Lasso estimate characterization, cfr. Lemma A.1 in [23]). *A vector $\hat{\boldsymbol{\theta}}$ is a solution to convex optimization problem in Equation (6) if and only if there exists $\boldsymbol{\tau} \in \mathbb{R}^{p(p-1)K}$ such that $[\mathbf{X}^T\mathbf{D}^2(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\theta}})] = \lambda\boldsymbol{\tau}$ and*

$$\boldsymbol{\tau}_{[ij]} = \begin{cases} dir\left(\hat{\boldsymbol{\theta}}_{[ij]}\right), & if \quad \hat{\boldsymbol{\theta}}_{[ij]} \not\equiv 0 \\ \mathbf{u}, \mathbf{u} \in \mathbb{R}^{2K}, ||\mathbf{u}|| \leq 1 & if \quad \hat{\boldsymbol{\theta}}_{[ij]} \equiv 0, \end{cases}$$

where $dir(\mathbf{u}) = \mathbf{u}/||\mathbf{u}||$ is the directional vector of any non-zero vector $\mathbf{u}$.

We can now state the main result, which has been inspired by Theorem 4.1 of [23]. In the following, the analog of empirical covariance matrix $\mathbf{C}$ and some auxiliary stochastic matrices and vectors which will be part of the main theorem:

$$\begin{aligned} \mathbf{C} &= \mathbf{X}^T\mathbf{D}^2\mathbf{X}, & \dim(\mathbf{C}) &= p(p-1)K \times p(p-1)K \\ \boldsymbol{\zeta} &= \mathbf{X}^T\mathbf{D}^2\boldsymbol{\varepsilon}, & \dim(\boldsymbol{\zeta}) &= p(p-1)K \times 1 \\ \mathbf{w} &= \boldsymbol{\zeta}_{S^c} - \mathbf{C}_{S^cS}\mathbf{C}_{SS}^{-1}\boldsymbol{\zeta}_S, & \dim(\mathbf{w}) &= 2Kq \times 1 \\ \mathbf{v} &= \mathbf{C}_{SS}^{-1}\boldsymbol{\zeta}_S, & \dim(\mathbf{v}) &= 2Ks \times 1, \end{aligned}$$

where $\boldsymbol{\zeta}_A$ and $\mathbf{C}_{AA}$ denote the restriction of vector $\boldsymbol{\zeta}$ and matrix $\mathbf{C}$ to the rows and columns in the set A.

Theorem 1. *Let $\hat{\boldsymbol{\theta}}$ be the solution of problem in Equation (6), with $\mathbf{y} = \mathbf{X}\boldsymbol{\theta}^0 + \boldsymbol{\varepsilon}$. Suppose that there exists $\delta > 0$ such that, with probability at least $1 - e^{-\delta\log(p)/N}$, one has*

1. *$\mathbf{C}_{SS}$ is invertible.*
2. *(Irrepresentable condition): $\exists\ \alpha \in (0,1) : \forall (i,j) \in S^c$*
   - *(a)* $\left|\left|[\mathbf{C}_{S^cS}\mathbf{C}_{SS}^{-1}\boldsymbol{\tau}]_{ij}\right|\right| \leq \alpha\ \forall\boldsymbol{\tau} \in \mathbb{R}^{2Ks} : \max\limits_{(i,j)\in S}||\boldsymbol{\tau}_{[ij]}||_2 \leq 1$
   - *(b)* $\lambda \geq \dfrac{2}{1-\alpha}||\mathbf{w}_{[ij]}||$
3. *(Signal strength): $\forall\ (i,j) \in S$ it holds*

$$\lambda < \left\{||\boldsymbol{\theta}^0_{[ij]}||_2 - ||\mathbf{v}_{[ij]}||\right\}\left|\left|\left[\mathbf{C}_{SS}^{-1}\boldsymbol{\tau}\right]_{[ij]}\right|\right|^{-1} \quad \forall\boldsymbol{\tau} \in \mathbb{R}^{2Ks} : \max_{(i,j)\in S}||\boldsymbol{\tau}_{[ij]}||_2 \leq 1$$

then, $\mathbb{P}(\hat{E} = E) \geq 1 - e^{-\delta\log(p)/N}$, where
$E = \{(i,j) : \boldsymbol{\theta}^0_{[ij]} \not\equiv 0\}$ is the true edge set and
$\hat{E} = \{(i,j) : \hat{\boldsymbol{\theta}}_{[ij]} \not\equiv 0\}$ is the estimated edge set.

Proof. To prove set equality $\hat{E} = E$, we verify separately the two inclusions, $\hat{E} \subseteq E$ and $\hat{E} \supseteq E$. Let us first prove inclusion $\hat{E} \subseteq E \iff \hat{\boldsymbol{\theta}}_{[ij]} \equiv 0 \quad \forall (i,j) \in S^c$.

Define $\hat{\boldsymbol{\theta}}^S$ be the solution of the following restricted problem:

$$\hat{\boldsymbol{\theta}}^S := \underset{\boldsymbol{\theta}\in\mathbb{R}^{2Ks}}{\arg\min}\frac{1}{2}||\mathbf{y} - \mathbf{X}_{\cdot S}\boldsymbol{\theta}||^2_{\mathbf{D}^2} + \lambda \sum_{(i,j)\in S}||\boldsymbol{\theta}_{[ij]}||.$$

By Lemma 1, $\exists\ \boldsymbol{\tau}^S \in \mathbb{R}^{2Ks}$ such that $-\mathbf{X}_{\cdot S}^T\mathbf{D}^2(\mathbf{y} - \mathbf{X}_{\cdot S}\hat{\boldsymbol{\theta}}^S) + \lambda\boldsymbol{\tau}^S = 0$ and

$$\boldsymbol{\tau}^S_{[ij]} = \begin{cases} \hat{\boldsymbol{\theta}}^S_{[ij]}/||\hat{\boldsymbol{\theta}}^S_{[ij]}||, & if \quad \hat{\boldsymbol{\theta}}^S_{[ij]} \not\equiv 0 \\ \mathbf{u} \in \mathbb{R}^{2K}, ||\mathbf{u}|| \leq 1, & if \quad \hat{\boldsymbol{\theta}}^S_{[ij]} \equiv 0. \end{cases}$$

Define $\hat{\boldsymbol{\theta}} \in \mathbb{R}^{p(p-1)K}$ such that its restriction to the set of active groups coincides with $\hat{\boldsymbol{\theta}}^S$, while its restriction to the set of non-active groups is zero, i.e.,

$$\hat{\boldsymbol{\theta}}_{[ij]} = \begin{cases} \hat{\boldsymbol{\theta}}^S_{[ij]}, & if \quad (i,j) \in S \\ 0 & if \quad (i,j) \in S^c. \end{cases}$$

To get the first inclusion, we need to prove that $\hat{\boldsymbol{\theta}}$ is a solution of the full problem in Equation (6). By Lemma 1 it is sufficient to prove that $\exists\, \boldsymbol{\tau} \in \mathbb{R}^{p(p-1)K} : -\mathbf{X}^T\mathbf{D}^2(Y - \mathbf{X}\hat{\boldsymbol{\theta}}) + \lambda\boldsymbol{\tau} = 0$ and

$$\boldsymbol{\tau}_{[ij]} = \begin{cases} \hat{\boldsymbol{\theta}}_{[ij]}/||\hat{\boldsymbol{\theta}}_{[ij]}||, & if \quad \hat{\boldsymbol{\theta}}_{[ij]} \not\equiv 0 \\ \mathbf{u} \in \mathbb{R}^{2K}, ||\mathbf{u}|| \leq 1, & if \quad \hat{\boldsymbol{\theta}}_{[ij]} \equiv 0. \end{cases} \tag{10}$$

When $\hat{\boldsymbol{\theta}}_{[ij]} \not\equiv 0$, the conditions in Equation (10) are satisfied by construction of $\hat{\boldsymbol{\theta}}$. When $\hat{\boldsymbol{\theta}}_{[ij]} \equiv 0$, the conditions in Equation (10) need to be verified. To this aim, substitute $\mathbf{y} = \mathbf{X}\boldsymbol{\theta}^0 + \boldsymbol{\varepsilon}$ into $-\mathbf{X}^T\mathbf{D}^2(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\theta}}) + \lambda\boldsymbol{\tau} = 0$ and get

$$\begin{aligned} -\mathbf{X}^T\mathbf{D}^2(\mathbf{X}\boldsymbol{\theta}^0 + \boldsymbol{\varepsilon} - \mathbf{X}\hat{\boldsymbol{\theta}}) + \lambda\boldsymbol{\tau} &= 0 \\ -\mathbf{X}^T\mathbf{D}^2\mathbf{X}\boldsymbol{\theta}^0 - \mathbf{X}^T\mathbf{D}^2\boldsymbol{\varepsilon} + \mathbf{X}\hat{\boldsymbol{\theta}} + \lambda\boldsymbol{\tau} &= 0 \\ \underbrace{\mathbf{X}^T\mathbf{D}^2\mathbf{X}}_{\mathbf{C} \text{ by def}}(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^0) - \underbrace{\mathbf{X}^T\mathbf{D}^2\boldsymbol{\varepsilon}}_{\boldsymbol{\zeta} \text{ by def}} + \lambda\boldsymbol{\tau} &= 0 \\ \mathbf{C}(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^0) - \boldsymbol{\zeta} + \lambda\boldsymbol{\tau} &= 0 \end{aligned} \tag{11}$$

After properly permuting the indexes of $\mathbf{C}$, $\boldsymbol{\zeta}$ and $\boldsymbol{\tau}$, i.e., placing all the variables belonging to the active groups at the beginning and the non-active ones at the end, Equation (11) becomes

$$\begin{pmatrix} \mathbf{C}_{SS} & \mathbf{C}_{SS^c} \\ \mathbf{C}_{S^cS} & \mathbf{C}_{S^cS^c} \end{pmatrix} \begin{pmatrix} \hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^0 \\ 0 \end{pmatrix} - \begin{pmatrix} \boldsymbol{\zeta}_S \\ \boldsymbol{\zeta}_{S^c} \end{pmatrix} + \lambda \begin{pmatrix} \boldsymbol{\tau}_S \\ \boldsymbol{\tau}_{S^c} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

This is equivalent to

$$\begin{cases} \mathbf{C}_{SS}(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^0) - \boldsymbol{\zeta}_S + \lambda\boldsymbol{\tau}_S &= 0 \\ \mathbf{C}_{S^cS}(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^0) - \boldsymbol{\zeta}_{S^c} + \lambda\boldsymbol{\tau}_{S^c} &= 0 \end{cases}$$

Solving the first equation for $\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^0$, and substituting into the second, we obtain

$$\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^0 = \mathbf{C}_{SS}^{-1}(\boldsymbol{\zeta}_S - \lambda\boldsymbol{\tau}_S) \tag{12}$$

and then

$$\begin{aligned} \mathbf{C}_{S^cS}\mathbf{C}_{SS}^{-1}(\boldsymbol{\zeta}_S - \lambda\boldsymbol{\tau}_S) - \boldsymbol{\zeta}_{S^c} + \lambda\boldsymbol{\tau}_{S^c} &= 0 \\ \boldsymbol{\tau}_{S^c} &= -\frac{1}{\lambda}\mathbf{C}_{S^cS}\mathbf{C}_{SS}^{-1}(\boldsymbol{\zeta}_S - \lambda\boldsymbol{\tau}_S) + \frac{\boldsymbol{\zeta}_{S^c}}{\lambda} \\ \boldsymbol{\tau}_{S^c} &= \frac{1}{\lambda}\underbrace{(\boldsymbol{\zeta}_{S^c} - \mathbf{C}_{S^cS}\mathbf{C}_{SS}^{-1}\boldsymbol{\zeta}_S)}_{\mathbf{w}} + \mathbf{C}_{S^cS}\mathbf{C}_{SS}^{-1}\boldsymbol{\tau}_S, \end{aligned}$$

By using hypothesis 2, we get $\forall (i,j) \in S^c$

$$||\boldsymbol{\tau}^{S^c}_{[ij]}|| \leq \frac{1}{\lambda}||\mathbf{w}_{[ij]}|| + ||\left[\mathbf{C}_{S^cS}\mathbf{C}_{SS}^{-1}\boldsymbol{\tau}_S\right]_{[ij]}|| < \frac{\alpha + 1}{2} < 1.$$

The second inclusion requires $\hat{E} \supseteq E \iff \hat{\boldsymbol{\theta}}_{[ij]} \not\equiv 0 \; \forall (i,j) \in S$. We observe that it is implied by $||\hat{\boldsymbol{\theta}}_{[ij]} - \boldsymbol{\theta}^0_{[ij]}|| < ||\boldsymbol{\theta}^0_{[ij]}|| \; \forall \; (i,j) \in S$ which is a stronger requirement called *direction consistency* in the original paper [23]. Starting from Equation (12), we get

$$\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^0 = \underbrace{\mathbf{C}_{SS}^{-1}(\boldsymbol{\zeta}_S}_{\mathbf{v}} - \lambda \boldsymbol{\tau}_S) = \mathbf{v} - \lambda \mathbf{C}_{SS}^{-1} \boldsymbol{\tau}_S.$$

Then, by hypothesis 3, we have that $\forall (i,j) \in S$

$$||\hat{\boldsymbol{\theta}}_{[ij]} - \boldsymbol{\theta}^0_{[ij]}|| \leq ||\mathbf{v}_{[ij]}|| + \lambda ||\left[\mathbf{C}_{SS}^{-1} \boldsymbol{\tau}_S\right]_{[ij]}|| < ||\boldsymbol{\theta}^0_{[ij]}||.$$

□

Remark 3. *We stress that the hypotheses of Theorem 1 are weaker than the hypotheses of Theorem 4.1 in [23]. In fact, in our setting, the stochastic matrix* $\mathbf{C}$ *and vector* $\boldsymbol{\zeta}$ *do not inherit the Gaussian distribution from data. Therefore, our results are based on a probabilistic assumption on these stochastic objects and not on the underlying families of Gaussian distributions, i.e., on their covariance matrices* $\boldsymbol{\Sigma}^{(k)}$, $k = 1 \dots K$. *However, if, on one hand, this could be a limitation, on the other hand, our result gives explicit conditions on the data that, in principle, could be verified given an estimate of vector* $\boldsymbol{\theta}$. *The same would not be possible when the assumptions involve the population matrices* $\boldsymbol{\Sigma}^{(k)}$ *instead of the population vector* $\boldsymbol{\theta}^0$, *because we do not estimate covariance matrices.*

Remark 4. *In machine learning language, hypothesis 2 implies that* λ *must be chosen small enough to control the Type I error (i.e., the first inclusion) to avoid killing real edges. Hypothesis 3 implies that* λ *must be chosen large enough to control the Type II error (the second inclusion) to avoid including in the model false edges. Unfortunately, as it always happens in literature, from theoretical results we have no explicit expression for* λ, *thus we will select it through data-driven criteria, as exposed in the next section.*

2.5. Selection of Regularization Parameter

Like any other penalty-based method, *jewel* requires selecting the regularization parameter λ, which controls the resulting estimator's sparsity. A high value of λ results in a more sparse and interpretable estimator, but it may have many false-negative edges. By contrast, a small value of λ results in a less sparse estimator with many false-positive edges.

Some authors have proposed using $\lambda = \sqrt{\log p / n}$ or suggested empirical application-driven choices so that the resulting model is sufficiently complex to provide novel information and, at the same time, sufficiently sparse to be interpretable. However, the best choice remains to select λ by Bayesian Information Criterion (BIC), Cross-Validation (CV), or other data-driven criteria (e.g., quantile universal threshold (QUT) [24]). In this work, we propose the use of BIC and CV approaches.

Bayesian Information Criterion (BIC): Following the idea in [21], we can define the BIC for the K classes as the weighted sum of the BICs of the individual classes. For each class, the BIC comprises two terms: the logarithm of the residual sum of squares (RSS) and the degree of freedom. For any value of λ, Algorithm 1 provides not only the solution $\hat{\boldsymbol{\theta}}$, but also the RSS stored in the matrices $\mathbf{R}^{(k)}$ and the degree of freedom as the number of non-zero pairs in the **Active** matrix. Therefore, the expression for BIC is given by

$$BIC(\lambda) = \sum_{k=1}^{K} n_k \sum_{i=1}^{p} \log \left|\left| R^{(k)}_{.i}(\lambda) \right|\right|^2 + \#\{Active_{ij}(\lambda) \neq 0\} \sum_{k=1}^{K} \log n_k.$$

Algorithm 1 The *jewel* algorithm

INPUT: $\mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(K)}$, λ, tol and $t_{\max}$

INITIALIZE:
$\mathbf{\Theta}^{(1,0)}, \ldots, \mathbf{\Theta}^{(K,0)}$
$\mathbf{R}^{(k)} = \mathbf{X}^{(k)} - \mathbf{X}^{(k)}\mathbf{\Theta}^{(k,0)}, k = 1 \ldots K$

$$\mathbf{Active} = \begin{pmatrix} 0 & 1 & \ldots & 1 \\ 0 & 0 & & 1 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \ldots & 0 \end{pmatrix}$$

REPEAT UNTIL CONVERGENCE
for $j = 1 \ldots p$ **do**
  for $i = j+1 \ldots p$: **do**
    if $Active_{ij} \neq 0$
    evaluate $\mathbf{z} = \left(z_{ij}^{(1)}, z_{ji}^{(1)}, \ldots, z_{ij}^{(K)}, z_{ji}^{(K)}\right)$ by

$$z_{ij}^{(k)} = \frac{1}{n_k} {X_{.i}^{(k)}}^T R_{.j}^{(k)} + \Theta_{ij}^{(k,t)}$$
$$z_{ji}^{(k)} = \frac{1}{n_k} {X_{.j}^{(k)}}^T R_{.i}^{(k)} + \Theta_{ji}^{(k,t)}$$

    evaluate $threshold = 1 - \lambda / \|\mathbf{z}\|$
    if $threshold < 0$ **then**
      $Active_{ij} \leftarrow 0$ and $\mathbf{z} \leftarrow 0$
    else
      $\mathbf{z} \leftarrow \mathbf{z} \cdot threshold$
    end if
    update residuals

$$R_{.j}^{(k)} = R_{.j}^{(k)} + X_{.i}^{(k)}\left(\Theta_{ij}^{(k,t)} - z_{ij}^{(k)}\right)$$
$$R_{.i}^{(k)} = R_{.i}^{(k)} + X_{.j}^{(k)}\left(\Theta_{ji}^{(k,t)} - z_{ji}^{(k)}\right)$$

    update coefficients $(\Theta_{ij}^{(1,t)}, \Theta_{ji}^{(1,t)}, \ldots, \Theta_{ij}^{(K,t)}, \Theta_{ij}^{(K,t)}) \leftarrow \mathbf{z}$

  end for
end for
Stop if $\dfrac{\sum_k \left|\mathbf{\Theta}^{(k,t+1)} - \mathbf{\Theta}^{(k,t)}\right|}{\sum_k \left|\mathbf{\Theta}^{(k,t)}\right|} < tol$ or $t > t_{\max}$
OUTPUT: **Active**

Given a grid of parameters $\lambda_1 < \lambda_2 \cdots < \lambda_L$, we choose $\lambda_{BIC} = \underset{\lambda_l, l=1\ldots L}{\arg\min} BIC(\lambda_l)$.

Cross-Validation (CV): The idea of cross-validation (CV) is to split the data into F folds and consequentially use one fold as a testing set and all the others as training set. In our *jewel* procedure, we divide each data set $\mathbf{X}^{(k)}$ into F folds of dimension $n_k^f \times p$ and the f-th fold is the union of the f-th folds of each class. As in standard CV procedure, for each parameter λ_l of the grid $\lambda_1 < \cdots < \lambda_L$ and for each fold $(\mathbf{X}_f^{(k)})_{k=1,\ldots,K}$ we estimate $\hat{\mathbf{\Theta}}_{-f}^{(k)}(\lambda_l)$ and then evaluate its error as

$$err(f,l) = \sum_{k=1}^{K} \frac{1}{n_k^f} \left|\left|\mathbf{X}_f^{(k)} - \mathbf{X}_f^{(k)} \hat{\mathbf{\Theta}}_{-f}^{(k)}(\lambda_l)\right|\right|_F^2.$$

Errors are then averaged over folds $CV(\lambda_l) = \frac{1}{F}\sum_{f=1}^{F} err(f,l)$ and the optimal parameter is chosen as $\lambda_{CV} = \underset{\lambda_l, l=1\ldots L}{\arg\min} CV(\lambda_l)$.

As part of our numerical procedure, for both criteria, we start from the same grid of values $\lambda_1 < \cdots < \lambda_L$, and we adopt the warm start initialization procedure combined with the active shooting approach. Warm start initialization procedure means that we first apply Algorithm 1 with the smaller value of λ_l, obtaining solution $\hat{\boldsymbol{\theta}}_{\lambda_l}$ and $\mathbf{Active}_{\lambda_l}$. Then, when moving to the next value λ_{l+1}, we initialize Algorithm 1 with $\mathbf{Active}_{\lambda_l}$ and $\hat{\boldsymbol{\theta}}_{\lambda_l}$. Starting with a sparse **Active** matrix instead of a full one, allows the algorithm to cycle over a smaller number of groups, accelerating the iteration step. Note that with warm start initialization not only we reduce the computational cost but also get nested solutions.

2.6. Code Availability

jewel is implemented as an R package `jewel` which is freely available at https://github.com/annaplaksienko/jewel accessed on 11 July 2021.

3. Results

3.1. Simulation Studies

This section presents simulation results to demonstrate the empirical performance of *jewel* from different perspectives. Specifically, we conducted three types of experiments. In the first, we show the performance of *jewel* as a function of the number of classes K, assessing the advantages of using more than one dataset. In the second, given the same K datasets, we show that the joint approach is better than performing K independent analyses with a voting strategy or fitting a single concatenated dataset. Finally, in the third experiment, we compare the performance of *jewel* with two existing methods, the joint graphical lasso (JGL) [14] and the proposal of Guo et al. [13]. We used `JGL` package for the first and the code, kindly provided by the authors of [13], for the second.

Before presenting the results, we briefly describe the data generation and the metrics we use to measure performance.

Data generation: Given K, p, and n_k, we generated a "true" scale-free network $G = (V, E)$ with p nodes according to the Barabasi–Albert algorithm with the help of `igraph` package [25]. If not stated otherwise, the number of edges added at each step of the algorithm, m, and the power of the preferential attachment, *power*, were both set to 1. The resulting graph G was sparse and had $mp - (2m - 1)$ edges. Then, we generated K precision matrices $\boldsymbol{\Omega}^{(k)}$. To this purpose, for each k, we created a $p \times p$ matrix with 0 s on the elements not corresponding to the network edges and symmetric values sampled from the uniform distribution with support on $[-0.8, -0.2] \cup [0.2, 0.8]$ for the elements corresponding to the existing edges. To ensure positive definiteness of $\boldsymbol{\Omega}^{(k)}$, we set its diagonal elements equal to $|\mu_{min}(\boldsymbol{\Omega}^{(k)})| + 0.1$, with $\mu_{min}(\mathbf{A})$ being the minimum eigenvalue of matrix $\mathbf{A}$. We invert $\boldsymbol{\Omega}^{(k)}$ and set $\boldsymbol{\Sigma}^{(k)}$ with elements

$$\Sigma_{ij}^{(k)} = \frac{\left(\Omega^{(k)}\right)_{ij}^{-1}}{\sqrt{\left(\Omega^{(k)}\right)_{ii}^{-1}\left(\Omega^{(k)}\right)_{jj}^{-1}}}.$$

Finally, for each k, we sampled n_k independent, identically distributed observations from $\mathcal{N}(0, \boldsymbol{\Sigma}^{(k)})$.

Performance measures: We evaluated the estimate of the graph structure $\hat{E}$ using the *true positive rate* and the *false positive rate*, defined, respectively, as

$$TPR = \frac{TP}{TP + FN} \qquad \text{and} \qquad FPR = \frac{FP}{FP + TN},$$

with

$$TP = |\{(i,j) : (i,j) \in E \cap \hat{E}\}|, \qquad TN = |\{(i,j) : (i,j) \in E^c \cap \hat{E}^c\}|,$$
$$FP = |\{(i,j) : (i,j) \in E^c \cap \hat{E}\}|, \qquad TN = |\{(i,j) : (i,j) \in E \cap \hat{E}^c\}|,$$

where A^c is the complement of set A. TPR shows the proportion of edges correctly identified, and FPR shows the proportion of edges incorrectly identified. As usually done in the literature, to judge the method's performance without being influenced by λ, we used the ROC-curve (receiver operating characteristic), i.e., TPR against the FPR for different values of λ. Our experiments used a grid of λ_s equispaced in log scale, consisting of 50 values ranging from 0.01 to 1. We averaged both performance metrics and running time over 20 independent realizations of the above data generation procedure. Running time was measured on the 4-core 3.6 GHz processor and 16 GB RAM computer.

3.1.1. More Datasets Provide Better Performance

This first experiment aims to quantify the gain in estimating E when the number of datasets K increases. The simulation settings for this first experiment are as follows. We simulated 10 datasets as described above for two different dimensional cases: $p = 100$ with $n_k = 50 \; \forall k$ ($n_k/p = 1/2$) and $p = 500$ with $n_k = 100 \; \forall k$ ($n_k/p = 1/5$). We repeated the datasets generation 20 times. For each case, we first applied *jewel* to the $K = 10$ datasets and each of the 20 runs. We computed the average TPR and FPR. Then, we sampled $K = 5$ matrices (in each run) and repeated the procedure. We subsampled $K = 3$ matrices out of the previous 5, then $K = 2$ out of 3 and $K = 1$ out of 2. In other words, for each value of $K = 10, 5, 3, 2, 1$ we applied *jewel* to 20 realizations and evaluated the average TPR and FPR.

The average ROC curve in Figure 1 illustrates the performances as a function of K. Results in Figure 1 show the trend of improvement as K grows (which we expect, given the increasing amount of available data) and demonstrate that a limited increase in the number of datasets can provide a significant gain in performance. Indeed, we observed a remarkable improvement going from $K = 1$ to $K = 2$ or $K = 3$. Of course, this improvement comes at a price on an increasing computational time. However, this price is not excessive because it increases from ≈40 min for $K = 1$ to ≈1.5 h for $K = 3$ considering the whole grid of 50 λ parameters. The grid of λ is uniform in log-scale and starts from 0.01. Therefore, half of the values are between 0.01 and 0.1. Starting from a bigger λ_1 or using fewer values would decrease running time to minutes and make the price in terms of computational cost not excessive. Note also that these running times refers to the case where we use *jewel* over the entire grid of λ without the warm start procedure.

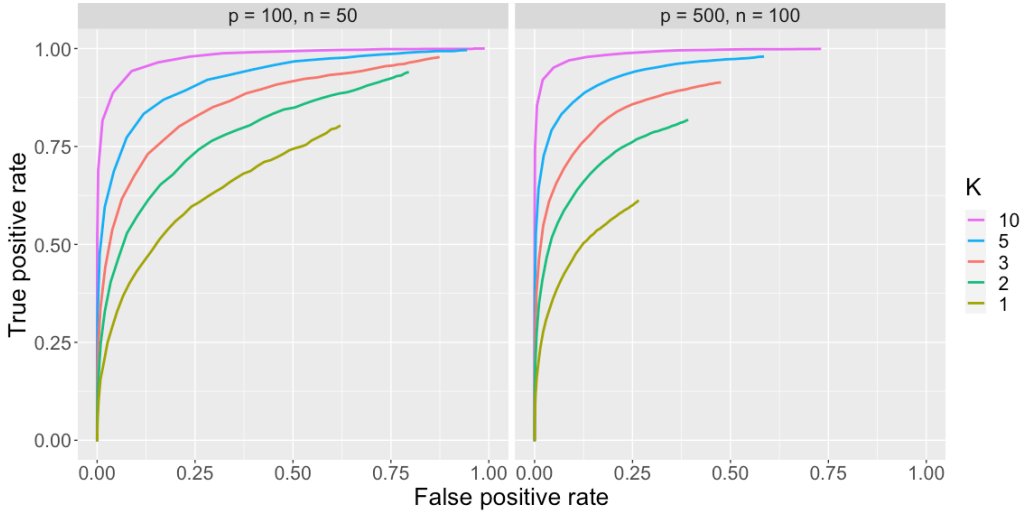

Figure 1. ROC-curve for *jewel* method applied to the different number of datasets K (denoted by different colors). **Left panel**: performance for $p = 100, n_k = 50 \; \forall k$. **Right panel**: performance for $p = 500, n_k = 100 \; \forall k$.

3.1.2. The Joint Approach Is Better That Voting and Concatenation

In the same spirit of [26], this second experiment shows the importance of considering a joint approach when analyzing multiple datasets instead of other naive alternatives.

More precisely, given K datasets sharing the same network structure E, we want to show that the joint analysis performed by *jewel* has important advantages with respect to the two following alternatives. The first is the *concatenation* approach, where all data sets are combined into one extended matrix of size $\sum_k n_\times p$ and *jewel* is applied with $K = 1$. The second is the *voting* approach, where each dataset is processed independently by *jewel* with $K = 1$ obtaining K estimates of the adjacency matrices. Then, we build a consensus matrix by setting an edge if it is present in at least $\lceil K/2 \rceil$ of the estimated matrices. Figure 2 illustrates a schematic representation of these approaches.

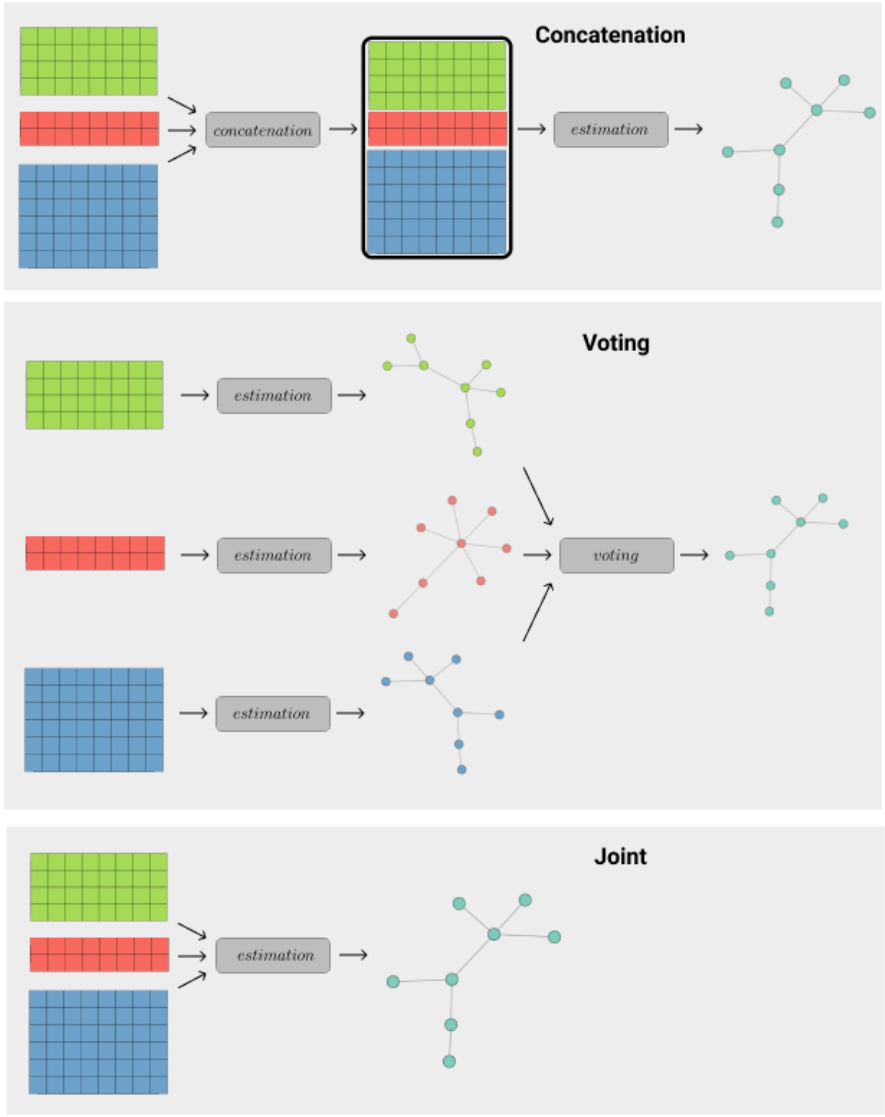

Figure 2. Different approaches that can be used for analyzing multiple datasets. **Top panel:** concatenation. **Middle panel:** voting. **Bottom panel:** joint approach.

The simulation setting for this second experiment is the following. We generated 20 independent runs, each with $K = 3$ data sets. We considered two dimensional scenario, $p = 100$ with $n_k = 50$ and $p = 500$ with $n_k = 100$. For the concatenation approach, as a first step, we constructed the long $Kn_k \times p$ matrix and then applied *jewel*. For the voting approach, we applied the method separately to each data matrix $\mathbf{X}^{(k)}$, $k = 1 \ldots 3$, and then

put an edge in the resulting adjacency matrix only if it was present in 2 out of 3 estimated adjacency matrices. The ROC curves represent each approach's performance in the left and right panel of Figure 3 for the two-dimensional scenario, respectively.

First, we note that performance in the first scenario (left panel) is superior to the second scenario (right panel) due to the high-dimensional regime that is more severe in the second case. Second, we observe a significant advantage in processing the datasets jointly with respect to the other two approaches. Indeed, with the same amount of data, *jewel* correctly exploits the commonalities and the differences in K distributions and provides a more accurate estimate of E. Instead, the concatenation approach ignores the distributional differences creating a single data matrix (from not identically distributed datasets), and the voting approach exploits the common structure of the network only during the post-processing (voting) of the estimator. As a consequence, both concatenation and voting approaches result in a loss of performance.

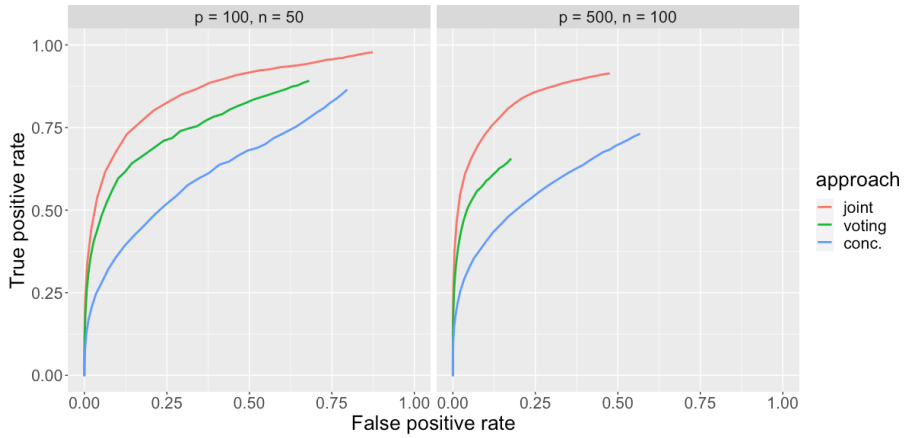

Figure 3. ROC curve for different approaches of inferring the graph from $K = 3$ datasets: joint estimation, voting and concatenation (denoted in different colors). **Left panel**: performance for $p = 100, n_k = 50\ \forall k$. **Right panel**: performance for $p = 500, n_k = 100\ \forall k$.

3.1.3. Comparison of *Jewel* with Other Joint Estimation Methods

In this third experiment, we compare the performance and runtime of *jewel* with two other methods for joint estimation: joint graphical lasso (JGL) with group penalty [14] and the proposal of Guo et al. [13]. JGL requires two tuning parameters λ_1 and λ_2 where the first one is responsible for differences in the supports of $\mathbf{\Theta}^{(k)}$, $k = 1 \dots K$. Thus, according to our hypothesis, as the patterns of non-zero elements are identical across classes, we set $\lambda_1 = 0$ and vary only λ_2. For the proposal of Guo et al., we consider the union of the supports of $\mathbf{\Theta}^{(k)}$ as the final adjacency matrix estimation (OR rule).

For sake of brevity, in this section we discuss only the dimensional setting $K = 3$, $p = 500$, $n_k = 100\ \forall k$ ($n/p = 1/5$). The other settings show analogous results and do not add value to our exposition. Instead, as an added value to this study, we explored the influence of the type of "true" graph on methods' performance. Specifically, we compared results obtained for different scale-free graphs obtained changing parameters m and *power*. The first one, m, controls the graph's sparsity as the number of edges in the resulting graph is equal to $mp - (2m - 1)$. The *power* parameter controls graph's hub structure—bigger *power*, bigger hubs. We considered six different $m - power$ scenarios, with parameter $m = 1, 2$ (resulting in 499 edges with 0.4% sparsity and 997 edges with 0.8% sparsity, respectively) and parameter $power = 0.5, 1, 1.5$. In each of these scenarios, we generated the "true" underlying graph for 20 independent realizations, see Figure 4 for a random realization in each scenario. We then proceeded with the same scheme described before, generating the data, to which we applied *jewel*, JGL, and Guo et al. methods, finally evaluating the average performance and running time.

In Figure 5, we show results of this third experiment and observe that on average *jewel* and JGL are comparable in performance, both being superior to the proposal of Guo et al. This observation remains true even in the worst-case scenario, i.e., *power* = 1.5. Overall, Figure 5 illustrates the good performance of this class of methods in the sparse regime, although increasing m, the performance decreases (in the worst case, it becomes similar to a random guess for all methods). More specifically, we note that increasing *power*, i.e., hubs size, leads to a significant loss in performance for all the methods. This observation agrees with the recent paper [27] for the case of one dataset that explores classical methods, like the one treated in this paper, for inferring a network with big hubs and comes to the same discovery. This observation is quite important since, in many real-world networks, the *power* is often estimated between 2 and 3. We could introduce degree-induced weights into the penalty to overcome this limitation, but this possibility is not explored in this paper.

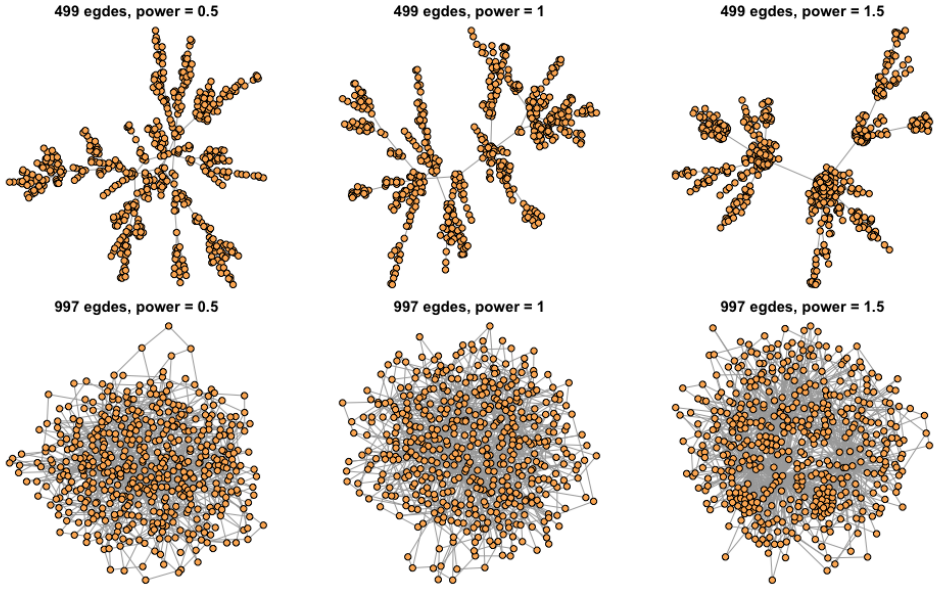

Figure 4. Scale-free graphs with $p = 500$ nodes generated with different values of parameters m (in rows) and *power* (in columns). The graphs correspond to one of the 20 random realizations generated in this simulation setup.

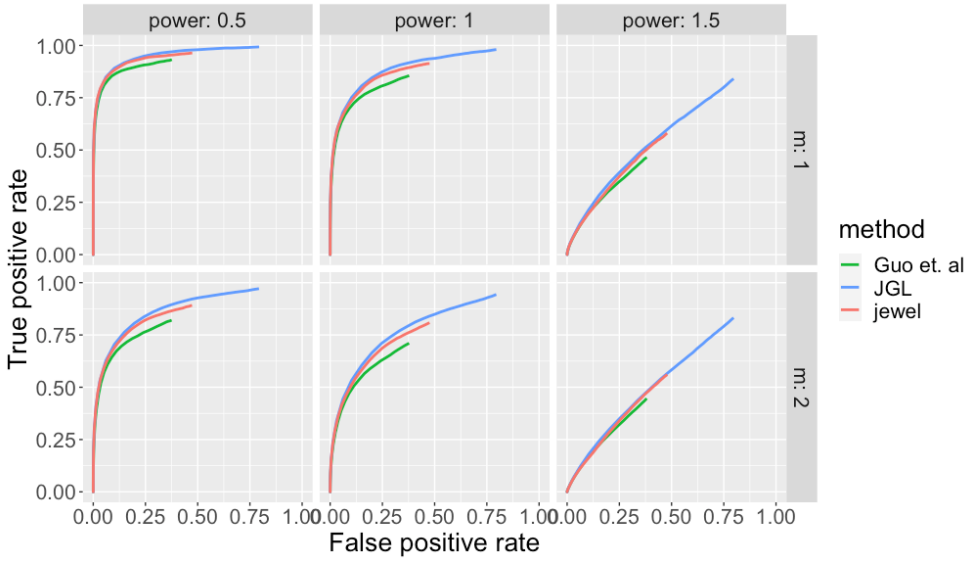

Figure 5. ROC-curve for different joint estimation methods: *jewel*, JGL [14] and Guo et al. proposal [13] for $K = 3$, $p = 500, n_k = 100\ \forall k$. Each panel demonstrates performance in different $m - power$ setting.

In Table 1, we report the running time results for some specific values of λ and for the entire grid of λ in the case $m = 1$, $power = 1$ (we omit other cases as these parameters do not influence the running time). As the tolerance influences the running time, we report that for each method we used its default stopping criteria threshold value which is $tol = 0.01$ for both *jewel* and Guo et al. method and $tol = 10^{-4}$ for JGL. As we can see, without using the warm start, *jewel* is approximately two times faster than JGL and several times faster than Guo et al. for small values of λ. This does not hold for the higher values of λ where *jewel* has to pay the price for the full initialization of the **Active** matrix. However, in real data applications, it is unlikely to use such large values of λ since it implies setting most of the connections to zero and, hence, many false negatives. In practical applications, interesting values for λ lay in the range for which *jewel* is faster than both its competitors.

Table 1. Running time for different joint estimation methods: *jewel*, JGL [14], and Guo et al.'s proposal [13] for $K = 3$, $p = 500$, $n_k = 100$, $m = 1$ and $power = 1$ over the uniform in log-scale grid of 50 parameters λ from 0.01 to 1.

	jewel	**JGL**	**Guo et al.**
$\lambda = 0.01$	≈7 min	≈11.5 min	≈66 min
$\lambda = 0.1$	41.16 s	80.16 s	≈2.6 min
$\lambda = 0.2$	26.28 s	73.76 s	73.68 s
$\lambda = 0.52$	26.32 s	0.31 s	22.91 s
$\lambda = 0.83$	26.3 s	0.099 s	12.08 s
$\lambda = 1$	22.65 s	0.099 s	10.88 s
grid of 50 λ	≈1.5 h	≈3.4 h	≈8.4 h

To summarize, we can assert that *jewel* demonstrated performance comparable to JGL and superior to Guo et al.'s proposal while showing a significant advantage in terms of running time in respect to both methods.

3.1.4. Tuning Parameter Estimation

Here, we show results obtained using BIC and CV criteria described in Section 2.5. `jewel` package has both criteria built-in. By default, we fixed 5-folds for the CV and implemented parallelization on a 4-core machine. Warm start procedure was implemented for both criteria.

The simulation setting is the following: for $p = 500$, $n_k = 100$, $K = 3$ we generated 20 independent runs as described before with default values $m = 1$ and $power = 1$. We used a grid of 50 λs uniformly spaced in log-scale from 0.1 to 1. We set the stopping criterion threshold $tol = 10^{-4}$ instead of default value $tol = 10^{-2}$ to achieve higher accuracy for the estimation of regression coefficients $\hat{\mathbf{\Theta}}^{(k)}$ and residuals $\mathbf{R}^{(k)}$, $k = 1 \dots K$, which are required by both criteria BIC and CV.

In Figure 6, we plot values of BIC and CV error for each λ_l value for one realization of data randomly chosen out of twenty independent runs. In Table 2, instead, we report results averaged over all 20 runs. For each run, we first estimated λ_{BIC} and λ_{CV} by the two criteria, then ran *jewel* with these values and evaluated performance in terms of accuracy, precision, recall ($accuracy = (TP + TN)/(TP + TN + FP + FN)$, $precision = TP/(TP + FP)$, $recall = TP/(TP + FN)$) and running time.

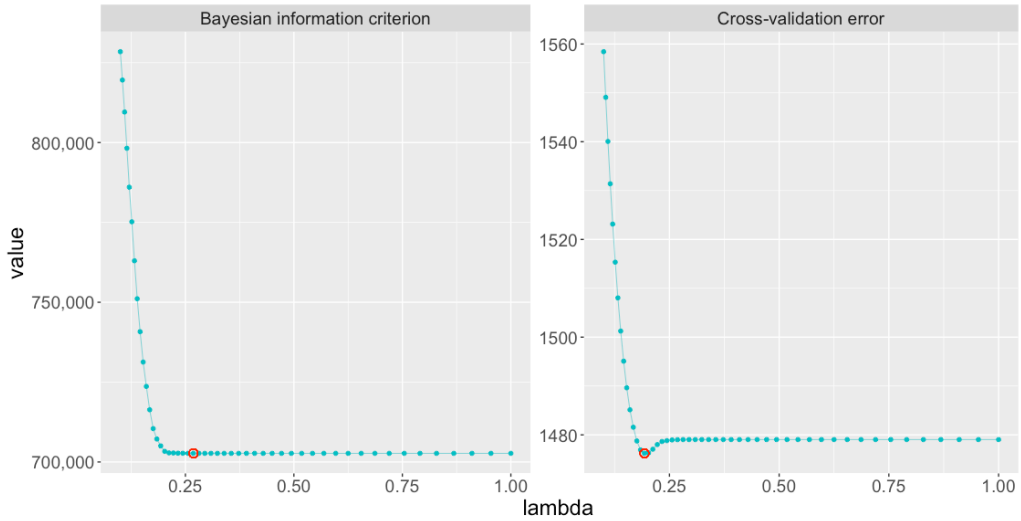

Figure 6. Left panel: values of BIC obtained with warm start for *jewel*. **Right panel:** CV error obtained with warm start. Results are reported for one randomly chosen realization with $K = 3$, $p = 500$, $n_k = 100$, $m = 1$, $power = 1$. Red circles denote the estimated optimal λ_{OPT}.

Table 2. Results of BIC and CV procedures with and without warm start for *jewel* in the case $K = 3$, $p = 500$, $n_k = 100$, $m = 1$, $power = 1$. Performance metrics and runtime were evaluated with estimated λ_{OPT}.

	λ_{OPT}	Accuracy	Precision	Recall	Runtime
BIC with w.s.	0.268	0.996	0.548	0.134	≈3.6 min
CV with w.s.	0.193	0.992	0.229	0.382	≈3.5 min

As we can see from Figure 6, both curves show a clear and well-defined minimum and provide λ_{OPT} estimation. From Table 2 we also observe that runtime is around ≈3.5 min, while the estimation without warm start would take around 3.5 h. Thus, we suggest using warm start initialization for both criteria.

On average both estimates λ_{BIC} and λ_{CV} are quite close to $\lambda = \sqrt{\log p/n} = \sqrt{\log 500/100} = 0.249$,which some authors adopt as regularization parameter (without applying data-driven selection criteria). However, this choice seems too large because it eliminates too many correct edges reaching high accuracy but relatively low precision and recall. This fact indicates that the choice of regularization parameter is still an open problem and a crucial one, and thus we conclude that, although BIC and CV are fundamental for choosing λ in the absence of any other information, their performance is not yet optimal.

3.2. Real Data Analysis

This section shows the application of *jewel* to gene expression datasets of patients with glioblastoma, which is the most common type of malignant brain tumor. We used three microarray datasets from Gene Expression Omnibus (GEO) database [28]: GSE22866 [29] (Agilent Microarray), GSE4290 [30], and GSE7696 [31,32] (both Affymetrix Array). We annotated the probes using the `biomaRt` R package. In case of multiple matches between probes and Ensembl gene ids, we gave preference to genes in common among all datasets or, in case of further uncertainty, to those present in selected pathways (see below). Then, we converted Ensembl gene ids to gene symbols, and we averaged gene expression over the probes, obtaining $K = 3$ matrices with dimensions $40 \times 20{,}861$, $77 \times 16{,}801$, $80 \times 16{,}804$, respectively. For the sake of simplicity, we considered only the $p = 13{,}323$ genes in common to all three datasets.

For this illustrative analysis, we limited the attention to the genes belonging to seven pathways from the Kyoto Encyclopedia of Genes and Genomes database [33] which were as-

sociated with cancer: p53 signaling pathway (hsa04115), glutamatergic synapse (hsa04724), chemokine signaling pathway (hsa04062), PI3K-Akt signaling pathway (hsa04151), glioma pathway (hsa05214), mTOR signaling pathway (hsa04150), and cytokine–cytokine receptor interaction (hsa04060). These pathways involve 920 genes in total; out of them, $p = 483$ were present in our datasets. Therefore, we applied *jewel* on this subset of genes. As described in the previous section, we selected the regularization parameter λ with both BIC and CV procedures. Finally, we compared the two estimated networks with a network obtained from the STRING database. In the following are the details.

First, when we used BIC (with the warm start) to estimate the optimal value of λ, we obtained $\lambda_{BIC} = 0.2223$ (see Figure 7). Therefore, the estimated graph G_{BIC} is the solution of *jewel* corresponding to this parameter. It has 3113 edges (about 2.7% of all possible edges), and all 483 vertices have a degree of at least 1.

When we used CV (with the warm start) to estimate the optimal value of λ, we obtained $\lambda_{CV} = 0.1151$ (see Figure 7). We ran *jewel* with this value of the regularization parameter. Resulting graph G_{CV} has 7272 edges (about 6.2% of all possible edges) and all 483 vertices have a degree of at least 1. As, in this example $\lambda_{CV} < \lambda_{BIC}$, G_{CV} has more connections than G_{BIC}.

Then, to better understand the identified connections, we analyzed the $p = 483$ genes in the STRING database [34]. STRING is a database of known and predicted protein–protein interactions that can be physical and functional and derived from lab experiments, known co-expression, and genomic context predictions and knowledge in the databases text mining. We limited the query to connections from "experiments" and "databases" as active interaction sources setting the minimum required interaction score to the highest value of 0.9. The resulting STRING network had 415 out of 483 vertices connected to any other node and 4134 edges.

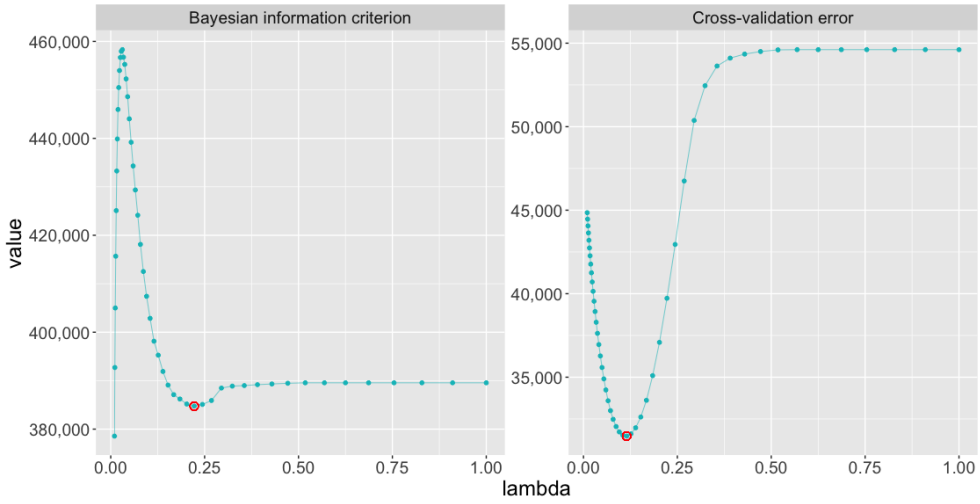

Figure 7. Left panel: values of BIC obtained for glioblastoma datasets with $K = 3$, $p = 483$ and $n_1 = 40$, $n_2 = 77$, $n_3 = 80$ over the uniform in log-scale grid of 50 parameters of λ from 0.01 to 1. **Right panel:** CV error obtained with the same settings. Red circles denote estimated optimal λ_{OPT}.

We measured the number of connections common to our estimated network and the network from the STRING database. For each case, Figure 8 shows the connections identified by *jewel* that were present also in the STRING database. For G_{BIC}, we observed 170 edges in common, while for G_{CV}, we had 297 common edges (see all the results summarized in Table 3).

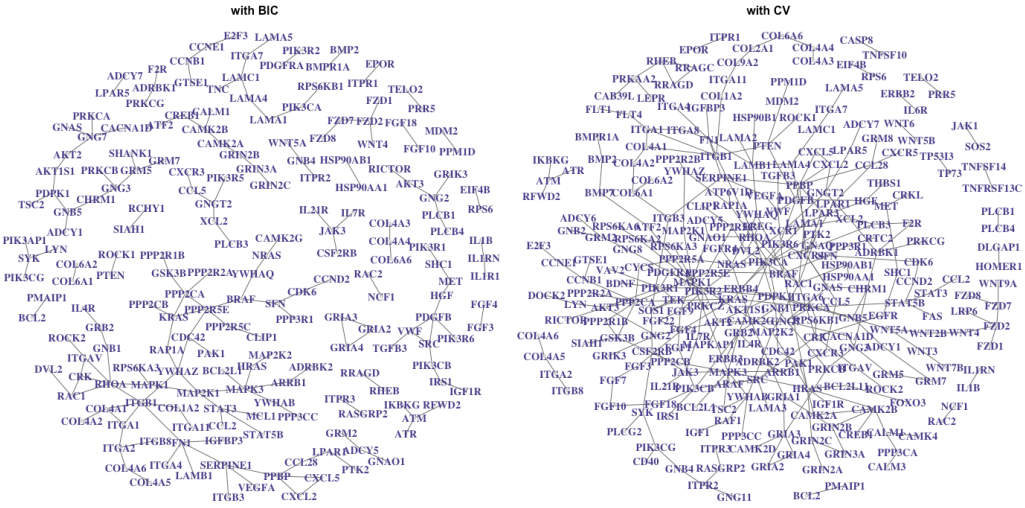

Figure 8. Intersection of the networks estimated with *jewel* from glioblastoma datasets and the one obtained from the STRING database. Regularization parameter, used in the estimation, was obtained with BIC (on the left) and with CV (on the right).

Table 3. Results of BIC and CV procedures obtained for $K = 3$ glioblastoma datasets with $p = 483$ over the uniform in log-scale grid of 50 parameters of λ from 0.01 to 1. The p-values is the results of the hyper-geometric test to assess the significance of the edge overlap.

	λ_{OPT}	# Est. Edges	# Edges in Intersection	p-Value
BIC	0.2223	3113/116,403	170/4134	3.29255×10^{-8}
CV	0.1151	7272/116,403	297/4134	0.00697

Although the number of edges in the intersection can be considered low at first sight, it is significant according to the hypergeometric test. Nevertheless, we should note two things: First, *jewel* seeks to identify conditional correlation among variables or equivalently linear relationships between two genes not mediated by other factors (i.e., other genes). Meanwhile, connections from the STRING database are not necessarily of such nature. Second, STRING contains general protein–protein interactions, i.e., interactions that are not necessarily present in the tissue/condition studied in used datasets. Therefore, we do not expect to identify mechanisms that might occur in other biological conditions (our gene expressions are glioblastoma).

However, we notice many groups of genes identified consistently, such as collagen alpha chains, ionotropic glutamate receptors, frizzled class receptors, interleukin 1 receptors, and fibroblast growth factors, collagen, and others. The biggest hubs in G_{BIC} include PPP3CC (frequently underexpressed in gliomas), RCHY1 (vice versa, typically highly expressed in this condition), and IL4R (is associated with better survival rates). In G_{CV}, the biggest hubs are TNN (that is considered a therapeutic target since an increase in expression can suppress brain tumor growth), CALML6, and BCL-2 (that can block apoptosis, i.e., cell death, and therefore may influence tumor prognosis).

To conclude, *jewel* demonstrated its ability to identify connections from the gene expression microarray data. However, it is possible that the choice of the regularization parameter still deserves improvements to achieve better results.

4. Discussion

The proposed method *jewel* is a methodological contribution to the GGM inference in the context of multiple datasets. It provides a valid alternative if the user is interested only in the structure of the GGM and not in covariance estimation. The proposed method is easy to use with the R package `jewel`.

There are still some aspects that can be improved and constitute directions for future work. As the performance of any regularization method depends on the choice of the tuning parameter, `jewel` could be improved with the better choice of λ. For example, in [24] the authors suggest quantile universal threshold, λ_{QUT}, which is the upper α-quantile of the introduced zero-thresholding function under the null model. When not only the response vector but also the design matrix is random (as in *jewel*), bootstrapping within the Monte Carlo simulation can be used to evaluate λ_{QUT}.

Moving to more methodological improvements, we can incorporate degree-induced weights into the minimization problem to account for the underlying graph's hub structure. In this way, we could overcome the decrease in performance demonstrated by all analyzed methods. Furthermore, we can consider other grouping approaches as the neighbor-dependent synergy described in [35]. Another possible improvement, when the underlying graphs are not the same across all the K datasets, is to decompose $\boldsymbol{\Theta}^{(k)}$ as a sum of two factors, one describing the common part and another the differences between the K graphs, then add a second group Lasso penalty term to capture differences between the networks. Other intriguing improvements regard the incorporation of specific prior knowledge, which would lead to different initialization of the **Active** matrix. For example, using variable screening procedures, i.e., a preliminary analysis of the input data that identifies connections that are not "important" with high probability, we can reduce the problem's dimensionality. Other aspects concern implementing the block-diagonalization approach, i.e., identifying blocks in the underlying graph and perform independent execution of *jewel* to each block. Such a choice does not influence performance but can significantly decrease the running time, especially if we parallelize the execution of *jewel* to different blocks.

Finally, another point that might greatly impact the applications of *jewel* to the analysis of gene expression has to do with the Gaussian assumption. Nowadays, RNA-seq data have become popular and are replacing the old microarray technology. However, RNA-seq are counts data. Therefore the Gaussian assumption does not hold. Methods such as *voom* [36] can be used to transform the RNA-seq data and stabilize the variance. *voom* estimates the mean-variance relationship of the log-counts, generates a precision weight for each observation and enters these into the *limma* [37] empirical Bayes analysis pipeline. With this transformation, the RNA-seq can be analyzed using similar tools as for microarrays, *jewel* included. As a more appealing alternative, one could develop a joint graphical approach in the context of count data, such as in [38–40].

Author Contributions: Conceptualization, C.A. and D.D.C.; methodology, C.A., D.D.C. and A.P.; software, A.P.; formal analysis, A.P.; investigation, C.A., D.D.C. and A.P.; resources, C.A.; data curation, A.P.; writing—original draft preparation, C.A., D.D.C. and A.P.; writing and editing, C.A., D.D.C. and A.P.; supervision, C.A. and D.D.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded in part by grant from the Regione Campania ADViSE Project.

Data Availability Statement: The article includes analysis of three publicly available datasets from Gene Expression Omnibus (GEO) database with accession numbers GSE22866 [29], GSE4290 [30], and GSE7696 [31,32].

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Barabasi, A.L. *Network Science*; Cambridge University Press: Cambridge, UK, 2018.
2. Pržulj, N. *Analyzing Network Data in Biology and Medicine*; Cambridge University Press: Cambridge, UK, 2019.
3. Shang, Y. On the likelihood of forests. *Phys. A Stat. Mech. Appl.* **2016**, *456*, 157–166. [CrossRef]
4. Bühlmann, P.; van de Geer, S. *Statistics for High-Dimensional Data: Methods, Theory and Applications*; Springer: Berlin/Heidelberg, Germany, 2011. [CrossRef]
5. Giraud, C. *Introduction to High-Dimensional Statistics*; Springer: Berlin/Heidelberg, Germany, 2015. [CrossRef]
6. Hastie, T.; Tibshirani, R.; Wainwright, M.J. *Statistical Learning with Sparsity: The Lasso and Generalizations*; Chapman and Hall/CRC: Boca Raton, FL, USA, 2015. [CrossRef]

7. Wainwright, M.J. *High Dimensional Statistics: A Non-Asymptotic Viewpoint*; Cambridge University Press: Cambridge, UK, 2019. [CrossRef]
8. Yuan, M.; Lin, Y. Model selection and estimation in the Gaussian graphical model. *Biometrika* **2007**, *94*, 19–35. [CrossRef]
9. Friedman, J.; Hastie, T.; Tibshirani, R. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics* **2008**, *9*, 432–441. [CrossRef] [PubMed]
10. Meinshausen, N.; Bühlmann, P. High-dimensional graphs and variables selection with lasso. *Ann. Stat.* **2006**, *34*, 1436–1462. [CrossRef]
11. Lin, D.; Zhang, J.; Li, J.; Hao, H.; Deng, H.W.; Wang, Y.P. Integrative analysis of multiple diverse omics datasets by sparse group multitask regression. *Front. Cell Dev. Biol.* **2014**, *2*, 62. [CrossRef] [PubMed]
12. Rohart, F.; Gautier, B.; Singh, A.; Cao, K.A.L. mixOmics: An R package for 'omics feature selection and multiple data integration. *PLoS Comput. Biol.* **2017**, *13*, e1005752. [CrossRef] [PubMed]
13. Guo, G.; Levina, E.; Michailidis, G.; Zhu, J. Joint estimation of multiple graphical models. *Biometrika* **2011**, *98*, 1–15. [CrossRef]
14. Danaher, P.; Wang, P.; Witten, D. The joint graphical lasso for inverse covariance across multiple classes. *J. R. Stat. Soc. B* **2014**, *76*, 373–397. [CrossRef]
15. Shan, L.; Kim, I. Joint estimation of multiple Gaussian graphical models across unbalanced classes. *Comput. Stat. Data Anal.* **2018**, *121*, 89–103. [CrossRef]
16. Huang, F.; Chen, S.; Huang, S. Joint Estimation of Multiple Conditional Gaussian Graphical Models. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *29*, 3034–3046. [CrossRef]
17. Ma, J.; Michailidis, G. Joint Structural Estimation of Multiple Graphical Models. *J. Mach. Learn.* **2016**, *17*, 1–48.
18. Chiquet, J.; Grandvalet, Y.; Ambroise, C. Inferring multiple graphical structures. *Stat. Comput.* **2011**, *21*, 537–553. [CrossRef]
19. De Canditiis, D.; Guardasole, A. Learning Gaussian Graphical Models by symmetric parallel regression technique. In Proceedings of the 15th Meeting on Applied Scientific Computing and Tools (MASCOT 2018), Rome, Italy, 2–5 October 2018.
20. Breheny, P.; Huang, J. Group descent algorithms for nonconvex penalized linear and logistic regression models with grouped predictors. *Stat Comput.* **2015**, *25*, 173–187. [CrossRef] [PubMed]
21. Peng, J.; Wang, P.; Zhou, N.; Zhu, J. Partial Correlation Estimation by Joint Sparse Regression Models. *J. Am. Stat. Assoc.* **2009**, *104*, 735–746. [CrossRef]
22. Mohan, K.; London, P.; Fazel, M.; Witten, D.; Lee, S.I. Node-based learning of multiple Gaussian graphical models. *J. Mach. Learn. Res.* **2014**, *15*, 445–488.
23. Basu, S.; Shojaie, A.; Michailidis, G. Network Granger causality with inherent grouping structure. *J. Mach. Learn. Res.* **2015**, *16*, 417–453.
24. Giacobino, C.; Sardy, S.; Diaz-Rodriguez, J. Quantile universal threshold. *Electron. J. Stat.* **2017**, *11*, 4701–4722. [CrossRef]
25. Csardi, G.; Nepusz, T. The igraph software package for complex network research. *InterJ. Complex Syst.* **2006**, *1695*, 1–9.
26. Rohart, F.; Eslami, A.; Matigian, N.; Bougeard, S.; Cao, K.-A.L. MINT: A multivariate integrative method to identify reproducible molecular signatures across independent experiments and platforms. *BMC Bioinform.* **2017**, *18*, 128, [CrossRef]
27. Sulaimanov, N.; Kumar, S.; Burdet, F.; Ibberson, M.; Pagni, M.; Koeppl, H. Inferring gene expression networks with hubs using a degree weighted Lasso approach. *Bioinformatics* **2019**, *35*, 987–994. [CrossRef] [PubMed]
28. Barrett, T.; Suzek, T.O.; Troup, D.B.; Wilhite, S.E.; Ngau, W.-C.; Ledoux, P.; Rudnev, D.; Lash, A.E.; Fujibuchi, W.; Edgar, R. NCBI GEO: Mining millions of expression profiles—Database and tools. *Nucleic Acids Res.* **2005**, *33*, D562–D566. [CrossRef]
29. Atcheverry, E.; Aubry, M.; de Tayrac, M.; Vauleon, E.; Boniface, R.; Guenot, F.; Saikali, S.; Hamlat, A.; Riffaud, L.; Menei, P.; et al. DNA methylation in glioblastoma: Impact on gene expression and clinical outcome. *BMC Genom.* **2010**, *11*, 701. [CrossRef]
30. Sun, L.; Hui, A.M.; Su, Q.; Vortmeyer, A.; Kotliarov, Y.; Pastorino, S.; Passaniti, A.; Menon, J.; Walling, J.; Bailey, R.; et al. Neuronal and glioma-derived stem cell factor induces angiogenesis within the brain. *Cancer Cell* **2006**, *9*, 287–300. [CrossRef] [PubMed]
31. Murat, A.; Migliavacca, E.; Gorlia, T.; Lambiv, W.L.; Shay, T.; Hamou, M.F.; de Tribolet, N.; Regli, L.; Wick, W.; Kouwenhoven, M.C.M.; et al. Stem cell-related "self-renewal" signature and high epidermal growth factor receptor expression associated with resistance to concomitant chemoradiotherapy in glioblastoma. *J. Clin. Oncol.* **2008**, *26*, 3015–3024. [CrossRef]
32. Lambiv, W.L.; Vassallo, I.; Delorenzi, M.; Shay, T.; Diserens, A.C.; Misra, A.; Feuerstein, B.; Murat, A.; Migliavacca, E.; Hamou, M.F.; et al. The Wnt inhibitory factor 1 (WIF1) is targeted in glioblastoma and has a tumor suppressing function potentially by induction of senescence. *Neuro-Oncology* **2011**, *13*, 736–747. [CrossRef]
33. Kanehisa, M.; Furumichi, M.; Sato, Y.; Ishiguro-Watanabe, M.; Tanabe, M. KEGG: Integrating viruses and cellular organisms. *Nucleic Acids Res.* **2021**, *49*, D545–D551. [CrossRef]
34. Jensen, L.J.; Kuhn, M.; Stark, M.; Chaffron, S.; Creevey, C.; Muller, J.; Doerks, T.; Julien, P.; Roth, A.; Simonovic, M.; et al. STRING 8—A global view on proteins and their functional interactions in 630 organisms. *Nucleic Acids Res.* **2009**, *37*, D412–D416. [CrossRef] [PubMed]
35. Shang, Y. Consensus formation in networks with neighbor-dependent synergy and observer effect. *Commun. Nonlinear Sci. Numer. Simul.* **2021**, *95*, 105632. [CrossRef]
36. Law, C.W.; Chen, Y.; Shi, W.; Smyth, G.K. voom: Precision weights unlock linear model analysis tools for RNA-seq read counts. *Genome Biol.* **2014**, *15*, R29. [CrossRef] [PubMed]
37. Ritchie, M.; Phipson, B.; Wu, D.; Hu, Y.; Law, C.W.; Shi, W.; Smyth, G.K. Limma powers differential expression analyses for RNA-sequencing and microarray studies. *Nucleic Acids Res.* **2015**, *43*, e47. [CrossRef]

38. Chiquet, J.; Mariadassou, M.; Robin, S. Variational Inference of Sparse Network from Count Data. In Proceedings of the 36th International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019.
39. Yang, E.; Ravikumar, P.; Allen, G.I.; Liu, Z. Graphical Models via Generalized Linear Models. In Proceedings of the 25th International Conference on Neural Information Processing Systems (NIPS), Lake Tahoe, NV, USA, 3–6 December 2012.
40. Hue, N.T.K.; Chiogna, M. Structure Learning of Undirected Graphical Models for Count Data. *J. Mach. Learn. Res.* **2021**, *22*, 1–53.