Chapter 1
A multiattribute Gaussian graphical model for inference of multiscale regulatory network: an application in breast cancer

Julien Chiquet, Guillem Rigaill, Martina Sundqvist

1.1 Introduction

Write the damn thing

Gaussian Graphical Models (GGMs) **??** are a very convenient tool for describing the patterns at play in complex data sets. Indeed, through the notion of partial correlation, they provide a well-studied framework for spotting direct relationships between variables, and thus reveal the latent structure in a way that can be easily interpreted. Application areas are very broad and include for instance gene regulatory network inference in biology (using gene expression data) as well as spectroscopy, climate studies, functional magnetic resonance imaging, etc. Estimation of GGMs in a sparse, high-dimensional setting has thus received much attention recently.

write a paragraph about generalization of GGM towards regulatory network inference

[...]

A deeper generalization of GGM comes by integrating multiple types of data measured from diverse platforms, what is sometimes referred to as *horizontal* integration: not only does this means a better treatment of the heterogeneity of the data, but it also makes the network reconstruction more robust.

1.2 Background

We should probably shorten this section

This section provides an overview on the state-of-the-art ℓ_1-regularization methods for sparse GGM inference and their most recent striking variants, insisting on their computational and statistical properties. This provides the reader with the necessary material to approach the model at play in this chapter, dedicated to multiattribute GGM.

1.2.1 Basics on Gaussian graphical models

Let $\mathscr{P} = \{1, \dots, p\}$ be a set of fixed vertices and $X = (X_1, \dots, X_p)^\intercal$ a random vector describing a signal over this set. The vector $X \in \mathbb{R}^p$ is assumed to be multivariate Gaussian with unknown mean and unknown covariance matrix $\Sigma = (\Sigma_{ij})_{(i,j)\in\mathscr{P}^2}$. No loss of generality is involved when centering X, so we may assume that $X \sim \mathscr{N}(\mathbf{0}_p, \Sigma)$. The covariance matrix Σ, equal to $\mathrm{e}(XX^\intercal)$ under the assumption that X is centered, belongs to the set $\mathscr{S}_p^+$ of positive definite symmetric matrices of size $p \times p$.

Graph of conditional dependencies.

GGMs endow Gaussian random vectors with a graphical representation $\mathscr{G}$ of their *conditional dependency structure*: two variables i and j are linked by an undirected edge (i, j) if, conditional on all other variables indexed by $\mathscr{P}\backslash\{i, j\}$, random variables X_i and X_j remain or become dependent. Thanks to the Gaussian assumption, conditional independence actually boils down to a zero conditional covariance $\mathrm{cov}(X_i, X_j | X_{\mathscr{P}\backslash\{i,j\}})$, or equivalently to a zero partial correlation which we denote by ρ_{ij}, the latter being a normalized expression of the former.

Concretely, the inference of a GGM is based upon a classical result originally emphasized in **?** stating that partial correlations ρ_{ij} are actually proportional to the corresponding entries in the *inverse* of the covariance matrix $\Sigma^{-1} = \Theta$, also known as the *concentration matrix*. More precisely, we have

$$\rho_{ij} = -\Theta_{ij}/\sqrt{\Theta_{ii}\Theta_{jj}}, \qquad \Theta_{ii} = \mathrm{Var}(X_i | X_{\mathscr{P}\backslash i})^{-1}; \tag{1.1}$$

thus Θ directly describes the conditional dependency structure of X. Indeed, after a simple rescaling, Θ can be interpreted as the adjacency matrix of an undirected weighted graph representing the partial covariance (or correlation) structure between variables $X_1, \dots, X_p$. Formally, we denote by $\mathscr{G} = (\mathscr{P}, \mathscr{E})$ this graph, the edges of which are characterized by

$$(i, j) \in \mathscr{E} \Leftrightarrow \Theta_{ij} \neq 0, \quad \forall (i, j) \in \mathscr{P}^2 \text{ such that } i \neq j.$$

In words, $\mathscr{G}$ has no self-loop and contains all edges (i, j) such that Θ_{ij} is nonzero. Therefore recovering nonzero entries of Θ is equivalent to inferring the graph of conditional dependencies $\mathscr{G}$, and the correct identification of nonzero entries is the main issue in this framework.

Maximum Likelihood inference.

GGMs fall into the family of exponential models for which the whole range of classical statistical tools applies. As soon as the sample size n is greater than the number

p of variables, the likelihood admits a unique maximum over $\mathscr{S}_p^+$, defining a maximum likelihood estimator (MLE): suppose we observe a sample $\{X^1,\dots,X^n\}$ composed of n i.i.d. copies of X, stored row-wise once centered in a matrix $\mathbf{X} \in \mathbb{R}^{n\times p}$ such that $(X^i)^\top$ is the ith row of $\mathbf{X}$. The empirical covariance matrix is denoted by $\mathbf{S}_n = \mathbf{X}^\intercal\mathbf{X}/n$. Maximizing the likelihood is equivalent to

$$\widehat{\Theta}^{\mathrm{mle}} = \underset{\Theta\in\mathscr{S}_p^+}{\arg\max} \quad \log\det(\Theta) - \mathrm{Tr}(\Theta\mathbf{S}_n). \tag{1.2}$$

When $n > p$, Problem (1.2) admits a unique solution equal to $\mathbf{S}_n$. The scaled empirical covariance matrix $\mathbf{S}_n$ follows a Wishart distribution while its inverse $\mathbf{S}_n^{-1}$ follows an inverse Wishart distribution with computable parameters.

There are two major limitations with the MLE regarding the objective of graph reconstruction by recovering the pattern of zeroes in Θ. First, it provides an estimate of the saturated graph: all variables are connected to each other; second, we need n to be larger than p to be able to even define this estimator, which is rarely the case in genomics. In any case, the need for regularization and feature selection is huge. A natural assumption is that the true set of direct relationships between the variables remains small, that is, the true underlying graph is sparse (say, of the order of p rather than the order of p^2). Sparsity makes estimation feasible in the case where $n < p$ since we can concentrate on sparse or shrinkage estimators with fewer degrees of freedom than in the original problem. Henceforth, the question of selecting the correct set of edges in the graph is treated as a question of variable selection.

High-dimensional inference of GGM.

The different methods for the inference of sparse GGMs in high-dimensional settings fall into roughly three categories. The first contains constraint-based methods, performing statistical tests **?????**. However, they either suffer from the excessive computational burden **??** or strong assumptions **??** that correspond to regimes never attained in real situations. The second of these categories is composed of Bayesian approaches, see for instance **????**. However, constructing priors on the set of concentration matrices is not a trivial task and the use of MCMC procedures limits the range of applications to moderate-sized networks. The third category contains regularized estimators, which add a penalty term to the likelihood in order to reduce the complexity or degrees of freedom of the estimator and more generally regularize the problem: throughout this chapter we focus on ℓ_1-regularized procedures, which are freed from any test procedure – and thus multiple testing issues – since they directly perform estimation and selection of the most significant edges by zeroing entries in the estimator of Θ. The remainder of this section is dedicated to a quick review of the state-of-the-art methods of this kind.

1.2.2 Sparse methods for GGM inference

The idea underlying sparse methods for GGM is the same as for the Lasso in linear regression (**?**): it basically uses ℓ_1-regularization as a convex surrogate of the ideal but computationally intensive ℓ_0-regularized problem:

$$\underset{\Theta \in \mathscr{S}_p^+}{\arg\max} \quad \log\det(\Theta) - \mathrm{Tr}(\Theta \mathbf{S}_n) - \lambda \, \|\Theta\|_{\ell_0}. \tag{1.3}$$

Problem (1.3) achieves a trade-off between the maximization of the likelihood and the sparsity of the graph within a single optimization problem. The penalty term can also be interpreted as a log prior on the coefficients in a Bayesian perspective. BIC or AIC criteria are special cases of such ℓ_0 regularized problems, except that the maximization is made upon a restricted subset of candidates $\{\tilde{\Theta}_1, \dots, \tilde{\Theta}_m\}$ and the choice of λ is fixed ($\log(n)$ for BIC and $1/2$ for AIC). Actually solving (1.3) would require the exploration of all possible 2^p graphs. On the contrary, by preserving the convexity of the optimization problem, ℓ_1-regularization paves the way to fast algorithms. For the price of a little bias on all the coefficients, we get to shrink some coefficients to exactly 0, operating selection and estimation in one single step as hoped in Problem (1.3).

Graphical-Lasso.

The criterion optimized by the graphical-Lasso was simultaneously proposed in **?** and **?**. It corresponds to the estimator obtained by fitting the ℓ_1-penalized Gaussian log-likelihood, *i.e.* the tightest convex relaxation of (1.3):

$$\widehat{\Theta}_\lambda^{\mathrm{glasso}} = \underset{\Theta \in \mathscr{S}_p^+}{\arg\max} \; \log\det(\Theta) - \mathrm{Tr}(\Theta \mathbf{S}_n) - \lambda \, \|\Theta\|_{\ell_1}. \tag{1.4}$$

In this regularized problem, the ℓ_1-norm drives some coefficients of Θ to zero. The non-negative parameter λ tunes the global amount of sparsity: the larger the λ, the fewer edges in the graph. A large enough penalty level produces an empty graph. As λ decreases towards zero, the estimated graph tends towards the saturated graph and the estimated concentration matrix tends towards the usual MLE (1.2). By construction, this approach guarantees a well-behaved estimator of the concentration matrix *i.e.* sparse, symmetric and positive-definite, which is a great advantage of this method.

Ever since Criterion (1.4) was proposed, many efforts have been dedicated to developing efficient algorithms for its optimization. In the original proposal of **?**, it is shown that solving for one row of matrix Θ in (1.4) while keeping other rows fixed boils down to a Lasso problem. The global problem is solved by cycling over the matrix rows until convergence. Thus, if one considers that L passes over the whole matrix are needed to reach convergence, a rough estimation of the overall cost is of the order of $Lp \times$ (cost for solving for one row). With a block-coordinate update each iteration over a row has $\mathcal{O}(p^3)$ complexity and their implementation is $\mathcal{O}(Lp^4)$ for L sweeps over the whole matrix $\widehat{\Theta}$. In **?** again, a rigorous analysis is conducted in Nesterov's framework **?** showing that the complexity for a single λ reaches $\mathcal{O}(p^{4.5}/\varepsilon)$ where ε is the desired accuracy of the final estimate.

The *Graphical-Lasso* algorithm of **?** follows the same line but builds on a coordinate descent algorithm to solve each underlying Lasso problem. While no precise complexity analysis is possible with these methods, empirical results tend to show that this algorithm is faster than the original proposal of **?**. Additional insights on the convergence of the graphical-Lasso are provided in **?**, simultaneously with **?**, showing how to take advantage of the problem sparsity by decomposing (1.4) into block diagonal problems depending on λ: this considerably reduces the computational burden in practice. Implementations of the graphical-Lasso algorithm are available in the R-packages **glasso**, **huge ?**, or **simone ??**. The most recent notable efforts related to the optimization of (1.4) are due to **??** and the QUIC (then BIG&QUIC) algorithm, a quadratic approximation which allows (1.4) to be solved up to $p = 1,000,000$ with a super-linear rate of convergence and with bounded memory. The R-package **quic** implements the first version of this algorithm.

On the statistical side, the most striking results are due to **?**: they show that selection consistency of the estimator defined by (1.4) – that is, recovery of the true underlying graphical structure –, is met in the sub-Gaussian case when, for an appropriate choice of λ, the sample size n is of the same order as $\mathcal{O}(d^2 \log(p))$, where d is the highest degree in the target graph. Additional conditions on the empirical covariance between relevant and irrelevant features are required, known as the "irrepresentability conditions" in the Lasso case. Such statistical results are important since they provide insights on the "data" situations where such methods may either be successful or completely hopeless. More on this is discussed in **?**. For instance, this should prevent blindly applying the graphical-Lasso in situations where the sample size n is too small compared to p. Similarly, when the presence of hub nodes with high degree is suspected, the estimated graph should be interpreted with care.

Neighborhood selection.

This approach, proposed in **?**, determines the graph of conditional dependencies by solving a series of p independent Lasso problems, successively estimating the neighborhoods of each variable and then applying a final reconciliation step as post-treatment to recover a symmetric adjacency matrix. Concretely, a given column $\mathbf{X}_j$ of the data matrix is "explained" by the remaining columns $\mathbf{X}_{\backslash j}$ corresponding to the remaining variables: the set $\mathrm{ne}(j)$ of neighbors of variable j in the graph $\mathscr{G}$ is estimated by the support of the vector solving

$$\hat{\beta}_j = \underset{\beta \in \mathbb{R}^{p-1}}{\arg\min} \frac{1}{2n} \left\| \mathbf{X}_j - \mathbf{X}_{\backslash j}\beta \right\|_{\ell_2}^2 + \lambda \, \|\beta\|_{\ell_1}. \tag{1.5}$$

Indeed, if each row of $\mathbf{X}$ is drawn from a multivariate Gaussian $\mathscr{N}(\mathbf{0},\Theta^{-1})$, then the best linear approximation of $\mathbf{X}_j$ by $\mathbf{X}_{\backslash j}$ is given by

$$\mathbf{X}_j = \sum_{k\in \text{ne(j)}} \beta_{jk}\mathbf{X}_k = -\sum_{k\in \text{ne(j)}} \frac{\Theta_{jk}}{\Theta_{jj}}\mathbf{X}_k, \tag{1.6}$$

thus coefficients β_j and column Θ_j – once its diagonal elements are removed – share the same support. By support, we mean the set of nonzero coefficients. Adjusting (1.5) for each $j = 1,\dots,p$ allow us to reconstruct the full graph $\mathscr{G}$. Because the neighborhoods of the p variable are selected separately, a post symmetrization must be applied to manage inconsistencies between edge selections; **?** suggests AND or OR rules.

Let us fill the gap with Criterion (1.4). First, note that the p regression problem can be rewritten as a unique matrix problem, where $\mathbf{B}$ contains p vectors $\beta_j, j = 1,\dots,p$:

$$\hat{\mathbf{B}}^{\text{ns}} = \underset{\mathbf{B}\in\mathbb{R}^{p\times p}, \text{diag}(\mathbf{B})=\mathbf{0}_p}{\arg\min} \frac{1}{2}\text{Tr}(\mathbf{B}^\top\mathbf{S}_n\mathbf{B}) - \text{Tr}(\mathbf{B}^\top\mathbf{S}_n) + \lambda\|\mathbf{B}\|_{\ell_1}. \tag{1.7}$$

In fact, it can be shown **???** that the optimization problem (1.7) corresponds to the minimization of a penalized, negative *pseudo*-likelihood: the joint distribution of X is approximated by the product of the p distributions of the p variables conditional on the other ones, that is

$$\log \mathbb{P}(\mathbf{X};\Theta) = \sum_{j=1}^{p}\sum_{i=1}^{n} \log \mathbb{P}(X_j^i | X_{\backslash j}^i; \Theta_j).$$

This pseudo-likelihood is based upon the (false) assumption that conditional distributions are independent. Moreover, all variables are assumed to share the same variance in this formulation. Building on these remarks, **?** amend criterion (1.7) by the adjunction of an additional symmetry constraint, and introduce additional parameters to account for different variances between the variables.

Concerning the computational aspect, this approach has very efficient implementation as it basically boils down to solving p Lasso problems. Suppose for instance that the target neighborhood size is k per variable: fitting the whole solution path of a Lasso problem using the Lars algorithm can be done in $\mathscr{O}(npk)$ complexity **?**. This must be multiplied by p for the whole network, yet we underline that a parallel implementation is straightforward in this case. This makes this approach quite competitive, especially when coupled with additional bootstrap or resampling techniques **?**.

On the statistical side, neighborhood selection has been reported to be sometimes empirically more accurate in terms of edge detection than is the graphical-Lasso **??** on certain types of data. This is somewhat supported by the statistical analysis of **?**, who show that under the classical irrepresentability conditions for the Lasso **??** and for an appropriate choice of λ, neighborhood selection achieves selection consistency with high probability when the sample size n is of the order of $\mathscr{O}(d\log(p))$ with d the maximal degree of the target graph $\mathscr{G}$. This is to be compared with the $\mathscr{O}(d^2\log(p))$ required by the graphical-Lasso (even if the corresponding "irrepresentability conditions" are not strictly comparable). A rough explanation for this difference on the asymptotic is that the graphical-Lasso intends to estimate the concentration matrix on top of selecting the nonzero entries, while neighborhood selection focuses on the selection problem.

Sparse PArtial Correlation Estimation (SPACE).

In **?**, the gap is completely filled between linear regression, Gaussian graphical model and neighborhood selection with a method that directly penalizes the partial correlations within the linear model. Indeed, by combining firstly Relationship 1.1 between the partial correlations and the concentration matrix, and secondly, Relationship 1.6 between the coefficients in linear regression and concentration matrix, one has

$$\mathbf{X}_j = \sum_{k\in\text{ne(j)}} \beta_{jk}\mathbf{X}_k + \varepsilon = \sum_{k\in\text{ne(j)}} \rho_{jk}\sqrt{\frac{\Theta_{kk}}{\Theta_{jj}}}\mathbf{X}_k + \varepsilon,$$

which suggests the following optimization problem

$$\left(\widehat{\rho}_{\lambda}^{\text{space}}, \operatorname{diag}(\Theta)\right) = \underset{\rho\in\mathbb{R}p(p-1),\operatorname{diag}(\Theta)}{\arg\min}\ \frac{1}{2}\sum_{j=1}^{p}\omega_j\left\|\mathbf{X}_j - \sum_{k=1}^{p}\rho_{jk}\sqrt{\frac{\Theta_{kk}}{\Theta_{jj}}}\mathbf{X}_k\right\|_{\ell_2}^{2} + \lambda\ \|\rho\|_{\ell_1}, \tag{1.8}$$

where ρ is a vector containing all the pairwise partial correlations, $\operatorname{diag}(\Theta)$ contains the diagonal elements of Θ, that is to say, the partial covariances of all the variables, and finally ω_j are some positive (given) weights.

Although the optimization of (1.8) is more demanding than is neighborhood selection, the problem is jointly convex in $(\operatorname{diag}(\Theta),\rho)$. When $\operatorname{diag}(\Theta)$ is fixed, the problem has the same complexity as does neighborhood selection, and the authors claim that only a couple of iterations alternating over each of the two parameters $(\operatorname{diag}(\Theta),\rho)$ are needed for convergence. It thus remains a lot more efficient than the graphical-Lasso. On top of that, the method intrinsically imposes symmetry over the partial correlations ρ. In short, it embeds the computational advantage of neighborhood selection while estimating the conditional variance as in the graphical-Lasso. It is available in the `R`-package **space**. Further refinements and statistical analyses have been recently proposed in **?**.

Model selection issues.

Up to this point, we have completely avoided the fundamental model selection issue, that is, the choice of the tuning parameter λ, which is at play in all the sparse methods mentioned thus far. The first possibility is to rely on information criteria of the form

$$\mathrm{IC}_\lambda = -2\mathrm{loglik}(\widehat{\Theta}_\lambda;\mathbf{X}) + \mathrm{pen}(\mathrm{df}(\widehat{\Theta}_\lambda)),$$

where "pen" is a function penalizing the model complexity, described by df, the degrees of freedom of the current estimator. We meet the AIC by choosing $\mathrm{pen}(x) = 2x$ and the BIC by choosing $\mathrm{pen}(x) = \log(n)x$. However, AIC and BIC are based upon assumptions which are not suited to high-dimensional settings (see **?**). Moreover, the notion of degrees of freedom for sparse methods has to be specified, not to mention that one has to adapt these criteria to the case of GGMs. An example of a criterion meeting these prerequisites is the extended BIC for sparse GGMs **?**:

$$\mathrm{EBIC}_\gamma(\widehat{\Theta}_\lambda) = -2\mathrm{loglik}(\widehat{\Theta}_\lambda;\mathbf{X}) + |\mathscr{E}_\lambda|(\log(n) + 4\gamma\log(p)), \qquad (1.9)$$

where the function df is equal to $|\mathscr{E}|$, the total number of edges in the inferred graph. The parameter $\gamma \in [0,1]$ is used to adjust the tendency of the usual BIC – recovered for $\gamma = 0$ – to choose overly dense graphs in the high-dimensional setting. Further justification can be found in **?**. A competing approach, designed to compare a family of GGM – possibly inferred with different methods –, is GGMSelect **??**.

Another possibility is to rely on resampling/subsampling procedures to select a set of edges which are robust to small variations of the sample. The most popular approach is the *Stability Selection* procedure proposed in **?**, also related to the bootstrapped procedure of **?**. A similar approach, called StaRS (Stability approach to Regularization Selection) is developed specifically in the context of GGM in **?**. The basic idea is as follows: for a given range of the tuning parameter $\Lambda = [\lambda_{\min}, \lambda_{\max}]$, the same method is fitted on many subsamples (with or without replacement) with size, say $n/2$. The idea is then to construct a score indexed on Λ that measures stability – or instability – of the selected variables. The selected edges are those matching a given score, for which the probability of false discovery is controlled. This requires an additional threshold in place of a choice of λ, but the authors in **??** claim that such a threshold is typically much less sensitive than is the tuning parameter λ. An application of such resampling techniques to the inference of biological networks has been pursued with success in **?**, advocating for the use of stability methods on real problems.

A final possibility — that remains somewhat confidential while writing these lines — is to rely on sparse procedures which are less sensitive to λ: among these, we may cite the "scaled-Lasso" **?** for linear regression, adapted to the context of network inference in a neighborhood-selection-like fashion in **?**.

Extensions towards non Gaussian settings.

As hopefully illustrated throughout this section, sparse GGM is a mature and well controlled framework, with solid contributions both on the statistical and the computational sides. There is also expanding innovative literature tending to broaden the applicability of GGMs, especially to overcome the Gaussian assumption. Indeed, particularly in genomics, there is a growing interest for the multivariate modeling of discrete random vectors, as sequencing techniques provide us with count data. In this perspective, some attempts were made for a Poisson version of the above techniques: in **?** for instance, the neighborhood selection approach is extended to a sparse generalized linear model setup; still, interpretability of the inferred network is questionable, as a null partial correlation does not mean conditional dependency in the non-Gaussian case. In a recent paper **?**, a review of existing Poisson graphical models is provided, where the notion of conditional dependency is more carefully specified.

Finally, there is much interest for pretreatment methods which change the original data into more "Gaussian" data via simple transformations. Hence, we can still take advantage of the well-controlled sparse GGM framework. A successful work based on Gaussian copulas is the nonparanormal distribution developed in **?**. It is implemented within the `R`-package **huge**, at a negligible cost compared to that of the inference process itself.

1.3 Accounting for multiscale data: multi-attribute GGM

We now place ourselves in the situation where, for our collection of features $\mathscr{P}$, we observe not one but several attributes. The question at hand remains the same, that is to say, unraveling strong interactions between these features according to the observation of their attributes. Such networks are known as "association networks", which are systems of interacting elements, where a link between two different elements indicates a sufficient level of similarity between element attributes. In this section, we are interested in reconstructing such networks based upon n observations of a set of K attributes of the p elements composing the vertices of the network. To this end, we propose a natural generalization of sparse GGM to sparse *multi-attribute* GGMs.

This work was planned for submission as a research paper I came across an independent work **?** on the `arXiv` that proposes nearly the same approach for the Graphical Lasso. The difference here is to consider the framework of neighborhood selection approach. We should mention this at some point of the chapter.

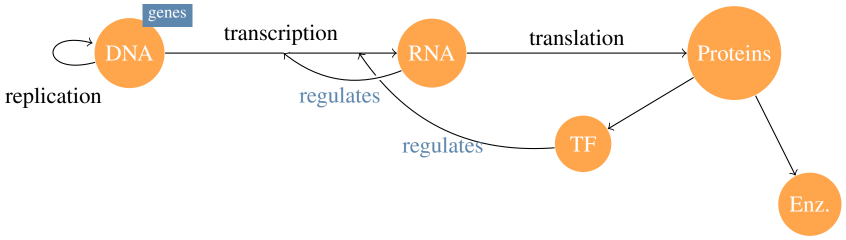


Fig. 1.1 Basic example of a multi-attribute network in genomics: activity of a gene can be measured at the transcriptomic and proteomic levels, and gene regulation affected accordingly

Why multi-attribute networks?

The need for multi-attribute networks is relevant in many application fields, but seems particularly applicable in genomics. Indeed, with the plurality of emerging technologies and sequencing techniques, it is possible to record many signals related to the same set of biological features at various scales or locations of the cell. Consider for instance the simplifying – still hopefully didactic – central dogma of molecular biology, sketched in Figure 1.1: basically, expression of a gene encoding for a protein can be measured either at the transcriptome level, in terms of its quantity of mRNA, or at the protein level, in terms of the concentration of the associated protein. Still, different technologies are used to measure either the transcriptome or the proteome, typically, microarray or sequencing technology for gene expression levels and cytometric or spectrometric experiments for protein concentrations. Although these signals are very heterogeneous (different levels of noise, count vs. continuous data, etc.), they do share commonality as they undergo common biological processes. We then put an edge in the network if it is supported in both spaces (gene and protein spaces). Our hope is that molecular profiles combined on the same set of biological samples can be *synergistic*, in order to identify a "consensus" and hopefully more robust network.

Multi-attribute GGM.

Let $\mathscr{P} = \{1,\dots,p\}$ be a set of variables of interest, each of them having some K attributes. Consider the random vector $X = (X_1,\dots,X_p)^\intercal$ such as $X_i = (X_{i1},\dots,X_{iK})^\intercal \in \mathbb{R}^K$ for $i \in \mathscr{P}$. The vector $X \in \mathbb{R}^{pK}$ describes the K recorded signals for the p features. We assume that X is a multivariate centered Gaussian vector, that is, $X \sim \mathscr{N}(\mathbf{0},\Sigma)$, with covariance and concentration matrices defined block-wise

$$\Sigma = \begin{bmatrix} \Sigma_{11} & & \Sigma_{1p} \\ & \ddots & \\ \Sigma_{p1} & & \Sigma_{pp} \end{bmatrix}, \quad \Theta = \begin{bmatrix} \Theta_{11} & & \Theta_{1p} \\ & \ddots & \\ \Theta_{p1} & & \Theta_{pp} \end{bmatrix}, \quad \Sigma_{ij}, \Theta_{ij} \in \mathscr{M}_{K,K}, \ \forall (i,j) \in \mathscr{P}^2,$$

where $\mathscr{M}_{a,b}$ is the set of real-valued matrices with a rows, b columns. Such a multi-attribute framework has been studied in **?** with a reconstruction method based upon canonical correlations in order to test dependencies between pairs (i,j) at the attribute level using covariance. Here, we propose to rely on partial correlations in a multivariate framework rather than (canonical) correlations to describe relationships between the features, and thus extend GGM to a multi-attribute framework. The objective is to define a "canonical" version of partial correlations. In our setting, the target network $\mathscr{G} = (\mathscr{P},\mathscr{E})$ is defined as the multivariate analog of the conditional graph for univariate GGM, that is

$$(i,j) \in \mathscr{E} \Leftrightarrow \Theta_{ij} \neq \mathbf{0}_{KK}, \quad \forall i \neq j. \tag{1.10}$$

In words, there is no edge between two variables i and j when their attributes are all conditionally independent.

A multivariate version of neighborhood selection.

Our idea for performing sparse multi-attribute GGM inference is to define a multivariate analog of the neighborhood selection approach **?** (see Section 1.2.2, Equations (1.5) and (1.7)). Indeed, it seems to be the most natural and convenient setup toward multivariate generalization. Nevertheless, we think that the graphical-Lasso (1.4), CLIME (**??**) or SPACE (1.8) settings may have a close equivalent multi-attribute version.

To this end, we look at the multivariate analog of equation (1.6): in a multivariate linear regression setup, it is a matter of straightforward algebra to see that the conditional distribution of $X_j \in \mathbb{R}^K$ on the other variables is

$$X_j \,|\, X_{\backslash j} = x \sim \mathcal{N}(-\Theta_{jj}^{-1}\Theta_{j\backslash j}x, \Theta_{jj}^{-1}) \ .$$

Equivalently, letting $\mathbf{B}_j^T = -\Theta_{jj}^{-1}\Theta_{j\backslash j}$, one has

$$X_j \,|\, X_{\backslash j} = \mathbf{B}_j^T X_{\backslash j} + \varepsilon_j \quad \varepsilon_j \sim \mathcal{N}(\mathbf{0}, \Theta_{jj}^{-1}), \quad \varepsilon_j \perp X,$$

where $\mathbf{B}_j \in \mathcal{M}_{(p-1)K,K}$ is defined block-wise

$$\mathbf{B}_j = \left[\begin{array}{c} \mathbf{B}_j^{(1)} \\ \hline \vdots \\ \hline \mathbf{B}_j^{(p-1)} \end{array}\right] = \Theta_{jj}^{-1} \times \left[\begin{array}{c} \Theta_j^{(1)} \\ \hline \vdots \\ \hline \Theta_j^{(p-1)} \end{array}\right],$$

and where each $\mathbf{B}_j^{(i)}$ is a $K \times K$ matrix which links attributes of variables (i, j). We see that recovering the support of $\mathbf{B}_j$ block-wise is equivalent to reconstructing the network defined in (1.10). Estimation of $\mathbf{B}_j$ is thus typically achieved through sparse methods. To this end, we consider an i.i.d. sample $\{X^\ell\}_{\ell=1}^n$ of X such that each attribute is observed n times for the p variable, each X^n being a pK-size row vector staked in a $\mathcal{M}_{n,pK}$ data matrix $\mathbf{X}$, so that $\mathbf{X}_j \in \mathcal{M}_{n,K}$ is a real-value, $n \times K$ block matrix containing the data related to the jth variable:

$$\mathbf{X} \quad = \quad \left[\begin{array}{c} \mathbf{x}^1 \\ \hline \vdots \\ \hline \mathbf{x}^N \end{array}\right] \quad = \quad \left[\mathbf{X}_1 \ldots \mathbf{X}^p\right] \quad = \quad \left[\begin{array}{cc|c|cc} X_1^{11} & X_1^{1K} & \ldots & X_1^{p1} \ \ldots & X_1^{pK} \\ \hline \vdots & \vdots & \ldots & & \\ \hline X_n^{11} & X_n^{1K} & \ldots & X_n^{1K} \ \ldots & X_n^{pK} \end{array}\right].$$

Using these notations, a direct generalization of the neighborhood selection is to predict for each $j = 1, \ldots, p$ the data block $\mathbf{X}_j$ by regressing on $\mathbf{X}_{\backslash j}$. In matrix form, this can be written as the optimization problem

$$\arg\min_{\mathbf{B}_j\in\mathbb{R}} J(\mathbf{B_j}), \qquad J(\mathbf{B}_j) = \frac{1}{2n}\left\|\mathbf{X}_j - \mathbf{X}_{\setminus j}\mathbf{B}_j\right\|_F^2 + \lambda\ \Omega(\mathbf{B}_j), \tag{1.11}$$

where $\|\mathbf{A}\|_F = \sqrt{\sum_{i,j}\mathbf{A}_{ij}^2}$ is the Frobenius norm of matrix $\mathbf{A}$ and Ω is a penalty which constrains $\mathbf{B}_j$ block-wise.

Choosing a penalizer.

Various choices for Ω in (1.11) seem relevant: by simply setting $\Omega_0(A) = \sum_{i,j}|A_{i,j}|$, we just encourage sparsity among the $\mathbf{B}_i$ and thus do not couple the attributes. A clever choice would be to activate a set of attributes all together: hence, the group is defined by all the K attributes between variables i and j, therefore the penalizer turns to a group-Lasso like penalty

$$\Omega_1(\mathbf{B}_j) = \sum_{i\in\mathscr{P}\setminus j}\|\mathbf{B}_j^{(i)}\|_F, \tag{1.12}$$

in which case convex analysis and subdifferential calculus (see **?**) can be used to show that a $\mathbf{B}_i$ is optimal for Problem (1.11) if and only if

$$\begin{cases} \forall i : \mathbf{B}_j^{(i)} \neq 0, & \left(\mathbf{S}_{ij} + \frac{\lambda}{\|\mathbf{B}_j^{(i)}\|_F} I\right)^{-1}\mathbf{S}_{ij} = \mathbf{B}_j^{(i)} \\ \forall i : \mathbf{B}_j^{(i)} = \mathbf{0}_{KK}, & \|\mathbf{S}_{ij}\|_F \leq \lambda \end{cases}, \tag{1.13}$$

where $\mathbf{S}_{ij} \in \mathscr{M}_{KK}$ is a $K\times K$ block in the empirical covariance matrix $\mathbf{S}_n = n^{-1}\mathbf{X}^\top\mathbf{X}$, which shows the same block-wise decomposition as Σ or Θ. This paves the way for an optimization algorithm like block-coordinate descent which we implemented, although we omit details here.

1.4 Numerical study

We propose a simple simulation to illustrate the interest of using multi-attribute networks and the efficiency of our proposal. The simulations are set up as follows:

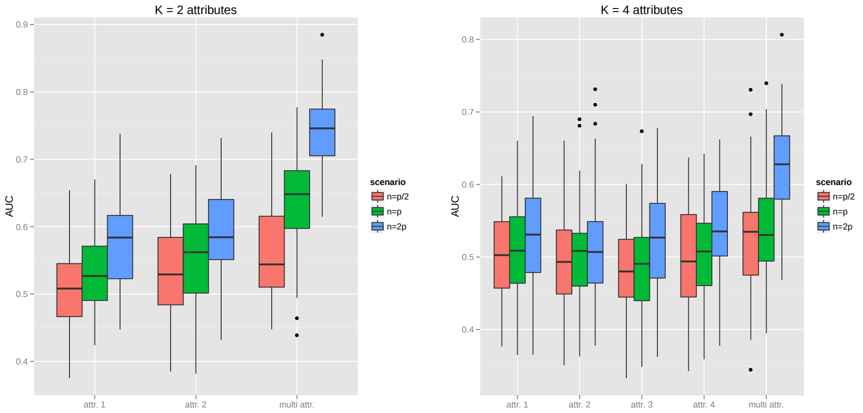
1. Draw a random undirected network with p nodes from the Erds-Renyi model;
2. Expand the associated adjacency matrix to multivariate space with

   $$\mathbf{A} = (\mathbf{A} + I)\otimes\mathbb{I}_{K\times K};$$

   where $\otimes$ is the Kronecker product.
3. Compute Θ a positive definite approximation of $\mathbf{A}$ by replacing null and negative eigenvalues by a small constant;
4. Control the difficulty of the problem with $\gamma > 0$ such that $\Theta = \Theta + \gamma I$;

5. Draw an i.i.d. sample $\mathbf{X}$ of $X \sim \mathcal{N}\left(0, \Theta^{-1}\right)$.

We choose small networks with $p = 20$, with 20 edges on average and vary n from $p/2$ to $2p$. We consider cases where the number of attributes is $K = 2$ or $K = 4$. We either apply the usual neighborhood selection procedure on each dimension separately, or its multi-attribute counterpart with group-like penalty (1.12) on the multivariate data. We compute the AUC for each method and replicate the experiment 50 times. On Figure 1.3, it is clear that aggregation improves upon single-attribute methods.



$K = 2$ $K = 4$

Fig. 1.2 Simple simulation study for the multi-attribute network inference problem: the multivariate procedure improves over the univariate procedures in every situation when networks are close for each attribute.

Illustration: Gene/Protein regulatory network inference.

As an illustration, we applied our sparse multi-attribute GGM approach to the NCI-60 cancer line data set. This data set consists in molecular profiles on a panel of 60 diverse human cancer cell lines. We use both protein and gene profiling experiments. For the former, we have samples for 92 antibodies from reverse-phase lysate arrays (RPLA); for the latter, expression is measured for 9,000 RNA with Human Genome U95 affymetrix. A *consensus set* composed of 91 protein and the corresponding gene profiles is retained for the $n = 60$ samples.

We infer a sparse GGM on each attribute (gene and protein), separately to start with, and then on its multi-attribute version. We do this on a large grid of the tuning parameter and thus have three families of networks indexed by their number of edges. Figure 1.4 demonstrates that our sparse multi-attribute method capture the

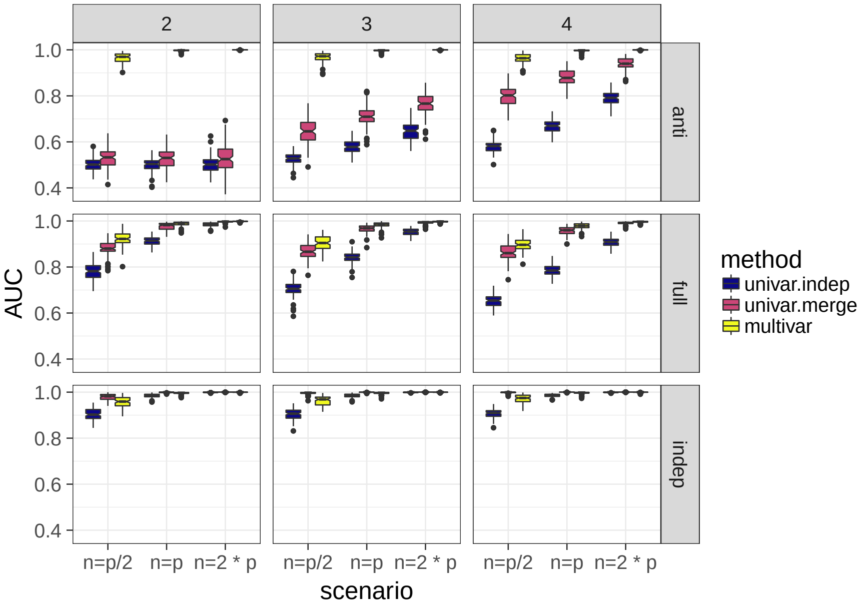


Fig. 1.3 New simu... discuss this

characteristics of both univariate networks, as the Jaccard similarity index is high between each uni-attribute network and the multi-attribute network, while it remains low when comparing uni-attribute networks together. This tends to prove that this multi-attribute version proposes a consensus version of the interactions at hand in the cell, and one which is hopefully more robust to noise and small misregulations.

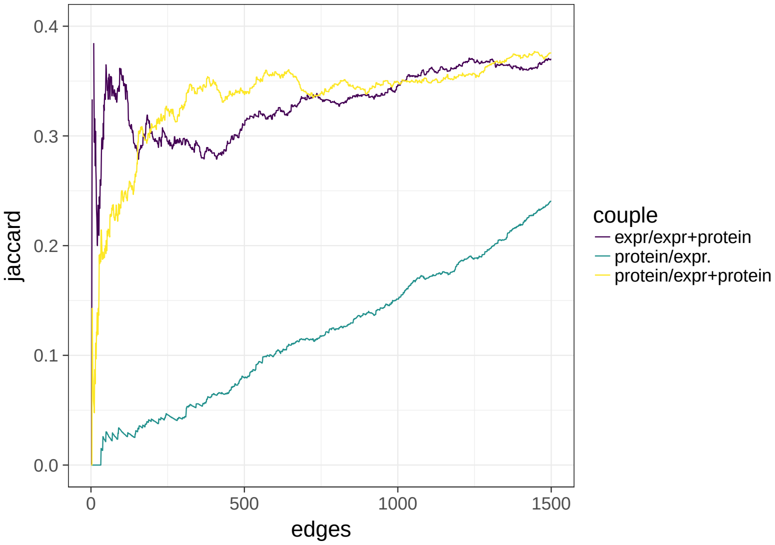


Fig. 1.4 Jaccard's similarity index $J(A,B) = \frac{|A \cap B|}{|A \cup B|}$: multi-attribute network shares a high Jaccard index with both uni-attribute networks.