Journal of Machine Learning Research 25 (2024) 1-41 Submitted 3/23; Revised 11/23; Published 4/24

Exploration of the Search Space of Gaussian Graphical Models for Paired Data

Alberto Roverato ALBERTO.ROVERATO@UNIPD.IT
Department of Statistical Sciences
University of Padova, Italy.

Dung Ngoc Nguyen NGOCDUNG.NGUYEN@CSIRO.AU
Department of Statistical Sciences
University of Padova, Italy.
CSIRO Agriculture and Food
Canberra, ACT, Australia.

Editor: Jin Tian

Abstract

We consider the problem of learning a Gaussian graphical model in the case where the observations come from two dependent groups sharing the same variables. We focus on a family of coloured Gaussian graphical models specifically suited for the paired data problem. Commonly, graphical models are ordered by the submodel relationship so that the search space is a lattice, called the model inclusion lattice. We introduce a novel order between models, named the twin order. We show that, embedded with this order, the model space is a lattice that, unlike the model inclusion lattice, is distributive. Furthermore, we provide the relevant rules for the computation of the neighbours of a model. The latter are more efficient than the same operations in the model inclusion lattice, and are then exploited to achieve a more efficient exploration of the search space. These results can be applied to improve the efficiency of both greedy and Bayesian model search procedures. Here, we implement a stepwise backward elimination procedure and evaluate its performance both on synthetic and real-world data.

Keywords: Brain network, coloured graphical model, lattice, partial order, principle of coherence, RCON model.

1. Introduction

A Gaussian graphical model (GGM) is a family of multivariate normal distributions whose conditional independence structure is represented by an undirected graph. The vertices of the graph correspond to the variables and every edge missing from the graph implies that the corresponding entry of the concentration matrix, that is the inverse of the covariance matrix, is equal to zero; see Lauritzen (1996). Gaussian graphical models are widely applied to the joint learning of multiple networks, where the observations come from two or more groups sharing the same variables. The association structure of each group is represented by a network and it is expected that there are similarities between the groups. In this framework, the literature has mostly focused on the case where the groups are independent so that every network is a distinct unit, disconnected from the other networks; see Tsai et al. (2022) for a recent review. We consider the case where groups cannot be assumed to be independent and, more specifically, we focus on the case of paired data, with exactly two dependent groups. The rest of this section is devoted to the presentation of the problem considered in this paper, whereas an overview of related works in this field, as well as of a description of some areas of application, are deferred to Section 4.

Coloured GGMs (Højsgaard and Lauritzen, 2008) are undirected graphical models with additional symmetry restrictions in the form of equality constraints on the parameters, which are then depicted on the dependence graph of the model by colouring of edges and vertices. Equality constraints allow one to disclose symmetries concerning both the structure of the network and the values of parameters associated with vertices and edges and, in addition, have the practical advantage of reducing the number of parameters. Roverato and Nguyen (2022) introduced a subfamily of coloured GGMs specifically designed to suit the paired data problem that they called RCON models for paired data (pdRCON). They approached the problem by considering a single coloured GGM comprising the variables of both the first and the second group. In this way, the resulting model has a graph for each of the two groups and the cross-graph dependence is explicitly represented by the edges across groups; see also Ranciati et al. (2021) and Ranciati and Roverato (2023).

Although the symmetry restrictions implied by a coloured GGM may usefully reduce the model dimension, the problem of model identification is much more challenging than with classical GGMs because both the dimensionality and the complexity of the search spaces highly increase; see Gao and Massam (2015); Massam et al. (2018); Li et al. (2020) and references therein. For the construction of efficient model selection methods, it is therefore imperative to understand the structure of model classes.

A statistical model is a family of probability distributions, and if a model is contained in another model then it is called a submodel of the latter. We can also say that a model is “larger” than any of its submodels, and model inclusion is typically used to embed a model class with a partial order. In this way, one can easily obtain that the family of GGMs forms a complete distributive lattice. Gehrmann (2011) considered four relevant subfamilies of coloured GGMs and showed that they constitute a lattice with respect to model inclusion. However, the structure of such lattices is rather complicated and this makes the identification of neighbouring models, and therefore the implementation of procedures for the exploration of model spaces, much less efficient than for classical GGMs. Furthermore, none of the existing lattices of coloured GGMs satisfies the distributivity property, which is a fundamental property that facilitates the implementation of efficient procedures and representation in lattices; see, among others, Habib et al. (2001) and Davey and Priestley (2002). Roverato and Nguyen (2022) showed that, under the model inclusion order, the class of pdRCON models identifies a proper complete sublattice of the lattice of coloured graphical models, although also in this case the distributive property is not satisfied.

We introduce a novel partial order for the class of pdRCON models that coincides with the model inclusion order if two models are model inclusion comparable but that also includes order relationships between certain models which are model inclusion incomparable. We show that the class of pdRCON models forms a complete lattice also with respect to this order, that we call the *twin lattice*. The twin lattice is distributive and its exploration is more efficient than that of the model inclusion lattice. Hence, the twin lattice can be used to improve the efficiency of procedures, either Bayesian and frequentist, which explore the model space moving between neighbouring models. More specifically, the focus of this paper is on stepwise greedy search procedures, and we show how the twin lattice can be exploited to improve efficiency in the identification of neighbouring submodels.

One way to increase the efficiency of greedy search procedures is by applying the, so-called, *principle of coherence* (Gabriel, 1969) that is used as a strategy for pruning the search space. We show that for the family of pdRCON models the twin lattice allows a more straightforward implementation of the principle of coherence.

We implement a stepwise backward elimination procedure with local moves on the twin lattice which satisfies the coherence principle, and we show that it is more efficient than an equivalent procedure on the model inclusion lattice. This procedure is implemented in the statistical programming language R and its behavior is investigated both on synthetic and real-world data.

The rest of the paper is organized as follows. Background on lattice theory, coloured graphical models and the structure of their model space is given in Section 2, whereas Section 3 introduces the family of RCON models for paired data. Section 4 contains an overview of the related literature and a discussion of some issues concerning the applications. In Section 5, we introduce the twin lattice, derive its properties and describe its relationships with the model inclusion lattice. Section 6 deals with the dimension of the search space and the implementation of the principle of coherence. The greedy search procedure is described in Section 7, and then its application to both synthetic and real-world data is presented in Section 8. Finally, Section 9 contains a brief discussion. Proofs are deferred to Section E of the Appendix.

2. Background and Notation

In this section, we review the elements and notation of lattice theory and of coloured graphical models, as required for this paper; for a more comprehensive account we refer to Davey and Priestley (2002), Lauritzen (1996) and Højsgaard and Lauritzen (2008).

2.1 Partial Orders and Lattices

A partially ordered set, or *poset*, $\langle A, \preceq \rangle$ is a structure where A is a set and $\preceq$ is a partial order on A. If the elements $a, b \in A$ are such that $a \preceq b$ with $a \neq b$, then we write $a \prec b$ and say that a is *smaller* than b. Furthermore, if it also holds that there is no element $c \in A$ such that $a \prec c \prec b$ then we say that a is *covered* by b and write $a \prec\!\!\!\cdot\, b$. If both $a \not\preceq b$ and $b \not\preceq a$ then a and b are *incomparable.*

An element $a \in A$ is called an *upper bound* of the subset $H \subseteq A$ if it has the property that $h \preceq a$ for all $h \in H$ and, furthermore, it is the *supremum* of H, denoted by $a = \sup H$, if every upper bound b of H satisfies $a \preceq b$. Accordingly, $a \in A$ is a *lower bound* of H if $a \preceq h$ for all $h \in H$, and it is the *infimum* of H, $a = \inf H$, if for every lower bound b of H it holds that $b \preceq a$. If for a poset $\langle A, \preceq \rangle$ the element $\sup A$ exists, then it is called the *maximum element*, or the *unit*, of A and denoted by $\hat{1}$. Dually, if $\inf A$ exists, then this element is called the *minimum element*, or the *zero*, of A and denoted by $\hat{0}$.

A poset $\langle A, \preceq \rangle$ is called a *lattice* if $\inf H$ and $\sup H$ exist for every finite nonempty subset H of A. A lattice is called *complete* if also both $\inf A$ and $\sup A$ exist. If $\langle A, \preceq \rangle$ is a lattice, then for every pair of elements $a, b \in A$, we write $a \wedge b$ for $\inf\{a, b\}$ and $a \vee b$ for $\sup\{a, b\}$

and refer to $\wedge$ as the *meet operation* and to $\vee$ as the *join operation*. A lattice A is called *distributive* if the operations of join and meet distribute over each other; formally, if for all $a, b, c \in A$, $a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$. Finally, it is useful to represent the structure of a lattice graphically by means of a *Hasse diagram* that is a graph with A as vertex set and where two elements $a, b \in A$ are joined by an undirected edge, with the vertex b appearing above a, whenever $a \prec b$; see Figures 3 and 4 for examples.

2.2 Graphical Models and Coloured Graphical Models

For a finite set $V = \{1, \ldots, p\}$ we let $Y = Y_V$ be a continuous random vector indexed by V and we denote by $\Sigma = (\sigma_{ij})_{i,j \in V}$ and $\Sigma^{-1} = \Theta = (\theta_{ij})_{i,j \in V}$ the covariance and concentration matrix of Y, respectively. An undirected graph with vertex set V is a pair $G = (V, E)$ where E is an edge set that is a set of unordered pairs of distinct vertices; formally $E \subseteq F_V$ with $F_V = \{(i, j) \mid i, j \in V \text{ and } i < j\}$. Thus, F_V is the edge set of the *complete* graph, that is the graph in which each pair of distinct vertices is connected by an edge. Note that, when it is not clear from the context which graph is under consideration, we will write V_G and E_G to denote the vertex set and edge set of the graph G.

We say that the concentration matrix Θ is *adapted* to the graph G if every missing edge of G corresponds to a zero entry in Θ; formally, $(i, j) \notin E$, with $i < j$, implies that $\theta_{ij} = \theta_{ji} = 0$. A *Gaussian graphical model* (GGM) with graph G is the family of Gaussian distributions whose concentration matrix is adapted to G; see Lauritzen (1996). We denote by $M \equiv M(V)$ the family of GGMs for Y_V and by $M(G) \in M$ the GGM represented by the graph $G = (V, E)$.

A *colouring* of $G = (V, E)$ is a pair $(\mathcal{V}, \mathcal{E})$ where $\mathcal{V} = \{V_1, \ldots, V_v\}$ is a partition of V into vertex colour classes and, similarly, $\mathcal{E} = \{E_1, \ldots, E_e\}$ is a partition of E into edge colour classes. Accordingly, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a *coloured graph*. Similarly to the notation used for uncoloured graphs, we may write $\mathcal{V}_\mathcal{G}$ and $\mathcal{E}_\mathcal{G}$ to denote the vertex and edge colour classes of $\mathcal{G}$, respectively. In the graphical representation, all the vertices belonging to a same colour class are depicted of the same colour, and similarly for edges. Furthermore, in order to make coloured graphs readable also in black and white printing, we put a common symbol next to every vertex or edge of the same colour. The only exception to this rule is for vertices and edges belonging to colour classes with a single element, called *atomic*, which are all depicted in black with no symbol next to them.

Højsgaard and Lauritzen (2008) introduced *coloured* GGMs, which are GGMs with additional restrictions on the parameter space. The models are represented by coloured graphs, where parameters that are associated with edges or vertices of the same colour are restricted to being identical. In this paper, we focus on the family of coloured GGMs called RCON models by Højsgaard and Lauritzen (2008), because they place equality Restrictions on the entries of the CONcentration matrix. More specifically, in the RCON model with coloured graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ every vertex colour class V_i, $i = 1, \ldots, v$, identifies a set of diagonal concentrations whose value is constrained to be equal, and similarly for edge colour classes which identify subsets of off-diagonal concentrations. We denote by $\mathcal{M} \equiv \mathcal{M}(V)$ the family of RCON models for Y_V and by $\mathcal{M}(\mathcal{G}) \in \mathcal{M}$ the RCON model represented by $\mathcal{G}$.

We close this section by noticing that for the families of graphical models we consider every model is uniquely represented by a coloured graph, and in the rest of this paper, with a slight abuse of notation, we will not make an explicit distinction between sets of models and sets of graphs, thereby equivalently writing, for example, $\mathcal{M}(\mathcal{G}) \in \mathcal{M}$ and $\mathcal{G} \in \mathcal{M}$.

2.3 Exploration of Model Spaces

The implementation of most model selection procedures requires the exploration of the space of candidate structures, i.e., of the model space. It is therefore fundamental to understand the structure of model spaces so as to improve efficiency and to identify suitable neighbouring relationships to be used in greedy search procedures. In the families of graphical models we consider, this is typically achieved by embedding the model space with the partial order defined through the *model inclusion*, i.e., the *submodel*, relationship thereby obtaining a lattice structure. In the rest of this section, we review the relevant properties of the model inclusion lattices of both undirected and coloured graphical models, and discuss their use in the exploration of model spaces.

Two lattices which become relevant in this context are the so-called *subset* and *partition* lattices. The power set $\mathcal{P}ow(A)$ of any finite set A is naturally embedded with the set inclusion order, $\subseteq$, and forms a complete distributive lattice. This is a well-known lattice structure whose meet and join operations can be efficiently computed because they are the set intersection $\cap$ and the set union $\cup$ operations, respectively (see Davey and Priestley, 2002). A partition of A is a collection of nonempty, pairwise disjoint, subsets of A whose union is A. The family of partitions $\mathcal{P}art(A)$ of a set A is typically embedded with the partial order where the partition P_1 is smaller than P_2 if P_1 is finer than P_2, that is if every set in P_2 can be expressed as a union of sets in P_1. In this way, one obtains the so-called partition lattice, and it is well-understood that distributivity does not hold for this lattice and, furthermore, the implementation of the meet and of the join operations is more involved, and considerably less efficient, than in the subset lattice; see Canfield (2001) and Pittel (2000).

Given two graphs G and H, with vertex set V, we write $M(H) \preceq_s M(G)$ to denote that $M(H)$ is a submodel of $M(G)$. For the family of GGMs, model inclusion coincides with the subset relationship between edge sets, so that for every pair of graphs, G and H, with vertex set V, it holds that $M(H) \preceq_s M(G)$ if and only if $E_H \subseteq E_G$. Hence, $\langle M, \preceq_s \rangle$ coincides with the subset lattice of F_V and, consequently, the meet and the join operations take an especially simple form because $M(G) \vee M(H)$ and $M(G) \wedge M(H)$ are the models represented by the graphs with edge sets $E_G \cup E_H$ and $E_G \cap E_H$, respectively. Furthermore, the lattice $\langle M, \preceq_s \rangle$ is complete and distributive.

Consider now the family $\mathcal{M}$ of RCON models for Y_V and let $\mathcal{G}$ and $\mathcal{H}$ be two coloured graphs with vertex set V. It was shown by Gehrmann (2011) that $\mathcal{M}(\mathcal{H})$ is a submodel of $\mathcal{M}(\mathcal{G})$, i.e., $\mathcal{M}(\mathcal{H}) \preceq_s \mathcal{M}(\mathcal{G})$, if and only if all of the three following conditions hold true,

(S1) the edge set of $\mathcal{H}$ is a subset of the edge set of $\mathcal{G}$;

(S2) every colour class in $\mathcal{V}_{\mathcal{H}}$ is a union of colour classes in $\mathcal{V}_{\mathcal{G}}$;

(S3) every colour class in $\mathcal{E}_{\mathcal{H}}$ is a union of colour classes in $\mathcal{E}_{\mathcal{G}}$;

and we remark that condition (S1) can be considered as redundant because, in fact, it is implied by (S3). Furthermore, Gehrmann (2011) showed that $\langle \mathcal{M}, \preceq_s \rangle$ is a complete lattice, although non-distributive. Colour classes are partitions of the vertex and edge set, respectively, and Gehrmann (2011, eqn. 9) provided explicit rules for the computation of the meet $\mathcal{M}(\mathcal{G}) \wedge \mathcal{M}(\mathcal{H})$ and the join $\mathcal{M}(\mathcal{G}) \vee \mathcal{M}(\mathcal{H})$, which follow from the same operations in the partition lattice and share an equivalent level of computational complexity. The exploration of the space of RCON models is thus much more challenging than the exploration of GGMs because, firstly, as shown by Gehrmann (2011) the dimension of $\langle \mathcal{M}, \preceq_s \rangle$ is much larger than that of $\langle M, \preceq_s \rangle$ and it grows super-exponentially with the number of variables and, secondly, the exploration of the space through neighbouring models, which requires the application of the meet and the join of models, is more computationally demanding than in $\langle M, \preceq_s \rangle$. Indeed, the existing procedures for learning coloured graphical models can deal with a small number of variables (Gehrmann, 2011) or impose substantial approximations (Li et al., 2018).

3. RCON Models for Paired Data

Roverato and Nguyen (2022) introduced pdRCON models, which is a subfamily of RCON models specifically designed to deal with paired data. In this section, we introduce the twin-pairing function that allows us to deal efficiently with the paired data setting, and use it to define the family of pdRCON models. We start with a small instance of paired data problem that will be used hereafter as a running example.

Example 1 (Frets' Heads) *Whittaker (1990, Section 8.4) fitted a GGM to the Frets' Heads data which consist of measurements of the head* Height *and head* Breadth *of the first and second adult sons in a sample of 25 families. Variables are therefore naturally split into the variables associated with the first son, Y_H and Y_B, and the second son, $Y_{H'}$ and $Y_{B'}$, thereby identifying the Left and Right groups depicted in Figure 1a, whereas the corresponding block partition of the concentration matrix Θ is given in Figure 2i. Note that, for the presentation of the results below it is convenient to use a numerical vertex set, that is $V = \{1, 2, 3, 4\}$; however, in order to improve readability, in this example we also add subscripts and write $L = \{1_H, 2_B\}$ and $R = \{3_{H'}, 4_{B'}\}$.*

In paired data problems, every variable in Y_V has a homologous, or *twin*, variable in Y_V, and the *twin-pairing* function τ, or twin function for short, associates every element $i \in V$ with its twin $\tau(i) \in V$, in such a way that if $\tau(i) = j$ then $\tau(j) = i$. We then naturally extend the use of the twin function to edges as well as to collections of vertices and edges. Formally, for $(i, j) \in F_V$ we let $\tau(i, j) = (\tau(i), \tau(j))$ whereas for $A \subseteq V$ we set $\tau(A) = \{\tau(i) \mid i \in A\}$ and if $E \subseteq F_V$ then $\tau(E) = \{\tau(i, j) \mid (i, j) \in E\}$, with the convention that $\tau(\emptyset) = \emptyset$. Note that it may happen that $\tau(i, j) = (\tau(i), \tau(j))$ is such that $\tau(i) > \tau(j)$ and in this case we will implicitly assume that the endpoints of such pairs are reversed, so as to obtain a proper edge belonging to F_V.

The twin function can be used to partition the set V into two sets, $L \cup R = V$, with $L \cap R = \emptyset$, such that $\tau(L) = R$. This partition is not unique, and the theory developed in this paper requires the identification of one of such partitions, which can be arbitrarily chosen. Nevertheless, there typically exists a natural partition where L is associated with the first group and R with the second group and in our examples we will, without loss of generality, always refer to such natural partition. Accordingly, to facilitate the interpretation of a

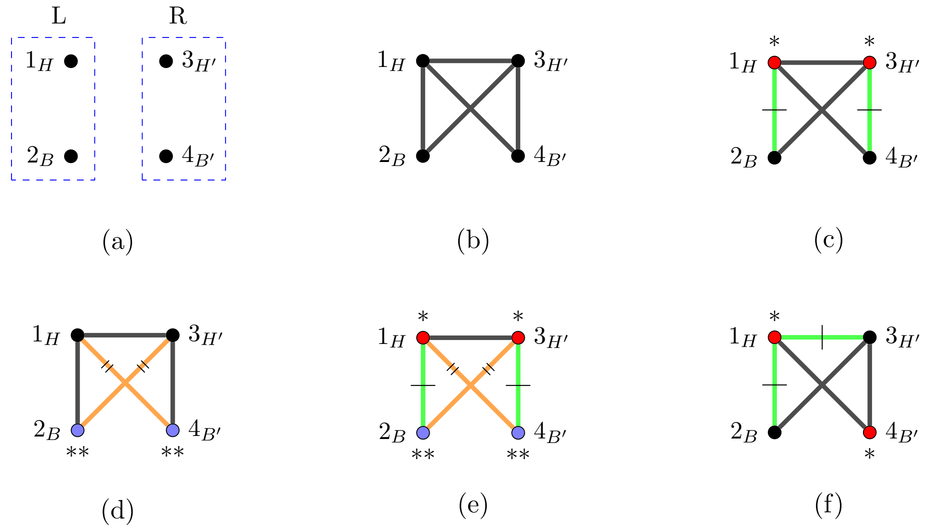


Figure 1: Frets' Heads example: (a) gives the partition of the vertex set into the two groups; the graphs (b) to (e) are pdCGs whereas (f) is a coloured graph but not a pdCG.

model, we will depict the coloured graph with vertices L on the Left and vertices R on the Right.

In this context, the focus is on similarities and differences between the two groups. Any vertex $i \in V$ and its twin $\tau(i)$ are associated with the diagonal entries θ_{ii} and $\theta_{\tau(i)\tau(i)}$ of Θ, respectively, and thus a hypothesis of interest is given by the equality restriction $\theta_{ii} = \theta_{\tau(i)\tau(i)}$. The latter is encoded by the vertex colour class $\{i, \tau(i)\}$, which we call *twin-pairing*. Similarly, the edge (i, j) and its twin $\tau(i, j)$ are associated with the off-diagonal concentrations θ_{ij} and $\theta_{\tau(i)\tau(j)}$, respectively, and the interest is for the equality $\theta_{ij} = \theta_{\tau(i)\tau(j)}$ implied by the edge twin-pairing colour class $\{(i, j), \tau(i, j)\}$. More formally, a twin-pairing colour class has cardinality two and contains either a pair of twin vertices $\{i, \tau(i)\}$ or a pair of twin edges $\{(i, j), \tau(i, j)\}$, with $i \neq \tau(j)$ so that $(i, j) \neq \tau(i, j)$.

$$\begin{pmatrix} \Theta_{LL} & \Theta_{LR} \\ \Theta_{RL} & \Theta_{RR} \end{pmatrix} \qquad \left[\begin{array}{cc|cc} \boldsymbol{\alpha} & \boldsymbol{\delta} & \theta_{13} & \theta_{14} \\ & \theta_{22} & \theta_{23} & 0 \\ \hline & & \boldsymbol{\alpha} & \boldsymbol{\delta} \\ & & & \theta_{44} \end{array}\right] \qquad \left[\begin{array}{cc|cc} \theta_{11} & \theta_{12} & \theta_{13} & \boldsymbol{\gamma} \\ & \boldsymbol{\beta} & \boldsymbol{\gamma} & 0 \\ \hline & & \theta_{33} & \theta_{34} \\ & & & \boldsymbol{\beta} \end{array}\right] \qquad \left[\begin{array}{cc|cc} \boldsymbol{\alpha} & \boldsymbol{\delta} & \theta_{13} & \boldsymbol{\gamma} \\ & \boldsymbol{\beta} & \boldsymbol{\gamma} & 0 \\ \hline & & \boldsymbol{\alpha} & \boldsymbol{\delta} \\ & & & \boldsymbol{\beta} \end{array}\right]$$

(i) (ii) (iii) (iv)

Figure 2: Frets' Heads example: (i) shows the block partition of the concentration matrix, whereas (ii) to (iv) give the concentration matrices for the three pdCGs of Figure 1, with vertical and horizontal lines to highlight the four block partition. More specifically, matrix (ii) corresponds to graph (c); matrix (iii) to graph (d) and matrix (iv) to graph (e).

Definition 1 (Roverato and Nguyen, 2022) *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a coloured graph with vertex set V and let τ be a twin-pairing function on V. We say that $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a* coloured graph for paired data *(pdCG) if every colour class is either atomic or twin-pairing. If an RCON model is defined by a pdCG then we say that it is an* RCON model for paired data *(pdRCON) and denote it by $\mathcal{P}(\mathcal{G})$. Moreover, we will denote by $\mathcal{P} = \mathcal{P}(V)$ the family of pdRCON models for Y_V.*

The family of pdRCON models $\mathcal{P}$ is a subfamily of RCON models $\mathcal{M}$, i.e., $\mathcal{P} \subseteq \mathcal{M}$, and therefore it makes sense to embed this space with the model inclusion order $\preceq_s$. Roverato and Nguyen (2022) showed that $\langle \mathcal{P}, \preceq_s \rangle$ is a proper sublattice of $\langle \mathcal{M}, \preceq_s \rangle$; however, as well as $\langle \mathcal{M}, \preceq_s \rangle$ also $\langle \mathcal{P}, \preceq_s \rangle$ is non-distributive.

For the interpretation of a pdRCON model, it is useful to distinguish between two different types of edge pairings, which we now describe for the case where L and R correspond to the natural group partition of V. Indeed, in this case the comparison of the association structure within the first group with that within the second group is carried out considering the pairs $i < j$ such that $i, j \in L$, because in this case $\tau(i), \tau(j) \in R$. Hence, the simultaneous absence or presence of the edges (i, j) and $\tau(i, j)$ identifies a similarity in the internal association structure of the two groups. Moreover, if (i, j) and $\tau(i, j)$ are both present, then it may further hold true that $\theta_{ij} = \theta_{\tau(i)\tau(j)}$, which is a constraint encoded by the twin-pairing colour class $\{(i, j), \tau(i, j)\}$. On the other hand, the analysis of the across-group relationships involves the vertices $i, j \in V$ with $i \in L$, $j \in R$ and $i \neq \tau(j)$, and also in this case the simultaneous absence or presence of both (i, j) and $\tau(i, j)$ identifies a similarity in the association structure between the across-group variables Y_i and Y_j and that of their across-group twins $Y_{\tau(i)}$ and $Y_{\tau(j)}$ with, potentially, $\theta_{ij} = \theta_{\tau(i)\tau(j)}$. We also note that symmetries involving the internal structure of the two groups correspond to similarities between the diagonal blocks Θ_{LL} and Θ_{RR} of Θ, whereas every across-group symmetry corresponds to a similarity between two entries of Θ belonging to the upper and lower triangular part, respectively, of the off-diagonal block Θ_{LR}.

Example 2 (Frets' Heads continued) *The five graphs of Figure 1 have all the same edge set and thus differ only for the colour classes. The graph in Figure 1b has only atomic colour classes and represents an uncoloured GGM or, equivalently, a trivial pdRCON model with no equality restrictions. The graph in Figure 1c is associated with the concentration matrix in Figure 2ii, and it represents a pdRCON model with one vertex twin-pairing colour class $\{1_H, 3_{H'}\}$, and one edge twin-pairing colour class $\{(1_H, 2_B), (3_{H'}, 4_{B'})\}$, the latter involving one edge within the first group and one edge within the second group. The graph in Figure 1d is associated with the concentration matrix in Figure 2iii, and it represents a pdRCON model with one vertex twin-pairing colour class $\{2_B, 4_{B'}\}$, and one edge twin-pairing colour class $\{(1_H, 4_{B'}), (2_B, 3_{H'})\}$, the latter involving edges across groups. The graph in Figure 1e is associated with the concentration matrix in Figure 2iv, and it represents a pdRCON model with all possible twin-paring classes and, finally, the graph in Figure 1f is not a pdCG because neither the vertex colour class $\{1_H, 4_{B'}\}$ nor the edge colour class $\{(1_H, 3_{H'}), (1_H, 2_B)\}$ are twin-pairing.*

4. Related Works and Application Issues

The problem of comparing the distribution of a set of variables between two experimental conditions, or groups, is common to many statistical applications. In this context, paired data commonly arise in paired design studies, where each subject is measured twice at two different time points or under two different situations, as well as from matched observational studies. For instance, in biomedical studies paired designs are frequently used because they make it possible to account for individual-specific effects, and consequently to reduce the variability in the observations due to differences between individuals. This also extends to high-throughput experimental techniques, with a relevant example provided by cancer genomics where control samples are often obtained from histologically normal tissues adjacent to the tumor (NAT) (Hardcastle and Kelly, 2013; Aran et al., 2017).

When the interest is for the association structure represented by a GGM, then the analysis of paired data amounts to the joint learning of the structure of a graph for each of the two groups, by accounting for the cross-graph association structure. We approach this problem by considering the family of pdRCON models, introduced by Roverato and Nguyen (2022). More specifically, Roverato and Nguyen (2022) showed that pdRCON models form a proper sublattice of the lattice of coloured graphical models, gave rules for the computation of the meet and joint operations, and implemented such rules in a stepwise model search procedure. Hence, Roverato and Nguyen (2022) considered the results provided by Gehrmann (2011) for RCON models, and derived the specific form they take when restricted to the subset of pdRCON models. In this way, some improvement in terms of efficiency could be achieved but, nevertheless, the pdRCON model inclusion sublattice still behaves like a partition lattice, and shares the inefficiencies of the latter. In the next section, we introduce a lattice for the family of pdRCON models that behaves like the subset lattice, and this allows us to exploit the properties of the latter in the exploration of the model space, as well as to gain insight into the structure of this model class.

One way to avoid the explicit exploration of the model space is by using penalized likelihood methods, which can be applied to problems of larger dimensions. Ranciati and Roverato (2023), elaborating on previous work by Ranciati et al. (2021), introduced a graphical lasso method for learning pdRCON models; see also Li et al. (2021) and Wit and Abbruzzo (2015) for applications of penalized likelihood methods to coloured graphical modelling. The applications considered in Ranciati et al. (2021) and Ranciati and Roverato (2023) concern the identification of brain networks from fMRI data and of gene networks from breast cancer gene expression data, respectively, which are contexts where variables are measured on the same scale. Indeed, the scale the variables are measured plays a relevant role in the procedures for learning pdRCON models. Højsgaard and Lauritzen (2008, Section 8) remarked that the comparison of concentration values is meaningful only when variables are measured on comparable scales, and they recommended that RCON models should be used only in this case. We note, however, that a less stringent condition is required in pdRCON models because, the fact that only twin-pairing colour classes are allowed, implies that only homologous variables need to have comparable scales, that is a condition easily satisfied in practice. Hence, pdRCON models can be meaningfully applied also when the scales of non-homologous variables are not comparable. However, in this case the application of graphical lasso methods is problematic. Indeed, on the one hand the result of the graphical lasso is not invariant to scalar multiplications of the variables and, for this reason, it is common practice to apply it to standardized data; see, among others, Hastie et al. (2015, p. 8) and Carter et al. (2024). On the other hand, as noticed by Højsgaard and Lauritzen (2008, Section 3.4), RCON models are not invariant under rescaling, in the sense that standardization will not preserve the original structure of colour classes. In Section 8, we present two applications: the first to the same fMRI data previously analysed by Ranciati et al. (2021) and Roverato and Nguyen (2022), and the second to a dataset on air quality data where non-homologous variables have different scales, and a comparison with the graphical lasso procedure of Ranciati and Roverato (2023) is carried out.

An alternative approach to the joint learning of dependent GGMs was introduced by Xie et al. (2016) and Zhang et al. (2022) where multiple groups are considered and the cross-graph dependence is modelled by means of a latent vector representing systemic variation manifesting simultaneously in all groups. The latter methods have the advantage that they can deal with an arbitrary number of dependent groups but, on the other hand, they do not provide an explicit representation of the association structure between groups. Wit and Abbruzzo (2015) introduced factorial GGMs, which are suited for dealing with longitudinal multivariate data observed at T time points; see also Vinciotti et al. (2016). In factorial GGMs the concentration matrix is partitioned into blocks in such a way that each of the T diagonal blocks gives the graph structure at a different time point, whereas the off-diagonal blocks encode the cross-graphs dependence structure. More specifically, factorial GGMs are RCON models because they also allow to include equality constraints between entries of the concentration matrix and, thus, pdRCON models could be regarded as a subfamily of factorial GGMs. However, the available theory of factorial GGMs does not include methods for learning the colour classes, which are assumed to be known.

5. An Alternative Lattice Structure of $\mathcal{P}$

We now introduce a novel lattice structure for the family of pdRCON models that is based on an alternative representation of pdCGs.

5.1 An Alternative Characterization of Coloured Graphs for Paired Data

The classical representation of coloured graphs is through their colour classes. In this section we introduce a different representation of pdCGs that will allow us to deal more efficiently with these objects. A detailed example that may help to follow the steps required for the construction of the novel representation is given in Section A of the Appendix.

If (L, R) is a partition induced by the twin function τ we can, without loss of generality, number the variables so that $L = \{1, \dots, q\}$ and $R = \{q+1, \dots, p\}$. The latter numbering allows us to identify the following three subsets of F_V,

$$F_L = \{(i,j) \in F_V \mid i < \tau(j)\},\ F_R = \{(i,j) \in F_V \mid i > \tau(j)\},\ F_T = \{(i,j) \in F_V \mid i = \tau(j)\}.$$

It is straightforward to see that the triplet (F_L, F_R, F_T) forms a partition of F_V with the property that $F_R = \tau(F_L)$, $F_L = \tau(F_R)$ and $F_T = \tau(F_T)$. The subscripts we use follow from the fact that, for every $(i,j) \in F_V$, it holds that if $i, j \in L$ then $(i,j) \in F_L$ whereas if $i, j \in R$ then $(i, j) \in F_R$. We remark, however, that both F_L and F_R contain edges (i, j) such that $i \in L$ and $j \in R$. More specifically, the partition of the cross-graph edges between F_L and F_R is not unique, because any partition with the property that if $(i, j) \in F_L$ then $\tau(i, j) \in F_R$ can be equivalently used. Indeed, the partition implied by τ merely depends on the variable numbering. Finally, the subscript of F_T recalls that every edge in this set links a vertex to its twin.

For a coloured graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, we set

$$V_{\mathcal{G}} = \cup_{j=1}^{v} V_j \qquad \text{and} \qquad E_{\mathcal{G}} = \cup_{j=1}^{e} E_j, \tag{1}$$

and call $(V_{\mathcal{G}}, E_{\mathcal{G}})$ the *uncoloured version* of $\mathcal{G}$. Hence, if $G = (V, E)$ is the uncoloured version of $\mathcal{G}$ then the above partition of F_V naturally induces a partition (E_L, E_R, E_T) of E where $E_L = E \cap F_L$, $E_R = E \cap F_R$ and $E_T = E \cap F_T$. Then, we introduce the following two sets which can be associated to a pdCG $\mathcal{G} = (\mathcal{V}, \mathcal{E})$,

$$\mathbb{L} = \{i \in L \mid \{i\} \in \mathcal{V}\} \quad \text{and} \quad \mathbb{E} = \{(i, j) \in E_L \cap \tau(E_R) \mid \{(i, j)\} \in \mathcal{E}\}, \tag{2}$$

so that $\mathbb{L} \subseteq L$ is made up of the vertices in L which belong to an atomic colour class, whereas an edge (i, j) belongs to $\mathbb{E}$ if and only if (i) (i, j) belongs to E_L, and (ii) its twin $\tau(i, j)$ is present in the graph and (iii) both (i, j) and $\tau(i, j)$ form atomic colour classes. Hence, through equations (1) and (2), we can associate to any pdCG $\mathcal{G}$ two sets of vertices, that is V and $\mathbb{L}$, and two sets of edges, that is E and $\mathbb{E}$.

Example 3 (Frets' Heads continued) *All the pdCGs of Figure 1 have common vertex set $V = \{1_H, 2_B, 3_{H'}, 4_{B'}\}$ and edge set $E = \{(1_H, 2_B), (1_H, 3_{H'}), (1_H, 4_{B'}), (2_B, 3_{H'}), (3_{H'}, 4_{B'})\}$. The graph in Figure 1b has no twin-pairing colour classes so that $\mathbb{L} = L = \{1_H, 2_B\}$ and $\mathbb{E} = E_L \cap \tau(E_R) = \{(1_H, 2_B), (1_H, 4_{B'})\}$. For the graph in Figure 1c one has $\mathbb{L} = \{2_B\}$ and $\mathbb{E} = \{(1_H, 4_{B'})\}$ whereas the graph Figure 1d is such that $\mathbb{L} = \{1_H\}$ and $\mathbb{E} = \{(1_H, 2_B)\}$. Finally, the graph in Figure 1e has all possible twin-pairing colour classes and therefore $\mathbb{L} = \mathbb{E} = \emptyset$.*

We will show below that every pdCG can be uniquely represented by its uncoloured version, together with the subset of vertices $\mathbb{L}$ that characterizes the vertex colour classes, and the subset of edges $\mathbb{E}$ that characterizes the edge colour classes. Hence, the quadruplet $(V, E, \mathbb{L}, \mathbb{E})$ provides an alternative representation of pdCGs. Firstly, we show that from the four sets $(V, E, \mathbb{L}, \mathbb{E})$ it is possible to recover the coloured graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ from where they were computed.

Proposition 2 *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a pdCG and let $(V, E, \mathbb{L}, \mathbb{E})$ the quadruplet obtained from the application of (1) and (2) to $\mathcal{G}$ with respect to a partition (L, R) induced by the twin function τ. Then, the colour classes $(\mathcal{V}, \mathcal{E})$ of $\mathcal{G}$ can be recovered from the quadruplet as follows,*

(i) the twin-pairing colour classes in $\mathcal{V}$ are the sets $\{i, \tau(i)\}$ for all $i \in L \setminus \mathbb{L}$, and the remaining vertices in V form vertex atomic classes;

(ii) the twin-pairing colour classes in $\mathcal{E}$ are the sets $\{(i, j), \tau(i, j)\}$ for all $(i, j) \in (E_L \cap \tau(E_R)) \setminus \mathbb{E}$ and the remaining edges in E form edge atomic classes.

Proof See Section E.1 of the Appendix. ■

The quadruplet $(V, E, \mathbb{L}, \mathbb{E})$ characterizes a pdCG by means of an uncoloured undirected graph together with a subset of its vertices, $\mathbb{L}$, and a subset of its edges, $\mathbb{E}$. However, not all quadruplets of this type can be used to represent a pdCG and we need the following.

Definition 3 *Let $G = (V, E)$ be an undirected graph and τ a twin function on V. We say that the collection of sets $(V, E, \mathbb{L}, \mathbb{E})$ is* compatible *with the partition (L, R) induced by τ if $\mathbb{L} \subseteq L$ and $\mathbb{E} \subseteq E_L \cap \tau(E_R)$.*

And we can now give the main result of this section.

Theorem 4 *Let τ a twin function on V and (L, R) a partition of V induced by τ. Then, equations (1) and (2) establish a one-to-one correspondence between the family of pdCGs $\mathcal{P}$ and the collection of quadruplets $(V, E, \mathbb{L}, \mathbb{E})$ compatible with (L, R).*

Proof See Section E.2 of the Appendix. ■

We have thus shown that the family of compatible quadruplets $(V, E, \mathbb{L}, \mathbb{E})$ provides a representation of $\mathcal{P}$ alternative to colour classes. In fact, such alternative representation may look less intuitive than the traditional representation. However, it is easier to implement in computer programs and, as shown below, this representation naturally leads to the definition of a useful alternative lattice structure of $\mathcal{P}$.

5.2 The Twin Lattice

In this section, we introduce a novel partial order for pdCGs and show that, embedded with this order, the set $\mathcal{P}$ forms a distributive lattice that we call the twin lattice. Unlike the model inclusion lattice that inherits its properties from the partition lattice, the twin lattice behaves in a way similar to the subset lattice and therefore it satisfies the distributivity property, and the join and meet operations can be efficiently computed as union and intersection of sets, respectively.

For the theory developed in the following, we need to identify a partition (L, R) of V that is induced by τ, but otherwise arbitrary. Furthermore, a pdCG $\mathcal{G} \in \mathcal{P}$ will be equivalently represented by using its colour classes, $(\mathcal{V}, \mathcal{E})$ or its quadruplet $(V, E, \mathbb{L}, \mathbb{E})$. We can now introduce the novel twin order.

Definition 5 *Let $\mathcal{H} = (V, E_{\mathcal{H}}, \mathbb{L}_{\mathcal{H}}, \mathbb{E}_{\mathcal{H}})$ and $\mathcal{G} = (V, E_{\mathcal{G}}, \mathbb{L}_{\mathcal{G}}, \mathbb{E}_{\mathcal{G}})$ be two pdCGs in $\mathcal{P}$. Then we say that $\mathcal{H} \preceq_t \mathcal{G}$ if and only if*

(T1) $E_{\mathcal{H}} \subseteq E_{\mathcal{G}}$, (T2) $\mathbb{L}_{\mathcal{H}} \subseteq \mathbb{L}_{\mathcal{G}}$, (T3) $\mathbb{E}_{\mathcal{H}} \subseteq \mathbb{E}_{\mathcal{G}}$.

and call $\preceq_t$ the twin order.

We can thus see that the twin order is naturally associated with the alternative representation of pdCGs introduced in the previous section, and that, when using that representation, it is straightforward to check whether the order relationship holds for two graphs. Because the twin order generalizes the subset order, also the associated lattice turns out to be a natural extension of the subset lattice and to share its properties.

Theorem 6 *The family of pdCGs on the vertex set V, equipped with the twin order, that is $\langle \mathcal{P}, \preceq_t \rangle$, forms a complete distributive lattice, that we call the* twin lattice*, where if $\mathcal{G}, \mathcal{H} \in \mathcal{P}$,*

(i) the meet of $\mathcal{G}$ and $\mathcal{H}$ can be computed as

$$\mathcal{G} \wedge_t \mathcal{H} = (V,\ E_{\mathcal{G}} \cap E_{\mathcal{H}},\ \mathbb{L}_{\mathcal{G}} \cap \mathbb{L}_{\mathcal{H}},\ \mathbb{E}_{\mathcal{G}} \cap \mathbb{E}_{\mathcal{H}})\,;$$

(ii) the join of $\mathcal{G}$ and $\mathcal{H}$ can be computed as

$$\mathcal{G} \vee_t \mathcal{H} = (V,\ E_{\mathcal{G}} \cup E_{\mathcal{H}},\ \mathbb{L}_{\mathcal{G}} \cup \mathbb{L}_{\mathcal{H}},\ \mathbb{E}_{\mathcal{G}} \cup \mathbb{E}_{\mathcal{H}})\,.$$

Furthermore, the unit is $\hat{1} = (V,\ F_V,\ L,\ F_L)$, that is the uncoloured complete graph, whereas the zero is $\hat{0} = (V,\ \emptyset,\ \emptyset,\ \emptyset)$, that is the graph with no edge and such that all vertices belong to twin-pairing classes.

Proof See Section E.3 of the Appendix. ■

5.3 Application of the Twin Lattice in Model Search

In order to exploit the properties of the twin lattice in model search it is useful to investigate the existing relationships between the twin lattice and the model inclusion lattice. We first show that the twin order $\preceq_t$ is a refinement of the model inclusion order $\preceq_s$.

Proposition 7 *For any $\mathcal{H}, \mathcal{G} \in \mathcal{P}$, the following relationships between the model inclusion order and the twin order exist,*

(i) if $\mathcal{H} \preceq_s \mathcal{G}$ then $\mathcal{H} \preceq_t \mathcal{G}$

(ii) if $\mathcal{H} \preceq_s \mathcal{G}$ and $\mathcal{H} \prec_t \mathcal{G}$ then $\mathcal{H} \prec_s \mathcal{G}$.

Proof See Section E.4 of the Appendix. ■

We remark that the reverse of (i) in Proposition 7 does not hold true, and it is not difficult to find two pdCGs, $\mathcal{G}$ and $\mathcal{H}$, such that $\mathcal{G} \preceq_t \mathcal{H}$ but that are model inclusion incomparable.

For the exploration of the model space it is useful to be able to identify the neighbours of a model $\mathcal{P}(\mathcal{G})$. Here we focus on stepwise backward elimination procedures and therefore on the neighbouring submodels of $\mathcal{P}(\mathcal{G})$, which are the models $\mathcal{P}(\mathcal{H}) \subset \mathcal{P}(\mathcal{G})$ such that there is no graph $\mathcal{F} \in \mathcal{P}$ with $\mathcal{P}(\mathcal{H}) \subset \mathcal{P}(\mathcal{F}) \subset \mathcal{P}(\mathcal{G})$ or, equivalently, the models $\mathcal{P}(\mathcal{H})$ such as $\mathcal{H}$ is covered by $\mathcal{G}$ in the model inclusion lattice, $\mathcal{H} \prec_s \mathcal{G}$.

Proposition 8 *Let $\mathcal{P}(\mathcal{G})$ be the pdRCON model represented by the graph $\mathcal{G} = (V, E_{\mathcal{G}}, \mathbb{L}_{\mathcal{G}}, \mathbb{E}_{\mathcal{G}})$. Then the set of pdCGs representing the neighbouring submodels of $\mathcal{P}(\mathcal{G})$, that is the subset of graphs $\mathcal{H} \in \mathcal{P}$ such that $\mathcal{H} \prec_s \mathcal{G}$, is made up of,*

(a) all the graphs obtained by merging exactly two vertex atomic colour classes of $\mathcal{G}$ to obtain a vertex twin-pairing colour class, which can be formally computed as

(i) $\mathcal{H} = (V, E_{\mathcal{G}}, \mathbb{L}_{\mathcal{G}} \setminus \{i\}, \mathbb{E}_{\mathcal{G}})$ *for all $i \in \mathbb{L}_{\mathcal{G}}$;*

(b) all the graphs obtained by merging exactly two edge atomic colour classes of $\mathcal{G}$ to obtain an edge twin-pairing colour class, which can be formally computed as

(ii) $\mathcal{H} = (V, E_{\mathcal{G}}, \mathbb{L}_{\mathcal{G}}, \mathbb{E}_{\mathcal{G}} \setminus \{(i,j)\})$ *for all* $(i,j) \in \mathbb{E}_{\mathcal{G}}$*;*

(c) all the graphs obtained by removing exactly one edge atomic colour class from $\mathcal{G}$, which can be formally computed as

(iii) $\mathcal{H} = (V, E_{\mathcal{G}} \setminus (i,j), \mathbb{L}_{\mathcal{G}}, \mathbb{E}_{\mathcal{G}} \setminus \{(i,j)\})$ *for all* $(i,j) \in \mathbb{E}_{\mathcal{G}}$*;*
(iv) $\mathcal{H} = (V, E_{\mathcal{G}} \setminus \tau(i,j), \mathbb{L}_{\mathcal{G}}, \mathbb{E}_{\mathcal{G}} \setminus \{(i,j)\})$ *for all* $(i,j) \in \mathbb{E}_{\mathcal{G}}$*;*
(v) $\mathcal{H} = (V, E_{\mathcal{G}} \setminus (i,j), \mathbb{L}_{\mathcal{G}}, \mathbb{E}_{\mathcal{G}})$ *for all* $(i,j) \in E_{\mathcal{G}}$ *such that* $\tau(i,j) \notin E_{\mathcal{G}}$*;*
(vi) $\mathcal{H} = (V, E_{\mathcal{G}} \setminus \{(i,\tau(i))\}, \mathbb{L}_{\mathcal{G}}, \mathbb{E}_{\mathcal{G}})$ *for all* $i \in V$ *such that* $(i,\tau(i)) \in E_{\mathcal{G}}$*;*

(d) all the graphs obtained by removing exactly one edge twin-pairing colour class from $\mathcal{G}$, which can be formally computed as

(vii) $\mathcal{H} = (V, E_{\mathcal{G}} \setminus \{(i,j), \tau(i,j)\}, \mathbb{L}_{\mathcal{G}}, \mathbb{E}_{\mathcal{G}})$ *for all* $(i,j), \tau(i,j) \in E_{\mathcal{G}}$ *such that*
$(i,j) \neq \tau(i,j)$*and both* $(i,j) \notin \mathbb{E}_{\mathcal{G}}$
and $\tau(i,j) \notin \mathbb{E}_{\mathcal{G}}$*.*

Proof See Section E.5 of the Appendix. ■

Example 4 (Frets' Heads continued) *Figure 3 gives the structure of a portion of the Hasse diagram of the model inclusion lattice of pdRCON models in the Frets' heads example. The graph on the top represents the saturated model, and it is linked to its neighbouring submodels, represented by the coloured graphs in the highlighted area. If we number the graphs in the highlighted area from left to right, then graphs 1 and 2 are obtained from the graph of the saturated model by applying (i) of Proposition 8, graphs 5 and 8 from point (ii), graphs 6, 7, 9 and 10 from points (iii) and (iv) and, finally, graphs 3 and 4 from point (vi). Points (v) and (vii) cannot be applied to the saturated model but they can be applied to obtain the two graphs depicted in the lowest level. More specifically, the two graphs can be obtained both by applying (v) to the graphs 6, 7, 9 and 10, and by applying (vii) to the graphs 5 and 8.*

The usefulness of this proposition stands on the fact that it provides a way to construct all of the neighbouring submodels of a given pdRCON model $\mathcal{P}(\mathcal{G})$ and, furthermore, because it is stated by using the alternative representation of pdCGs, it allows us to carry out a comparison between the model inclusion and the twin lattice. It is obvious, by construction, that any two neighbouring submodels of $\mathcal{P}(\mathcal{G})$ are $\preceq_s$–incomparable. However, they are not typically $\preceq_t$–incomparable, as shown as follows.

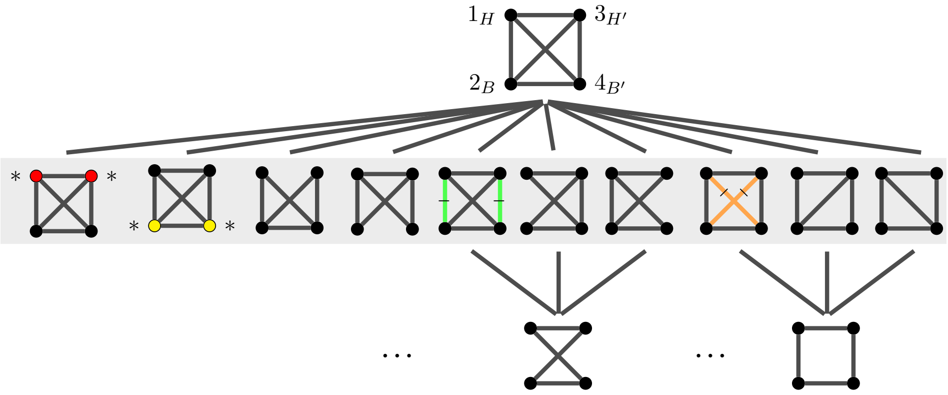


Figure 3: Part of the Hasse diagram of the model inclusion lattice for the Frets' heads example. The graph representing the saturated model is depicted on the top and followed, below, by the graphs representing its neighbouring submodels, which are in the highlighted area. Then, two graphs of the third level are given.

Corollary 9 *Let $\mathcal{H}_1$, $\mathcal{H}_2$ and $\mathcal{G}$ be three pdCGs such that $\mathcal{H}_1, \mathcal{H}_2 \prec_s \mathcal{G}$ or, equivalently, such that $\mathcal{P}(\mathcal{H}_1)$ and $\mathcal{P}(\mathcal{H}_2)$ are two neighbouring submodels of $\mathcal{P}(\mathcal{G})$. Then $\mathcal{H}_1 \preceq_t \mathcal{H}_2$ if and only if, for a given edge $(i,j) \in \mathbb{E}_{\mathcal{G}}$, $\mathcal{H}_2$ is obtained from (ii) of Proposition 8 and $\mathcal{H}_1$ is obtained from either (iii) or (iv) of the same proposition. More specifically, in the latter case it holds that $\mathcal{H}_1 \prec_t \mathcal{H}_2$ whereas, in all other cases, $\mathcal{H}_1$ and $\mathcal{H}_2$ are $\preceq_t$–incomparable.*

Proof This result follows immediately from the application of the twin order in Definition 5 to the comparison of all the pairs of neighbouring submodels of $\mathcal{G}$ as given in Proposition 8. ■

In the graphical representation of the model inclusion lattice provided by the Hasse diagram every model is depicted above its submodels, and Proposition 7 shows that this is also true in the Hasse diagram of the twin lattice. Furthermore, in the Hasse diagram of the model inclusion lattice every model is linked by an edge to each of its neighbouring submodels, which are pairwise $\preceq_s$–incomparable. Theorem 8 and Corollary 9 show that the twin order $\preceq_t$ can be use to partition the set of neighbouring submodels of any model $\mathcal{P}(\mathcal{G})$ into two subsets, that we refer to as the *upper* and *lower-layer* neighbouring submodels of $\mathcal{P}(\mathcal{G})$. The upper-layer neighbouring submodels are obtained by applying to $\mathcal{G}$ the points (i), (ii), (v), (vi) and (vii) of Proposition 8, whereas the lower-layer submodels come from points (iii) and (iv). The names upper and lower layer are suggested by the fact that every graph in the lower layer is $\preceq_t$–smaller than some graph in the upper layer, and therefore in the Hasse diagram of the twin lattice the lower-layer graphs are represented below the upper-layer graphs. More specifically, within each of the two layers the models are both $\preceq_s$– and $\preceq_t$–incomparable, whereas every model in the lower layer is smaller, in the twin order sense, than some model in the upper layer. Thus, the twin order induces a partial ordering within the set of neighbouring submodels of any given model $\mathcal{P}(\mathcal{G})$, which will prove useful both in the implementation of the coherence principle and in the exploration of the model space.

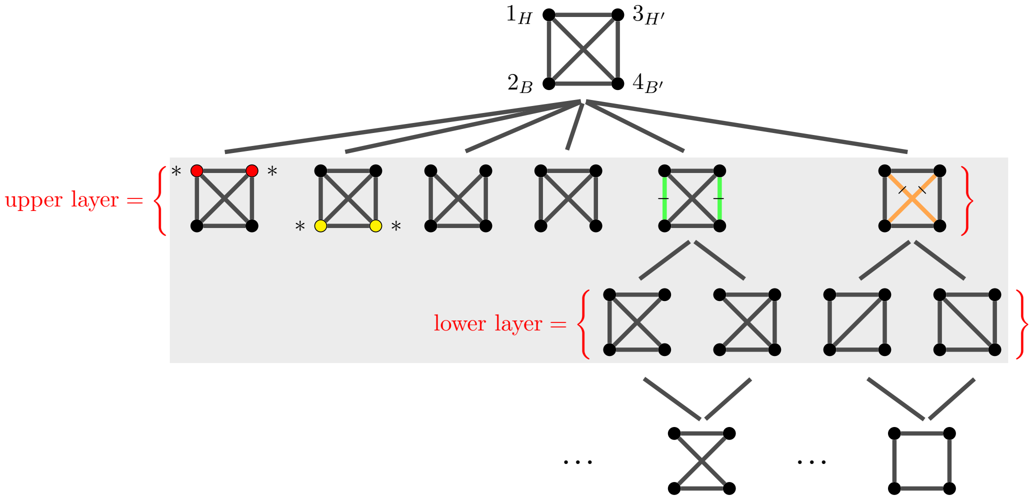


Figure 4: Part of the Hasse diagram of the twin lattice for the Frets' heads example. The graph representing the saturated model is depicted on the top. The graphs in the highlighted area are those representing the model inclusion neighbouring submodels of the saturated model, which are organized in a two-layer structure, to be compared with Figure 3.

Example 5 (Frets' Heads continued) *Figure 4 gives the portion of the Hasse diagram of the twin lattice that refines the model inclusion lattice of Figure 3 into the two-layer structure, so that every model in the lower layer is $\preceq_t$–smaller than some models in the upper, and within each layer, models are pairwise both $\preceq_s$– and $\preceq_t$–incomparable.*

The stepwise procedure that we consider in the forthcoming sections involves the iterative computation of suitable subsets of neighbouring submodels obtained by meet operation within the model inclusion lattice, that is $\wedge_s$. The rest of this section provides some results that allow us to compute such neighbouring submodels by using the more efficient meet operation within the twin lattice, $\wedge_t$, which amounts to straightforward set intersection operations.

Firstly, we show that the $\wedge_s$ and the $\wedge_t$ meet operations are equivalent when applied to two neighbouring submodels which are $\preceq_t$–incomparable.

Corollary 10 *Let $\mathcal{H}_1$, $\mathcal{H}_2$ and $\mathcal{G}$ be three pdCGs such that $\mathcal{H}_1, \mathcal{H}_2 \prec_s \mathcal{G}$ or, equivalently, such that $\mathcal{P}(\mathcal{H}_1)$ and $\mathcal{P}(\mathcal{H}_2)$ are two neighbouring submodels of $\mathcal{P}(\mathcal{G})$. Then, if $\mathcal{H}_1$ and $\mathcal{H}_2$ are $\preceq_t$–incomparable it holds that $\mathcal{H}_1 \wedge_s \mathcal{H}_2 = \mathcal{H}_1 \wedge_t \mathcal{H}_2$. On the other hand, if $\mathcal{H}_1 \preceq_t \mathcal{H}_2$ then $\mathcal{H}_1 \wedge_s \mathcal{H}_2 = \mathcal{H}_1 \wedge_t \mathcal{H}'_1$ where, $\mathcal{H}_1$ and $\mathcal{H}'_1$ are obtained one from (iii) of Proposition 8 and the other from (iv) of the same proposition.*

Proof See Section E.6 of the Appendix. ■

Finally, we provide a result that makes it possible to use the meet $\wedge_t$ in place of $\wedge_s$ at every step of the greedy search procedure described in Section 7.

Corollary 11 *Let $\mathcal{G}$ be a pdCG and, furthermore, let $\mathcal{A}$ be a set of pdCGs such that (i) the graphs in $\mathcal{A}$ are pairwise $\preceq_t$–incomparable and, (ii) for every $\mathcal{F} \in \mathcal{A}$ it holds that $\mathcal{F} \prec_s \mathcal{G}$, so that $\mathcal{P}(\mathcal{F})$ is a neighbouring submodel of $\mathcal{P}(\mathcal{G})$. Then, for every $\mathcal{H} \in \mathcal{A}$ the set*

$$\mathcal{B} = \{\mathcal{F} \wedge_s \mathcal{H} \mid \mathcal{F} \in \mathcal{A} \setminus \{\mathcal{H}\}\} = \{\mathcal{F} \wedge_t \mathcal{H} \mid \mathcal{F} \in \mathcal{A} \setminus \{\mathcal{H}\}\}$$

has the same properties as $\mathcal{A}$, that is, (i) the graphs in $\mathcal{B}$ are pairwise $\preceq_t$–incomparable and, (ii) for every $\mathcal{F} \in \mathcal{B}$ it holds that $\mathcal{F} \prec_s \mathcal{H}$, so that $\mathcal{P}(\mathcal{F})$ is a neighbouring submodel of $\mathcal{P}(\mathcal{H})$.

Proof See Section E.7 of the Appendix. ■

6. Dimension of the Search Space and Implementation of the Principle of Coherence

One major challenge in learning the structure of a coloured graphical model is the dimension of the search space that is extremely large even when the number of variables, $|V| = p$, is small. The dimension of the space $M(V)$ of GGMs is well-known to be $|M(V)| = 2^{\binom{p}{2}}$. The dimension of the search space $\mathcal{M}$ of RCON models was computed in Gehrmann (2011, eqn. (7)), where it is shown, for example, that for $p = 5$ there are 1 024 undirected GGMs but 35 285 640 RCON models. The family of pdRCON models $\mathcal{P}$ forms a proper subset of RCON models, $\mathcal{P} \subset \mathcal{M}$, however the dimension of $\mathcal{P}$ is still much larger than that of GGMs. It is shown in Section B of the Appendix that the dimension of $\mathcal{P}(V)$ can be computed as

$$|\mathcal{P}(V)| = 2^{p/2} \sum_{i=0}^{p(p-2)/4} \binom{\frac{p(p-2)}{4}}{i} 2^{\binom{p}{2}-2i},$$

so that, for example, in the application of Section 8.2 where $p = 36$, the number of pdRCON models is 10^{35} times larger than that of GGMs, formally, $|\mathcal{P}| > |M| \times 10^{35}$.

One way to increase the efficiency of greedy search procedures is by applying the, so-called, *principle of coherence* that is used as a strategy for pruning the search space. The latter was introduced in Gabriel (1969) where it is stated that: "in any procedure involving multiple comparisons no hypothesis should be accepted if any hypothesis implied by it is rejected". We remark that, for convenience, we say "accepted" instead of the more correct "non-rejected". Consider some goodness-of-fit test for testing models at a given level α so that for every model in a given class we can apply the test and determine whether the model is rejected or accepted. In graphical modelling, the principle of coherence is typically implemented by requiring that we should not accept a model while rejecting a larger model; see, among others, Edwards (2000, Chapter 6), Madigan and Raftery (1994), Cowell et al. (1999, p. 256). Thus, if a model is rejected then also all its submodels should be rejected, and the model inclusion lattice allows a straightforward implementation of this pruning procedure because it is sufficient to remove from the Hasse diagram all the paths descending from the rejected model. However, we note that for pdRCON models this implementation of the coherence principle is not sufficient to avoid incoherent steps. This is due to the fact that, unlike the lattice of GGMs, the model inclusion lattice of pdRCON models is non-distributive. To clarify this issue, consider the five graphs $\mathcal{G}_i$, $i = 1, \dots, 5$ in Figure 5a, which depicts the relevant portion of the Hasse diagram of the model inclusion lattice. In lattice theory, the sublattice of Figure 5a is said to have the diamond structure, and its presence in a Hasse diagram causes the lattice to be non-distributive (see Davey and Priestley, 2002, Theorem 4.10). The diamond structure, that is thus present in the Hasse diagram of the pdRCON model inclusion lattice, but not in that of the lattice of GGMs, requires additional attention in the implementation of the principle of coherence of Gabriel (1969), because it may give rise to a type of incoherence that involves $\preceq_s$–incomparable neighbouring submodels. To see this, consider Figure 5a and assume that $\mathcal{P}(\mathcal{G}_2)$ is rejected. In this case, under the above interpretation of the coherence principle also $\mathcal{P}(\mathcal{G}_5) \subseteq \mathcal{P}(\mathcal{G}_2)$ should be considered rejected. On the other hand, nothing can be said with respect to $\mathcal{P}(\mathcal{G}_3)$ and $\mathcal{P}(\mathcal{G}_4)$ because $\mathcal{P}(\mathcal{G}_2)$, $\mathcal{P}(\mathcal{G}_3)$ and $\mathcal{P}(\mathcal{G}_4)$ are neighbouring submodels of $\mathcal{P}(\mathcal{G}_1)$ and are thus model-inclusion incomparable. It is therefore possible that $\mathcal{P}(\mathcal{G}_3)$ and $\mathcal{P}(\mathcal{G}_4)$ are both accepted whereas $\mathcal{P}(\mathcal{G}_2)$, and thus $\mathcal{P}(\mathcal{G}_5)$, are rejected. However, this is clearly against the coherence principle because $\mathcal{G}_5 = \mathcal{G}_3 \wedge_s \mathcal{G}_4$. More generally, it holds that

$$\mathcal{G}_5 = \mathcal{G}_2 \wedge_s \mathcal{G}_3 = \mathcal{G}_3 \wedge_s \mathcal{G}_4 = \mathcal{G}_2 \wedge_s \mathcal{G}_4,$$

and therefore it would be incoherent to reject any of the models $\mathcal{P}(\mathcal{G}_2)$, $\mathcal{P}(\mathcal{G}_3)$ or $\mathcal{P}(\mathcal{G}_4)$ while accepting the remaining two. Consider now the Hasse diagram of the twin lattice for the same models, given in Figure 5b. The three neighbouring submodels of $\mathcal{P}(\mathcal{G}_1)$ are now partitioned into a two-layer structure that can be exploited to correctly implement the coherence principle. Following the structure of the twin lattice, model $\mathcal{P}(\mathcal{G}_2)$ is tested first and, if it is accepted, then models $\mathcal{P}(\mathcal{G}_3)$ and $\mathcal{P}(\mathcal{G}_4)$ should be either both accepted or rejected and this hypothesis can be verified directly from $\mathcal{P}(\mathcal{G}_5)$. On the other hand, if $\mathcal{P}(\mathcal{G}_2)$ is rejected we can consider $\mathcal{P}(\mathcal{G}_3)$ and $\mathcal{P}(\mathcal{G}_4)$ by recalling that it would be incoherent to accept both.

It is not difficult to use the result of Proposition 8 to generalise this idea to any set of neighbouring submodels. It follows that, in the implementation of the principle of coherence by using the twin lattice, the general rule is the same as in the model inclusion lattice: if a model is rejected then all the models in the paths descending from it should be excluded from the set of candidate models. In addition, there are specific rules for upper-layer models. If an upper-layer model is rejected then all the lower-layer models directly linked to it cannot be excluded from the candidate models. On the other hand, if it is accepted, one can exclude all the lower-layer models directly linked to it, and this can greatly reduce the number of candidate models, as shown by the simulations in Section 8.1.

Finally, it is worth remarking that the organization of the neighbouring submodels into the two-layer structure, provided by the twin lattice, is useful in the implementation of the coherence principle, but it has also the advantage of eliminating the diamond structure, thereby making the lattice distributive.

7. A Coherent Greedy Search Procedure

We introduce a stepwise backward elimination procedure that exploits the twin lattice both for the computation of the meet operation and the implementation of the coherence principle.

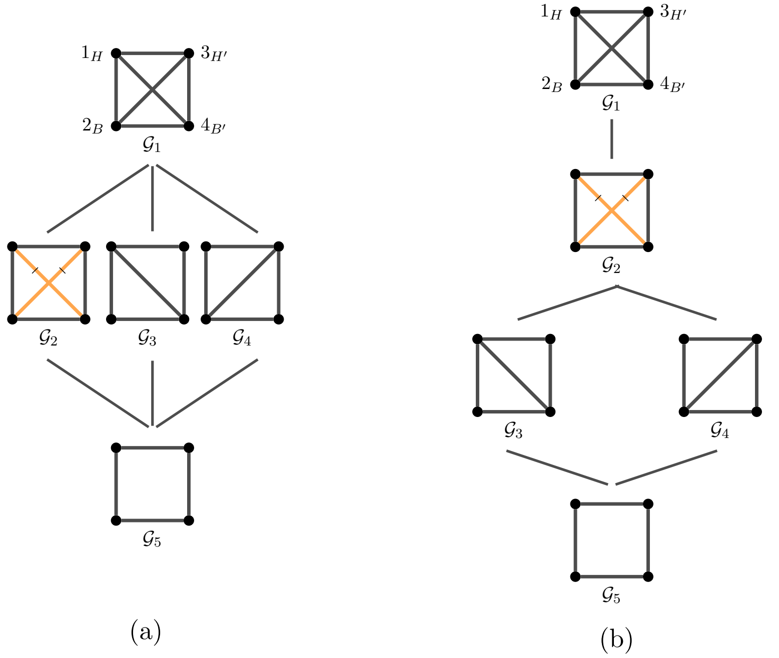


Figure 5: Frets' Heads example: Comparison of the Hasse diagrams of two sublattices induced by the same five pdCGs under (a) the model inclusion order, and (b) the twin order.

Every step of the procedure starts from a model, defined by a graph labelled as $\mathcal{G}^{\text{best}}$, and computes a set of candidate neighbouring submodels so as to obtain a set $\mathcal{A}$ of accepted models, according to a pre-established criterion. Specifically, we label as accepted the models with p-value of the likelihood ratio test against the saturated model, computed on the asymptotic chi-squared distribution, larger than 0.05. Then a new $\mathcal{P}(\mathcal{G}^{\text{best}})$ is selected from $\mathcal{A}$. In this implementation, we choose as best model in $\mathcal{A}$ the model with the largest p-value. We set the saturated model as a starting point and then the procedure is iterated until either $\mathcal{A}$ is empty or a maximum number of iterations is reached. The pseudocode of the procedure is given in Algorithms 1 and 2 where, in order to make the code more readable, we have kept the technical level low. The pseudocode with all the technical details can be found in Algorithms 3 and 4 of the Appendix.

A key issue concerns the identification, at every step, of the candidate models, which are all the coherent neighbouring submodels of $\mathcal{G}^{\text{best}}$. At the first step, efficiency is achieved by considering the upper layer first, and then the lower layer, so as to apply the principle of coherence as described in Section 6 and implemented in lines 5–11 of Algorithm 1. This can significantly reduce the dimension of the initial set of candidate models and, in turn, the dimension of the sets of candidate models of all the subsequent steps. In addition, the implementation of the coherence principle from the first step implies that the models in $\mathcal{A}$ are pairwise $\preceq_t$–incompatible and therefore, by Corollary 10, the next set of candidate models can be computed by using the more efficient $\wedge_t$ meet operation. Furthermore, Corollary 11 guarantees that the same can be done at every step of the procedure; see lines 14–17 of Algorithm 2.

Algorithm 1 Coherent stepwise backward elimination procedure

```
$\mathcal{G}^{\text{best}} \leftarrow$ complete graph with all colour classes atomic
$\mathcal{A} \leftarrow \emptyset$
$K \leftarrow$ maximum number of steps
$k \leftarrow 1$
for all upper-layer neighbouring submodels $\mathcal{H}$ of $\mathcal{G}^{\text{best}}$ do
    if $\mathcal{H}$ is accepted then $\mathcal{A} \leftarrow \mathcal{A} \cup \{\mathcal{H}\}$
end for
for all lower-layer neighbouring submodels $\mathcal{H}$ of $\mathcal{G}^{\text{best}}$ which are $\preceq_t$–incomparable with
        every graph in $\mathcal{A}$ do
    if $\mathcal{H}$ is accepted then $\mathcal{A} \leftarrow \mathcal{A} \cup \{\mathcal{H}\}$
end for
while $\mathcal{A} \neq \emptyset$ and $k < K$ do
    $\mathcal{A}$, $\mathcal{G}^{\text{best}} \leftarrow$ UPDATE($\mathcal{A}$, $\mathcal{G}^{\text{best}}$)        ▷ see Algorithm 2
    $k \leftarrow k + 1$
end while
if $\mathcal{A} \neq \emptyset$ and $k = K$ then $\mathcal{G}^{\text{best}} \leftarrow$ best model in $\mathcal{A}$
return $\mathcal{G}^{\text{best}}$
```

Algorithm 2 Update() procedure called by Algorithm 1

```
procedure UPDATE($\mathcal{A}$, $\mathcal{G}^{\text{best}}$)
    $\mathcal{G}^{\text{old}} \leftarrow \mathcal{G}^{\text{best}}$
    $\mathcal{G}^{\text{best}} \leftarrow$ best model in $\mathcal{A}$
    if $\mathcal{G}^{\text{best}}$ is obtained from $\mathcal{G}^{\text{old}}$ by removing exactly one edge $(i, j)$ with $i \neq \tau(j)$ then
        $\mathcal{H} \leftarrow$ graph obtained by removing from $\mathcal{G}^{\text{old}}$ the edge $\tau(i, j)$
        $\mathcal{A} \leftarrow \mathcal{A} \setminus \{\mathcal{H}\}$
    else if $\mathcal{G}^{\text{best}}$ is obtained from $\mathcal{G}^{\text{old}}$ by merging two atomic classes to obtain the
            twin-pairing class $\{(i, j), \tau(i, j)\}$ then
        $\mathcal{H} \leftarrow$ graph obtained by removing from $\mathcal{G}^{\text{old}}$ the edges $(i, j)$ and $\tau(i, j)$
        $\mathcal{A} \leftarrow \mathcal{A} \cup \{\mathcal{H}\}$
    end if
    $\mathcal{A}^{\text{old}} \leftarrow \mathcal{A} \setminus \{\mathcal{G}^{\text{best}}\}$
    $\mathcal{A} \leftarrow \emptyset$
    for all $\mathcal{F} \in \mathcal{A}^{\text{old}}$ do
        $\mathcal{H} \leftarrow \mathcal{F} \wedge_t \mathcal{G}^{\text{best}}$
        if $\mathcal{H}$ is accepted then $\mathcal{A} \leftarrow \mathcal{A} \cup \{\mathcal{H}\}$
    end for
    return $\mathcal{A}$ and $\mathcal{G}^{\text{best}}$
end procedure
```

8. Applications

The greedy search procedure of the previous section has been implemented in the program language R, and here we describe its application to both synthetic and real-world data, including an empirical comparison with the penalized likelihood method of Ranciati and Roverato (2023).

8.1 Simulations

In this section, we analyse the behaviour of the search procedure of Section 7 on synthetic data. More specifically, we compare it with the stepwise backward elimination procedure given in Roverato and Nguyen (2022) that does not exploit the twin lattice for the computation of the set of candidate models, and where the principle of coherence is naively implemented by only considering the submodel relationship.

We considered two scenarios that differ for the sparsity degree, computed as $|E|/|F_V|$. For each of the two scenarios we generated four pdCGs with $p = 8, 12, 16, 20$ and density degrees approximatively equal to 0.18 for scenario A and to 0.35 for scenario B. Next, for every pdCG $\mathcal{G}$ we randomly generated a concentration matrix Θ such that the normal distribution $N(0, \Theta^{-1}) \in \mathcal{P}(\mathcal{G})$ and, finally, we randomly selected 20 samples of size 100 from such normal distribution; see Section D.1 of the Appendix for details.

For each of the 160 synthetic data sets generated, we ran the two greedy search procedures, which always terminated before the maximum number of iterations was reached. The performance of the two procedures is summarized in Table 2 of the Appendix. The point of main interest is the comparison in terms of efficiency, that we quantify with respect to the average execution time and the average number of fitted models. These can be found in the last two columns of Table 2 and, furthermore, the growth rates of these measurements are displayed in Figures 6 and 7. We can see that the procedure on the twin lattice is considerably more efficient. Specifically, the procedure on the twin lattice was more than five times faster, requiring between 16% and 20% of the time required by the procedure on the model inclusion lattice. Furthermore, the proper implementation of the coherence principle allowed us to fit a much smaller number of models, ranging between 37% and 54% of the models fitted under the naive implementation of principle of coherence. It is also interesting to notice that the latter proportions appear to decrease as p increases. Table 2 also gives the average values over the 20 samples of the positive predicted value, the true-positive rate and the true-negative rate, both for the edges and for the colour classes of the selected graphs. These have satisfying values and there are not relevant differences between the two procedures, thereby showing that the increase in efficiency is not achieved at the cost of a lower level of performance of the selected model.

8.2 Brain Networks from fMRI Data

Functional MRI is a non invasive technique for collecting data on brain activity that measures the increase in the oxygenation level at some specific brain region, as long as an increase in blood flow occurs, due to some brain activity. The construction of a network from fMRI data requires first the identification of a set of functional vertices, such as spatial regions of interest (ROIs), and then the analysis of connectivity patterns across ROIs.

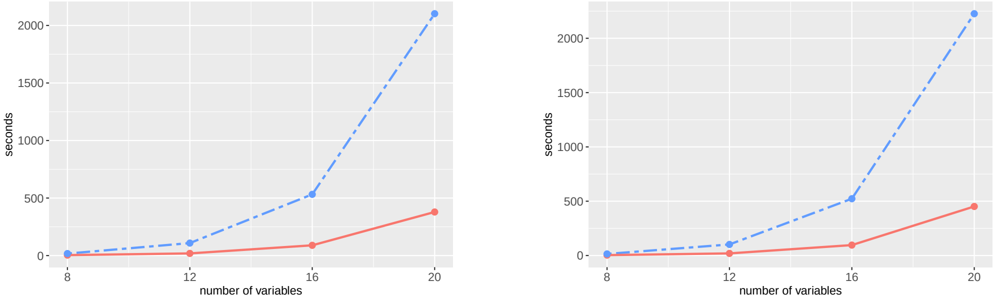


Figure 6: Average computational time in the simulations for the procedures based on the twin lattice (full red line) and on the model inclusion lattice (dashed blue line), for scenario A (left panel) and B (right panel).

The data set we use for this application comes from a pilot study of the Enhanced Nathan Kline Institute-Rockland Sample project that are time series recorded on 70 ROIs at 404 equally spaced time points. A detailed description of the project, scopes, and technical aspects can be found at `http://fcon_1000.projects.nitrc.org/indi/enhanced/`. Following Ranciati et al. (2021) we apply our method to the residuals estimated from the vector autoregression models, carried out to remove the temporal dependence. We consider two subjects indexed by 14 and 15, who have the same psychological traits with no neuropsychiatric diseases and right-handedness. The main difference is that subject 14 is 19 years old whereas the subject 15 is 57 years old. The human brain has a natural symmetric structure so that for every spatial ROI on the left hemisphere there is an homologous ROI on the right hemisphere. Accordingly, we identify the left and right groups with the left and the right hemispheres, respectively, and consider 36 cortical brain regions, that are 22 regions in the frontal lobe and 14 regions in the anterior temporal lobe. We have therefore $|V| = 36$ with $|L| = |R| = 18$.

This section describes the analysis for the subject 15 whereas the analysis of subject 14 is given in Section D.2 of the Appendix. We applied the procedure of Section 7 and selected the model defined by the pdCG given in Figure 8. The graph of the selected

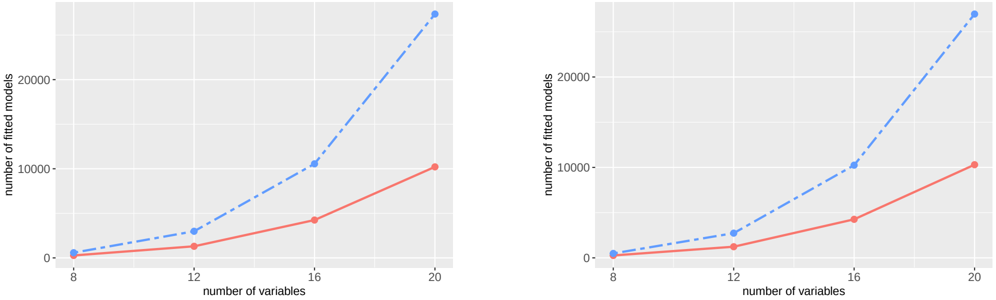
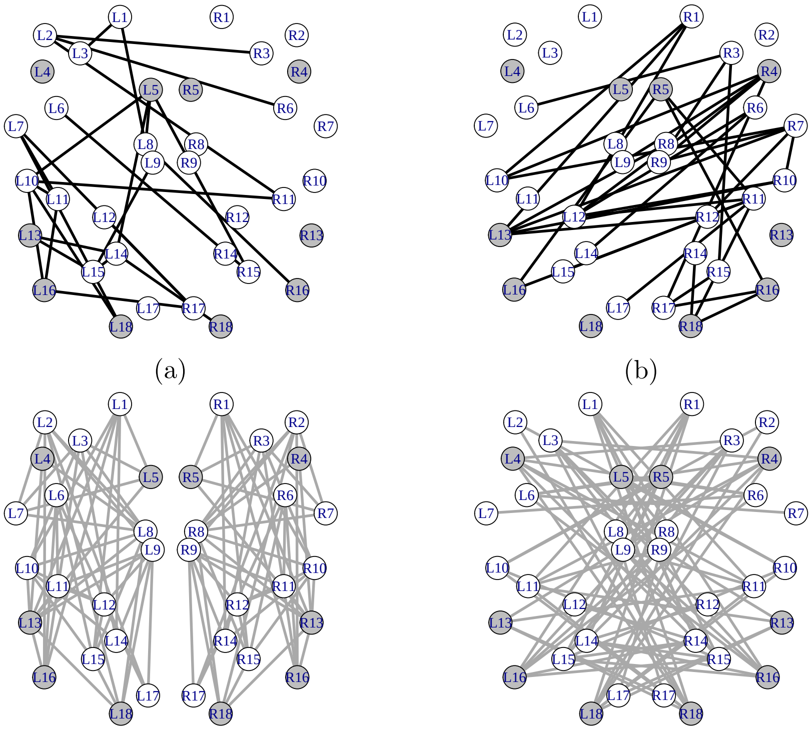


Figure 7: Averaged numbers of fitted models in the simulations for the procedures based on the twin lattice (full red line) and on the model inclusion lattice (dashed blue line), for scenario A (left panel) and B (right panel).



(a) (b) (c) (d)

Figure 8: Selected pdCG for subject 15 with separate panels for: (a) edges in E_L forming atomic classes; (b) edges in E_R forming atomic classes; (c) twin-pairing classes between groups; (d) twin-pairing classes across groups. Twin-pairing classes are depicted in gray, also for vertices, and the edges in E_T are not depicted.

model, denoted by $\mathcal{G}$, has density equal to 52.2%, the number of edges is 329 and there are 5 vertex and 85 edge twin-pairing colour classes, respectively. It provides an adequate fit with a p-value$= 0.064$, computed on the asymptotic chi-squared distribution on 275 degrees of freedom of the likelihood ratio test for the comparison with the saturated model. Note that Figure 8 splits the selected graph into 4 panels. The top-left panel gives the edges of $\mathcal{G}$ that belong to E_L and form atomic colour classes, and similarly for the top-right panel that gives the atomic colour classes in E_R. Furthermore, the bottom-left and right panels give the edge twin-pairing colour classes between and across groups, respectively. We use the gray colour for both vertex and edge twin-pairing classes and, finally, we have omitted to represent the edges in E_T because this set is almost complete, in the sense that $E_T = F_T \setminus \{(L4, R4)\}$. We deem that this representation can effectively illustrate the main features of the model, for example highlighting a high degree of symmetry in $\mathcal{G}$, given that 51.67% of the edges of $\mathcal{G}$ belong to twin-pairing classes. The selected model has 275 parameters and it is considerably more parsimonious than the saturated model that has 666 parameters. It is also interesting to notice that the GGM defined by the undirected version of $\mathcal{G}$ has 90 parameters more than the selected pdRCON model.

8.3 Comparison with Penalized Likelihood Methods

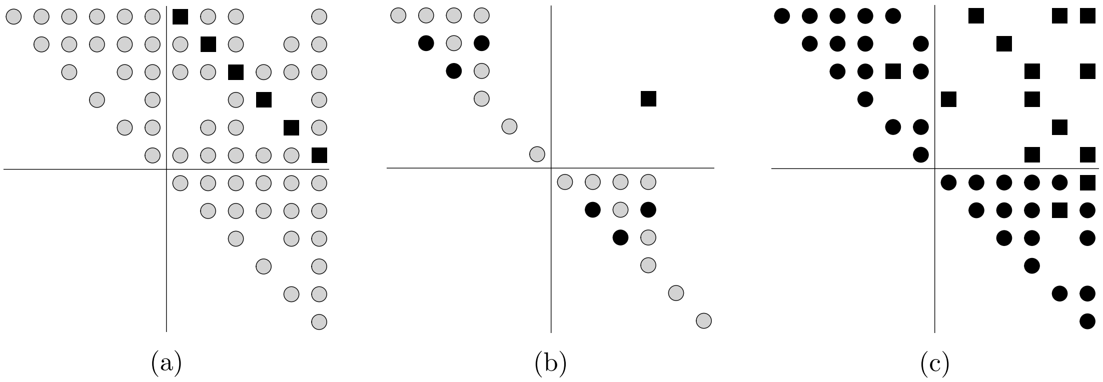
We now carry out a comparison of our greedy search procedure with the graphical lasso for paired data (pdglasso) method of Ranciati and Roverato (2023), with special attention to the role played by the scale of the variables. The methods are applied to an Air Quality dataset containing average hourly measurements from a multi-sensor gas device for one year (De Vito et al., 2008). The device was located in a heavily polluted area of an Italian city, at road level. Additional details, including the dataset, can be found at `https://archive.ics.uci.edu/ml/datasets/Air+Quality`.

We consider 6 variables, relative to the 4 substances CO, $C6H6$, $NO2$, $O3$, and the 2 meteorological measurements RH (relative humidity) and AH (absolute humidity). In order to analyse the different behaviour during the night and day hours, for every day the measure at 1am is matched to that at 1pm, so that $|V| = 12$ with $|L| = |R| = 6$. The dataset obtained after removing missing values is made up of 373 observations, and we model the residual structure of a lag-1 autoregression model (Epskamp et al., 2018).

These data present a considerable difference in the scale of the variables, with a much larger scale for variables CO, $C6H6$, $NO2$ and $O3$ compared to the scale of RH and AH. As explained in Section 8.3, in this case the application of the pdglasso is problematic and, in order to clarify this issue, we apply the method to both the unscaled and the standardized data. Models are chosen on the basis of the Bayesian information criterion (BIC). Figure 9b gives the model obtained from the application of the pdglasso to the sample covariance matrix, and it is evident that the result strongly depends on the scale of the variables with the variables RH and AH independent of both each other and all the remaining variables. This seems unrealistic if one notes, for example, that the sample correlation of RH and AH is equal to 0.66 at 1am, and to 0.61 at 1pm. Indeed, the model selected from the sample correlation matrix, in Figure 9c, has a denser structure with the variables RH and AH connected both between them and with some of the remaining variables. This model has 46 parameters and, compared with the saturated model, its p-value is almost equal to 0. Furthermore, no twin-pairing class is identified, but it is not clear how to interpret this result if one recalls that standardization may affect the structure of twin-pairing classes. Interestingly, the model in Figure 9a, selected by the greedy search procedure, has a fully symmetric structure, thereby suggesting that there is no difference between the association structure in the night and day hours. The selected graph looks denser than that in Figure 9c, nevertheless the model of Figure 9a is more parsimonious with 36 parameters, and it shows an adequate fit, with p-value equal to 0.39.

9. Conclusions

We have considered the problem of structure learning of GGMs for paired data by focusing on the family of RCON models defined by coloured graphs named pdCGs. The main results of this paper provide insight into the structure of the model inclusion lattice of pdCGs. We have introduced an alternative representation of these graphs that facilitates the computation of neighbouring models. Furthermore, this alternative representation is naturally associated with a novel order relationship that has led to the construction of the twin lattice, whose structure resembles that of the well-known set inclusion lattice, and that facilitates the exploration of the search space. These results can be applied in



(a) (b) (c)

Figure 9: Matrix representation of the selected pdRCON models for the Air Quality data by: the greedy search procedure (a), the paired data graphical lasso procedure applied to unscaled (b) and standardized (c) data. The left and right blocks correspond to the 1am and 1pm measurements, respectively, with variable ordering CO, $C6H6$, $NO2$, $O3$, RH and AH. The twin-pairing classes are denoted by gray dots (○), whereas black symbols are used for atomic classes, with black dots (●) in the case where homologous vertices/edges are both present and black squares (■) for edges whose twin edge is not present.

the implementation of both greedy and Bayesian model search procedures. Here, we have shown how they can be used to improve the efficiency of stepwise backward elimination procedures. This has also made it clear that the use of the twin lattice facilitates the correct application of the principle of coherence. Finally, we have applied our procedure to learn a brain network on 36 variables. This model dimension could be regarded as somehow small, compared with the number of variables that can be dealt with by penalized likelihood methods. This is due to the fact that, as shown in Section 6, the number of pdRCON models is much larger than that of GGMs and the same is the number of neighbouring submodels that need to be identified at every step of the algorithm. Furthermore, for every model considered, the computation of the maximum likelihood estimate is not available in closed form, but it involves an iterative procedure. Efficiency improvement is object of current research and could be achieved, for instance, by both implementing a procedure that deals with candidate submodels in parallel, and a procedure for the computation of maximum likelihood estimates explicitly designed for pdRCON models. We recall, however, that, as explained in Sections 4 and 8.3, although penalized likelihood methods are considerably more efficient, their use is problematic when variables are not measured on comparable scales. Finally, we also remark that the range of application of our results does not restrict to pdRCON models. In fact, the colouring of vertices and edges of pdCGs can be associated with different types of equality restrictions, and thus to other types of graphical models for paired data for which penalized likelihood methods are not available. For instance, they could be used to identify a subfamily of RCOR models, which impose equality restrictions between specific partial variances and correlations (Højsgaard and Lauritzen, 2008).

Acknowledgments

We would like to thank two anonymous referees for their valuable comments, and Saverio Ranciati and Veronica Vinciotti for useful discussion. Financial support was provided by the MUR-PRIN grant 2022SMNNKY (CUP C53D23002580006).

Appendix A. On the Partition of the Edge Set Induced by the Twin-pairing Function

In this section we provide a detailed example of the partition of the edge set of an undirected graph $G = (V, E)$ as described in Section 5.1. We let $p = 6$ so that $V = \{1, 2, 3, 4, 5, 6\}$ and, furthermore, we set $L = \{1, 2, 3\}$. Hence, $R = \{4, 5, 6\}$ and we assume that $\tau(i) = 3 + i$, for every $i \in L$, so that $\tau(1) = 4$, $\tau(2) = 5$ and $\tau(3) = 6$.

In this case, the edge set of the complete graph is $F_V = \{(i, j) \mid i, j = 1, \ldots, 6;\ i < j\}$, and it can be partitioned into the three sets F_L, F_R and F_T as follows,

$$\begin{aligned} F_L &= \{(1,2),(1,3),(2,3),(1,5),(1,6),(2,6)\}, \\ F_R &= \{(4,5),(4,6),(5,6),(2,4),(3,4),(3,5)\}, \\ F_T &= \{(1,4),(2,5),(3,6)\}, \end{aligned}$$

which are graphically represented in Figure 10. Note that one can easily check that $F_R = \tau(F_L)$, $F_L = \tau(F_R)$ and $F_T = \tau(F_T)$. In order to understand the meaning of this partition of F_V it is useful to note that the set F_L can be seen as the union of two disjoint subsets, $F_L = F_L^{(within)} \cup F_L^{(across)}$ where

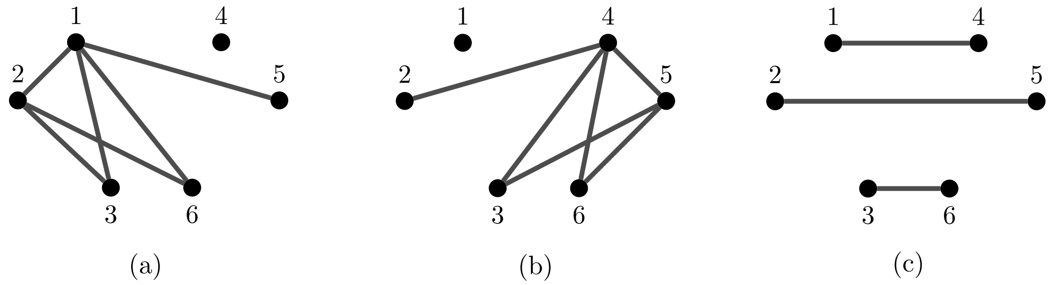
$$\begin{aligned} F_L^{(within)} &= \{(i,j) \in F_V \mid i, j \in L\} \\ F_L^{(across)} &= \{(i,j) \in F_V \mid i \in L, j \in R, i < \tau(j)\}. \end{aligned}$$

Dually, $F_R = F_R^{(within)} \cup F_R^{(across)}$ with

$$\begin{aligned} F_R^{(within)} &= \{(i,j) \in F_V \mid i, j \in R\} \\ F_R^{(across)} &= \{(i,j) \in F_V \mid i \in L, j \in R, i > \tau(j)\}. \end{aligned}$$

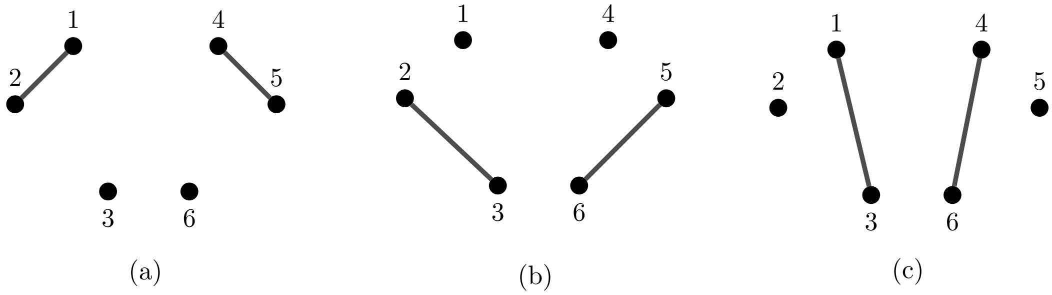
Note that $F_L^{(within)} = \tau(F_R^{(within)})$ and thus $F_R^{(within)} = \tau(F_L^{(within)})$, so that every edge in $F_L^{(within)}$ has a twin in $F_R^{(within)}$. The edges in $F_L^{(within)}$ and $F_R^{(within)}$ belong to the left and right group, respectively, and thus the twin-pairing classes characterized by these edges encode similarities involving edges within the two groups. For the example on $p = 6$ these two sets become $F_L^{(within)} = \{(1,2),(1,3),(2,3)\}$ and $F_R^{(within)} = \{(4,5),(4,6),(5,6)\}$ and the relevant twin-pairing classes are represented in Figure 11.

The edges in $F_L^{(across)}$ and $F_R^{(across)}$ encode the cross-group association structure. Also in this case $F_L^{(across)} = \tau(F_R^{(across)})$ and $F_T^{(across)} = \tau(F_T)^{(across)}$ so that every edge in $F_L^{(across)}$ has a twin in $F_R^{(across)}$. For the case $p = 6$ these sets become $F_L^{(across)} = \{(1,5),(1,6),(2,6)\}$ and $F_R^{(across)} = \{(2,4),(3,4),(3,5)\}$ and the corresponding colour classes are depicted in Figure 12.



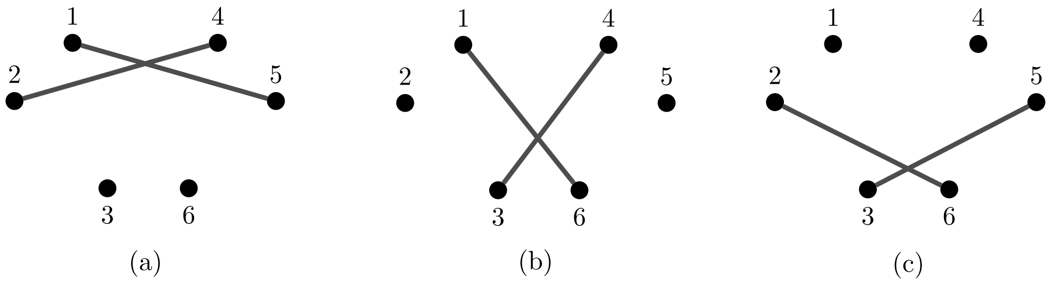
(a) (b) (c)

Figure 10: Graphs with vertex set $V = \{1,2,3,4,5,6\}$ and edge sets: (a) F_L, (b) F_R and (c) F_T.



(a) (b) (c)

Figure 11: Edge twin-pairing classes involving edges within groups: each of the graphs (a), (b) and (c) contains exactly one edge $(i,j) \in F_L^{(within)} = \{(1,2),(1,3),(2,3)\}$ and its twin $\tau(i,j) \in F_R^{(within)} = \{(4,5),(4,6),(5,6)\}$.



(a) (b) (c)

Figure 12: Edge twin-pairing classes involving edges across groups: each of the graphs (a), (b) and (c) contains exactly one edge $(i,j) \in F_L^{(across)} = \{(1,5),(1,6),(2,6)\}$ and its twin $\tau(i,j) \in F_R^{(across)} = \{(2,4),(3,4),(3,5)\}$.

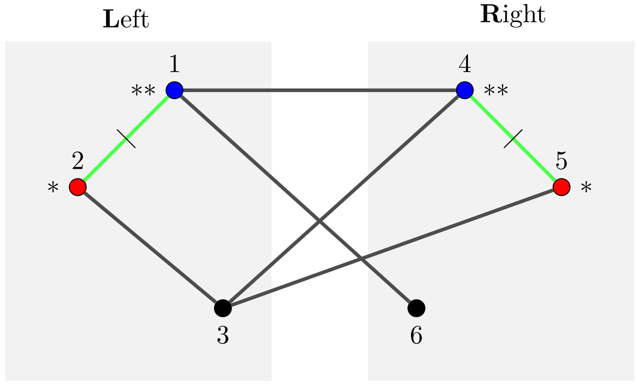


Figure 13: Example of coloured graph for paired data.

Consider now the pdCG in Figure 13. This is denoted by $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with

$$\begin{aligned}\mathcal{V} &= \{\{1,4\},\{2,5\},\{3\},\{6\}\}\\ \mathcal{E} &= \{\{(1,2),(4,5)\},\{(1,4)\},\{(1,6)\},\{(2,3)\},\{(3,4)\},\{(3,5)\}\}.\end{aligned}$$

We now compute the equivalent representation $\mathcal{G} = (V, E, \mathbb{L}, \mathbb{E})$. First, we can trivially obtain that $V = \{1,2,3,4,5,6\}$ and $E = \{(1,2),(1,4),(1,6),(2,3),(3,4),(3,5),(4,5)\}$. Next, because $L = \{1,2,3\}$ and the only vertex atomic colour classes are $\{3\}$ and $\{6\}$, we can see that $\mathbb{L} = \{3\}$. Finally, we compute $E_L = E \cap F_L = \{(1,2),(1,6),(2,3)\}$ and $E_R = E \cap F_R = \{(4,5),(3,4),(3,5)\}$. It follows that $\tau(E_R) = \{(1,2),(1,6),(2,6)\}$ so that $E_L \cap \tau(E_R) = \{(1,2),(1,6)\}$. Because the edge $(1,2)$ belongs to a twin-pairing colour class with its twin $(4,5)$, whereas $(1,6)$ belongs to an atomic colour class, then we obtain that $\mathbb{E} = \{(1,6)\}$.

Appendix B. Dimension of the Search Space of pdRCON Models

In this section, we provide a formula for the computation of the number of pdRCON models $\mathcal{P}(V)$ with $|V| = p$.

Firstly, we notice that in a pdRCON model with p variables the maximum numbers of vertex and edge twin-pairing colour classes are $p/2$ and $|F_L| = p(p-2)/4$, respectively. Next, we compute $|\mathcal{P}(V)|$ by steps, starting from the case where there are no twin-pairing colour classes and then increasing the number of twin-twin pairing classes.

- the number of pdCGs where all colour classes are atomic coincides with the number of GGMs and is equal to $2^{\binom{p}{2}}$,
- the number of pdCGs where vertex twin-pairing classes are allowed but edge colour classes are all atomic is $2^{p/2} \times 2^{\binom{p}{2}}$,
- the number of pdCGs where vertex twin-pairing classes are allowed and at most 1 edge twin-pairing class is present is

$$2^{p/2}\binom{p(p-2)/4}{1}2^{\binom{p}{2}-2},$$

- the number of pdCGs where vertex twin-pairing classes are allowed and at most 2 edge twin-pairing classes are present is
$$2^{p/2}\binom{p(p-2)/4}{2}2^{\binom{p}{2}-4},$$
- the number of pdCGs where vertex twin-pairing classes are allowed and at most i edge twin-pairing classes are present, with $0 \leq i \leq p(p-2)/4$, is
$$2^{p/2}\binom{p(p-2)/4}{i}2^{\binom{p}{2}-2i}.$$

So finally, the number of pdRCON models is

$$|\mathcal{P}(V)| = 2^{p/2}\sum_{i=0}^{p(p-2)/4}\binom{p(p-2)/4}{i}2^{\binom{p}{2}-2i}.$$

Appendix C. Technical Details on the Greedy Search procedure

Here we provide Algorithms 3 and 4 which contain a fully detailed, expanded version of the pseudocode given in Algorithms 1 and 2. Note that the pseudocode calls the procedures Is.Accepted() and Best.Model() that are left unspecified. The procedure Is.Accepted() takes a model as input and returns `true` if the model satisfies a given, arbitrary, criterion and `false` otherwise. The procedure Best.Model() accepts a non-empty set of models as input and returns one of the models of such set that is identified as "best", according to a specified criterion. Finally, an implementation of the procedure in the programming language R can be found at `https://github.com/NgocDung-NGUYEN/backwardCGM-PD`.

Appendix D. Additional Details on the Applications

D.1 Simulations

In this section, we provide additional details on the simulations described in Section 8.1; see also `https://github.com/NgocDung-NGUYEN/backwardCGM-PD`.

Table 1 considers the 8 pdCGs used in the simulations and gives details on edges, pairs of twin edges present in the graph and colour classes.

For each pdCG in Table 1 above, we generated a concentration matrix by exploiting the R package `gRc` (Højsgaard and Lauritzen, 2007). More specifically, we applied the function `rcox()` by giving in input the $p \times p$ equicorrelation matrix with the diagonal entries equal to 1, and all off-diagonal entries equal to 0.5. Each of resulting concentration matrices was used to generate 20 random samples of size $n = 100$ from the relevant multivariate normal distribution, with mean vector equal to zero. The two procedures were then applied to the $20 \times 8 = 160$ datasets so obtained, and we also remark that we exploited the package `gRc` for the computation of maximum likelihood estimates.

Table 2 summarizes the average results over all repetitions of the simulated data of the performance scores which measure the identifications of the zeros and symmetric structures

Algorithm 3 Expanded version of the pseudocode in Algorithm 1.

```
Require: $V$ and $L$
$\mathcal{G}^{\text{best}} \leftarrow (V, F_V, L, F_L)$
$\mathcal{A} \leftarrow \emptyset$
$K \leftarrow$ maximum number of steps ($\geq 1$)
$k \leftarrow 1$
for all $i \in L$ do
    $\mathcal{H} \leftarrow (V, F_V, L \setminus \{i\}, F_L)$
    if Is.Accepted($\mathcal{H}$) then
        $\mathcal{A} \leftarrow \mathcal{A} \cup \{\mathcal{H}\}$
    end if
    $\mathcal{H} \leftarrow (V, F_V \setminus \{(i, \tau(i))\}, L, F_L)$
    if Is.Accepted($\mathcal{H}$) then
        $\mathcal{A} \leftarrow \mathcal{A} \cup \{\mathcal{H}\}$
    end if
end for
for all $(i, j) \in F_L$ do
    $\mathcal{H} \leftarrow (V, F_V, L, F_L \setminus \{(i, j)\})$
    if Is.Accepted($\mathcal{H}$) then
        $\mathcal{A} \leftarrow \mathcal{A} \cup \{\mathcal{H}\}$
    else
        $\mathcal{H} \leftarrow (V, F_V \setminus \{(i, j)\}, L, F_L \setminus \{(i, j)\})$
        if Is.Accepted($\mathcal{H}$) then
            $\mathcal{A} \leftarrow \mathcal{A} \cup \{\mathcal{H}\}$
        end if
        $\mathcal{H} \leftarrow (V, F_V \setminus \{\tau(i, j)\}, L, F_L \setminus \{(i, j)\})$
        if Is.Accepted($\mathcal{H}$) then
            $\mathcal{A} \leftarrow \mathcal{A} \cup \{\mathcal{H}\}$
        end if
    end if
end for
while $\mathcal{A} \neq \emptyset$ and $k < K$ do
    $\mathcal{A}$, $\mathcal{G}^{\text{best}} \leftarrow$ Update($\mathcal{A}$, $\mathcal{G}^{\text{best}}$)          ▷ see Algorithm 4
    $k \leftarrow k + 1$
end while
if $\mathcal{A} \neq \emptyset$ and $k = K$ then $\mathcal{G}^{\text{best}} \leftarrow$Best.Model($\mathcal{A}$)
return $\mathcal{G}^{\text{best}}$
```

Algorithm 4 Expanded version of the pseudocode in Algorithm 2.

```
procedure Update($\mathcal{A}$, $\mathcal{G}^{\text{best}}$)
    $\mathcal{G}^{\text{old}} \leftarrow \mathcal{G}^{\text{best}}$
    $\mathcal{G}^{\text{best}} \leftarrow$Best.Model($\mathcal{A}$)        ▷ Note that:
                                            $\mathcal{G}^{\text{old}} = (V, E^{\text{old}}, \mathbb{L}^{\text{old}}, \mathbb{E}^{\text{old}})$
                                            $\mathcal{G}^{\text{best}} = (V, E^{\text{best}}, \mathbb{L}^{\text{best}}, \mathbb{E}^{\text{best}})$
    $e \leftarrow E^{\text{best}} \setminus E^{\text{old}}$
    if $|e| = 1$ and $e \neq \tau(e)$ then
        $\mathcal{H} \leftarrow (V, E^{\text{old}} \setminus \{\tau(e)\}, \mathbb{L}^{\text{old}}, \mathbb{E}^{\text{old}} \setminus \{e, \tau(e)\})$
        $\mathcal{A} \leftarrow \mathcal{A} \setminus \{\mathcal{H}\}$
    else if $e = \emptyset$ and $\mathbb{E}^{\text{best}} \neq \mathbb{E}^{\text{old}}$ then
        $e \leftarrow \mathbb{E}^{\text{old}} \setminus \mathbb{E}^{\text{best}}$
        $\mathcal{H} \leftarrow (V, E^{\text{old}} \setminus \{e, \tau(e)\}, \mathbb{L}^{\text{old}}, \mathbb{E}^{\text{old}} \setminus \{e\})$
        $\mathcal{A} \leftarrow \mathcal{A} \cup \{\mathcal{H}\}$
    end if
    $\mathcal{A}^{\text{old}} \leftarrow \mathcal{A} \setminus \{\mathcal{G}^{\text{best}}\}$
    $\mathcal{A} \leftarrow \emptyset$
    for all $\mathcal{F} \in \mathcal{A}^{\text{old}}$ do
        $\mathcal{H} \leftarrow \mathcal{F} \wedge_t \mathcal{G}^{\text{best}}$
        if Is.Accepted($\mathcal{H}$) then $\mathcal{A} \leftarrow \mathcal{A} \cup \{\mathcal{H}\}$
    end for
    return $\mathcal{A}$ and $\mathcal{G}^{\text{best}}$
end procedure
```

of the resulting models recovered from the model selection procedure. Specifically, for the accuracy of the graph structures, we considered the quantities such as the edge positive-predicted value (ePPV), the edge true-positive rate (eTPR), and the edge true-negative rate (eTNR), which are formally specified as

$$\text{ePPV} = \frac{\text{eTP}}{\#\text{edges}}, \quad \text{eTPR} = \frac{\text{eTP}}{\text{eP}}, \quad \text{eTNR} = \frac{\text{eTN}}{\text{eN}},$$

where

- eTP: the number of true edges in the selected graph,
- #edges: the number of edges in the selected graph,
- eP: the number of edges in the true graph,
- eTN: the number of true missing edges in the selected graph,
- eN: the number of missing edges in the true graph.

Similarly, for the identification of the symmetries, that is of the twin-pairing classes, we considered the symmetry positive-predicted value (sPPV), the symmetry true-positive rate

p	Scen.	Structure				Symmetries			
		den.$_{\%}$	$\|E\|$	$\|E_T\|$	$\|E_L \cap \tau(E_R)\|$	$\|\mathbb{E}\|$	$\|\mathbb{E}^c\|$	$\|\mathbb{L}\|$	$\|\mathbb{L}^c\|$
8	A	17.85	5	0	2	1	1	3	1
	B	35.71	10	0	4	1	3	1	3
12	A	18.18	12	0	5	4	1	4	2
	B	34.85	23	1	9	3	6	2	4
16	A	18.33	22	1	9	7	2	6	2
	B	35.00	42	3	15	5	10	2	6
20	A	17.94	34	2	14	11	3	8	2
	B	34.74	66	6	24	8	16	2	8

Table 1: Details on the pdCGs used in the simulations: den.$_{\%}$ is the density computed from $|E|/|F_V|$; $|E|$ is the number of edges; $|E_T|$ is the number of edges connecting pairs of twin vertices; $|E_L \cap \tau(E_R)|$ is the number of edges for which a twin is present in the graph; $|\mathbb{E}|$ and $|\mathbb{L}|$ are the number of edge and vertex atomic colour classes, respectively; $|\mathbb{E}^c| = |(E_L \cap \tau(E_R)) \setminus \mathbb{E}|$ and $|\mathbb{L}^c| = |L \setminus \mathbb{L}|$ are the number of edge and vertex twin-pairing classes, respectively.

(sTPR), and the symmetry true-negative rate (sTNR), as defined by Ranciati et al. (2021). Specifically, they are computed as

$$\text{sPPV} = \frac{\text{sTP}}{\#\text{sym}}, \quad \text{sTPR} = \frac{\text{sTP}}{\text{sP}}, \quad \text{sTNR} = \frac{\text{sTN}}{\text{sN}},$$

where

- sTP: the number of pairs of true twin-pairing edges in the selected graph,
- #sym: the number of pairs of twin-pairing edges in the selected graph,
- sP: the number of pairs of twin-pairing edges in the true graph,
- sTN:the number of pairs of twin-pairing edges that are missing in the selected graph,
- sN: the number of pairs of twin-pairing edges that are missing in the true graph.

Finally, on the last two columns of the table we report the computational time of the procedures and the total number of fitted models during the execution of the algorithms.

D.2 Additional Details on the Application to fMRI Data

We applied our method to a multimodal imaging dataset coming from a pilot study of the Enhanced Nathan Kline Institute-Rockland Sample project, and provided by Greg Kiar and Eric Bridgeford from NeuroData at Johns Hopkins University, who pre-processed the

S	p	Order	Graph structure				Symmetries				Time$_{(s)}$	#models
			#edges	ePPV$_{\%}$	eTPR$_{\%}$	eTNR$_{\%}$	#sym	sPPV$_{\%}$	sTPR$_{\%}$	sTNR$_{\%}$		
A	8	$\preceq_t$	7(2)	76.68	100.00	91.52	2(1)	41.67	95.00	89.44	4	273
		$\preceq_s$	7(2)	75.41	100.00	91.30	2(1)	46.67	95.00	85.56	17	580
	12	$\preceq_t$	17(3)	71.22	97.92	90.37	6(1)	15.99	90.00	87.61	19	1300
		$\preceq_s$	17(3)	70.23	98.75	90.00	5(1)	17.34	90.00	83.91	109	2985
	16	$\preceq_t$	27(4)	74.83	88.64	92.70	9(1)	18.53	85.00	89.43	89	4245
		$\preceq_s$	28(4)	70.98	87.05	91.48	8(1)	19.32	77.50	84.77	532	10554
	20	$\preceq_t$	44(8)	64.24	82.21	89.49	16(3)	13.47	70.00	86.18	379	10212
		$\preceq_s$	46(7)	60.11	78.97	88.04	13(3)	11.97	51.67	80.00	2102	27356
B	8	$\preceq_t$	11(2)	84.54	89.50	89.72	5(1)	64.08	93.33	92.50	4	264
		$\preceq_s$	11(2)	83.59	89.00	89.44	4(1)	64.83	85.00	85.83	15	486
	12	$\preceq_t$	23(4)	81.78	80.00	89.65	9(2)	56.28	79.17	87.35	19	1230
		$\preceq_s$	23(4)	81.25	78.48	89.53	7(2)	63.26	73.33	83.53	102	2729
	16	$\preceq_t$	34(5)	72.49	57.86	87.63	12(2)	52.38	64.00	86.09	96	4259
		$\preceq_s$	31(4)	74.50	55.24	89.49	9(2)	63.36	54.00	82.97	523	10247
	20	$\preceq_t$	51(9)	69.74	53.41	87.02	18(2)	48.17	54.38	84.07	452	10300
		$\preceq_s$	48(7)	67.81	48.64	87.22	12(2)	52.97	39.38	78.98	2226	26961

Table 2: Performance measures of the model selection procedures on the lattice structures ordered by $\preceq_t$ and $\preceq_s$. Results are recorded as mean computed across the 20 replicated datasets for each scenario. Here, the columns of #edges and #sym include the average numbers of edges and of pairs of symmetric edges, respectively, and the numbers in the round bracket are the corresponding standard deviations. The columns of **Time** and **#models** measure the efficiency of the procedures that are the average of computational time and the average of numbers of fitted models, respectively.

raw DTI and R-fMRI imaging data available at `http://fcon_1000.projects.nitrc.org/indi/CoRR/html/nki_1.html`, using the pipelines ndmg and C-PAC. Particularly, the R-fMRI monitors brain functional activity at different regions via dynamic changes in the blood oxygenation level dependent (BOLD) signal, when, in this study, the subjects are simply asked to stay awake with eyes open. The data sets that we apply our methods are residuals estimated from the vector autoregression models carried out to remove the temporal dependence, see Ranciati et al. (2021).

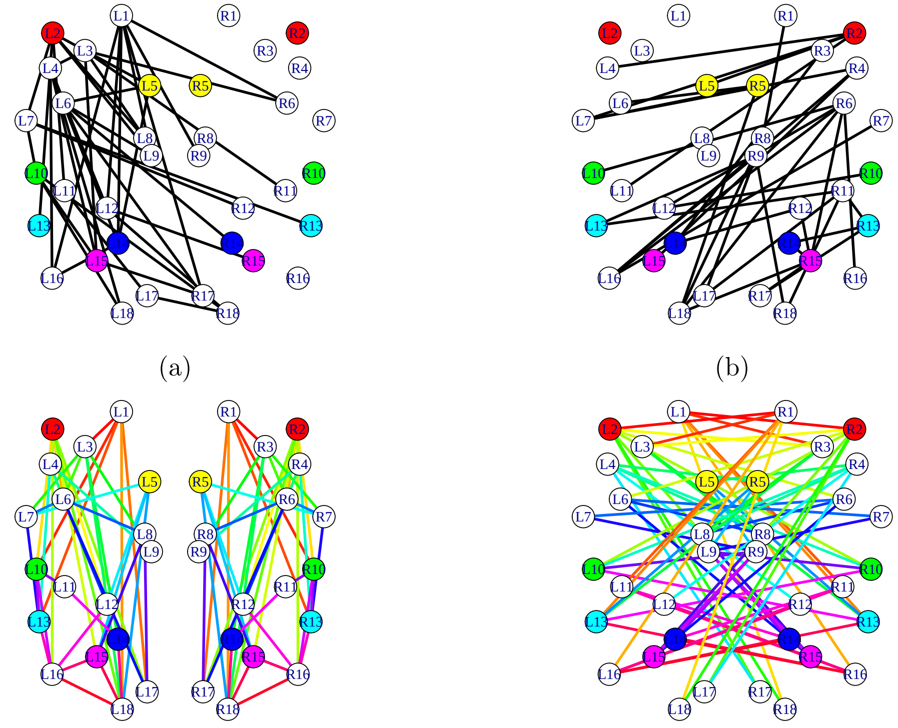
We considered 36 cortical brain regions including 22 regions in the frontal lobe and 14 regions in the anterior temporal lobe for two subjects 14 and 15. In this section, we provide the analysis of subject 14 while the analysis of subject 15 is given in Section 8.2. In particular, the graph of the selected model by our method, called by $\mathcal{G}$, has density equal to 54.28% with 342 edges present in which there are 76 edge twin-pairing colour classes, which to make up approximately 44.44% on present edges, and 6 vertex twin-pairing colour classes. Based on the likelihood ratio test for the comparison with the saturated model, the model $\mathcal{G}$ has p-values equal to 0.055 with 296 degrees of freedom computed on the asymptotic chi-squared distribution.

Figure 14 presents the coloured graphical representation of the model $\mathcal{G}$ that is split into 4 panels. The top-left panel gives the edges of $\mathcal{G}$ that belong to E_L and form atomic colour classes, and similarly for the top-right panel that gives the atomic colour classes in E_R. Furthermore, the bottom-left and right panels give the edge twin-pairing colour classes between and across groups, respectively. Here, in this case, we have omitted to represent the edges in E_T because this set is complete, in the sense that $E_T = F_T$.

Appendix E. Proofs

E.1 Proof of Proposition 2

We now show that from the quadruplet $(V, E, \mathbb{L}, \mathbb{E})$ obtained from (1) and (2) one can recover the representation $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ by applying (i) and (ii). Recall that every colour class of both $\mathcal{V}$ and $\mathcal{E}$ is either atomic or twin-pairing. We consider first the vertex colour classes $\mathcal{V}$. Every twin-pairing vertex colour class can be written as $\{i, \tau(i)\}$ with $i \in L$ and $\tau(i) \in R$, and it follows from the definition of $\mathbb{L}$ in (2) that a vertex $i \in L$ belongs to a twin-pairing colour class if and only if $i \in L \setminus \mathbb{L}$. Hence, from the pair L and $\mathbb{L}$ we can obtain the twin-pairing colour classes in $\mathcal{V}$ which are the sets $\{i, \tau(i)\}$ for all $i \in L \setminus \mathbb{L}$. It is straightforward that all the vertices in V not belonging to any of the above twin-pairing classes must form atomic colour classes. We now turn to the edge colour classes $\mathcal{E}$. The partition (E_L, E_R, E_T) of E is obtained in such a way that every twin-pairing edge colour class can be written as $\{(i, j), \tau(i, j)\}$ with $(i, j) \in E_L$ and $\tau(i, j) \in E_R$ and therefore it holds that $(i, j) \in E_L \cap \tau(E_R)$. Furthermore, by construction, the set $\mathbb{E}$ in (2) is made up of all the edges in $E_L \cap \tau(E_R)$ which form atomic colour classes. It follows that the twin-pairing edge colour classes of $\mathcal{G}$ are given by the pairs $\{(i, j), \tau(i, j)\}$ for all $(i, j) \in (E_L \cap \tau(E_R)) \setminus \mathbb{E}$, as required. Also in this case it is straightforward to see that all the edges in E not belonging to any of the above twin-pairing classes must form atomic colour classes.



(a)

(b)

(c)

(d)

Figure 14: The pdCG of the selected models for subject 14 with separate panels: (a) edges in E_L forming atomic classes; (b) edges in E_R forming atomic classes; (c) twin-pairing classes within groups; and (d) twin pairing classes across groups.

E.2 Proof of Theorem 4

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}) \in \mathcal{P}$ be a pdCG and let $(V, E, \mathbb{L}, \mathbb{E})$ the quadruplet obtained from the application of (1) and (2). It follows immediately from the construction of $G = (V, E)$ in (1) and of $\mathbb{L}$ and $\mathbb{E}$ in (2) that $(V, E, \mathbb{L}, \mathbb{E})$ is compatible with (L, R). Furthermore, the pair $(\mathcal{V}, \mathcal{E})$ can be recovered from $(V, E, \mathbb{L}, \mathbb{E})$ as shown in Proposition 2.

Consider now a compatible quadruplet $(V, E, \mathbb{L}, \mathbb{E})$ and let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ the graph obtained from the application of Proposition 2. It follows immediately from the constructing procedure that $\mathcal{G} \in \mathcal{P}$ and we have to show that if we apply (1) and (2) to $\mathcal{G}$ then we recover the quadruplet $(V, E, \mathbb{L}, \mathbb{E})$. It is straightforward to see that (1) gives $V_{\mathcal{G}} = V$ and $E_{\mathcal{G}} = E$. By (i) of Proposition 2, the twin-pairing vertex colour classes of $\mathcal{G}$ are $\{i, \tau(i)\}$ for all $i \in L \setminus \mathbb{L}$ and therefore we obtain from (2) that $\mathbb{L}_{\mathcal{G}} = L \cap \mathbb{L} = \mathbb{L}$ because $\mathbb{L} \subseteq L$ by compatibility. Similarly, we obtain from (ii) of Proposition 2 and (2) that $\mathbb{E}_{\mathcal{G}} = (E_L \cap \tau(E_R)) \cap \mathbb{E} = \mathbb{E}$ because $\mathbb{E} \subseteq E_L \cap \tau(E_R)$ by compatibility. And this completes the proof.

E.3 Proof of Theorem 6

The proof of this theorem is as follows: we firstly prove that $\langle \mathcal{P}, \preceq_t \rangle$ is a lattice by determining the explicit forms of the corresponding meet and join operations. We then identify the zero and the unit of the lattice $\langle \mathcal{P}, \preceq_t \rangle$, which implies completeness. Finally, we prove that $\langle \mathcal{P}, \preceq_t \rangle$ is distributive.

We first consider (i) and show that $\mathcal{L} = (V, E_{\mathcal{G}} \cap E_{\mathcal{H}}, \mathbb{L}_{\mathcal{G}} \cap \mathbb{L}_{\mathcal{H}}, \mathbb{E}_{\mathcal{G}} \cap \mathbb{E}_{\mathcal{H}})$ is the meet of $\mathcal{G}$ and $\mathcal{H}$, $\mathcal{G} \wedge_t \mathcal{H}$. It can be easily checked that, by construction, the quadruplet $(V, E_{\mathcal{G}} \cap E_{\mathcal{H}}, \mathbb{L}_{\mathcal{G}} \cap \mathbb{L}_{\mathcal{H}}, \mathbb{E}_{\mathcal{G}} \cap \mathbb{E}_{\mathcal{H}})$ is compatible so that $\mathcal{L} \in \mathcal{P}$. Furthermore, because $E_{\mathcal{G}} \cap E_{\mathcal{H}} \subseteq E_{\mathcal{G}}$, $\mathbb{L}_{\mathcal{G}} \cap \mathbb{L}_{\mathcal{H}} \subseteq \mathbb{L}_{\mathcal{G}}$ and $\mathbb{E}_{\mathcal{G}} \cap \mathbb{E}_{\mathcal{H}} \subseteq \mathbb{E}_{\mathcal{G}}$, then it follows immediately from Definition 5 that $\mathcal{L} \preceq_t \mathcal{G}$ and, similarly, one can show that $\mathcal{L} \preceq_t \mathcal{H}$, so that $\mathcal{L} \preceq_t \mathcal{G}, \mathcal{H}$. In this way, we have shown that $\mathcal{L}$ is a lower bound of $\mathcal{G}$ and $\mathcal{H}$ and now, in order to prove that $\mathcal{L} = \mathcal{G} \wedge_t \mathcal{H}$, we have to show that $\mathcal{L}$ is the greatest lower bound, or infimum, of $\mathcal{G}$ and $\mathcal{H}$. More formally, we have to show that if $\mathcal{F} = (V, E_{\mathcal{F}}, \mathbb{L}_{\mathcal{F}}, \mathbb{E}_{\mathcal{F}})$ is an arbitrary lower bound of $\mathcal{G}$ and $\mathcal{H}$ then $\mathcal{F} \preceq_t \mathcal{L}$. For every lower bound $\mathcal{F}$ of $\mathcal{G}$ and $\mathcal{H}$ it holds that $\mathcal{F} \preceq_t \mathcal{G}, \mathcal{H}$ and, therefore, that

1) both $E_{\mathcal{F}} \subseteq E_{\mathcal{G}}$ and $E_{\mathcal{F}} \subseteq E_{\mathcal{H}}$,

2) both $\mathbb{L}_{\mathcal{F}} \subseteq \mathbb{L}_{\mathcal{G}}$ and $\mathbb{L}_{\mathcal{F}} \subseteq \mathbb{L}_{\mathcal{H}}$,

3) both $\mathbb{E}_{\mathcal{F}} \subseteq \mathbb{E}_{\mathcal{G}}$ and $\mathbb{E}_{\mathcal{F}} \subseteq \mathbb{E}_{\mathcal{H}}$.

In turn, this implies that $E_{\mathcal{F}} \subseteq E_{\mathcal{G}} \cap E_{\mathcal{H}}$, $\mathbb{L}_{\mathcal{F}} \subseteq \mathbb{L}_{\mathcal{G}} \cap \mathbb{L}_{\mathcal{H}}$ and $\mathbb{E}_{\mathcal{F}} \subseteq \mathbb{E}_{\mathcal{G}} \cap \mathbb{E}_{\mathcal{H}}$ and therefore that $\mathcal{F} \preceq_t \mathcal{L}$, as required.

We now consider (ii) and show that $\mathcal{U} = (V, E_{\mathcal{G}} \cup E_{\mathcal{H}}, \mathbb{L}_{\mathcal{G}} \cup \mathbb{L}_{\mathcal{H}}, \mathbb{E}_{\mathcal{G}} \cup \mathbb{E}_{\mathcal{H}})$ is the join of $\mathcal{G}$ and $\mathcal{H}$, $\mathcal{G} \vee_t \mathcal{H}$. It can be easily checked that, by construction, the quadruplet $(V, E_{\mathcal{G}} \cup E_{\mathcal{H}}, \mathbb{L}_{\mathcal{G}} \cup \mathbb{L}_{\mathcal{H}}, \mathbb{E}_{\mathcal{G}} \cup \mathbb{E}_{\mathcal{H}})$ is compatible so that $\mathcal{U} \in \mathcal{P}$. Furthermore, because $E_{\mathcal{G}} \subseteq E_{\mathcal{G}} \cup E_{\mathcal{H}}$, $\mathbb{L}_{\mathcal{G}} \subseteq \mathbb{L}_{\mathcal{G}} \cup \mathbb{L}_{\mathcal{H}}$ and $\mathbb{E}_{\mathcal{G}} \subseteq \mathbb{E}_{\mathcal{G}} \cup \mathbb{E}_{\mathcal{H}}$, then it follows immediately from Definition 5 that $\mathcal{G} \preceq_t \mathcal{U}$ and, similarly, one can show that $\mathcal{H} \preceq_t \mathcal{U}$ so that $\mathcal{G}, \mathcal{H} \preceq_t \mathcal{U}$. In this way, we have shown that $\mathcal{U}$ is an upper bound of $\mathcal{G}$ and $\mathcal{H}$ and now, in order to prove that $\mathcal{U} = \mathcal{G} \vee_t \mathcal{H}$, we have to show that $\mathcal{U}$ is the least upper bound, or supremum, of $\mathcal{G}$ and $\mathcal{H}$. More formally, we have to show that if $\mathcal{F} = (V, E_{\mathcal{F}}, \mathbb{L}_{\mathcal{F}}, \mathbb{E}_{\mathcal{F}})$ is an arbitrary upper bound of $\mathcal{G}$ and $\mathcal{H}$ then $\mathcal{U} \preceq_t \mathcal{F}$. For every upper bound $\mathcal{F}$ of $\mathcal{G}$ and $\mathcal{H}$ it holds that $\mathcal{G}, \mathcal{H} \preceq_t \mathcal{F}$ and, therefore, that

1) both $E_{\mathcal{G}} \subseteq E_{\mathcal{F}}$ and $E_{\mathcal{H}} \subseteq E_{\mathcal{F}}$,

2) both $\mathbb{L}_{\mathcal{G}} \subseteq \mathbb{L}_{\mathcal{F}}$ and $\mathbb{L}_{\mathcal{H}} \subseteq \mathbb{L}_{\mathcal{F}}$,

3) both $\mathbb{E}_{\mathcal{G}} \subseteq \mathbb{E}_{\mathcal{F}}$ and $\mathbb{E}_{\mathcal{H}} \subseteq \mathbb{E}_{\mathcal{F}}$.

In turn, this implies that $E_{\mathcal{G}} \cup E_{\mathcal{H}} \subseteq E_{\mathcal{F}}$, $\mathbb{L}_{\mathcal{G}} \cup \mathbb{L}_{\mathcal{H}} \subseteq \mathbb{L}_{\mathcal{F}}$ and $\mathbb{E}_{\mathcal{G}} \cup \mathbb{E}_{\mathcal{H}} \subseteq \mathbb{E}_{\mathcal{F}}$ and therefore that $\mathcal{U} \preceq_t \mathcal{F}$, as required.

We turn now to the $\hat{0}$ and the $\hat{1}$. It can be easily checked that both $(V,\, \emptyset,\, \emptyset,\, \emptyset)$ and $(V,\, F_V,\, L,\, F_L)$ are compatible and therefore they both belong to $\mathcal{P}$. We have then to show that for every $\mathcal{F} \in \mathcal{P}$ it holds that $\hat{0} \preceq_t \mathcal{F} \preceq_t \hat{1}$ and this follows immediately from the fact the $\emptyset \subseteq E_{\mathcal{F}} \subseteq F_V$, $\emptyset \subseteq \mathbb{L}_{\mathcal{F}} \subseteq L$ and $\emptyset \subseteq \mathbb{E}_{\mathcal{F}} \subseteq F_L$.

Finally, we show that $\langle \mathcal{P}, \preceq_t \rangle$ is distributive, that is the operations of meet and the join distribute over each other. Formally, we have to show that for all $\mathcal{F}, \mathcal{G}, \mathcal{H} \in \mathcal{P}$ it holds that

$$\mathcal{F} \wedge_t (\mathcal{G} \vee_t \mathcal{H}) = (\mathcal{F} \wedge_t \mathcal{G}) \vee_t (\mathcal{F} \wedge_t \mathcal{H})$$

and this follows from the fact that the operations of union and intersection between sets distribute over each other so that

$$\mathcal{F} \wedge_t (\mathcal{G} \vee_t \mathcal{H}) = (V,\, E_{\mathcal{F}} \cap (E_{\mathcal{G}} \cup E_{\mathcal{H}}),\, \mathbb{L}_{\mathcal{F}} \cap (\mathbb{L}_{\mathcal{G}} \cup \mathbb{L}_{\mathcal{H}}),\, \mathbb{E}_{\mathcal{F}} \cap (\mathbb{E}_{\mathcal{G}} \cup \mathbb{E}_{\mathcal{H}}))$$

is equal to

$$\begin{aligned}&(\mathcal{F} \wedge_t \mathcal{G}) \vee_t (\mathcal{F} \wedge_t \mathcal{H})\\ &= (V,\, (E_{\mathcal{F}} \cap E_{\mathcal{G}}) \cup (E_{\mathcal{F}} \cap E_{\mathcal{H}}),\, (\mathbb{L}_{\mathcal{F}} \cap \mathbb{L}_{\mathcal{G}}) \cup (\mathbb{L}_{\mathcal{F}} \cap \mathbb{L}_{\mathcal{H}}),\, (\mathbb{E}_{\mathcal{F}} \cap \mathbb{E}_{\mathcal{G}}) \cup (\mathbb{E}_{\mathcal{F}} \cap \mathbb{E}_{\mathcal{H}})).\end{aligned}$$

E.4 Proof of Proposition 7

We first show (i) that $\mathcal{H} \preceq_s \mathcal{G}$ implies $\mathcal{H} \preceq_t \mathcal{G}$. The pdCGs $\mathcal{H}$ and $\mathcal{G}$ are such that $\mathcal{H} \preceq_s \mathcal{G}$ if and only if conditions (S1), (S2) and (S3) hold true, and therefore we have to prove that the latter three conditions imply (T1), (T1) and (T3). Conditions (S1) and (T1) are trivially equivalent. We now show that (S2) implies (T2). By construction, all the vertices in $\mathbb{L}_{\mathcal{H}}$ belong to atomic colour classes in $\mathcal{V}_{\mathcal{H}}$ and, more precisely, for every $i \in \mathbb{L}_{\mathcal{H}}$ it holds that both $\{i\}$ and $\{\tau(i)\}$ are atomic colour classes in $\mathcal{V}_{\mathcal{H}}$. Hence, by (S2), $\{i\}$ and $\{\tau(i)\}$ are atomic colour classes also in $\mathcal{V}_{\mathcal{G}}$ which implies that $i \in \mathbb{L}_{\mathcal{G}}$ and therefore that $\mathbb{L}_{\mathcal{H}} \subseteq \mathbb{L}_{\mathcal{G}}$ as required. Finally, we show that (S1) and (S3) implies (T3). By construction, every edge $(i, j) \in \mathbb{E}_{\mathcal{H}}$ is such that $\tau(i, j) \in E_{\mathcal{H}}$ and, furthermore, that $\{(i, j)\}$ and $\{\tau(i, j)\}$ are both atomic classes in $\mathcal{E}_{\mathcal{H}}$. It follows by (S1) that $(i, j), \tau(i, j) \in E_{\mathcal{G}}$ and thus, by (S3), that $\{(i, j)\}$ and $\{\tau(i, j)\}$ are both atomic classes also in $\mathcal{E}_{\mathcal{G}}$. In turn, this implies that $(i, j) \in \mathbb{E}_{\mathcal{G}}$ and therefore that $\mathbb{E}_{\mathcal{H}} \subseteq \mathbb{E}_{\mathcal{G}}$.

We now consider the statement (ii) and show it by contradiction. Assume that both $\mathcal{H} \preceq_s \mathcal{G}$ and $\mathcal{H} \prec_t \mathcal{G}$ but $\mathcal{H} \not\prec_s \mathcal{G}$. Then, there exists a graph $\mathcal{F} \in \mathcal{P}$ such that $\mathcal{H} \preceq_s \mathcal{F} \preceq_s \mathcal{G}$ and therefore, by (i) that $\mathcal{H} \preceq_t \mathcal{F} \preceq_t \mathcal{G}$, and this contradicts the assumption that $\mathcal{H} \prec_t \mathcal{G}$.

E.5 Proof of Proposition 8

The relationship between submodels and coloured graphs is given in (S1), (S2) and (S3) of Section 2.3, and we recall that (S1) is in fact redundant. Hence, if we consider a pdCG

$\mathcal{G} \in \mathcal{P}$ then the graph $\mathcal{H} \in \mathcal{P}$ identifies a submodel of $\mathcal{P}(\mathcal{G})$ if and only if the colour classes of $(\mathcal{V}_{\mathcal{H}}, \mathcal{E}_{\mathcal{H}})$ of $\mathcal{H}$ are obtained from the classes $(\mathcal{V}_{\mathcal{G}}, \mathcal{E}_{\mathcal{G}})$ of $\mathcal{G}$ by applying one of the following operations:

(a) Exactly one pair of atomic vertex colour classes of $\mathcal{V}_{\mathcal{G}}$ are merged to obtain a vertex twin-pairing colour class;

(b) exactly one pair of atomic edge colour classes of $\mathcal{E}_{\mathcal{G}}$ are merged to obtain an edge twin-pairing colour class;

(c) exactly one atomic colour class is removed from $\mathcal{E}_{\mathcal{G}}$;

(d) exactly one twin-pairing colour class is removed from $\mathcal{E}_{\mathcal{G}}$;

(e) more than one, not necessarily distinct, of the above operations (a) to (d) are applied.

It is straightforward to see that the graph $\mathcal{H}$ obtained from the application of one of the conditions from (a) to (d) is a neighboring submodel of $\mathcal{G}$, i.e., $\mathcal{H} \prec_s \mathcal{G}$ whereas (e) produces a graph $\mathcal{H} \preceq_s \mathcal{G}$ but such that $\mathcal{H} \not\prec_s \mathcal{G}$. We now consider each of the cases from (a) to (d), in turn, and show that these give the corresponding graphs from (i) to (vii) as given in the text of the proposition. Consider condition (a), and assume, without loss of generality, that $\{i\}, \{\tau(i)\} \in \mathcal{V}_{\mathcal{G}}$ for $i \in \mathbb{L}_{\mathcal{G}}$. If these two classes are merged to obtain $\{i, \tau(i)\} \in \mathcal{V}_{\mathcal{H}}$ then the resulting graph $\mathcal{H}$ is such that $\mathcal{E}_{\mathcal{H}} = \mathcal{E}_{\mathcal{G}}$ so that both $E_{\mathcal{H}} = E_{\mathcal{G}}$ and $\mathbb{E}_{\mathcal{H}} = \mathbb{E}_{\mathcal{G}}$. On the other hand, $\mathbb{L}_{\mathcal{H}} = \mathbb{L}_{\mathcal{G}} \setminus \{i\}$, thereby giving (i). Consider condition (b) and assume, without loss of generality, that $\{(i,j)\}, \{\tau(i,j)\} \in \mathcal{E}_{\mathcal{G}}$ for $(i,j) \in \mathbb{E}_{\mathcal{G}}$. If the latter two classes are merged to obtain $\{(i,j), \tau(i,j)\} \in \mathcal{E}_{\mathcal{H}}$ then the resulting graph $\mathcal{H}$ has the same edge set as $\mathcal{G}$, $E_{\mathcal{H}} = E_{\mathcal{G}}$ and, obviously, also $\mathcal{V}_{\mathcal{H}} = \mathcal{V}_{\mathcal{G}}$ so that $\mathbb{L}_{\mathcal{H}} = \mathbb{L}_{\mathcal{G}}$. Thus, the only difference between $\mathcal{G}$ and $\mathcal{H}$ is that $\mathbb{E}_{\mathcal{H}} = \mathbb{E}_{\mathcal{G}} \setminus \{(i,j)\}$ thereby giving (ii). Consider now condition (c) and assume that the vertex $e \in E_{\mathcal{G}}$ forms an atomic colour class $\{e\} \in \mathcal{E}_{\mathcal{G}}$. Removing $\{e\}$ from $\mathcal{E}_{\mathcal{G}}$ leaves the vertex classes unchanged so that $\mathcal{V}_{\mathcal{H}} = \mathcal{V}_{\mathcal{G}}$ and consequently, $\mathbb{L}_{\mathcal{H}} = \mathbb{L}_{\mathcal{G}}$ whereas the corresponding edge has to be removed so that $E_{\mathcal{H}} = E_{\mathcal{G}} \setminus e$. Furthermore, if $e \neq \tau(e)$ and $\{\tau(e)\} \in \mathcal{E}_{\mathcal{G}}$ then either $e \in \mathbb{E}_{\mathcal{G}}$, so that $\mathbb{E}_{\mathcal{H}} = \mathbb{E}_{\mathcal{G}} \setminus e$, or $\tau(e) \in \mathbb{E}_{\mathcal{G}}$, so that $\mathbb{E}_{\mathcal{H}} = \mathbb{E}_{\mathcal{G}} \setminus \tau(e)$. More specifically, we can consider three different types of atomic edge colour classes. The first type of atomic edge colour class $\{(i,j)\} \in \mathcal{E}_{\mathcal{G}}$ is such that $(i,j) \neq \tau(i,j)$ and $\{\tau(i,j)\} \in \mathcal{E}_{\mathcal{G}}$. Hence, if $(i,j) \in \mathbb{E}_{\mathcal{G}}$ then we obtain (iii) whereas $\tau(i,j) \in \mathbb{E}_{\mathcal{G}}$ gives (iv). The second type of atomic edge colour class $\{(i,j)\} \in \mathcal{E}_{\mathcal{G}}$ is such that $\{\tau(i,j)\} \notin \mathcal{E}_{\mathcal{G}}$ so that $E_{\mathcal{H}} = E_{\mathcal{G}} \setminus \{(i,j)\}$ and $\mathbb{E}_{\mathcal{H}} = \mathbb{E}_{\mathcal{G}}$ as in (v). Finally, the third type of atomic edge colour class has the form $\{(i, \tau(i))\} \in \mathcal{E}_{\mathcal{G}}$ for $i \in V$ so that $E_{\mathcal{H}} = E_{\mathcal{G}} \setminus \{(i, \tau(i))\}$ and $\mathbb{E}_{\mathcal{H}} = \mathbb{E}_{\mathcal{G}}$ as in (vi). We turn now to the case (d). Removing the colour class $\{(i,j), \tau(i,j)\} \in \mathcal{E}_{\mathcal{G}}$ leaves the vertex colour classes unchanged so that $\mathbb{L}_{\mathcal{H}} = \mathbb{L}_{\mathcal{G}}$ and, furthermore, also $\mathbb{E}_{\mathcal{H}} = \mathbb{E}_{\mathcal{G}}$ because both $(i,j) \notin \mathbb{E}_{\mathcal{G}}$ and $\tau(i,j) \notin \mathbb{E}_{\mathcal{G}}$. On the other hand, $E_{\mathcal{G}} \setminus \{(i,j), \tau(i,j)\}$. Hence, in order to obtain (vii) it is sufficient to recall that $\{(i,j), \tau(i,j)\} \in \mathcal{E}_{\mathcal{G}}$ if and only if $i \neq \tau(i)$, both $(i,j) \in E_{\mathcal{G}}$ and $\tau(i,j) \in E_{\mathcal{G}}$ and, furthermore, both $(i,j) \notin \mathbb{E}_{\mathcal{G}}$ and $\tau(i,j) \notin \mathbb{E}_{\mathcal{G}}$.

E.6 Proof of Corollary 10

The first statement can be easily obtained by applying the meet operation as given in point (i) of Theorem 6 to all the pairs of $\preceq_t$–incomparable neighbouring submodels of $\mathcal{G}$ as given in Corollary 9, so as to check that the submodel encoding both the constraints of $\mathcal{H}_1$ and those of $\mathcal{H}_2$ is given by $\mathcal{P}(\mathcal{H}_1 \wedge_t \mathcal{H}_2)$. We turn now to the case where $\mathcal{H}_1$ and $\mathcal{H}_2$ are comparable. As shown in Corollary 9, if $\mathcal{H}_1 \preceq_t \mathcal{H}_2$ then, for a given edge $(i,j) \in \mathbb{E}_{\mathcal{G}}$, $\mathcal{H}_2$ is obtained from (ii) of Proposition 8 and $\mathcal{H}_1$ is obtained from either (iii) or (iv) of the same proposition. If $\mathcal{H}_1$ is obtained from (iii) we let $\mathcal{H}_1'$ be the graph obtained from (iv) whereas if $\mathcal{H}_1$ is obtained from (iv) we let $\mathcal{H}_1'$ be the graph obtained from (iii). Then, one can check that $\mathcal{H}_1 \wedge_s \mathcal{H}_2 = \mathcal{H}_1 \wedge_s \mathcal{H}_1'$ and the result follows because $\mathcal{H}_1$ and $\mathcal{H}_1'$ are $\preceq_t$–incomparable.

E.7 Proof of Corollary 11

First, we notice that the equality $\{\mathcal{F} \wedge_s \mathcal{H} \mid \mathcal{F} \in \mathcal{A} \setminus \{\mathcal{H}\}\} = \{\mathcal{F} \wedge_t \mathcal{H} \mid \mathcal{F} \in \mathcal{A} \setminus \{\mathcal{H}\}\}$ follows immediately from Corollary 10 because we the graphs in $\mathcal{A}$ are pairwise $\preceq_t$–incomparable. Next we show that, (a) for every $\mathcal{F} \in \mathcal{A} \setminus \{\mathcal{H}\}$ it holds that $\mathcal{H} \wedge_t \mathcal{F} \prec_s \mathcal{H}$, and (b) for every $\mathcal{F}_1, \mathcal{F}_2 \in \mathcal{A} \setminus \{\mathcal{H}\}$ it holds that $\mathcal{H} \wedge_t \mathcal{F}_1$ and $\mathcal{H} \wedge_t \mathcal{F}_2$ are $\preceq_t$–incomparable.

We start from (a). Because $\mathcal{A}$ contains neighbouring submodels of $\mathcal{G}$ it follows that both $\mathcal{F}$ and $\mathcal{H}$ are obtained from $\mathcal{G}$ by applying one of the points from (i) to (vii) of Proposition 8. Then, it is easy to check that for all possible combinations of $\mathcal{F}$ and $\mathcal{H}$ it holds that $\mathcal{F} \wedge_t \mathcal{H} \prec_s \mathcal{H}$, with the exception where, for a given edge $(i,j) \in \mathbb{E}_{\mathcal{G}}$, $\mathcal{F}$ and $\mathcal{H}$ are obtained one from (ii) and the other from either (iii) or (iv), but this is not possible because, as shown in Corollary 9, in this case $\mathcal{F}$ and $\mathcal{H}$ would not be $\preceq_t$–incomparable.

We now show (b) contradiction. We have shown above that both $\mathcal{H} \wedge_t \mathcal{F}_1 \prec_s \mathcal{H}$ and $\mathcal{H} \wedge_t \mathcal{F}_2 \prec_s \mathcal{H}$ so that, if we assume that $\mathcal{H} \wedge_t \mathcal{F}_1$ and $\mathcal{H} \wedge_t \mathcal{F}_2$ are $\preceq_t$–comparable, then Corollary 9 implies that, without loss of generality, for a given edge $(i,j) \in \mathbb{E}_{\mathcal{H}}$, $\mathcal{H} \wedge_t \mathcal{F}_1$ is obtained from $\mathcal{H}$ by applying (ii) of Proposition 8 and $\mathcal{H} \wedge_t \mathcal{F}_2$ by either (iii) or (iv) of the same proposition. If $\mathcal{H} \wedge_t \mathcal{F}_2$ is obtained from (iii) then,

$$\mathcal{H} \wedge_t \mathcal{F}_1 = (V, E_{\mathcal{H}} \cap E_1, \mathbb{L}_{\mathcal{H}} \cap \mathbb{L}_1, \mathbb{E}_{\mathcal{H}} \cap \mathbb{E}_1) = (V, E_{\mathcal{H}} \phantom{\setminus \{(i,j)\}}, \mathbb{L}_{\mathcal{H}}, \mathbb{E}_{\mathcal{H}} \setminus \{(i,j)\}), \tag{3}$$

$$\mathcal{H} \wedge_t \mathcal{F}_2 = (V, E_{\mathcal{H}} \cap E_2, \mathbb{L}_{\mathcal{H}} \cap \mathbb{L}_2, \mathbb{E}_{\mathcal{H}} \cap \mathbb{E}_2) = (V, E_{\mathcal{H}} \setminus \{(i,j)\}, \mathbb{L}_{\mathcal{H}}, \mathbb{E}_{\mathcal{H}} \setminus \{(i,j)\}), \tag{4}$$

where $\mathcal{F}_1 = (V, E_1, \mathbb{L}_1, \mathbb{E}_1)$ and $\mathcal{F}_2 = (V, E_2, \mathbb{L}_2, \mathbb{E}_2)$ are both the neighbouring submodels of $\mathcal{G}$. The equation (3) implies that $E_1 = E_{\mathcal{G}}$, $\mathbb{L}_1 = \mathbb{L}_{\mathcal{G}}$ and $\mathbb{E}_1 = \mathbb{E}_{\mathcal{G}} \setminus \{(i,j)\}$ so that $\mathcal{F}_1 = (V, E_{\mathcal{G}}, \mathbb{L}_{\mathcal{G}}, \mathbb{E}_{\mathcal{G}} \setminus \{(i,j)\})$. Moreover, the equation (4) implies that $E_2 = E_{\mathcal{G}} \setminus \{(i,j)\}$, $\mathbb{L}_2 = \mathbb{L}_{\mathcal{G}}$ and $\mathbb{E}_2 = \mathbb{E}_{\mathcal{G}} \setminus \{(i,j)\}$ so that $\mathcal{F}_2 = (V, E_{\mathcal{G}} \setminus \{(i,j)\}, \mathbb{L}_{\mathcal{G}}, \mathbb{E}_{\mathcal{G}} \setminus \{(i,j)\})$. Hence, $\mathcal{F}_1 \preceq_t \mathcal{F}_2$ which contradicts the assumption that $\mathcal{F}_1$ and $\mathcal{F}_2$ are the elements in $\mathcal{A}$ so that they are $\preceq_t$–incomparable. The conclusion is the same when $\mathcal{H} \wedge_t \mathcal{F}_2$ is obtained from (iv).

References

Dvir Aran, Roman Camarda, Justin Odegaard, Hyojung Paik, Boris Oskotsky, Gregor Krings, Andrei Goga, Marina Sirota, and Atul Butte. Comprehensive analysis of normal adjacent to tumor transcriptomes. *Nature Communications*, 8(1):1–14, 2017.

Earl Rodney Canfield. Meet and join within the lattice of set partitions. *The Electronic Journal of Combinatorics*, 8(1):R15, 2001.

Jack Storror Carter, David Rossell, and Jim Q. Smith. Partial correlation graphical LASSO. *Scandinavian Journal of Statistics*, 51(1):32–63, 2024.

Robert Cowell, Philip Dawid, Steffen Lauritzen, and David John Spiegelhalter. *Probabilistic Networks and Expert Systems.* Springer, 1999.

Brian A. Davey and Hilary A. Priestley. *Introduction to Lattices and Order*. Cambridge University Press, 2nd edition, 2002.

Saverio De Vito, Ettore Massera, Marco Piga, Luca Martinotto, and Girolamo Di Francia. On field calibration of an electronic nose for benzene estimation in an urban pollution monitoring scenario. *Sensors and Actuators B: Chemical*, 129(2):750–757, 2008.

David Edwards. *Introduction to Graphical Modelling.* Springer, 2nd edition, 2000.

Sacha Epskamp, Lourens Waldorp, René Mõttus, and Denny Borsboom. The Gaussian graphical model in cross-sectional and time-series data. *Multivariate Behavioral Research*, 53(4):453–480, 2018.

K. Ruben Gabriel. Simultaneous test procedures – some theory of multiple comparisons. *The Annals of Mathematical Statistics*, 40(1):224–250, 1969.

Xin Gao and Hélène Massam. Estimation of symmetry-constrained Gaussian graphical models: application to clustered dense networks. *Journal of Computational and Graphical Statistics*, 24(4):909–929, 2015.

Helene Gehrmann. Lattices of graphical Gaussian models with symmetries. *Symmetry*, 3 (3):653–679, 2011.

Michel Habib, Raoul Medina, Lhouari Nourine, and George Steiner. Efficient algorithms on distributive lattices. *Discrete Applied Mathematics*, 110(2-3):169–187, 2001.

Thomas J. Hardcastle and Krystyna A. Kelly. Empirical Bayesian analysis of paired high-throughput sequencing data with a beta-binomial distribution. *BMC Bioinformatics*, 14 (135):1–11, 2013.

Trevor Hastie, Robert Tibshirani, and Martin Wainwright. *Statistical Learning with Sparsity: The Lasso and Generalizations.* CRC Press, 2015.

Søren Højsgaard and Steffen Lauritzen. Inference in graphical Gaussian models with edge and vertex symmetries with the gRc package for R. *Journal of Statistical Software*, 23(6): 1–26, 2007.

Søren Højsgaard and Steffen Lauritzen. Graphical Gaussian models with edge and vertex symmetries. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(5):1005–1027, 2008.

Steffen Lauritzen. *Graphical Models.* Oxford University Press, 1996.

Qiong Li, Xin Gao, and Hélène Massam. Approximate Bayesian estimation in large coloured graphical Gaussian models. *Canadian Journal of Statistics*, 46(1):176–203, 2018.

Qiong Li, Xin Gao, and Hélène Massam. Bayesian model selection approach for coloured graphical Gaussian models. *Journal of Statistical Computation and Simulation*, 90(14): 2631–2654, 2020.

Qiong Li, Xiaoying Sun, Nanwei Wang, and Xin Gao. Penalized composite likelihood for colored graphical Gaussian models. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 2021.

David Madigan and Adrian Raftery. Model selection and accounting for model uncertainty in graphical models using Occam's window. *Journal of the American Statistical Association*, 89(428):1535–1546, 1994.

Hélène Massam, Qiong Li, and Xin Gao. Bayesian precision and covariance matrix estimation for graphical Gaussian models with edge and vertex symmetries. *Biometrika*, 105 (2):371–388, 2018.

Boris Pittel. Where the typical set partitions meet and join. *The Electronic Journal of Combinatorics*, 7(1):R5, 2000.

Saverio Ranciati and Alberto Roverato. On the application of Gaussian graphical models to paired data problems. *arXiv preprint arXiv:2307.14160*, 2023.

Saverio Ranciati, Alberto Roverato, and Alessandra Luati. Fused graphical lasso for brain networks with symmetries. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 70(5):1299–1322, 2021.

Alberto Roverato and Dung Ngoc Nguyen. Model inclusion lattice of coloured Gaussian graphical models for paired data. In Antonio Salmerón and Rafael Rumí, editors, *Proceedings of the 11th International Conference on Probabilistic Graphical Models*, volume 186 of *Proceedings of Machine Learning Research*, pages 133–144. PMLR, 2022. URL `https://proceedings.mlr.press/v186/roverato22a.html`.

Katherine Tsai, Oluwasanmi Koyejo, and Mladen Kolar. Joint Gaussian graphical model estimation: A survey. *WIREs Computational Statistics*, 14(6):e1582, 2022.

Veronica Vinciotti, Luigi Augugliaro, Antonino Abbruzzo, and Ernst Wit. Model selection for factorial Gaussian graphical models with an application to dynamic regulatory networks. *Statistical Applications in Genetics and Molecular Biology*, 15(3):193–212, 2016.

Joe Whittaker. *Graphical Models in Applied Multivariate Analysis.* John Wiley & Sons, Chichester, 1990.

Ernst Wit and Antonino Abbruzzo. Factorial graphical models for dynamic networks. *Network Science*, 3(1):37—57, 2015.

Yuying Xie, Yufeng Liu, and William Valdar. Joint estimation of multiple dependent Gaussian graphical models with applications to mouse genomics. *Biometrika*, 103(3):493–511, 2016.

Hongmei Zhang, Xianzheng Huang, and Hasan Arshad. Comparing dependent undirected Gaussian networks. *Bayesian Analysis*, pages 1–26, 2022.