Guided structure learning of DAGs for count data

(Article begins on next page)

22 November 2024

# Guided structure learning of DAGs for count data

## Thi Kim Hue Nguyen [1], Monica Chiogna [2], Davide Risso [1] and Erika Banzato [1]

[1] Department of Statistical Sciences, University of Padova, Padova, Italy

[2] Department of Statistical Sciences "Paolo Fortunati", University of Bologna, Bologna, Italy

---

**Address for correspondence:** Nguyen Thi Kim Hue, Department of Statistical Sciences, University of Padova, Via Cesare Battisti, 241, 35121 Padova, Italy.

**E-mail:** nguyen@stat.unipd.it.

**Phone:** (+39) 0498274196.

**Fax:** (+39) 0498274170.

---

**Abstract:** In this paper, we tackle structure learning of Directed Acyclic Graphs (DAGs), with the idea of exploiting available prior knowledge of the domain at hand to guide the search of the best structure. In particular, we assume to know the topological ordering of variables in addition to the given data. We study a new algorithm for learning the structure of DAGs, proving its theoretical consistency in the limit of infinite observations. Furthermore, we experimentally compare the proposed algorithm to several popular competitors, to study its behaviour in finite samples.

Biological validation of the algorithm is presented through the analysis of non-small cell lung cancer data.

---

**Key words:**   Consistency; Directed acyclic graphs; Graphical models; Guided structure learning; Topological ordering

# 1   Introduction

Current demand for modelling complex interactions between variables, combined with the greater availability of high-dimensional discrete data, possibly showing a large number of zeros and measured on a small number of units, has led to an increased focus on structure learning for discrete data in high dimensional settings. In some applications, directed graphs are preferable, as they translate naturally into domain-specific concepts. One notable example are biological pathways, i.e., directed, possibly cyclic, graphs, representing biological systems. It is widely acknowledged that various types of biological pathways, including those related to gene expression, cell signalling, metabolic processes, development, and ecological systems, are used as valuable tools for capturing causal relationships (see, for instance, Palumbo et al., 2006). In these contexts, arrows between two variables picture the information flows from the parent nodes to their descendants. From an alternative viewpoint, and considering solely the realm of probability, directed acyclic graphs (DAGs) serve as graphical tools for depicting the probabilistic conditional independence relationships among variables. Within this framework, the arrows represent associations between variables and allow for the exchange of directional relationships.

When the task at hand involves the estimation from observational data of properties of a system of variables, not allowing for cycles, excluding latent variables and (nonlinear) functional relationships between the variables, the problem is commonly referred to as structure learning of DAGs. Some solutions are nowadays available in the literature for learning (sparse) DAGs for discrete data. Hadiji et al. (2015) introduce a novel family of non-parametric Poisson graphical models, called Poisson Dependency Networks (PDN), trained using functional gradient ascent; Park and Raskutti (2015) define general Poisson DAG models which are identifiable from observational data, and present a polynomial-time algorithm that learns the Poisson DAG model under suitable regularity conditions.

The previously cited approaches, and, more broadly, typical approaches to structure learning of DAGs, usually assume no knowledge of the graph structure to be learned other than sparseness. However, in some contexts, such as that of learning gene networks, a wealth of information is available, usually stored in repositories such as KEGG (Kanehisa and Goto, 2000), about a myriad of interactions, reactions, and regulations. Such information is often identified piecemeal over extended periods and by a variety of researchers, and can therefore be not fully precise. Nevertheless, it allows some topological ordering variables.

When an ordering of variables can be assumed, then the strategy of neighbourhood recovery turns the problem of learning the structure of a DAG into a straightforward task. The graph selection problem is split into a sequence of feature selection problems by assuming that the conditional distribution of each variable given its precedents in the topological ordering follows the chosen distribution. To learn the structure of a DAG, it is sufficient to perform a (sparse) regression for each variable, treating

all preceding variables as covariates. It is known that simply performing lasso-type $\ell_1$-penalized regressions yields consistency for both the coefficients and the sparsity pattern (the set of nonzero coefficients) in regression (Li et al., 2015), and thus, yields consistency for the DAG structure.

The idea of assuming a known ordering of the variables is not novel and several authors have considered decoupling the search over orderings from the graph estimation given the ordering (Friedman and Koller, 2003). However, coming up with a good ordering of variables usually requires a significant amount of domain knowledge, which is not commonly available in many practical applications. As a consequence, various approaches exploiting the topological ordering of the variables implement, in different ways, a search over the space of topological orderings. In the Gaussian setting, Bühlmann et al. (2014) estimate a superset of the skeleton of the underlying DAG, then search a topological ordering using (restricted) maximum likelihood estimation based on an additive structural equation model with Gaussian errors, and finally, exploiting the estimated order of the variables, use sparse additive regression to estimate the functions in an additive structural equation model. Teyssier and Koller (2005) propose to learn a DAG by a search, not over the space of structures, but over the space of orderings, selecting for each ordering  the best consistent network (OS algorithm); Schmidt et al. (2007) couple the OS algorithm with a sparsity-promoting $\ell_1$-regularization.

In this work, we propose one algorithm for structure learning of DAGs for count data which is not affected by the non-uniqueness of the topological ordering and overcomes some of the shortcomings induced by the use of penalized procedures. It is the case that penalization is scale-variant, a condition that often interferes with some of the

filtering steps that are commonly performed in the analysis of complex datasets, such as those arising in genomics. Moreover, it could suffer from the over-shrinking of small but significant covariate effects. Our proposal, named Or-PPGM (PC-based learning of Oriented Poisson Graphical Models), is based on a modification of the PC algorithm (Kalisch and Bühlmann, 2007; Spirtes et al., 1993). In Or-PPGM, we assume to know whether a variable, $i$ say, comes before or after $j$ ($i \neq j$) in an ordering of the available variables that describes the fundamental mechanisms operative in the physical situation. Furthermore, we substitute penalized estimation of the local regressions with a testing procedure on the regression parameters, following the lines of the PC algorithm. Provided that the assumed topological ordering belongs to the space of true topological orderings, we give a theoretical proof of convergence of Or-PPGM that shows that the proposed algorithm consistently estimates the edges of the underlying DAG, as the sample size $n \rightarrow \infty$, irrespective of the choice of the topological ordering. The iterative testing procedure performed within the PC algorithm allows to guarantee scale-invariance of the procedure and avoids over-shrinking of small effects.

The paper is organized as follows. Some essential concepts on DAG models and Poisson DAG models are given in Section 2. Section 3 is devoted to the illustration of the proposed algorithm. We then provide statistical guarantees in Section 4, and, in Section 5, experimental results that illustrate the performance of our methods in finite samples. Section 6 provides an application to gene expression data. Some conclusions and remarks are provided in Section 7. Results needed to prove the main theorem in the paper, another structure learning algorithm and additional simulation results can be found in Supplementary Material.

## 2   Background on Poisson DAG models

In this section, we review, setting up the required notation, some essential concepts on DAG models and present Poisson DAG models according to the model specification introduced by Park and Raskutti (2015).

Consider a $p$-dimensional random vector $\mathbf{X} = (X_1, \ldots, X_p)$ such that each random variable $X_s$ corresponds to a node of a directed graph $G = (V, E)$ with index set $V = \{1, 2, \ldots, p\}$. A directed edge from node $k$ to node $j$ is denoted by $k \to j$, $k$ is called a parent of node $j$, and $j$ is a child of $k$. The set of parents of a vertex $j$, denoted $pa(j)$, consists of all parents of node $j$; its descendants, i.e., nodes that can be reached from $j$ by repeatedly moving from parent to child, are denoted $de(j)$. Non-descendants of $j$ are $nd(j) = V \backslash (\{j\} \cup de(j))$.

A DAG is a directed graph that does not have any directed cycles. In other words, there is no pair $(j, k)$ such that there are directed paths from $j$ to $k$ and from $k$ to $j$. A topological ordering $j_1, \ldots, j_p$ is an order of $p$ nodes such that there are no directed paths from $j_k$ to $j_t$ if $k > t$.

In a DAG, independence is encoded by the relation of d-separation, defined as in Lauritzen (1996). A random vector $\mathbf{X}$ satisfies the local Markov property with respect to (w.r.t.) a DAG $G$ if $X_v \perp\!\!\!\perp \mathbf{X}_{nd(v) \backslash pa(v)} | \mathbf{X}_{pa(v)}$ for every $v \in V$, where $\mathbf{X}_U = \{X_t, t \in U\}$. Similarly, $\mathbf{X}$ satisfies the global Markov property w.r.t. $G$ if $\mathbf{X}_A \perp\!\!\!\perp \mathbf{X}_B | \mathbf{X}_C$ for all triples of pairwise disjoint subsets $A, B, C \subset V$ such that $C$ d-separates $A$ and $B$ in $G$, which we denote by $A \perp\!\!\!\perp_G B | C$. In this work, we make the assumption that the DAG $G$ is a perfect map, i.e., it satisfies the global Markov property and its reverse implication, known as faithfulness. A distribution $P_{\mathbf{X}}$ is said to be faithful to graph

$G$ if $\mathbf{X}_A \perp\!\!\!\perp \mathbf{X}_B | \mathbf{X}_C \Rightarrow A \perp\!\!\!\perp_G B | C$, for all disjoint vertex sets $A, B, C$.

In the Poisson case, the distribution of $\mathbf{X}$ has the form

$$\mathbb{P}_{\boldsymbol{\theta}}(\mathbf{x}) = \prod_{s=1}^{p} \mathbb{P}_{\boldsymbol{\theta}_s}(x_s | \mathbf{x}_{pa(s)}) \tag{2.1}$$

$$= \exp\Big\{ \sum_{s=1}^{p} \sum_{t \in pa(s)} \theta_{st} x_s x_t - \sum_{s=1}^{p} \log(x_s!) - \sum_{s=1}^{p} e^{\sum_{t \in pa(s)} \theta_{st} x_t} \Big\}.$$

where $\mathbf{x}$ is a realization of the random variable $\mathbf{X}$, $\boldsymbol{\theta}_s = \{\theta_{st} | t \in pa(s)\}$, and $\boldsymbol{\theta} = \{\boldsymbol{\theta}_s, s \in V\}$ denotes the set of conditional dependence parameters of the local Poisson regression models characterizing the conditional densities $\mathbb{P}_{\boldsymbol{\theta}_s}(x_s | \mathbf{x}_{pa(s)})$,

$$\mathbb{P}_{\boldsymbol{\theta}_s}(x_s | \mathbf{x}_{pa(s)}) = \exp\Big\{ \sum_{t \in pa(s)} \theta_{st} x_s x_t - \log(x_s!) - e^{\sum_{t \in pa(s)} \theta_{st} x_t} \Big\}.$$

If we zero-pad the parameter $\boldsymbol{\theta}_s \in \mathbb{R}^{|pa(s)|}$ to include zero weights over $V \backslash \{\{s\} \cup pa(s)\}$, then the resulting parameter would lie in $\mathbb{R}^{p-1}$. Therefore, Poisson conditional densities can be written as,

$$\mathbb{P}_{\boldsymbol{\theta}_s}(x_s | \mathbf{x}_{pa(s)}) = \exp\Big\{ \sum_{t \in pa(s)} \theta_{st} x_s x_t - \log(x_s!) - e^{\sum_{t \in pa(s)} \theta_{st} x_t} \Big\} \tag{2.2}$$

$$= \exp\Big\{ x_s \langle \boldsymbol{\theta}_s, \mathbf{x}_{V \backslash \{s\}} \rangle - \log(x_s!) - D(\langle \boldsymbol{\theta}_s, \mathbf{x}_{V \backslash \{s\}} \rangle) \Big\},$$

where $\langle ., . \rangle$ denotes the dot product, and $D(\langle \boldsymbol{\theta}_s, \mathbf{x}_{V \backslash \{s\}} \rangle) = e^{\sum_{t \in V \backslash \{s\}} \theta_{st} x_t}$. This specification puts an edge from node $t$ to node $s$ if $\theta_{st} \neq 0$. A missing edge $t \to s$ corresponds to the condition $\theta_{st} = 0$, implying conditional independence of $X_s$ and $X_t$ given the parents of $s$, i.e., $X_s \perp\!\!\!\perp X_t | \mathbf{x}_{pa(s)}$. As we are only interested in the structure of the graph $G$, without loss of generality we have assumed that the local Poisson regression models characterizing the conditional densities $\mathbb{P}_{\boldsymbol{\theta}_s}(x_s | \mathbf{x}_{pa(s)})$ have zero intercept. Specification (2.2) is similar to that used in Allen and Liu (2013) for the undirected version of Poisson graphical models. The only difference lies in the identification

of the parameter space for $\boldsymbol{\theta}$ that guarantees the existence of the joint distribution. While the distribution represented in (2.1) is always a valid distribution, in the undirected case a joint distribution compatible with the local specifications exists only if all parameters assume non-positive values.

It is worth noting that, to have a perfect map, it is enough to assume faithfulness of the Poisson node conditional distributions to the graph $G$, as this guarantees faithfulness of the joint distributions thanks to the equivalence between local and global Markov property.

# 3   The Or-PPGM algorithm

In this section, we tackle structure learning of DAGs, with the idea of exploiting available prior knowledge of the domain at hand to guide the search for the best structure. In particular, we will assume to know the topological ordering of variables. In what follows, we adopt the convention of using superscripts, e.g., $\mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(n)}$, to denote independent copies of the $p$-random vector $\mathbf{X}$, where $\mathbf{X}^{(i)} = (X_{i1}, \ldots, X_{ip})$. We denote with $\mathbb{X} = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(n)}\}$ the collection of $n$ observed samples drawn from the random vectors $\mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(n)}$, with $\mathbf{x}^{(i)} = (x_{i1}, \ldots, x_{ip})$, $i = 1, \ldots, n$.

Let $i_1, i_2, \ldots, i_p$ indicate one of the possible topological orderings of the variables. The conditional distribution of each variable $X_{i_s}$ given its precedents, denoted $pre(i_s)$, in the topological ordering $i_1, i_2, \ldots, i_p$ can be written as

$$
\begin{aligned}
\mathbb{P}_{\boldsymbol{\theta}_{i_s|pre(i_s)}}(x_{i_s}|\mathbf{x}_{pre(i_s)}) \;=\;\; & \exp\big\{ x_{i_s}\langle \boldsymbol{\theta}_{i_s|pre(i_s)}, \mathbf{x}_{pre(i_s)}\rangle - \log(x_{i_s}!) \\
& - D(\langle \boldsymbol{\theta}_{i_s|pre(i_s)}, \mathbf{x}_{pre(i_s)}\rangle)\big\},
\end{aligned}
\tag{3.1}
$$

where $\boldsymbol{\theta}_{s|\mathbf{K}} = \{\theta_{st|\mathbf{K}} : t \in \mathbf{K}\}$ denote the set of conditional parameters on conditional set $\mathbf{K}$. Then, a rescaled negative node conditional log-likelihood formed by products of all the conditional distributions is as follows

$$
\begin{aligned}
l(\boldsymbol{\theta}_{i_s|pre(i_s)}, \mathbb{X}_{i_s}; \mathbb{X}_{pre(i_s)}) \quad &= -\frac{1}{n}\log\prod_{i=1}^{n}\mathbb{P}_{\boldsymbol{\theta}_{i_s|pre(i_s)}}\big(x_{ii_s}|\mathbf{x}^{(i)}_{pre(i_s)}\big) \\
&= \frac{1}{n}\sum_{i=1}^{n}\Big[-x_{ii_s}\langle\boldsymbol{\theta}_{i_s|pre(i_s)}, \mathbf{x}^{(i)}_{pre(i_s)}\rangle - \log(x_{ii_s}!) \\
&\quad - D(\langle\boldsymbol{\theta}_{i_s|pre(i_s)}, \mathbf{x}^{(i)}_{pre(i_s)}\rangle)\Big].
\end{aligned}
\tag{3.2}
$$

The absence of an edge $t \to i_s$ implies $\theta_{i_st|pre(i_s)}$ to be equal to zero.

When the topological ordering is assumed to be known, the expression of the joint distribution in (2.1) suggests that the structure of the network might be recovered from observed data by disjointly maximizing the single factors in the log-likelihood $\ell(\boldsymbol{\theta}, \mathbb{X})$ since the log-likelihood is decomposable as the sum of partial log-likelihoods over all nodes. Therefore, structure learning could be based on solving local convex optimization problems. Each local estimated conditional dependence parameter $\hat{\boldsymbol{\theta}}_s$ is then combined to form the global estimate.

To tackle this problem, we propose a new algorithm, called Or-PPGM, based on a modification of the well-known PC algorithm (Kalisch and Bühlmann, 2007). Here, we exploit the idea that the consistency of the PC algorithm ultimately depends upon the consistency of the tests of conditional independence employed in the learning process. In our case, consistent tests can be constructed from Wald-type tests on the parameters $\theta_{st|\mathbf{K}}$ (see also Nguyen and Chiogna (2021)). We combine this idea with that of making use of topological ordering to determine the sequence of tests to be performed. Assuming that the order of variables is specified beforehand considerably reduces the number of conditional independence tests to be performed. Indeed, for

each $s \in V$, it is sufficient to test if the data support the existence of the conditional independence relation $X_s \perp\!\!\!\perp X_t | \mathbf{X_S}$ only for $t \in pre(s)$ and for any $\mathbf{S} \subseteq pre(s) \backslash \{t\}$. In detail, we assume that the distribution of each variable $X_s$, conditional to all possible subsets of variables $\mathbf{X_K}$, $\mathbf{K} \subseteq pre(s)$ is a Poisson distribution:

$$X_s | \mathbf{x_K} \sim \text{Pois}\big(\exp\big\{\sum_{t \in \mathbf{K}} \theta_{st|\mathbf{K}} x_t\big\}\big).$$

Then, the algorithm starts from the complete DAG obtained by directing all edges of a complete undirected graph as suggested by the topological ordering. At each level of the cardinality of the conditioning variable set $\mathbf{S}$, we test, at some pre-specified significance level, the null hypothesis $H_0 : \theta_{st|\mathbf{K}} = 0$, where $\mathbf{S} = \mathbf{K} \backslash \{s\}$. If the null hypothesis is not rejected, the edge $t \rightarrow s$ is considered to be absent from the graph. We note that the cardinality of the set $\mathbf{S}$ increases from 0 to $\min\{ord(s) - 1, m\}$, where $ord(s)$ is the position of node $s$ in the topological ordering and $m$ an upper bound on the cardinality of conditional sets. It is worth noting that the value of $m$ is chosen based on prior knowledge about the sparsity of the graph. In the case of no prior knowledge, it will be set to $p - 2$. For a description of the conditional independence test, as well as the definition of an appropriate test statistic, we refer readers to Nguyen and Chiogna (2021). The pseudo-code of the Or-PPGM algorithm is given in Algorithm 1.

## 4    Statistical Guarantees

In this section, we address the property of statistical consistency of Or-PPGM. In detail, we study the limiting behaviour of our estimation procedure as the sample size $n$, and the model size $p$ go to infinity. In what follows, we derive uniform consistency

---

**Algorithm 1** The Or-PPGM algorithm.

---

1: **Input**: $n$ independent realizations of the $p$-random vector $\mathbf{X}$, $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(n)}$; a topological ordering $Ord$, (and a stopping level $m$).

2: **Output**: An estimated DAG $\hat{G}$.

3:  Form the complete undirected graph $\tilde{G}$ on the vertex set $V$.

4:  Orient edges on $\tilde{G}$ respecting the topological ordering to form DAG $G'$.

5: $l = -1; \quad \hat{G} = G'$

6: **repeat**

7:    $l = l + 1$

8:    **for** all vertices $s \in V$, **do**

9:       let $\mathbf{K}_s = pa(s)$

10:    **end for**

11:    **repeat**

12:       Select a (new) edge $t \to s$ in $\hat{G}$ such that

13:       $|\mathbf{K}_s \backslash \{t\}| \geq l$.

14:       **repeat**

15:          choose a (new) set $\mathbf{S} \subset \mathbf{K}_s \backslash \{t\}$ with $|\mathbf{S}| = l$.

16:          **if** $H_0 : \theta_{st|\mathbf{K}} = 0$ not rejected

17:             delete edge $t \to s$ from $\hat{G}$

18:          **end if**

19:       **until** edge $t \to s$ is deleted or all $\mathbf{S} \subset \mathbf{K}_s \backslash \{t\}$ with $|\mathbf{S}| = l$ have been considered.

20:    **until** all edge $t \to s$ in $\hat{G}$ such that $|\mathbf{K}_s \backslash \{t\}| \geq l$ and $\mathbf{S} \subset \mathbf{K}_s \backslash \{t\}$ with $|\mathbf{S}| = l$ have been tested for conditional independence.

21: **until** $l = m$ or for each edge $t \to s$ in $\hat{G}$: $|\mathbf{K}_s \backslash \{t\}| < l$.

---

of our estimators explicitly as a function of the sample size, $n$, the number of nodes, $p$, (and of $m$) by assuming that the true distribution is faithful to the graph. We acknowledge that our results are based on the work of Yang et al. (2012) for exponential family models, and leverage the proof of Lemma 4 in Kalisch and Bühlmann (2007) in the proof of our main theorem.

For the readers' convenience, before stating the main result, we summarize some notation that will be used throughout this proof. Given a vector $u \in \mathbb{R}^p$, and a parameter $q \in [0, \infty]$, we write $\|u\|_q$ to denote the usual $\ell_q$ norm. Given a matrix $A \in \mathbb{R}^{p \times p}$, denote the largest and smallest eigenvalues as $\Lambda_{\max}(A)$, $\Lambda_{\min}(A)$, respectively. We use $|||A|||_2 = \sqrt{\Lambda_{\max}(A^T A)}$ to denote the spectral norm, corresponding to the largest singular value of $A$, and the $\ell_\infty$ matrix norm is defined as $|||A|||_\infty = \max_{i=1,\ldots,a} \sum_{j=1}^{a} |A_{i,j}|$.

## 4.1    Assumptions

We will begin by stating the assumptions that underlie our analysis, and then give a precise statement of the main results.

Denote the population Fisher information matrix and the sample Fisher information matrix corresponding to the covariates in model (2.2) with $\mathbf{K} = V \backslash \{s\}$ as follows $I_s(\boldsymbol{\theta}_s) = -\mathbb{E}_{\boldsymbol{\theta}} \left( \nabla^2 \log \left( \mathbb{P}_{\boldsymbol{\theta}_s}(X_s | \mathbf{X}_{V \backslash \{s\}}) \right) \right)$, and $Q_s(\boldsymbol{\theta}_s) = \nabla^2 l(\boldsymbol{\theta}_s, \mathbf{X}_s; \mathbf{X}_{V \backslash \{s\}})$. We note that we will consider the problem of maximum likelihood on a closed and bounded dish $\boldsymbol{\Theta} \subset \mathbb{R}^{(p-1)}$. For $\boldsymbol{\theta}_{s|\mathbf{K}} \in \mathbb{R}^{|\mathbf{K}|}$, we can immerse $\boldsymbol{\theta}_{s|\mathbf{K}}$ into $\boldsymbol{\Theta} \subset \mathbb{R}^{(p-1)}$ by zero-pad $\boldsymbol{\theta}_{s|\mathbf{K}}$ to include zero weights over $V \backslash \{\mathbf{K} \cup \{s\}\}$.

**Assumption 1.** *The coefficients $\boldsymbol{\theta}_{s|\mathbf{K}} \in \boldsymbol{\Theta}$ for all sets $\mathbf{K} \subseteq V \backslash \{s\}$ and all $s \in V$ have*

*an upper bound norm,* $\max_{s,t,\mathbf{K}} |\theta_{st|\mathbf{K}}| \leq M$, *and a lower bound norm,* $\min_{s,t,\mathbf{K}} |\theta_{st|\mathbf{K}}| \geq c,\ \forall t \in \mathbf{K}$.

**Assumption 2.** *The Fisher information matrix corresponding to the covariates in model* (2.2) *with* $\mathbf{K} = V\backslash\{s\}$ *has bounded eigenvalues, i.e., there exists a constant* $\lambda_{\min} > 0$ *such that* $\Lambda_{\min}(I_s(\boldsymbol{\theta}_s)) \geq \lambda_{\min},\ \forall\ \boldsymbol{\theta}_s \in \boldsymbol{\Theta}$. *Moreover, we require that* $\Lambda_{\max}\left(\mathbb{E}_{\boldsymbol{\theta}}\left(\mathbf{X}^T_{V\backslash\{s\}}\mathbf{X}_{V\backslash\{s\}}\right)\right) \leq \lambda_{\max}, \forall s \in V, \forall\ \boldsymbol{\theta} \in \boldsymbol{\Theta}$, *where* $\lambda_{\max}$ *is some constant such that* $\lambda_{\max} < \infty$.

Assumption 1 simply bounds the effects of covariates in all local models. In other words, we consider that the parameters $\theta_{st|\mathbf{K}}$ belong to a compact set bounded by $M$. Note that the upper bound value $M$ can be arbitrarily large. Hence, this assumption does not limit the general applicability of the method. Being the expected value of the rescaled negative log-likelihood twice differentiable, the lower bound on the eigenvalues of the Fisher information matrix in Assumption 2 guarantees strong convexity in all partial models. Condition on the upper eigenvalue of the covariance matrix guarantees that the relevant covariates do not become overly dependent, a requirement which is commonly adopted in these settings.

It is worth noting that for a given topological ordering, consistency of the proposed algorithm requires that condition in Assumption 1 is satisfied only on all subsets $\mathbf{K} \subseteq pre(i_s)$. However, the topological ordering may not be unique and, as a consequence, different topological orderings may lead to different results. To prove consistency uniformly over all topological orderings, a stronger assumption is needed, that requires the condition in Assumption 1 to be satisfied on all subsets $\mathbf{K} \subseteq V\backslash\{s\}$. This is the solution adopted here.

**Assumption 3.** *Suppose* $\mathbf{X}$ *is a p-random vector with node conditional distribution*

*specified in* (2.2). *Then, for any positive constant* $\delta$, *there exists some constant* $c_1 > 0$, *such that* $\mathbb{P}_{\boldsymbol{\theta}_s}(X_s \geq \delta \log n) \leq c_1 n^{-\delta}, \ \forall \ s \in V, \ \forall \ \boldsymbol{\theta}_s \in \boldsymbol{\Theta}$.

**Assumption 4.** *Suppose* $\mathbf{X}$ *is a p-random vector with node conditional distribution specified in* (2.2). *Then, for any* $\boldsymbol{\theta} \in \boldsymbol{\Theta}$, *there exists some positive constants* $\nu, c_2$, *and* $\gamma < 1/3$, *such that* $\mathbb{P}_{\boldsymbol{\theta}}(\nu + \langle \boldsymbol{\theta}, \mathbf{X} \rangle \geq \gamma \log n) \leq c_2 \kappa(n, \gamma)$, *where* $\kappa(n, \gamma) = o_p(n^a)$ *for some* $a < -1$.

The condition on the marginal distribution in Assumption 3 guarantees that the considered variables do not have heavy tails, a common condition permitting to achieve consistency. Assumption 4 specifies the parameter space on which we can prove the consistency of local estimators. Compared to Assumption 5 in Yang et al. (2012) and Condition 4 in Yang et al. (2015), Assumption 4 appears to be much weaker. Indeed, Yang et al. (2012) require $\gamma < \dfrac{1}{4}$ and $\|\boldsymbol{\theta}\|_2 \leq \dfrac{\log n}{18 \log(\max\{n, p\})}$, whereas we only require $\gamma < \dfrac{1}{3}$ and no specified bound is put on $\|\boldsymbol{\theta}\|_2$ (since the negative elements of $\boldsymbol{\theta}$ can be arbitrarily small). Moreover, Condition 4 in Yang et al. (2012) is written in analytical form, i.e., a form more restrictive than the probability form here employed.

When conditional dependencies are all positive, a condition also known as "additive relationship" among variables, Assumption 4 also implies the sparsity of the graphs.

## 4.2    Consistency of the Or-PPGM algorithm

DAG models can be defined only up to their Markov equivalence class, a set consisting of all DAGs encoding the same set of conditional independence. Here, we consider a particular parametric distributions, as specified in Equation (2.2). Under this stronger assumption, Poisson DAG models in (2.2) are identifiable, as shown

in Appendix, Theorem A1, where we provide an alternative proof of identifiability benefiting from the ideas developed in the work of Peters and Bühlmann (2013), and avoiding a condition in Park and Raskutti (2015) related to the conditional variance of the general Poisson DAG models (see Theorem 3.1) that become redundant under our specificied Poisson generalized linear model. Identifiability has important consequences in our setting. Indeed, it guarantees that the true graph is unique, and, consequently, that Or-PPGM converges to the true unique graph irrespective of which ordering among the true existing ones is chosen to inform the algorithm.

**Theorem 1.** *Assume 1- 4. Denote by $\hat{G}(\alpha_n)$ the estimator resulting from Algorithm 1, and by $G$ the true graph. Then, there exists a numerical sequence $\alpha_n \longrightarrow 0$, such that $\mathbb{P}_{\boldsymbol{\theta}}(\hat{G}(\alpha_n) = G) = 1, \ \forall \ \boldsymbol{\theta} \in \Omega(\boldsymbol{\Theta})$, when $n \longrightarrow \infty$, where $\Omega(\boldsymbol{\Theta})$ is the space such that the faithfulness assumption is satisfied.*

**Proof.** See Appendix, Section B.

The proof of the above-given Theorem 1 does not depend on which topological order is considered. This implies that, even if for different topological orderings $T_1, T_2, \ldots, T_k$, Algorithm 1 performs different sequences of tests $S_1, S_2, \ldots, S_k$, resulting respectively in estimated graphs $\hat{G}^{T_1}(\alpha_n), \hat{G}^{T_2}(\alpha_n), \ldots, \hat{G}^{T_k}(\alpha_n)$, there exists a numerical sequence $\alpha_n \to 0$, such that the estimators $\hat{G}^T(\alpha_n), \ T = T_1, T_2, \ldots, T_k$ converge to the true unique graph.

It is worth noting that for structure learning of undirected graphs, Nguyen and Chiogna (2021) derived statistical guarantees based on the assumption that the node-wise data generating process belongs to the truncated Poisson distribution. In the case of Poisson node conditional distributions, a proof of consistency of PC-LPGM

with proper test statistic can be provided in the situation of "competitive relationships" between variables, and it is still an unsolved question in the case of unrestricted conditional interaction parameters. Here, we consider DAGs, a situation that guarantees the existence of a joint distribution without the need of restricting conditional interaction parameters, i.e., considering both positive and negative parameters. Moreover, in Nguyen and Chiogna (2021), for each pair of nodes $s$ and $t$, we test $\theta_{st|\mathbf{K}} = 0$, where $\mathbf{K}$ could be all possible subsets of $V \backslash \{s\}$. Here, for each ordered pair of nodes $s$ and $t$, we test $\theta_{st|\mathbf{K}} = 0$, with $\mathbf{K}$ as a subset of $pre(s)$. Therefore, the number of conditional independent tests performing during the run of PC procedure reduces. This difference ensures the validation of the proof of Theorem 1 when moving from undirected graphs to DAGs.

## 5 Empirical study

Here, we empirically evaluate the ability of our proposal to retrieve the true DAG. As a measure of the ability to recover the true structure of the graphs, we adopt three criteria including Precision $P$; Recall $R$; and their harmonic mean, known as $F_1$-score, respectively defined as

$$P = \frac{TP}{TP + FP}, \ R = \frac{TP}{TP + FN}, \ F_1 = 2\frac{P.R}{P + R},$$

where TP (true positive), FP (false positive), and FN (false negative) refer to the number of inferred edges (Liu et al., 2010).

We also aim to compare Or-PPGM to possible contestants. To evaluate the effect of limiting the cardinality of the conditional set, we consider a variant of our proposal, that we call Oriented-Local Poisson Graphical Models (Or-LPGM), that for each

$i_s \in V$ fixes the set of parents of node $i_s$ to be

$$\hat{pa}(i_s) = \{t \in pre(i_s) \text{ such that } H_0 : \ \theta_{i_s t | pre(i_s)} = 0 \text{ is rejected}\}.$$

Moreover, we compare Or-PPGM to several popular competitors. As competitors, we consider structure learning algorithms for both Poisson and non-Poisson variables. Some of the considered competitors are adaptations to our specific setting of established algorithms and are, therefore, firstly scrutinised in this simulation exercise. In detail, as representatives of algorithms for Poisson data, we consider: i) one variant of the K2 algorithm (Cooper and Herskovits, 1992), PKBIC, able to deal with Poisson data and based on a scoring criterion frequently used in model selection (see Supplementary Material, Section B for details); ii) the PDN (Poisson Dependency Networks) algorithm in Hadiji et al. (2015); iii) the overdispersion scoring (ODS) algorithm in Park and Raskutti (2015). It is worth noting that PKBIC is indeed a new structure learning algorithm for Poisson data, whose consistency is proved in Supplementary Material, Section B. Moreover, we consider a structure learning method dealing with the class of categorical data, namely the Max Min Hill Climbing (MMHC) algorithm (Tsamardinos et al., 2006). To apply such algorithms, we categorize our data using Gaussian mixture models on log-transformed data shifted by 1 (Fraley and Raftery, 2002). Finally, taking into account that structure learning for discrete data is usually performed by employing methods for continuous data after suitable data transformation, we consider two representatives of approaches based on the Gaussian assumption, that are, the PC algorithm (Kalisch and Bühlmann, 2007), and the Bayesian network structure learning (Kuipers et al., 2022) using BGe score, applied to log-transformed data shifted by 1.

## 5.1    Data generation

For two different cardinalities, $p = 10$ and $p = 100$, we consider three graphs of different structure:    (i) a scale-free graph, in which the node degree distribution follows a power law; (ii) a hub graph, where each node is connected to one of the hub nodes; (iii) an Erdos-Renyi graph, where the presence of the edges is drawn from independent and identically distributed Bernoulli random variables.  To construct the scale-free and Erdos-Renyi graphs, we employed the R package *igraph* (Csardi et al., 2006). For the scale-free graphs, we followed the Barabasi-Albert model with parameter $power = 0.01, zero.appeal = p$. For the Erdos-Renyi graphs, we followed the Erdos-Renyi model with probability to draw one edge between two vertices $\gamma = 0.2$ for $p = 10$ and $\gamma = 0.02$ for $p = 100$.  To construct the hub graphs, we assumed 2 hub nodes for $p = 10$, and 5 hub nodes for $p = 100$. To convert them into DAGs, we fixed a topological ordering for each graph by taking a permutation of considered variables.  Once the order was defined, undirected edges were oriented to form a DAG. See Figure 1 and 2 for plots of the three chosen DAGs for $p = 10$ and $p = 100$, respectively.

To simulate data, we first construct an adjacency matrix $Adj = (\theta_{ij})$ as follows:

1. fill in the adjacency matrix $Adj$ with zeros;

2. replace every entry corresponding to a directed edge by one;

3. replace each entry equal to 1 with an independent realization from a Uniform random variable $U([-0.5, 0.5])$, representing the true values of parameter $\theta_{st}$.

This yields a matrix $Adj$ whose entries are either zeros or in the range $[-0.5, 0.5]$,
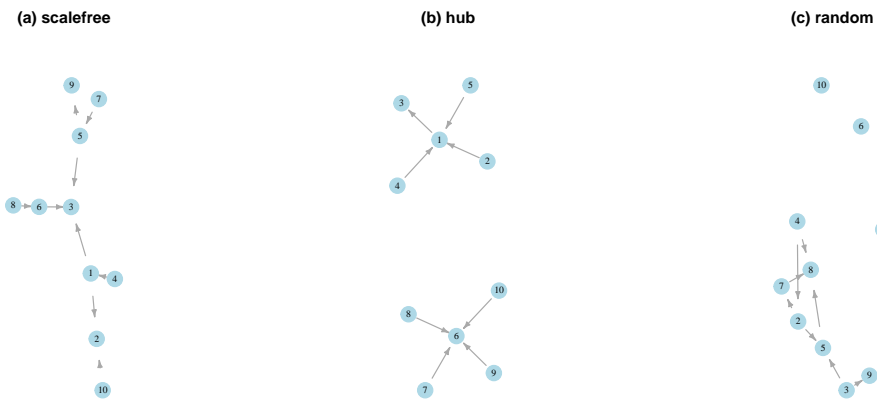
**Figure 1:** The graph structures for $p = 10$ employed in the simulation studies: (a) scale-free; (b) hub; (c) Erdos-Renyi graph.
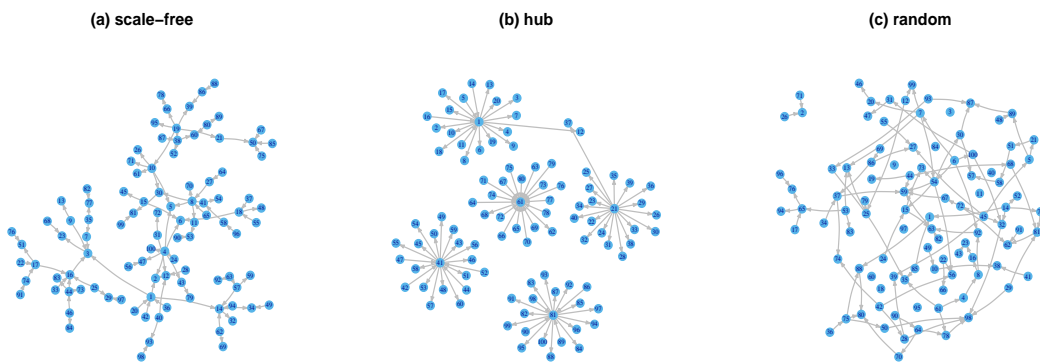


**Figure 2:** The graph structures for $p = 100$ employed in the simulation studies: (a) scale-free; (b) hub; (c) random graph.

representing positive and negative relations among variables. For each DAG corresponding to an adjacency matrix $Adj$, 50 datasets are sampled for four sample sizes, $n = 100, 200, 500, 1000$ with $p = 10$, and $n = 200, 500, 1000, 2000$ with $p = 100$ as follows. The realization of the first random variable $X_{i_1}$ in the topological ordering $i_1, i_2, \ldots, i_p$ is sampled from a $\mathrm{Pois}(\exp\{\theta_1\})$, where the default value of $\theta_1$ is 0. Realizations of the following random variables are recursively sampled from

$$X_{i_j}^{(t)} \sim \mathrm{Pois}(\exp\{\sum_{k=i_1}^{i_{(j-1)}} \theta_{i_j k} x_{tk}\}).$$

## 5.2    Learning algorithms

Acronyms of the considered algorithms are listed below, along with specifications, if needed, of tuning parameters. In this study, besides the topological ordering, we also specify an additional input, the upper limit for the cardinality of conditional sets, $m$, which in this study was set to $m = 8$ for $p = 10$ and $m = 3$ for $p = 100$, respectively.

- **Or-PPGM**: PC-based learning of Oriented Poisson Graphical Models (Section 3);

- **PKBIC**: variant of K2 tailored on Poisson data based on the use of BIC (Supplementary Material, Section B);

- **Or-LPGM**: Oriented Local Poisson Graphical Model, variant of Or-PPGM with no restriction on cardinality of the conditioning set (Section 5);

- **PDN**: Poisson Dependency Networks algorithm (Hadiji et al., 2015), implemented in the R function `learnPDN` (see https://sfb876.tu-dortmund.de/auto?self=%24eon9ai8e80) with n.trees = 20;

- **ODS**: Overdispersion Scoring (ODS) algorithm (Park and Raskutti, 2015) with $k$-fold cross validation ($k = 10$);

- **MMHC**: Max Min Hill Climbing algorithm (Tsamardinos et al., 2006), implemented in the `R` package `bnlearn`, applied to data categorized by mixture models, using $\chi^2$ tests of independence.

- **PC**: PC algorithm (Kalisch and Bühlmann, 2007), implemented in the `R` package `pcalg`, applied to log-transformed data, using Gaussian conditional independent tests.

- **GBiDAG**: Bayesian network structure learning (Kuipers et al., 2022), implemented in the `R` package `BiDAG` (Suter et al., 2023), with an iterative order MCMC algorithm on an expanded search space using BGe score, using the order as an input, and applied to log-transformed data.

We note that ODS, PDN and MMHC employ a preliminary step aimed to estimate the topological ordering. This makes the comparison with our algorithms not completely fair. Nevertheless, we decided to consider these algorithms in our numerical studies to get a measure of the impact of the knowledge of the true topological ordering.

It is also worth noting that the PC algorithm returns PDAGs that consist of both directed and undirected edges. In this case, we borrow the idea of Dor and Tarsi (1992) to extend a PDAG to DAG. This procedure is guaranteed to find a solution for CPDAGs but not for more general PDAGs, as a directed extension of the PDAG may not exist, and the procedure will pick only one DAG from the equivalence class. However, we can prove that the Poisson DAG is identifiable (see Appendix, Theorem A1), i.e., there is a unique DAG equivalent to the set of conditional independence

relations between considered variables. Hence, there is no problem if the procedure picks only one DAG from the equivalence class because the equivalence class has only one element. For details of the algorithm, we refer the interested reader to the paper by Dor and Tarsi (1992).

## 5.3   Results

For the two considered vertex cardinalities, $p = \{10, 100\}$, and the chosen sample sizes, $n = \{100, 200, 500, 1000, 2000\}$, Table 1 and Table 2 report, respectively, Monte Carlo means of TP, FP, FN, P, R and $F_1$ score for each considered method. Each value is computed as an average of the 150 values obtained by simulating 50 samples for each of the three networks. Results disaggregated by network types are given in Supplementary Material, Section D, Tables D.1, and Tables D.2. These results indicate that the proposed algorithm (Or-PPGM), along with Or-LPGM and the modification of K2 (PKBIC) described in Supplementary Material, Section B, outperforms, on average, Gaussian-based competitors (GBiDAG, PC), category-based competitors (MMHC), as well as the state-of-the-art algorithms that are specifically designed for Poisson graphical models (ODS, PDN).

When $p = 10$, the algorithms PKBIC, Or-PPGM and Or-LPGM reach the highest $F_1$ score, followed by the ODS, GBiDAG, and the PC algorithms. When $n \geq 1000$, the three first algorithms recover almost all edges, see Figure 3. A closer look at the Precision $P$ and Recall $R$ plot (see Figure D.1 in Supplementary, Section D.2) provides further insight into the behaviour of considered methods. The PKBIC, Or-LPGM and Or-PPGM algorithms always reach the highest Precision $P$ and Recall $R$.

It is interesting to note that the performance of PKBIC, Or-PPGM and Or-LPGM appears to be far better than that of the competing algorithms employing the Poisson assumption (PDN and ODS). The use of topological ordering overcomes the inaccuracies of the first step of the ODS algorithm, i.e., the identification of the order of variables, as well as the uncertainties in recovering the direction of interactions in PDN.



**Figure 3:** $F_1$-score of the considered algorithms: PKBIC; Or-PPGM; Or-LPGM; PDN; ODS; MMHC; PC; and BiDAG for the three types of graphs in Figure 1 with $p = 10$ and sample sizes $n = 100, 200, 500, 1000$.

When considering other methods, category-based methods (MMHC), and Gaussian-based methods (GBiDAG, PC), both perform less accurately than the three leading methods, i.e., PKBIC, Or-PPGM and Or-LPGM. Moreover, the GBiDAG is the closest method to our proposal, i.e., using the topological ordering as an input to search for the underlying structure. However, this algorithm works well only for the

hub graph with $p = 10$. This result can be explained by the loss of information due to the data transformation, an approach can be ill-suited, possibly leading to wrong inferences in some circumstances (Gallopin et al., 2013). Another variant of BiDAG that uses BDe score with categorical representation proved to be sensitive to the categorisation of the data, and in particular, not effective when the employed categorisation was that of MMHC.
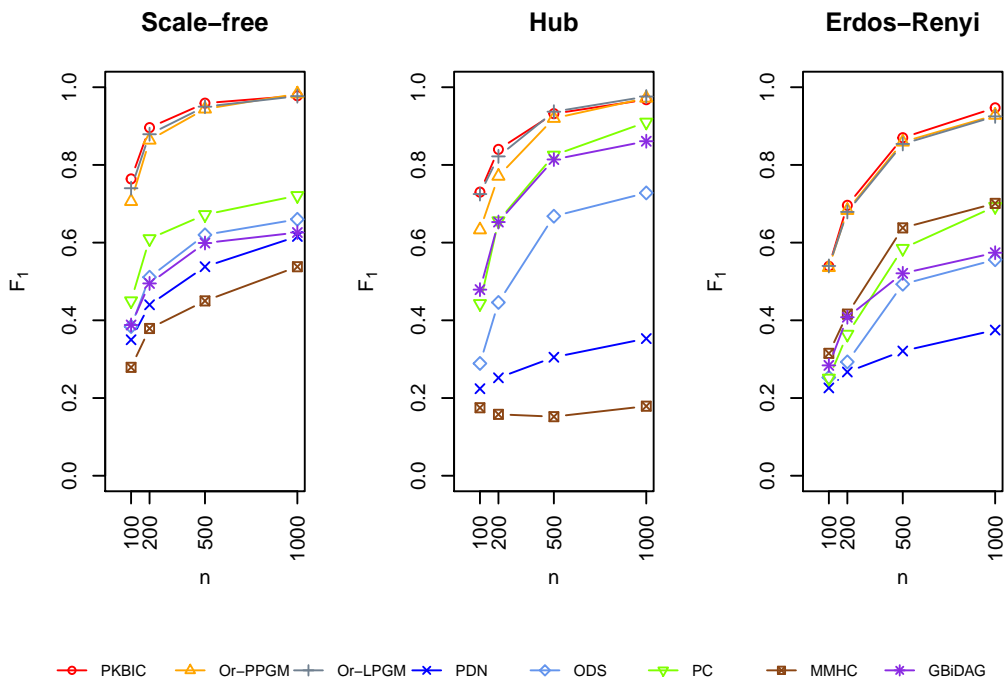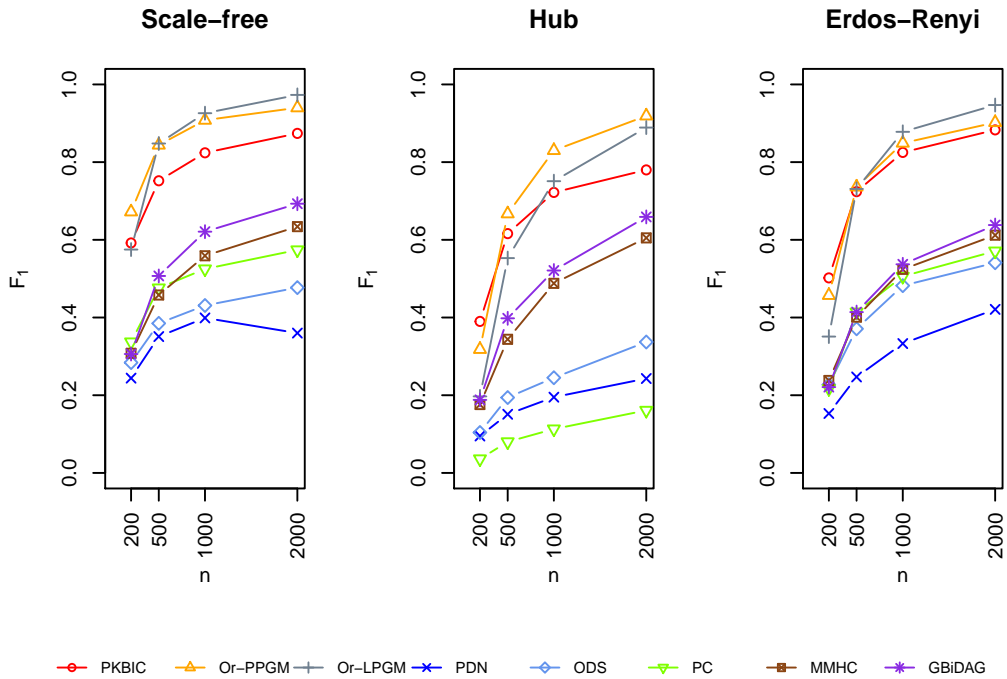


Figure 4: $F_1$-score of the considered algorithms: PKBIC; Or-PPGM; Or-LPGM; PDN; ODS; MMHC; PC; and BiDAG for the three types of graphs in Figure 2 with $p = 100$ and sample sizes $n = 200, 500, 1000, 2000$.

Results for the high dimensional setting ($p = 100$) are somehow comparable to the ones of the previous setting, as it can be seen in Figure 4, and Figure D.2 in Supplementary, Section D.2. The performance of the considered algorithms are clustered into two different groups. In detail, PKBIC, Or-PPGM, and Or-LPGM still rank as the top three best algorithms, with Or-LPGM scoring as the best-performing one for

the highest sample size. Overall, their $F_1$ scores become already reasonable when $n$ approaches 1000 observations.

We also need to stress the good performances of Or-PPGM related to the difference between penalization and restriction of the conditional sets. In the PDN algorithm, as well as in the ODS algorithm, a prediction model is fitted locally on all other variables, using a series of independent penalized regressions. In contrast, Or-PPGM controls the number of variables in the conditional sets for node $s$, which is progressively increased from 0 to $\min\{m, ord(s) - 1\}$.

As a final remark, we note that the performances of ODS are overall less accurate than expected. A reason for it is that ODS uses the LPGM model (Allen and Liu, 2013) to search the candidate parent sets for each node. As a consequence, the performance of ODS is highly dependent on the result obtained by the LPGM algorithm. However, this result depends on the tuning of its parameters ($\beta$, $\gamma$, $sth$, etc). Here, we used the best combination of parameters that we managed to find in Nguyen and Chiogna (2021), i.e., $B = 50$, $nlambda = 20$, $\frac{\lambda_{min}}{\lambda_{max}} = 0.01$, $\gamma = 10^{-6}$, $sth = 0.6$, $\beta = 0.1$ for $p = 10$ and $\beta = 0.05$ for $p = 100$.

# 6 Results on Non-small cell lung cancer data

Here, we show an application of our proposed algorithm to the problem of learning gene interactions starting from gene expression measurements on a set of lung cancer cells.

Specifically, we aim at reconstructing the connected part of the manually curated network in Figure 2c of Xue et al. (2020) from gene expression data, exploiting the topological ordering deriving from the non-small cell lung cancer (Kanehisa and Goto, 2000), which is directly connected to the scope of the original analysis.

Briefly, the data consists of gene expression measurements of individual cells by *RNA sequencing*, which yields discrete counts as a measure of the activity of each gene. We followed the filtering procedure described in the original publication (see Xue et al., 2020, for details).

Xue et al. (2020) identified 10 different clusters of cells based on their sensitivity to treatment. We selected only the cells in clusters 1, 3, 4, 5, and 10 as described in Figure 1 of Xue et al. (2020), which leads to a total of $n = 5505$ cells. These clusters correspond to the cells that showed resistance to the treatment and are of particular biological interest.

The network in Figure 2c of Xue et al. (2020) presents a total of 11 genes, of which, only 8 belong to a connected component of the non-small cell lung cancer pathways, and hence further considered for the analysis, see Figure 5a. It is important to highlight that these 8 genes are part of the initial (upstream) segment of the signaling pathway. These genes are situated at the beginning of the directional flow of information and are not influenced by downstream signaling. This characteristic helps to mitigate concerns related to potential confounding from other genes in the pathway, although it does not solve the issue of relations of this pathway with the other pathways. The topological ordering of the considered genes  has been abstracted from the KEGG pathway database.   In light of the localized nature of this topological assumption, it is important to acknowledge that the ordering may not entirely reflect

the physical properties of the variables involved. However, it does significantly contribute to the estimation process by providing valuable information to the search for associations. Our objective is to assess the capability of our algorithm to identify and reconstruct some of the connections involving these variables that are documented and widely acknowledged in the relevant literature. It is crucial to emphasize, however, that the algorithm's capabilities do not encompass the determination of causality for the estimated links.
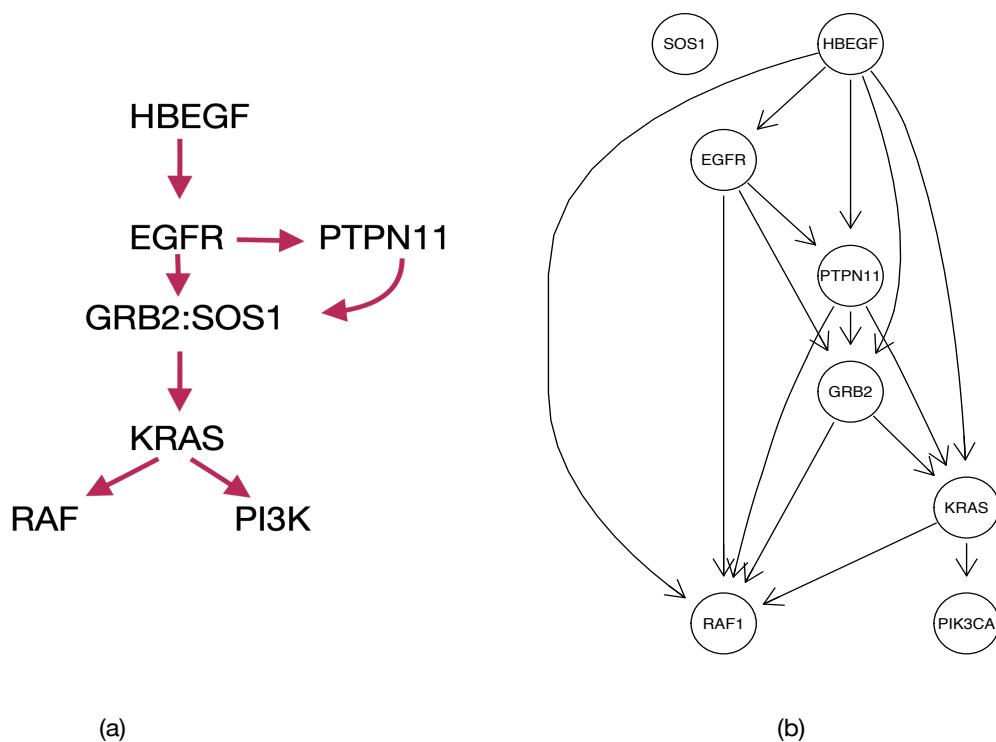


Figure 5: (a) Non-small cell lung cancer network manually curated by Xue et al. (2020); (b) Non-small cell lung cancer network estimated by Or-PPGM algorithm.

After removing values that are more than three standard deviations away from the mean, we applied the Or-PPGM algorithm, with a significance level of 5%, and the results are shown in Figure 5b. By visually comparing our estimated DAG with the manually curated network of Xue et al. (2020), we confirm that our algorithm

can reconstruct, from gene expression data, a biologically meaningful structure that confirms several known biological processes. For instance, the expression of heparin-binding epidermal growth factor (HBEGF) mRNA encodes a ligand of the epidermal growth factor receptor (EGFR) (Lemmon and Schlessinger, 2010). In detail, the ligand drives changes in EGFR depending on the expression of secreted HBEGF, and EGFR plays a potential role in mediating adaptation (Xue et al., 2020). This is consistent with EGFR being a descendant of HBEGF. Moreover, by activating EGFR, the secretion of HBEGF affects a population of cells in an autocrine and/or paracrine fashion, which drives nucleotide exchange to activate RAS. Indeed, Xue et al. (2020) showed that stimulation with recombinant EGF induced KRAS activation in sorted quiescent cells and enhanced signalling in an EGFR- and PTPN11-dependent manner. This is coherent with the path from HBEGF through EGFR and PTPN11 to KRAS. Aside from this, it is well known that GRB2 mediates the EGF-dependent activation of guanine nucleotide exchange on RAS (Gale et al., 1993). The fact that our algorithm reconstructs this known signalling pathway holds promise for novel biological insight that could be provided by inspecting other, lesser-known, gene interactions.

However, we should note that inherent limitations with the available gene expression data prevent us to confidently assert that the estimated associations are causal in nature. Indeed, even if the available set of variables were causally sufficient, it is essential to bear in mind the continuous-time nature of the world when reconstructing causal DAGs from gene expression data. Our guided structure learning strategy is entirely independent of how measurements are made. Consequently, the learned conditional independence relationships may not necessarily reflect the true causal structure. See, for example, Aalen et al. (2016).

# 7   Conclusions and remarks

We have considered structure learning of DAGs for count data in a scenario where we know one possible topological ordering of the variables. We have proposed and compared various guided structure learning algorithms that owe their attractiveness to the improvement in accuracy and the reduction of computational costs due to exploitation of the topological ordering, an ingredient that considerably reduces the search space. For the new proposals, estimators enjoy strong statistical guarantees under assumptions considerably weaker than those employed in related works. Following the empirical comparison with several different approaches, our proposals appear to be promising algorithms as far as prediction accuracy is concerned.

Here, we consider the probabilistic interpretation of DAGs, which differs from the causal interpretation (see Dawid (2010) for more details). Indeed, in the applications that motivate this work, we cannot assume that the set of observed variables is causally sufficient, that is every common direct cause of the observed variables is also observed. Taking this under consideration, we can not guarantee full appropriateness of a causal interpretation (see also Pearl et al. (2000)). Nonetheless, as we show in Section 6, learning DAGs is informative and aids the interpretation of the results compared to undirected graphs, ultimately generating causal hypotheses that can be further explored with subsequent experiments.

It is worth remarking that when $p$ is small (such as $p = 10$), Or-LPGM performs similarly to Or-PPGM while the average of the runtime of Or-PPGM is around 12 times that of Or-LPGM (see Supplementary, Table D.1). Hence, in low-dimensional regimes, Or-LPGM is the worth-considering variant. However, when the number of

variables is large (for example $p = 100$), and the sample size is not large enough (for example $n = 200$), the performance of Or-LPGM is overall less accurate than Or-PPGM (see Supplementary, Table D.2), an effect due to inclusion of covariates which are spuriously related to the outcome reduces the estimated residual variance when the variance is unknown. Here, Or-PPGM is the preferable solution.

Or-PPGM makes the stronger assumption that $X_s$ conditional on all possible subsets of its precedents follows a Poisson distribution, while Or-LPGM relaxes this assumption, requiring that $X_s$ conditional on its precedents is Poisson. However, the stronger assumption of Or-PPGM did not negatively affect the performances of the algorithm on simulations built on Or-LPGM assumptions.

An important side effect of our empirical study was shedding some light on the effect of data transformation finalized to the use of structure learning under model specifications irrespective of the discrete nature of the data, such as those for continuous or categorical data. We have noticed that making the data continuous by log transformation is better than categorizing them when the PC algorithm is used and that mixture-based categorization is better than cut points-based categorization with K2. This is an important empirical conclusion that we draw from this study.

Results from Empirical study showed that the two considered variants of BiDAG (Kuipers et al., 2022) that use BDe or BGe scores perform less accurately than the proposed method because of the loss of information due to the data transformation. In future work, it will be worth considering the BiDAG with the score defined in the PKBIC algorithm, an approach does not suffer from the loss of information due to the data transformation.

Overall, our exploration has consolidated avenues for learning DAGs, the properties and applications of which leave much room for future research. For example, the topological ordering may be misspecified, or only a partial order on the set of nodes might be specified due to several reasons. How to tackle these and other extensions of our setting is the core of our current research.

Table 1: Monte Carlo marginal means of $TP, FP, FN, P, R$, and $F_1$ score obtained by simulating 50 samples from each of the three networks shown in Figure 1 ($p = 10$). The levels of significance of tests $\alpha = 2(1-\Phi(n^{0.15}))$ .

| $n$ | Algorithm | $TP$ | $FP$ | $FN$ | $P$ | $R$ | $F_1$ |
|-----|-----------|------|------|------|-----|-----|-------|
| 100 | PKBIC | 5.133 | 1.393 | 3.200 | 0.791 | 0.613 | 0.678 |
|     | Or-PPGM | 4.573 | 1.380 | 3.760 | 0.774 | 0.546 | 0.625 |
|     | Or-LPGM | 5.200 | 1.740 | 3.133 | 0.758 | 0.621 | 0.669 |
|     | PDN | 6.187 | 32.613 | 2.147 | 0.164 | 0.738 | 0.267 |
|     | ODS | 1.791 | 0.721 | 6.581 | 0.786 | 0.211 | 0.315 |
|     | PC | 2.734 | 2.604 | 5.626 | 0.511 | 0.325 | 0.392 |
|     | MMHC | 1.723 | 3.088 | 6.635 | 0.381 | 0.205 | 0.260 |
|     | GBiDAG | 3.080 | 4.094 | 5.283 | 0.434 | 0.368 | 0.392 |
| 200 | PKBIC | 6.293 | 0.693 | 2.040 | 0.907 | 0.750 | 0.811 |
|     | Or-PPGM | 5.853 | 0.680 | 2.480 | 0.904 | 0.698 | 0.773 |
|     | Or-LPGM | 6.173 | 0.867 | 2.160 | 0.887 | 0.737 | 0.794 |
|     | PDN | 6.820 | 28.820 | 1.513 | 0.201 | 0.814 | 0.320 |
|     | ODS | 2.667 | 1.236 | 5.681 | 0.714 | 0.315 | 0.422 |
|     | PC | 4.062 | 2.000 | 4.283 | 0.654 | 0.484 | 0.550 |
|     | MMHC | 2.329 | 3.859 | 6.007 | 0.392 | 0.276 | 0.319 |
|     | GBiDAG | 4.139 | 3.368 | 4.194 | 0.559 | 0.497 | 0.521 |
| 500 | PKBIC | 7.593 | 0.520 | 0.740 | 0.940 | 0.909 | 0.920 |
|     | Or-PPGM | 7.340 | 0.433 | 0.993 | 0.947 | 0.879 | 0.907 |
|     | Or-LPGM | 7.387 | 0.387 | 0.947 | 0.955 | 0.884 | 0.914 |
|     | PDN | 7.067 | 22.180 | 1.267 | 0.258 | 0.844 | 0.388 |
|     | ODS | 4.347 | 1.813 | 3.987 | 0.714 | 0.520 | 0.593 |
|     | PC | 5.450 | 1.839 | 2.886 | 0.746 | 0.654 | 0.694 |
|     | MMHC | 3.338 | 4.365 | 5.000 | 0.444 | 0.398 | 0.417 |

**Table 1 – continued from previous page**

| $n$ | Algorithm | $TP$ | $FP$ | $FN$ | $P$ | $R$ | $F_1$ |
|---|---|---|---|---|---|---|---|
| | GBiDAG | 5.351 | 2.932 | 2.986 | 0.656 | 0.643 | 0.646 |
| 1000 | PKBIC | 8.093 | 0.353 | 0.240 | 0.962 | 0.970 | 0.964 |
| | Or-PPGM | 7.907 | 0.180 | 0.427 | 0.979 | 0.948 | 0.961 |
| | Or-LPGM | 7.880 | 0.180 | 0.453 | 0.980 | 0.944 | 0.959 |
| | PDN | 7.307 | 17.927 | 1.027 | 0.309 | 0.873 | 0.448 |
| | ODS | 5.213 | 2.527 | 3.120 | 0.681 | 0.625 | 0.648 |
| | PC | 6.233 | 1.513 | 2.100 | 0.805 | 0.749 | 0.775 |
| | MMHC | 4.000 | 4.327 | 4.340 | 0.484 | 0.477 | 0.478 |
| | GBiDAG | 5.799 | 2.772 | 2.537 | 0.681 | 0.698 | 0.688 |

Table 2: Monte Carlo marginal means of $TP, FP, FN, P, R,$ and $F_1$ score obtained by simulating 50 samples from each of the three networks shown in Figure 2 ($p = 100$). The levels of significance of tests $\alpha = 2(1 - \Phi(n^{0.2}))$ for $n = 500, 1000, 2000$, and $\alpha = 2(1 - \Phi(n^{0.225}))$ for $n = 200$.

| $n$ | Algorithm | $TP$ | $FP$ | $FN$ | $P$ | $R$ | $F_1$ |
|---|---|---|---|---|---|---|---|
| 200 | PKBIC | 60.220 | 81.113 | 40.780 | 0.424 | 0.595 | 0.495 |
| | Or-PPGM | 35.307 | 5.320 | 65.693 | 0.854 | 0.349 | 0.482 |
| | Or-LPGM | 26.107 | 6.340 | 74.893 | 0.780 | 0.258 | 0.374 |
| | PDN | 50.580 | 547.487 | 50.420 | 0.128 | 0.498 | 0.164 |
| | ODS | 23.413 | 91.947 | 77.587 | 0.201 | 0.229 | 0.205 |
| | PC | 14.399 | 16.601 | 86.895 | 0.403 | 0.141 | 0.205 |
| | MMHC | 27.527 | 98.873 | 73.473 | 0.216 | 0.272 | 0.241 |
| | GBiDAG | 36.553 | 167.740 | 64.447 | 0.178 | 0.362 | 0.238 |
| 500 | PKBIC | 80.933 | 49.520 | 20.067 | 0.619 | 0.800 | 0.697 |
| | Or-PPGM | 63.180 | 3.333 | 37.820 | 0.950 | 0.625 | 0.749 |
| | Or-LPGM | 58.813 | 2.493 | 42.187 | 0.956 | 0.580 | 0.711 |
| | PDN | 64.773 | 419.773 | 36.227 | 0.216 | 0.637 | 0.250 |
| | ODS | 36.040 | 84.007 | 64.960 | 0.298 | 0.353 | 0.317 |
| | PC | 26.693 | 26.127 | 74.307 | 0.444 | 0.259 | 0.323 |
| | MMHC | 47.993 | 89.340 | 53.007 | 0.348 | 0.474 | 0.401 |
| | GBiDAG | 57.773 | 103.533 | 43.227 | 0.358 | 0.573 | 0.440 |

| n | Algorithm | $TP$ | $FP$ | $FN$ | $P$ | $R$ | $F_1$ |
|---|---|---|---|---|---|---|---|
| 1000 | PKBIC | 89.000 | 34.685 | 12.013 | 0.719 | 0.879 | 0.790 |
| | Or-PPGM | 77.658 | 1.208 | 23.356 | 0.985 | 0.769 | 0.862 |
| | Or-LPGM | 76.153 | 0.200 | 24.847 | 0.997 | 0.751 | 0.852 |
| | PDN | 69.713 | 323.793 | 31.287 | 0.286 | 0.685 | 0.309 |
| | ODS | 45.413 | 83.947 | 55.587 | 0.355 | 0.444 | 0.386 |
| | PC | 34.087 | 32.827 | 66.913 | 0.459 | 0.331 | 0.381 |
| | MMHC | 62.447 | 74.787 | 38.553 | 0.455 | 0.618 | 0.524 |
| | GBiDAG | 69.173 | 77.180 | 31.827 | 0.473 | 0.686 | 0.559 |
| 2000 | PKBIC | 91.833 | 23.713 | 9.167 | 0.794 | 0.906 | 0.846 |
| | Or-PPGM | 87.273 | 1.493 | 13.727 | 0.984 | 0.865 | 0.920 |
| | Or-LPGM | 89.267 | 0.020 | 11.733 | 1.000 | 0.882 | 0.936 |
| | PDN | 67.733 | 237.620 | 33.267 | 0.338 | 0.664 | 0.341 |
| | ODS | 54.767 | 82.687 | 46.233 | 0.398 | 0.538 | 0.452 |
| | PC | 41.400 | 39.180 | 59.600 | 0.478 | 0.402 | 0.435 |
| | MMHC | 72.100 | 60.813 | 28.900 | 0.544 | 0.714 | 0.617 |
| | GBiDAG | 78.420 | 57.313 | 22.580 | 0.579 | 0.778 | 0.663 |

Table 2 – continued from previous page

# 8   Software

The methods presented in this article are available in the *learn2count* R package, available at https://github.com/drisso/learn2count. The code to reproduce the analyses of this paper is available at https://github.com/kimhuenguyen/guided_structure_learning.

# Appendix

# A   Identifiability

In what follows, we provide a proof of identifiability of models specified in Section 2 of the main paper.

**Proposition A1.** *Let* $\mathbf{X}$ *be a p-random vector defined as in* (2.1) *and* $G = (V, E)$ *be a DAG. Consider a variable* $X_j$, $j \in V$, *and one of its parents* $k \in pa_G(j)$. *For all set* $S$ *with* $pa_G(j) \backslash \{k\} \subseteq S \subseteq nd_G(j) \backslash \{k\}$, *we have* $X_j \not\perp\!\!\!\perp X_k | \mathbf{X}_S$.

*Proof.* This proposition can be proved easily by using the definition of d-connection and the faithfulness assumption. Indeed, for a fixed node $j \in V$, for all $k \in pa_G(j)$ and for all set $S$ satisfies $pa_G(j) \backslash \{k\} \subseteq S \subseteq nd_G(j) \backslash \{k\}$, there always exists the path $k \to j$ satisfies the definition of d-connection. Hence, $X_j \not\perp\!\!\!\perp X_k | \mathbf{X}_S$.     □

**Theorem A1.** *The Poisson DAG model defined as in Equation* 2.2 *is identifiable.*

*Proof.* Assume there are two structure models as in 2.2 which both encode the same set of conditional independences, one with graph $G$, and the other with graph $G'$. We will show that $G \equiv G'$.

Since DAGs do not contain any cycles, we can always find one node without any child. Indeed, assume to start at some node, and follow a directed path that contains the chosen node. After at most $|V - 1|$ steps, a node without any child is reached. Eliminating such a node from the graph leads to a new DAG.

We repeat this process on $G$ and $G'$ for all nodes that have: (i) no children, (ii) the same parents in $G$ and $G'$. This process terminates with one of two possible outputs: (a) no nodes left; (b) a subset of variables, which we call again $\mathbf{X}$, two sub-graphs, which we call again $G$ and $G'$, and a node $j$ that has no children in $G$ such that either $pa_G(j) \neq pa_{G'}(j)$ or $ch_{G'}(j) \neq \emptyset$. If (a) occurs, the two graphs are identical and the result is proved. In what follows, we consider the case that (b) occurs.

For such a $j$ node, we have

$$X_j \perp\!\!\!\perp X_{V \backslash (pa_G(j) \cup \{j\})} | \mathbf{X}_{pa_G(j)}, \tag{A.1}$$

thanks to the Markov properties with respect to $G$. To make our argument clear, we divide the set of parents $pa_G(j)$ into three disjoint partitions $W, Y, Z$ representing, respectively, the set of common parents in both graphs; the set of parents in $G$ being a subset of children in $G'$; the set of parents in $G$ which are not parents in $G'$. Formalizing,

- $Z = pa_G(j) \cap pa_{G'}(j)$;

- $Y \subset pa_G(j)$ such that $ch_{G'}(j) = Y \cup T$;

- $W \subset pa_G(j)$ such that $W$ are not adjacent to $j$ in $G'$.

Thus,

$$pa_G(j) = W \cup Y \cup Z, \qquad ch_G(j) = \emptyset,$$

$$pa_{G'}(j) = D \cup Z, \qquad ch_{G'}(j) = T \cup Y,$$

where $D$ is not adjacent to $j$ in $G$. Let $U = W \cup Y$ and consider the following two cases:

- $U = \emptyset$. Then, there exists a node $d \in D$ or a node $t \in T$, otherwise $j$ would have been discarded.

  – If there exists a node $d \in D$, (A.1) implies $X_j \perp\!\!\!\perp X_d | \mathbf{X}_Q$, for $Q = Z \cup D \backslash \{d\}$, which contradicts Proposition (A1) applied to $G'$.

  – If $D = \{\emptyset\}$, and there exists a node $t \in T$, then (A.1) implies $X_j \perp\!\!\!\perp X_t | \mathbf{X}_Q$, for $Q = Z \cup pa_{G'}(t) \backslash \{j\}$, which contradicts Proposition (A1) applied to $G'$.

- $U \neq \emptyset$. We note that, within the structure of the graph $G'$, the Poisson assumption implies

$$\text{Var}\big(X_j | \mathbf{X}_{pa_{G'}(j)}\big) = \mathbb{E}\big(X_j | \mathbf{X}_{pa_{G'}(j)}\big). \tag{A.2}$$

However, by applying the law of total variance we get

$$
\begin{aligned}
\text{Var}\big(X_j | \mathbf{X}_{pa_{G'}(j)}\big) \;=\; & \text{Var}\big(\mathbb{E}(X_j | \mathbf{X}_{pa_{G'}(j)} \cup \mathbf{X}_{pa_G(j)}) | \mathbf{X}_{pa_{G'}(j)}\big) \\
& + \mathbb{E}\big(\text{Var}(X_j | \mathbf{X}_{pa_{G'}(j)} \cup \mathbf{X}_{pa_G(j)}) | \mathbf{X}_{pa_{G'}(j)}\big).
\end{aligned}
$$

By applying Property (A.1) we can rewrite

$$\text{Var}\big(X_j | \mathbf{X}_{pa_{G'}(j)}\big) = \text{Var}\big(\mathbb{E}(X_j | \mathbf{X}_{pa_G(j)}) | \mathbf{X}_{pa_{G'}(j)}\big) + \mathbb{E}\big(\text{Var}(X_j | \mathbf{X}_{pa_G(j)}) | \mathbf{X}_{pa_{G'}(j)}\big). \tag{A.3}$$

Let $f_s(\mathbf{X}_{pa(s)}) = \exp\{\sum_{t \in pa(s)} \theta_{st} X_t\}, \forall\, s \in V$. In graph $G$, we have $X_j | \mathbf{X}_{pa_G(j)} \sim \text{Pois}(f_j(\mathbf{X}_{pa_G(j)}))$, so that

$$\mathbb{E}(X_j | \mathbf{X}_{pa_G(j)}) = \text{Var}(X_j | \mathbf{X}_{pa_G(j)}) = f_j(\mathbf{X}_{pa_G(j)}).$$

Hence, from Equation (A.3), we get

$$
\begin{aligned}
\text{Var}\big(X_j | \mathbf{X}_{pa_{G'}(j)}\big) \;=\; & \text{Var}\big(f_j(\mathbf{X}_{pa_G(j)}) | \mathbf{X}_{pa_{G'}(j)}\big) + \mathbb{E}\big(\mathbb{E}(X_j | \mathbf{X}_{pa_G(j)}) | \mathbf{X}_{pa_{G'}(j)}\big) \quad (A.4) \\
=\; & \text{Var}\big(f_j(\mathbf{X}_{pa_G(j)}) | \mathbf{X}_{pa_{G'}(j)}\big) + \mathbb{E}\big(\mathbb{E}(X_j | \mathbf{X}_{pa_G(j)} \cup \mathbf{X}_{pa_{G'}(j)}) | \mathbf{X}_{pa_{G'}(j)}\big) \\
=\; & \text{Var}\big(f_j(\mathbf{X}_{pa_G(j)}) | \mathbf{X}_{pa_{G'}(j)}\big) + \mathbb{E}\big(X_j | \mathbf{X}_{pa_{G'}(j)}\big),
\end{aligned}
$$

by applying (A.1). Equation (A.4) implies

$$\mathrm{Var}\big(X_j|\mathbf{X}_{pa_{G'}(j)}\big) > \mathbb{E}\big(X_j|\mathbf{X}_{pa_{G'}(j)}\big),$$

since $\mathrm{Var}\big(f_j(\mathbf{X}_{pa_G(j)})|\mathbf{X}_{pa_{G'}(j)}\big) > 0$ in general, except at the root node.

$\square$

# B    Proof of Theorem 1

*Proof.* Given a topological ordering, let $\hat{\theta}_{st|\mathbf{K}}$, and $\theta^*_{st|\mathbf{K}}$ denote the estimated and true partial weights between $X_s$ and $X_t$ given $X_r, r \in \mathbf{S}$, where $\mathbf{S} = \mathbf{K}\backslash\{t\} \subseteq pre(s)$. For a fixed-ordered pair of nodes $s, t$, the conditioning sets are elements of

$$K_{st}^m = \{\mathbf{S} \subseteq pre(s)\backslash\{t\} : |\mathbf{S}| \leq \min\{ord(s) - 1, m\}\}.$$

The cardinality is bounded by

$$|K_{st}^m| \leq Cp^{\min\{ord(s)-1,m\}} \leq Cp^m, \quad \text{for some } 0 < C < \infty.$$

Let $E_{st|\mathbf{K}}$ denote type I or type II errors occurring when testing $H_0: \theta_{st|\mathbf{K}} = 0$. Thus

$$E_{st|\mathbf{K}} = E^I_{st|\mathbf{K}} \cup E^{II}_{st|\mathbf{K}}, \tag{B.1}$$

in which, for $n$ large enough

- type I error $E^I_{st|\mathbf{K}}$: $|Z_{st|\mathbf{K}}| > \Phi^{-1}(1 - \alpha/2)$ and $\theta^*_{st|\mathbf{K}} = 0$;

- type II error $E^{II}_{st|\mathbf{K}}$: $|Z_{st|\mathbf{K}}| \leq \Phi^{-1}(1 - \alpha/2)$ and $\theta^*_{st|\mathbf{K}} \neq 0$;

where $Z_{st|\mathbf{K}} = \dfrac{\sqrt{n}\hat{\theta}_{st|\mathbf{K}}}{\sqrt{\left[J(\hat{\boldsymbol{\theta}}_{s|\mathbf{K}})^{-1}\right]_{tt}}}$ was defined in Nguyen and Chiogna (2021), and $\alpha$ is a

chosen significance level. Consider an arbitrary matrix $\boldsymbol{\theta}_{|\mathbf{K}} = \{\boldsymbol{\theta}_{s|\mathbf{K}}\}_{s\in V}^{T} \in \Omega(\boldsymbol{\Theta})$, such

that $|\theta_{st|\mathbf{K}}| \geq \delta$, for some $\delta > 0$. Let $\boldsymbol{\theta}_{|\mathbf{K}}^0$ be the matrix that has the same elements

as $\boldsymbol{\theta}_{|\mathbf{K}}$ except $\theta_{st|\mathbf{K}} = \theta_{st|\mathbf{K}}^0 = 0$. Choose $\alpha_n = 2(1 - \Phi(n^b))$, where $0 < b < 1/2$ will

be chosen later, then

$$
\begin{aligned}
\sup_{s,t,\mathbf{K}\in K_{st}^m} \mathbb{P}_{\boldsymbol{\theta}_{|\mathbf{K}}^0}(E_{st|\mathbf{K}}^I) &= \sup_{s,t,\mathbf{K}\in K_{st}^m} \mathbb{P}_{\boldsymbol{\theta}_{|\mathbf{K}}^0}\left(|\hat{\theta}_{st|\mathbf{K}}| > n^{b-1/2}\sqrt{\left[J(\hat{\boldsymbol{\theta}}_{s|\mathbf{K}})^{-1}\right]_{tt}}\right) \\
&= \sup_{s,t,\mathbf{K}\in K_{st}^m} \mathbb{P}_{\boldsymbol{\theta}_{|\mathbf{K}}^0}\left(|\hat{\theta}_{st|\mathbf{K}} - \theta_{st|\mathbf{K}}^0| > n^{b-1/2}\sqrt{\left[J(\hat{\boldsymbol{\theta}}_{s|\mathbf{K}})^{-1}\right]_{tt}}\right) \\
&\leq \exp\{-cn\} + c_2 n\kappa(n,\gamma) + c_1 n^{-2},
\end{aligned}
\tag{B.2}
$$

using Theorem C.6, Supplementary, and the fact that $n^{b-1/2}\sqrt{\left[J(\hat{\boldsymbol{\theta}}_{s|\mathbf{K}})^{-1}\right]_{tt}} \longrightarrow 0$ as

$n \longrightarrow \infty$. Furthermore, with the choice of $\alpha_n$ above, and $\delta \geq 2n^{b-1/2}\sqrt{\left[J(\hat{\boldsymbol{\theta}}_{s|\mathbf{K}})^{-1}\right]_{tt}}$,

$$
\begin{aligned}
\sup_{s,t,\mathbf{K}\in K_{st}^m} \mathbb{P}_{\boldsymbol{\theta}_{|\mathbf{K}}}(E_{st|\mathbf{K}}^{II}) &= \sup_{s,t,\mathbf{K}\in K_{st}^m} \mathbb{P}_{\boldsymbol{\theta}_{|\mathbf{K}}}\left(|\hat{\theta}_{st|\mathbf{K}}| \leq n^{b-1/2}\sqrt{\left[J(\hat{\boldsymbol{\theta}}_{s|\mathbf{K}})^{-1}\right]_{tt}}\right) \\
&= \sup_{s,t,\mathbf{K}\in K_{st}^m} \mathbb{P}_{\boldsymbol{\theta}_{|\mathbf{K}}}\left(|\theta_{st|\mathbf{K}}| - |\hat{\theta}_{st|\mathbf{K}}| \geq |\theta_{st|\mathbf{K}}| - n^{b-1/2}\sqrt{\left[J(\hat{\boldsymbol{\theta}}_{s|\mathbf{K}})^{-1}\right]_{tt}}\right) \\
&\leq \sup_{s,t,\mathbf{K}\in K_{st}^m} \mathbb{P}_{\boldsymbol{\theta}_{|\mathbf{K}}}\left(|\theta_{st|\mathbf{K}} - \hat{\theta}_{st|\mathbf{K}}| \geq |\theta_{st|\mathbf{K}}| - n^{b-1/2}\sqrt{\left[J(\hat{\boldsymbol{\theta}}_{s|\mathbf{K}})^{-1}\right]_{tt}}\right) \\
&\leq \sup_{s,t,\mathbf{K}\in K_{st}^m} \mathbb{P}_{\boldsymbol{\theta}_{|\mathbf{K}}}\left(|\hat{\theta}_{st|\mathbf{K}} - \theta_{st|\mathbf{K}}| \geq n^{b-1/2}\sqrt{\left[J(\hat{\boldsymbol{\theta}}_{s|\mathbf{K}})^{-1}\right]_{tt}}\right),
\end{aligned}
$$

Finally, by Theorem C.6, Supplementary, we then obtain

$$
\sup_{s,t,\mathbf{K}\in K_{st}^m} \mathbb{P}_{\boldsymbol{\theta}_{|\mathbf{K}}}(E_{st|\mathbf{K}}^{II}) \leq \exp\{-cn\} + c_2 n\kappa(n,\gamma) + c_1 n^{-2},
\tag{B.3}
$$

as $n \longrightarrow \infty$. Now, by (B.1)-(B.3), we get

$$\mathbb{P}_{\boldsymbol{\theta}}( \text{ a type I or II error occurs in testing procedure})$$

$$\leq \mathbb{P}_{\boldsymbol{\theta}_{|\mathbf{K}}}(\cup_{s,t,\mathbf{K}\in K_{st}^m} E_{st|\mathbf{K}})$$

$$\leq O_p(p^{m+2}) \sup_{s,t,\mathbf{K}\in K_{st}^m} \mathbb{P}_{\boldsymbol{\theta}_{|\mathbf{K}}}(E_{st|\mathbf{K}})$$

$$\leq O_p(p^{m+2}) \left[ \exp\{-cn\} + c_2 n\kappa(n,\gamma) + c_1 n^{-2} \right]$$

$$\to 0,$$

as $n \longrightarrow \infty$.

$\square$

# Acknowledgements

# References

Aalen, O. O., Røysland, K., Gran, J. M., Kouyos, R., and Lange, T. (2016). Can we believe the dags? a comment on the relationship between causal dags and mechanisms. *Statistical methods in medical research*, **25**(5), 2294–2314.

Allen, G. and Liu, Z. (2013). A local Poisson graphical model for inferring networks from sequencing data. *NanoBioscience, IEEE Transactions on*, **12**(3), 189–198.

Bühlmann, P., Peters, J., and Ernest, J. (2014). CAM: causal additive models, high-dimensional order search and penalized regression. *The Annals of Statistics*, **42**(6), 2526–2556.

Cooper, G. F. and Herskovits, E. (1992). A bayesian method for the induction of probabilistic networks from data. *Machine learning*, **9**(4), 309–347.

Csardi, G., Nepusz, T., et al. (2006). The igraph software package for complex network research. *InterJournal, complex systems*, **1695**(5), 1–9.

Dawid, A. P. (2010). Beware of the dag! In *Causality: objectives and assessment*, pages 59–86. PMLR.

Dor, D. and Tarsi, M. (1992). A simple algorithm to construct a consistent extension of a partially oriented graph. *Technicial Report R-185, Cognitive Systems Laboratory, UCLA*.

Fraley, C. and Raftery, A. E. (2002). Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association*, **97**(458), 611–631.

Friedman, N. and Koller, D. (2003). Being bayesian about network structure. a bayesian approach to structure discovery in bayesian networks. *Machine learning*, **50**, 95–125.

Gale, N. W., Kaplan, S., Lowenstein, E. J., Schlessinger, J., and Bar-Sagi, D. (1993). Grb2 mediates the egf-dependent activation of guanine nucleotide exchange on ras. *Nature*, **363**(6424), 88–92.

Gallopin, M., Rau, A., and Jaffrézic, F. (2013). A hierarchical poisson log-normal model for network inference from rna sequencing data. *PloS one*, **8**(10), e77503.

Hadiji, F., Molina, A., Natarajan, S., and Kersting, K. (2015). Poisson dependency networks: Gradient boosted models for multivariate count data. *Machine Learning*, **100**(2-3), 477–507.

Kalisch, M. and Bühlmann, P. (2007). Estimating high-dimensional directed acyclic graphs with the PC-algorithm. *Journal of Machine Learning Research*, **8**(Mar), 613–636.

Kanehisa, M. and Goto, S. (2000). KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Research*, **28**(1), 27–30. ISSN 0305-1048. URL http://www.ncbi.nlm.nih.gov/pubmed/10592173.

Kuipers, J., Suter, P., and Moffa, G. (2022). Efficient sampling and structure learning of bayesian networks. *Journal of Computational and Graphical Statistics*, **31**(3), 639–650.

Lauritzen, S. L. (1996). *Graphical Models*, volume 17. Clarendon Press, Oxford.

Lemmon, M. A. and Schlessinger, J. (2010). Cell signaling by receptor tyrosine kinases. *Cell*, **141**(7), 1117–1134.

Li, Y.-H., Scarlett, J., Ravikumar, P., and Cevher, V. (2015). Sparsistency of $\ell_1$-regularized m-estimators. In *Artificial Intelligence and Statistics*, pages 644–652. PMLR.

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

Nguyen, T. K. H. and Chiogna, M. (2021). Structure learning of undirected graphical models for count data. *Journal of Machine Learning Research*, **22**(50), 1–53. URL http://jmlr.org/papers/v22/18-401.html.

Palumbo, M. C., Farina, L., Colosimo, A., Tun, K., Dhar, P. K., and Giuliani, A. (2006). Networks everywhere? some general implications of an emergent metaphor. *Current Bioinformatics*, **1**(2), 219–234.

Park, G. and Raskutti, G. (2015). Learning large-scale Poisson DAG models based on overdispersion scoring. In *Advances in Neural Information Processing Systems*, pages 631–639.

Pearl, J. et al. (2000). Models, reasoning and inference. *Cambridge, UK: Cambridge-UniversityPress*, **19**(2), 3.

Peters, J. and Bühlmann, P. (2013). Identifiability of Gaussian structural equation models with equal error variances. *Biometrika*, **101**(1), 219–228.

Schmidt, M., Niculescu-Mizil, A., and Murphy, K. (2007). Learning graphical model structure using $\ell_1$-regularization paths. In *AAAI*, volume 7, pages 1278–1283.

Spirtes, P., Glymour, C., and Scheines, R. (1993). Causation, prediction, and search. 1993. *Lecture Notes in Statistics*, **81**.

Suter, P., Kuipers, J., Moffa, G., and Beerenwinkel, N. (2023). Bayesian structure learning and sampling of bayesian networks with the r package bidag. *Journal of Statistical Software*, **105**, 1–31.

Teyssier, M. and Koller, D. (2005). Ordering-based search: A simple and effective algorithm for learning bayesian networks. In *Proceedings of the Twenty-First Con-*

*ference on Uncertainty in Artificial Intelligence*, UAI'05, page 584–590, Arlington, Virginia, USA, 2005. AUAI Press. ISBN 0974903914.

Tsamardinos, I., Brown, L. E., and Aliferis, C. F. (2006). The max-min hill-climbing Bayesian network structure learning algorithm. *Machine learning*, **65**(1), 31–78.

Xue, J. Y., Zhao, Y., Aronowitz, J., Mai, T. T., Vides, A., Qeriqi, B., Kim, D., Li, C., de Stanchina, E., Mazutis, L., Risso, D., and Lito, P. (2020). Rapid non-uniform adaptation to conformation-specific kras (g12c) inhibition. *Nature*, **577**(7790), 421–425.

Yang, E., Allen, G., Liu, Z., and Ravikumar, P. K. (2012). Graphical models via generalized linear models. In *Advances in Neural Information Processing Systems*, pages 1358–1366.

Yang, E., Ravikumar, P., Allen, G. I., and Liu, Z. (2015). Graphical models via univariate exponential family distributions. *Journal of Machine Learning Research*, **16**(1), 3813–3847.