# Psychometric network inference: a comparative analysis

Claudia Delli Colli, Blerina Sinaimeri, Giuseppe F. Italiano, Igor Branchi, Catherine Matias

# PSYCHOMETRIC NETWORK INFERENCE: A COMPARATIVE ANALYSIS

Claudia Delli Colli[1], Blerina Sinaimeri[2], Giuseppe F. Italiano[2], Igor Branchi[1] and Catherine Matias[3]

1. italian institute of health, rome, italy

2. luiss university, rome, italy

3. sorbonne université, université paris cité & cnrs, paris, france

September 18, 2024

This work was partially supported by ERANET Neuron and Istituto Superiore di Sanità, project EnviroMood.

The authors declare no competing financial interests.

The authors thank Aurelia Viglione for her helpful inputs and comments.

Correspondence should be sent to Catherine Matias.
E-Mail: catherine.matias@math.cnrs.fr

**Abstract**

Recently, network-based approaches have provided an important contribution for the understanding of mental disorders. A growing number of statistical models, developed in the context of continuous variables in high-dimensional settings, are currently being used to infer dependencies between network elements (e.g., symptoms or behavioral elements) in psychometrics. However, psychometric datasets typically correspond to low-dimensional statistical settings, namely with a low number of variables collected from a large enough sample size and the variables collected are ordinal rather than Gaussian. In this large-scale simulation study, we tested and compared the performance of 14 methodological approaches including several that, to our knowledge, have never been tested in the context of psychometrics network inference. We assessed the impact of various factors such as the sample size, the number of variables (i.e., network elements), the density of the true underlying graph and the number of ordinal levels. We conclude that the simple and classic statistical methods are undervalued in the current practice, while polychoric correlations appear to have limited additional benefits. We recommend researchers to systematically rely on more than one method in their analyses.

Key words: Gaussian graphical model, Mental health, Network inference, Ordinal data, Psychopathology, Symptom networks

## Introduction

Psychological research examines how complex interactions between biological, psychological, environmental aspects influence health and well-being. To effectively capture and analyze this complexity, network-based approaches have emerged as particularly suitable tools. These models allow researchers to visualize and quantify the relationships between various elements within a system, offering a comprehensive framework to understand how these different elements interact and contribute to overall health outcomes (Borsboom, 2017; Robinaugh et al., 2019; Delli Colli et al., 2024). In the context of psychometrics, network nodes may represent symptoms, affects states or broader behavioral elements, while network edges may represent a relationship such as co-occurrence or correlation (whether direct, partial, positive or negative) between two nodes (Borsboom, 2017; Borsboom et al., 2021; Isvoranu et al., 2022). These relationships are not directly observed but are statistically estimated from data collected through patients' or individuals' responses to questionnaire items or interview questions (Borsboom, 2017).

Among a wide range of statistical models used to represent dependencies between the elements, graphical models are the most commonly used as they provide a visual and intuitive way of understanding complex interactions among multiple variables (Koller and Friedman, 2009). These models show which variables statistically depend on and potentially predict one-another, thereby highlighting potential causal relationships between observed variables (Borsboom, 2017). In psychometric networks, the relationships typically considered are measured through partial correlations, which provide estimates of the relationship between two variables or nodes, controlling for all other nodes in the network.

Over the past twenty years, numerous advancements have been made in network inference through graphical models. However, these developments have predominantly been driven by new demands in bioinformatics (Sinoquet and Mourad, 2014), a field in which the datasets characteristics significantly differ from the ones inherent to psychometric data. Indeed, recent developments in the statistical literature have focused on very high-dimensional (i.e. much more variables than observations) and normally distributed (possibly after transformation) data. Nevertheless, psychometrics data typically lie in small dimensions, with the number of variables ranging from a few

to about a dozen, collected from samples of a few hundred to a few thousand individuals (Hakulinen et al., 2020; McNally et al., 2015; Rhemtulla et al., 2016; Spiller et al., 2017). Moreover, the variables are not continuous (and thus not Gaussian), but rather discrete and finite, and most importantly, are measured on an ordinal scale. This is the case when considering the clinical questionnaires based on the Likert scales (Likert, 1932) used to measure symptoms. For instance, the review by Robinaugh et al. (2019) that focuses on psychopathology mentions "170 empirical articles [that] used network psychometrics to estimate network structure, including 141 articles that examined cross-sectional data in 176 samples (mean [number of individuals] $n = 2169$; median $n = 508$) and 32 articles that examined time-series data in 44 samples (mean $n = 185$; median $n = 76$)". Its supplementary material further contains a table describing a subset of "18 studies estimating the depression symptom network in isolation, [in which] researchers used 12 different pre-existing scales, with the number of symptoms ranging from 9 to 28". Most surprisingly, the low dimensional setting (i.e., low number of variables collected from large sample size) and the ordinal nature of the measurements have been limitedly considered when importing statistical methods of network inference into the psychometrics field.

Among the graphical models, the Gaussian graphical model (GGM), based on an undirected network driving partial correlation coefficients is one of the most popular ways to model statistical relationships between observed variables in the psychometrics field (Epskamp and Fried, 2018; Epskamp et al., 2018; Epskamp, 2020) and as the name indicates, it assumes Gaussian observations. Moreover, recent statistical developments were focused on sparse GGMs (i.e. most of the partial correlations between the variables are equal to zero), giving rise to regularized inference methods such as the famous Graphical Least Absolute Shrinkage and Selection Operator (Glasso) estimator (Meinshausen and Bühlmann, 2006). As an advantage, these methods provide easy-to-interpret results as well as feasible solutions, in particular for high-dimensional settings. However, they come at the cost of a high instability (Meinshausen and Bühlmann, 2010). In the psychometrics context, Epskamp et al. (2018) proposed to input polychoric correlations (Olsson, 1979) inside the EBICglasso algorithm (Foygel and Drton, 2010); this is currently the default estimation method for psychometric networks. It is based on $l_1$ regularization of the maximum likelihood (ML) criterion and provides a parsimonious (i.e., sparse) estimation of the partial correlation structure of

the variables at stake. Let us recall that more generally, regularized methods refer to statistical inference methods that assume that the signal is sparse (i.e., has many zero values), thus producing a sparse estimator; while unregularized methods do not make such an assumption and produce estimated value that will never be exactly zero. As already stressed, psychometric data neither are continuous or Gaussian, nor lie in high-dimensional settings. Nevertheless, the psychometrics literature has mainly focused on sparse GGMs and other regularized methods to infer psychometric networks (Epskamp and Fried, 2018).

Some authors have already pointed out the inadequacy of the above methods due to two fundamental characteristics of psychometric data: the lack of continuity/normality in the observations and the lack of a high-dimensional statistical setting. For example, it has been shown that analyzing ordinal data with metric models can systematically lead to errors (Liddell and Kruschke, 2018). In a similar vein, Williams and Rast (2020) argue that the high-dimensional setting is uncommon in psychological applications and they urge psychometricians to go "back to the basics" when inferring partial correlation networks. In particular, they highlighted that regularization is not required in this context and it may even lead to poor estimation when the setting is not high-dimensional. Another rarely discussed point concerns the advantages offered by polychoric correlations. Indeed, to take into account the discrete nature of the observations, some authors have proposed to replace the classical Pearson correlations by polychoric ones as input in GGMs (Epskamp and Fried, 2018). However, in the statistical literature, it has been advocated that relying on polychoric correlations is inadequate in the case of ordinal variables, as it requires multivariate normality of underlying latent variables and only reflects a linear association (Liu et al., 2021). To finish this list of works questioning the current state-of-the-art, we mention that Lee et al. (2022) not only proposed an ordinal (as opposed to Gaussian) graphical model to better fit psychometrics data but also introduced heterogeneity in the individuals, thus accounting for sub-populations estimated from the data. However, these newer methods at the forefront of research are not yet available as simple packages that could be routinely used by researchers.

Recently, noting the lack of consensus and the numerous variants in psychometric network inference methods, Isvoranu and Epskamp (2023) proposed a first large-scale simulation study aimed to compare the performance of several algorithms. These authors also questioned the use

of polychoric correlations in maximum-likelihood based methods, suggesting that it may be not optimal, particularly when applied to structural equation models. In those models, weighted least squares (WLS) methods better account for sampling variation (Muthén, 1984), whereas polychoric correlations can introduce significant variation, especially with small sample sizes. Additionally, they claim that psychometrics data tend to be skewed, a specificity rarely accounted for in the context of network inference. It is worth noting that transformation of skewed data has been previously discussed in the context of psychometric analysis (Norris and Aroian, 2004).

Our goal is to extend and supplement the study of Isvoranu and Epskamp (2023). First and most importantly, our list of compared methods includes straightforward estimation procedures (namely, in the present article as `poly.mle, poly.wls, pears`) which, though not previously tested within the context of psychometric networks, surprisingly perform well in various situations. Second, we choose to focus on ordinal variables and explore the possible impact of a) the number of levels used to measure the observations, a quantity which, to the best of our knowledge, has never been varied in previous studies; b) the sparsity parameter that rules the number of relationships/interactions in the underlying psychometric graph. Third, while Isvoranu and Epskamp (2023) did not focus on ordinal data and also considered continuous and Gaussian observations, we explore more sophisticate discrete distributions considering heterogeneity in the individuals (see simulation Scenario 4).

In the present contribution the first section describes the methodology including the different simulation scenarios, their driving parameters and the methods we compared. We restrict attention to methods that are implemented in statistical software packages (in fact `R` libraries) and can be routinely used by psychometrics researchers. Additionally, we describe the performance measures selected to assess the quality of each method. The second section describes the main results, followed by the discussion section. We also provide in Appendix A mathematical definitions that should help the reader interested in understanding the statistical details behind the simulations and the methods. The scripts to replicate our experiments are available at `https://github.com/clacollins/PSYCHOMETRIC-NETWORK-INFERENCE`

## Methodology

The three main ingredients of the methodology of this contribution are described in the next subsections: the data simulation scenarios, the methods that we compared and last, the performance measures underlying these comparisons.

### *Simulations*

We considered a sample of $n$ individuals, on which we measure $p$ ordinal variables with $L$ different levels (e.g., answers measured on a Likert scale). Our analysis is conducted in a low-dimensional context, from a classical statistical perspective, where $p$ is much smaller than $n$ ($p << n$). This context is characteristic of psychometric data (Hakulinen et al., 2020). We used sample sizes $n \in \{100; 250; 500; 1,000; 2,500\}$; number of variables $p \in \{6; 7; 8; 9; 10\}$ (and in a specific case, up to $p = 20$) and number of levels $L \in \{4; 5; 6\}$.

To compare different graph inference procedures, we rely on simulations where the ground-truth is controlled. The advantages of using simulated data are that: i) we control exactly the true partial correlations between variables; ii) our analyses are reproducible. Furthermore we can use sophisticate models to mimic psychometric data.

We present 4 different simulation scenarios, whose characteristics are summarized in Table 1. For each scenario, we made 100 experiments and averaged the results over these repetitions. The first and second scenarios both rely on the choice of a polychoric correlation structure between the ordinal variables that we generate. The key difference is that the structure in the first scenario is synthetic, while in the second, it is based on a polychoric correlation structure previously estimated from a real dataset. In the case of a synthetic correlation structure, we varied the density (i.e., the proportion of actual edges number over the maximum possible) of the underlying graph, with values in $\{0.1; 0.2; 0.3; 0.4; 0.5; 1\}$. On the contrary when working with a correlation estimated on a dataset, we chose the simplest method to estimate these polychoric correlations which is not a regularized approach. As a result, we obtained a complete graph structure (i.e., all possible edges are present) with a fixed density of 1. The first scenario corresponds to an *ideal* model, where data has a sparse polychoric partial correlation structure, which aligns with the assumptions

underlying most of the methods. The second scenario, however, was designed to be closer to real-world conditions. The third scenario plays the role of the control scenario: it relies on independent variables, corresponding to an empty graph to infer. This is the case where there is no signal in the data and we expect the methods not to detect spurious relations. The fourth and last scenario is more elaborate and considers a heterogeneous dataset where the individuals come from 2 different unknown populations. In this case, the true underlying structure is unfortunately unknown and we used a proxy for its true value. This scenario is certainly the most realistic among the fourth. The simulation details may be found in Appendix B. In all the scenarios, we computed the average resulting skewness of the distribution of the ordinal variables. As we expect real psychometric dataset to be skewed, our goal is to ensure that we simulated skewed, thus realistic data. The skewness is not part of the initial settings, we rather monitor its values during the experiments.

TABLE 1.

Characteristics of the 4 simulation scenarios.

| Scenario | Data distribution | Expected network density |
|---|---|---|
| 1 | ordinal, with synthetic polychoric correlation structure | `prob` $\in \{0.1; 0.2; 0.3; 0.4; 0.5; 1\}$ |
| 2 | ordinal, with polychoric correlation structure derived from real data | 1 |
| 3 | ordinal, no correlation | 0 |
| 4 | ordinal, no polychoric correlation | 1 |

*Methods compared*

We selected a total of 14 different graph inference methods from the literature, grouped into 4 different categories of approaches for inferring partial correlation graphs. A summary of their characteristics is given in Table 2.

The first group of methods encompasses the simplest and most straightforward approaches, which have been overlooked in the psychometrics network literature. Given the low-dimensional context, these methods involve direct estimation of partial correlations, whether based on polychoric or Pearson's correlations. The second group of methods – relying on ML estimation in GGMs – contains the current default methods used by psychometricians, which consist in functions

TABLE 2.

The 14 methods considered in the present study, divided into 4 groups. We indicate the name of the method used in the study; the name of the R package and principal function used. Note that when needed, transformations from correlations matrices to partial correlations are not explicitly indicated. The fourth column specifies the input of the method, while the fifth gives (whenever applicable) the parameters that have been used. The last column indicates whether the method has been included in Isvoranu and Epskamp (2023).

| Method | R Package | Function | Input | Parameters | New? |
|---|---|---|---|---|---|
| poly.mle | qgraph | cor_auto | raw data | - | Yes |
| poly.wls | psychonetrics | ggm %>% prune %>% modelsearch | raw data | - | No |
| pears | stats | cor | raw data | - | Yes |
| Glasso.poly | qgraph | EBICglasso | poly.mle output | default | No |
| Glasso.pears | qgraph | EBICglasso | pearson output | default | Yes |
| ggmMS.poly | qgraph | ggmModSelect | poly.mle output | default | No |
| ggmMS.pears | qgraph | ggmModSelect | pearson output | default | Yes |
| GGMnr.neighsel | GGMnonreg | ggm_inference | raw data | boot=FALSE | No |
| GGMnr.boot.poly | GGMnonreg | ggm_inference | raw data | default | No |
| GGMnr.boot.pears | GGMnonreg | ggm_inference | raw data | method="polychoric" | Yes |
| BGGM.explore | BGGM | explore | raw data | type="ordinal", impute = FALSE | No |
| BGGM.estimate | BGGM | estimate | raw data | type="ordinal", impute = FALSE | No |
| ggmSS | GeneNet | ggm.estimate.pcor | raw data | default | Yes |
| PAsso | PAsso | PAsso | raw data | default | Yes |

implemented in the `qgraph R` package (Epskamp et al., 2012). The input of these functions can either be polychoric or Pearson's correlation matrix estimators. In the third group of methods, we consider unregularized approaches for GGMs, as proposed by Williams et al. (2019, 2020) and implemented in the `R` package `GGMnonreg` (Williams, 2021b), together with Bayesian GGMs (BGGM) methods introduced by Williams and Mulder (2020); Williams (2021a) and implemented in the `BGGM R` package (Williams and Mulder, 2019). Finally, in the fourth group we explore a method that has produced interesting results in the field of bioinformatics (Schäfer and Strimmer, 2004) and which inspired a new method in the psychometrics context (Liu et al., 2021). The former is implemented in the `R` package `GeneNet` (Schäfer et al., 2021); while the latter is in the `R` package `PAsso` (Zhu et al., 2021), specifically designed for ordinal data. None of these two methods were ever tested before in the context of psychometric data. Details about all the methods, their differences and possibly parameter choices are given in Appendix C.

*Performance measures*

To assess the qualities of each method, we considered a set of performance measures capturing different aspect of the methods' quality.

First, we assessed the recovery of the actual edge weights, namely the correlation coefficients, by measuring the mean-squared error (MSE) between true and estimated values as follows:

$$MSE(\hat{\Theta}; \Theta) = \frac{2}{p(p-1)} \sum_{i=1}^{p} \sum_{j>i} (\hat{\theta}_{ij} - \theta_{ij})^2,$$

where $\hat{\Theta} = (\hat{\theta}_{ij})_{1 \leq i < j \leq p}$ is the matrix of estimated partial correlations and $\Theta$ is the true ones, as specified in each simulation setting. This quantity measures the ability to recover the exact value of each partial correlation coefficient.

Second, we focused on recovering the structure (i.e., the topology) of the partial correlation graph, aiming to accurately identify the presence or absence of edges between variables, without considering the weight of those connections. This means that rather than focusing on the value of each partial correlation coefficient $\theta_{ij}$ we are interested in the binary variable $1\{\theta_{ij} \neq 0\}$ that is 1 whenever the coefficient $\theta_{ij}$ is non zero, and zero otherwise. We thus introduced measures of

binary classification performance. Edges were categorized as true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) as follows:

$$TP = \sum_{i<j} 1\{\hat{\theta}_{ij} \neq 0; \theta_{ij} \neq 0\}, \qquad TN = \sum_{i<j} 1\{\hat{\theta}_{ij} = 0; \theta_{ij} = 0\},$$
$$FP = \sum_{i<j} 1\{\hat{\theta}_{ij} \neq 0; \theta_{ij} = 0\}, \qquad FN = \sum_{i<j} 1\{\hat{\theta}_{ij} = 0; \theta_{ij} \neq 0\}.$$

From these quantities, we obtain sensitivity, specificity and precision as follows

$$\text{sensitivity} = \frac{TP}{TP + FN}; \quad \text{specificity} = \frac{TN}{TN + FP}; \quad \text{precision} = \frac{TP}{TP + FP}.$$

Sensitivity captures the ability of a method to correctly detect a (partial) correlation between symptoms out of pairs of (partially) correlated symptoms, while specificity focuses on the ability to correctly reject (partially) uncorrelated pairs of symptoms. Finally, precision is the fraction of relevant instances (partially correlated pairs of symptoms) among the retrieved instances. Denoting by $|E|$ the cardinality of the number of edges in the true partial correlation graph and $|\hat{E}|$ the same quantity in the estimated graph, we have that

$$\text{sensitivity} = \frac{TP}{|E|}; \quad \text{specificity} = \frac{TN}{p(p-1)/2 - |E|}; \quad \text{precision} = \frac{TP}{|\hat{E}|},$$

where we recall that $p$ is the number of variables/symptoms. Observe that in simulation settings where the true partial correlation graph is complete and thus no edge weight is zero (i.e., Scenario 2, Scenario 4 and Scenario 1 when `prob`=1), we have $\theta_{ij} \neq 0$ for all $i < j$ and thus TN=FP=0 and thus the precision equals 1, the sensitivity equals the density of the estimated graph and the specificity is not defined. In the same way, when considering the independent setting (Scenario 3) where there are no edges in the true partial correlation graph and thus TP=FN=0, sensitivity is undefined, specificity is equal to 1 minus the density of the estimated graph and precision is either zero or undefined.

All these performance measures are computed for each simulation in each scenario and further averaged over the 100 repetitions. We also tracked the number of execution errors encountered by each method, when the function tested did not produce any result. Indeed, some of them failed to process certain datasets and so for each simulation setting, we recorded the count of unsuccessful runs out of the 100 repetitions.

## Results

### *Main findings*

There are three main findings from our simulation study: 1) The most sophisticate methods are not the ones giving the best results, even in ideal scenarios. In particular `poly.wls` performs as well as (and sometimes better than) `Glasso.poly`; 2) polychoric correlations input does not always give better results than relying on simple Pearson's correlations; 3) for real-world datasets, which are usually more complex than ideal cases, no single method consistently performs the best for estimating partial correlation structure. Therefore, we recommend that practitioners use multiple methods and include simple ones such as `poly.wls` and `pears`.

### *Detailed results*

*Scenario 1 - Ideal case.* Our Scenario 1 is the default scenario, that corresponds to the ideal GGM, which is the basis of most of the methods. In this Scenario 1, we first observe that the simulated datasets exhibit a wide range of different skews, either positive or negative, across the different settings. We recall that skewness absolute values within the range of 0.5 and 1 (whether negative or positive) indicate slightly skewed data distributions, while absolute values greater than 1 (negative or positive) correspond to data considered highly skewed. Across the different settings, we observe variables with distributions that range from no skew to slight or high skew, either positively or negatively (see Figure A in Appendix D for an illustration). Based on these observations, the results presented for this scenario are not expected to be biased toward an unrealistic situation.

Out of the 14 methods tested, 7 show execution errors in different settings. More precisely, a first group comprising `GGMnr.boot.poly` and the `BGGM` family often shows a high execution error rate, while a second group with `poly.wls, Glasso.poly, ggmMS.poly` and `PAsso` exhibit a low execution error rate in a small number of settings. However, no specific pattern (with respect to sample size $n$, number of levels $L$, number of variables/symptoms $p$ or density of the true graph `prob`) seems to drive these execution error occurrences, see Figure 1 for an illustration. We also observe that in some settings, the execution error rate may reach 1, meaning that the limits of the method are clearly exceeded. This is the case in particular for `BGGM.explore` and `BGGM.estimate`

(data not shown).

Considering the accuracy of the methods with respect to their MSE, it is important to note that the MSE values (i.e., $y$-scales of the plots) vary significantly across different settings. In general, the mean and the variance of the MSE of the methods decrease with the sample size $n$, and increases with the number of variables/symptoms $p$, and with the density of the true underlying graph. However these values do not appear to be correlated with the number of levels $L$, see Figure 2 for an illustration. Also, this behaviour with respect to $n, p$ and $L$ is observed consistently across all scenarios.

Comparing the MSE of the methods, `Glasso.poly` and `poly.wls` are overall the best methods in this Scenario 1; and in many situations `poly.wls` can be as good and even better than `Glasso.poly`, while being a much simpler approach, see Figure 3. As expected under this scenario, `Glasso.poly` performs better than `Glasso.pears`, with smallest MSE (data not shown). The results also confirm the claim in Muthén (1984) that a WLS approach (i.e., `poly.wls`) is better than a ML based one (i.e., `poly.mle`) for estimating polychoric correlations, see Figure 4. For large sample size ($n$ = 2500), we observe that `poly.wls` is better than (`poly.mle` that is better than `pears`). However, for small sample size ($n$ = 100), the `pears` method seems competitive to `poly.mle`. Surprisingly, in this simulation setting where the true underlying correlations are generated through a polychoric-based approach, inputing polychoric correlations estimates is not always a better approach than relying on Pearson's estimates, see for e.g. Figure 2. In particular, while `Glasso.pears` has high variability and `Glasso.poly` is always better, the methods `ggmMS.poly` and `ggmMS.pears` have similar performances, whereas `GGMnr.boot.poly` performs worse than `GGMnr.boot.pears` (Figure 2). To finish, we observe that the remaining methods, namely the three methods `GGMnr` from the `GGMnonreg` package plus the `BGGM`family and `ggmSS, PAsso`, can be quite competitive in some situations (see for e.g. Figure 2). However no specific pattern may predict when these methods will be appropriate. Moreover, these methods are time consuming (data not shown) and we recall that some of them exhibit execution error rates that prevent their use in a wide range of situations (see Figure 1).

Concerning specificity – the capacity to detect true negatives among unselected edges in the graph – and sensitivity – the ability to detect true positives among all edges in the graph – the

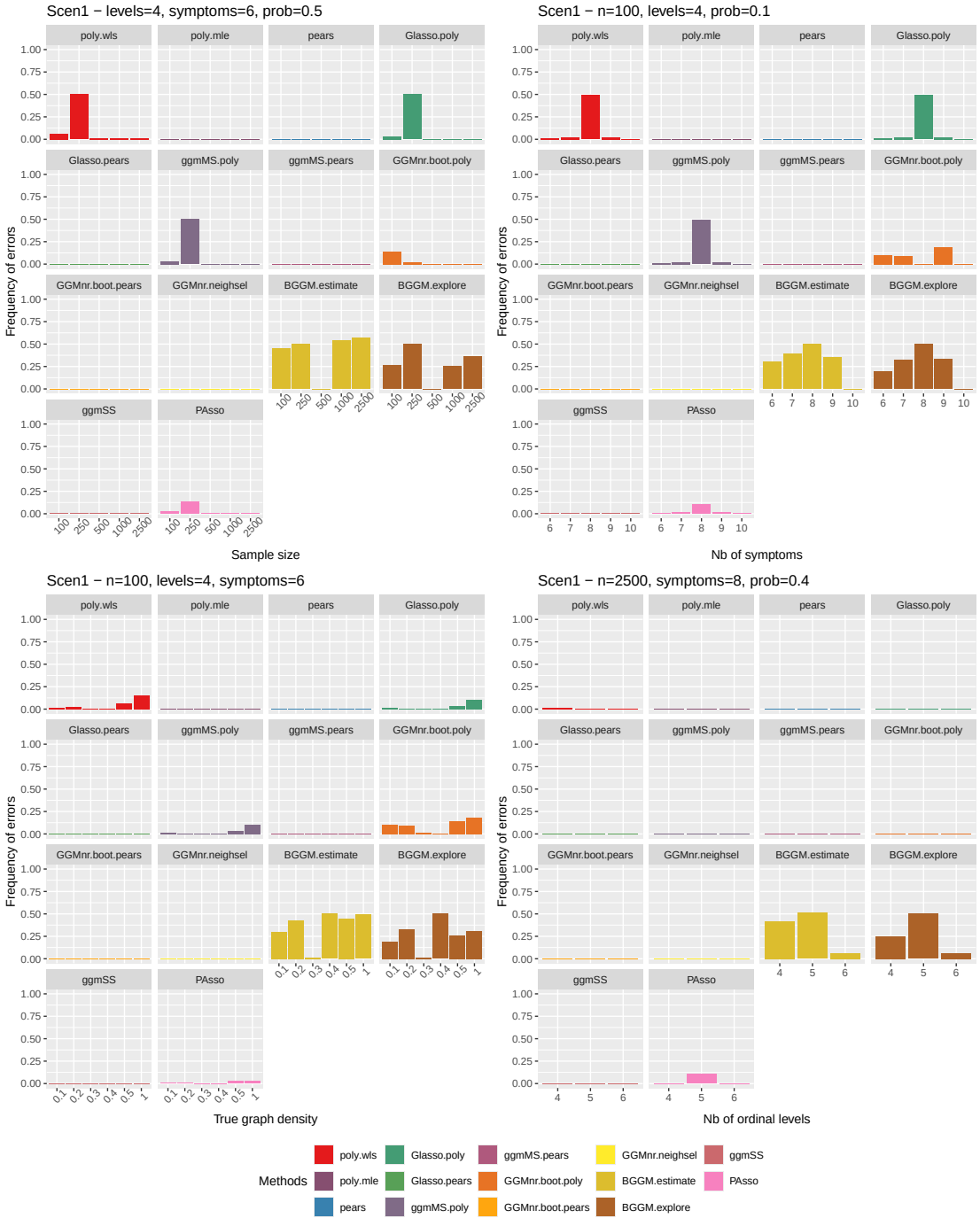


FIGURE 1.

Scenario 1: Examples of the execution error rates (boxplot over 100 replicates) of all the methods (displayed as 14 graphics in each of the 4 pictures) with respect to evolving parameters (displayed on the $x$-axis): sample sizes $n \in \{100; 250; 500; 1,000; 2500\}$ on top left; number of variables $p \in \{6; 7; 8; 9; 10\}$ on top right; density of the true underlying graph `prob`$\in \{0.1; 0.2; 0.3; 0.4; 0.5; 1\}$ on bottom left and number of ordinal levels $L \in \{4, 5, 6\}$ on bottom right.

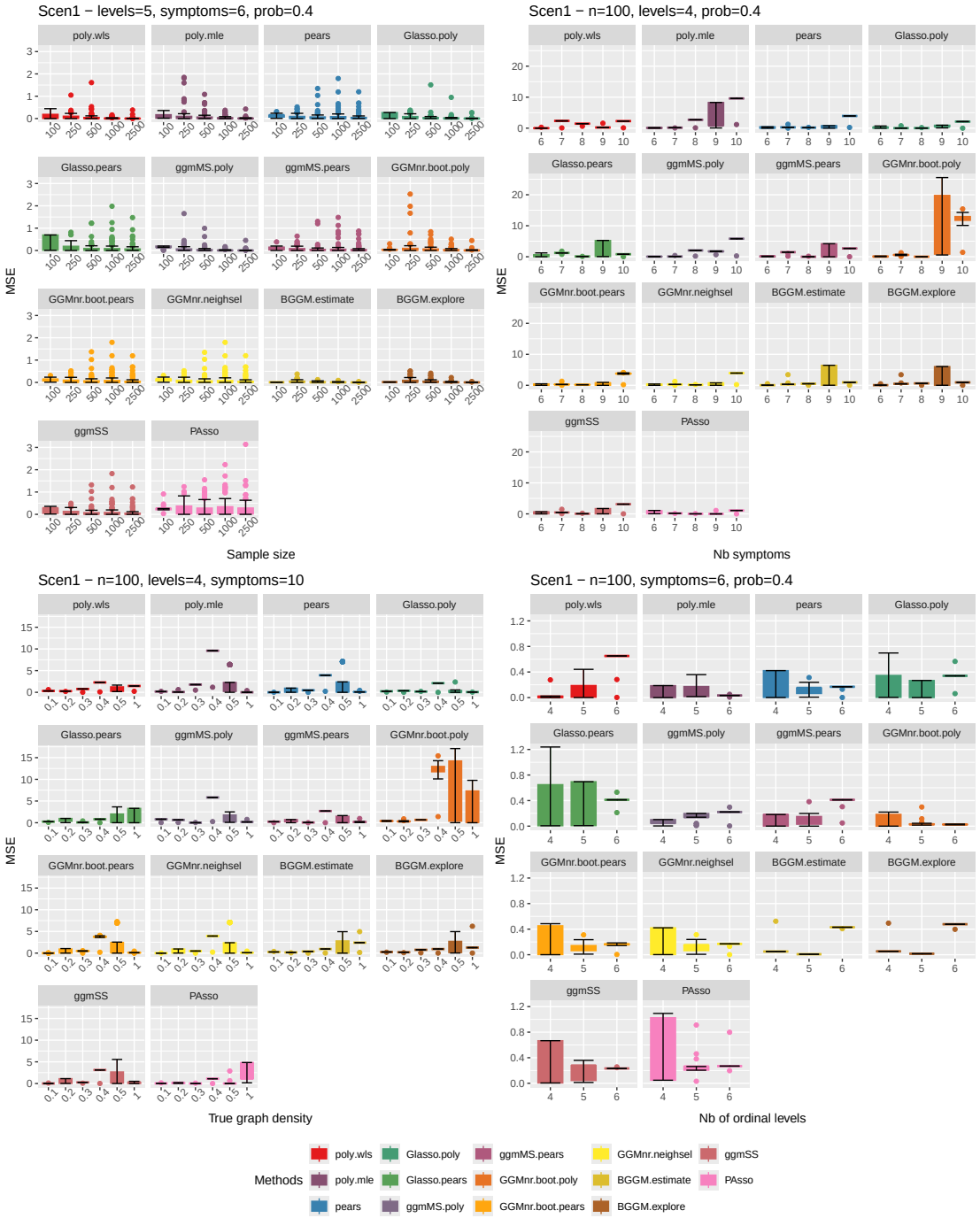


FIGURE 2.

Scenario 1: Examples of the behaviour of the MSE (boxplots over 100 replicates) of all the methods (displayed as 14 graphics in each of the 4 pictures) with respect to evolving parameters (displayed on the $x$-axis): sample sizes $n \in \{100; 250; 500; 1,000; 2500\}$ on top left; number of variables $p \in \{6; 7; 8; 9; 10\}$ on top right; density of the true underlying graph $\texttt{prob} \in \{0.1; 0.2; 0.3; 0.4; 0.5; 1\}$ on bottom left and number of ordinal levels $L \in \{4, 5, 6\}$ on bottom right.

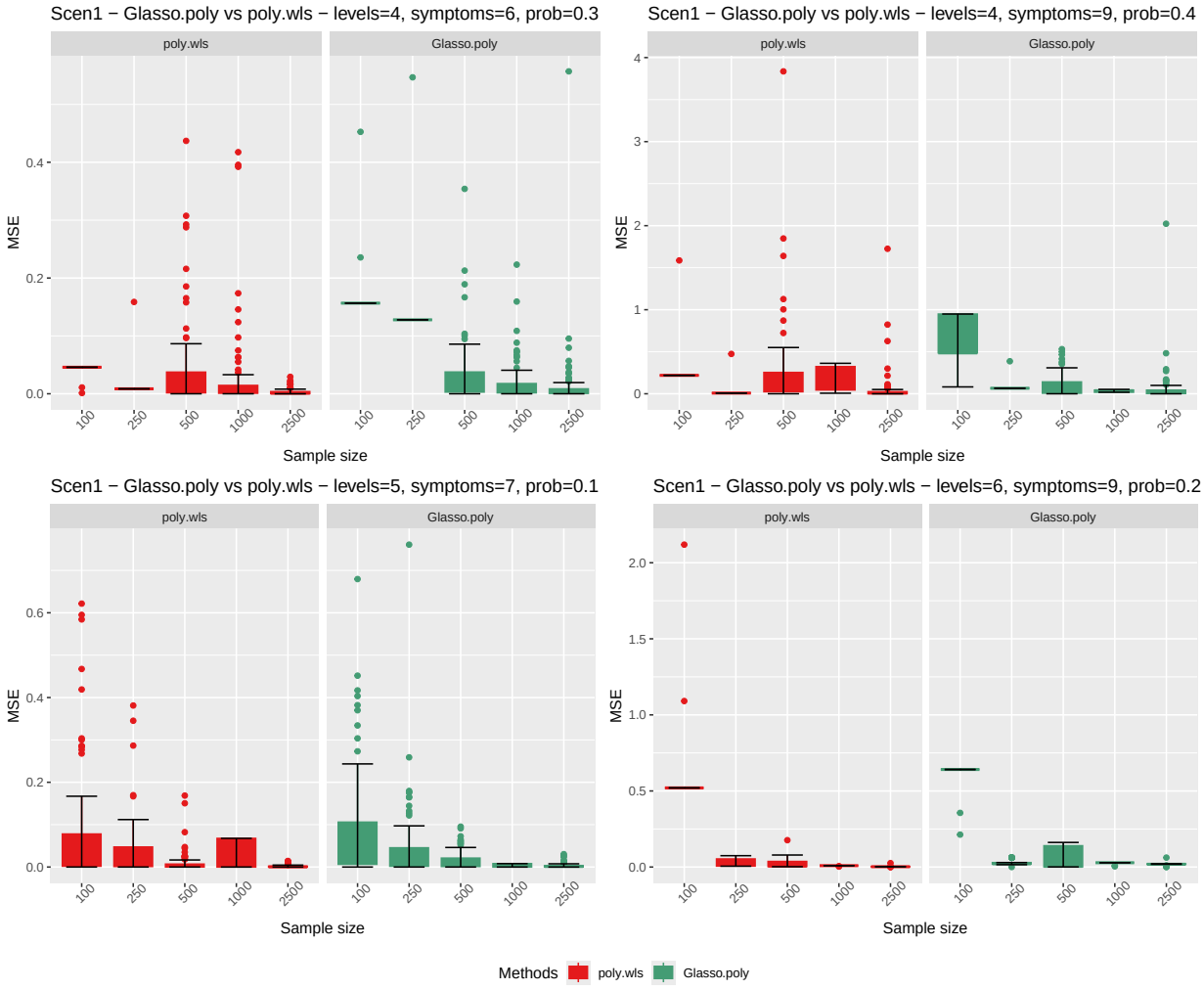


FIGURE 3.

Scenario 1: A selection of settings where `poly.wls` often outperforms `Glasso.poly` in terms of MSE (boxplots over 100 replicates); $x$-axis shows sample sizes $n \in \{100; 250; 500; 1,000; 2500\}$.

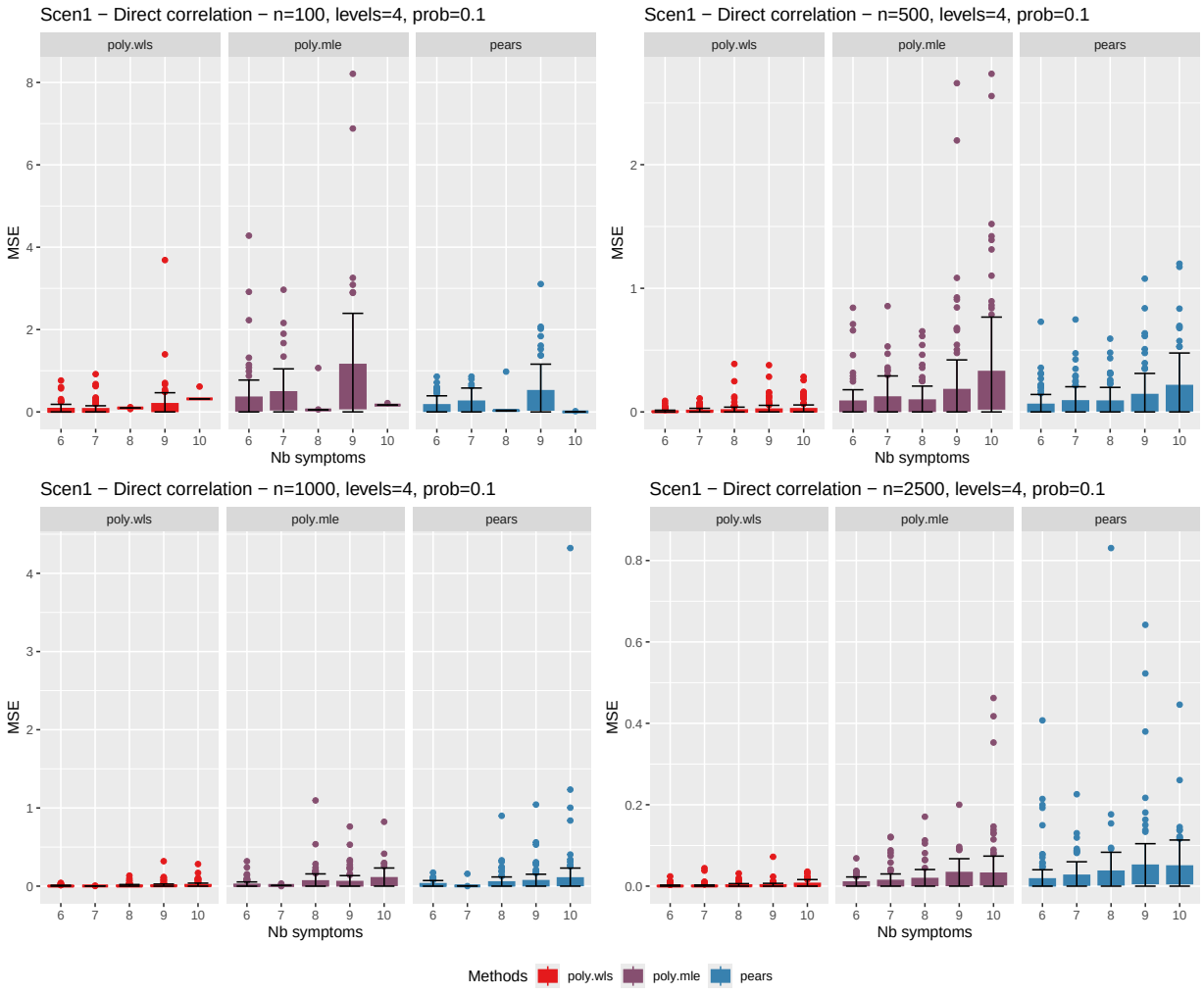


FIGURE 4.

Scenario 1: A selection of settings to compare the performances in terms of MSE (boxplots over 100 replicates) of the direct estimation correlation methods, `poly.wls`,`poly.mle` and `pears`; sample sizes $n$ = 100 (top left), $n$ = 500 (top right), $n$ = 1000 (bottom left) and $n$ = 2500 (bottom right). In each graphic, the $x$-axis shows number of symptoms $p \in \{6; 7; 8; 9; 10\}$.

methods considered split into two groups: those that may estimate by zero a partial correlation (`poly.wls`, the `Glasso, BGGM` and `ggmMS` families) and those whose estimator will always be non-zero (i.e., `poly.mle, pears, ggmSS, PAsso` and the `GGMnr` family). For the second group we observe, as expected, a bad zero specificity and a perfect sensitivity of one, see Figures 5 and 6. This is in fact valid independently of the scenario considered (as soon as these quantities are defined for the scenario). Indeed, these methods are not designed to get a compromise between sensitivity and specificity. In the first group of methods, expected to produce such a compromise between sensitivity and specificity, we observe that `poly.wls` and `BGGM.explore` are obtaining the best specificities (close to 1) while `Glasso.poly` obtains the best sensitivities. We also note that the performances of the `Glasso` family methods do not improve with the number of symptoms (top right part of Figures 5 and 6) contrarily to what could be expected, as these methods are dedicated to high-dimensional settings; while they tend to degrade with the density of the true underlying graph (bottom left part of Figures 5 and 6). While the `BGGM` family of methods shows good specificities, we recall that these methods exhibit very large execution error rates (Figure 1).

Finishing with precision – the fraction of true positives among selected edges – the results do not cluster into two groups of methods anymore and `BGGM.explore` and `poly.wls` obtain the best performances, see Figure 7. Considering that `poly.wls` is a much simpler approach with less execution errors, it represents the best method from the precision point of view.

*Scenario 2 - Ideal case with complete graph.* This scenario overlaps with Scenario 1, and relies on a correlation structure preliminary estimated on a real dataset (see Appendix B for details concerning the dataset used). Therefore, it resembles Scenario 1 with the density parameter `prob` equal to 1, except that the underlying correlation structure is expected to be more realistic here. It is important to note that there are no true negatives in this setup. Overall, the datasets simulated from this second scenario exhibit a wide range of skewness values across the different settings, mimicking in this way real datasets (data not shown). We note that only three methods, `GGMnr.boot.poly, BGGM.estimate` and `BGGM.explore` exhibit some non null execution error rates in a few different settings. Those error rates are globally lower than in Scenario 1 and always staying below 40% (see Figure B in Appendix D for an illustration).

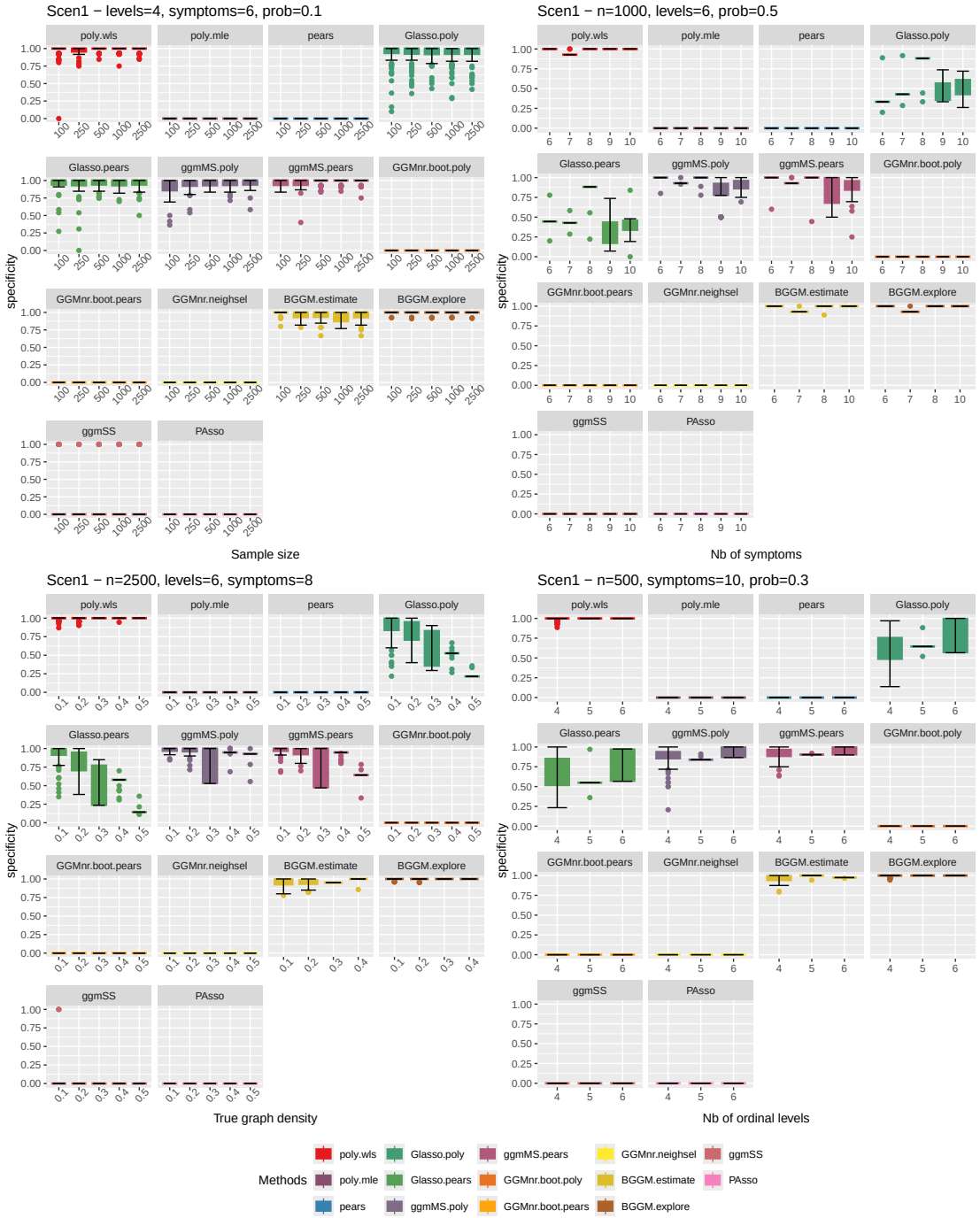


FIGURE 5.

Scenario 1: Examples of the behaviour of the specificity (boxplots over 100 replicates) of all the methods (displayed as 14 graphics in each of the 4 pictures) with respect to evolving parameters (displayed on the $x$-axis): sample sizes $n \in \{100; 250; 500; 1,000; 2500\}$ on top left; number of variables $p \in \{6; 7; 8; 9; 10\}$ on top right; density of the true underlying graph `prob`$\in \{0.1; 0.2; 0.3; 0.4; 0.5; 1\}$ on bottom left and number of ordinal levels $L \in \{4, 5, 6\}$ on bottom right.

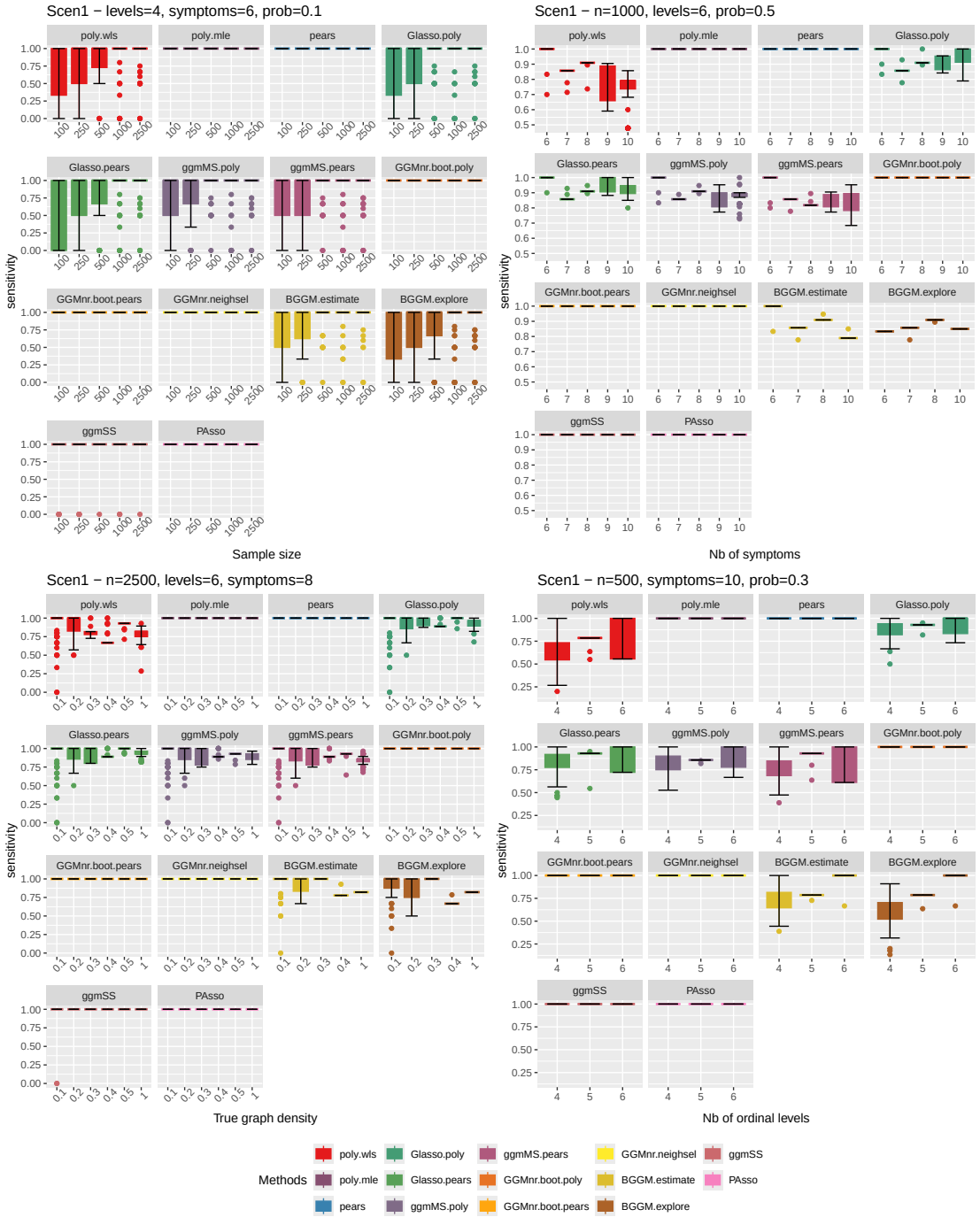


FIGURE 6.

Scenario 1: Examples of the behaviour of the sensitivity (boxplots over 100 replicates) of all the methods (displayed as 14 graphics in each of the 4 pictures) with respect to evolving parameters (displayed on the $x$-axis): sample sizes $n \in \{100; 250; 500; 1,000; 2500\}$ on top left; number of variables $p \in \{6; 7; 8; 9; 10\}$ on top right; density of the true underlying graph `prob`$\in \{0.1; 0.2; 0.3; 0.4; 0.5; 1\}$ on bottom left and number of ordinal levels $L \in \{4, 5, 6\}$ on bottom right.

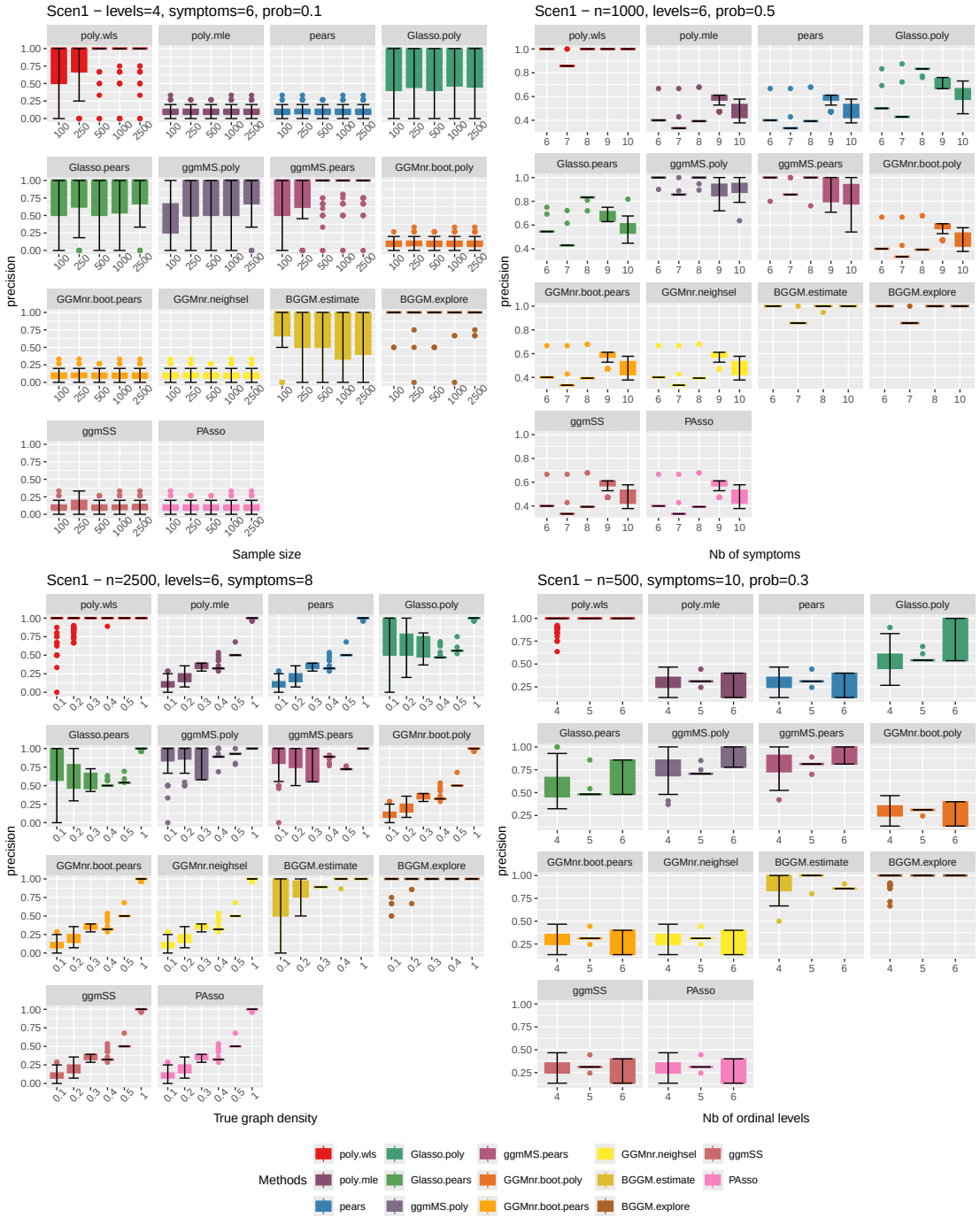


FIGURE 7.

Scenario 1: Examples of the behaviour of the precision (boxplots over 100 replicates) of all the methods (displayed as 14 graphics in each of the 4 pictures) with respect to evolving parameters (displayed on the $x$-axis): sample sizes $n \in \{100; 250; 500; 1,000; 2500\}$ on top left; number of variables $p \in \{6; 7; 8; 9; 10\}$ on top right; density of the true underlying graph `prob`$\in \{0.1; 0.2; 0.3; 0.4; 0.5; 1\}$ on bottom left and number of ordinal levels $L \in \{4, 5, 6\}$ on bottom right.

Comparing accuracy in terms of MSE, the results are quite different from the Scenario 1, see Figure 8. In line with what is observed in Scenario 1, the mean value and the variance of the MSE decrease with the sample size and increase with the number of symptoms. However, here `poly.mle` exhibits a lower MSE than `poly.wls` and `pears` is competitive with `poly.mle`. The methods `Glasso.poly` and `Glasso.pearson` are not competitive with `poly.mle` or with `pears` and the `BGGM` family of methods. In this scenario 2, the `Glasso` family is overall the worse group of methods. The `GGMnr` family of methods exhibits pretty good results, the 3 variants showing similar performances. While their performances are competitive with respect to `poly.mle` and `pears`, we recall that these methods are less direct and simple than the latter. The remaining methods, namely the `ggmMS` family, `ggmSS` and `PAsso`, while performing better than the `Glasso` approaches, are not competitive to the best ones (`poly.mle` and `pears`).

Sensitivity confirms the results observed for Scenario 1, see Figure 9. We recall that neither specificity nor precision are defined in this scenario.

*Scenario 3 - Control case.* This scenario relies on independent variables, corresponding to an empty graph to infer. The datasets exhibit a wide range of skewness values, this way mimicking real datasets (data not shown). In this Scenario 3, `GGMnr.boot.poly` and the `BGGM` family show some execution errors, that can be above 50% when $n$ = 100 and $p$ = 20 and decrease with increasing sample size. The method `poly.wls` exhibits execution errors in only 2 settings: when the number of symptoms $p$ = 20 is very large and the sample size is small, i.e. either $n$ = 100 (up to 40% execution error rate) or $n$ = 250 (less than 5% execution error rate, see Figure C in Appendix D).

From the MSE perspective, the `Glasso` methods are the best in this no signal case, with a quite small advantage for `Glasso.pears`, see Figure 10 for an illustration.

Specificity in this scenario gives pretty much the same conclusions as for Scenario 1 (except for the method `ggmSS`, whose specificity takes exactly two values 0 and 1 over the 100 replicates). More precisely, the `Glasso` methods and `poly.wls` obtain the best results, with the former ones exhibiting a specificity very close to 1 when the number of symptoms $p$ = 20, while the latter is better when $p = 6, 7$, see Figure 11.

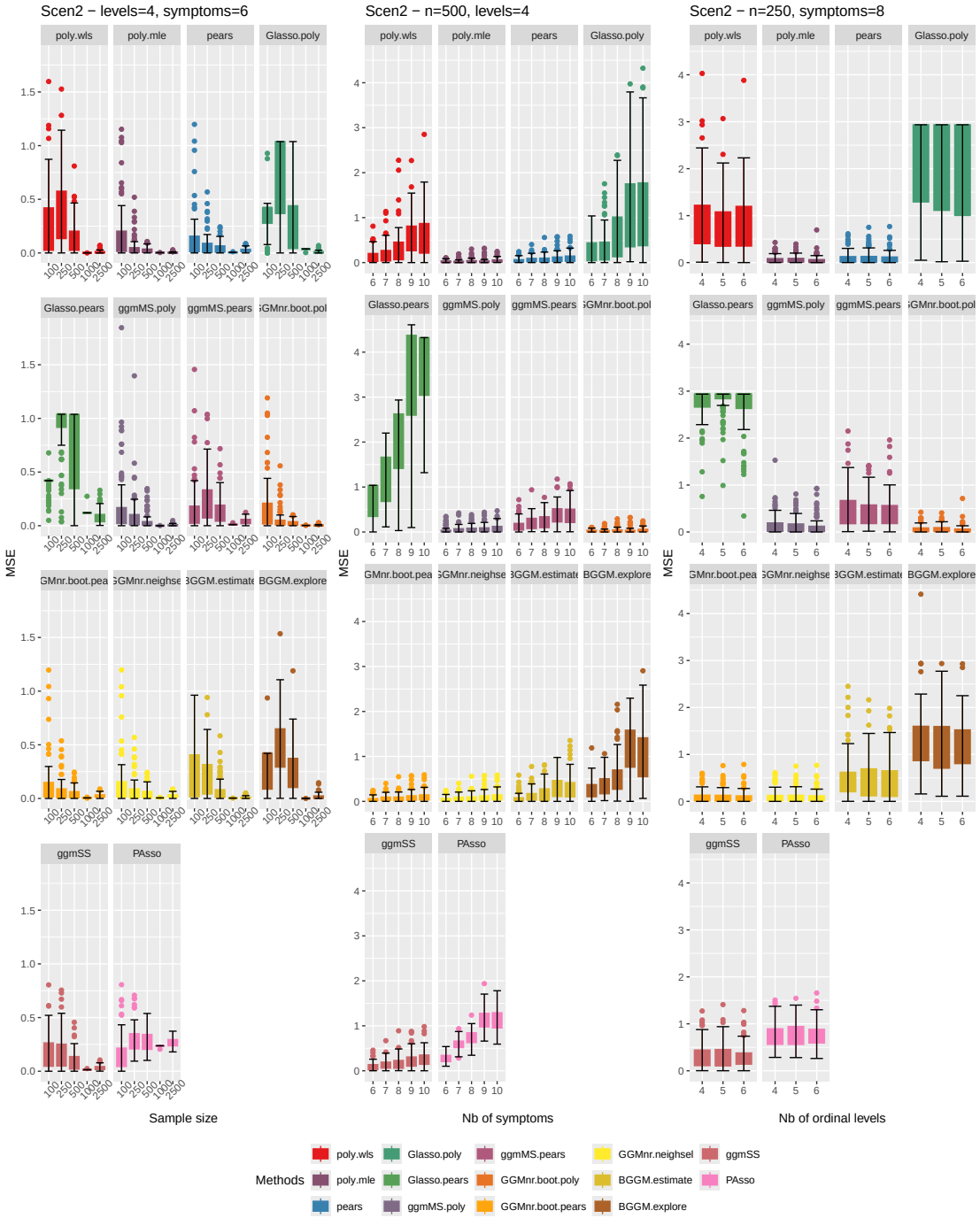


FIGURE 8.

Scenario 2: Examples of the behaviour of the MSE (boxplots over 100 replicates) of all the methods (displayed as 14 graphics in each of the 3 pictures) with respect to evolving parameters (displayed on the $x$-axis): sample sizes $n \in \{100; 250; 500; 1,000; 2500\}$ on the left; number of variables $p \in \{6; 7; 8; 9; 10\}$ in the center and number of ordinal levels $L \in \{4, 5, 6\}$ on the right.

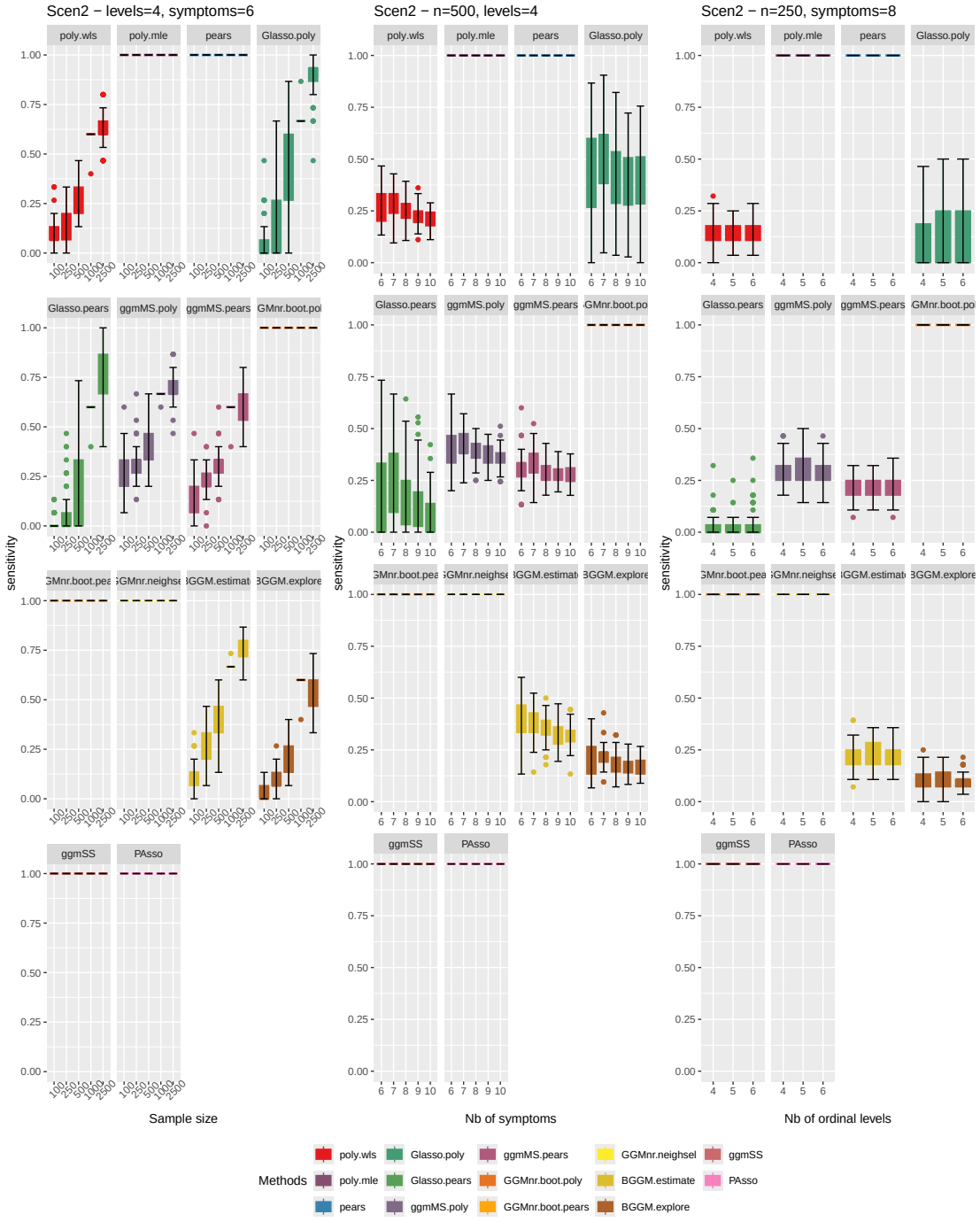


FIGURE 9.

Scenario 2: Examples of the behaviour of the sensitivity (boxplots over 100 replicates) of all the methods (displayed as 14 graphics in each of the 3 pictures) with respect to evolving parameters (displayed on the $x$-axis): sample sizes $n \in \{100; 250; 500; 1,000; 2500\}$ on the left; number of variables $p \in \{6; 7; 8; 9; 10\}$ in the center and number of ordinal levels $L \in \{4, 5, 6\}$ on the right.

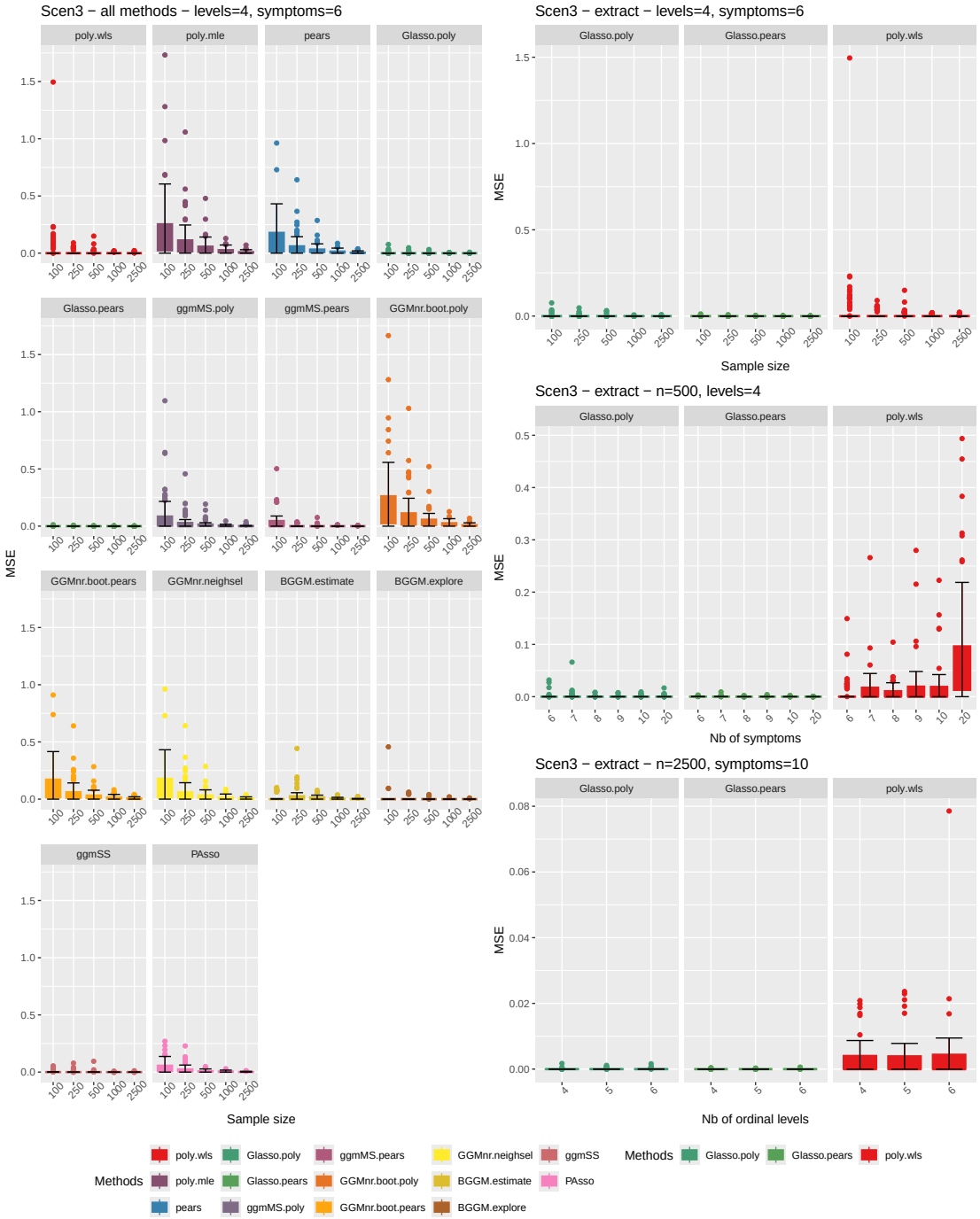


FIGURE 10.

Scenario 3: Examples of the behaviour of the MSE (boxplots over 100 replicates). On the left, all the methods are displayed as 14 graphics and the $x$-axis shows sample sizes $n \in \{100; 250; 500; 1,000; 2500\}$, in the case of $p = 6$ symptoms and $L = 4$ levels. On the right, a zoom on the 3 methods `Glasso.pears,Glasso.poly,poly.wls`; wrt sample size and in the same setting as on the left (top), wrt the number of symptoms on the $x$-axis (center) and wrt the number of ordinal levels (bottom).

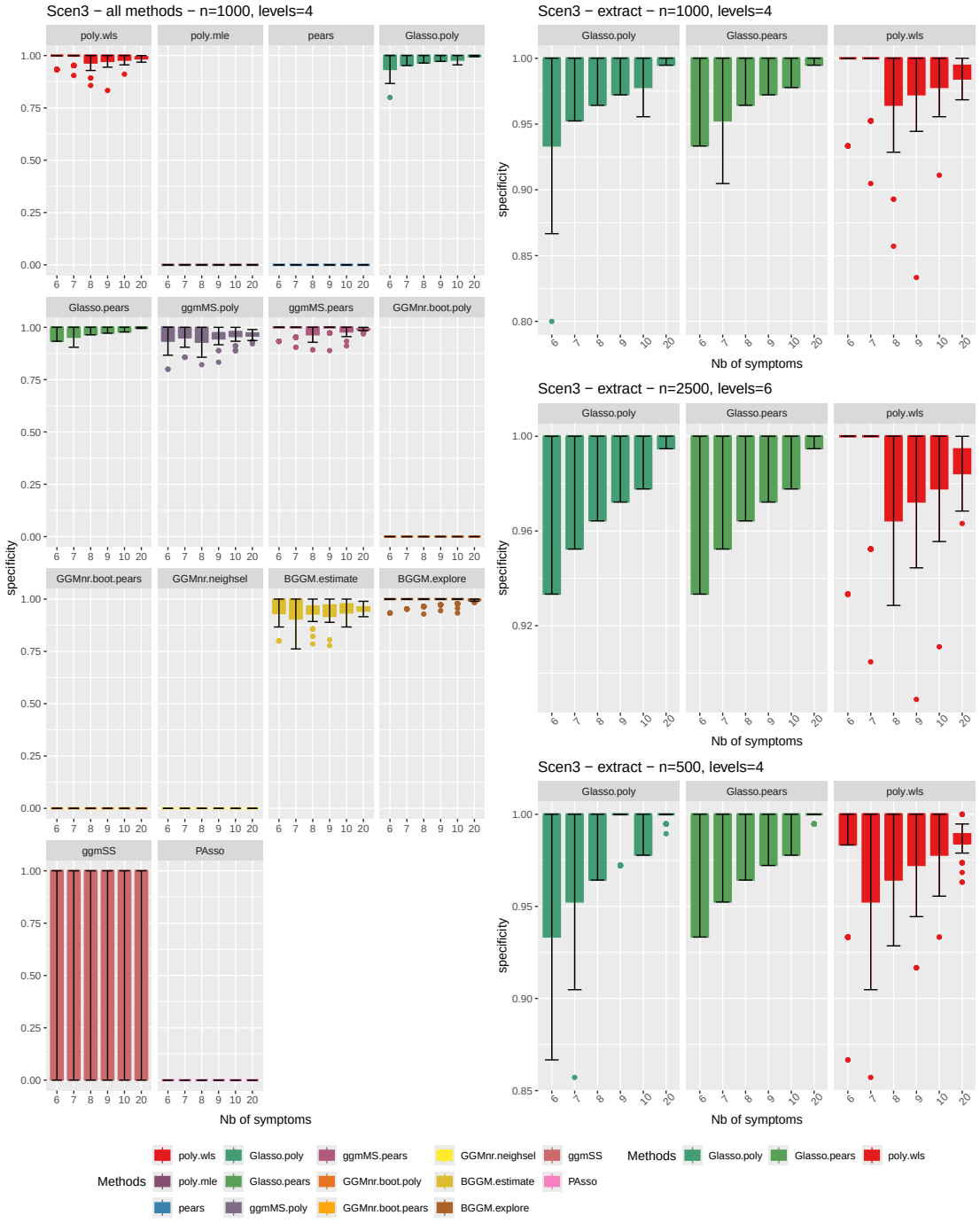


FIGURE 11.

Scenario 3: Examples of the behaviour of the specificity (boxplots over 100 replicates) wrt the number of symptoms on the $x$-axis. On the left, all the methods are displayed as 14 graphics in the case of sample size $n = 1000$ and $L = 4$ levels. On the right, a zoom on the 3 methods `Glasso.pears,Glasso.poly,poly.wls`; in the same setting as on the left (top), for $n = 2500$ and $L = 6$ (center) and for $n = 500$ and $L = 4$ (bottom).

*Scenario 4 - Most realistic case.* This scenario is the most realistic one, as it integrates some heterogeneity in the sample, by simulating two sub-populations of individuals with different behavior. Again the datasets exhibit very diverse skewness values, with absolute values that can be much higher than in the previous scenario (data not shown). This is expected as the datasets include heterogeneity. In this scenario, the `BGGM` methods never worked, with an execution error rate equal to 1 in all settings. In particular, we obtained no MSE or sensitivity for these methods. Also, `GGMnr.boot.poly` has a high execution error rate in most settings. For small sample sizes $n$ and increasing values of $p$, the methods `poly.wls, Glasso.poly, ggmMS.poly` and `PAsso` may exhibit large execution error rates. The problem fades with increasing sample size and curiously, it also seems to diminish with increasing number of levels, see Figure 12.

Concerning MSE, the comparison of the methods excludes `BGGM` methods that produced no results and `GGMnr.boot.poly` that has too few results for being relevant. Also, we decided to remove 1% of all the largest MSE values to remove some outliers and obtain clearer results. We observe that the results are highly variable, with the direct methods `poly.wls, poly.mle, pears`, the `Glasso` family and `ggmMS.poly` alternatively being the best approaches, see Figure 13.

Considering sensitivity, among the methods that aim at a compromise with specificity (which is not defined here, neither precision is), `Glasso.poly` shows the best results, see Figure 14.

This most realistic scenario shows that for real datasets that are expected to be more complex than ideal case situations, no method is uniformly best for estimating partial correlation structure, though `Glasso.poly` gives the best sensitivities (among the methods trying to find a compromise with specificity).

## Discussion

Overall these results highlight the importance of direct and simple methods for estimating partial polychoric and Pearson correlations, like `poly.wls` and `pears`. Our findings show these methods are well-suited to most psychometric datasets and should be systematically explored. Our study advocates for using different methods, as the complexity of real datasets is obviously not captured by the models underlying the different approaches, leading to none of the methods dominating the others for complex scenarios (e.g., Scenario 4). In particular, we suggest that

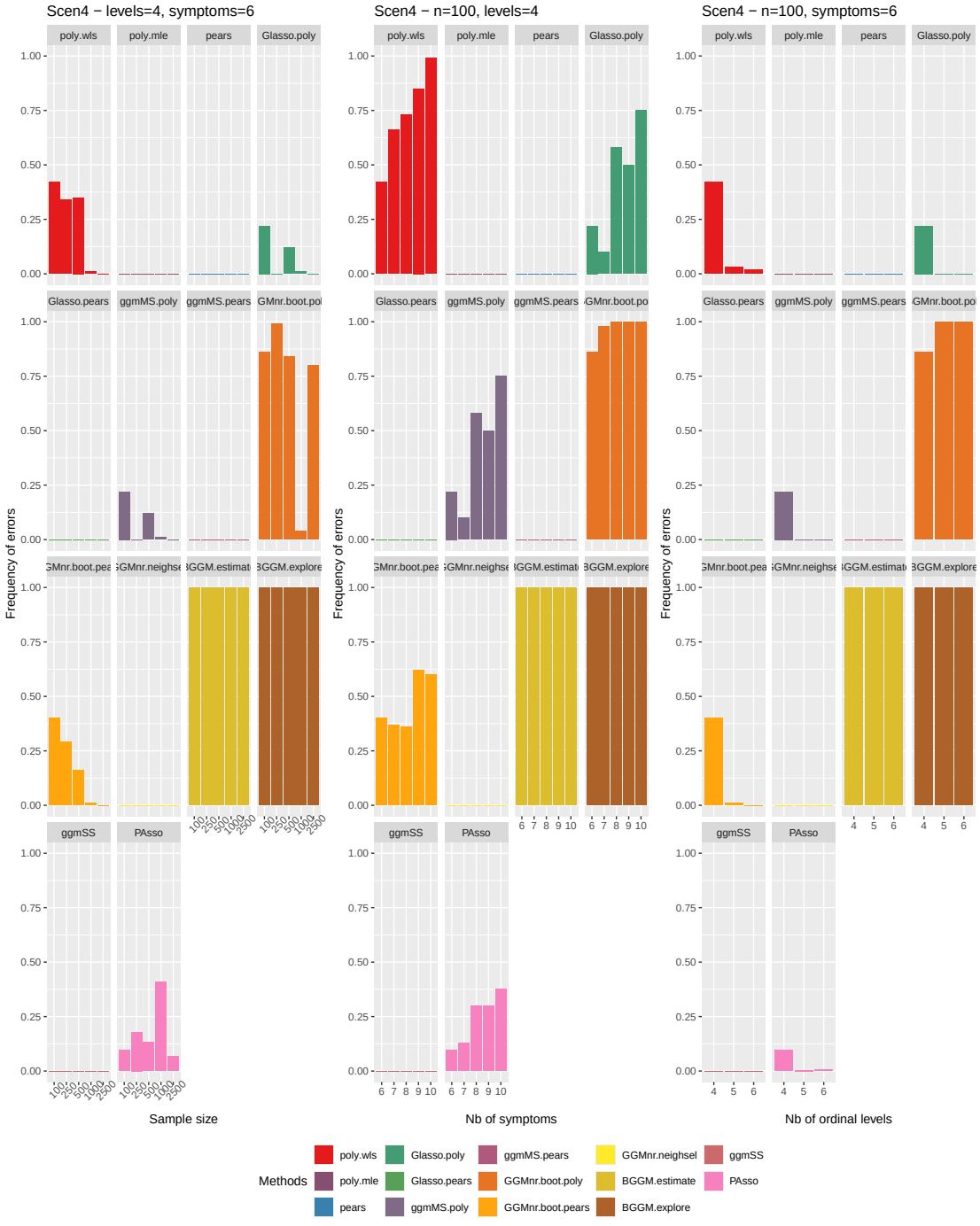


FIGURE 12.

Scenario 4: Examples of the execution error rates (averaged over 100 replicates) of all the methods (displayed as 14 graphics in each picture). The $x$-axis displays sample sizes (left picture), number of symptoms (center picture) and number of levels (right picture).

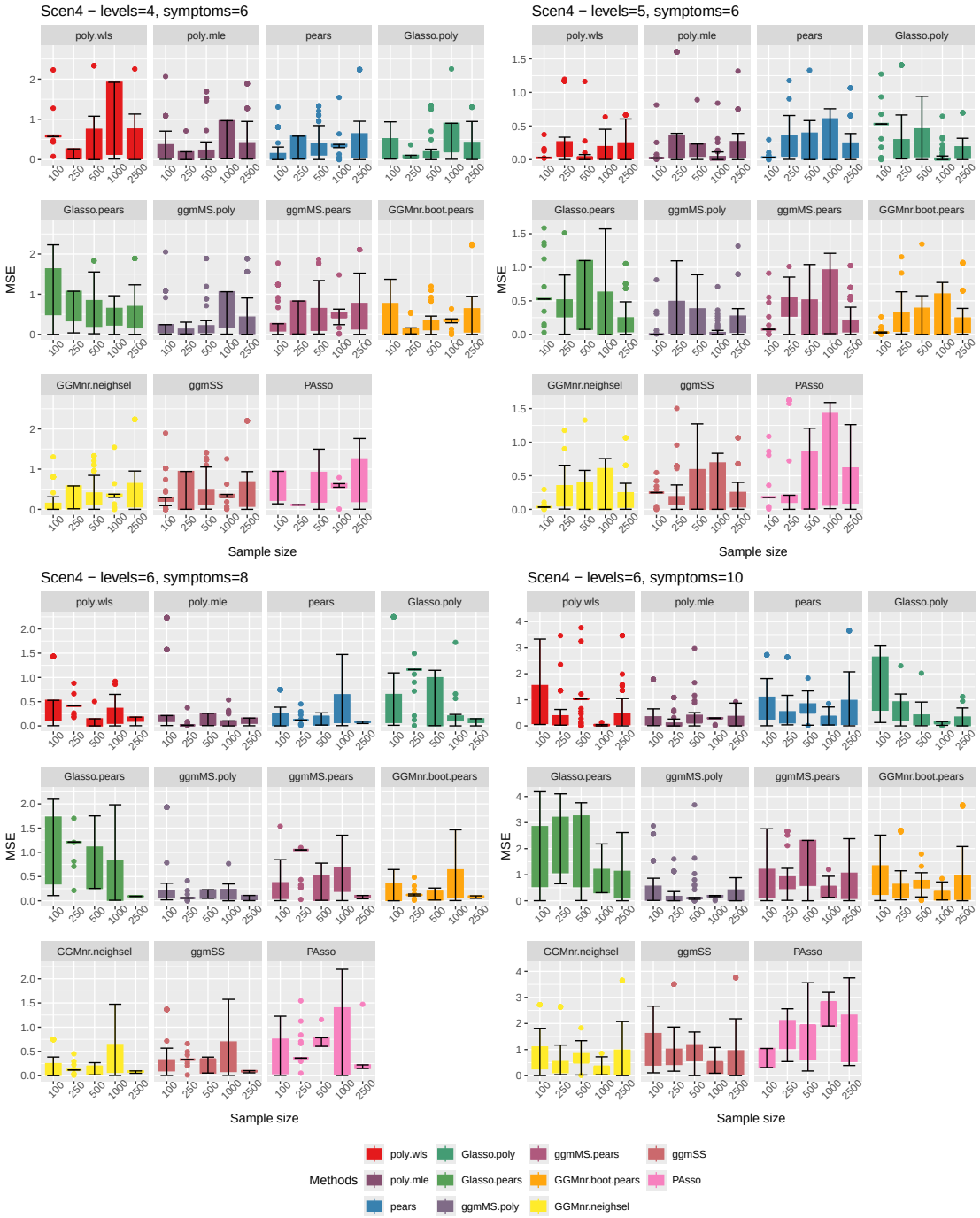


FIGURE 13.

Scenario 4: Examples of the behaviour of the MSE (boxplots over 100 replicates) of the working methods (displayed as 11 graphics in each picture). The $x$-axis displays sample sizes. The 4 pictures display different settings wrt number of levels $L$ and number of symptoms $p$. None of the methods appears to be uniformly best.

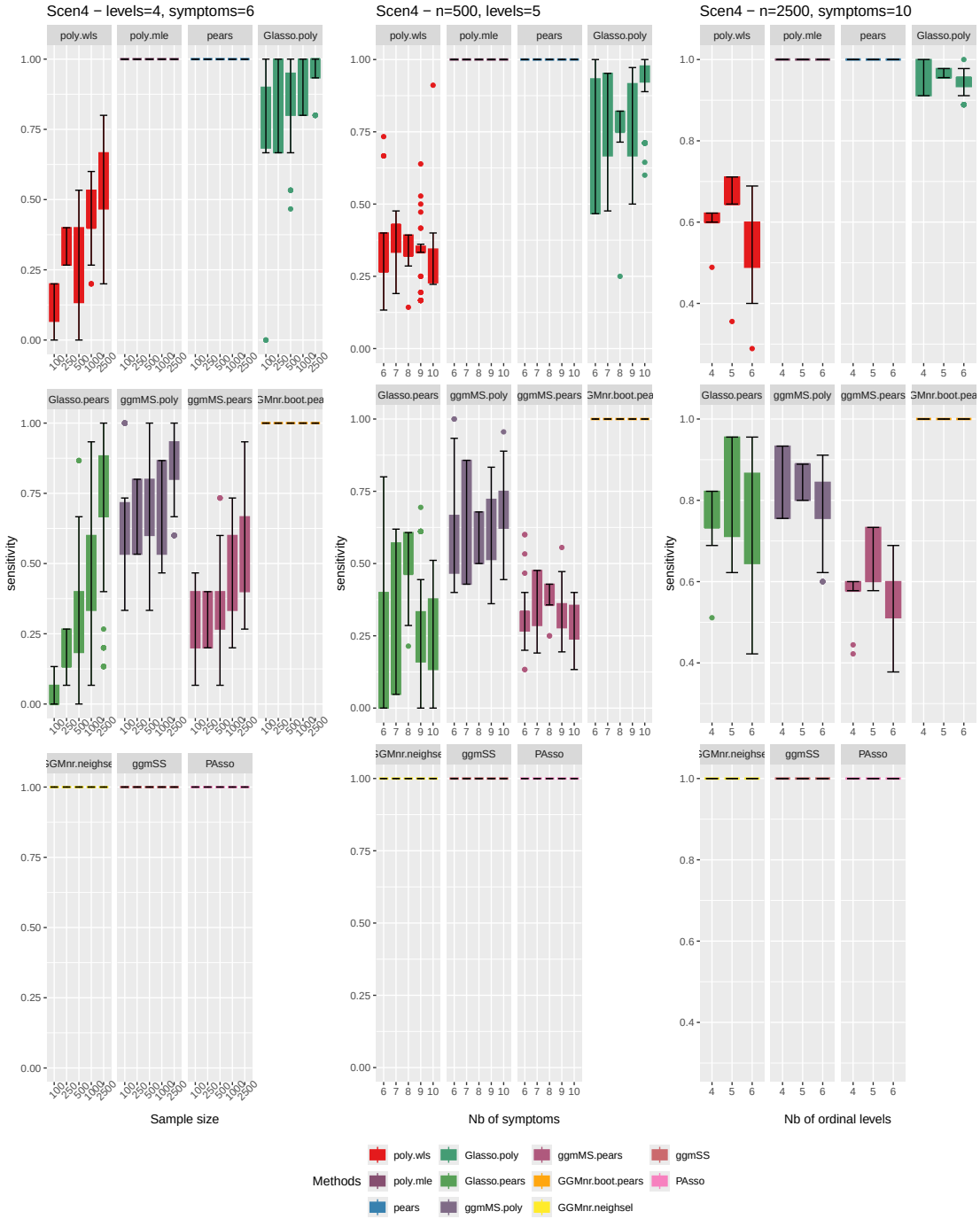


FIGURE 14.

Scenario 4: Examples of the behaviour of the sensitivity (boxplots over 100 replicates) of the working methods (displayed as 11 graphics in each picture). The $x$-axis displays sample sizes (left picture), number of symptoms (center picture) and number of levels (right picture).

researchers systematically explore the data using at least one direct method (`poly.wls, pears`) and one sparse approach (`Glasso.poly, Glasso.pears, ggmMS.poly`). When possible, we also recommend to try an unregularized approach (`GGMnr` family).

Most of the comparative studies published focus on introducing new methods that address the limitations of older ones (see for e.g. Williams et al., 2020; Williams and Rast, 2020). To the best of our knowledge, only one large-scale comparative study was recently published in Isvoranu and Epskamp (2023) but did not include some of the simplest and direct approaches (i.e.,`poly.mle, pears`). We expanded this latter work, focusing on ordinal variables, concentrating on low-dimensional settings and exploring a sophisticate heterogeneous scenario that mimics real datasets (Scenario 4).

Our study is an additional step in the quest for a better understanding of the behaviour of psychometric networks inference methods although it could be extended in several directions. We explored a quite restricted set of values for the number of levels ($L = \{4, 5, 6\}$) which may be too limited to observe the full impact of this parameter. Also, a natural extension is considering a heterogeneous number of levels per ordinal variable (i.e., not the same for all variables), though this drastically increases the size of the experimental design and complicates the analysis. Most of the methods tested are too slow to enable simulations that could go up to $p = 20$ variables, a realistic situation though, see for e.g. Fried et al. (2019); Zhou et al. (2022).

## References

Borsboom, D. (2017). A network theory of mental disorders. *World Psychiatry 16*(1), 5–13.

Borsboom, D., M. Deserno, M. Rhemtulla, S. Epskamp, and et al. (2021). Network analysis of multivariate data in psychological science. *Nat Rev Methods Primers 1*, 58.

Delli Colli, C., F. Chiarotti, P. Campolongo, A. Giuliani, and I. Branchi (2024). Towards a network-based operationalization of plasticity for predicting the transition from depression to mental health. *Nat. Mental Health 2*, 200–208.

Epskamp, S. (2020). Psychometric network models from time-series and panel data. *Psychometrika 85*, 206–231.

Epskamp, S. (2024). *psychonetrics: Structural Equation Modeling and Confirmatory Network Analysis*. R package version 2.4.

Epskamp, S., A. O. J. Cramer, L. J. Waldorp, V. D. Schmittmann, and D. Borsboom (2012). qgraph: Network visualizations of relationships in psychometric data. *Journal of Statistical Software 48*(4), 1–18.

Epskamp, S. and E. I. Fried (2018). A tutorial on regularized partial correlation networks. *Psychological Methods 23*(4), 617–634.

Epskamp, S., L. J. Waldorp, R. Mõttus, and D. Borsboom (2018). The Gaussian graphical model in cross-sectional and time-series data. *Multivariate Behavioral Research 53*(4), 453–480.

Foygel, R. and M. Drton (2010). Extended Bayesian information criteria for gaussian graphical models. In J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta (Eds.), *Advances in Neural Information Processing Systems*, Volume 23. Curran Associates, Inc.

Fried, E., S. von Stockert, J. Haslbeck, F. Lamers, R. Schoevers, and B. Penninx (2019). Using network analysis to examine links between individual depressive symptoms, inflammatory markers, and covariates. *Psychological Medicine 50(16)*, 2682–2690.

Hakulinen, C., E. Fried, L. Pulkki-Råback, M. Virtanen, J. Suvisaari, and M. Elovainio (2020). Network structure of depression symptomology in participants with and without depressive disorder: the population-based Health 2000–2011 study. *Soc Psychiatry Psychiatr Epidemiol 55*, 1273–1282.

Isvoranu, A., S. Epskamp, L. Waldorp, and D. Borsboom (Eds.) (2022). *Network Psychometrics with R: A Guide for Behavioral and Social Scientists*. Routledge.

Isvoranu, A.-M. and S. Epskamp (2023). Which estimation method to choose in network psychometrics? Deriving guidelines for applied researchers. *Psychological Methods 28*(4), 925–946.

Koller, D. and N. Friedman (2009). *Probabilistic graphical models*. Adapt. Comput. Mach. Learn. Cambridge, MA: MIT Press.

Lee, K., Q. Chen, and W. DeSarbo (2022). Estimating finite mixtures of ordinal graphical models. *Psychometrika 87*, 83–106.

Liddell, T. M. and J. K. Kruschke (2018). Analyzing ordinal data with metric models: What could possibly go wrong? *Journal of Experimental Social Psychology 79*, 328–348.

Likert, R. (1932). A technique for the measurement of attitudes. *Archives of Psychology 22*(140), 5–55.

Liu, D., S. Li, Y. Yu, and I. Moustaki (2021). Assessing partial association between ordinal variables: Quantification, visualization, and hypothesis testing. *Journal of the American Statistical Association 116*(534), 955–968.

McNally, R. J., D. J. Robinaugh, G. W. Y. Wu, L. Wang, M. K. Deserno, and D. Borsboom (2015). Mental disorders as causal systems: A network approach to posttraumatic stress disorder. *Clinical Psychological Science 3*(6), 836–849.

Meinshausen, N. and P. Bühlmann (2006). High-dimensional graphs and variable selection with the lasso. *Ann. Statist. 34*(3), 1436–1462.

Meinshausen, N. and P. Bühlmann (2010). Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 72*(4), 417–473.

Mohammadi, R., E. C. Wit, and A. Dobra (2024). *BDgraph: Bayesian Structure Learning in Graphical Models using Birth-Death MCMC*. R package version 2.72.

Muthén, B. (1984). A general structural equation model with dichotomous, ordered categorical, and continuous latent variable indicators. *Psychometrika 49*(1), 115–132.

Norris, A. E. and K. J. Aroian (2004). To Transform or Not Transform Skewed Data for Psychometric Analysis: That Is the Question! *Nursing Research 53*(1), 67–71.

Olsson, U. (1979). Maximum likelihood estimation of the polychoric correlation coefficient. *Psychometrika 44*, 443–460.

Rhemtulla, M., E. I. Fried, S. H. Aggen, F. Tuerlinckx, K. S. Kendler, and D. Borsboom (2016). Network analysis of substance abuse and dependence symptoms. *Drug and Alcohol Dependence 161*, 230–237.

Robinaugh, D. J., R. H. A. Hoekstra, E. R. Toner, and D. Borsboom (2019). The network approach to psychopathology: a review of the literature 2008–2018 and an agenda for future research. *Psychological Medicine 50*(3), 353–366.

Schäfer, J., R. Opgen-Rhein, and K. Strimmer (2021). *GeneNet: Modeling and Inferring Gene Networks*. R package version 1.2.16.

Schäfer, J. and K. Strimmer (2004). An empirical Bayes approach to inferring large-scale gene association networks. *Bioinformatics 21*(6), 754–764.

Sinoquet, C. and R. Mourad (2014). *Probabilistic graphical models for genetics, genomics and postgenomics*. Oxford: Oxford University Press.

Spiller, T. R., M. Schick, U. Schnyder, R. A. Bryant, A. Nickerson, and N. Morina (2017). Symptoms of posttraumatic stress disorder in a clinical sample of refugees: A network analysis. *European Journal of Psychotraumatology 8*(Suppl 3), 1318032.

Trivedi, M. H., A. J. Rush, S. R. Wisniewski, A. A. Nierenberg, D. Warden, L. Ritz, G. Norquist, R. H. Howland, B. Lebowitz, P. J. McGrath, K. Shores-Wilson, M. M. Biggs, G. K. Balasubramani, M. Fava, and STAR*D Study Team (2006). Evaluation of outcomes with citalopram for depression using measurement-based care in STAR*D: Implications for clinical practice. *American Journal of Psychiatry 163*(1), 28–40.

Williams, D. and P. Rast (2020). Back to the basics: Rethinking partial correlation network methodology. *Br J Math Stat Psychol. 73*(2), 187–212.

Williams, D. R. (2021a). Bayesian estimation for Gaussian graphical models: Structure learning, predictability, and network comparisons. *Multivariate Behavioral Research 56*(2), 336–352.

Williams, D. R. (2021b). *GGMnonreg: Non-Regularized Gaussian Graphical Models*. R package version 1.0.0.

Williams, D. R. and J. Mulder (2019). *BGGM: Bayesian Gaussian Graphical Models in R*. R package version 2.1.1.

Williams, D. R. and J. Mulder (2020). Bayesian hypothesis testing for Gaussian graphical models: Conditional independence and order constraints. *Journal of Mathematical Psychology 99*, 102441.

Williams, D. R., R. Philipe, P. R. Luis, and J. Mulder (2020). Comparing Gaussian graphical models with the posterior predictive distribution and Bayesian model selection. *Psychological Methods 25*(5), 653–672.

Williams, D. R., M. Rhemtulla, A. C. Wysocki, and P. Rast (2019). On nonregularized estimation of psychological networks. *Multivariate behavioral research 54*(5), 719–750.

Zhou, J., S. Liu, T. Mayes, Y. Feng, M. Fang, L. Xiao, and W. G (2022). The network analysis of depressive symptoms before and after two weeks of antidepressant treatment. *J Affect Disord 15*, 126–134.

Zhu, X., S. Li, D. Liu, and Y. Chen (2021). *PAsso: Assessing the Partial Association Between Ordinal Variables*. R package.

## Appendix A: Statistical correlations

We provide here statistical definitions around the notion of correlation.

First, covariance between two real-valued variables $X, Y$ is defined as $Cov(X,Y) = \mathbb{E}[(X - \mathbb{E}(X))(Y - \mathbb{E}(Y)] = \mathbb{E}(XY) - \mathbb{E}(X)\mathbb{E}(Y)$. Pearson's correlation is a normalized version of the covariance, defined as $\rho_{X,Y} = Cov(X,Y)/\sigma_X\sigma_Y$, where $\sigma_X$ is the standard deviation of $X$. It is a measure of linear correlation between the 2 variables. Partial correlation between 2 variables $X, Y$ adjusting for the vector of covariates $\mathbf{W}$ (denoted $\rho_{X,Y.\mathbf{W}}$) is defined as the correlation between the residuals obtained from the linear regression of $X$ (resp. $Y$) over $\mathbf{W}$. Now for a whole vector of variables $(X_1, \ldots, X_p)$, in order to compute at once all the partial correlations between any pair $X_i, X_j$ adjusting for all other variables $\{X_k\}_{k\neq i,j}$, we proceed as follows. Partial correlation (resp. empirical partial correlations) coefficients between a set of variables $(X_1, \ldots, X_p)$ may be obtained from the (resp. empirical) precision matrix $\Omega = (\omega_{ij})$, i.e. the inverse of the (empirical) covariance matrix $\Sigma = \Omega^{-1}$ in the following way

$$\rho_{X_i,X_j.(X_k)_{k\neq i,j}} = \frac{-\omega_{ij}}{\sqrt{\omega_{ii}}\sqrt{\omega_{jj}}}.$$

In the Gaussian setting only, the partial correlation coefficient translates into conditional independence. More precisely, if $(X, Y, \mathbf{W})$ is jointly Gaussian, then $\rho_{X,Y.\mathbf{W}} = 0 \iff X \perp\!\!\!\perp Y|\mathbf{W}$ (i.e. $X, Y$ are independent conditional on $\mathbf{W}$).

In the case of discrete variables, regression models rely on a *link function* $G^{-1}$, which is the inverse of a continuous cumulative distribution function (cdf) $G$, chosen by the statistician. For e.g., the regression of $X$ over the vector of covariates $\mathbf{W}$ is done through the model

$$G^{-1}(\mathbb{P}(X \leq j)) = \alpha_j + f(\mathbf{W}, \boldsymbol{\beta}), \tag{A1}$$

where $\boldsymbol{\beta}$ is a vector of parameters and $f(\mathbf{W}, \boldsymbol{\beta}) = \mathbf{W}^\intercal\boldsymbol{\beta}$ in the linear regression setting. Here, Eq. (A1) means that there exists some random variable $\epsilon$ distributed according to the cdf $G$ and independent of $\mathbf{W}$ such that setting $Z = f(\mathbf{W}, \boldsymbol{\beta}) + \epsilon$, we have that $X$ is obtained from $Z$ through a binning operation, relying on the cut points $-\infty = \alpha_0 < \alpha_1 < \cdots < \alpha_L = +\infty$. In other words, we set $X = j$ whenever $Z$ falls in the bin $(\alpha_{j-1}, \alpha_j]$.

Polychoric correlations are defined with the above model using linear regression, when $G$ is the cdf of the standardized Gaussian distribution. More precisely, if $(X_1, \ldots, X_p)$ are ordinal variables and we want to define partial polychoric correlations, here is how we proceed. We first start with only 2 variables. Say we want the partial polychoric correlation between $X_1, X_2$ adjusting for $\mathbf{X}_{k\geq 3} := \{X_k\}_{k\geq 3}$ then we assume that there exist some latent (i.e. not observed) continuous variables $(Z_1, Z_2)$ whose joint distribution is a two-dimensional Gaussian with mean $(\mathbf{X}_{k\geq 3}^\intercal \boldsymbol{\beta}_1, \mathbf{X}_{k\geq 3}^\intercal \boldsymbol{\beta}_2)^\intercal$ and covariance matrix $\Sigma_{1,2.k\geq 3}$. Let $\Omega_{1,2.k\geq 3} = (\Sigma_{1,2.k\geq 3})^{-1} = (\omega_{ij})$ denote the corresponding precision matrix. The link between $(X_1, X_2)$ and $(Z_1, Z_2)$ is that we assume that there exist $L_i - 1$ cut points on the real line ($L_i$ is the number of levels of the ordinal variable $X_i$, for $i = 1, 2$) such that $X_i$ is obtained from $Z_i$ by binning the variable into $L_i$ bins, through the $L_i - 1$ cut points. Then the polychoric partial correlation between $X_1, X_2$ adjusting for $\mathbf{X}_{k\geq 3} := \{X_k\}_{k\geq 3}$ is given as $-\omega_{12}/\sqrt{\omega_{11}\omega_{22}}$. Now for a set of ordinal random variables $(X_1, \ldots, X_p)$, polychoric correlations assume that there exist some latent (i.e. not observed) continuous variables $(Z_1, \ldots, Z_p)$ whose joint distribution is multivariate Gaussian with some mean vector (whose $k$-th coordinate is just the linear regression of $Z_k$ over all other $Z_i, i \neq k$) and covariance matrix $\Sigma = (\sigma_{ij})_{i,j}$. Denote $\Omega = \Sigma^{-1} = (\omega_{ij})$ the corresponding precision matrix. Then $X_k$ is a binned version of $Z_k$ obtained thanks to some cut points $-\infty = \alpha_{k,0} < \alpha_{k,1} < \cdots < \alpha_{k,L_k} = +\infty$. Finally, the (non partial) polychoric correlations between $X_i$ and $X_j$ is defined as $\rho_{i,j} = \sigma_{ij}/\sigma_{ii}\sigma_{jj}$. It is the correlation between the corresponding latent variables $Z_i, Z_j$. Note that this quantity is not adjusting for all other $X_k, k \neq i, j$. Finally, we define partial polychoric correlation between $X_i, X_j$ as the quantity $-\omega_{ij}/\sqrt{\omega_{ii}\omega_{jj}}$ where $\Omega = \Sigma^{-1}$.

## Appendix B: Simulation scenarios

It is not possible to simulate directly categorical variables with a prescribed correlation structure. So we rely on the simulation of continuous variables which are then randomly binned into categorical (ordinal) values. We have a total of 4 scenarios.

*First simulation setting: purely synthetic, polychoric correlation structure*

In this setting, we draw a random partial correlation matrix $\Theta$ from a $G$-Wishart distribution, using the `R` package `BDgraph` (Mohammadi et al., 2024). Indeed, correlation and partial correlation matrices have the constraint that they are definite positive matrices and the $G$-Wishart distribution is the only known distribution that outputs definite positive matrices. More precisely, we first draw a skeleton (i.e. positions of non-zero values), with some probability parameter `'prob'`. It is the density of the true graph and we explore different values for this parameter, with low values corresponding to sparse graphs while larger values correspond to denser graphs. Then, for each non-zero interaction, we draw a weight so that the joint distribution of the resulting matrix is $G$-Wishart. Finally, from this partial correlation matrix $\Theta$, we draw Gaussian random variables having this correlation structure and then get (with random bins) categorical (ordinal) variables.

This setting is exactly the one corresponding to the idea behind using polychoric correlations as input in graph inference methods. All those steps are done automatically at once when relying on the function `bdgraph.sim`. We vary the sparsity parameter `'prob'` of the graph in the set $\{0.1; 0.2; 0.3; 0.4; 0.5; 1\}$.

*Second simulation setting: relying on an initial real dataset, polychoric correlation structure*

Here, we start from the Sequenced Treatment Alternatives to Relieve Depression (STAR*D) dataset (Trivedi et al., 2006). It is a randomized clinical trial of outpatients with major depressive disorder (MDD), designed to prospectively evaluate the effectiveness of pharmacological and psychotherapeutic treatment. In the dataset, MDD symptom severity is assessed using the 16-items Quick Inventory of Depressive Symptomatology – Clinician-Rating scale (QIDS-16). The 16 items have been collapsed into the 9 criterion symptom domains that define MDD according to the Diagnostic and Statistical Manual of Mental Disorders. Those 9 symptoms (i.e., value of $p$) score from 0 (i.e., no problem) to 3 points (i.e., severe problem) resulting in $L = 4$ ordinal levels. In this setting, we choose to infer the partial polychoric correlations among $p$ variables starting from the 9 symptoms collected in the STAR*D dataset. More precisely, for different values of the number of symptoms $p$, we sample $p$ variables from the original dataset and estimate their partial polychoric

correlations with the function `cor_auto()` from the `R` package `qgraph` (Epskamp et al., 2012). Note that the result is not sparse and no value will be zero (i.e. the density of the underlying graph is 1). Then, starting from this matrix, say $\Theta_{\text{real}}$, we input it in the `BDgraph` package (Mohammadi et al., 2024) in order to generate datasets with this fixed partial correlation structure, exactly as in the first simulation setting (but skipping the random generation of $\Theta$).

*Third simulation setting: independent variables (no correlation)*

We consider the case of independent variables, i.e. the correlation matrix is the identity matrix and the partial correlation graph is empty. This is an unrealistic setup, that we keep as a control experiment, in order to understand how the methods behave in this extreme case where there is no signal in the data. As this scenario is easier to run for the inference methods, we could investigate the extra case of $p = 20$ variables.

*Fourth simulation setting: non polychoric correlation structure*

Following the example of Liu et al. (2021), we explore a heterogeneous dataset that deviates from the polychoric correlations based setting in that the underlying continuous variables are not jointly Gaussian. More precisely, we rely on the mixture of two $p$-dimensional Gaussian distributions, further binned into levels. We thus consider the mixture with equal proportions of samples drawn as in the first simulation setting: `prob1,prob2` are the sparsity levels of each component partial correlation graph. For each component of the mixture, we draw the non-zero entries of the covariance matrices $\Sigma_1, \Sigma_2$ according to a $G$-Wishart distribution with respective sparsity levels `prob1,prob2`. Then we draw two $p$-dimensional Gaussian distributions with corresponding covariances $\Sigma_1, \Sigma_2$. Binding the two samples gives us the desired mixture of two-components $p$-dimensional Gaussian distributions. Binning the Gaussian variables into discrete levels then further creates ordinal variables.

In this setting, there is no "true" partial correlation for the mixture. Instead, we take the empirical partial correlation of the Gaussian random variables as the true value for $\Theta$. As a consequence, our proxy for the true correlation graph has density 1 (no estimated correlation is 0,

all edges are present).

## Appendix C: Methods compared

We grouped the 14 methods into 4 different groups.

### *Group 1: Direct estimation of partial correlations*

In this group of methods, we first rely on polychoric correlations, for which there are 2 different estimators and we contrasted those 2 methods with a computation of Pearson's partial correlations, that treat the data as continuous variables.

Our first method `poly.mle` is a maximum likelihood estimator (MLE) of the polychoric correlation, as proposed in Olsson (1979). It is implemented in function `cor_auto` from the package `qgraph` (Epskamp et al., 2012). The second method `poly.wls` is based on a weighted least squares estimator of the polychoric correlation. It's implemented in the package `psychonetrics` (Epskamp, 2024), using function `ggm`. The third and last method in this group, `pears`, simply estimates Pearson's partial correlations, through its empirical estimator.

### *Group 2: methods from `qgraph`*

This group contains 2 categories of methods, that both correspond to finding the ML estimator in a GGM and using EBIC criterion for model selection: the `EBICGlasso` proposed in Epskamp and Fried (2018) and the `ggmModSelect` method discussed in (Isvoranu and Epskamp, 2023). Each method is available in 2 versions, either relying on polychoric or on Pearson correlations estimates as input. We call the methods `Glasso.poly, Glasso.pears` for the first 2 based on `EBICGlasso` and `ggmMS.poly, ggmMS.pears` for the 2 others based on `ggmModSelect`. In the case of polychoric correlations, we use as input for the methods the output from the `poly.mle` method above, as it is the most widely used method.

### *Group 3: methods from `GGMnonreg` and `BGGM`*

This category first includes methods from the `R` package `GGMnonreg` (Williams, 2021b) implementing estimators proposed in Williams et al. (2019, 2020). Three variants are considered. The

first variant assumes a GGM and first performs $p$ independent (non regularized) regressions of each of the variables on the others. Then, it relies on the neighborhood selection procedure (Meinshausen and Bühlmann, 2006) to obtain a sparse estimate. We call this method `GGMnr.neighsel`. Note that this method implicitly relies on Pearson's partial correlations (because each of the $p$ regression procedures estimates a Pearson's correlation coefficient). The second variant relies on a bootstrap procedure. It starts performing a singular valued decomposition (SVD) of the empirical covariance matrix $\hat{\Sigma}$ and use this SVD to define a generalized inverse $\hat{\Theta}$ of $\hat{\Sigma}$ and thus corresponding partial correlations. Note that up to this point, it corresponds exactly to the proposal of Schäfer and Strimmer (2004) discussed below. Also at this stage, the graph is complete (all edges are present because none of the estimated partial correlations is 0). Finally, relying on bootstrap, one obtains a confidence interval on each partial correlation coefficient and use this for thresholding the coefficients and sparsifying the estimator. We call this method `GGMnr.boot.pears`. Note that Schäfer and Strimmer (2004) also used a bootstrap step, however in a different way, to robustify their estimator instead of thresholding it. The last and third variant is same as previous, relying on polychoric correlations. We thus call it `GGMnr.boot.poly`.

We additionally consider in this group 2 methods from the `BGGM` (Bayesian Gaussian Graphical Models) package (Williams and Mulder, 2019) that were also included in the comparison by Isvoranu and Epskamp (2023), namely the functions `BGGM_explore` and `BGGM_estimate`.

*Group 4: methods from `geneNet` and `PAsso`*

We consider in this group the method of Schäfer and Strimmer (2004). Originally developed in the context of gene associations networks, it is a regularization method for GGMs. It corresponds to the function `ggm.estimate.pcor` from the `R` package `GeneNet` (Schäfer et al., 2021). We call the method `ggmSS`.

Liu et al. (2021) claim that relying on polychoric associations for ordinal variables is a bad idea. They propose a new method, based on surrogate residuals. It consists in simulating a continuous latent variable (with some specific distribution) and construct the corresponding residual (at the end, the results are averaged over many simulations, around 30 simulations). Then, many different types of correlations can be looked at for these residuals (they propose 3 different measures, that

capture not only linear correlations). The method corresponds to the `R` package `PAsso` (Zhu et al., 2021). We call it `PAsso`.

## Appendix D: Additional figures

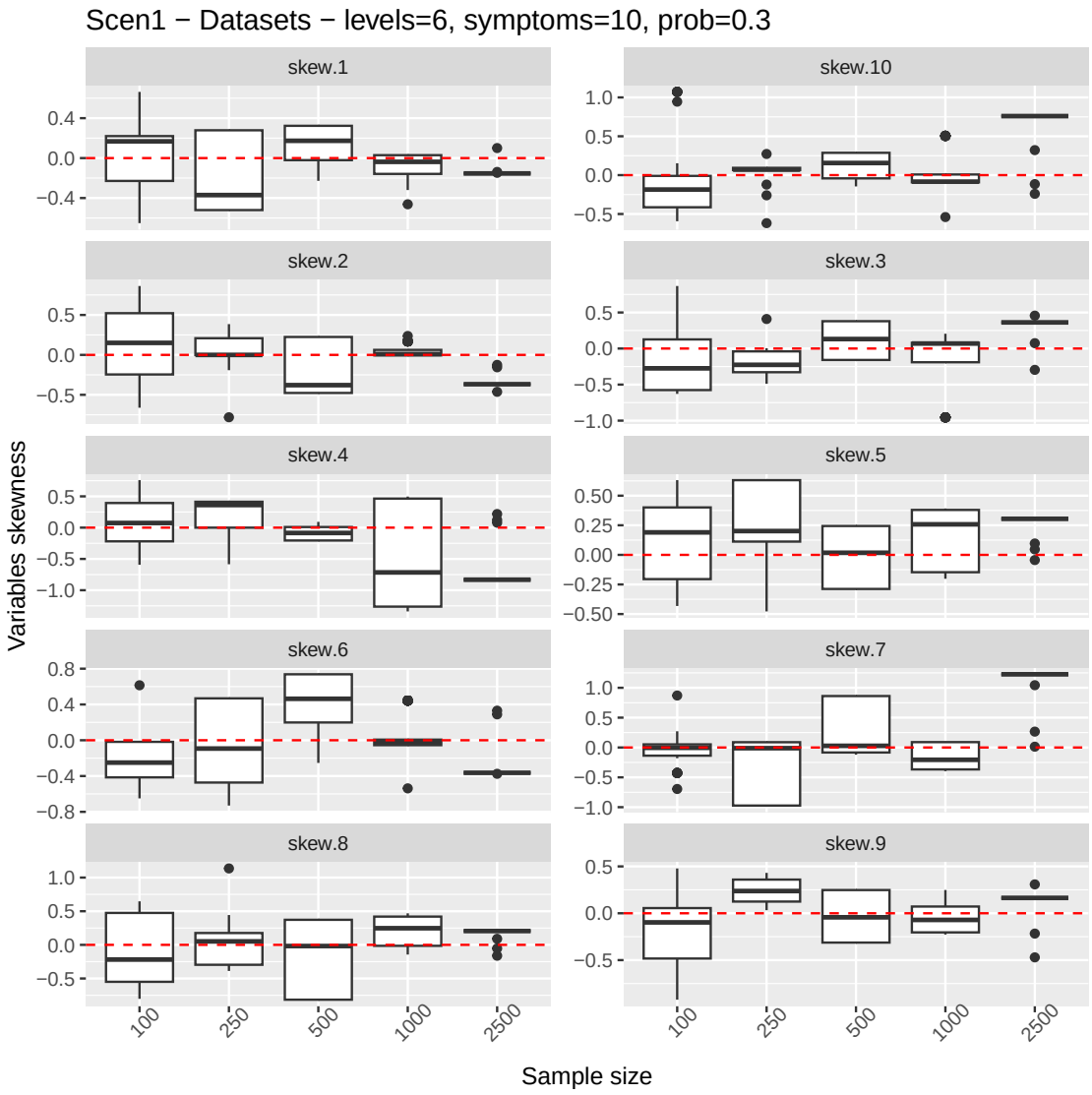


FIGURE A.

Scenario 1: An example of the skewness values (boxplots over 100 replicates) of the distributions of $p = 10$ symptoms (displayed as 10 graphics) for the different values of sample size $n \in \{100; 250; 500; 1,000; 2500\}$ (displayed on the $x$-axis). The setting corresponds to $L = 6$ ordinal levels and density of the true partial correlation graph `prob`=0.3.

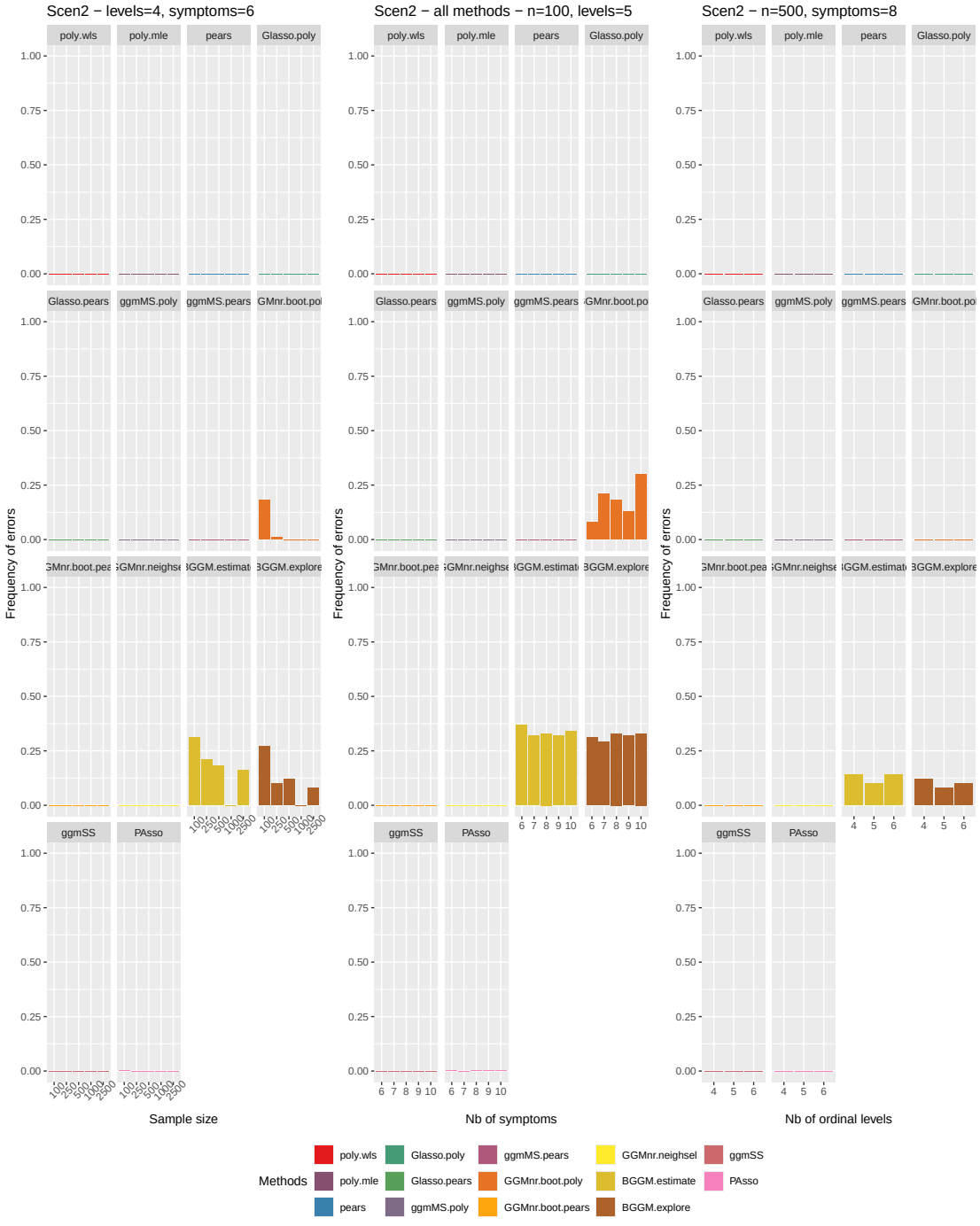


FIGURE B.

Scenario 2: Examples of execution error rates (averaged over 100 replicates) of all the methods (displayed as 14 graphics in each picture). The $x$-axis displays sample sizes (left picture), number of symptoms (center picture) and number of levels (right picture).

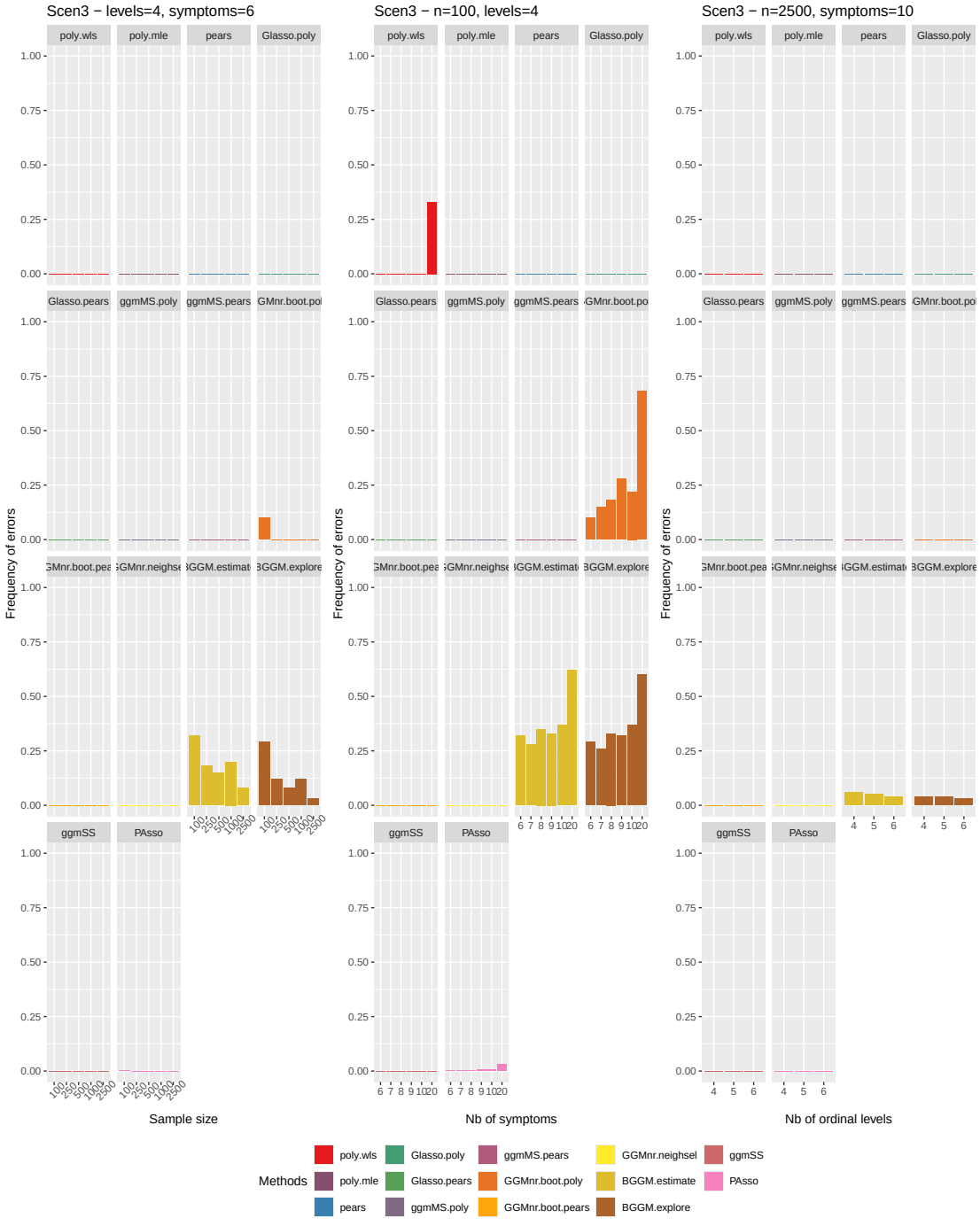


FIGURE C.

Scenario 3: Examples of the execution error rates (averaged over 100 replicates) of all the methods (displayed as 14 graphics in each picture). The $x$-axis displays sample sizes (left picture), number of symptoms (center picture) and number of levels (right picture).