Multi-omics network inference with a Gaussian copula model

Ekaterina Tomilina, Gildas Mazo, Florence Jaffrézic

Multi-omics network inference with a Gaussian copula model

Ekaterina Tomilina[1,2*], Gildas Mazo[1] and Florence Jaffrézic[2]

[1*]Université Paris-Saclay, INRAE, MaIAGE, Jouy-en-Josas, 78350, France.

[2]Université Paris-Saclay, INRAE, AgroParisTech, GABI, Jouy-en-Josas, 78350, France.

*Corresponding author(s). E-mail(s): ekaterina.tomilina@inrae.fr;
Contributing authors: gildas.mazo@inrae.fr; florence.jaffrezic@inrae.fr;

Abstract

Background: Inferring partial correlation networks is essential in systems biology to uncover direct interactions between biological entities. Traditional Gaussian graphical models rely on the assumption of normally distributed data, which limits their applicability to multi-omics datasets comprising heterogeneous data types such as continuous and discrete variables.

Results: We propose a novel likelihood-based approach for network inference using a Gaussian copula model with semiparametric pairwise-likelihood estimation of the correlation matrix. The inferred correlation structure is then inverted and regularized via the graphical lasso to recover partial correlations. Compared to a moment-based approach employing bridge functions, our method demonstrates significantly improved computational efficiency and estimation accuracy, particularly for discrete data with many categories and/or large values, such as count data. This result is important for biological applications, especially for the integration of RNA-seq count data. An application to a breast cancer data set from the International Cancer Genome Consortium (ICGC) successfully identified biologically relevant interactions.

Conclusions: The proposed approach, based on the Gaussian copula and likelihood-based estimation, provides a novel, effective and computationally efficient mathematical framework for integrative multi-omics data analysis and network inference.

Keywords: Partial correlation networks, multi-omics data, Gaussian copula, Graphical lasso

1 Introduction

Systems biology is based on the analysis of complex, large-scale data of diverse nature. A key biological challenge is to understand the interactions and regulatory links between the different types of omics data. Their heterogeneity constitutes a major analysis issue. Some can easily be modeled with Gaussian distributions (transcriptomic data from microarrays), others are continuous but non-Gaussian (epigenomic data from methylation chips), or discrete (transcriptomic and epigenomic data from sequencing, genotyping). However, mathematical and statistical models are highly dependent on the nature of the data, which is why the models proposed to date have mainly been carried out independently for each omics level.

When the data are Gaussian, the networks are assimilated to a set of conditional independence relationships between variables. These relationships are encoded in the inverse of the covariance matrix, called the precision matrix, of the underlying Gaussian distribution. In this case, inference is made by penalized maximum likelihood. A commonly used algorithm for that purpose is called glasso [1]. When the data are continuous, but not necessarily Gaussian, a non-linear scaling of the data allows to recover Gaussian data [2]. This approach has been used, for example, to analyze microarray data. When the data are discrete, like the counts generated by RNA-seq technology, there is no one-to-one transformation to recover Gaussian data. An alternative is to build a hierarchical model, such as Poisson models based on latent Gaussian variables [3]. The network is reconstructed from the precision matrix of latent variables. This approach has the advantage of not requiring any transformation, but is limited to the analysis of one type of data with strong parametric assumptions regarding the observed distribution.

In the context of multi-omics network inference, it is necessary to be able to study the relationships between data of various types (Gaussian, continuous non-Gaussian, counts, etc.) Two main approaches have been proposed to tackle this issue [4]. The first one corresponds to Mixed Graphical Models, which are an extension of Gaussian graphical models allowing to take into account links between discrete and continuous variables by regressing each variable with respect to the others. It corresponds to an extension of the graphical lasso, with one or more regularization parameters [5, 6]. However, they require strong parametric assumptions on the marginal distributions. The second one corresponds to Gaussian copula-based graphical models that are particularly well-suited for the integration of data with various types and can be extended to the semi-parametric case where only the copula parameter is to be estimated, with no assumptions regarding the marginals. Parameter inference for these models has so far been mostly performed in a Bayesian framework by MCMC approaches [7, 8], which have a high computational cost. Some methods also rely on bridge functions that link an extension of Kendall's tau (estimated on the observed variables) to the copula correlation coefficient [9].

The goal of this article is to propose a parameter inference algorithm based on the pairwise likelihood. We estimate the copula parameters directly from the observed data before applying the graphical Lasso in order to invert and regularize the inferred correlation matrix. The proposed algorithm is evaluated in an extensive simulation study in which we compare our performance to bridge function methods, and then

applied to real biological data from the International Cancer Genome Consortium [10]. The method is implemented in an R package called `heterocop`, available on the CRAN.

2 Methods

2.1 Gaussian copula model

Let $X^i = (X_1^i, \ldots, X_d^i)$, $i = 1, \ldots, n$, be a sample of i.i.d. observations in $\mathbb{R}^d$, and let F denote the cumulative distribution function (CDF) of X^i. We assume

$$\begin{aligned} F(x_1, ..., x_d) &= C_\Sigma(F_1(x_1), \ldots, F_d(x_d)) \\ &\equiv \Phi_\Sigma(\Phi^{-1}(F_1(x_1)), ..., \Phi^{-1}(F_d(x_d))) \end{aligned} \tag{1}$$

where $F_1, \ldots, F_d$ denote the marginal CDFs of $X_1^i, \ldots X_d^i$, Φ_Σ the centered Gaussian multivariate CDF of correlation matrix Σ, and Φ^{-1} the inverse standard Normal CDF.

In other words, model (1) corresponds to a latent Gaussian variable structure. Let $Z^i = (Z_1^i, ..., Z_d^i) \sim \mathcal{N}(0, \Sigma)$ denote a centered standardized Gaussian vector of correlation matrix Σ. Let $F_j^{\leftarrow}$ denote the generalized inverse function such that $F_j^{\leftarrow}(u) = \inf\{x : F_j(x) \geq u\}$. If $X_j^i = F_j^{\leftarrow}(\Phi(Z_j^i))$, then it can be shown that X^i has the CDF given by (1).

We are interested in the estimation of the partial covariances between the latent Gaussian variables Z^i,

$$Cov(Z_j^i, Z_{j'}^i | Z_{-(j,j')}^i) = \mathbb{E}(Z_j^i Z_{j'}^i | Z_{-(j,j')}^i) - \mathbb{E}(Z_j^i | Z_{-(j,j')}^i)\mathbb{E}(Z_{j'}^i | Z_{-(j,j')}^i), \tag{2}$$

where $Z_{-(j,j')}^i$ denotes the set of variables $Z_1^i, \ldots, Z_d^i$ without Z_j^i and $Z_{j'}^i$. Note that for a given pair of variables Z_j^i and $Z_{j'}^i$, this value is deterministic because it does not depend on the values taken by $Z_{-(j,j')}^i$ [11]. It can be shown that

$$\begin{aligned} Cor(Z_j^i, Z_{j'}^i | Z_{-(j,j')}^i) &= \frac{Cov(Z_j^i, Z_{j'}^i | Z_{-(j,j')}^i)}{\sqrt{Var(Z_j^i | Z_{-(j,j')}^i) Var(Z_{j'}^i | Z_{-(j,j')}^i)}} \\ &= \frac{-\Omega_{jj'}}{\sqrt{\Omega_{jj}}\sqrt{\Omega_{j'j'}}}, \end{aligned}$$

where $\Omega_{jj'}$ denotes the element at the j-th row and j'-th column of the precision matrix $\Omega = \Sigma^{-1}$.

2.2 Pairwise-likelihood and Graphical Lasso

The first step is to obtain an estimation of matrix Σ. We propose to use the pairwise pseudo maximum likelihood estimator (PPMLE) studied by Mazo et al. [12], extended

to the case of non-parametric marginals [13]:

$$\hat{\Sigma} = \underset{\Sigma}{\operatorname{argmax}} \left(\frac{1}{n} \sum_{i=1}^{n} \sum_{j<j'} \log \hat{f}_{jj'}(X_j^i, X_{j'}^i, \Sigma_{jj'}) \right) \tag{3}$$

where $\hat{f}_{jj'}(\cdot, \cdot, \Sigma_{jj'}) = f_{jj'}(\cdot, \cdot, \hat{F}_j, \hat{F}_{j'}, \Sigma_{jj'})$ denotes an estimate of $f_{jj'}(\cdot, \cdot, F_j, F_{j'}, \Sigma_{jj'})$, the density of the bivariate marginal CDF from model (1) corresponding to the pair $(X_j, X_{j'})$ with respect to the $\lambda \otimes \lambda$ measure if both variables are continuous, $\mu \otimes \mu$ measure if both variables are discrete, and $\lambda \otimes \mu$ measure if X_j is continuous and $X_{j'}$ is discrete. λ denotes the Lebesgue measure and μ denotes the counting measure. Note that the marginal CDFs F_j and $F_{j'}$ have been replaced by their empirical counterparts $\hat{F}_j$ and $\hat{F}_{j'}$. The expressions of the densities $\hat{f}_{jj'}$ can be found in Appendix B.2. This approach does not require any parametric assumption on the marginals, and handles the cost of dimensionality by a pairwise approach. It also has the advantage of estimating $\hat{\Sigma}$ from the observed data in a single step.

In order to obtain the partial correlations, once an estimator $\hat{\Sigma}$ of Σ is obtained, it can be inverted via a penalized approach such as graphical Lasso (gLasso) [1] as shown in equation (4).

$$\hat{\Omega}_\lambda = \underset{\Omega}{\operatorname{argmin}} \left(\operatorname{tr}(\hat{\Sigma}_{PD}\Omega) - \log(|\Omega|) + \lambda ||\Omega||_1 \right) \tag{4}$$

where $\hat{\Sigma}_{PD}$ denotes the projection of $\hat{\Sigma}$ on the space of positive definite matrices. Note that the expression from equation (4) stems from maximum likelihood inference of Ω in the latent space as described in Appendix C.

2.3 Comparison with a moment-based method

Another possible approach to estimate Σ is the use of bridge functions [9, 14, 15], which are based on a one-to-one correspondence between a statistic of the observed data, which we denote by r, and the matrix Σ. Note that this method does not require either any parametric assumption on the marginal distributions.

The first step consists in the estimation of r, which is a vector with as many components $r_{jj'}$ as there are pairs of variables $(X_j, X_{j'})$. For each pair, the estimator $\hat{r}_{jj'}$ of $r_{jj'}$ depends on the nature of the variables. When both variables X_j and $X_{j'}$ are continuous, it actually coincides with Kendall's τ and is computed as

$$\hat{r}_{jj'} = \frac{2}{n(n-1)} \sum_{1 \le i < i' \le n} \operatorname{sign}(X_j^i - X_j^{i'}) \operatorname{sign}(X_{j'}^i - X_{j'}^{i'}) \tag{5}$$

When X_j is continuous and $X_{j'}$ is discrete, it is computed as

$$\hat{r}_{jj'} = \frac{2}{n(n-1)} \sum_{1 \le i < i' \le n} \operatorname{sign}(X_j^i - X_j^{i'})(X_{j'}^i - X_{j'}^{i'}) \tag{6}$$

Finally, when the pair is discrete, it is computed as

$$\hat{r}_{jj'} = \frac{2}{n(n-1)} \sum_{1 \leq i < i' \leq n} (X_j^i - X_j^{i'})(X_{j'}^i - X_{j'}^{i'}). \tag{7}$$

In a second step, estimation of Σ is performed by solving $F(\hat{\Sigma}_{jj'}) = \hat{r}_{jj'}$ for $\hat{\Sigma}_{jj'}$, where F here momentarily denotes the so-called bridge function that makes the one-to-one correspondence. The expressions of the bridge functions for a continuous, discrete, mixed pair, denoted by F^{cc}, F^{dd}, F^{dc}, respectively, can be found in Appendix B.1.

However, several obstacles can arise. Indeed, the equation $F(\hat{\Sigma}_{jj'}) = \hat{r}_{jj'}$ may not have any solution in the interval $[-1, 1]$, in which case a manual correction may be needed. Moreover, the computation time of the bridge function increases with the number of categories of the discrete variables.

3 Simulation study

The goal of this simulation study is to compare the performance of our proposed pairwise pseudo likelihood-based method and the moment-based approach with bridge functions.

3.1 Simulation protocol

First, we simulate a partial correlation structure $\tilde{\Omega}$. We set a $d \times d$ matrix of zeroes and we choose a number r of non-null coefficients. Then, we randomly position $r/2$ non-null coefficients via a Watts-Storgatz approach [16] in the upper triangular part of the matrix. We fill them with uniform values $\mathcal{U}(0.2, 0.7)$ and we symmetrize our matrix before setting its diagonal to 1. Finally, we make it nonnegative definite by adding the absolute value of its smallest eigenvalue to its diagonal. We compute $\tilde{\Sigma} = \tilde{\Omega}^{-1}$ and standardize it to be a correlation matrix Σ as required in model (1). The detailed algorithm is presented in Appendix A. Once generated, the matrices Σ and $\Omega = \Sigma^{-1}$ are kept fixed throughout the whole simulation study.

We simulated a Gaussian vector $(Z_1, \ldots, Z_d) \sim \mathcal{N}(0, \Sigma)$ of correlation matrix Σ. Then, we considered two simulation protocols in order to obtain our data set $(X_1, \ldots, X_d)$. The first one is similar to the setting presented by [9]. We simulated a data set that contains one half of Gaussian variables and another half of discrete variables with at most three categories. To do so, we simulated $d/2$ uniform cutoff values $C_{j1} \sim \mathcal{U}(0.25, 0.85)$ and $d/2$ uniform cutoff values $C_{j2} \sim \mathcal{U}(1.5, 2)$. We set $X_j = Z_j$ for $j = 1, \ldots, d/2$, and then we set $X_j = \mathbb{1}(Z_j > C_{j1}) + \mathbb{1}(Z_j > C_{j2})$ for $j = d/2 + 1 \ldots, d$.

We then consider a more realistic second simulation protocol to complexify the structure of the dataset, in which one third of the variables are Gaussian $\mathcal{N}(0, 1000)$, one third are discrete following a Negative Binomial distribution $NB(1000, 0.3)$ and one third are binary $\mathcal{B}(\frac{1}{2})$. For $j = 1, \ldots, d$, we set $X_j = F_j^{\leftarrow}(\Phi(Z_j))$, where $F_j^{\leftarrow}$ corresponds to the generalized inverse of the corresponding marginal distribution.

In our simulation study, we have computed $N = 100$ replications, for $d \in \{30, 300\}$ variables, for sample sizes $n \in \{20, 50, 200, 500\}$. On each generated dataset, Σ was

estimated according to the two inference methods from section 2. Then, a family of estimates $\hat{\Omega}_\lambda$ was calculated as in equation (4) for a grid of $\lambda \in \log\{1.01, 1.02 \ldots, 2.99, 3\}$ for $d = 30$, and $\lambda \in \{0.05, 0.06 \ldots, 0.74, 0.75\}$ for $d = 300$.

3.2 Performance Metrics

Three different performance metrics were computed in order to compare the two methods. First, the computational efficiency was evaluated by recording the CPU time for the computation of $\hat{\Sigma}$ and its inversion for one value of λ. Note that for the bridge functions estimation method (subsection 2.3), the original code can be found on https://github.com/Aiying0512/LGCM and corresponds to the article by Zhang et al [9]. We have parallelized it in order to have a fair comparison of computation time with the code for the PPMLE estimation method that is implemented in the `heterocop` R package [17], with 19 cores used by default.

Our main goal is to construct a biological network, in which an edge between two nodes corresponds to a non-null latent conditional correlation between the corresponding variables. A key step therefore consists in recovering the set of non-null conditional correlations between the latent variables. The simulated precision matrix Ω has a sparsity (proportion of zeroes) of about 0.8 for $d = 30$, corresponding to 75 non-zeroes to be identified, and around 0.98 for $d = 300$, corresponding to 750 non-zeroes. A zero corresponds to no conditional correlation, while a non-zero corresponds to a non-null conditional correlation. For each estimation $\hat{\Omega}_\lambda$, the penalization enables to set the estimated coefficients exactly to zero without a thresholding step. Note that the proportion of estimated zeroes increases along with the penalization parameter λ. For each λ, two main indicators were measured. The sensitivity, also known as true positive rate $\mathrm{TPR}(\lambda)$, represents the proportion of correctly detected non-zeroes in $\hat{\Omega}_\lambda$ among the true non-zeroes in Ω. Similarly, the specificity, also known as the true negative rate $\mathrm{TNR}(\lambda)$, corresponds to the proportion of correctly detected zeroes among the true zeroes. More specifically, we are going to consider the false positive rate, $FPR(\lambda) = 1 - TNR(\lambda)$. The Receiver Operating Characteristic (ROC) curve is a graphical representation of the FPR against the TPR depending on the parameter λ. When λ is high, $\hat{\Omega}_\lambda$ is strongly penalized and all non-diagonal coefficients are zero, leading to $TPR(\lambda) = 0$ and $FPR(\lambda) = 0$. When λ is low, $\hat{\Omega}_\lambda$ is not penalized and there might be no zero coefficients, leading to $TPR(\lambda) = 1$ and $FPR(\lambda) = 1$. The Area Under Curve (AUC) criterion computes the area under the ROC curve in order to evaluate the performance of the classifier.

Finally, to assess the performance of the estimator $\hat{\Sigma}$, we shall examine the distribution of the squared distances between the replicated estimates $\hat{\Sigma}^k$, $k = 1, \ldots, N$ and the true Σ, given by:

$$\sum_{j<j'} (\Sigma_{jj'} - \hat{\Sigma}^k_{jj'})^2. \tag{8}$$

3.3 Simulation results

First, the results are presented for $d = 30$ variables, for both simulation protocols, for N=100 replications.

3.3.1 Comparison of computation times

Figure 1 shows that for the first simulation protocol, when the discrete variable only has three categories $\{0, 1, 2\}$, the bridge function performs similarly until $n = 200$, and slightly better for larger sample sizes n, needing under 10 seconds per iteration when $n = 500$ while the PPMLE estimation needs about 30 seconds for this sample size. A much larger difference is observed in the second simulation protocol. Indeed, this simulation setting leads to a discrete variable that can take several dozens of discrete values between 2000 and 2500. In this case, the computational time of the bridge function method drastically increases from around two minutes for $n = 50$ to over ten minutes when $n = 200$, and close to an hour per iteration for $n = 500$, while the PPMLE estimation remains, on average, under two minutes for all sample sizes.

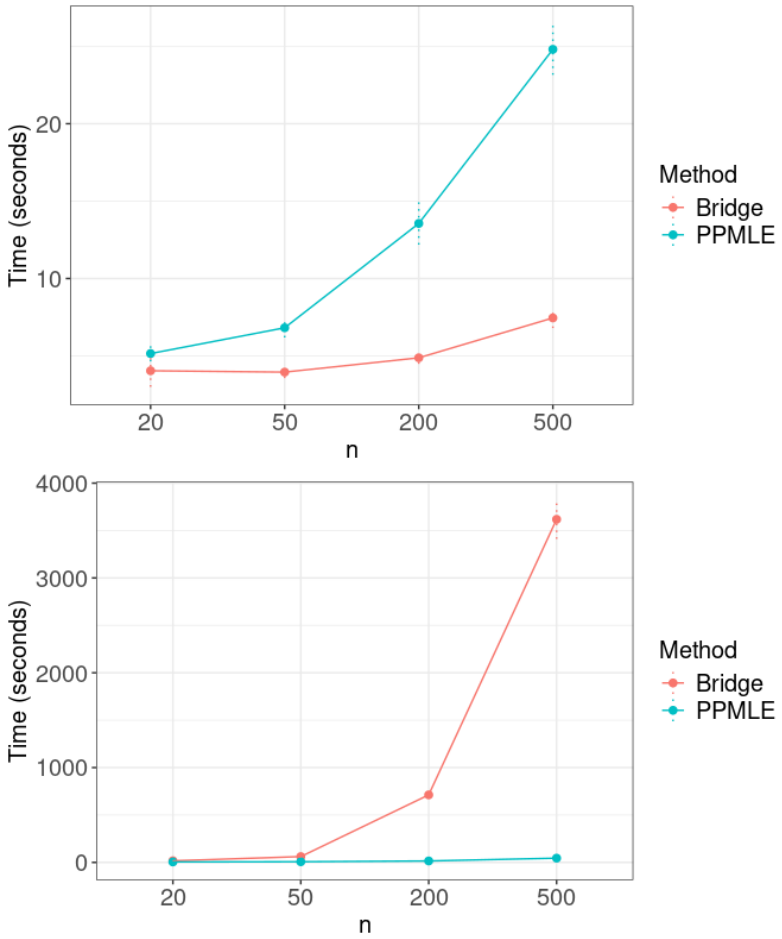

Fig. 1: Computation time for $d = 30$ variables, for both methods, for the first simulation protocol (top) and for the second simulation protocol (bottom).

3.3.2 AUC values

The presence of conditional correlations has been detected via ROC curves, which AUC values for both simulation protocols and for both methods are given in Table 1. We can see that in the case of the first simulation protocol, the results are quite similar, even if the PPMLE estimation seems to perform slightly better for $n = 20$ and $n = 50$. Larger differences are observed for the second scenario, especially when discrete variables are involved, as presented in Table 2. An explanation might be that bridge functions are not well suited for the analysis of categorical data when the number and values of the categories can be large, which is typically the case for count data, especially for RNA-seq data which are often assumed to come from negative binomial distributions.

	Sample size	20	50	200	500
Scenario 1	PPMLE	0.66	0.70	0.76	0.78
	Bridge	0.62	0.67	0.75	0.77
Scenario 2	PPMLE	0.67	0.71	0.77	0.78
	Bridge	0.63	0.66	0.72	0.76

Table 1: Estimated AUC for $d = 30$ variables, for each method, sample size and scenario. The standard errors of the estimates were calculated and were found to be less than 10^{-2}.

	AUC on discrete pairs		AUC on CD pairs	
Sample size	PPMLE	Bridge	PPMLE	Bridge
20	0.65	0.60	0.84	0.73
50	0.70	0.60	0.91	0.76
200	0.77	0.64	0.93	0.83
500	0.80	0.69	0.93	0.92

Table 2: Estimated AUC for $d = 30$ variables, over the discrete-discrete and continuous-discrete (CD) pairs, in the second simulation scenario, for each method and sample size. The standard errors of the estimates were calculated and were found to be less than 10^{-2}

3.3.3 Squared Error

The overall Squared Error (SE) values, computed as in equation (8) for both simulation scenarios, are given in Figure 2. In the first simulation scenario, both methods perform similarly, except for a sample size of 20 for which the bridge functions are slightly better. On the other hand, for the second simulation scenario, for which the discrete data have larger values and number of categories, much better performances

are observed for the proposed PPMLE approach, as shown in Figure 2. This difference is even more drastic in the presence of discrete variables, either for discrete-discrete pairs or continuous-discrete pairs, as presented in Figure 3.

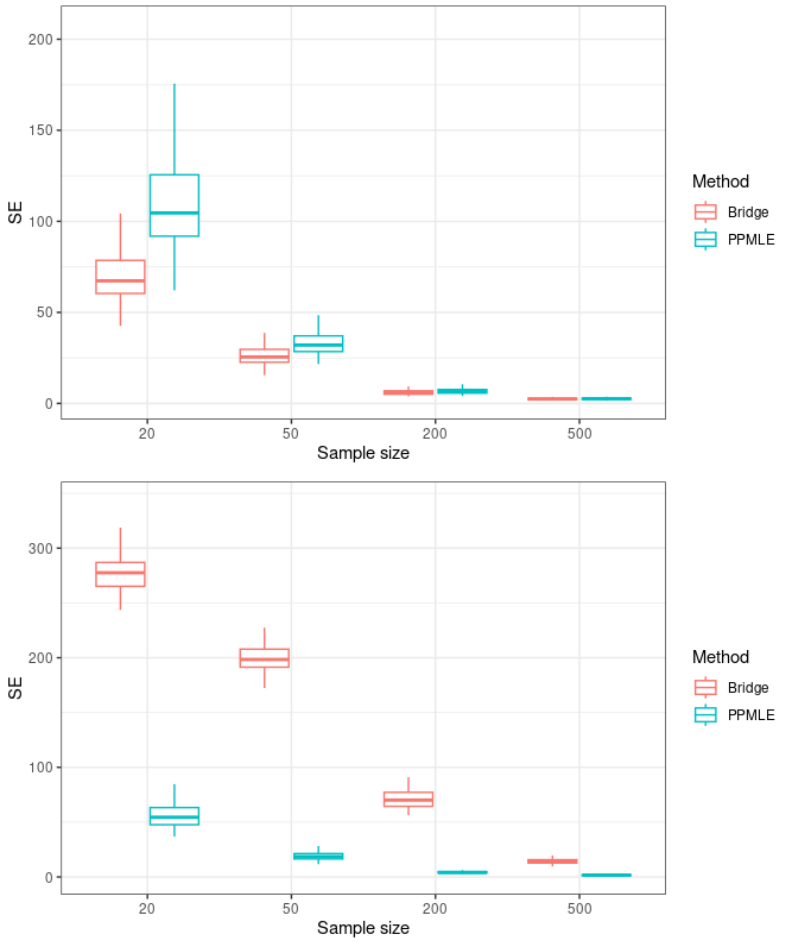

Fig. 2: Boxplot of the $N = 100$ SE values for $d = 30$ for scenarios 1 (top) and 2 (bottom).

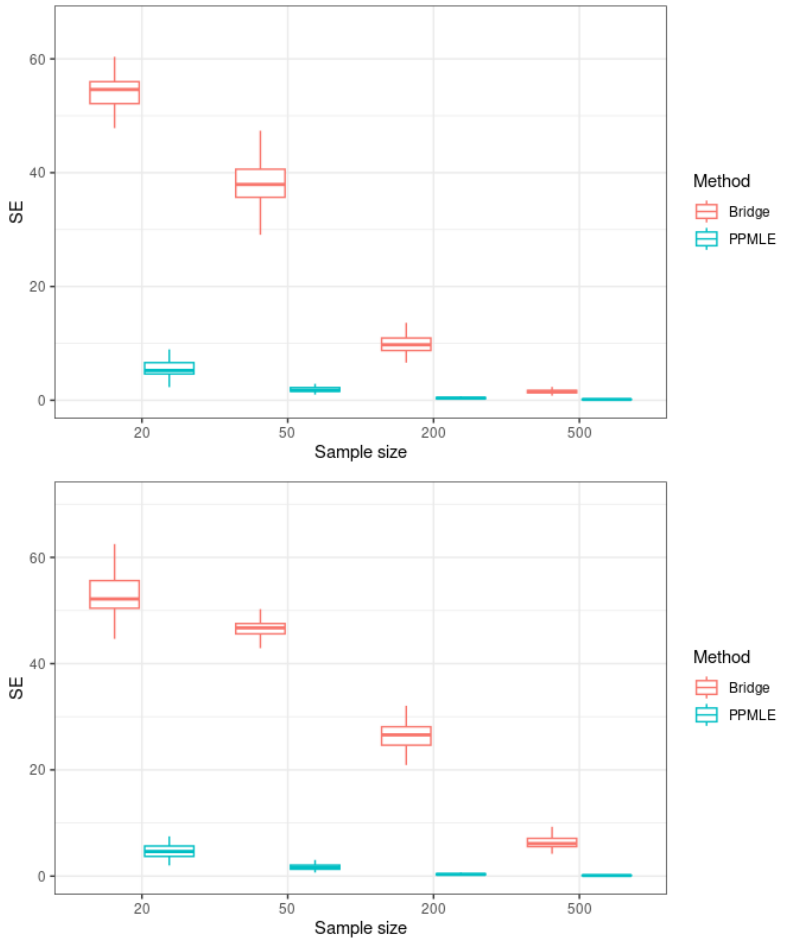

Fig. 3: Boxplot of the $N = 100$ SE values for $d = 30$ for continuous-discrete (top) and discrete-discrete (bottom) pairs in the second simulation scenario.

3.3.4 Simulation results for $d = 300$

Simulations were run for a larger number of variables ($d = 300$). Figure 4 shows the average computation time for both simulation scenarios and both estimation methods. For the first simulation scenario where the discrete variables only take values in $\{0, 1, 2\}$, the computation time is lower for the bridge function estimator for all sample sizes. Both remain under five minutes for $n = 20, 50, 200$. In the second simulation scenario, on the other hand, in which the discrete variables have a much larger number of categories, the computational time required for the bridge function approach is much larger than for the PPMLE approach, which prevents to run this method for sample sizes larger than 50 in this simulation study.

The estimated AUC values and the distributions of the SE values are presented in Table 3 and Figure 5. Both methods present similar results for the first simulation scenario. For the second scenario, the bridge function approach could not be evaluated for sample sizes larger than $n = 50$, due to the required computing time, which limits the interpretation of the results. Our approach performs well in terms of AUC values and SE even for $d = 300$ variables for the different sample sizes evaluated, up to $n = 500$. The computational time required for the bridge functions for the analysis of such data is, however, a great difficulty as it prohibits its use for real data analysis, such as RNA-seq data, as presented in the section below.

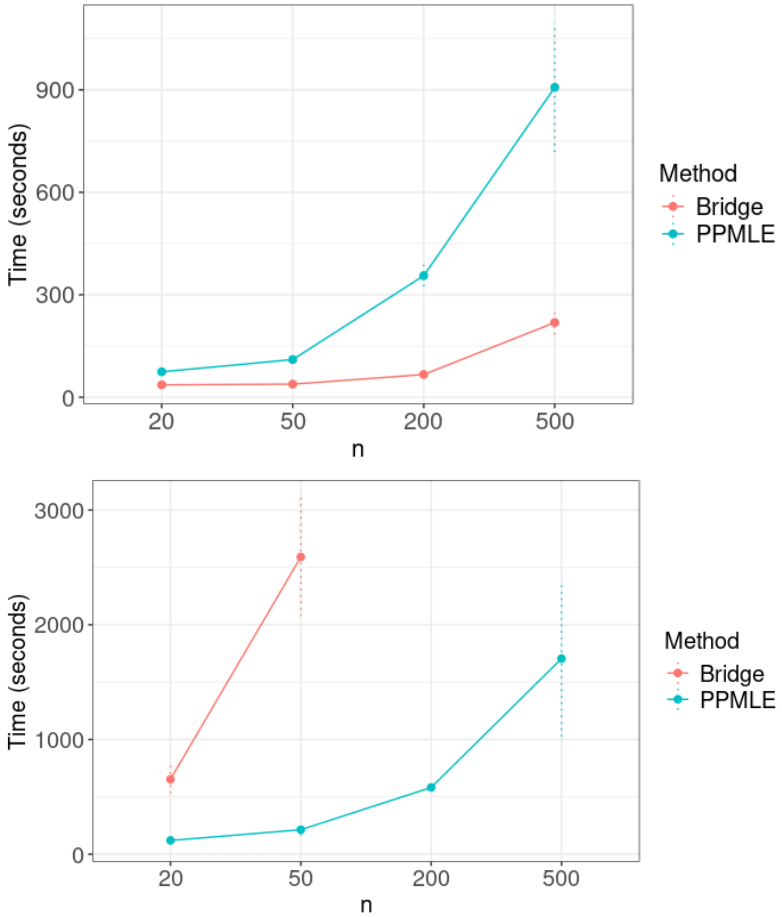

Fig. 4: Computation time for $d = 300$ variables, for both methods, for the first simulation protocol (top) and for the second simulation protocol (bottom).

	Sample size	20	50	200	500
Scenario 1	PPMLE	0.63	0.71	0.84	0.90
	Bridge	0.62	0.70	0.84	0.90
Scenario 2	PPMLE	0.65	0.74	0.87	0.92
	Bridge	0.64	0.70	–	–

Table 3: Estimated AUC for $d = 300$ variables, for each method, sample size and scenario. The standard errors of the estimates were calculated and were found to be less than 10^{-2}.

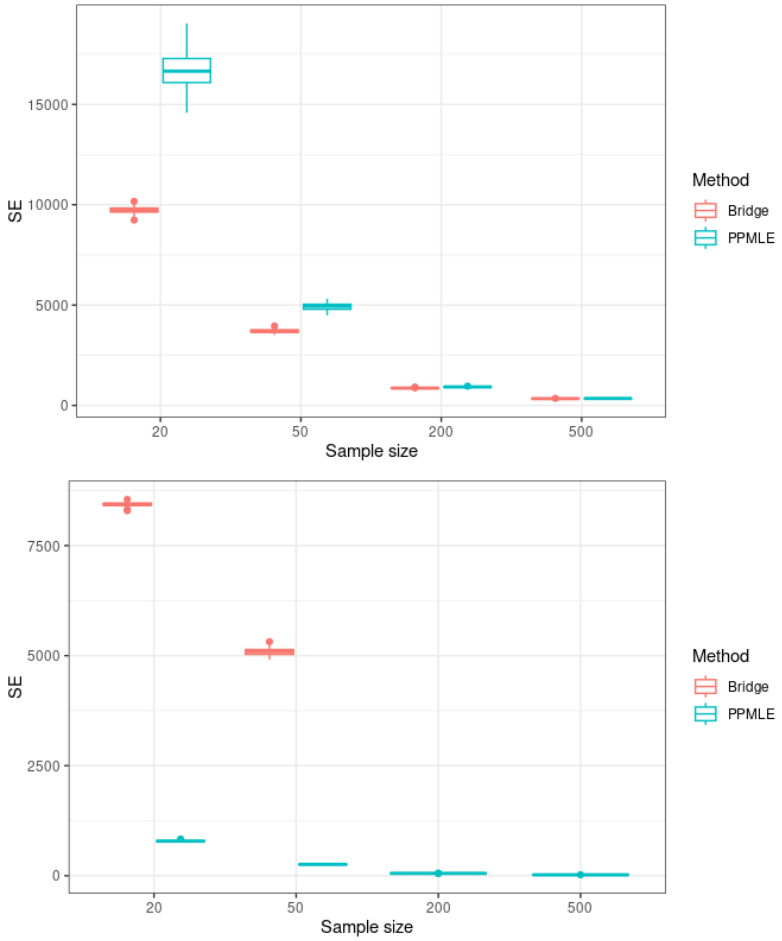

Fig. 5: Boxplot of the $N = 100$ SE values for $d = 300$ for scenarios 1 (top) and 2 (bottom)

4 Real data application

To evaluate the practical performance of our proposed inference method, we applied it to breast cancer multi-omics data from the International Cancer Genome Consortium (ICGC) [10]. This dataset contains various molecular profiles, including gene expression, protein abundance and mutation data, providing a comprehensive view of tumor heterogeneity. Our objective was to infer regulatory interactions between these molecular features and identify key network structures associated with breast cancer. The pre-processed dataset used in this study is available on HAL [13]. Our variables of interest includes protein expression (continuous), RNA sequencing (discrete) and mutations (binary) measured on breast cancer tumoral tissue. The initial data set included normalized protein abundance for 115 genes measured on 260 individuals, RNA-seq counts for 20 501 genes observed on 939 individuals, and presence of 107 249 mutations observed on 918 individuals. As there were several samples per individual, we averaged the RNA-seq and protein expression values. The binary variables encoded if the mutations were present for at least one of the samples. Then, we used the `DESeq2` R package [18] to normalize the RNA-seq counts. The intersection of available data for all type of variables left us with 250 individuals. Finally, the 108 genes found in common between the RNA-seq and protein data were kept, as well as the 62 mutations that were present in at least two donors. Note that among the 62 mutations, 58 are present on genes that are not concerned by the RNA-seq and protein expression measurements. In the end, the final dataset contained 278 variables (108 protein expression, 108 RNA-seq counts, 62 mutations) observed on 250 individuals. Note that our RNA-seq counts have a median value of 3706.

The precision matrix Ω was inferred via the graphical lasso applied to the estimate of the copula correlation matrix obtained from pseudo pairwise likelihood estimation. Estimates $\hat{\Omega}_\lambda$ of Ω were computed for each value of λ in a grid with values between zero and one. As the dataset comprise count data with a large number of distinct values, and large values of counts, the method of bridge functions was too computationally expensive and could not be applied.

Figure 6 shows the proportion of kept edges per variable type depending on the values of the penalization parameter λ. We can see that the proportion of detected edges decreases faster for RNA-protein and mutation-mutation pairs, while the other curves behave similarly.

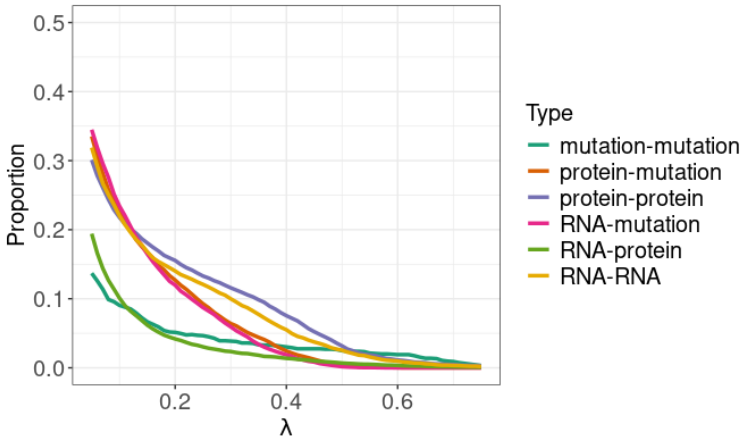

Fig. 6: Proportion of detected edges for each pair of variable types as a function of the penalization parameter λ.

The HBIC criterion was used to choose the penalization parameter, and an optimal value of $\lambda = 0.53$ was found, as illustrated in Figure 7. The conditional correlation graph obtained for this optimal value is given in Figure 8. Table 4 shows the number of edges per variable type. As shown in Figure 8 and Table 4, the larger number of links identified in the network is within proteins for one major cluster, and within RNA-seq counts for a second major cluster. Note that both clusters are strongly connected, with 64 links between them. A fewer number of links are identified within mutations (45), and only 9 (resp. 4) links between mutations and proteins (resp. genes).

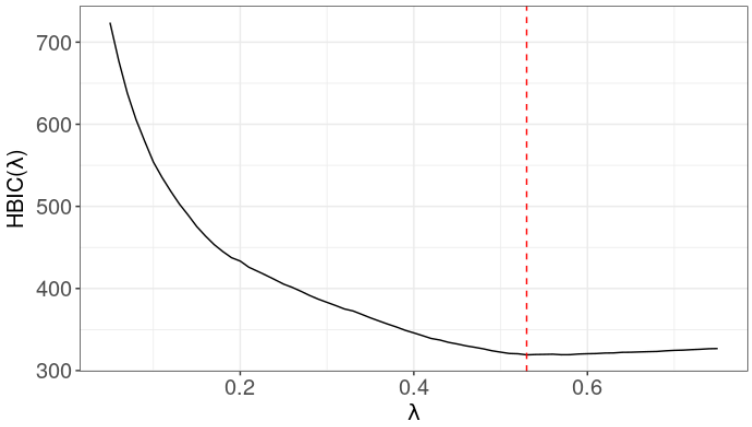

Fig. 7: Values of HBIC(λ), $0.05 < \lambda \leq 0.75$.

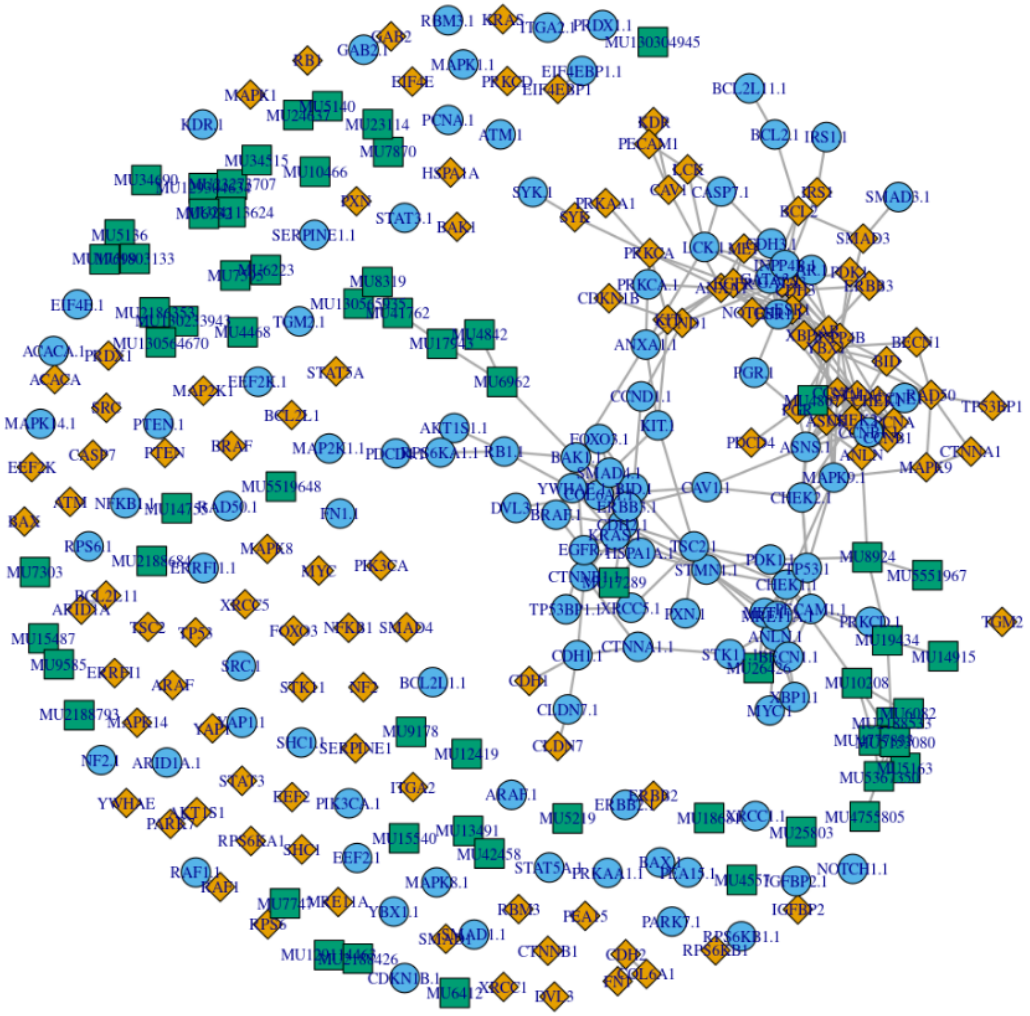

Fig. 8: Estimated network between the RNA-seq counts (yellow diamonds), proteins (blue circles) and mutations (green squares) with the optimal value of $\lambda = 0.53$ returned by the HBIC criterion.

Variable type	Number of edges
protein-protein	125
RNA-RNA	106
RNA-protein	64
mutation-mutation	45
protein-mutation	9
RNA-mutation	4
Total	353

Table 4: Number of edges for the optimal lambda.

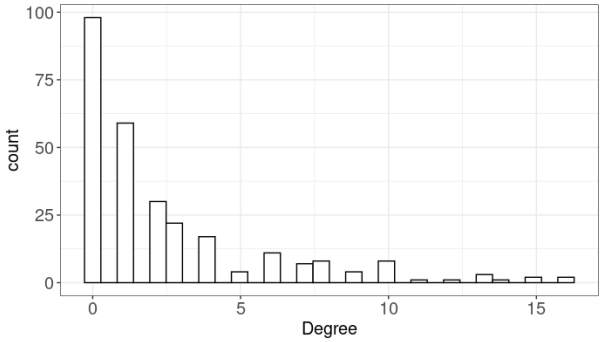

Fig. 9: Histogram of the degrees of the nodes in the selected network.

Figure 9 shows a histogram of the degrees of the nodes in the obtained graph. Only ten of them have over ten connected edges. These nodes correspond to the genes and proteins given in Table 5. It is interesting to note that all of them have been found to be related to breast cancer. The scientific references presenting these results are given in Table 5. Furthermore, for several of these nodes, we identified both the gene and the related protein. It is the case for GATA3 and ESR1. Figure 10 presents the subgraph corresponding to the neighborhoods of genes and proteins from Table 5. Overall, the nodes with the highest number of edges are linked to each other in two main hubs: one consisting of proteins, the other one of mostly RNA-seq counts. Two mutations are included in the graph: MU4807 and MU17289. They are related to genes TP53 and CDKN1B, which are known to have an influence on breast cancer [19, 20]. One of the scientific articles [21] also showed a link between gene CCNE1 and gene ANLN. This link was successfully identified with our network inference procedure as shown in Figure 10.

Variable	Type	degree	Reference
CCNE1	RNA-seq	16	[22]
GATA3	RNA-seq	16	[23]
ESR1	RNA-seq	16	[24]
INPP4B	RNA-seq	15	[25]
YBX1	RNA-seq	14	[26]
ANXA1	RNA-seq	13	[27]
ESR1	protein	13	[24]
GATA3	protein	13	[23]
STMN1	RNA-seq	12	[28]
CDH2	protein	11	[29]

Table 5: Genes and proteins with degree larger than ten.

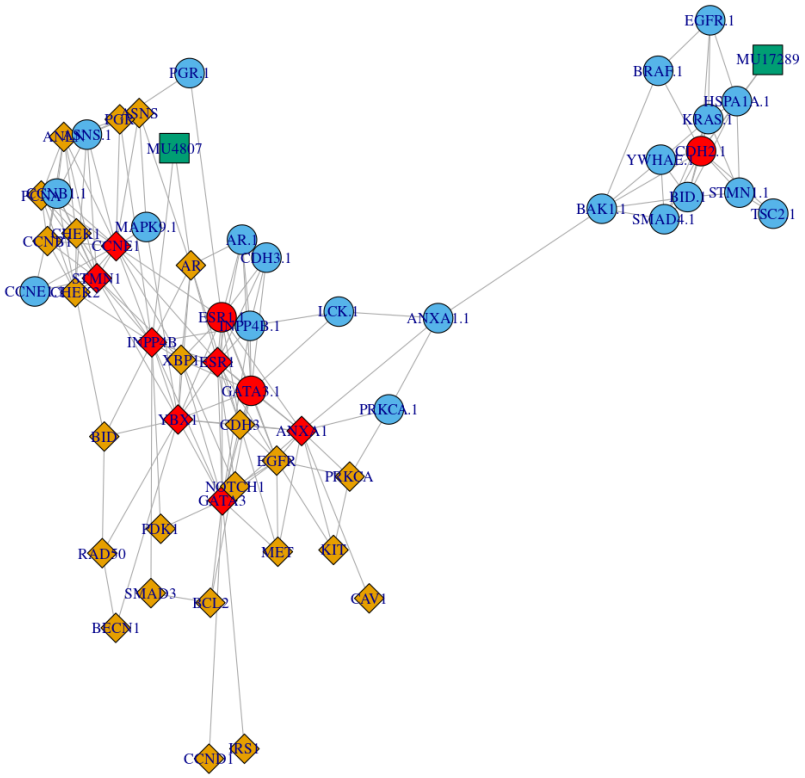

Fig. 10: Subgraph corresponding to the neighborhoods of genes (resp. proteins) from Table 5, represented as red diamonds (resp. circles). The mutations are represented as green squares, the other genes as yellow diamonds and the other proteins as blue circles.

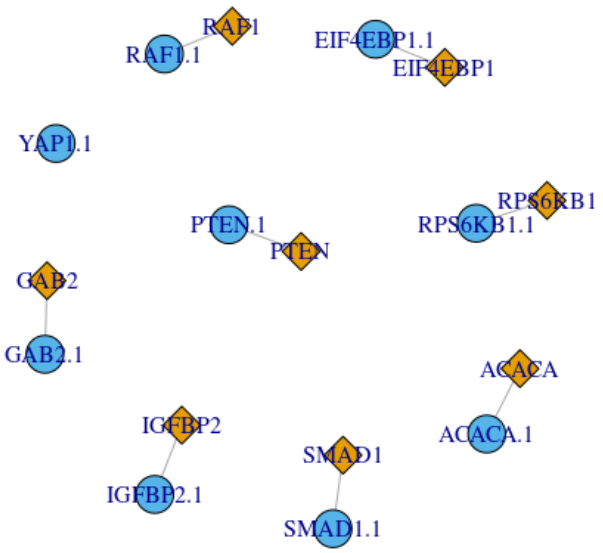

Fig. 11: Pairs of proteins (blue circles) and RNA-seq (yellow diamonds) not linked to any other node in the main graph presented in Figure 8.

Additionally, among the nodes outside the main clusters of the network in Figure 8, a few independent pairs of proteins and genes have been identified and are presented in Figure 11. It is interesting to notice that each of these pairs of RNA-seq and protein corresponds to a single gene.

5 Discussion

In this study, we proposed a Gaussian copula model for multi-omics network inference, with a pairwise-likelihood estimation and graphical lasso regularization. In an extensive simulation study, we showed that the proposed pairwise-likelihood estimation method proved to be more effective as compared to moment-based methods using bridge functions. Its computational time is not impacted by the analysis of discrete variables with a large number of categories. It performs well in terms of AUC even in the case of large discrete values, which are poorly handled by the bridge functions estimation.

The above results have shown relevance of PPMLE estimation as opposed to bridge functions when inferring biological networks that involve, for instance, RNA-seq count data which often have large values and many categories. When applied to ICGC breast cancer data, our method has successfully identified biologically relevant interactions. For instance, it recovered direct links between proteins and RNA-seq counts belonging to a same gene. It also highlighted hubs of biologically important variables linked to breast cancer.

Several challenges will be worthwhile investigating in future work. The selection of the optimal regularization parameter remains crucial for balancing sparsity and accuracy in network estimation. We proposed in this study to use the HBIC selection criterion, but further exploration of adaptive penalization techniques may enhance model selection.

In order to improve the biological interpretation of the results, it would be interesting to further investigate the link between the precision matrix of the copula and the conditional independence relationships of the observed variables.

Overall, our approach provides a computationally efficient and robust tool for network inference in high-dimensional multi-omics data, enriching the toolkit for genomic data analysis.

6 Acknowledgements

This work was supported by a public grant from the Fondation Mathématique Jacques Hadamard.

References

[1] Friedman, J., Hastie, T., Tibshirani, R.: Sparse inverse covariance estimation with the graphical lasso. Biostatistics **9**(3), 432–441 (2008) https://doi.org/10.1093/biostatistics/kxm045

[2] Liu, H., Lafferty, J., Wasserman, L.: The nonparanormal: Semiparametric estimation of high dimensional undirected graphs. Journal of Machine Learning Research **10**, 2295–2328 (2009)

[3] Chiquet, J., Mariadassou, M., Robin, S.: Variational inference for sparse network reconstruction from count data. Proceedings of the 36th International Conference on Machine Learning **97**, 1162–1171 (2019)

[4] Hawe, J.S., Theis, F.J., Heinig, M.: Inferring interaction networks from multi-omics data. Frontiers in Genetics **10**(535) (2019) https://doi.org/10.3389/fgene.2019.00535

[5] Lee, J.D., Hastie, T.J.: Learning the structure of mixed graphical models. Journal of Computational and Graphical Statistics **24**(1), 230–253 (2015) https://doi.org/10.1080/10618600.2014.900500

[6] Sedgewick, A.J., Shi, I., Donovan, R.M., Benos, P.V.: Learning mixed graphical models with separate sparsity parameters and stability-based model selection. BMC Bioinformatics **17**(S175) (2016) https://doi.org/10.1186/s12859-016-1039-0

[7] Dobra, A., Lenkoski, A.: Copula Gaussian graphical models and their application to modeling functional disability data. The Annals of Applied Statistics **5**, 969–993 (2011) https://doi.org/10.1214/10-AOAS397

[8] Mohammadi, A., Abegaz, F., Heuvel, E.V.D., Wit, E.C.: Bayesian modelling of Dupuytren disease by using Gaussian copula graphical models. Applied Statistics **66**, 629–645 (2017) https://doi.org/10.1111/rssc.12171

[9] Zhang, A., Fang, J., Hu, W., Calhoun, V., Wang, Y.: A latent gaussian copula model for mixed data analysis in brain imaging genetics. IEEE/ACM transactions on computational biology and bioinformatics **18**(4), 1350–1360 (2021) https://doi.org/10.1109/TCBB.2019.2950904

[10] Zhang, J., Bajari, R., Andric, D.: The International Cancer Genome Consortium Data Portal (2019). https://doi.org/10.1038/s41587-019-0055-9

[11] Lauritzen, S.: Graphical Models. Oxford Science Publications, Oxford (1996)

[12] Mazo, G., Karlis, D., Rau, A.: A randomized pairwise likelihood method for complex statistical inferences. Journal of the American Statistical Association **119**(547), 2317–2327 (2024) https://doi.org/10.1080/01621459.2023.2257367

[13] Tomilina, E., Jaffrézic, F., Mazo, G.: Gaussian copula correlation network analysis with application to multi-omics data. working paper or preprint (2025). https://hal.inrae.fr/hal-04847648

[14] Liu, H., Han, F., Yuan, M., Lafferty, J., Wasserman, L.: High-dimensional semiparametric gaussian copula graphical models. The Annals of Statistics **40**(4), 2293–2326 (2012)

[15] Fan, J., Liu, H., Ning, Y., Zou, H.: High dimensional semiparametric latent graphical model for mixed data. Journal of the Royal Statistical Society Series B: Statistical Methodology **79**(2), 405–421 (2016) https://doi.org/10.1111/rssb.12168

[16] Watts, D., Strogatz, S.: Collective dynamics of small-world networks. Nature **393**, 440–442 (1998) https://doi.org/10.1038/30918

[17] Tomilina, E., Cartier, J., Jaffrézic, F., Mazo, G.: Heterocop: Semi-Parametric Estimation with Gaussian Copula. (2025). R package version 1.0.0. https://CRAN.R-project.org/package=heterocop

[18] Love, M.I., Huber, W., Anders, S.: Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. Genome Biology **15**(550) (2014) https://doi.org/10.1186/s13059-014-0550-8

[19] Shahbandi, A., Nguyen, H.D., Jackson, J.G.: TP53 mutations and outcomes in breast cancer: Reading beyond the headlines. Trends in Cancer **6**(2), 98–110 (2020) https://doi.org/10.1016/j.trecan.2020.01.007

[20] Cusan, M., Mungo, G., De Marco Zompit, M., Segatto, I., Belletti, B., Baldassarre, G.: Landscape of CDKN1B mutations in luminal breast cancer and other hormone-driven human tumors. Frontiers in Endocrinology **9**(393) (2018) https://doi.org/10.3389/fendo.2018.00393

[21] Dai, S., Li, L., Guo, G., Peng, Y., Yuan, H., Li, J.: CCNE1 stabilizes ANLN by counteracting FZR1-mediated the ubiquitination modification to promotes triple negative breast cancer cell stemness and progression. Cell Death Discovery **11**(228) (2025) https://doi.org/10.1038/s41420-025-02518-5

[22] Marra, A., Selenica, P., Zhu, Y., Safonov, A., Razavi, P., Roulston, A., Koehler, M., Curigliano, G., Ross, D.S., Weigelt, B., Reis-Filho, J., Schram, A.M., Chandarlapaty, S., Rosen, E.: CCNE1 amplification as marker of poor prognosis and novel therapeutic target in advanced breast cancer. Journal of Clinical Oncology **42**(16), 1040–1040 (2024) https://doi.org/10.1200/JCO.2024.42.16_suppl.1040

[23] Takaku, M., Grimm, S.A., Wade, P.A.: GATA3 in breast cancer: Tumor suppressor or oncogene? Gene Expression **16**(4), 163–168 (2015) https://doi.org/10.3727/105221615X14399878166113

[24] Brett, J.O., Spring, L.M., Bardia, A., Wander, S.A.: ESR1 mutation as an emerging clinical biomarker in metastatic hormone receptor-positive breast cancer. Breast Cancer Research **23**(85) (2021) https://doi.org/10.1186/s13058-021-01462-3

[25] Rodgers, S.J., Ooms, L.M., Oorschot, V.M.J., Schittenhelm, R.B., Nguyen, E.V., Hamila, S.A., Rynkiewicz, N., Gurung, R., Eramo, M.J., Sriratana, A., Fedele, C.G., Caramia, F., Loi, S., Kerr, G., Abud, H.E., Ramm, G., Papa, A., Ellisdon, A.M., Daly, R.J., McLean, C.A., Mitchell, C.A.: INPP4B promotes pi3kα-dependent late endosome formation and wnt/β-catenin signaling in breast cancer. Nature Communications **12**(1), 3140 (2021) https://doi.org/10.1038/s41467-021-23241-6

[26] Shibata, T., Tokunaga, E., Hattori, S., Watari, K., Murakami, Y., Yamashita, N., Oki, E., Itou, J., Toi, M., Maehara, Y., Kuwano, M., Ono, M.: Y-box binding protein YBX1 and its correlated genes as biomarkers for poor outcomes in patients with breast cancer. Oncotarget **9**(98), 37216–37228 (2018) https://doi.org/10.18632/oncotarget.26469

[27] Al-Ali, H.N., Crichton, S.J., Fabian, C., Pepper, C., Butcher, D.R., Dempsey, C.F., Parris, C.N.: A therapeutic antibody targeting annexin-A1 inhibits cancer cell growth in vitro and in vivo. Oncogene **43**, 608–614 (2024) https://doi.org/10.1038/s41388-023-02919-9

[28] Ruiqi, L., Xiaodong, L., Haiwei, G., Shuang, L., Weiping, Y., Chenfang, D., Jiajun, W., Yanwei, L., Jianming, T., Haibo, Z.: STNM1 in human cancers: role, function and potential therapy sensitizer. Cellular Signalling **109**, 110775 (2023)

https://doi.org/10.1016/j.cellsig.2023.110775

[29] Sinha, G., Ferrer, A.I., Ayer, S., El-Far, M.H., Pamarthi, S.H., Naaldijk, Y., Barak, P., Sandiford, O.A., Bibber, B.M., Yehia, G., Greco, S.J., Jiang, J.G., Bryan, M., Kumar, R., Ponzio, N.M., Etchegaray, J., Rameshwar, P.: Specific N-cadherin-dependent pathways drive human breast cancer dormancy in bone marrow. Life Science Alliance **4**(7) (2021) https://doi.org/10.26508/lsa.202000969

[30] Csárdi, G., Nepusz, T., Traag, V., Horvát, S., Zanini, F., Noom, D., Müller, K.: igraph: Network Analysis and Visualization in R. (2025). https://doi.org/10.5281/zenodo.7682609 . R package version 2.1.4. https://CRAN.R-project.org/package=igraph

[31] Kruskal, W.H.: Ordinal measures of association. Journal of the American Statistical Association **53**(284), 814–861 (1958) https://doi.org/10.1080/01621459.1958.10501481

Appendix A Simulation of the precision matrix

First, a matrix $\tilde{\Omega}$ is simulated via Algorithm 1. Note that the Watts-Storgatz approach is implemented in the `igraph` package [30], and that the last step is necessary to ensure positive definiteness of $\tilde{\Omega}$.

Algorithm 1 Simulation of $\tilde{\Omega}$ for p variables (Zhang et al. [9])

Require: $p > 0$ the number of variables, $r > 0$ the number of non-null coefficients
Define a $p \times p$ matrix of zeroes $\tilde{\Omega}$
Randomly assign $r/2$ non-null coefficients by a Watts-Storgatz approach [16] in the upper triangular part of $\tilde{\Omega}$
Fill these coefficients with $\mathcal{U}(0.2, 0.7)$ values
$\tilde{\Omega} \leftarrow \tilde{\Omega} + \tilde{\Omega}^T$
$\text{diag}(\tilde{\Omega}) \leftarrow 1$
Compute $\lambda_m = \min(Sp(\tilde{\Omega}))$
$\text{diag}(\tilde{\Omega}) \leftarrow |\lambda_m| + 0.01$

Then, compute $\tilde{\Sigma} = \tilde{\Omega}^{-1}$. Because we suppose model (1) to be parametrized by a correlation matrix, standardize it to obtain $\Sigma = \Lambda^{-\frac{1}{2}}\tilde{\Sigma}\Lambda^{-\frac{1}{2}}$ where $\Lambda = \text{diag}(\tilde{\Sigma})$. Finally, compute our matrix of interest $\Omega = \Sigma^{-1} = \Lambda^{\frac{1}{2}}\tilde{\Omega}\Lambda^{\frac{1}{2}}$.

Appendix B Details of inference methods

B.1 Estimation based on bridge functions

A first frequentist approach in order to estimate Ω has been given by Liu et al. in the nonparanormal SKEPTIC [14] and consists in the following steps:

1. Estimate Kendall's τ on the observed variables
2. Find the root of the corresponding bridge function in order to obtain the covariance matrix Σ between the latent Gaussian variables
3. Invert Σ in order to obtain Ω as described in section 2.2

Note that the difference with our estimation method comes from the additional step while we directly estimate Σ on the observed variables by PPMLE before inversion.

A theoretical expression for Kendall's tau between two variables X_j and X_k is given by:

$$\tau_{jk} = \text{Corr}\left(\text{sign}(X_j - \tilde{X}_j)\text{sign}(X_k - \tilde{X}_k)\right) \tag{B1}$$

where $\tilde{X}_j$ and $\tilde{X}_k$ denote two independent copies of X_j and X_k. It can be estimated by

$$\hat{\tau}_{jk} = \frac{2}{n(n-1)} \sum_{1 \leq i < i' \leq n} \text{sign}(x_j^i - x_j^{i'})\text{sign}(x_k^i - x_k^{i'}) \tag{B2}$$

Moreover, when both variables are continuous, we have the following result [31]:

$$\Sigma_{jk} = 2\sin\left(\frac{\pi}{2}\tau_{jk}\right) \tag{B3}$$

and it is sufficient to plug the estimator from equation (B2) into equation (B3) in order to estimate $\hat{\Sigma}_{jk}$.

If at least one of the variables is binary, the estimator from equation (B2) holds because $\text{sign}(x_j^i - x_j^{i'}) = x_j^i - x_j^{i'}$. However, the bridge function becomes [15]:

$$\begin{aligned} \tau_{jk} &= F(\Sigma_{jk}, \Delta_j) \\ &= 4(\Phi_{\Sigma_{jk}/\sqrt{2}}(\Delta_j, 0)) - 2\Phi(\Delta_j) \end{aligned} \tag{B4}$$

where Δ_j denotes the cutoff value for a latent variable Z_j behind the binary variable X_j. In practice, one can estimate Δ_j by $\hat{\Delta}_j = \Phi^{-1}(1 - \bar{X}_j)$ and $\hat{\Sigma}_{jk}$ by solving $F(t, \hat{\Delta}_j) - \hat{\tau}_{jk} = 0$. Similarly, in the case where both variables are binary, the bridge function becomes

$$\begin{aligned} \tau_{jk} &= F(\Sigma_{jk}, \Delta_j, \Delta_k) \\ &= 2\Phi_{\Sigma_{jk}}(\Delta_j, \Delta_k) - 2\Phi(\Delta_j)\Phi(\Delta_k) \end{aligned} \tag{B5}$$

Finally, these expressions have been extended to the case of discrete variables with more than two modalities [9]. In place of Kendall's τ, a new rank statistic r_{jk} is introduced for the discrete-discrete and mixed cases in order to facilitate computation of the bridge functions. When both variables are multinomial, we compute $\hat{r}_{jk}$ on the observed data as below:

$$\hat{r}_{jk} = \frac{2}{n(n-1)} \sum_{1 \leq i < i' \leq n} (x_j^i - x_j^{i'})(x_k^i - x_k^{i'}) \tag{B6}$$

We also get the associated bridge function

$$\begin{aligned} r_{jk} &= \mathbb{E}(\hat{r}_{jk}) \\ &= F(\Sigma_{jk}, \Delta_j, \Delta_k) \\ &= 2\{\sum_{l=1}^{L}\sum_{m=1}^{L}\Phi_{\Sigma_{jk}}(\Delta_{jl}, \Delta_{km}) - \sum_{l=1}^{L}\Phi(\Delta_{jl})\sum_{m=1}^{L}\Phi(\Delta_{km})\} \end{aligned} \tag{B7}$$

When X_j is discrete and X_k is continuous, we compute:

$$\hat{r}_{jk} = \frac{2}{n(n-1)}\sum_{1\leq i<i'\leq n}(x_j^i - x_j^{i'})\text{sign}(x_k^i - x_k^{i'}) \tag{B8}$$

The associated bridge function is given by:

$$\begin{aligned} r_{jk} &= \mathbb{E}(\hat{r}_{jk}) \\ &= F(\Sigma_{jk}, \Delta_j, \Delta_k) \\ &= 2\{\sum_{l=1}^{L}\Phi_{\Sigma_{jk}/\sqrt{2}}(\Delta_{jl}, 0) - 2\sum_{l=1}^{L}\Phi(\Delta_{jl})\} \end{aligned} \tag{B9}$$

The main limit of this method is the lack of roots for the bridge functions when the sample size n is not large enough, and the needed adaptation of the code as in https://github.com/Aiying0512/LGCM. Also, the case when a binary variable takes the same modality for all observations is not taken into consideration.

B.2 Pairwise densities

The densities $\hat{f}_{jj'}$ introduced in section 2.2 take the form below depending on the case. For easier notation, let $\hat{F}_j(x_j) = u$, $\hat{F}_{j'}(x_{j'}) = v$, $\hat{F}_j(x_j-) = u-$ and $\hat{F}_{j'}(x_{j'}-) = v-$, where x_j- and $x_{j'}-$ denote the points before x_j and $x_{j'}$ in the supports of $\hat{F}_j$ and $\hat{F}_{j'}$. If both variables are continuous, we have

$$\hat{f}_{jj'}(x_j, x_{j'}, \Sigma_{jj'}) = \frac{1}{\sqrt{2\pi(1-\Sigma_{jj'}^2)}}\exp\left(-\frac{\Phi^{-1}(u)^2 + \Phi^{-1}(v)^2 - 2\Sigma_{jj'}\Phi^{-1}(u)\Phi^{-1}(v)}{2(1-\Sigma_{jj'}^2)}\right)$$

If X_j is continuous and $X_{j'}$ is discrete, we have

$$\hat{f}_{jj'}(x_j, x_{j'}, \Sigma_{jj'}) = \Phi\left(\frac{\Phi^{-1}(v) - \Sigma_{jj'}\Phi^{-1}(u)}{1-\Sigma_{jj'}^2}\right) - \Phi\left(\frac{\Phi^{-1}(v-) - \Sigma_{jj'}\Phi^{-1}(u)}{1-\Sigma_{jj'}^2}\right)$$

If both variables are discrete, we have

$$\hat{f}_{jj'}(x_j, x_{j'}, \Sigma_{jj'}) = \int_{u-}^{u} \Phi\left(\frac{\Phi^{-1}(v) - \Sigma_{jj'}\Phi^{-1}(z)}{1-\Sigma_{jj'}^2}\right) - \Phi\left(\frac{\Phi^{-1}(v-) - \Sigma_{jj'}\Phi^{-1}(z)}{1-\Sigma_{jj'}^2}\right) dz$$

Appendix C Graphical Lasso

Suppose that $\mathbf{Z} = (Z_1, ..., Z_d) \sim \mathcal{N}(0, \Sigma)$ is a centered standardized Gaussian vector of correlation matrix Σ. Its density is expressed as, for $\mathbf{z} = (z_1, \ldots, z_d)$:

$$f(\mathbf{z}) = \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} \exp\left(-\frac{\mathbf{z}^T \Sigma^{-1} \mathbf{z}}{2}\right).$$

Let $\hat{\Sigma}^P$ denote the estimator of the Pearson correlation matrix with elements:

$$(\hat{\Sigma}^P)_{jk} = \frac{1}{n}\sum_{i=1}^{n} Z_j^i Z_k^i.$$

A natural log-MLE estimator of $\Omega = \Sigma^{-1}$, for n independent realizations $Z_1^i, \ldots, Z_d^i$, $i = 1, \ldots, n$ is:

$$\begin{aligned}
\hat{\Omega} &= \underset{\Omega}{\text{argmax}} \frac{1}{n}\sum_{i=1}^{n} \log\left(\frac{1}{\sqrt{(2\pi)^d |\Omega^{-1}|}} \exp\left(-\frac{(Z_1^i, \ldots, Z_d^i)^T \Omega (Z_1^i, \ldots, Z_d^i)}{2}\right)\right) \\
&= \underset{\Omega}{\text{argmax}} \left(\frac{1}{n}\sum_{i=1}^{n} \log|\Omega| - \frac{1}{2}(z_1^i, \ldots, z_d^i)^T \Omega (z_1^i, \ldots, z_d^i)\right) \\
&= \underset{\Omega}{\text{argmax}} \left(\frac{1}{n}\sum_{i=1}^{n} \log|\Omega| - \sum_{j=1}^{d}\sum_{k=1}^{d} \frac{z_j^i z_k^i \Omega_{jk}}{2}\right) \\
&= \underset{\Omega}{\text{argmax}} \left(\log|\Omega| - \sum_{j=1}^{d}\sum_{k=1}^{d} \Omega_{jk} \frac{1}{n}\sum_{i=1}^{n} z_j^i z_k^i\right) \\
&= \underset{\Omega}{\text{argmax}} \left(\log|\Omega| - \sum_{j=1}^{d}\sum_{k=1}^{d} \Omega_{jk} \hat{\Sigma}_{kj}^P\right) \\
&= \underset{\Omega}{\text{argmax}} \left(\log|\Omega| - tr(\Omega \hat{\Sigma}^P)\right).
\end{aligned}$$

Equation (4) is then obtained by penalizing the above likelihood.