NP-MuScL: Unsupervised global prediction of interaction networks from multiple data sources

Kriti Puniyani and Eric P. Xing

School of Computer Science, Carnegie Mellon University
{kpuniyan,epxing}@cs.cmu.edu

Abstract. Inference of gene interaction networks from expression data usually focuses on either supervised or unsupervised edge prediction from a single data source. However, in many real world applications, multiple data sources, such as microarray and ISH measurements of mRNA abundances, are available to offer multi-view information about the same set of genes. We propose NP-MuScL (nonparanormal multi-source learning) to estimate a gene interaction network that is consistent with such multiple data sources, which are expected to reflect the same underlying relationships between the genes. NP-MuScL casts the network estimation problem as estimating the structure of a sparse undirected graphical model. We use the semiparametric Gaussian copula to model the distribution of the different data sources, with the different copulas sharing the same precision (i.e., inverse covariance) matrix, and we present an efficient algorithm to estimate such a model in the high dimensional scenario. Results are reported on synthetic data, where NP-MuScL outperforms baseline algorithms significantly, even in the presence of noisy data sources. Experiments are also run on two real-world scenarios: two yeast microarray data sets, and three Drosophila embryonic gene expression data sets, where NP-MuScL predicts a higher number of known gene interactions than existing techniques.

Keywords: interaction networks, gene expression, multi-source learning, sparsity, Gaussian graphical models, nonparanormal, copula

1 Introduction

With the prevalence of high throughput technologies such as microarray and RNA-seq for measuring gene expressions, computational inference of gene regulatory or interaction networks from large-scale gene expression datasets has emerged as a popular technique to improve our understanding of cellular systems [1,2,3]. In numerous studies, gene interactions reverse engineered from analysis of such high-throughput data have been experimentally validated [4,5], demonstrating the credibility of such data-driven algorithmic approaches.

There have been two popular approaches to reverse engineering gene networks. The first approach is to build a generative model of the data, and learn a graphical model that captures the conditional independencies in the data. Learning the structure of a graphical model under a multivariate Gaussian assumption

of the data has received wide attention in recent years [6,7,8]; various algorithms have been proposed [6,7,8], many with theoretical analysis offering asymptotic guarantee of consistent estimation of the interactions between genes in the network. Empirically, these algorithms are computationally efficient and the results obtained have been encouraging.

However, a limitation of this class of network inference approach is that, it assumes data are identically and independently distributed (i.e., *iid*), which implicitly means that they are from a single experimental source. In reality, many real world biological problems sit on multiple sources of information that can be used to predict interactions between genes. For example, there can be multiple microarray data sets from different laboratories available for the same organism, sometimes measured at the same conditions where the main differences lie in the data sampling strategy or measurement technologies. Biologically, it is often plausible to assume that multiple experimental means resulting in the different datasets may have captured the same information from different viewpoints, e.g., both microarray and *in-situ* hybridization can capture gene expression information, even though the technology used to measure mRNA abundances is different. It remains unclear how to integrate such multiple sources of data in a statistically valid and computational efficient way to infer the underlying network. One may imagine inferring independently a network from each data source, and then averaging across multiple resultant networks, but such an *ad hoc* method is not only un-robust (e.g., each view may have only a small amount of samples), but also lacks statistically justification and consistence guarantee (e.g., on the "average" operator). In this paper, we address the question of inferring a network by analyzing multiple sources of information simultaneously.

An alternative approach to tackle this problem is via supervised learning methods, where a classifier (e.g., SVM) is trained by using examples of known gene interactions (edges in the network) as training data to learn the importance of each data source in predicting unknown interactions between other gene-pairs [9]. This approach suffers from some intrinsic limitations which prevent it from being widely applicable. First, while such an approach works well for problems where there are sufficient examples of known edges in the network, e.g., in the form of a *reference network* or *reference interactions* obtained from reliable sources, it fails for problems where few or no examples of known edges are available. Gene networks for humans or yeast may be learned by supervised methods where reference interactions are available from extensive prior studies; but for organisms where prior research is limited, this approach cannot be used. Furthermore, one can argue that predicting gene networks is of high importance for such organisms with few known edges, to help biologists who are starting research for regulatory mechanisms of these organisms.

Secondly, using a classifier to predict edges implicitly utilizes the notion of *marginal independence* between nodes. To classify an edge as "positive", i.e., to predict an edge between a given pair of nodes, the correlation between the data for these nodes must be high. Gene networks usually have pathways in which genes interact with each other in a sequential order, which results in high

marginal correlation between all pairs of genes in the same pathway. Predicting each edge locally and independently of all other edges will often result in an non-stringent prediction of a clique for all genes in the same pathway, leading to high false positive rates. To reduce such false positives and increase accuracy, we wish to analyze *conditional independence* between the genes instead, which must be done by building a global graphical model that captures simultaneously all the conditional independencies among genes. Each edge resultant from such an estimator enjoys *global* statistical interpretability and consistency guarantee, and such an estimator does not require supervised training, although prior knowledge of interactions on the "reference gene pairs" can still be utilized via introducing a prior over the model, if desired. Thus, it is desirable to develop an *unsupervised* and global inference method which can incorporate multiple data sources to predict a consensus graphical model that explains all the data sources, without using any examples of known edges for training the model.

This paper proposes NP-MuScL (NonParanormal Multi-Source Learning), a machine learning technique for estimating the structure of a sparse undirected graphical model that is consistent with multiple sources of data. The multiple data sources are all defined over the same feature space, and it is assumed that they share the same underlying relationships between the genes (nodes). We use the semiparametric Gaussian copula to model the distribution of the different data sources, where the copula for each data source has its own mean and transformation functions, but all data sources share the same precision matrix (i.e., the inverse covariance matrix, which captures the topological structure of the network). We propose an efficient algorithm to estimate such a model in the high dimensional scenario. The likelihood-related objective function used in NP-MuScL is convex, and results in a globally optimal estimator. Furthermore, the implementation of our algorithm is simple and efficient, computing a network over 2000 nodes using 3 data sources in a matter of minutes. Results are reported on synthetic data, where NP-MuScL outperforms baseline algorithms significantly, even in the presence of noisy data sources. We also use NP-MuScL to estimate a gene network for yeast using two microarray data sets: one over time series expression, and the other over knockout mutants. Finally, we run NP-MuScL on three data sets of Drosophila embryonic gene expression using ISH images and microarray. In both yeast and Drosophila, we find that NP-MuScL predicts a higher number of gene interactions that are known to interact in the literature, than existing techniques.

1.1 Related Work

Previous work on analyzing multiple data sources for network prediction has either specifically taken time into account[10,11], or has different source and target organisms via transfer learning [12]. Katenka et. al.[13] propose a strategy to learn a network from multi-attribute data, where aligned vector observations are made for each node. The NP-MuScL algorithm on the other hand works for data sources which are not aligned, hence each data source may have a different number of observations. Honorio et. al. [14] proposed techniques for multi-task

structure learning of Gaussian Graphical Models, to share knowledge across multiple problems, using multi-task learning. However, their method estimates a separate graphical model for each data source, unlike our problem which requires a consensus network common to all data sources. To the best of our knowedge, the NP-MuScL algorithm is the first work that builds a consensus graphical model to explain the relationship between genes by combining information from multiple data sources without explicitly constraining the data to be time-series, or about different organisms.

2 Nonparanormal Multi-Source Learning (NP-MuScL)

Let the k input data sources be defined as $\mathbf{X}^{(1)} \in \mathbb{R}^{n_1 \times d}$, $\mathbf{X}^{(2)} \in \mathbb{R}^{n_2 \times d}$, $\ldots$, $\mathbf{X}^{(k)} \in \mathbb{R}^{n_k \times d}$ with total number of data samples $n = \sum_{i=1}^{k} n_i$. Each data source i may have a different number of measurements or samples n_i, but they all measure information about the same feature space of d genes. The goal of NP-MuScL is to learn the structure of a graphical model over the feature space, such that the graphical model will encapsulate global conditional independencies between the genes.

2.1 Glasso

Given a single source of data $\mathbf{X} \in \mathbb{R}^{n \times d}$ drawn from a Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$, a Gaussian graphical model (GGM) may be estimated by computing the inverse covariance matrix $\mathbf{\Sigma}^{-1}$ of the Gaussian. Zeros in the inverse covariance matrix imply conditional independence between the features, and thus the absence of an edge between them in the corresponding GGM. Given the empirical covariance matrix $\mathbf{S}$ of the data, the inverse covariance matrix may be computed by maximizing the log likelihood of the data, with an L_1 regularizer to encourage sparsity.

$$\hat{\mathbf{\Sigma}}^{-1} = \arg\max_{\mathbf{\Theta} \succ 0} \{\log\det \mathbf{\Theta} - \mathrm{tr}(\mathbf{S\Theta}) - \lambda ||\mathbf{\Theta}||_1\} \tag{1}$$

where λ is a tuning parameter that controls the sparsity of the solution; as λ increases, fewer edges are predicted in the GGM. Rothman et. al. [15] showed the consistence of such estimators in Frobenius and Operator norms in high dimensions when $d >> n$; Friedman et. al.[8] proposed a block coordinate descent algorithm for this objective - they named their technique glasso. The glasso algorithm uses a series of L_1 penalized regressions, called Lasso regressions [16], that can be solved in time $O(d^3)$.

2.2 Joint estimation of the GGM

Given k data sources $\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \cdots \mathbf{X}^{(k)}$ with corresponding sample covariances $\mathbf{S}^{(1)}, \mathbf{S}^{(2)}, \cdots, \mathbf{S}^{(k)}$, a joint estimator of the underlying GGM may be computed as

$$\hat{\mathbf{\Sigma}}^{-1} = \arg\max_{\mathbf{\Theta} \succ 0} \sum_{i=1}^{k} w_i \left\{ \log\det \mathbf{\Theta} - \text{tr}(\mathbf{S}^{(i)}\mathbf{\Theta}) \right\} - \lambda ||\mathbf{\Theta}||_1 \tag{2}$$

where w_i defines the relative importance of each data source, and must be defined by the user such that $\sum_{i=1}^{k} w_i = 1$. Assuming the data in each data source is drawn i.i.d., an appropriate choice for the weights may be $w_i = \frac{n_i}{n}$. It can be seen that if each data source is assumed to have mean 0, then for this choice of w_i

$$\begin{aligned}
\hat{\mathbf{\Sigma}}^{-1} &= \arg\max_{\mathbf{\Theta} \succ 0} \log\det \mathbf{\Theta} - \sum_{i=1}^{k} \frac{n_i}{n} \text{tr}\left(\mathbf{S}^{(i)}\mathbf{\Theta}\right) - \lambda ||\mathbf{\Theta}||_1 \\
&= \arg\max_{\mathbf{\Theta} \succ 0} \log\det \mathbf{\Theta} - \text{tr}\left(\frac{1}{n} \sum_{i=1}^{k} \sum_{l=1}^{n_i} \mathbf{X}^{(i)}(l,\cdot)^T \mathbf{X}^{(i)}(l,\cdot)\ \mathbf{\Theta} \right) - \lambda ||\mathbf{\Theta}||_1
\end{aligned} \tag{3}$$

Thus, our objective function is equivalent to calling glasso with covariance matrix $\frac{1}{n}\sum_{i=1}^{k}\sum_{l=1}^{n_i} \mathbf{X}^{(i)}(l,\cdot)^T \mathbf{X}^{(i)}(l,\cdot)$. We call this method "glasso-bag of data". With an appropriate choice of weights, this model concatenates the data from all data sources into a single matrix, and uses the second moment of the data to estimate the inverse covariance matrix.

Such a procedure highlights the underlying assumption of Gaussianity of the data. If we assume that all data is being drawn from the same Gaussian distribution, then it is reasonable to construct a single sample covariance matrix from the data to estimate the network. However, real data is not always Gaussian; and such an assumption can be limiting, especially when analyzing multiple data sources simultaneously, since non-Gaussianity in a single data source will result in the non-Gaussianity of the combined data. A lot of previous work has been done to drop the Gaussianity assumption in the solution to classic problems like sparse regression [17], estimating GGMs [18], sparse CCA [19] etc., and propose non-parametric solutions to the same. We will also drop the assumption that the data is drawn from the same Gaussian distribution in the next section.

However, if the data is not drawn from the same Gaussian distribution, then how can we characterize the underlying network that generated the data? We propose a generative model where we assume that each data source is drawn from a semi-parametric Gaussian copula, where the copulas for the different data sources share the same covariance matrix, but have different functional transformations. To justify this model, we assume that for each data source, the data is sampled from a multi-variate Gaussian, but this sample is not directly observed. Instead, due to non-linearities introduced during data measurement, a transformed version of the data is measured. Each data source will have its own transformation, hence, the observed distribution of each data source will be different. The key idea of NP-MuScL is then to estimate the non-linear transformation, so that all data can be assumed Gaussian, and the network can be estimated using Equation 2.

Algorithm 1 Data generation model for NP-MuScL

Input: True covariance matrix $\mathbf{\Sigma}$ with $\sigma_{jj} = 1 \quad \forall\, j \in \{1, \cdots, d\}$
Input: Transformation function g_{ij}, mean μ_{ij} and variance ρ_{ij} for each feature j for each data source i.
for $i = 1$ to k **do**
  for $l = 1$ to n_i **do**
    $y \sim N(0, \mathbf{\Sigma})$
    for $j = 1$ to d **do**
      $\mathbf{X}^{(i)}(l, j) = \mu_{ij} + \rho_{ij} g_{ij}(y(j))$
    end for
  end for
end for
return Observed data $\mathbf{X}^{(i)}$ from k data sources.

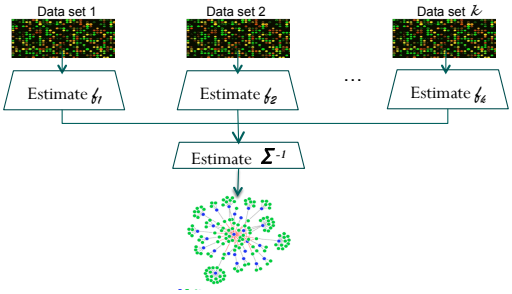


Fig. 1. The overall algorithm for NP-MuScL. Each data source is transformed into a Gaussian, using a nonparanormal, and the Gaussian data is then used to jointly estimate a inverse covariance matrix, giving the structure of the Gaussian Graphical Model, underlying the data.

2.3 Dropping the Gaussianity assumption

We model that each data source is drawn from an underlying Gaussian distribution with mean 0, and covariance matrix $\mathbf{\Sigma}$, where the variance of each feature $\sigma_{jj} = 1,\ \forall\, j \in \{1, \cdots, d\}$. However, the observed data may be some unknown transformation of the Gaussian data; thus, if $y \sim \mathcal{N}(0, \mathbf{\Sigma})$, then the observed data is $X^{(i)}(j) = \mu_{ij} + \rho_{ij} g_{ij}(y(j))$ where μ_{ij} and ρ_{ij} is the mean and standard deviation respectively of feature j in data source i.

The function g_{ij} is some (unknown) transformation that depends on the data source, our task is to estimate $f_{ij} = g_{ij}^{-1}$ from the data, so that $f_{ij}(X_j^{(i)})$ is Gaussian. The data generation process is then described in Algorithm 1.

2.4 NP-MuScL algorithm

A random vector X has a nonparanormal distribution $NPN(\mu, \mathbf{\Sigma}, f)$ if there exists a function $f(X) = (f_1(X_1), f_2(X_2), \cdots, f_d(X_d))$ such that $f(X)$ has a multi-variate Gaussian distribution $\mathcal{N}(\mu, \mathbf{\Sigma})$ [18]. To preserve identifiability, we constrain each f_j to have mean 0 and standard deviation 1. The nonparanormal distribution is a Gaussian copula when the fs are monotone and differentiable. For our model, we assume that each data source $X^{(i)} \sim NPN(\mathbf{0}, \mathbf{\Sigma}, f_i)$, that is, while each data source has its own functional transformation, they all share the same underlying relationship between the nodes, represented by $\mathbf{\Sigma}$. The mean of each copula is zero, since we constrain the estimated functions f_j to have zero means. Then, for nonparanormal data, it can be shown that conditional independence in the corresponding graph is equivalent to zeros in the inverse covariance matrix $\mathbf{\Sigma}^{-1}$ [18].

This suggests the following two step algorithm. For each data source i and each feature j, we first estimate the sample mean μ_{ij} and sample variance ρ_{ij}.

$$\hat{\mu}_{ij} = \frac{1}{n_i}\sum_{l=1}^{n_i} \mathbf{X}^{(i)}(l,j); \qquad \hat{\rho}_{ij}^2 = \frac{1}{n_i}\sum_{l=1}^{n_i}\left(\mathbf{X}^{(i)}(l,j) - \hat{\mu}_{ij}\right)^2 \tag{4}$$

The data in each data source is normalized by the appropriate μ and ρ to have mean 0 and standard deviation 1. Non-parametric functions f_{ij} are estimated for each data source i and feature j, so that $f_{ij} \sim \mathcal{N}(0,1)$. The details of estimating f are discussed in Sec. 2.6.

In the second step, the inverse covariance matrix is estimated jointly from the transformed f_is. We can define $\mathbf{Y}^{(i)} \in \mathbb{R}^{n_i \times d}$ as

$$\mathbf{Y}^{(i)}(\cdot, j) = \hat{f_{ij}}\left(\mathbf{X}^{(i)}(\cdot, j)\right) \forall j \in \{1, 2, \cdots d\} \tag{5}$$

The distribution of $\mathbf{Y}^{(i)}$ is then Gaussian with covariance matrix $\mathbf{\Sigma}$. The graphical model corresponding to all data sources can be jointly estimated as

$$\mathbf{\hat{\Sigma}^{-1}} = \arg\max_{\mathbf{\Theta}\succeq 0} \sum_{i=1}^{k} w_i \left\{\log\det\mathbf{\Theta} - \mathrm{tr}(\mathbf{\Theta}\ \mathbf{\hat{S}_f}^{(\mathbf{i})})\right\} - \lambda||\mathbf{\Theta}||_1 \tag{6}$$

where

$$\mathbf{\hat{S}_f}^{(\mathbf{i})} = \frac{1}{n_i}\sum_{l=1}^{n_i} \mathbf{Y}^{(i)}(l,\cdot)^T\mathbf{Y}^{(i)}(l,\cdot) \tag{7}$$

Setting the weights $w_i = \frac{n_i}{n}$ is equivalent to the data in each data source being drawn i.i.d. from the corresponding Gaussian copula; while setting different weights suggests that the effective sample size of a data source is not the observed sample size.

2.5 Optimization

The objective function in Equation 6 can be rewritten as

$$\mathbf{\hat{\Sigma}^{-1}} = \arg\max_{\mathbf{\Theta}\succeq 0}\ \log\det\mathbf{\Theta} - \mathrm{tr}(\mathbf{\Theta}\ \sum_{i=1}^{k} w_i\mathbf{\hat{S}_f}^{(\mathbf{i})}) - \lambda||\mathbf{\Theta}||_1 \tag{8}$$

Thus, by using $\sum_{i=1}^{k} w_i\mathbf{\hat{S}_f}^{(\mathbf{i})}$ as the covariance matrix, we can optimize the above objective by using efficient, known algorithms like glasso. The overall NP-MuScL algorithm is summarized in Figure 1.

2.6 Estimating $\hat{f}$

For each feature j in data source i, we can compute the empirical distribution function as (where $\mathbb{I}$ is the indicator function)

$$\hat{F}_{ij}(t) = \frac{1}{n_i}\sum_{l=1}^{n_i} \mathbb{I}(X^{(i)}(l,j) \leq t) \tag{9}$$

The variance of such an estimate may be very large, when computed in the high dimensional scenario $d >> n$. Liu et. al.[18] propose using a Windsorized estimator, for the same, where very small and large values of $\hat{F}_{ij}(t)$ are bounded away from 0 and 1 respectively. Thus,

$$\tilde{F}_{ij}(t) = \begin{cases} \delta_n & \hat{F}_{ij}(t) < \delta_n \\ \hat{F}_{ij}(t) & \delta_n \leq \hat{F}_{ij}(t) \leq 1 - \delta_n \\ 1 - \delta_n & \hat{F}_{ij}(t) \geq 1 - \delta_n \end{cases} \tag{10}$$

where δ_n is a truncation parameter. A value of δ_n chosen to be $\delta_n = \frac{1}{4n^{1/4}\sqrt{\pi \log n_i}}$ is found to give good convergence properties for estimating the network for a single data source [18]; and we use the same estimate for NP-MuScL.

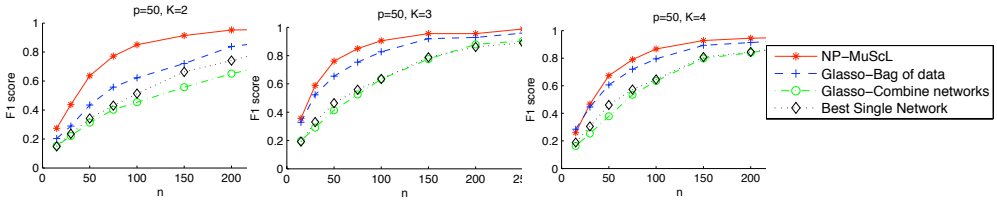
Now, for any continuous pdf f, the distribution of the cdf $F(x) = P(X \leq x)$ is uniform. Then, the distribution of $\Phi^{-1}(F(x))$ is Gaussian with mean zero, and standard deviation one, as required (where Φ is the cdf of the standard Gaussian). Thus, we can estimate the required function by using the marginal empirical distribution function defined above: $\hat{f}_{ij}(x) = \Phi^{-1}(\tilde{F}_{ij}(x))$.

3 Results

We first demonstrate that when multiple data sources have different distributions, NP-MuScL can extract the underlying network more accurately than other methods. Next, we show that NP-MuScL can identify the correct network, even when one of the data sources is noise. To analyze NP-MuScL on real data, we run NP-MuScL on two microarray yeast data sets, and find that the network obtained by NP-MuScL predicts more known edges of the yeast interaction network than other methods. Finally, we analyze NP-MuScL on Drosophila embryonic gene expression data from 3 data sets of ISH images and microarray.

3.1 Multiple data sources with different distributions

Data generation The details of generating the data for different experiments is described in detail in the supplementary material. In brief, we construct an inverse covariance matrix with an equivalent random sparse Gaussian graphical model. Data is sampled from the Gaussian, and then transformed into non-Gaussian distribution using different transformations. For $d = 50$ with $k = 2$ data sources, we use the Gaussian cdf ($\mu_0 = 0.05, \sigma_0 = 0.4$) and power transform ($\alpha = 3$) for the two data sources respectively (see supp. material for details). The task then is to jointly use the data from the two sources to extract the network. For $k = 3$ data sources, we use the identity transform for the third data source, so that the data sampled from the third source is truly Gaussian. For $k = 4$ data sources, the fourth data source is Gaussian noise, to test the performance of the algorithms in the presence of noise. We generate the same amount of data in each source (n), and run the experiment as n varies. Each result is reported as the average of 10 randomized runs of the experiment.



(a) (b) (c)

Fig. 2. $F1$ score for predicting edges in simulated data, as n is varied, for (a) $k = 2$, (b) $k = 3$, and (c) $k = 4$ data sources. The standard deviation in the results is small and almost constant across the different experiments; it ranges from (0.01-0.03), and is hence not displayed on the plot.

Metrics We report the $F1$ measure, which is the harmonic mean of precision and recall, as a measure of the accuracy of predicting the edges in the network.

Baselines We report three baselines. The first baseline is to report the best accuracy found by a single data source (Best Single Network). We assume that an oracle tells us which data source is most predictive. In our data experiments, we found that it was not possible to predict the most informative data source without using an oracle. Even when $k = 3$, the identity transformed source was not always the most informative. The second baseline is the glasso-bag of data, described in Section 2.2. The third baseline is to compute a separate network for each data source using glasso, and combine the networks to predict a single network (glasso-combine networks). An edge in the final network is present if it is present in m out of the k networks from the k data sources. We assume an oracle defines the best value of m for a given data set, the best value of m varied with different data sets.

As can be seen in Figure 2, NP-MuScL outperforms all three baselines significantly in all three scenarios. Interestingly, using the best single source outperforms estimating separate networks, and combining them in a second step. Note that an oracle is used for identifying the best source, as well as the optimal m used to combine networks. Hence, in a real world scenario, we may expect combining different data sources to perform as well as using only the best single data source for network prediction. When $k = 4$ (Figure 2(c)), one of the data sources is Gaussian noise, however, the use of the oracle in the "Glasso-combine networks" and the "single best source" baselines allows these baselines to ignore the noise source completely. However, NP-MuScL is still able to identify more correct edges in the network. Using a paired t-test, we found that the difference in $F1$ scores between NP-MuScL and "glasso-bag of data" is significant in all conditions, with P-value $p = 10^{-4}$.

3.2 Yeast data

In this experiment, we look at two different yeast microarray data sets, and make joint predictions via NP-MuScL. Data source 1 is a set of 18 expression profiles from Cho et. al. [20], where each expression corresponds to a different stage in the

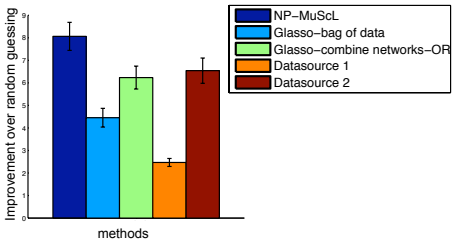


Fig. 3. Performance of different methods on predicting edges in the yeast network.

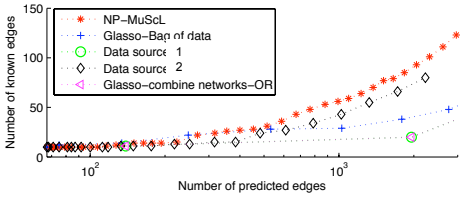


Fig. 4. Effect of varying tuning parameter on different methods. For a fixed number of predicted edges, the NP-MuScL method predicts more known edges than the other methods.

cell cycle of the the yeast. Data source 2 is a set of 300 expression profiles from Hughes et. al.[21], where each expression corresponds to a different knockout mutant of the yeast. Both data sets are processed using standard microarray processing algorithms [22].

We use a list of known interactions from BioGrid [23] to test how well do the different algorithms predict the known edges. Note that since the known gene interactions is an incomplete set, predicted gene interactions may be interactions that have not been observed yet, and thus, have not been added to the BioGrid data base. Hence, measuring recall is no longer appropriate, and we report the improvement in accuracy over random prediction of edges, as suggested by Liben-Nowell & Kleinberg [24].

The total data is over 6120 genes, we sample 1000 genes at a time, and run the algorithms for them. Results are reported for 10 random sub-samples of the genes. Figure 3 shows the improvement over random prediction for edges predicted by each method. Due to the amount of data available, the knockout mutant expression profiles capture more information (and hence more known edges) than the time series expression. Surprisingly, both methods of combining information without taking non-Gaussianity into account, perform worse than using only data source 2. NP-MuScL is the only method where using both data sets into account increases the number of correctly predicted edges. The same results were found to hold true when the network is predicted over the entire set of 6120 genes - NP-MuScL did significantly better than all other methods, and both glasso bag-of-data and glasso-OR did worse than using only data set 2.

To test the effect of varying tuning parameter λ, Figure 4 plots the number of known edges predicted by each method, versus the total number of edges predicted, as λ is varied. For very large values of λ when few edges are predicted, NP-MuScL and "glasso-Bag of data" perform equally well, however, as the amount of predictions increase, NP-MuScL outperforms other methods significantly.

Figure 5 shows the transformations learned for the two data sets by NP-MuScL for 4 random genes. A straight line corresponds to Gaussian data, non-

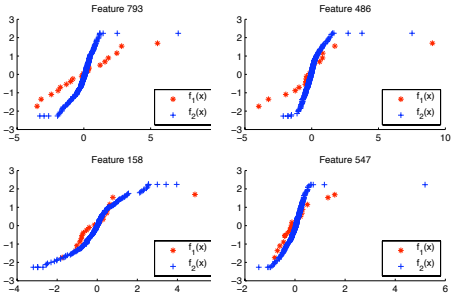


Fig. 5. Examples of the transformations made for data in source 1 (red) and source 2 (blue) for different features.

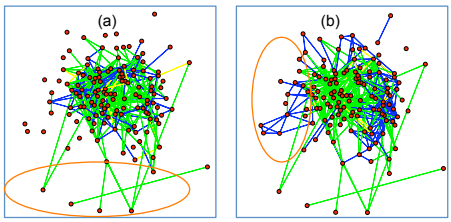


Fig. 6. Difference between the NP-MuScL network and (a) the 13-16 ISH network alone and (b) microarray network alone. Green edges are only predicted in the NP-MuScL network. Blue edges are only present in the (a) 13-16 ISH network and (b) microarray network.

NP-MuScL	Glasso Bag-of-data	Glasso OR	ISH 13-16	ISH 9-10	Microarray
7.29	4.88	4.06	5.98	2.35	3.66

Table 1. Improvement in prediction over random guessing for predicting gene interactions using Drosophila embryonic data.

linearities are clearly detected by the NP-MuScL algorithm. The transformations also seem to be damping extremely large values observed in the features.

3.3 Drosophila embryonic data

We study three data sets of Drosophila embryonic gene expression for 146 genes [25]. The first data set measures spatial gene expression in embryonic stage 9-10 of Drosophila development via in-situ hybridization (ISH) images (4.3 to 5.3 hours after fertilization), when germ band elongation of the embryo is observed. The second data set also studies ISH images measuring spatial gene expression in the 13-16 stage of embryonic development (9.3 to 15 hours after fertilization), when segmentation has already been established. The last data set is of microarray expression at 12 time points spaced evenly in embryonic development.

The ISH images were processed to extract 311 data points for each data set, as described in Puniyani & Xing [26]. The microarray data was processed using standard microarray processing algorithms. Since the number of data points extracted from the ISH data is dependent on the image processing algorithm used, using weights proportional to the number of data points is no longer suitable. We expect the microarray data to be as informative as the ISH data, hence we use $w_i = 0.25$ for each of the two ISH data sources, and $w_i = 0.5$ for the microarray data. The results in Table 1 show that NP-MuScL outperforms using the data separately, and glasso bag-of-data and glasso-combine networks (m=1, called glasso-OR).

We visualized the differences in edge prediction between the NP-MuScL network and the networks predicted by analyzing only one single data source at a

time. The orange ellipse in Figure 6(a) highlights gene interactions predicted by NP-MuScL by analyzing all 3 data sources, which were not predicted by any single data source. Figure 6(b) highlights interactions predicted by the microarray data that were not predicted either by the ISH data or the NP-MuScL network. The 9-10 ISH network is similar to the 13-16 ISH network, and hence, is not shown. A detailed analysis of the specific differences in the gene interactions predicted by the different methods is ongoing.

4 Conclusions

We proposed NP-MuScL, an algorithm that predicts gene interaction networks in a global, unsupervised fashion by jointly analyzing multiple data sources to capture the conditional independencies observed in the data. NP-MuScL models each data source as a non-parametric Gaussian copula, with all data sources having different mean and transformation functions, but sharing the covariance matrix across the underlying copulas. The network can then be efficiently estimated in a two step process, of transforming each data source into Gaussian, and then estimating the inverse covariance matrix of the Gaussian using all data sources jointly. We found that NP-MuScL significantly outperforms baseline methods in both synthetic data, and two experiments predicting a gene interaction network from two yeast microarray data sets, and three Drosophila ISH images and microarray data sets.

One limitation of NP-MuScL is that the weights giving the importance of each data source must be assigned by the user. While a good estimate of the weights may be obtained if all data sources are truly drawn i.i.d. from their nonparanormal distributions, and have similar noise levels; in practice, some data sources may be known to be noisier than others, or known to not be i.i.d. (eg. microarray experiments over time are not truly independent draws from the distribution). The question of automatically learning the weights from data remains an open challenge.

References

1. Segal, E., Koller, D., Friedman, N.: Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data. Nature genetics **34** (2003) 166–176
2. Basso, K., Magolin, A., Califano, A.: Reverse engineering of regulatory networks in human b cells. Nature Genetics **37** (2005) 382–390
3. Morrissey, E.R., Juárez, M.A., Denby, K.J., Burroughs, N.J.: On reverse engineering of gene interaction networks using time course data with repeated measurements. Bioinformatics **26**(18) (2010) 2305–2312
4. Carro, M.S., Califano, A., Iavarone, A.: The transcriptional network for mesenchymal transformation of brain tumours. Nature **463** (2010) 318–325
5. Wang, K., Saito, M., Califano, A.: Genome-wide identification of post-translational modulators of transcription factor activity in human b-cells. Nature Biotechnology **27**(9) (2009) 829–839

6. Meinshausen, N., Bühlmann, P.: High-dimensional graphs and variable selection with the lasso. Annals of Statistics (2006)
7. Banerjee, O., Ghaoui, L.E., d'Aspremont, A., Natsoulis, G.: Convex optimization techniques for fitting sparse gaussian graphical models. In: ICML. (2006)
8. Friedman, J., Hastie, T., Tibshirani, R.: Sparse inverse covariance estimation with the graphical lasso. Biostatistics (2007)
9. Ben-Hur, A., Noble, W.S.: Kernel methods for predicting protein–protein interactions. In: ISMB. Volume 21. (2005) i38–i46
10. Wang, Y., Joshi, T., Zhang, X.S., Xu, D., Chen, L.: Inferring gene regulatory networks from multiple microarray datasets. Bioinformatics **22**(19) (2006) 2413–2420
11. Ahmed, A., Xing, E.P.: Tesla: Recovering time-varying networks of dependencies in social and biological studies. Proc. Natl. Acad. Sci. **106** (2009) 11878–11883
12. Xu, Q., Hu, D.H., Yang, Q., Xue, H.: Simpletrppi: A simple method for transferring knowledge between interaction networks for ppi prediction. In: Bioinformatics and Biomedicine Workshops. (2012)
13. Katenka, N., Kolaczyk, E.D.: Inference and characterization of multi-attribute networks with application to computational biology. Arxiv (2012)
14. Honorio, J., Samaras, D.: Multi-task learning of gaussian graphical models. In: ICML. (2011)
15. Rothman, A.J., Bickel, P.J., Levina, E., Zhu, J.: Sparse permutation invariant covariance estimation. Electronic Journal of Statistics **2** (2008)
16. Tibshirani, R.: Regression shrinkage and selection via the lasso. J R Statist. Soc B **58**(1) (1996) 267–288
17. Ravikumar, P., Liu, H., Lafferty, J., Wasserman, L.: Spam: Sparse additive models. In: NIPS. (2007)
18. Liu, H., Lafferty, J., Wasserman, L.: The nonparanormal: Semiparametric estimation of high dimensional undirected graphs. Journal of Machine Learning Research **10** (2009) 2295–2328
19. Balakrishnan, S., Puniyani, K., Lafferty, J.: Sparse additive functional and kernel cca. In: ICML. (2012)
20. Cho, R., Campbell, M., Winzeler, E., Davis, R.: A genome-wide transcriptional analysis of the mitotic cell cycle. Mol Cell **2**(1) (1998) 65–73
21. Hughes, T., Marton, M., Jones, A., Roberts, C., Friend, S.: Functional discovery via a compendium of expression profiles. Cell **102**(1) (2000)
22. Hibbs, M., Hess, D., Myers, C., Troyanskaya, O.: Exploring the functional landscape of gene expression: directed search of large microarray compendia. Bioinformatics (2007)
23. Stark, C., Breitkreutz, B., Chatr-Aryamontri, A., Boucher, L., Tyers, M.: The biogrid interaction database: 2011 update. Nucleic Acids Res. **39**(D) (2011) 698–704
24. Liben-Nowell, D., Kleinberg, J.: The link prediction problem for social networks. In: CIKM. (2003)
25. Tomancak, P., Beaton, A., Weiszmann, R., Kwan, E., Shu, S., Lewis, S., Richards, S., Celniker, S., Rubin, G.: Systematic determination of patterns of gene expression during drosophila embryogenesis. Genome Biol **3**(2) (2002) 14
26. Puniyani, K., Xing, E.P.: Inferring gene interaction networks from ish images via kernelized graphical models. In: 13th ECCV. (2012)