BIOINFORMATICS

Vol. 00 no. 00 2005
Pages 1–33

Supplementary Material: Comparative Evaluation of Reverse Engineering Gene Regulatory Networks with Relevance Networks, Graphical Gaussian Models and Bayesian Networks

Adriano V. Werhli [a,b,*], Marco Grzegorczyk [c] and Dirk Husmeier [a]

[a]BioSS - Biomathematics and Statistics Scotland, Edinburgh, United Kingdom
[b]School of Informatics, University of Edinburgh, United Kingdom
[c]Department of Statistics, University of Dortmund, Germany

ABSTRACT

This supplementary material is divided as follows: Section 1 has a description about how to simulate Genetic Regulatory Networks and Section 2 explains how to perform these simulations using Netbuilder.

Section 3 presents details about the Graphical Gaussian Models and Section 4 presents details about Bayesian Networks. In Section 5 we discuss how the data was pre-processed for our analysis. Section 6 presents the v-structure network, which was used to create simulated data to give some idea of how the network topology influences the different inference methods.

In Section 7 we present an overview of all the results. Sections 8 and 9 present tables for comparing the performance between methods. These sections present comparisons for AUC scores and TP scores respectively.

Sections 10 and 11 present tables for comparing the performance between observational and interventional data sets. These sections present comparisons for AUC scores and TP scores respectively.

Sections 12 and 13 present tables for comparing the performance between different network topologies. These sections present comparisons for AUC scores and TP scores respectively.

1 SIMULATING GENE REGULATORY NETWORKS

All cells in an organism carry the same DNA, but synthesized protein can be totally different. This is due to genetic regulation. Protein synthesis is regulated by control mechanisms at different stages: transcription, RNA splicing, translation and post-translational modifications. Since the advent of microarray experiments, a huge amount of data has been produced. In microarray experiments an organism is exposed to different conditions and the expression level of their genes are measured. One of the main applications to this data is to infer the set of relationships between this genes, the so called Genetic Regulatory Network (GRN). The real GRN normally is not known, being very difficult to evaluate the performance of learning algorithms. The advantage of simulated data is that the network structure is known, making possible to access the learning algorithm's performance. Often the simulated data are drawn from the multivariate Gaussian distribution while biological data rarely is Gaussian distributed. Another problem is the intricacies of the regulation by complex *cis*-regulatory modules what makes the data far from being linearly dependent (Pournara, 2005). A model to simulate GRNs must be simple, possible to parametrize, and yet produce data that resemble biological realistic data.

1.1 Genetic Networks Dynamics

The machinery inside the cell is responsible for synthesizing proteins from DNA. In a simplistic view this involves:

1. Begin with a DNA strand.
2. Transcription: The process of building an RNA copy of a DNA sequence. This process starts when one or more transcription factors (TF) bind to a *cis*-regulatory domain of the gene.
3. Translation: The process of matching amino acids to corresponding sets of three bases (codons). During translation messenger RNA (mRNA) sequences are used to manufacture proteins. Translation occurs at special structures in the cell called ribosomes. Ribosomes are the "factories" where RNA is used to manufacture proteins.
4. Post-translational modifications: These are modifications that occur in proteins after they are released from the ribosomes.
5. Finishes with a new protein.

To model these processes inside the cell, it is necessary to remember some concepts from chemical kinetics. As an example, consider the first order reaction where a reactant A is converted into a product B. It is represented by:

$$A \xrightarrow{k} B$$

The velocity, or rate of the reaction accordingly with the law of mass action, u is given by:

$$u = \frac{d[B]}{dt} = -\frac{d[A]}{dt} = k[A] \tag{1}$$

Applying the law of conservation of mass:

$$[A]_0 = [A] + [B] \tag{2}$$
$$[A] = [A]_0 - [B]$$

*to whom correspondence should be addressed

$$u = \frac{d[B]}{dt} = -\frac{d[A]}{dt} = k[A] = k([A_0] - [B]) \quad (3)$$

where $[A]$ and $[B]$ are concentrations at a time t, $[A]_0$ is the initial concentration, and k is the rate constant. If we have a reversible reaction like,

$$A \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} B$$

where k_1 is the forward rate constant and k_{-1} is the backward rate constant, then the reaction rate is expressed as:

$$u = \frac{d[B]}{dt} = k_1[A] - k_{-1}[B] \quad (4)$$

Applying the law of conservation mass:

$$\begin{aligned} [A]_0 &= [A] + [B] \quad (5)\\ [A] &= [A]_0 - [B] \\ \frac{d[B]}{dt} &= k_1([A]_0 - [B]) - k_{-1}[B] \\ &= k_1[A]_0 - [B](k_1 + k_{-1}) \end{aligned}$$

Knowing how to calculate the reaction's rate to first order reactions, a concept that was developed to describe small substrate-enzyme systems is used to model the process of a protein (TF) binding to a TF binding site and starting transcription. Note that as this concept was developed for small molecules, and the systems that we are investigating are much larger (TF, TF binding sites and transcription), the transcriptional and translational time delays are being ignored.

Michaelis and Menten (1913) proposed a way to model how enzymes act as catalysers speeding up the conversion of a substrate into a product. They showed that as a concentration of a substrate increases, the rate of the reaction increases only up to a certain extent. They proposed the following mechanism for an enzyme substract reaction:

$$S + E \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} ES \xrightarrow{k_2} E + P$$

where E is the enzyme, S is the substrate, P is the product and k_1 and k_2 are the association rates for the enzyme-substrate complex, ES, and the product respectively and k_{-1} is the dissociation rate constant for the enzyme substract complex. Assuming that a steady-state will be reached, and then concentration of ES will be constant, Briggs and Haldane (1925) derived the Michaelis and Mentem equation as follows:

$$\frac{d[ES]}{dt} = k1[E][S] - k_{-1}[ES] - k_2[ES] \quad (6)$$

applying the law of conservation mass:

$$\begin{aligned} [E]_0 &= [ES] + [E] \quad (7)\\ [E] &= [E]_0 - [ES] \end{aligned}$$

then,

$$\frac{d[ES]}{dt} = k_1([E_0] - [ES])[S] - k_{-1}[ES] - k_2[ES] = 0 \quad (8)$$

and it follows that:

$$[ES] = \frac{k_1[E_0][S]}{k_{-1} + k_2 + k_1[S]} \quad (9)$$

so, the rate of reaction is given by,

$$u = k_2[ES] = \frac{k_2 k_1[E_0][S]}{k_{-1} + k_2 + k_1[S]} = \frac{k_2[E_0][S]}{\frac{k_{-1}+k_2}{k_1} + [S]} \quad (10)$$

Making $V = k_2[E_0]$, where V is the limiting rate constant and $K_M = \frac{k_{-1}+k_2}{k_1}$ where K_M is the Michaelis constant, the equation can be rewritten as:

$$u = \frac{V[S]}{K_M + [S]} \quad (11)$$

The above derivations are for one enzyme and one substrate. When we use this concepts to model a TF and a TF binding site we should remember that TFs can act in various different forms to start transcription. Only as an example let's consider that there are three TFs T_a, T_b and T_c. Some hypothetical possibilities are:

1. T_a alone gives rise to a transcription rate of a certain fraction of the maximum.
2. T_b alone doesn't initiate any transcription.
3. T_a and T_b together give rise to transcription at the maximum.
4. T_a and T_b and T_c repress the gene transcription.

In these few examples above we can see that the combined effect of different TFs is not necessarily the sum of their individual effects. We take an example from Pournara (2005) where two transcription factors are considered, one activating, T_a, and other inhibiting, T_i, controlling the transcription of gene G. The system is described as:

$$T_i \underset{k_{13}}{\overset{k_{31}}{\rightleftharpoons}} G \underset{k_{21}}{\overset{k_{12}}{\rightleftharpoons}} T_a$$

Then the concentration of the mRNA of gene G is given by:

$$[G] = \frac{k_{12}[T_a]k_{31}}{k_{21}k_{31} + k_{12}[T_a]k_{31} + k_{13}[T_i]k_{21}} \quad (12)$$

Another possibility is the existence of more than one binding site, giving origin to what is called enzyme cooperativity. In this case the rate of reaction does not follow the Michaelis Menten equation (11), but instead follows the equation proposed by Hill (1910), the so called Hill equation, where the sigmoidal characteristic is evident:

$$u = \frac{V[S]^h}{K_{0.5}^h + [S]^h} \quad (13)$$

where $K_{0.5}^h$ is the Michaelis constant only when $h = 1$ and h is the Hill coefficient which gives an upper limit for the number of binding sites.

1.2 Simulation Methods

To model this dynamic process of protein production there are various approaches. A very refined model is the description of the biophysical processes as a system of coupled differential equations assuming the system is in a steady state. While this model keeps the

realism of the true network it is very difficult to parametrize due to the large number of parameters of which only few are known.

Noting similarities between GRNs and electronic circuits, several authors tried to design GRNs using techniques that were originally developed for electronic circuits. The first attempts using this approach used boolean networks. The boolean network consist of digital switches that can either assume values 0 or 1 and these switches can be combined to form logical operations as, for example AND and OR. However the pure boolean networks have low prediction power because in real GRNs the values of transcription range from 0 to maximum, and these rates influence the rates of others transcriptions, and hence the network dynamics.

Another approach was tried in Vohradsky Vohradsky (2001) using neural networks, where the genes are the nodes and the connections between the genes have weights expressing the relationship between them, if the weight is zero there is no connection. Furthermore with this approach it is possible to have positive or negative weights meaning activation or inhibition of a gene respectively, and the values of the transcription rate are continuous. The problem with this approach is related to the way the TFs can act together to initiate the transcription. As mentioned before the action of more than one TF is not necessarily the sum of their individual action.

In order to combine the positive aspects of the boolean networks and the neural networks representations, a combination of both was proposed in Yuh et al. (1998) and Yuh et al. (2001). In this approach genes are modeled as sigma-pi units, which were introduced by Rumelhart Rummelhart (1987) as nodes in higher order neural networks to avoid linear separability constraints associated with first-order neural networks. Boolean functions and logic gates can be expressed in a sigma-pi formalism, and their input and output are not restricted to boolean values. Sigma-pi units are combinatorial, so simpler units connected can lead to a very complex module.

We remember then, from the discussion in section 1.1, that following a simple model of enzyme-substrate interaction (ignoring time delays) and using chemical kinetics leads us to a set of ODEs describing the biophysical system. Assuming a steady state of this system, it is possible to derive a set of equations that describe the concentration of products as non-linear functions of combination of substrates. The resulting equations are a combination of multiplications and sums of sigmoidals. So instead of solving the steady-state approximation to ODEs explicitly, it is possible to model the system using the sigma-pi formalism, what makes the modeling much simpler with less parameters. This is the approach implemented in Netbuilder.

2 GENERATING DATA WITH NETBUILDER

NetBuilder (Yuh et al., 1998, 2001) is an interactive graphical tool for representing and simulating genetic regulatory networks in multicellular organisms. Some concepts used to create genetic networks diagrams in NetBuilder are borrowed from electronic engineering.

Conventional models are based on detailed descriptions of the individual chemical reactions that form a biochemical pathway but the number of parameters necessary to specify it is extremely large. Simplifying models while maintaining their main characteristics is often extremely difficult.

The main idea of Netbuilder is instead of solving the steady-state approximation to ODEs explicitly we approximate them with a qualitatively equivalent combination of multiplications and sums of sigmoidal transfer functions.

In NetBuilder as in many electronic circuit design packages pathways are represented as series of linked modules. Each module has specific input-output characteristics. As long as these characteristics conform to experimental observations, the exact transformations occurring inside the modules can be safely neglected. The result is a significant reduction in the number of parameters. Thus, NetBuilder aims to provide a way of quantifying intuitively drawn diagrams, and making experimentalists hypotheses testable.

There are four possible types of network nodes: Genes, Components, Functions and Receptors. The interaction between this nodes are represented by links which can be activators or repressors. Complex interactions that are difficult to model just with links make use of functions to represent them. It follows a brief explanation of each one of the possible network nodes:

- **Functions.** Functions are objects used to combine inputs, there are several functions implemented in Netbuilder but with particular interest for us are the logical functions *AND* and *OR*. These can have multiple entries and their outputs can be combined in order to perform other logical operations. The use of these logical functions are not restricted to binary values; it is also possible to use continuous values as inputs. As an example when two inputs x_1 and x_2 are combined using an *AND* or *OR* ports the outputs can be obtained with the following equations:

$$y_{\text{AND}} = x_1 \times x_2 \tag{14}$$

$$y_{\text{OR}} = x_1 + x_2 \times (1 - x_1) \tag{15}$$

- **Genes.** Each gene has a user specified number of inputs and one output. The inputs represents the *cis*-regulatory domain where one or more TFs can bind to initiate the transcription and the output represents protein production. A gene transforms its inputs into an output through some function, there are genes with pre-built functions *AND* or *OR*, and it is also possible to model the interaction within the inputs in a more complex way using a combination of functions. A gene with *AND* ports is used to model a process where if any of the TFs is low then the protein production is also low. Conversely a gene with an *OR* port is highly expressed if any of the TFs present in the input are high. Suppose we have a TF1 connected to one gene, the expression of this gene is calculated by:

$$y_1 = \frac{x_1}{x_1 + 1} \tag{16}$$

  where y_1 is the average occupation of the binding site of TF1, and x_1 is the concentration of TF1 relative to its equilibrium dissociation constant. If a TF2 is connected to a gene via a negated link, the expression is:

$$y_2 = 1 - (\frac{x_2}{x_2 + 1}) \tag{17}$$

  where y_2 is the TF2 binding site occupation, and x_2 is the concentration of TF2 relative to the dissociation constant of the gene-TF2 complex.

- **Components.** Components can be used, among other things, to specify external inputs to the network.

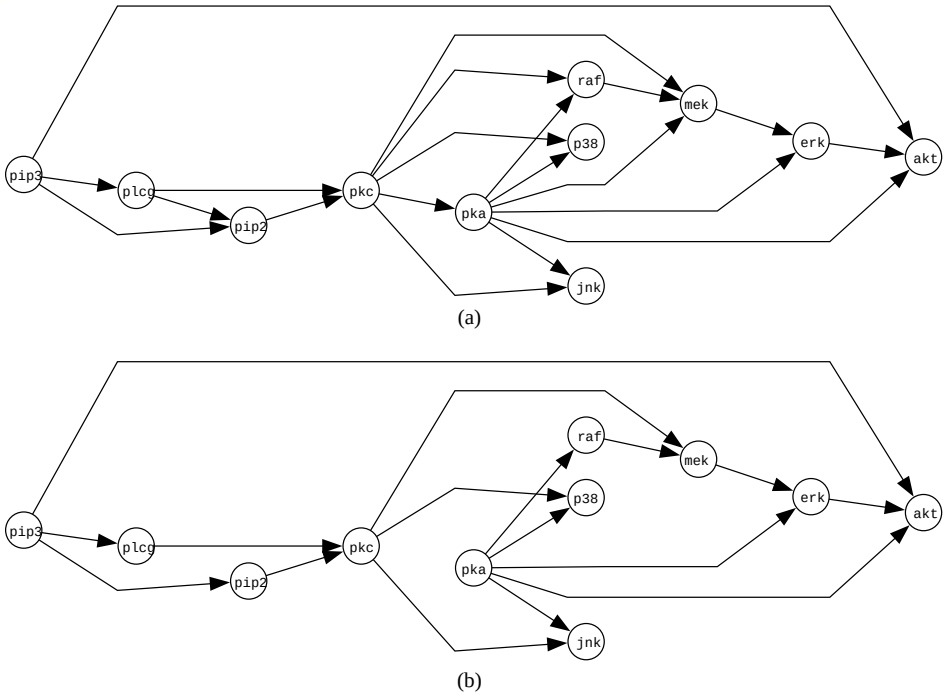


Fig. 1. Network structure. Panel (a) shows the original RAF network. Panel (b) shows the modified RAF network where the edges *PKC* → *RAF*, *PKC* → *PKA*, *PKA* → *MEK* and *PLCg* → *PIP2* were removed to increase the number of v-structures in the network.

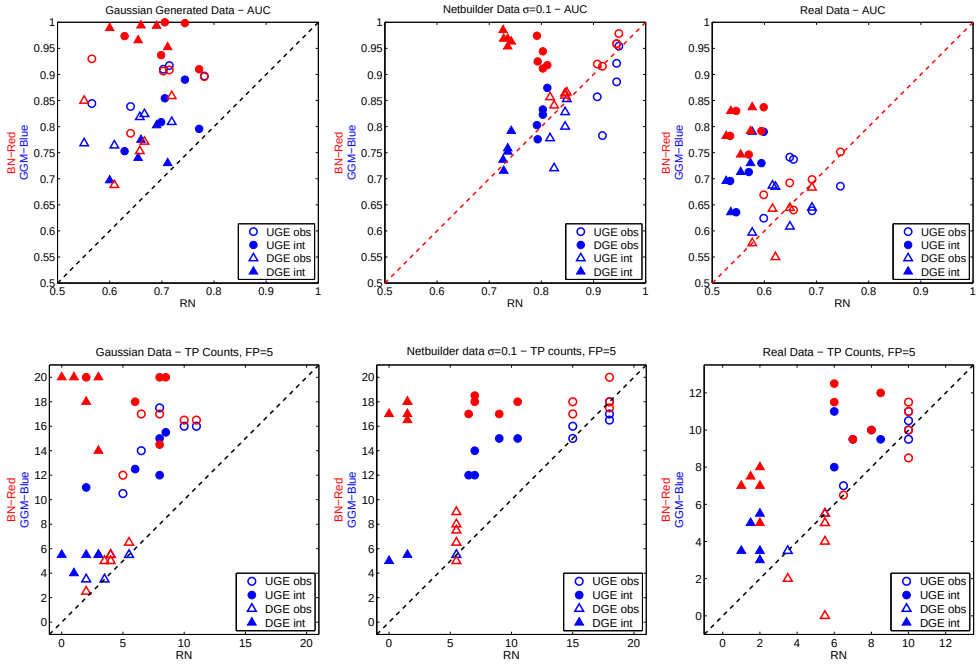
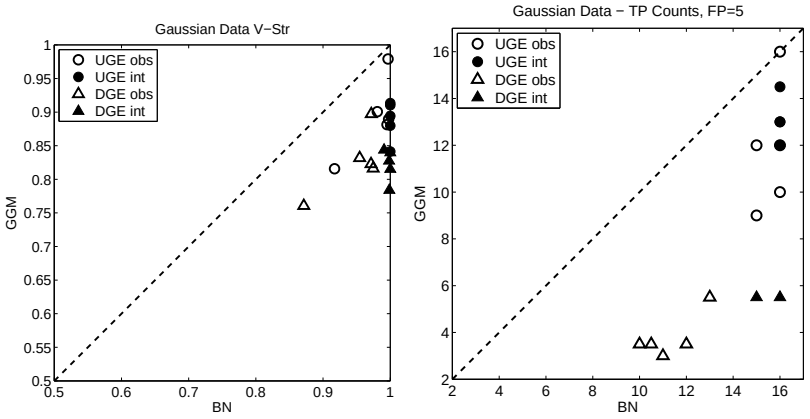


Fig. 2. GGMs and BNs versus RNs. This figure compares the performance of GGMs and BNs (vertical axis) with RNs (horizontal axis). The columns refer to different data sets. *Left column:* Gaussian data. *Centre column:* Data generated with Netbuilder, subject to additive Gaussian noise with $\sigma = 0.1$. *Right column:* Cytometry data. The two rows refer to different scoring criteria, discussed in Section 5 of the main paper.*Top row:* AUC score. *Bottom row:* TP count. The symbols of the six scatterplots are explained in the caption of Figure 3 of the main paper The colours refer to different comparisons. *Red:* BNs versus RNs. *Blue:* GGMs versus RNs.



(a) (b)

Fig. 3. GGMs vs BNs on Gaussian V-structure data. Scatter plots comparing the performance of GGMs (vertical axis) with BNs (horizontal axis). The diagonal line represents equal performance. Symbols above the line indicate that GGMs outperform BNs. Conversely symbols below that line point to a better performance of BNs over GGMs. Each subfigure compares the results obtained from two different data types, using only passive observations (empty symbols) and including active interventions (filled symbols). Two different evaluation criteria have been applied, based on directed graphs (DGE, represented by triangles) and their undirected skeletons (UGE, represented by circles). The two panels refer to two different scoring criteria. Left: AUC scores. Rigth: TP counts.

- **Receptors.** Receptors are used to transfer data between cells. In principle, NetBuilder Receptors can be used to represent any process in which signals get exchanged between cells, even those that do not actually involve biological cell surface receptors.

2.1 Simulating the RAF network

In order to simulate the Raf network we linked 11 genes with the same structure as we have in the network presented in Sachs et al. (2005), see figure 1(a). All the links between genes represent activations and all the interactions between TFs were set to *OR* regulation. A gene with an *OR* port is highly expressed if any of the TFs present in the input are high. The eleven genes we have in our network are: *RAF*, *MEK*, *PLCg*, *PIP2*, *PIP3*, *ERK*, *AKT*, *PKA*, *PKC*, *P38* and *JNK*.

For data generation we sample values from a uniform distribution, Uniform $\sim (0, 1)$, for all root nodes. Root nodes are nodes without any parents. These values are then propagated to the children nodes where they will be processed and then propagated further down in the network hierarchy until they reach the leave nodes. Leave nodes are nodes without any children. Every node that is not a root node has a sum function added to its output. This sum represents that the output of a node is subject to some additive noise. The values to be added as noise are sampled from a normal distribution $N \sim (0, \sigma^2)$. Having this way of adding dynamical noise we then generated data sets with three different noise levels, low, medium and high corresponding to, $\sigma = 0.01$, $\sigma = 0.1$ and $\sigma = 0.3$. Using this procedure we generated observational data. In total we generated 5 data sets with 100 data point for each noise level; these are called the Netbuilder observational data.

When generating interventional data sets some modifications of the network are needed. When an inhibition is simulated, the inhibited gene has its output forced to be zero independent of its inputs. So the output of such an inhibited node will be only the added noise. In the case where we want to activate an gene we set its value to one, again independent of its input. The output of an activated gene will be 1 plus the noise. The added noise in the nodes subject to inhibitions or activations was always sampled from $N \sim (0, \sigma^2)$ with $\sigma = 0.01$.

Data Points	Interventions
$1 \sim 16$	No Intervention
$17 \sim 30$	*AKT* inhibited
$31 \sim 44$	*PKC* inhibited
$45 \sim 58$	*PIP2* inhibited
$59 \sim 72$	*MEK* inhibited
$73 \sim 86$	*PKC* activated
$87 \sim 100$	*PKA* activated

Table 1. Interventional data set. Table showing how one interventional data set is built.

For this study we generated 5 interventional data sets for each noise level. These are called Netbuilder interventional data. Each interventional data set is composed by a total of 100 data points where some of the genes were intervened, see the table 2.1 for a detailed explanation of how the interventional data set is built. Our interventions try to mimic the ones that were used in Sachs et al. (2005).

3 DETAILS ON THE GAUSSIAN GRAPHICAL MODEL APPROACH

Gaussian graphical models (GGMs) are based on the assumption that the observed data are distributed according to a multivariate Gaussian distribution $N(\mu, \Sigma)$. The (i,j)-th element Σ_{ij} in the covariance matrix Σ is proportional to the correlation coefficient between node X_i and X_j. But a high correlation coefficient between two nodes must not necessarily indicate a direct causal association. Not rarely a high correlation coefficient is simply due to an indirect

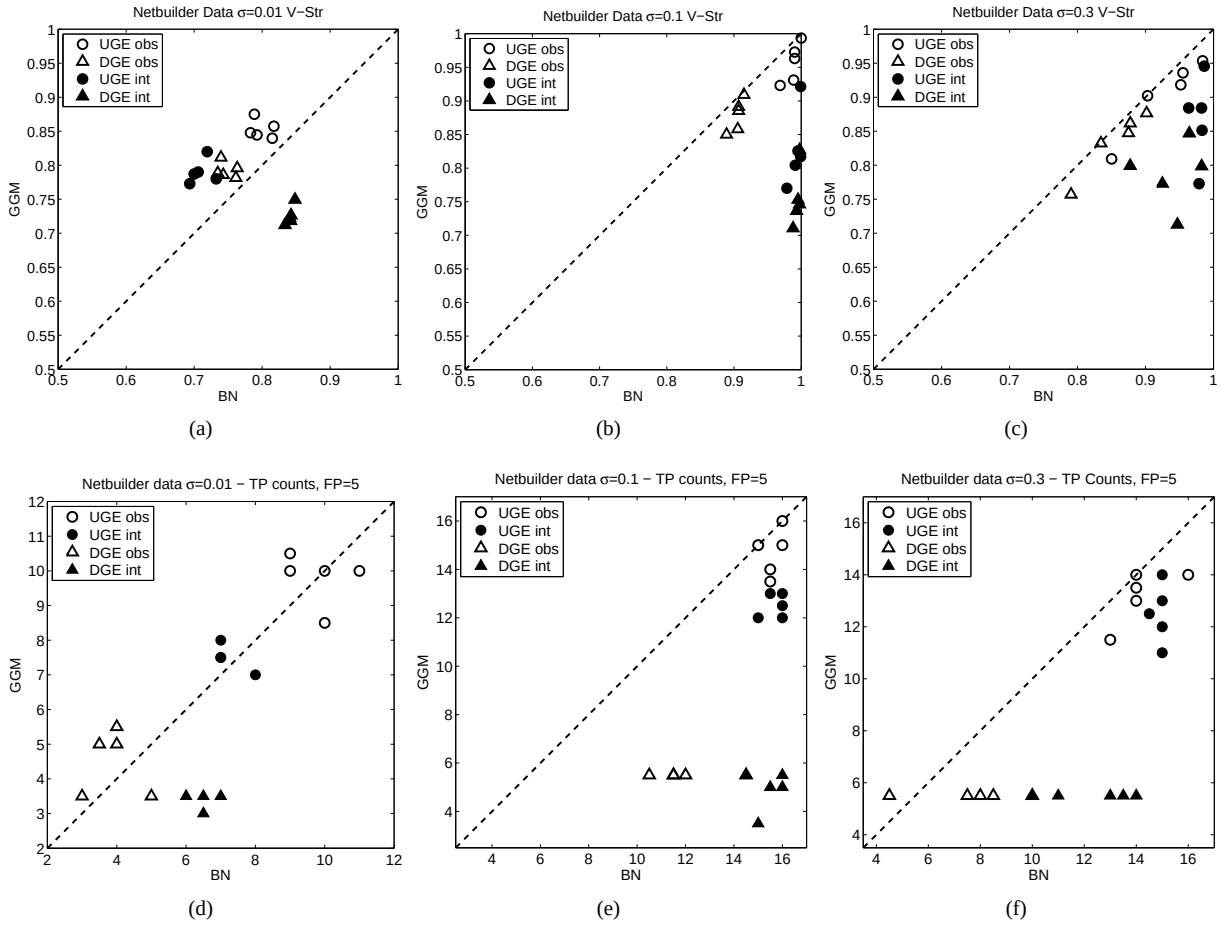


Fig. 4. GGMs vs BNs on Netbuilder V-structure data. This figure compares the performance of GGMs and BNs on the synthetic data generated with Netbuilder, for the topology with some edges removed. The columns refer to different standard deviations of the additive Gaussian noise. *Left column* $\sigma = 0.01$, *Centre column* $\sigma = 0.1$, *Right column* $\sigma = 0.3$. The two rows refer to different scoring criteria. *Top row:* AUC score. *Bottom row:* TP counts. A detailed explanation of the symbols is given in the caption of figure 3

association, e.g. both variables just depend on another network variable. Consequently, a high correlation coefficient between two variables provides only weak evidence for a direct association. And actually only the direct dependencies between variables (nodes) are of interest for the construction of regulatory networks. To avoid this shortcoming of Relevance network (RN) methodology, partial correlations are considered in Gaussian graphical models (GGMs) instead. That is, the strength of a direct association between two nodes X_i and X_j is measured by the partial correlation coefficient ρ_{ij} which describes the correlation between these nodes conditional on all the other network nodes. From the theory of normal distributions it is known that the partial correlation coefficients ρ_{ij} can be easily computed from the inverse $\Omega = \Sigma^{-1}$ of the covariance matrix Σ. More precisely, it holds:

$$\rho_{ij} = \frac{-\omega_{ij}}{\sqrt{\omega_{ii} \cdot \omega_{jj}}},$$

whereby ω_{ij} are the elements of the matrix $\Omega = \Sigma^{-1}$.
Hence, in order to reconstruct a Gaussian graphical model (GGM) from a given data set D, one typically employs the following procedure. From the data set D, the empirical covariance matrix is estimated and inverted to obtain Ω, subsequently the entries ρ_{ij} of the partial correlation matrix Π can be computed using the formula for ρ_{ij} given above. Afterwards the interpretation is as follows: Small elements ρ_{ij} in the resulting partial correlation matrix Π correspond to weak partial correlations, and the corresponding nodes become not connected by an edge. On the other hand, high entries correspond to strong partial correlations, so that there is reason to believe that there must be *direct* associations between the corresponding nodes.

In Schäfer and Strimmer (2005b) the authors present a novel regularized shrinkage covariance estimator which is based on the concept of shrinkage and exploits the Ledoit Wolf lemma for analytic calculation of the optimal shrinkage. This novel shrinkage estimator $\widehat{\Sigma}$ for the covariance matrix Σ is guaranteed to be non-singular, so that it can be inverted to obtain a new estimator $\widehat{\Omega} = (\widehat{\Sigma})^{-1}$ for the matrix Ω.
The new shrinkage estimator is based on the following theoretical idea. It is known that the unconstrained maximum likelihood

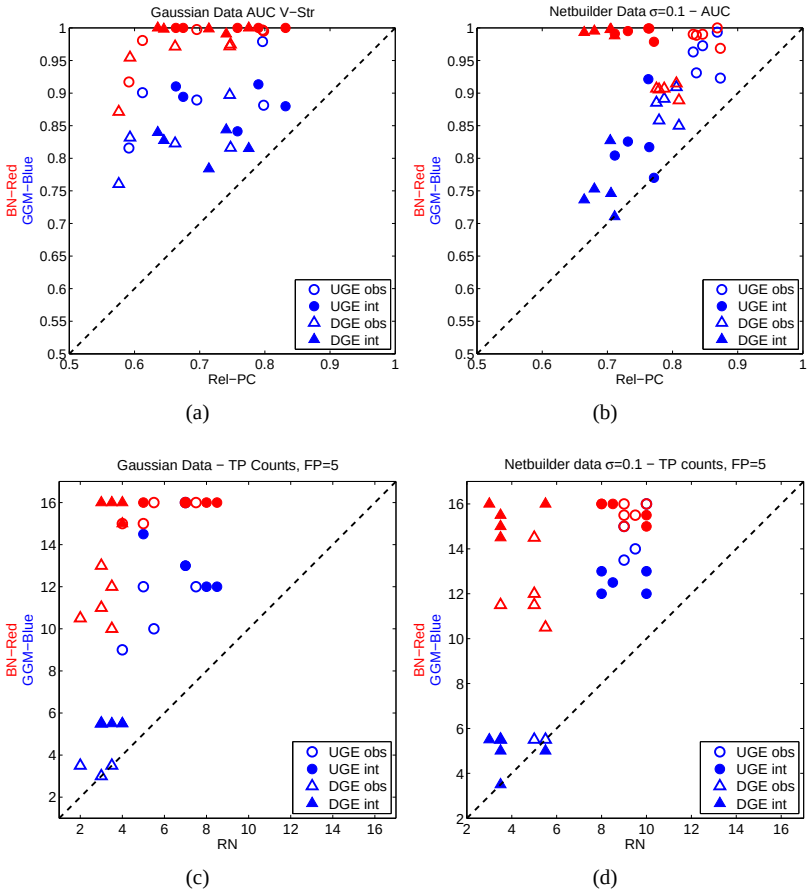


Fig. 5. GGMs and BNs vs RNs. V-structure data. This figure compares the performance of GGMs and BNs (vertical axis) with RNs (horizontal axis). The columns refer to different data sets. *Left column:* Gaussian data. *Right column:* Netbuilder data with additive Gaussian noise with $\sigma = 0.1$. The two rows refer to different scoring criteria. *Top row:* AUC score. *Bottom row:* TP counts. A detailed explanation of the symbols is given in the caption of figure 3. The colours refer to different comparisons. *Red:* BNs versus RNs. *Blue:* GGMs versus RNs

estimator $\widehat{\Sigma}_{ML}$ for the covariance matrix Σ has a high variance if the number of nodes exceeds the number of observations $(n > m)$. On the other hand there are lots of possible constrained estimators that have a certain bias but a lower variance. The shrinkage approach combines the maximum likelihood estimator with one of these constrained estimators $\widehat{\Sigma}_C$ in a weighted average:

$$\widehat{\Sigma} = (1 - \lambda) \cdot \widehat{\Sigma}_{ML} + \lambda \cdot \widehat{\Sigma}_C,$$

where $\lambda \in [0, 1]$ denotes the shrinkage intensity. The authors show that this regularized estimator outperforms both single estimators $\widehat{\Sigma}_{ML}$ and $\widehat{\Sigma}_C$ in terms of accuracy and statistical efficiency. Furthermore they show that the Ledoit Wolf lemma can be used to estimate the optimal shrinkage intensity λ, and recommend to restrict the constrained estimator $\widehat{\Sigma}_C$ by assuming that the network variables (nodes) are pairwise uncorrelated ($\Sigma_{C_{ik}} = 0$ for $i \neq k$) but may have unequal variances ($\Sigma_{C_{ii}} \neq \Sigma_{C_{kk}}$ for $i \neq k$). More details can be found in Schäfer and Strimmer (2005b). The computations for the Gaussian graphical models (GGMs) in our comparative evaluation study were carried out with the software provided by Schäfer and Strimmer (2005a).

4 DETAILS ON THE BAYESIAN NETWORK APPROACH

We sampled Bayesian Networks from the posterior distribution with the order-MCMC method of Friedman and Koller (2003). This method is a Markov Chain Monte Carlo (MCMC) sampling scheme that can be used to generate a sample of network node orderings $O_1, O_2, O_3, ...$ from the posterior distribution $P(O|D)$ over node orderings in the context of Bayesian networks. So, the state space is the set of all $n!$ possible orderings of the network nodes. Afterwards in a second step a sample of DAGs $G_1, G_2, G_3, ...$ can be obtained by sampling DAGs out of the sampled node orderings. Each network node ordering $O = (X_{\sigma(1)}, ..., X_{\sigma(n)})$ represents the set of all directed acyclic graphs which are consistent with the ordering, whereby a graph is consistent with O if and only if all its edges point rightwards with respect to the ordering. Friedman and Koller (2003) show that the likelihood $P(D|O)$ of a given node ordering O can be computed efficiently. In the space of node orderings the *flip-operator* which exchanges two nodes in the ordering for each other, can be used to construct a Markov chain over the space of node orderings whose distribution converges to $P(O|D)$. See Friedman and Koller (2003) for details.

	Gaussian				Netbuilder				Real			
	Obs		Int		Obs		Int		Obs		Int	
	DGE	UGE	DGE	UGE	DGE	UGE	DGE	UGE	DGE	UGE	DGE	UGE
BN	0.74	0.82	0.93	0.89	0.78	0.86	0.88	0.78	0.70	0.79	0.77	0.74
GGM	0.77	0.89	0.76	0.87	0.74	0.84	0.72	0.82	0.79	0.93	0.72	0.81
RN	0.49	0.49	0.53	0.54	0.54	0.55	0.55	0.57	0.77	0.88	0.57	0.60

Table 2. Separation Scores. The separation score is defined as $S = T/(T + F)$, where T is the average score of a true edge, and F is the average score of a false edge. The perfect separation score of $S = 1$ is obtained when assigning a zero score to all false edges. Conversely, a method that cannot distinguish between true and false edges leads to an average separation score of $S = 0.5$. The abbreviations *Obs* and *Int* refer to observational and interventional data, respectively. The numbers are averages over all simulations carried out in the indicated category.

As it takes 'some time' until a Markov chain converges to its stationary distribution, one has to sample from the chain for "long enough" to ensure that it has reached its stationary distribution. After the *burn-in time* the orderings can be seen as sampled from the posterior distribution $P(O|D)$.

As mentioned before, the idea of Order-MCMC is to use this sample or orderings to obtain a sample of DAGs. To this end for each sampled ordering O_i a DAG G_i is sampled out of the posterior distribution $P(G|O_i, D)$, that is the posterior distribution over DAGs given the ordering O_i and the data D. Thereby, as conditioned on the ordering, for each network node its parent set can be sampled independently with respect to its valid parent-sets in the ordering O_i.

To ensure convergence we have performed each Order-MCMC run *independently* twice with two different random node orderings as initialisations. Thereby the burn-in lengths were set to 20,000, and afterwards further 80,000 Order-MCMC simulation steps were performed, whereby from each 200-th node ordering a directed acyclic graph (DAG) was sampled, leading to a graph sample of size 400 per run. We used the results of both Order-MCMC runs not only to assert convergence, we also averaged the outputed edge posterior probabilities of both runs. Consequently, all Bayesian network results are based on a directed acyclic graph sample of size 800 sampled from two independent Order-MCMC runs. It turned out that this run length setting led to a sufficient degree of convergence for all test data sets, that is there has never been a remarkable difference in the outputs of the two independent Order-MCMC runs.

5 PREPROCESSING OF REAL CYTOMETRIC DATA

For interventions we occasionally observed a clear discrepancy between expected and observed concentrations for intervened nodes, eg. some inhibitions hadn't led to low concentrations while some activations hadn't led to high concentrations. The missing changes in concentrations are not surprising, as most of the experimental interventions affected the activity of its target instead of its concentration. Correspondingly, for intervened nodes the measured concentrations do not reflect the strength of the true activity of the corresponding node (Karen Sachs, personal communication). Therefore, we decided to replace in each real interventional cytometric data set the values of the activated (inhibited) nodes by the maximal (minimal) concentration of that node measured under observational conditions. Afterwards, we have used quantile-normalisation to normalise each real interventional data set. That is for each of the 11 variables (proteins) we replaced its 100 realisations by quantiles of the standard normal distribution $N(0, 1)$. More precisely, for each of the 11 variables (proteins) its j-th highest realisation was replaced by the $\left(\frac{j}{100}\right)$-quantile of the standard normal distribution, whereby the ranks of identical realisations were averaged.

6 THE V-STRUCTURE NETWORK

In this study we always compare the methods for the same network topology. In order to have an idea about the influence of the network topology we slightly modified it. With these modifications we have added v-structures in the network. Four edges were removed from the original topology and we have then 4 new v-structures in this new network. This so called v-structure network is presented along with the original in figure 1. For this new network structure, we generated the same data sets as for the original network. We have generated 5 Gaussian data sets with observations only and 5 with interventions. For Netbuilder we generated 5 data sets with observations and 5 with interventions for each one of the three noise levels. All sets contain 100 data points.

7 OVERVIEW OF THE COMPLETE SET OF RESULTS

Our supplementary material contains the complete set of results obtained in our simulation studies, with a statistical significance estimation on a per-comparison basis. We have not corrected these values for multiple testing. This is because our tables contain many 'uninteresting' comparisons that were not followed up in the main paper, and are included only for the sake of completeness. More importantly, given the abundance of cytometry data and the fact that Netbuilder data are generated synthetically, it is straightforward to retest our hypotheses on new but independently generated data. We found that such a regeneration confirms the trends and significance levels reported in the main paper; this becomes evident by comparing Figure 4 in the main paper with Figure 4 in the supplementary material.

8 CROSS-METHOD COMPARISON OF AUC SCORES

This section of the supplementary material provides nine tables, numbered from 3 to 11, which summarise and cross-compare the performances of the three Machine Learning methods under comparison in terms of the outputed AUC scores. Thereby for each table multiple rows indicate the four combinations of figure of merit (UGE and DGE) and data set type (observational and interventional). For each of these four combinations and for each of the three methods (Bayesian networks (BN), Gaussian graphical models (GGM), and Relevance Networks (RN)) the mean μ[AUC] and the the standard deviations σ(AUC) of the five outputed AUC scores can be found. The last three columns provide one-sample t-test p-values p(.) for the hypothesis: H_0: $\mu[\text{AUC}(M_i)] = \mu[\text{AUC}(M_j)]$ against its two-sided alternative: H_1: $\mu[\text{AUC}(M_i)] \neq \mu[\text{AUC}(M_j)]$ given the combination indicated in the multiple row above. M_i and M_j represent the methods mentioned in the row and column. Low p-values p(.) indicate that there may be a significant difference in the AUC score between these two methods for the particular combination of figure of merit and data set type. In these cases it can be seen from the entries in the mean score column μ[AUC] which of the two methods performed (significantly) better than the other one.

Method	μ[AUC]	σ(AUC)	p(BN)	p(GGM)	p(RN)
UGE - Observational					
BN	0.8848	0.0543	-	0.8815	0.0079
GGM	0.8814	0.0373	0.8815	-	0.0015
RN	0.6809	0.0816	0.0079	0.0015	-
DGE - Observational					
BN	0.7817	0.0711	-	0.6704	0.0239
GGM	0.7967	0.0286	0.6704	-	0.0015
RN	0.6407	0.0635	0.0239	0.0015	-
UGE - Interventional					
BN	0.9661	0.0391	-	0.0024	0.0018
GGM	0.8203	0.0532	0.0024	-	0.0082
RN	0.7097	0.0541	0.0018	0.0082	-
DGE - Interventional					
BN	0.9796	0.0187	-	0.0002	0.0002
GGM	0.7488	0.0409	0.0002	-	0.0081
RN	0.6631	0.0421	0.0002	0.0081	-

Table 3. AUC score. Cross method comparison Gaussian data sets. Original graph topology.

Method	μ[AUC]	σ(AUC)	p(BN)	p(GGM)	p(RN)
UGE - Observational					
BN	0.9775	0.0345	-	0.0087	0.0013
GGM	0.8933	0.0583	0.0087	-	0.0043
RN	0.6987	0.0981	0.0013	0.0043	-
DGE - Observational					
BN	0.9487	0.0440	-	0.0012	0.0004
GGM	0.8257	0.0487	0.0012	-	0.0043
RN	0.6649	0.0814	0.0004	0.0043	-
UGE - Interventional					
BN	1.000	0.0000	-	0.0010	0.0014
GGM	0.8878	0.0293	0.0010	-	0.0199
RN	0.7436	0.0730	0.0014	0.0199	-
DGE - Interventional					
BN	0.9976	0.0038	-	0.0001	0.0004
GGM	0.8220	0.0001	0.0001	-	0.0196
RN	0.7021	0.0004	0.0004	0.0196	-

Table 4. AUC score. Cross method comparison Gaussian data sets. V-structure graph topology.

Method	μ[AUC]	σ(AUC)	p(BN)	p(GGM)	p(RN)
UGE - Observational					
BN	0.6904	0.0376	-	0.8754	0.2957
GGM	0.6854	0.0542	0.8754	-	0.6175
RN	0.6680	0.0546	0.2957	0.6175	-
DGE - Observational					
BN	0.6231	0.0564	-	0.5316	0.7276
GGM	0.6443	0.0419	0.5316	-	0.6139
RN	0.6307	0.0425	0.7276	0.6139	-
UGE - Interventional					
BN	0.7912	0.0335	-	0.0552	0.0003
GGM	0.7129	0.0559	0.0552	-	0.0010
RN	0.5686	0.0286	0.0003	0.0010	-
DGE - Interventional					
BN	0.6969	0.0676	-	0.4802	0.0076
GGM	0.6656	0.0437	0.4802	-	0.0010
RN	0.5533	0.0222	0.0076	0.0010	-

Table 5. AUC score. Cross method comparison Real cytoflow data sets.

Method	μ[AUC]	σ(AUC)	p(BN)	p(GGM)	p(RN)
UGE - Observational					
BN	0.7901	0.0336	-	0.0764	0.0444
GGM	0.8143	0.0191	0.0764	-	0.0009
RN	0.7434	0.0081	0.0444	0.0009	-
DGE - Observational					
BN	0.6808	0.0703	-	0.0669	0.7977
GGM	0.7446	0.0150	0.0669	-	0.0010
RN	0.6893	0.0063	0.7977	0.0010	-
UGE - Interventional					
BN	0.7047	0.0221	-	0.0675	0.0076
GGM	0.7297	0.0183	0.0675	-	0.0410
RN	0.7537	0.0063	0.0076	0.0410	-
DGE - Interventional					
BN	0.8280	0.0097	-	0.0001	0.0000
GGM	0.6793	0.0144	0.0001	-	0.0468
RN	0.6973	0.0049	0.0000	0.0468	-

Table 6. AUC score. Cross method comparison Nebuilder data sets low noise level($\sigma = 0.01$). Original graph topology.

Method	μ[AUC]	σ(AUC)	p(BN)	p(GGM)	p(RN)
UGE - Observational					
BN	0.9564	0.0273	-	0.0247	0.0469
GGM	0.8803	0.0656	0.0247	-	0.0909
RN	0.9323	0.0188	0.0469	0.0909	-
DGE - Observational					
BN	0.8572	0.0100	-	0.0288	0.0116
GGM	0.7957	0.0508	0.0288	-	0.0891
RN	0.8362	0.0146	0.0116	0.0891	-
UGE - Interventional					
BN	0.9346	0.0254	-	0.0188	0.0006
GGM	0.8300	0.0438	0.0188	-	0.1466
RN	0.8003	0.0082	0.0006	0.1466	-
DGE - Interventional					
BN	0.9678	0.0114	-	0.0004	0.0000
GGM	0.7574	0.0339	0.0004	-	0.1359
RN	0.7336	0.0064	0.0000	0.1359	-

Table 7. AUC score. Cross method comparison Nebuilder data sets medium noise level($\sigma = 0.1$). Original graph topology.

Method	μ[AUC]	σ(AUC)	p(BN)	p(GGM)	p(RN)
UGE - Observational					
BN	0.9049	0.0150	-	0.2776	0.1310
GGM	0.8829	0.0486	0.2776	-	0.0750
RN	0.9163	0.0179	0.1310	0.0750	-
DGE - Observational					
BN	0.8208	0.0223	-	0.3024	0.8234
GGM	0.7979	0.0381	0.3024	-	0.0782
RN	0.8238	0.0139	0.8234	0.0782	-
UGE - Interventional					
BN	0.9053	0.0367	-	0.0168	0.0329
GGM	0.8571	0.0251	0.0168	-	0.7139
RN	0.8631	0.0273	0.0329	0.7139	-
DGE - Interventional					
BN	0.9219	0.0408	-	0.0013	0.0007
GGM	0.7776	0.0230	0.0013	-	0.7051
RN	0.7824	0.0212	0.0007	0.7051	-

Table 8. AUC score. Cross method comparison Nebuilder data sets high noise level($\sigma = 0.3$). Original graph topology.

Method	μ[AUC]	σ(AUC)	p(BN)	p(GGM)	p(RN)
UGE - Observational					
BN	0.7845	0.0184	-	0.0055	0.0018
GGM	0.8529	0.0139	0.0055	-	0.0000
RN	0.7170	0.0094	0.0018	0.0000	-
DGE - Observational					
BN	0.7354	0.0467	-	0.0748	0.0558
GGM	0.7927	0.0117	0.0748	-	0.0000
RN	0.6801	0.0078	0.0558	0.0000	-
UGE - Interventional					
BN	0.7102	0.0156	-	0.0008	0.3208
GGM	0.7900	0.0180	0.0008	-	0.0110
RN	0.7280	0.0279	0.3208	0.0110	-
DGE - Interventional					
BN	0.8413	0.0052	-	0.0000	0.0001
GGM	0.7258	0.0143	0.0000	-	0.0115
RN	0.6773	0.0217	0.0001	0.0115	-

Table 9. AUC score. Cross method comparison Nebuilder data sets low noise level($\sigma = 0.01$). V-structure graph topology.

Method	μ[AUC]	σ(AUC)	p(BN)	p(GGM)	p(RN)
UGE - Observational					
BN	0.9887	0.0114	-	0.0259	0.0002
GGM	0.9567	0.0294	0.0259	-	0.0024
RN	0.8513	0.0188	0.0002	0.0024	-
DGE - Observational					
BN	0.9674	0.0124	-	0.0002	0.0000
GGM	0.8788	0.0244	0.0002	-	0.0025
RN	0.7915	0.0156	0.0000	0.0025	-
UGE - Interventional					
BN	0.9927	0.0085	-	0.0019	0.0000
GGM	0.8277	0.0565	0.0019	-	0.0395
RN	0.7483	0.0257	0.0000	0.0395	-
DGE - Interventional					
BN	0.9944	0.0040	-	0.0002	0.0000
GGM	0.7547	0.0436	0.0002	-	0.0390
RN	0.6931	0.0200	0.0000	0.0390	-

Table 10. AUC score. Cross method comparison Nebuilder data sets medium noise level($\sigma = 0.1$). V-structure graph topology.

Method	μ[AUC]	σ(AUC)	p(BN)	p(GGM)	p(RN)
UGE - Observational					
BN	0.9332	0.0454	-	0.0437	0.0020
GGM	0.9038	0.0562	0.0437	-	0.2289
RN	0.9154	0.0460	0.0020	0.2289	-
DGE - Observational					
BN	0.8745	0.0452	-	0.0888	0.0931
GGM	0.8350	0.0466	0.0888	-	0.2135
RN	0.8447	0.0381	0.0931	0.2135	-
UGE - Interventional					
BN	0.9788	0.0090	-	0.0163	0.0018
GGM	0.8677	0.0630	0.0163	-	0.2972
RN	0.8214	0.0474	0.0018	0.2972	-
DGE - Interventional					
BN	0.9393	0.0406	-	0.0047	0.0013
GGM	0.7861	0.0489	0.0047	-	0.2943
RN	0.7500	0.0368	0.0013	0.2943	-

Table 11. AUC score. Cross method comparison Nebuilder data sets high noise level($\sigma = 0.3$). V-structure graph topology.

9 CROSS-METHOD COMPARISON OF TRUE POSITIVE COUNTS

This section of the supplementary material provides nine tables, numbered from 12 to 20, which summarise and cross-compare the performances of the three Machine Learning methods under comparison in terms of the true positive counts TP obtained when accepting 5 false positive counts (FP=5). Thereby in analogy to the last section for each table multiple rows indicate the four combinations of figure of merit (UGE and DGE) and data set type (observational and interventional). For each of these four combinations and for each of the three methods (Bayesian networks (BN), Gaussian graphical models (GGM), and Relevance Networks (RN)) the mean μ[TP] and the the standard deviations σ(TP) of the five true positive counts TP obtained for 5 false positive counts can be found in the first columns. The last three columns provide one-sample t-test p-values p(.) for the hypothesis: H_0: $\mu[\mathrm{TP}(M_i)] = \mu[\mathrm{TP}(M_j)]$ against its two-sided alternative: H_1: $\mu[\mathrm{TP}(M_i)] \neq \mu[\mathrm{TP}(M_j)]$ given the combination indicated in the multiple row above. M_i and M_j represent the methods mentioned in the row and column. Low p-values p(.) indicate that there may be a significant difference in the TP counts between these two methods for the particular combination of figure of merit and data set type. In these cases it can be seen from the entries in the mean score column μ[TP] which of the two methods performed (significantly) better than the other one. In contrast to the AUC score cross-method comparison this alternative true positive count cross-method comparison concentrates on a fixed inverse specificity point of the ROC curve.

Method	μ[TP]	σ(TP)	p(BN)	p(GGM)	p(RN)
UGE - Observational					
BN	15.8	2.1	-	0.1662	0.0010
GGM	14.8	2.7	0.1662	-	0.0012
RN	8.1	2.5	0.0010	0.0012	-
DGE - Observational					
BN	4.9	1.5	-	0.6885	0.0042
GGM	4.7	1.1	0.6885	-	0.0705
RN	3.8	1.3	0.0042	0.0705	-
UGE - Interventional					
BN	18.5	2.4	-	0.0074	0.0028
GGM	13.2	2.0	0.0074	-	0.0011
RN	6.5	2.7	0.0028	0.0011	-
DGE - Interventional					
BN	18.4	2.6	-	0.0005	0.0005
GGM	5.2	0.7	0.0005	-	0.0036
RN	1.8	1.3	0.0005	0.0036	-

Table 12. TP counts score. Cross method comparison Gaussian data sets. Original graph topology.

Method	μ[TP]	σ(TP)	p(BN)	p(GGM)	p(RN)
UGE - Observational					
BN	15.6	0.5	-	0.0270	0.0000
GGM	11.8	2.7	0.0270	-	0.0024
RN	5.8	1.4	0.0000	0.0024	-
DGE - Observational					
BN	11.3	1.2	-	0.0000	0.0001
GGM	3.8	1.0	0.0000	-	0.1951
RN	3.0	0.6	0.0001	0.1951	-
UGE - Interventional					
BN	16.0	0.0	-	0.0025	0.0001
GGM	12.9	1.0	0.0025	-	0.0054
RN	7.1	1.3	0.0001	0.0054	-
DGE - Interventional					
BN	15.8	0.4	-	0.0000	0.0000
GGM	5.5	0.0	0.0000	-	0.0008
RN	3.7	0.4	0.0000	0.0008	-

Table 13. TP counts score. Cross method comparison Gaussian data sets. V-structure graph topology.

Method	μ[TP]	σ(TP)	p(BN)	p(GGM)	p(RN)
UGE - Observational					
BN	9.5	2.0	-	0.7489	0.7174
GGM	9.6	1.6	0.7489	-	0.3046
RN	9.3	1.6	0.7174	0.3046	-
DGE - Observational					
BN	3.3	2.3	-	0.1369	0.1369
GGM	5.1	0.9	0.1369	-	NaN
RN	5.1	0.9	0.1369	NaN	-
UGE - Interventional					
BN	11.1	1.3	-	0.0951	0.0099
GGM	9.6	1.1	0.0951	-	0.0204
RN	7.1	1.1	0.0099	0.0204	-
DGE - Interventional					
BN	6.9	1.1	-	0.0065	0.0009
GGM	4.1	1.1	0.0065	-	0.0093
RN	1.7	0.4	0.0009	0.0093	-

Table 14. TP counts score. Cross method comparison Real cytoflow data sets.

Method	μ[TP]	σ(TP)	p(BN)	p(GGM)	p(RN)
UGE - Observational					
BN	11.0	2.0	-	0.2577	0.0366
GGM	12.0	1.2	0.2577	-	0.0040
RN	6.9	1.4	0.0366	0.0040	-
DGE - Observational					
BN	2.8	1.3	-	0.0077	0.0890
GGM	5.1	0.7	0.0077	-	0.0016
RN	0.8	0.8	0.0890	0.0016	-
UGE - Interventional					
BN	7.9	0.7	-	0.0008	0.0000
GGM	5.2	0.3	0.0008	-	0.0000
RN	2.0	0.0	0.0000	0.0000	-
DGE - Interventional					
BN	8.4	1.2	-	0.0019	0.0001
GGM	3.7	0.4	0.0019	-	0.0001
RN	0.0	0.0	0.0001	0.0001	-

Table 15. TP counts score. Cross method comparison. Nebuilder data sets low noise level($\sigma = 0.01$). Original graph topology.

Method	μ[TP]	σ(TP)	p(BN)	p(GGM)	p(RN)
UGE - Observational					
BN	18.1	1.1	-	0.0161	0.1216
GGM	16.5	1.1	0.0161	-	0.5291
RN	16.8	1.6	0.1216	0.5291	-
DGE - Observational					
BN	7.2	1.5	-	0.0673	0.0673
GGM	5.5	0.0	0.0673	-	NaN
RN	5.5	0.0	0.0673	NaN	-
UGE - Interventional					
BN	17.7	0.7	-	0.0046	0.0003
GGM	13.6	1.5	0.0046	-	0.0002
RN	8.0	1.7	0.0003	0.0002	-
DGE - Interventional					
BN	17.3	0.7	-	0.0000	0.0000
GGM	5.4	0.2	0.0000	-	0.0000
RN	1.2	0.7	0.0000	0.0000	-

Table 16. TP counts score. Cross method comparison. Nebuilder data sets medium noise level($\sigma = 0.1$). Original graph topology.

Method	μ[TP]	σ(TP)	p(BN)	p(GGM)	p(RN)
UGE - Observational					
BN	15.5	1.7	-	0.4468	0.2756
GGM	14.8	2.9	0.4468	-	0.0213
RN	16.6	2.3	0.2756	0.0213	-
DGE - Observational					
BN	4.1	2.0	-	0.5158	0.3844
GGM	4.7	1.1	0.5158	-	0.3739
RN	5.1	0.9	0.3844	0.3739	-
UGE - Interventional					
BN	16.0	1.6	-	0.0890	0.0143
GGM	14.5	1.5	0.0890	-	0.3672
RN	13.6	1.5	0.0143	0.3672	-
DGE - Interventional					
BN	14.1	4.5	-	0.0052	0.0073
GGM	5.5	0.0	0.0052	-	0.3739
RN	5.0	1.1	0.0073	0.3739	-

Table 17. TP counts score. Cross method comparison. Nebuilder data sets high noise level($\sigma = 0.3$). Original graph topology.

Method	μ[TP]	σ(TP)	p(BN)	p(GGM)	p(RN)
UGE - Observational					
BN	9.8	0.8	-	1.0000	0.0007
GGM	9.8	0.8	1.0000	-	0.0001
RN	5.2	0.3	0.0007	0.0001	-
DGE - Observational					
BN	3.9	0.7	-	0.3419	0.0019
GGM	4.5	0.9	0.3419	-	0.0020
RN	1.5	0.0	0.0019	0.0020	-
UGE - Interventional					
BN	7.2	0.4	-	0.4263	0.0102
GGM	7.5	0.4	0.4263	-	0.0078
RN	5.1	0.9	0.0102	0.0078	-
DGE - Interventional					
BN	6.6	0.4	-	0.0001	0.0000
GGM	3.4	0.2	0.0001	-	0.0002
RN	2.0	0.0	0.0000	0.0002	-

Table 18. TP counts score. Cross method comparison. Nebuilder data sets low noise level($\sigma = 0.01$). V-structure graph topology.

Method	μ[TP]	σ(TP)	p(BN)	p(GGM)	p(RN)
UGE - Observational					
BN	15.6	0.4	-	0.0876	0.0000
GGM	14.7	1.0	0.0876	-	0.0001
RN	9.3	0.4	0.0000	0.0001	-
DGE - Observational					
BN	12.0	1.5	-	0.0006	0.0007
GGM	5.5	0.0	0.0006	-	0.1079
RN	4.8	0.8	0.0007	0.1079	-
UGE - Interventional					
BN	15.7	0.4	-	0.0002	0.0005
GGM	12.5	0.5	0.0002	-	0.0021
RN	8.9	1.0	0.0005	0.0021	-
DGE - Interventional					
BN	15.4	0.7	-	0.0000	0.0000
GGM	4.9	0.8	0.0000	-	0.1302
RN	3.8	1.0	0.0000	0.1302	-

Table 19. TP counts score. Cross method comparison. Nebuilder data sets medium noise level($\sigma = 0.1$). V-structure graph topology.

Method	μ[TP]	σ(TP)	p(BN)	p(GGM)	p(RN)
UGE - Observational					
BN	14.2	1.1	-	0.0474	0.1087
GGM	13.2	1.0	0.0474	-	0.3375
RN	13.6	0.8	0.1087	0.3375	-
DGE - Observational					
BN	7.7	2.0	-	0.0714	0.0440
GGM	5.5	0.0	0.0714	-	0.2663
RN	5.0	0.9	0.0440	0.2663	-
UGE - Interventional					
BN	14.9	0.2	-	0.0093	0.0277
GGM	12.5	1.1	0.0093	-	0.5913
RN	12.8	1.4	0.0277	0.5913	-
DGE - Interventional					
BN	12.3	1.7	-	0.0009	0.0009
GGM	5.5	0	0.0009	-	NA
RN	5.5	0	0.0009	NA	-

Table 20. TP counts score. Cross method comparison. Nebuilder data sets high noise level($\sigma = 0.3$). V-structure graph topology.

10 COMPARISON BETWEEN PERFORMANCE ON OBSERVATIONAL AND INTERVENTIONAL DATA

This section of the supplementary material provides nine tables, numbered from 21 to 29, which compare the performance of each Machine Learning method on pure observational data with its performance on interventional data sets in terms of the outputed AUC scores. Thereby for each of the nine tables multiple rows indicate the two different figures of merit (UGE and DGE). For each figure of merit and for each of the three methods (Bayesian networks (BN), Gaussian graphical models (GGM), and Relevance Networks (RN)) the mean of the five AUC scores on pure observational data μ[AUC—OBS] as well as the mean of the five AUC scores on interventional data μ[AUC—INT] are given in the first two columns. Subsequently the hypothesis H_0: μ[AUC—OBS] = μ[AUC—INT] was tested against its two-sided alternative H_1: μ[AUC—OBS] $\neq$ μ[AUC—INT] using two-sample t-tests. The p-values p(.) of these tests can be found in the last column. Low p-values p(.) indicate that there may be a significant difference in the mean AUC score obtained for pure observational and interventional data sets for the method mentioned in the row and the particular figure of merit. In these cases it can be seen from the entries in the mean score columns μ[AUC—OBS] and μ[AUC—INT] on which data set type the method performed (significantly) better.

Method	μ[AUC—OBS]	μ[AUC—INT]	p(.)
UGE			
BN	0.8848	0.9661	0.0264
GGM	0.8814	0.8203	0.0683
RN	0.6809	0.7097	0.5285
DGE			
BN	0.7817	0.9796	0.0003
GGM	0.7967	0.7488	0.0643
RN	0.6407	0.6631	0.5285

Table 21. AUC score. Observational versus Interventional data. Gaussian data sets. Original graph topology.

Method	μ[AUC—OBS]	μ[AUC—INT]	p(.)
UGE			
BN	0.9775	1.0000	0.1822
GGM	0.8933	0.8878	0.8565
RN	0.6987	0.7436	0.4355
DGE			
BN	0.9487	0.9976	0.0384
GGM	0.8257	0.8220	0.8820
RN	0.6649	0.7021	0.4355

Table 22. AUC score. Observational versus Interventional data. Gaussian data sets. V-structure graph topology.

Method	μ[AUC—OBS]	μ[AUC—INT]	p(.)
UGE			
BN	0.6904	0.7912	0.0021
GGM	0.6854	0.7129	0.4534
RN	0.6680	0.5686	0.0069
DGE			
BN	0.6231	0.6969	0.0974
GGM	0.6443	0.6656	0.4532
RN	0.6307	0.5533	0.0069

Table 23. AUC score. Observational versus Interventional data. Real cytoflow data sets.

Method	μ[AUC—OBS]	μ[AUC—INT]	p(.)
UGE			
BN	0.7901	0.7047	0.0014
GGM	0.8143	0.7297	0.0001
RN	0.7434	0.7537	0.0553
DGE			
BN	0.6808	0.8280	0.0017
GGM	0.7446	0.6793	0.0001
RN	0.6893	0.6973	0.0553

Table 24. AUC score. Observational versus Interventional data. Netbuilder data sets low noise level ($\sigma = 0.01$). Original graph topology.

Method	μ[AUC—OBS]	μ[AUC—INT]	p(.)
UGE			
BN	0.9464	0.9346	0.4974
GGM	0.8803	0.8300	0.1918
RN	0.9323	0.8003	0.0000
DGE			
BN	0.8572	0.9678	0.0000
GGM	0.7957	0.7574	0.1979
RN	0.8362	0.7336	0.0000

Table 25. AUC score. Observational versus Interventional data. Netbuilder data sets medium noise level ($\sigma = 0.1$). Original graph topology.

Method	μ[AUC—OBS]	μ[AUC—INT]	p(.)
UGE			
BN	0.9049	0.9053	0.9813
GGM	0.8829	0.8571	0.3242
RN	0.9163	0.8631	0.0066
DGE			
BN	0.8208	0.9219	0.0013
GGM	0.7979	0.7776	0.3228
RN	0.8238	0.7824	0.0066

Table 26. AUC score. Observational versus Interventional data. Netbuilder data sets high noise level ($\sigma = 0.3$). Original graph topology.

Method	μ[AUC—OBS]	μ[AUC—INT]	p(.)
UGE			
BN	0.7845	0.7102	0.0001
GGM	0.8529	0.7900	0.0003
RN	0.7170	0.7280	0.4271
DGE			
BN	0.7354	0.8413	0.0010
GGM	0.7927	0.7258	0.0000
RN	0.6801	0.6773	0.7986

Table 27. AUC score. Observational versus Interventional data. Netbuilder data sets low noise level ($\sigma = 0.01$). V-structure graph topology.

Method	μ[AUC—OBS]	μ[AUC—INT]	p(.)
UGE			
BN	0.9887	0.9927	0.5464
GGM	0.9567	0.8277	0.0019
RN	0.8513	0.7483	0.0001
DGE			
BN	0.9674	0.9944	0.0017
GGM	0.8788	0.7547.	0.0005
RN	0.7915	0.6931	0.0000

Table 28. AUC score. Observational versus Interventional data. Netbuilder data sets medium noise level ($\sigma = 0.1$). V-structure graph topology.

Method	μ[AUC—OBS]	μ[AUC—INT]	p(.)
UGE			
BN	0.9332	0.9788	0.0583
GGM	0.9038	0.8677	0.3669
RN	0.9154	0.8214	0.0129
DGE			
BN	0.8745	0.9393	0.0443
GGM	0.8350	0.7861	0.1442
RN	0.8447	0.7500	0.0040

Table 29. AUC score. Observational versus Interventional data. Netbuilder data sets high noise level ($\sigma = 0.3$). V-structure graph topology.

11 COMPARISON BETWEEN PERFORMANCE ON OBSERVATIONAL AND INTERVENTIONAL DATA USING TP COUNTS

This section of the supplementary material provides nine tables, numbered from 30 to 38, which compare the performance of each Machine Learning method on pure observational data with its performance on interventional data sets in terms of the obtained true positive counts (TP) when accepting five false negative counts (FP=5). As in the last section for each of the nine tables multiple rows indicate the two different figures of merit (UGE and DGE). For each figure of merit and for each of the three methods (Bayesian networks (BN), Gaussian graphical models (GGM), and Relevance Networks (RN)) the mean of the five TP counts on pure observational data μ[TP—OBS] as well as the mean of the five TP counts on interventional data μ[TP—INT] are given in the first two columns. Subsequently the hypothesis H_0: μ[TP—OBS] = μ[TP—INT] was tested against its two-sided alternative H_1: μ[TP—OBS] $\neq$ μ[TP—INT] using two-sample t-tests. The p-values p(.) of these tests can be found in the last column. Low p-values p(.) indicate that there may be a significant difference in the mean TP count outputed for pure observational and interventional data sets for the method mentioned in the row and the particular figure of merit. In these cases it can be seen from the entries in the mean TP count columns μ[TP—OBS] and μ[TP—INT] on which data set type the method performed (significantly) better.

Method	μ[TP—OBS]	μ[TP—INT]	p(.)
UGE			
BN	15.8	18.5	0.0971
GGM	14.8	13.2	0.3152
RN	8.1	6.5	0.3553
DGE			
BN	4.9	18.4	0.0000
GGM	4.7	5.2	0.4094
RN	3.8	1.8	0.0386

Table 30. TP counts score. Observational versus Interventional data. Gaussian data sets. Original graph topology.

Method	μ[TP—OBS]	μ[TP—INT]	p(.)
UGE			
BN	15.6	16.0	0.1411
GGM	11.8	12.9	0.4167
RN	5.8	7.1	0.1780
DGE			
BN	11.3	15.8	0.0001
GGM	3.8	5.5	0.0045
RN	3.0	3.7	0.0729

Table 31. TP counts score. Observational versus Interventional data. Gaussian data sets. V-structure graph topology.

Method	μ[TP—OBS]	μ[TP—INT]	p(.)
UGE			
BN	9.5	11.1	0.1757
GGM	9.6	9.6	1.0000
RN	9.3	7.1	0.0347
DGE			
BN	3.3	6.9	0.0134
GGM	5.1	4.1	0.1502
RN	5.1	1.7	0.0001

Table 32. TP counts score. Observational versus Interventional data. Real cytoflow data sets.

Method	μ[TP—OBS]	μ[TP—INT]	p(.)
UGE			
BN	11.0	7.9	0.0102
GGM	12.0	5.2	0.0000
RN	6.9	2.0	0.0000
DGE			
BN	2.8	8.4	0.0001
GGM	5.1	3.7	0.0042
RN	0.8	0.0	0.0650

Table 33. TP counts score. Observational versus Interventional data. Netbuilder data sets low noise level ($\sigma = 0.01$). Original graph topology.

Method	μ[TP—OBS]	μ[TP—INT]	p(.)
UGE			
BN	18.1	17.7	0.5180
GGM	16.5	13.6	0.0088
RN	16.8	8.0	0.0000
DGE			
BN	7.2	17.3	0.0000
GGM	5.5	5.4	0.3466
RN	5.5	1.2	0.0000

Table 34. TP counts score. Observational versus Interventional data. Netbuilder data sets medium noise level ($\sigma = 0.1$). Original graph topology.

Method	μ[TP—OBS]	μ[TP—INT]	p(.)
UGE			
BN	15.5	16.0	0.6463
GGM	14.8	14.5	0.8408
RN	16.6	13.6	0.0397
DGE			
BN	4.1	14.1	0.0005
GGM	4.7	5.5	0.1411
RN	5.1	5.0	0.8798

Table 35. TP counts score. Observational versus Interventional data. Netbuilder data sets high noise level ($\sigma = 0.3$). Original graph topology.

Method	μ[TP—OBS]	μ[TP—INT]	p(.)
UGE			
BN	9.8	7.2	0.0003
GGM	9.8	7.5	0.0003
RN	5.2	5.1	0.8171
DGE			
BN	3.9	5.6	0.0001
GGM	4.5	3.4	0.0338
RN	1.5	2.0	NA

Table 36. TP counts score. Observational versus Interventional data. Netbuilder data sets low noise level ($\sigma = 0.01$). V-structure graph topology.

Method	μ[TP—OBS]	μ[TP—INT]	p(.)
UGE			
BN	15.6	15.7	0.7245
GGM	14.7	12.5	0.0020
RN	9.3	8.9	0.4468
DGE			
BN	12.0	15.4	0.0016
GGM	5.5	4.9	0.1411
RN	4.8	3.8	0.1078

Table 37. TP counts score. Observational versus Interventional data. Netbuilder data sets medium noise level ($\sigma = 0.1$). V-structure graph topology.

Method	μ[TP—OBS]	μ[TP—INT]	p(.)
UGE			
BN	14.2	14.9	0.1991
GGM	13.2	12.5	0.3347
RN	13.6	12.8	0.2907
DGE			
BN	7.7	12.3	0.0047
GGM	5.5	5.5	NA
RN	5.5	5.5	0.2328

Table 38. TP counts score. Observational versus Interventional data. Netbuilder data sets high noise level ($\sigma = 0.3$). V-structure graph topology.

12 COMPARISON OF PERFORMANCE ON ORIGINAL AND V-STRUCTURE NETWORK TOPOLOGY

This section of the supplementary material provides four tables, numbered from 39 to 42, with p-values we have used to compare the performance of Bayesian networks (BN), Gaussian graphical models (GGM), and Relevance networks (RN) on the original graph topology G_O and the v-structured graph topology G_V. We used this analysis for checking to which differences the inclusion of v-structures for the different methods leads. More precisely, we have been interested in answering the question whether the inclusion of v-structures leads to a larger improvement of the AUC scores for Bayesian networks than for the other two methods. To this end we have looked for each pair of methods M_i and M_j at the mean AUC score differences $AUC(M_i, G_O)$ - $AUC(M_j, G_O)$ and $AUC(M_i, G_V)$ - $AUC(M_j, G_V)$ to see whether the difference in performance alters between the two graph topologies. Then we computed the p-value of a two-sided two-sample t-test for the null hypothesis:

H_0: $\mu[AUC(M_i, G_O)\text{-}(M_j, G_O)] = \mu[AUC(M_i, G_V)\text{-}AUC(M_j, G_V)]$

against its two-sided alternative. Consequently, low p-values p(.) indicate that the mean AUC score differences alter on the two different graph topologies.

All four tables in this section have the same structure. After a row indicating the figure of merit (UGE or DGE) as well as the data set type (pure observational or interventional), there is one row for each of the three methods under comparison: Bayesian networks (BN), Gaussian graphical models (GGM), and Relevance networks (RN). In each of these rows the mean AUC scores for both directed acyclic graph topologies G_O and G_V as well as the p-values of the tests mentioned above can be found. Thereby the signs minus ('-') and plus ('+') have been added to the p-value entries to indicate whether for the method mentioned in the row the mean difference is higher for the graph topology with v-structures G_V ('+') or for the original graph G_O ('-'). So for example each plus sign ('+') indicates that the alteration of the differences introduced by v-structures is for the benefit of the method mentioned in the row.

Method	μ[AUC—G_O]	μ[AUC—G_V]	p(BN)	p(GGM)	p(RN)
UGE - Observational					
BN	0.8848	0.9775	-	+0.0187	+0.2018
GGM	0.8814	0.8933	-0.0187	-	+0.8900
RN	0.6809	0.6987	-0.2018	-0.8900	-
DGE - Observational					
BN	0.7817	0.9487	-	+0.0049	+0.0168
GGM	0.7967	0.8257	-0.0049	-	+0.8908
RN	0.6407	0.6649	-0.0168	-0.8908	-
UGE - Interventional					
BN	0.9661	1.0000	-	-0.2143	-0.9997
GGM	0.8203	0.8878	-0.2143	-	+0.4727
RN	0.7097	0.7436	-0.9997	+0.4727	-
DGE - Interventional					
BN	0.9796	0.9976	-	-0.0259	-0.5804
GGM	0.7488	0.8220	+0.0259	-	+0.3743
RN	0.6631	0.7021	+0.5804	-0.3743	-

Table 39. AUC score. Cross method differences between the original graph topology G_O and v-structure topology G_V. Gaussian data sets.

Method	μ[AUC—G_O]	μ[AUC—G_V]	p(BN)	p(GGM)	p(RN)
UGE - Observational					
BN	0.7901	0.7845	-	-0.0255	+0.2939
GGM	0.8143	0.8529	+0.0255	-	+0.0001
RN	0.7434	0.7170	-0.2939	-0.0001	-
DGE - Observational					
BN	0.6808	0.7354	-	+0.8580	+0.1256
GGM	0.7446	0.7927	-0.8580	-	+0.0000
RN	0.6893	0.6801	-0.1256	-0.0000	-
UGE - Interventional					
BN	0.7047	0.7102	-	-0.0034	+0.1316
GGM	0.7297	0.7900	+0.0034	-	+0.0007
RN	0.7537	0.7280	-0.1316	-0.0007	-
DGE - Interventional					
BN	0.8280	0.8413	-	-0.0111	+0.0183
GGM	0.6793	0.7258	+0.0111	-	+0.0008
RN	0.6973	0.6773	-0.0183	-0.0008	-

Table 40. AUC score. Cross method differences between the original graph topology G_O and v-structure topology G_V. Netbuilder data sets low noise level ($\sigma = 0.01$).

Method	μ[AUC—G_O]	μ[AUC—G_V]	p(BN)	p(GGM)	p(RN)
UGE - Observational					
BN	0.9464	0.9887	-	-0.1423	+0.0000
GGM	0.8803	0.9567	+0.1423	-	+0.0005
RN	0.9323	0.8513	-0.0000	-0.0005	-
DGE - Observational					
BN	0.8572	0.9674	-	+0.2027	+0.0000
GGM	0.7957	0.8788	-0.2027	-	+0.0004
RN	0.8362	0.7915	-0.0000	-0.0004	-
UGE - Interventional					
BN	0.9346	0.9927	-	+0.1280	+0.0004
GGM	0.8300	0.8277	-0.1280	-	+0.1488
RN	0.8003	0.7483	-0.0004	-0.1488	-
DGE - Interventional					
BN	0.9678	0.9944	-	+0.2979	+0.0005
GGM	0.7574	0.7547	-0.2979	-	+0.1553
RN	0.7336	0.6931	-0.0005	-0.1533	-

Table 41. AUC score. Cross method differences between the original graph topology G_O and v-structure topology G_V. Netbuilder data sets medium noise level ($\sigma = 0.1$).

Method	μ[AUC—G_O]	μ[AUC—G_V]	p(BN)	p(GGM)	p(RN)
UGE - Observational					
BN	0.9049	0.9332	-	+0.7264	+0.0021
GGM	0.8829	0.9038	-0.7264	-	+0.2127
RN	0.9163	0.9154	-0.0021	-0.2127	-
DGE - Observational					
BN	0.8208	0.8745	-	+0.5437	+0.1121
GGM	0.7979	0.8350	-0.5437	-	+0.2403
RN	0.8238	0.8447	-0.1121	-0.2403	-
UGE - Interventional					
BN	0.9053	0.9788	-	+0.0723	+0.0018
GGM	0.8571	0.8677	-0.0723	-	+0.2437
RN	0.8631	0.8214	-0.0018	-0.2437	-
DGE - Interventional					
BN	0.9219	0.9393	-	+0.7899	+0.1130
GGM	0.7776	0.7861	-0.7899	-	+0.2394
RN	0.7824	0.7500	-0.1130	-0.2394	-

Table 42. AUC score. Cross method differences between the original graph topology G_O and v-structure topology G_V. Netbuilder data sets high noise level ($\sigma = 0.3$).

13 COMPARISON OF PERFORMANCES ON ORIGINAL AND V-STRUCTURE NETWORK TOPOLOGY USING TRUE POSITIVE COUNTS

This section of the supplementary material provides four tables, numbered from 43 to 46, with p-values we have used to compare the performance of Bayesian networks (BN), Gaussian graphical models (GGM), and Relevance networks (RN) on the original graph topology G_O and the v-structured graph topology G_V. We used this analysis for checking to which differences the inclusion of v-structures for the different methods leads. More precisely, we have been interested in answering the question whether the inclusion of v-structures leads to a larger improvement of the sensitivity S outputed when accepting five false positive counts for Bayesian networks than for the other two methods. To this end we have looked for each pair of methods M_i and M_j at the mean sensitivity differences $S(M_i, G_O)$ - $S(M_j, G_O)$ and $S(M_i, G_V)$ - $S(M_j, G_V)$ to see whether the difference in performance alters between the two graph topologies. Then we computed the p-value of a two-sided two-sample t-test for the null hypothesis:
H_0: $\mu[S(M_i, G_O)\text{-}S(M_j, G_O)] = \mu[S(M_i, G_V)\text{-}AS(M_j, G_V)]$
against its two-sided alternative. Consequently, low p-values p(.) indicate that the mean sensitivities differences alter on the two different graph topologies.

All four tables in this section have the same structure. After a row indicating the figure of merit (UGE or DGE) as well as the data set type (pure observational or interventional), there is one row for each of the three methods under comparison: Bayesian networks (BN), Gaussian graphical models (GGM), and Relevance networks (RN). In each of these rows the mean sensitivity, when accepting five false positive counts, for both directed acyclic graph topologies G_O and G_V as well as the p-values p(.) of the tests mentioned above can be found. Thereby the signs minus ('-') and plus ('+') have been added to the p-value entries to indicate whether for the method mentioned in the row the mean difference is higher for the graph topology with v-structures G_V ('+') or for the original graph G_O ('-'). So for example each plus sign ('+') indicates that the alteration of the differences introduced by v-structures is for the benefit of the method mentioned in the row.

Method	μ[S—G$_O$]	μ[S—G$_V$]	p(BN)	p(GGM)	p(RN)
UGE - Observational					
BN	0.7900	0.9750	-	+0.0381	+0.0028
GGM	0.7400	0.7375	-0.0381	-	+0.5753
RN	0.4050	0.3625	-0.0028	-0.5753	-
DGE - Observational					
BN	0.2450	0.7063	-	+0.0000	+0.0000
GGM	0.2350	0.2375	-0.0000	-	+0.8960
RN	0.1900	0.1875	-0.0000	-0.8960	-
UGE - Interventional					
BN	0.9250	1.0000	-	-0.2696	-0.6690
GGM	0.6600	0.8063	+0.2696	-	+0.7308
RN	0.3250	0.4437	+0.6690	-0.7308	-
DGE - Interventional					
BN	0.9200	0.9875	-	-0.8122	-0.3898
GGM	0.2600	0.3438	+0.8122	-	-0.0963
RN	0.0900	0.2313	+0.3893	+0.0963	-

Table 43. TP counts score. Cross method differences between the original graph topology G_O and v-structure topology G_V. Gaussian data sets.

Method	μ[S—G$_O$]	μ[S—G$_V$]	p(BN)	p(GGM)	p(RN)
UGE - Observational					
BN	0.5500	0.6125	-	+0.3647	+0.2907
GGM	0.6000	0.6125	-0.3647	-	+0.5051
RN	0.3450	0.3250	-0.2907	-0.5051	-
DGE - Observational					
BN	0.1400	0.2437	-	+0.1009	+0.3401
GGM	0.2550	0.2813	-0.1009	-	-0.4938
RN	0.0400	0.0938	-0.3401	+0.4938	-
UGE - Interventional					
BN	0.3950	0.4500	-	-0.0004	-0.0009
GGM	0.2600	0.4688	+0.0004	-	-0.7546
RN	0.1000	0.3187	+0.0009	+0.7546	-
DGE - Interventional					
BN	0.4200	0.4125	-	-0.3407	-0.0019
GGM	0.1850	0.2125	+0.3407	-	-0.0000
RN	0.0000	0.1250	+0.0019	+0.0000	-

Table 44. TP counts score. Cross method differences between the original graph topology G_O and v-structure topology G_V. Netbuilder data sets low noise level ($\sigma = 0.01$).

Method	μ[S—G_O]	μ[S—G_V]	p(BN)	p(GGM)	p(RN)
UGE - Observational					
BN	0.9050	0.9750	-	-0.4794	+0.0000
GGM	0.8250	0.9187	+0.4794	-	+0.0000
RN	0.8400	0.5813	-0.0000	-0.0000	-
DGE - Observational					
BN	0.3600	0.7500	-	+0.0003	+0.0002
GGM	0.2750	0.3438	-0.0003	-	+0.0729
RN	0.2750	0.3000	-0.0002	-0.0729	-
UGE - Interventional					
BN	0.8850	0.9812	-	-0.9014	-0.3257
GGM	0.6800	0.7813	+0.9014	-	-0.1904
RN	0.4000	0.5563	+0.3257	+0.1904	-
DGE - Interventional					
BN	0.8650	0.9625	-	+0.0750	-0.0395
GGM	0.2700	0.3063	-0.0750	-	-0.0055
RN	0.0600	0.2375	+0.0395	+0.0055	-

Table 45. TP counts score. Cross method differences between the original graph topology G_O and v-structure topology G_V. Netbuilder data sets medium noise level ($\sigma = 0.1$).

Method	μ[S—G_O]	μ[S—G_V]	p(BN)	p(GGM)	p(RN)
UGE - Observational					
BN	0.7750	0.8875	-	+0.5750	+0.0859
GGM	0.7400	0.8250	-0.5750	-	+0.0889
RN	0.8300	0.8500	-0.0859	-0.0889	-
DGE - Observational					
BN	0.2050	0.4813	-	+0.0446	+0.0224
GGM	0.2350	0.3438	-0.0446	-	+0.1413
RN	0.2550	0.3125	-0.0224	-0.1413	-
UGE - Interventional					
BN	0.8000	0.9313	-	+0.1437	+0.8220
GGM	0.7250	0.7813	-0.1437	-	-0.2778
RN	0.6800	0.8000	-0.8220	+0.2778	-
DGE - Interventional					
BN	0.7050	0.7688	-	-0.9577	-0.7767
GGM	0.2750	0.3438	+0.9577	-	-0.3466
RN	0.2500	0.3438	+0.7767	+0.3466	-

Table 46. TP counts score. Cross method differences between the original graph topology G_O and v-structure topology G_V. Netbuilder data sets high noise level ($\sigma = 0.3$).

REFERENCES

Briggs, G. and Haldane, J. (1925). A note on the kinetics of enzyme action. *Biochemical Journal*, 19:339–339.

Friedman, N. and Koller, D. (2003). Being Bayesian about network structure. *Machine Learning*, 50:95–126.

Hill, A. (1910). The possible effects of the aggregation of the molecules of haemoglobin on its dissociation curves. *Journal of Physiology*, 40:4–7.

Michaelis, L. and Menten, M. (1913). Die kinetik der invertinwirkung. *Biochemische Zeitschrift*, 49:333–369.

Pournara, I. V. (2005). *Reconstructing gene networks by passive and active Bayesian learning*. PhD thesis, Birbeck College, University of London.

Rummelhart, D. E. (1987). *Parallel Distributed Processing*. MIT Press, Cambridge, MA, 2nd edition.

Sachs, K., Perez, O., Pe'er, D., Lauffenburger, D. A., and Nolan, G. P. (2005). Causal protein-signaling networks derived from multiparameter single-cell data. *Science, Vol 308, Issue 5721, 523-529 , 22 April 2005*, 308(5721):523–529.

Schäfer, J. and Strimmer, K. (2005a). An empirical Bayes approach to inferring large-scale gene association networks. *Bioinformatics*, 21(6):754–764.

Schäfer, J. and Strimmer, K. (2005b). An shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics. *Statistical Applications in Genetics and Molecular Biology*, page to appear.

Vohradsky, J. (2001). Neural network model of gene expression. *Faseb Journal*, 15:846–854.

Yuh, C. H., Bolouri, H., and Davidson, E. H. (1998). Genomic cis-regulatory logic: experimental and computational analysis of a sea urchin gene. *Science*, 279:1896–1902.

Yuh, C. H., Bolouri, H., and Davidson, E. H. (2001). Cis-regulatory logic in the endo16 gene: switching from a specification to a differentiation mode of control. *Development*, 128:617–629.