# Structure learning in Bayesian Networks using regular vines

Ingrid Hobæk Haff[a,*], Kjersti Aas[a], Arnoldo Frigessi[b], Virginia Lacal[c]

[a] *Norwegian Computing Center, PB 114 Blindern, NO-0373 Oslo, Norway*
[b] *Department of Biostatistics, University of Oslo, PB 1122 Blindern, NO-0317 Oslo, Norway*
[c] *University of Bergen, Department of Mathematics, P. O. Box 7800, N-5020 Bergen, Norway*

## Abstract

Learning the structure of a Bayesian Network from multidimensional data is an important task in many situations, as it allows understanding conditional (in)dependence relations which in turn can be used for prediction. Current methods mostly assume a multivariate normal or a discrete multinomial model. A new greedy learning algorithm for continuous non-Gaussian variables, where marginal distributions can be arbitrary, as well as the dependency structure, is proposed. It exploits the regular vine approximation of the model, which is a tree-based hierarchical construction with pair-copulae as building blocks. It is shown that the networks obtainable with our algorithm belong to a certain subclass of chordal graphs. Chordal graphs representations are often preferred, as they allow very efficient message passing and information propagation in intervention studies. It is illustrated through several examples and real data applications that the possibility of using non-Gaussian margins and a non-linear dependency structure outweighs the restriction to chordal graphs.

*Keywords:* Bayesian Networks, regular vines, pair-copula constructions, structure learning, chordal graph, junction tree

[*] Corresponding author

*Email addresses:* ingrihaf@math.uio.no (Ingrid Hobæk Haff), Kjersti.Aas@nr.no (Kjersti Aas), arnoldo.frigessi@medisin.uio.no (Arnoldo Frigessi), Virginia.Lacal@math.uib.no (Virginia Lacal)

## 1. Introduction

Bayesian Networks (BNs) (Pearl, 1988) or directed acyclic graphs (DAGs) allow to represent complex structured relations between random variables. They are successfully used in many areas, including genomics, where vertices represent genes and edges describe interactions in biomolecular mechanisms (Zhang et al., 2012), social sciences, where vertices are individuals and edges indicate contacts and collaborations (Banerjee et al., 2008), petroleum, where vertices stand for geographical locations and edges for geophysical relations (Martinelli et al., 2013), and telecommunication and internet networks (Bashar et al., 2010), where edges actually represent physical connections between instruments.

Learning the structure of a network from data is still one of the most exciting challenges in machine learning (Zhu et al., 2012). There are two main approaches for learning the structure of a BN, score-based and constraint-based approaches. Generally, the constraint-based methods require the existence of a faithful DAG for the data set (Tsamardinos et al., 2006), which is a rather strong assumption (Uhler et al., 2013), while the score-based methods typically do not have this restriction. On the other hand, the constraint-based algorithms may handle much larger dimensions. Since both methods have their advantages and disadvantages, researchers have tried to combine them in different ways, leading to so-called hybrid methods.

Most structure learning and inference methods for BNs have been developed for multinomial variables. Until recently, there were two ways of dealing with continuous BNs, either to first discretise the continuous variables and then work with the corresponding discrete model, or to assume joint normality. The first approach is limited by size and complexity, while the other is restricted by the normality assumption. There are many real-life situations where the dependencies between variables are far from linear, e.g. stock market prices, biometric variables, weather conditions. Assuming Gaussianity in such cases may produce incorrect networks that give a poor fit to the data. Hence, there have been some attempts at learning the structure of non-Gaussian BNs, see e.g. Margaritis (2005), Schwaighofer et al. (2007), Ma et al. (2012), Hanea et al. (2010), Elidan (2010b), Elidan (2010a), Bauer and Czado (2015) and Bauer et al. (2012). In the first three papers, non-parametric approaches are used to learn the structure, while the remaining papers combine the theory of copulae and BNs.

These previous attempts at using the framework of copulae all have some disadvantages. Hanea et al. (2010) restrict their attention to Gaussian copulae and hence to linear dependence, and the approach proposed by Elidan (2010a) is based on $k$-dimensional copulae. While the list of parametric bivariate copulae is long and varied, the choice is rather limited in higher dimensions (Genest et al., 2009). To obtain a proper likelihood, there are also restrictions regarding the copula types that can be combined. In the approach by Bauer et al. (2012), all copulae involved are bivariate and can belong to different families. However, the procedure for learning the structure requires involved computations.

In this paper we propose a new approach for learning the structure of a BN.

Like Bauer et al. (2012), we use pair-copula constructions (PCCs) (Aas et al., 2009), but we restrict ourselves to the subclass of regular vines (Joe, 1996; Bedford and Cooke, 2002), meaning that there is a well-defined and computationally efficient procedure for selecting the appropriate model. In turn, our approach allows to estimate only a restricted subclass of chordal graphs, which we characterise in this paper. However, we argue and show in a number of applications, that the benefits of assuming non-Gaussianity more than outweigh the disadvantages of being restricted to a certain type of graph structure.

The paper is organised as follows. Section 2 is a short review of pair-copula construction (PCC) and regular vines, while Section 3 treats the relationship between regular vines and chordal graphs. In Section 4, we discuss how the vine methodology may be used to learn the structure of a BN and in Section 5, we apply the methodology and compare it to other state-of-the-art structure learning approaches. Finally, Section 6 contains some concluding comments.

## 2. Pair-copula constructions and regular vines

Pair-copula constructions (PCCs), introduced by Joe (1996), are multivariate models, that decompose multivariate copulae into a product of bivariate ones. These structures have been studied by Bedford and Cooke (2001, 2002) and Kurowicka and Cooke (2006b) from a probabilistic point of view, and later by Aas et al. (2009) in an inferential context. PCCs have been shown to be useful in various applications, see, e.g., Chollete et al. (2009), Heinen and Valdesogo (2009), Berg and Aas (2009), Min and Czado (2011, 2010), Czado et al. (2012) and Smith et al. (2010). In this section we give an introduction to PCCs, focusing on the special case of regular vines.

A PCC is a multivariate copula, that is constructed from a set of bivariate ones, so-called *pair-copulae*. More specifically, the copula density is decomposed into a product of pair-copula densities. All these bivariate copulae may be selected completely freely as the resulting structure is guaranteed to be a valid copula. Hence, PCCs are highly flexible, and able to characterise a wide range of complex dependencies. Inference on PCCs is in general demanding, but the subclass of regular vines has many appealing computational properties, and hence constitutes an exception in the inferential context.

The notion of *regular vines* (R-vines) was introduced by Bedford and Cooke (2002), and described in more detail in Kurowicka and Cooke (2006b). It involves the specification of a sequence of trees, each edge of which corresponds to a pair-copula. These pair-copulae constitute the building blocks of the joint R-vine distribution. According to Definition 4.4 of Kurowicka and Cooke (2006b), an R-vine $\mathcal{V}$ on $d$ variables consists of the trees $T_1, ..., T_{d-1}$ with nodes $N_i$ and edges $E_i$ for $i = 1, ..., d-1$, which satisfy the following:

1. $T_1$ has nodes $N_1 = \{1, ..., d\}$ and edges $E_1$.
2. For $i = 2, ..., d-1$ the tree $T_i$ has nodes $N_i = E_{i-1}$.
3. Proximity condition: if two edges in tree $T_i$ are to be joined as nodes in tree $T_{i+1}$ by an edge, they must share a common node in $T_i$.

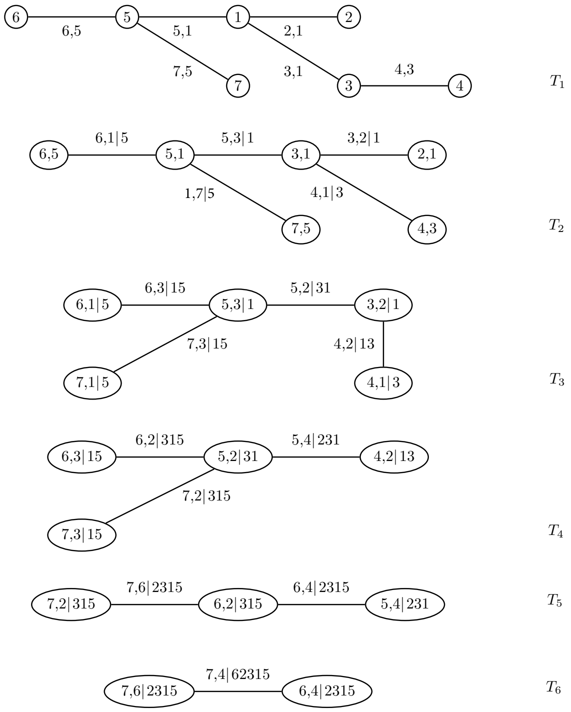


Figure 1: An R-vine tree specification on seven variables with edge indices.

To build an R-vine with node set $\mathcal{N} := \{N_1, ..., N_{d-1}\}$ and edge set $\mathcal{E} := \{E_1, ..., E_{d-1}\}$, one associates each edge $e$ in $E_i$ with a bivariate copula. Assume that $N_{ik}$ and $N_{il}$ are joined by the edge $e$ in $T_i$. As a consequence of the proximity condition, $N_{ij}$ and $N_{ik}$ share all but one node. Let $D(e)$ be the nodes they have in common, and $j(e)$ and $k(e)$ the ones they do not share, such that $N_{ij} = \{j(e), D(e)\}$ and $N_{ik} = \{k(e), D(e)\}$. Then, the nodes $j(e)$ and $k(e)$ are called the *conditioned nodes*, while $D(e)$ is denoted the *conditioning set* and the union $\{j(e), k(e), D(e)\}$ the *constraint set*. Further, the edge $e$ is associated with the bivariate copula $C_{j(e),k(e)|D(e)}$. Take for instance the edge $5, 4|231$ joining $\{5, 2|31\}$ and $\{4, 2|13\}$ in the fourth tree of Figure 1, displaying a seven-dimensional R-vine tree specification. The conditioned nodes are 5 and 4, the conditioning set is $\{1, 2, 3\}$ and the constraint set is $\{5, 4, 1, 2, 3\}$. As in Aas et al. (2009), it is assumed here that this copula is independent of the conditioning variable $\boldsymbol{X}_{D(e)}$. Hobæk Haff et al. (2010) denote the corresponding structure a *simplified* PCC. The copulae constituting the vine are organised in levels, also called trees, according to the size of their conditioning set. The copulae of level 1 have an empty set, on level 2 these sets consist of one node, on level 3 of two nodes, and so on.

Let the random vector $\boldsymbol{X}$ follow an R-vine distribution. Further let $\boldsymbol{X}_{D(e)}$ denote the subvector of $\boldsymbol{X}$ determined by the indices constituting $D(e)$. Then, Theorem 4.2 in Kurowicka and Cooke (2006b) states that the joint density of $\boldsymbol{X}$ can be written as

$$f(x_1, ..., x_d) = \left[\prod_{k=1}^{d} f_k(x_k)\right] \times \left[\prod_{i=1}^{d-1} \prod_{e \in E_i} c_{j(e),k(e)|D(e)}(F(x_{j(e)}|\boldsymbol{x}_{D(e)}), F(x_{k(e)}|\boldsymbol{x}_{D(e)}))\right]. \tag{1}$$

The right factor of the righthand side of (1) is a product of $d(d-1)/2$ bivariate copula densities, and is called an *R-vine copula.* Note that the arguments of the pair-copulae are conditional distributions in all trees but the first, where they are the univariate margins.

The key to the construction in (1) is that all copulae involved in the decomposition are bivariate and can belong to different families. There are no restrictions regarding the copula types that can be combined; the resulting structure is guaranteed to be valid anyhow. A further advantage with the R-vine copula is that the conditional distributions $F(x|\boldsymbol{v})$ constituting the pair-copula arguments can be evaluated using a recursive formula derived in Joe (1996):

$$F(x|\boldsymbol{v}) = \frac{\partial C_{xv_j|\boldsymbol{v}_{-j}}(F(x|\boldsymbol{v}_{-j}), F(v_j|\boldsymbol{v}_{-j}))}{\partial F(v_j|\boldsymbol{v}_{-j})}. \tag{2}$$

Here $C_{xv_j|\boldsymbol{v}_{-j}}$ is a bivariate copula, $v_j$ is an arbitrary component of $\boldsymbol{v}$ and $\boldsymbol{v}_{-j}$ denotes the vector $\boldsymbol{v}$ excluding $v_j$. By construction, R-vines have the important characteristic that the copulae in question always are present in the preceding trees of the structure, so that they are available without extra computations.

In order to find an expression for a general R-vine density, one needs an efficient way of storing the indices involved in the pair-copulae. One such approach was proposed by Morales-Napoles (2011) and explored in more detail in Dißmann et al. (2013). It involves the specification of a lower triangular matrix $M = (m_{i,j}|i, j = 1, ..., d) \in \{0, ..., d\}^{d \times d}$ whose diagonal entries $m_{i,i}$ are the nodes $1, ..., d$ of the first tree. Further, each row of $M$ from the bottom up represents a tree. The conditioned sets of a node is determined by a diagonal entry and the corresponding column entry of the row under consideration, while the conditioning set is given in by the column entries below this row. The R-vine matrix corresponding to the R-vine in Figure 1 is

$$M = \begin{pmatrix} 7 & & & & & & \\ 4 & 6 & & & & & \\ 6 & 4 & 5 & & & & \\ 2 & 2 & 4 & 4 & & & \\ 3 & 3 & 2 & 2 & 3 & & \\ 1 & 1 & 3 & 1 & 2 & 2 & \\ 5 & 5 & 1 & 3 & 1 & 1 & 1 \end{pmatrix}. \tag{3}$$

To determine the edges in $T_1$, we combine the numbers in the bottom row with the diagonal elements in the corresponding columns, i.e., the edges are (7,5), (6,5), (5,1) and so on. The edges of $T_2$ are given by the numbers in the second row from the bottom, associated with the diagonal elements, conditioning on the elements in the bottom row, namely (7,1|5), (6,1|5), etc. Proceeding like this, the only edge in $T_6$ is found by coupling the two upper elements in the leftmost column with the remaining 5 entries of the column as a conditioning set, i.e. (7,4|62315).

Based on $M$ the R-vine density may be written as (Dißmann et al., 2013)

$$f(x_1,...,x_d) = \left[\prod_{k=1}^{d} f_k(x_k)\right] \times \left[\prod_{j=d-1}^{1} \prod_{i=d}^{j+1} c_{m_{j,j},m_{i,j}|m_{i+1,j},...,m_{d,j}}\right], \qquad (4)$$

where the pair-copulae have arguments $F(x_{m_{j,j}}|x_{m_{i+1,j}},...,x_{m_{d,j}})$ and $F(x_{m_{i,j}}|x_{m_{i+1,j}},...,x_{m_{d,j}})$. Corresponding copula types and parameters can conveniently be stored in matrices similar to $M$.

Inference on R-vines consists in three tasks: (i) selecting the structure with all its trees, (ii) choosing a copula type for each of the $d(d-1)/2$ pair-copulae and (iii) estimating the parameters of each pair-copula. There are many possible pair-copula families, e.g. Gaussian, t, Gumbel, and Clayton. See Nelsen (2006) or Joe (1997) for a more comprehensive list. Ideally, the steps (i)-(ii) should be performed simultaneously. In practice, however, this has to be done stepwise, which is suboptimal.

The number of possible R-vines on $d$ variables is $2^{\binom{d-2}{2}-1}d!$ (Morales-Napoles, 2011). Finding the globally optimal R-vine structure for a given high-dimensional data set is therefore unfeasible, but several useful strategies have been proposed. Since the first trees can be estimated with more precision, a natural strategy is to build the structure starting from the bottom, trying to maximise the dependence in the first trees. Dißmann et al. (2013) propose such a procedure. Their algorithm starts by finding the maximum spanning tree over the $d$ nodes corresponding to the $d$ variables (using the well-known algorithm of Prim (1957)), which is a tree on all nodes that maximises the sum of the weights of the edges, using measures of pairwise dependence as weights. The subsequent trees are built in a similar manner, under the additional restriction that the proximity condition must be fulfilled. This procedure requires the simultaneous selection of pair-copula types, as well as the estimation of the parameters. There are alternatives to this bottom-up strategy, Kurowicka (2011a) starts e.g. with selecting the weakest conditional dependencies for the highest trees. To study theoretical advantages of one objective function with respect to another would be very interesting, but beyond the scope of this paper.

The copula types are typically chosen one by one, using either a model selection criterion, such as AIC, BIC or the copula specific CIC (Grønneberg, 2011), or a goodness-of-fit test. The parameters are usually estimated in a separate step. The copula parameters may be estimated using any multivariate copula estimator, such as the inference function for margins (Joe, 1997, 2005) and the semiparametric estimators (Genest et al., 1995; Shih and Louis, 1995), or the stepwise semiparametric estimator (Aas et al., 2009; Hobæk Haff, 2013), which is designed for R-vines.

The flexibility of R-vines comes at the price of the number of parameters exponentially increasing with the dimension. In high-dimensional applications, it is therefore necessary to reduce the number of parameters. One strategy is to identify as many pair-copulae as possible being equal to the independence copula, which amounts to specifying a series of conditional independencies. This may be done either by testing individual copulae for independence, so-called *pruning*, or by checking the contribution of all trees above a certain level, which is denoted *truncation*.

*Pruning:.* Pruning a particular copula $C_{jk|D}$ in the R-vine structure is the same as stating that $X_j$ and $X_k$ are conditionally independent given $\boldsymbol{X}_D$. Pruning may be performed using a copula goodness-of-fit test, e.g. the bivariate asymptotic test based on Kendall's tau (Genest and Favre, 2007). However, such a test is, strictly speaking, not an independence test unless the copulae are Gaussian, since $\tau = 0$ implies independence only for those copulae. Another option is therefore to use the Cramér-von Mises test proposed by Hobæk Haff and Segers (2015).

*Truncation:.* A truncated R-vine at level $K$ is an R-vine where all pair-copulae with conditioning set equal to or larger than $K$ are replaced by independence copulae. If $K = 1$, the truncated R-vine becomes a Markov tree distribution, that only models unconditional relationships. The density of an R-vine copula truncated at level $K$ is given by

$$c_{\text{tRV}(K)}(\boldsymbol{u}) = \prod_{j=d-1}^{1} \prod_{i=d}^{\max\{j+1,d-K+1\}} c_{m_{j,j},m_{i,j}|m_{i+1,j},\ldots,m_{d,j}}, \tag{5}$$

where $\boldsymbol{u} = (u_1, ..., u_d) \in [0,1]^d$.

The use of truncated R-vines may be justified as follows. As stated earlier in this section, the selection algorithm of Dißmann et al. (2013) builds the structure from the bottom up, trying to maximise the dependence in the first trees. Hence, if this procedure is successful, the most important and strongest (conditional) dependencies among the variables are captured by the pair-copulae in the first trees. At high levels of the structure, the parameters quantify conditional dependence with a very large number of conditioning variables. The uncertainty of the estimated copula parameters is large because of the repeated transformations of the original data using estimated conditional distribution functions (Hobæk Haff, 2012). Moreover, the parameter estimates for the upper levels do not seem to affect the lower order dependencies particularly. This indicates that it might be appropriate to truncate large structures after a certain level.

Several methods have been proposed for determining the optimal truncation level, see e.g. Kurowicka (2011b), Brechmann et al. (2012) and Brechmann and Joe (2015). In the experiments described in Section 5 we have used the approach by Brechmann et al. (2012). In this approach, one starts with $K = 1$ and fits the corresponding truncated R-vine (for $K = 0$ a pre-test of joint independence can be performed). $K$ is thereafter increased by one. If the gain from fitting the extra tree is negligible, one stops and uses the resulting specification. If not, one proceeds until one reaches a truncation level $K_0$, for which the contribution from an extra level is not significant. To assess whether the gain from fitting the extra tree is negligible we use the likelihood-ratio based test proposed by Vuong (1989), see Brechmann et al. (2012) for more details.

## 3. From R-vines to graphs

Regular vines and chordal graphs are closely connected. In this section we describe the relationship between them, but first, we briefly introduce some concepts from graph theory.

A graph $G = (V, E)$ is a set $V$ of vertices and a set $E$ of edges, which are pairs of vertices. $G$ is said to be *complete* if every pair of vertices is joined by an edge. A maximal complete subgraph is called a maximal *clique*, or simply clique. If two cliques $C_1, C_2 \subset V$ share nodes, i.e $C_1 \cap C_2 = S$ with $S \neq \emptyset$, then $S$ is the minimal complete *separator* of $C_1$ and $C_2$. The *degree* of a given node is the number of edges it has to other nodes. Two nodes are *adjacent* if they are connected by an edge.

*Bayesian Networks* (BNs) (Pearl, 1988), also known as Bayesian Belief networks, belief networks, or Bayes nets, belong to the family of probabilistic graphical models. More specifically, a BN is a directed acyclic graph (DAG). Each node in the graph represents a random variable, while the edges between the nodes represent probabilistic dependencies among the corresponding variables. An arrow from $X_i$ to $X_j$ means that variable $X_j$ is influenced by variable $X_i$. Node $X_j$ is then referred to as a *parent* of $X_j$, and $X_j$ as the *child* of $X_i$. Moreover, each variable is independent of its nondescendants in the graph given the state of its parents. This property is used to reduce, sometimes significantly, the number of parameters of the model.

BNs provide a compact representation of a high dimensional multivariate distribution. Let $\boldsymbol{X}_{PA(k)}$ be the parents of the variable $X_k$. The joint pdf is then given by

$$f(x_1, ..., x_d) = \prod_{k=1}^{d} f(x_k | \boldsymbol{x}_{PA(k)}). \tag{6}$$

If node $i$ has no parents, it is denoted a source node and $f(x_i|\boldsymbol{x}_{PA(i)}) = f(x_i)$. The *Markov blanket* $M$ of a variable $X$ consists of its parents, children and spouses (variables sharing children with $X$). Two nodes $X$ and $Y$ in a DAG are *d-separated* by a set $S$ if all paths between $X$ and $Y$ have either a fork or chain connection in $S$, or a collider connection outside $S$, that does not involve any descendants in $S$ (see Figure 2 for the different connections). Any node in a DAG is d-separated by its Markov blanket from all nodes outside the blanket. A DAG is said to be *faithful* if its d-separation statements are equivalent to the probability distribution associated with it. In particular, two distinct nodes $X$ and $Y$ in a faithful DAG are adjacent if and only if there is no subset $S \subseteq V \backslash \{X, Y\}$ such that $X$ and $Y$ are conditionally independent given $S$ (see for instance Theorem 1 in Koski and Noble (2012)). A comprehensive introduction to Bayesian networks may e.g. be found in Lauritzen (1996).

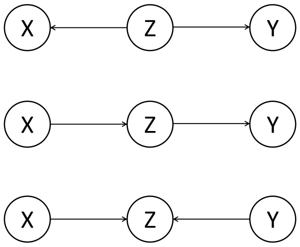


Figure 2: Different connections in a DAG: a fork on top, a chain in the middle and a collider in the bottom

An undirected graph is said to be *decomposable*, or *chordal*, if every cycle of length greater than three has a chord. Chordal graphs have been well studied, and are used in various fields such as optimisation, computer science, probability and statistics. If a graph $G$ describes a decomposable model, the joint probability density function (pdf) can be written as follows

$$p(\boldsymbol{x}) = \frac{\prod_C p(\boldsymbol{x}_c)}{\prod_S p(\boldsymbol{x}_S)},$$

where the products in the nominator and denominator run over the sets of cliques and separators, respectively. In such models, the pairwise conditional and unconditional dependencies are represented by the edges in the corresponding graph. If two variables are conditionally independent, there is no edge between them in the graph.

A decomposable model can be defined in several equivalent ways (Beeri et al., 1983):

- a Markov field whose underlying graph is chordal
- a BN with no V-structures
- a graphical model whose underlying (hyper)graph is a junction tree.

A *v-structure* consists of two of nodes that have an arrow pointing towards a third one, i.e. $X_1 \to X_2 \leftarrow X_3$ (see the last connection in Figure 2). A *Markov random field* is a set of random variables that satisfy a Markov property described by an undirected graph. More specifically, any two variables that are not adjacent, are conditionally independent given all the other variables. A *junction tree* is a tree-structured representation of an arbitrary graph. The vertices in a junction tree are the cliques from the original graph, and the edges are the separators that connect the cliques. Each junction tree defines a unique decomposable graph. However, decomposable graphs have multiple equivalent junction tree representations Thomas and Green (2009). Note that the cliques and separators are always the same, but the separators may be located in different places.

According to Deshpande et al. (2001), *decomposable models* possess several important characteristics that make them appealing. Among others, parameter estimation and statistical testing is much less demanding than in general undirected graphs. Moreover, Jensen and Jensen (1994) have shown that any scheme for exact belief updating must start by making the underlying graph decomposable. However, the problem of finding the optimal decomposable model for a given data set is known to be infeasible. Hence, heuristic search techniques are generally used.

In this section we study the relationship between truncated and/or pruned regular vines and chordal graphs. It has previously been proved (Kurowicka and Cooke, 2006a) that so-called m-saturated regular vines correspond to a chordal graph. M-saturated vines are $K$-truncated vines for which one or more of the copulae in levels 1 to $K$ are set to the independence copula. For instance, if the copulae $C_{16|5}$, $C_{14|3}$ and $C_{34}$ in the vine of Figure 1, truncated after level 2, are set to independence, the resulting vine is m-saturated, but no longer 2-truncated (note that in our definition, a $K$-truncated vine has has no independence copulae in levels 1 to $K$, which is a more restricted definition than in Brechmann et al. (2012)). Hence, truncated vines constitute a subclass of m-saturated vines, which leads us to the following proposition.

**Proposition 1.** *A d-dimensional R-vine, truncated after level $K$, defines a chordal graph whose cliques are the constraint sets of the copulae on level $K$.*

*Proof.* If $K = 1$, the graph corresponding to the vine is a tree, where the connected nodes are the variables joined by a copula. The cliques, that are all of size 2, are then obviously the constraint sets of these copulae.

Assume now that the R-vine truncated after level $K-1$ is a chordal graph whose cliques are the constraint sets of the copulae on level $K-1$. According to the proximity condition, the constraint set of each copula $C_{jk|D}$ on level $K$ is the union of the constraint sets $\mathcal{C}_i = \{j_i, k_i, D_i\}$ and $\mathcal{C}_l = \{j_l, k_l, D_l\}$ of exactly two copulae on level $K-1$, and these two constraint sets are the same except for one node. Assume without loss of generality that $j_i \neq j_l$ and $\{k_i, D_i\} = \{k_l, D_l\}$. Then, the conditioned set of $C_{jk|D}$ must be $\{j_i, j_l\}$ and the conditioning set $\{k_i, D_i\}$. The nodes of $\{j_i, k_i, D_i\}$ are all connected since they constitute a clique in the graph corresponding to the R-vine truncated after level $K-1$. Moreover, $\{j_i, j_l\}$ are connected in the graph of the R-vine truncated after level $K$ via the copula $C_{jk|D}$. Hence, $\{j_i, k_i, D_i, j_l\}$ must either be a clique or the subset of one. By construction, each pair of nodes is connected exactly once in a full R-vine. Hence, the pair $\{j_i, j_l\}$ cannot be part of any of the constraint sets of the copulae on level $K-1$. Thus, $\{j_i, k_i, D_i, j_l\}$ must be a maximal clique in the graph corresponding to the R-vine truncated after level $K$. □

It follows directly from Proposition 1 that the graph corresponding to a truncated R-vine has cliques that are all of size $K+1$, and since the proximity condition requires that the constraint sets of two copulae connected on a given level are the same except for one node, all separators must be of size $K$. Moreover, it entails that the number of cliques is $d-K$ and that they may be directly identified in the vine matrix $M$. If one sets all sub-diagonal entries above row $d-K$ to 0, the cliques are given by the non-zero entries of columns 1 to $d-K$.

Not all chordal graphs with $d-K$ cliques and $d-K-1$ separators may be represented by a $K$-truncated vine. Figure 3 provides a counterexample. The following proposition gives the characteristics of the subclass of chordal graphs corresponding to a $K$-truncated vine:

**Proposition 2.** *A $d$-dimensional regular vine truncated after level $K$ defines a chordal graph on $d$ nodes with cliques $\mathcal{C}_i$, $i = 1, \ldots, n_C$, with $n_C = d - K$, and separators $\mathcal{S}_i$, $i = 1, \ldots, n_S$, that fulfils the following conditions:*

1. *$|\mathcal{C}_i| = d - n_C + 1$, $i = 1, \ldots, n_C$.*
2. *$n_S = n_C - 1$ and $|\mathcal{S}_i| = d - n_C$, $i = 1, \ldots, n_S$.*
3. *For all $i$, let $\mathcal{S}_{i_1}, \ldots, \mathcal{S}_{i_n}$ be the separators connecting $\mathcal{C}_i$ to other cliques in the junction tree. If $n > 2$ and there exists a pair of indices $\{j, k\}$, such that $\mathcal{S}_{i_j} = \mathcal{S}_A \neq \mathcal{S}_B = \mathcal{S}_{i_k}$, then $\mathcal{S}_{i_m} = \mathcal{S}_A \vee \mathcal{S}_{i_m} = \mathcal{S}_B$, $m = 1, \ldots, n$.*
4. *Let $\mathcal{G}_k$ be the graph corresponding the R-vine truncated after level $k$. Then $\mathcal{G}_k$ must fulfill the three conditions 1, 2 and 3 for any $k \leq K$.*

*Proof.* Conditions 1 and 2 follow directly from Proposition 1. Condition 3 follows from the proximity condition of the regular vine. According to Proposition 1, the cliques are the constraint sets of the copulae on level $K$. Hence, if two cliques are adjacent, it means that the corresponding copulae on level $K$ are connected by a copula on level $K+1$ (an independence copula, since it is above the truncation level). The proximity condition requires that the constraint sets $\mathcal{C}_i = \{j_i, k_i, D_i\}$ and $\mathcal{C}_l = \{j_l, k_l, D_l\}$ of these two copulae have all nodes in common except one. Since the conditioned sets $\{j_i, k_i\}$ and $\{j_l, k_l\}$ cannot be exactly the same, the separator connecting the two cliques must be either $\mathcal{S}_A = \{j_i, D_i\}$ or $\mathcal{S}_B = \{k_i, D_i\}$. Hence, all separators connecting $\mathcal{C}_i$ to other cliques must be equal to either $\mathcal{S}_A$ or $\mathcal{S}_B$.

Condition 4 is a consequence of the fact that an R-vine may be truncated after any level. □

Condition 3 means that if a clique in the graph corresponding to a regular vine is adjacent to more than two others, the separators associated with this clique must be at most two different sets of nodes. This is necessary for the proximity condition to be fulfilled, and is the one that is violated by the two cliques $\{B, C, E\}$ and $\{B, G, E\}$ in the graph of Figure 3. The left panel of Figure 4 shows the junction tree of a graph that fulfills the first three conditions of Proposition 2, but not the fourth. This is the junction tree of $\mathcal{G}_4$. Hence, $\{1, 2, 3\}$ is a clique in $\mathcal{G}_3$ (shown in the right panel), that is connected to $\{1, 3, 6\}$ with separator $\{1, 3\}$, to $\{2, 3, 4\}$ with separator $\{3, 4\}$ and to $\{1, 2, 5\}$ with separator $\{1, 2\}$. This means that $\mathcal{G}_3$ violates Condition 3, and therefore, the graph in Figure 4 cannot correspond to an R-vine. Each of the three cliques in dashed boxes, including the simplical nodes 7, 8 and 9, is one of three possible

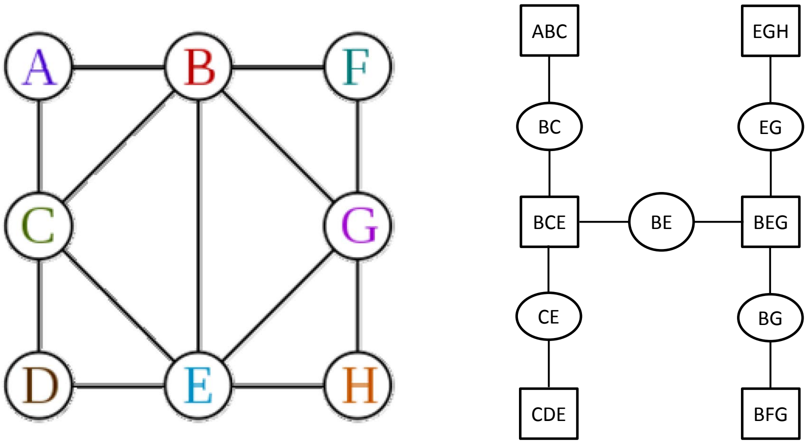


Figure 3: The left panel shows a chordal graph that does not correspond to an m-saturated vine (source: `http://en.wikipedia.org/wiki/Chordal_graph`). The right panel shows the corresponding junction tree. Squares and circles correspond to cliques and separators, respectively.

cliques. The choice of these does not influence the nonfulfillment of Condition 3.

When a truncated R-vine in addition is pruned, i.e. at least one of the $K$ first levels contains an independence copula, the cliques will in general have different sizes. More importantly, the corresponding graph may no longer be chordal. For the graph to remain chordal after the pruning, the statement of conditional independence must be compatible with the statements of conditional dependence implied by copulae in higher levels of the vine. This implies that the pruning must be performed in a certain order, starting with level $K$, which is the highest level where there are copulae that are not the independence copula. At this level all copulae may always be pruned. At the remaining levels, however, a copula may only be set to independence if the corresponding conditional distributions are used as arguments in independence copulae in the subsequent level. In Section 4 we give an algorithm for how to prune in a way that makes the resulting graph decomposable. The class of truncated vines that are pruned in this way corresponds to the class of m-saturated vines defined in Kurowicka and Cooke (2006a).

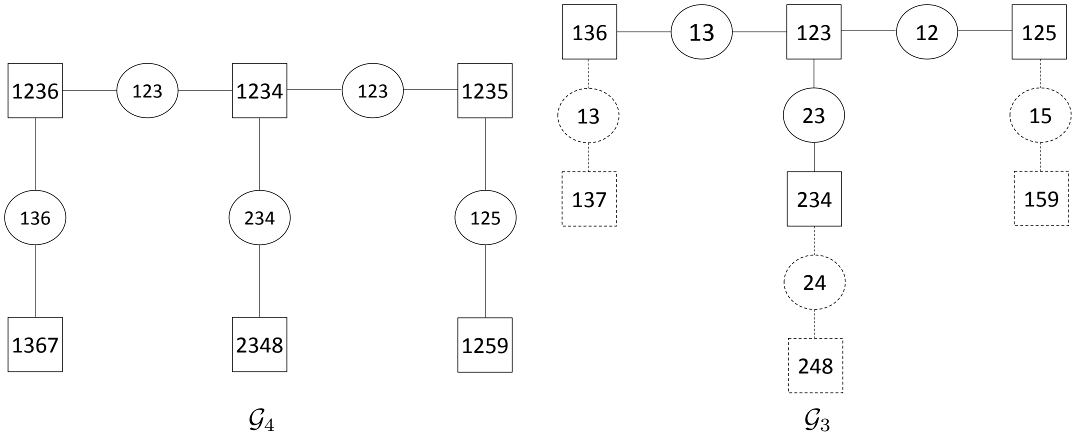


Figure 4: Left panel: junction tree of a graph that does not correspond to a truncated R-vine after level $K = 4$ although it fulfills Conditions 1 to 3 of Proposition 2. Right panel: junction tree of the graph that would have corresponded to truncation level $K = 3$, if this were an R-vine.

## 4. Structure learning using vines

In this section, we outline the specific procedure for structure learning using regular vines. Further, we discuss how this approach differs from the other copula-based approaches proposed in the literature.

The specific procedure for learning a BN using the R-vine methodology is as follows:

1. Fit a truncated R-vine to the data.
2. Convert the corresponding R-vine matrix to an adjacency matrix representing the chordal graph
3. Choose the BN as one of the multiple equivalent junction tree representations.

The first step is described in Section 2. Since we know from Section 3 that the undirected graph corresponding to the truncated vine really is chordal, we may find the junction tree simply by using the R-vine matrix to identify the cliques, as described in Section 3. More specifically, a so-called adjacency matrix $A$ is constructed from the R-vine matrix $M$ as follows:

**Algorithm 1**

```
Input: An R-vine matrix $M$
for $i = 1, \ldots, d$ do
    for $j = 1, \ldots, d$ do
        $a_{ij} = 0.$
    end for
end for
for $i = d - K + 1, \ldots, d$ do
    for $j = 1, \ldots, i - 1$ do
        $a_{m_{jj}, m_{ij}} = a_{m_{ij}, m_{jj}} = 1$
    end for
end for
Output: The adjacency matrix $A$
```

After step 2, we have a chordal graph. To obtain a BN, it remains to orient the edges. Since a chordal graph has no V-structures, these directions may be set in any way, under the restriction of acyclicity and compatibility with the R-vine decomposition of the joint pdf. Assume e.g. that the R-vine in Figure 1 has been truncated after level 2. After running through Algorithm 1, we obtain the following adjacency matrix

$$A = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}, \tag{7}$$

and the chordal graph shown in the left panel of Figure 5, with the corresponding junction tree in the right panel. The corresponding probability density of the R-vine is given by (for the readers' convenience we have omitted the copula arguments):

$$\begin{aligned} f(x_1, ..., x_7) = \left[\prod_{i=1}^{7} f(x_i)\right] & \cdot c_{65}\, c_{51}\, c_{21}\, c_{75}\, c_{31}\, c_{43} \\ & \cdot c_{61|5}\, c_{53|1}\, c_{32|1}\, c_{17|5}\, c_{41|3}. \end{aligned} \tag{8}$$

This density may also be written as follows

$$f(x_1, ..., x_7) = \frac{f(x_6, x_1, x_5)\, f(x_5, x_3, x_1)\, f(x_3, x_2, x_1)\, f(x_1, x_7, x_5)\, f(x_4, x_1, x_3)}{f(x_1, x_5)\, f(x_1, x_5)\, f(x_1, x_3)\, f(x_1, x_3)},$$

which can be recognised as the junction tree representation of a decomposable model. As shown in Figure 5, the cliques and separators may actually be directly identified from the nodes and edges at the truncation level in the R-vine, displayed in the middle panel.

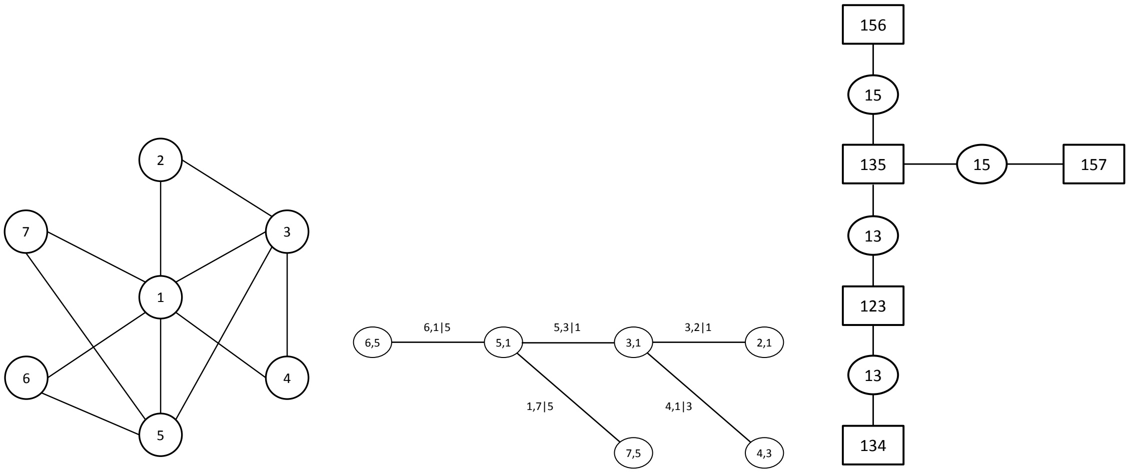


Figure 5: Left panel: Chordal graph corresponding to the R-vine in Figure 1, truncated after level 2. Middle panel: R-vine tree corresponding to the truncation level. Right panel: Junction tree corresponding to the graph in the left panel.

There are a number of ways of factorising the particular pdf above into a product of conditional pdfs, each resulting in a different BN. Note however that all these BNs correspond to exactly the same likelihood since they are composed of the same pair-copulae and marginal distributions. The decomposition one finally chooses to use may e.g. depend on which conditional distributions one is most interested in. In the example above, one possible BN-representation is:

$$\begin{aligned} f(x_1, ..., x_7) = f(x_1) \cdot f(x_3|x_1)\, f(x_2|x_1, x_3)\, f(x_4|x_1, x_3)\, f(x_5|x_1, x_3) \\ \cdot f(x_6|x_1, x_5)\, f(x_7|x_1, x_5). \end{aligned} \tag{9}$$

Comparing (8) and (9) we see that the different conditional densities may be written as follows

$$\begin{aligned} f(x_3|x_1) &= f(x_3) \cdot c_{31} \\ f(x_2|x_1, x_3) &= f(x_2) \cdot c_{21} \cdot c_{32|1} \\ f(x_4|x_1, x_3) &= f(x_4) \cdot c_{43} \cdot c_{41|3} \\ f(x_5|x_1, x_3) &= f(x_5) \cdot c_{51} \cdot c_{53|1} \\ f(x_6|x_1, x_5) &= f(x_6) \cdot c_{65} \cdot c_{61|5} \\ f(x_7|x_1, x_5) &= f(x_7) \cdot c_{75} \cdot c_{17|5}. \end{aligned}$$

If one wishes to investigate if any further conditional independence is present in the BN or to reduce the parameter space additionally, one may prune the R-vine after having truncated it. To ensure that the resulting graph still is decomposable, the pruning must be performed in a certain order. Assume that the truncation level is $K$. Then, the pruning procedure is as follows:

**Algorithm 2**

```
Input: A $K$-truncated vine
for levels $i = K, \ldots, 1$ do
    for edges $e \in E_i$ do
        if corresponding node at level $i$ is connected to independence edges only, then
            Test whether $C_{j(e),k(e)|D(e)}$ may be set to the independence copula
            using the test by Genest and Favre (2007).
        end if
    end for
end for
Output: A $K$-truncated and pruned vine
```

Assume for instance that the model obtained from the structure learning algorithm is the R-vine in Figure 1, truncated after the third level. This corresponds to a chordal graph with the four cliques $\{1, 3, 5, 7\}$, $\{1, 3, 5, 6\}$, $\{1, 2, 3, 5\}$ and $\{1, 2, 3, 4\}$. Starting at level 3, all copulae $C_{37|15}$, $C_{36|15}$, $C_{25|13}$ and $C_{24|13}$ may be pruned. At level 2, $C_{17|5}$, $C_{16|5}$ and $C_{14|3}$ can be set to independence if the copulae $C_{37|15}$, $C_{36|15}$, $C_{24|13}$, respectively, are independence copulae. The pair $\{2, 3\}$ is present in two copulae: $C_{25|13}$ or $C_{24|13}$. Hence, the copula $C_{23|1}$ can be pruned only if both these two copulae have been previously pruned. Finally, the last copula on level 2, $C_{35|1}$, can be pruned only if all three copulae $C_{37|15}$, $C_{36|15}$, $C_{25|13}$ have been. At level 1, $C_{57}$, $C_{56}$, $C_{12}$ and $C_{34}$ can be set to independence if the copulae $C_{17|5}$, $C_{16|5}$, $C_{23|1}$, and $C_{14|3}$, respectively, are independence copulae. The pairs $\{1, 5\}$ and $\{1, 3\}$ are both present in three copulae. Hence, for $C_{15}$ to be pruned, all of $C_{17|5}$, $C_{16|5}$ and $C_{35|1}$ must be independence copulae, and finally $C_{13}$ may only be tested for independence if $C_{35|1}$, $C_{14|3}$ and $C_{23|1}$ have been previously pruned.

When the vine is pruned, the cliques are no longer directly found in the R-vine matrix, but one may for instance use the algorithm described in Sections 4.3 and 4.4 of Cowell et al. (1999) to determine the junction tree. This algorithm has been implemented in the function `ug.to.jtree` of the R-package `lcd`.

The works of Elidan (2010a) and Bauer et al. (2012) are the ones bearing the strongest resemblance to ours, representing the relationship between a variable and its parents using copulae. We denote the method of Elidan (2010a) Copula Bayesian Networks (CBNs), the one of Bauer et al. (2012) Pair-copula Bayesian Networks (PCBNs) and ours Vine-copula Bayesian networks (VCBN). All three approaches model the joint pdf by (6). However, they model the conditional densities $f(x_k|\boldsymbol{x}_{PA(k)})$ differently. Let $y_1, \ldots y_{n_k}$ be the parents of variable $x_k$. In the CBN, the conditional density $f(x_k|y_1, \ldots y_{n_k})$ is given by

$$f(x_k|y_1, \ldots y_{n_k}) = f(x_k)\, \frac{c(F(x_k), F(y_1), \ldots, F(y_{n_k}))}{c(F(y_1), \ldots, F(y_{n_k}))}, \tag{10}$$

while in the PCBN and VCBN, it is given by

$$\begin{aligned} f(x_k|y_1,\dots y_{n_k}) = f(x_k) &\cdot c(F(x_k),F(y_1))\, c(F(x_k|y_1),F(y_2|y_1)) \\ &\cdot c(F(x_k|y_1,y_2),F(y_3|y_1,y_2))\cdots \\ &\cdot c(F(x_k|y_1,y_2,\dots,y_{n_k-1}),F(y_{n_k}|y_1,y_2,\dots,y_{n_k-1})). \end{aligned} \tag{11}$$

If the copulae in (10) and (11) are all chosen to be Gaussian, the two representations are equal. In general, however, the representation in (11) has two main advantages compared to the CBN. Firstly, in (11), all copulae are bivariate, while the CBN is based on higher-dimensional copulae. While the list of parametric bivariate copulae is long and varied, the choice is rather limited in higher dimensions (Genest et al., 2009). Secondly, in (11) all copulae can belong to different families. There are no restrictions regarding the copula types that can be combined; the resulting density is guaranteed to be valid anyhow. This is not the case for the CBN. Take for instance the R-vine in Figure 1, truncated after the third level, which corresponds to a chordal graph with the cliques $\{1,3,5,7\}$, $\{1,3,5,6\}$, $\{1,2,3,5\}$ and $\{1,2,3,4\}$. The construction of a CBN for this model requires the specification of the copulae $C_{1357}$, $C_{1356}$, $C_{1235}$ and $C_{1234}$. For the resulting probability density to be valid, the lower-dimensional margins that these copulae share must be the same. In other words, the margin $C_{135}$ must be the same for the three first copulae, and $C_{13}$ must be the same for all four. Achieving this, along with the specified conditional independencies, is very difficult, if possible, with other copula families than the Gaussian.

The main difference between the PCBN and the VCBN is that in the first, the parents of any node may be freely selected, while in the latter they must be chosen in a fashion that satisfies the R-vine structure. Bauer and Czado (2015) investigate the difference in performance between a PCBN and a VCBN. They find that the likelihood functions of the PCBNs have a consistently higher mode than those of the corresponding VCBNs, and the former also have fewer parameters. Typically, there are some copulae in the PCBNs that are not directly specified in the VCBNs, but unless these copulae have a strong dependence, the difference between the likelihood functions of the two models tends to be rather small. Moreover, the flexibility of the PCBNs comes at a price. In the VCBN, the conditional distributions $F(x|\boldsymbol{v})$ in (11) may be computed as the derivatives of copulae from the previous level of the vine, while in the PCBN, the determination of the conditional distributions may involve high-dimensional integration. Further, there is a well-defined and relatively fast procedure for determining the most appropriate structure for the VCBN, described in Section 2,while selecting the most appropriate model from the class of PCBNs is computationally very expensive. More specifically, this selection is made using the PC algorithm (see Section 5), replacing the independence test based on assumptions of Gaussianity with a more general test using the Rosenblatt transform. For example, to test whether $X_i$ and $X_j$ are independent given $\boldsymbol{X}_k$, one tests whether $W_{i|k} = F(X_i|\boldsymbol{X}_k)$ and $W_{j|k} = F(X_j|\boldsymbol{X}_k)$ are unconditionally independent. In order to compute the Rosenblatt transforms $W_{i|k}$ and $W_{j|k}$, one estimates a vine for the variables $X_i$, $X_j$ and $\boldsymbol{X}_k$, that has the copula $C_{ij|k}$ in its final level. This requires selecting and estimating the copulae of as many vines as there are independence tests in the procedure, i.e. a potentially very large number of copulae.

## 5. Experiments

To show the usefulness of the regular vine approach for learning networks, we have compared it with four state-of-the-art approaches on four different data sets: an 8-dimensional data set generated by simulation from a known Gaussian BN, an 8-dimensional data set generated by simulation from a known non-Gaussian BN, time series for 52 European stock indices, precipitation series from 22 sites and the Abalone data set, previously used for BN structure learning. In all examples we have used the following approaches: the Hill-Climbing greedy search algorithm, the Max-Min Hill Climbing algorithm, the grow-shrink Markov blanket algorithm, the PC algorithm and the R-vine structure selection algorithm. In the first two examples, we have also employed the PCBN method, and in the last example, both the PCBN and CBN approaches. As explained in Section 4, it is very difficult, if possible, to ensure that a CBN has a coherent probability distribution function if not all copulae are Gaussian. Therefore, we have only used the CBN approach with Gaussian copulae. Since that is equivalent to employing Hill-Climbing when the margins are normal, we have only included the CBN in the last example, where the margins are not transformed to normal distributions. Further, the PCBN is computationally very demanding. Hence, we have omitted it in the third and fourth examples, where the number of variables is quite large.

The PCBN and CBN approaches are sketched in Section 4. Below, we briefly describe the state-of-the-art methods and give an overview of the parameter settings that are common to all examples in Section.

Previously proposed approaches for learning the structure of a BN broadly fall into one of two categories: score-based and constraint-based approaches. Generally, the constraint-based methods require the existence of a faithful DAG for the data set (Tsamardinos et al., 2006), as opposed to score-based methods. On the other hand, they may handle much larger dimensions. Since both methods have their advantages and disadvantages, researchers have tried to combine them in different ways, leading to so-called hybrid methods.

Score-based methods treat the learning task as a combinatorial optimisation problem. They use a certain search technique to find candidate BNs. The algorithm finally picks the candidate that maximises a chosen scoring metric, for instance the Bayesian information criterion (BIC). The main problem with this approach is that the search space increases exponentially with the number of variables. In practice, one must therefore resort to a heuristic search algorithm. In this paper, we have considered the Hill-Climbing greedy search algorithm implemented in the R-package `bnlearn` (Scutari, 2010). This algorithm starts with an arbitrary network. Then it performs a local search by changing one element of the graph at a time in order to find a better fitting network. Hence, it either adds or removes an edge, or changes the direction of one of the existing edges. The different network structures are ranked with respect to their BIC scores, assuming that the joint distribution is multivariate normal.

Constraint-based methods consist in a series of tests for conditional independence between the variables. We have considered the PC algorithm (Spirtes et al., 2000), implemented in the R-package `pcalg` (Kalisch et al., 2012), and the grow-shrink Markov blanket algorithm (Margaritis, 2003) from the R-package `bnlearn`. Both algorithms test for conditional independence by testing whether the corresponding partial correlations are different from 0. A partial correlation equal to 0 does not imply conditional independence unless the joint distribution of the variables in question is the multivariate normal.

The difference between the two constraint-based methods is the order in which the conditional independence tests are performed. The PC algorithm starts with a complete undirected graph, and tests all pairs of adjacent nodes for conditional independence given an increasing number of conditioning variables. The grow-shrink algorithm, on the other hand, constructs the skeleton of the BN by trying to find the Markov blanket of each variable, that consists of its parents, children and spouses (variables sharing children with $X$). Starting with an empty blanket, the algorithm begins with a growing phase, adding variables to the blanket, and then switches to a shrinking phase, where variables are removed from it.

The constraint-based methods only find the so-called *skeleton* of the network, i.e. the undirected graph corresponding to the DAG. The directions are set afterwards by identifying v-structures and propagating directions to other edges in order to satisfy the condition of acyclicity. This is a completely deterministic procedure.

The hybrid methods try to combine ideas from constraint- and score-based techniques in an effective way. In this paper, we have used the Max-Min Hill-Climbing (MMHC) algorithm (Tsamardinos et al., 2006), which first reconstructs the skeleton of a Bayesian network and then performs a Bayesian-scoring greedy hill-climbing search to orient the edges.

To run the Hill-Climbing greedy search algorithm, we have applied the `hc`-function implemented in the R-package `bnlearn` with the Bayesian information criterion (BIC) as the score function. To run the Grow-Shrink Markov blanket algorithm, we have used the `gs`-function implemented in the R-package `bnlearn`, testing for conditional independence with Fisher's Z test. In examples 1, 2, 4 and 5, the chosen significance level was 5%, while in example 3, we had to use 1% in order to get convergence of the algorithm. We employed the function `pc`, implemented in the R-package `pcalg`, to run the PC-algorithm, with the function `gaussCItest` to test conditional independencies at significance level 1%. Finally, the function `mmhc` in `bnlearn` was used to run the Max-Min Hill-Climbing algorithm, and the R-vine structure was selected using the function `RVineStructureSelect` implemented in the R-package `VineCopula`. In principle the latter function allows to choose pair-copulae from 40 different families. However, to keep the computational time low, we have restricted the choice to two or three families, selected by visual inspection or prior knowledge. To determine the optimal truncation level we have used the likelihood-ratio based test described in Section 2, while the pruning was done using the copula independence test suggested by Genest and Favre (2007). In both tests the significance level was set to 5%, and the functions that were used were `RVineVuongTest` and `BiCopIndTest` from `VineCopula`, respectively. For the selection of PCBNs, we modified the function `pc` from `pcalg`, substituting the function `gaussCItest` with the independence test of Genest and Favre (2007). We used the functions `CDVineCopSelect` and `BiCopHfunc`, implemented in the `CDVine` package, to compute the Rosenblatt transforms, and a significance level of 5% in the independence tests. Note that in the original PCBN-approach, Bauer et al. (2012) search for the optimal C- or D-vine for the relevant variables when they compute the Rosenblatt transforms for the independence tests. Here, we simply choose a D-vine which has the needed conditional distributions in the top level. This may have affected the performance of the PCBN, but has also reduced the computation time significantly. For the CBN, we modified the `hc`-function from `bnlearn`, computing the BIC score based on Gaussian copulae instead of the multivariate Gaussian distribution.

To estimate the parameters of the networks obtained using `hc`, `gs`, `mmhc` and `pc`, we used the function `bn.fit` in the R-package `bnlearn`, that computes the maximum likelihood estimates. The parameters of the R-vine structure were estimated using the function `RVineSeqEst` in the R-package `VineCopula`, which performs the stepwise semiparametric estimation described in Section 2. To select the copulae and estimate the parameters of the PCBNs, we used the function `BiCopSelect` from `VineCopula`, choosing among the same copula families as for the vine approach. The multidimensional integrations, necessary for the computation of some of the conditional distributions that are pair-copula arguments, were performed with the function `cuhre` implemented in the R-package `R2Cuba`. The parameters of the CBN were estimated with the function `fitCopula`, implemented in the R-package `copula`.

In this paper we are mainly interested in the dependence structure of the data set in question and not the marginal fit. Hence, in all examples except the last, and for all approaches, we have fit the same parametric distributions to the univariate margins. Since the Hill-Climbing, MMHC, Grow-Shrink and PC-algorithms all implicitly assume that the margins are normal, we assume normal distributions for the margins also when using the regular vine approach. This allows us to isolate the effect of replacing the dependence structure of the multivariate normal distribution with an R-vine. However, it also reduces the potential leverage of the vine approach compared to the others, as one of the great advantages of this approach is precisely the possibility to choose the margins freely.

In order to examine the accuracy of the fitted Bayesian networks, we follow e.g. Acid et al. (2004), and use the Akaike (AIC) and the Bayesian information criteria. The BIC penalises the number of parameters more than the AIC, and therefore favours sparser models.

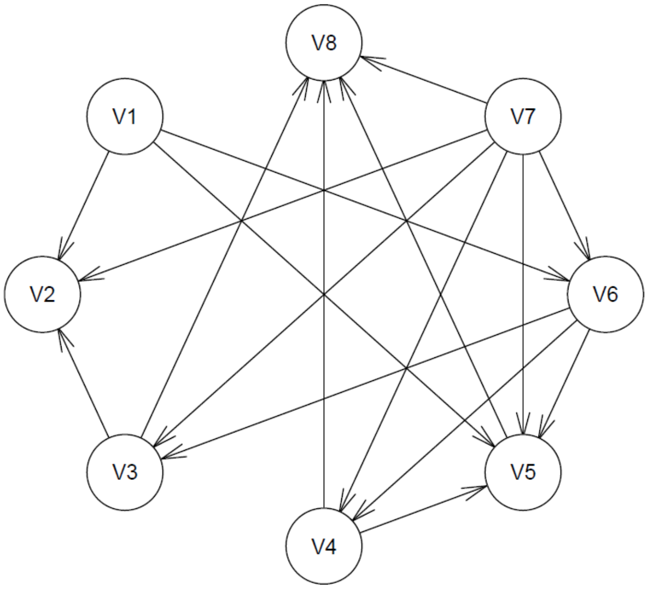


Figure 6: Network used in Experiment 1.

### *5.1. Simulated Gaussian data*

In the first example, we have generated 10,000 samples from a Gaussian BN represented by the graph shown in Figure 6. The reason why we generate this many samples is that we want to ensure that the data set gives an adequate representation of the BN. The probability density corresponding to this graph is

$$f(V_1)\,f(V_7)\,f(V_6|V_1,V_7)\,f(V_3|V_6,V_7)\,f(V_4|V_6,V_7)\,f(V_2|V_1,V_3,V_7)$$
$$\cdot\,f(V_5|V_1,V_4,V_6,V_7)\,f(V_8|V_3,V_4,V_5,V_7).$$

The corresponding graph is not chordal. The closest chordal graph (in terms of the least number of added edges) corresponds to an m-saturated, but not a $K$-truncated regular vine. Further, the parameters of the above distribution are such that there exists a DAG that is faithful to the distribution. Hence, the regular vine approach has a clear disadvantage compared to the traditional approaches in this case.

For this data set, the Gaussian copula would obviously have been chosen for most of the pairs if we had selected pair-copulae from all the 40 possible families. Hence, we fitted an R-vine structure merely with Gaussian copulae. Table 1 summarises the fit for all approaches. The R-vine obtained with the algorithm from Section 4 is the one called “Optimal vine”. As it turns out, it was not truncated and none of the copulae were pruned. For the sake of comparison

Table 1: Simulated Gaussian data: Number of edges, log-likelihood, number of parameters, AIC and BIC for the different approaches.

| *Method* | *No. of edg.* | *Log-lik* | *No. of par.* | *AIC* | *BIC* |
|---|---|---|---|---|---|
| True model | 17 | 86664.44 | 33 | -173262.90 | -173024.90 |
| Hill-climbing bic | 23 | 86667.98 | 39 | -173258.00 | -172976.80 |
| Max-Min HC bge | 17 | 85562.42 | 33 | -171058.80 | -173023.00 |
| Grow-shrink | 14 | 83241.75 | 30 | -166423.50 | -166207.20 |
| PC-algorithm | 13 | 81119.85 | 21 | -162181.70 | -161972.60 |
| PCBN | 13 | 79179.79 | 29 | -158301.60 | -158117.80 |
| Optimal vine ($K = 7$) | 28 | 86669.92 | 44 | -173251.80 | -172934.60 |
| 3-level vine | 18 | 84303.17 | 34 | -168538.30 | -168293.20 |

we have included the corresponding results for the true model and for a vine truncated after the third level. For all approaches, the number of parameters is equal to the number of edges plus twice the dimension. We see that the Hill-Climbing approach gives the best fit. However, the optimal vine is a very close competitor. The performance of the constraint-based approaches, including the PCBN, is not particularly good. All three methods are outperformed by a regular vine truncated after level 3. Hence, the regular vine approach is competitive even for this data set, which is designed to give the other methods an advantage.

Since we know the truth, a structural comparison may also be performed for this example. Table 2 shows the number of true positive, false positive and false negative edges for the different approaches (we ignore the direction of the edges in this comparison, since there are a number of ways to set the directions for the vine structures that all correspond to the same likelihood). As the optimal vine corresponds to a saturated graph, the number of false negative edges is 0. The number of false positive edges is however as large as 11. The reason why this vine nevertheless is a very close competitor in terms of the AIC and BIC, is that several of the edges in the corresponding graph correspond to very weak dependencies.

From Table 2 one may think that the PC and PCBN algorithms perform quite well compared to the optimal vine since the sum of false positive and false negative edges is much lower. Table 1 shows that it is in fact the other way around. The reason is that the graphs obtained by the PC and PCBN algorithms are missing some edges that are essential for the approximation to be appropriate, while the false positive edges in the optimal vine either correspond to very weak dependencies or cancel each other out.

### *5.2. Simulated non-Gaussian data*

In this example, we have generated 10,000 samples from a BN represented by the graph shown in Figure 7, which is a so-called Chow-Liu tree (Chow and

Table 2: Simulated Gaussian data: Number of true positive edges, number of false positive edges and number of false negative edges for the different approaches.

| *Method* | *TP* | *FP* | *FN* |
|---|---|---|---|
| True model | 17 | 0 | 0 |
| Hill-climbing bic | 16 | 7 | 1 |
| Max-Min HC bge | 14 | 3 | 3 |
| Grow-shrink | 13 | 1 | 4 |
| PC-algorithm | 13 | 0 | 4 |
| PCBN | 10 | 3 | 7 |
| Optimal vine ($K = 7$) | 17 | 11 | 0 |
| 3-level vine | 12 | 6 | 5 |

Liu, 1968). This graph may be represented by the following probability density

$$f(V_1)\, f(V_2|V_1)\, f(V_4|V_2)\, f(V_6|V_4)\, f(V_7|V_4)\, f(V_5|V_4)\, f(V_8|V_7)\, f(V_3|V_7).$$

which again may be written as

$$\left[\prod_{i=1}^{8} f(V_i)\right] c_{12}\, c_{24}\, c_{46}\, c_{45}\, c_{47}\, c_{73}\, c_{78}. \qquad (12)$$

If all pair-copulae $C_{ij}$ in (12) are chosen to be Gaussian and the margins $f(V_i)$ are assumed to be standard normal, all approaches recover the true structure exactly. However, if the pair-copulae are substituted by Clayton with parameter 3 (corresponding to a Kendall's tau of 0.6), keeping the standard normal margins, only the vine and the PCBN approaches succeed in determining the correct number of edges. Figure 8 shows scatter plots of the simulated data, while Table 3 shows the results. As can be seen from the table, the state-of-the-art methods detect many false edges. This may be explained as follows. If the true data are not multivariate normal, one might get a better approximation of the true density by adding more edges.

The PCBN and our approach find approximately the same model for this data set, and their performance is comparable. Actually, the PCBN is also a vine in this case. However, the CPU-time spent on the PCBN is 4150 times longer than the one spent on the VCBN, although no multidimensional integration was required for this model.

### *5.3. Stock data*

In the third example, we have used the Euro Stoxx 50 data set previously analysed in e.g. Brechmann (2013). The Euro Stoxx 50 index is a major barometer of the financial markets in the Eurozone. It covers stocks of 50 large Eurozone companies, selected based on their market capitalisation. In our experiment we have used the Euro Stoxx 50 index itself, 46 of its underlying stocks, and the German, French, Italian, Spanish and Dutch national indices. This results in a

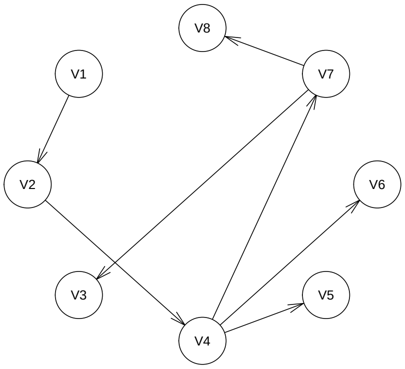


Figure 7: Network used in Experiment 2.

Table 3: Simulated Normal-Clayton data : Log-likelihood, number of parameters, AIC and BIC for the different approaches.

| *Method* | *No. of edg.* | *Log-lik* | *No. of par.* | *AIC* | *BIC* |
|---|---:|---:|---:|---:|---:|
| Hill-climbing bic | 25 | -81116.84 | 41 | 162315.70 | 162611.30 |
| Max-Min HC bge | 22 | -81167.91 | 38 | 162411.80 | 162685.80 |
| Grow-shrink | 21 | -84276.72 | 37 | 168894.20 | 168627.40 |
| PC-algorithm | 22 | -83712.63 | 38 | 167501.30 | 167775.20 |
| PCBN | 7 | -45702.01 | 23 | 91450.03 | 91546.97 |
| Optimal vine ($K = 1$) | 7 | -45064.79 | 23 | 90175.58 | 90272.52 |

52-dimensional data set. Like Brechmann (2013), we consider daily log returns over the 4-year period from May 22, 2006 to April 29, 2010, which corresponds to 985 observations. The log returns are preprocessed as follows before further modelling. First, the serial dependence in the conditional mean and in the conditional variance are modelled by ARMA- and GARCH-models (see Appendix A in Brechmann (2013) for further details). Then, the resulting residual vectors are converted to approximately uniform variables using the fitted marginal distribution functions. In the final step, each margin is transformed to a standard normal distribution, using the Gaussian quantile function. Figure 9 shows pair-plots of the first six variables from the preprocessed data. These are elliptically shaped, which should be compatible with the assumption of multivariate normality.

According to Brechmann (2013), the t-copula gave the best fit for the ma-

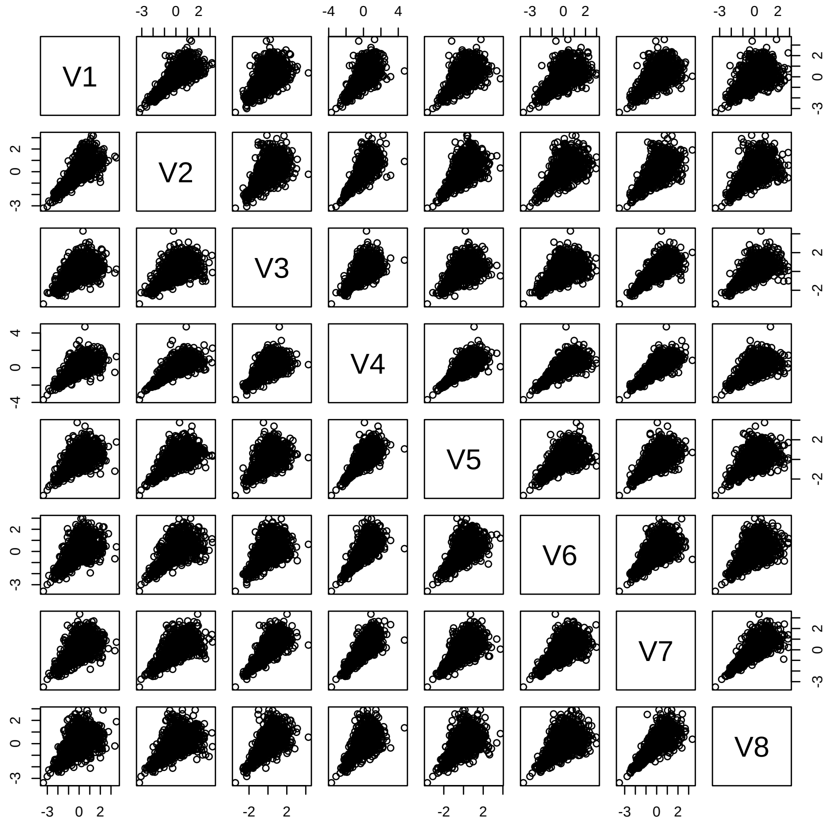


Figure 8: Scatter plot of the simulated normal-Clayton data.

jority of the pairs in the first 5 levels when they fitted an R-vine to the Euro Stoxx 50 data set. Hence, we fitted an R-vine structure consisting of Gaussian and t-copulae. Table 4 shows the fit for the resulting R-vine truncated after the first level in addition to the optimal one, and for the three other approaches. The optimal vine is truncated after level 8 and has 10 additional copulae set to independence. It outperforms the other methods in terms of the AIC score, while the Hill-Climbing algorithm gives a better BIC, due to the fact that it is sparser. The two constraint-based approaches perform even worse than the 1-level vine, that has fewer parameters. Hence, the performance of the Hill-Climbing algorithm and that of the vine approach are once more comparable, as one might expect from the elliptical pairwise dependencies. Had the two methods been compared including the marginal distributions, i.e. on the raw instead of the preprocessed data, the vine is likely to have been far superior to the Hill-Climbing graph.

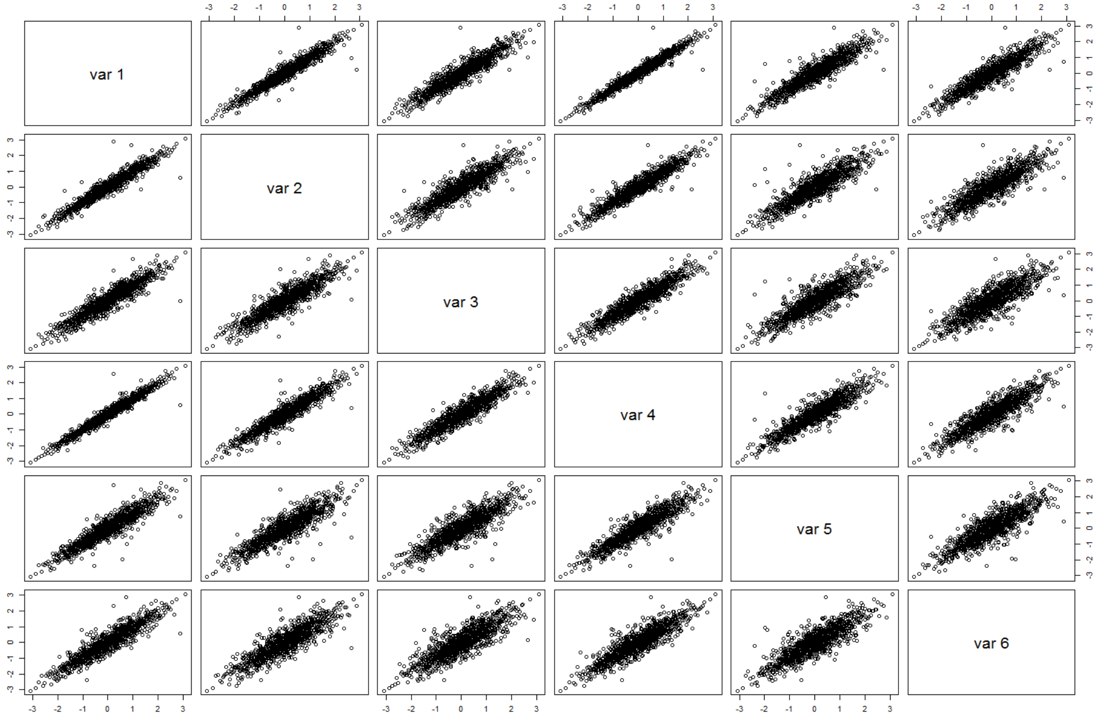


Figure 9: Pair-plot of the six first variables in the stock data.

Table 4: Stock data: log-likelihood, number of parameters, AIC and BIC for the different approaches.

| *Method* | *Log-lik* | *No. of par.* | *AIC* | *BIC* |
|---|---:|---:|---:|---:|
| Hill-climbing bic | -43984.44 | 354 | 88676.87 | 90408.88 |
| Max-Min HC bge | -44691.44 | 279 | 89940.89 | 91305.93 |
| Grow-shrink | -47621.82 | 224 | 95691.64 | 96787.59 |
| PC-algorithm | -51437.58 | 239 | 103353.20 | 104522.50 |
| Optimal vine ($K = 8$) | -43104.20 | 681 | 87570.39 | 90902.28 |
| 1-level vine | -45820.04 | 206 | 92052.08 | 93059.96 |

### *5.4. Precipitation data*

Our fourth data set consists of daily recordings of precipitation from January 1st, 1990 to December 31st, 2006, at 22 meteorological stations in Akershus county in Norway, obtained from the Norwegian Meteorological Institute. Subsets of these data set have previously been analysed in Berg and Aas (2009) and Hobæk Haff (2013). We have followed their example, and only modelled the positive precipitation. That is, we have discarded all observations for which at least one of the stations has recorded zero precipitation, which results in 4,928 data points. The precipitation data exhibit both serial dependence and seasonal patterns, typically with more precipitation during winter than summer. When the dry days are removed, most of the serial dependence is gone. The seasonal variation could be handled for instance by dividing the data into a summer and winter season, and treating these separately, as in Hobæk Haff et al. (2015). For simplicity, however, we have chosen to ignore the seasonal variation here.

To obtain approximately normal margins, we have transformed the margins first using the empirical distribution functions and then with the Gaussian quantile function. Figure 10 shows pair-plots of the six first variables after the preprocessing steps. There are strong indications of non-linear dependence. Hence, in this case, the regular vine approach is expected to be superior to the other ones.

From the appearance of the data, there are indications of upper, but not of lower tail dependence. Hence, the Gumbel copula seems to be a good candidate at the first level of the vine. Further, the data transformed with the estimated conditional cdfs from the preceding level seem to have an elliptical dependence structure (see Figure 11), meaning that the Student's t-copula would be reasonable for the remaining levels. Hence, we fitted an R-vine structure for which we allowed the pair-copulae to be either Gumbel or Student's t. As shown in Table 5, the fitted regular vine outperforms the other methods even when it is truncated at level 2. Again, the Hill-Climbing algorithm is far superior to the two constraint-based ones. However it produces a model with significantly more parameters and poorer scores than the 2-level vine. No doubt, the reason for this is the clearly non-Gaussian dependence structure. Moreover, notice that there is a large difference in the AIC-and BIC-values between the 2-level and the optimal vine, the latter being truncated after level 10 and having 3 pruned copulae. This means that although a 2-level vine performs very well compared to the other approaches, the fit to the data set is far from optimal.

### *5.5. Abalone data*

Finally, we have studied the Abalone data set (available at http://archive.ics.uci.edu/ml/datasets/Abalone), that has previously been used for Bayesian Network structure learning by e.g. Margaritis (2005), Steck (2008), and Ma et al. (2012). The data originate from a study by the Tasmanian Aquaculture and Fisheries Institute. An abalone is a kind of edible sea snail, the harvest of which is subject to quotas. These quotas are based partly on the age distribution of the abalones. To determine an abalone's age, one cuts

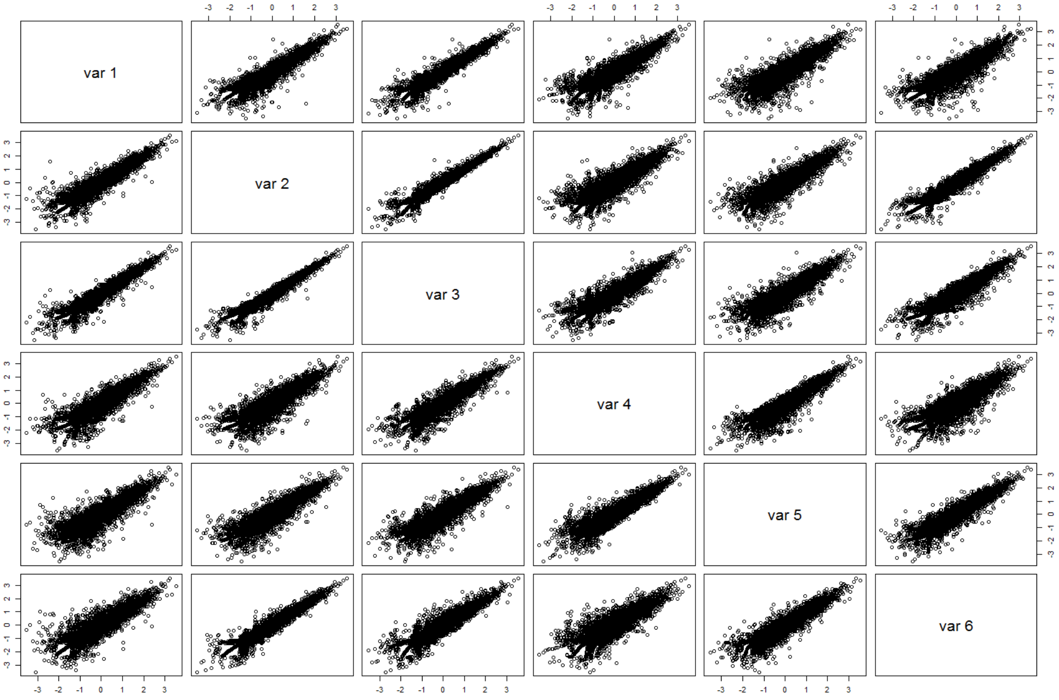
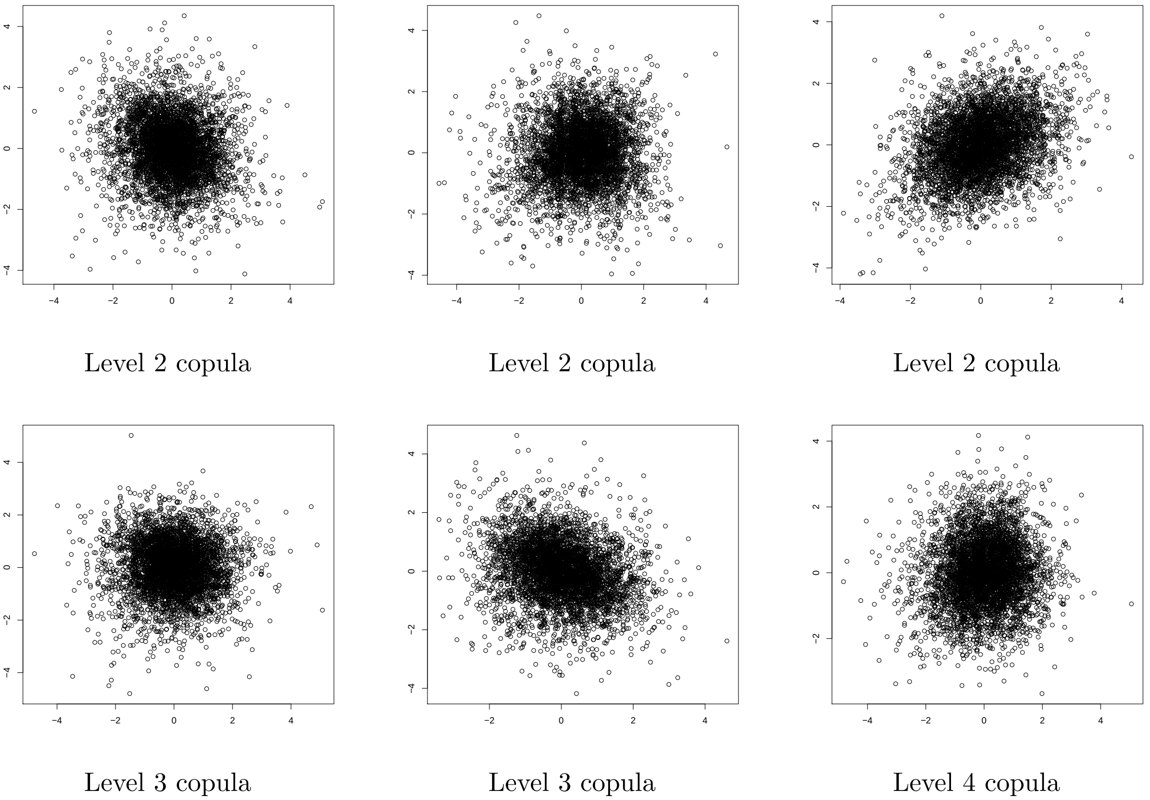


Figure 10: Pair-plot of the six first precipitation sites.

Table 5: Precipitation data: Log-likelihood, number of parameters, AIC and BIC for the different approaches.

| *Method* | *Log-lik* | *No. of par.* | *AIC* | *BIC* |
|---|---:|---:|---:|---:|
| Hill-climbing bic | -27589.85 | 165 | 55509.70 | 56582.64 |
| Max-Min HC bge | -32980.04 | 102 | 66164.08 | 66827.35 |
| Grow-shrink | -49105.43 | 79 | 98368.86 | 98882.57 |
| PC-algorithm | -51227.71 | 87 | 102629.40 | 103195.20 |
| Optimal vine ($K = 10$) | -10843.62 | 312 | 22311.24 | 24340.08 |
| 2-level vine | -26586.53 | 105 | 53383.07 | 54065.84 |



Level 2 copula

Level 2 copula

Level 2 copula

Level 3 copula

Level 3 copula

Level 4 copula

Figure 11: Data transformed with the estimated conditional cdfs from the preceding level, more specifically from levels 1, 2 and 3, representing copulae at level 2, 3 and 4, respectively, and then transformed to normal marginal distributions.

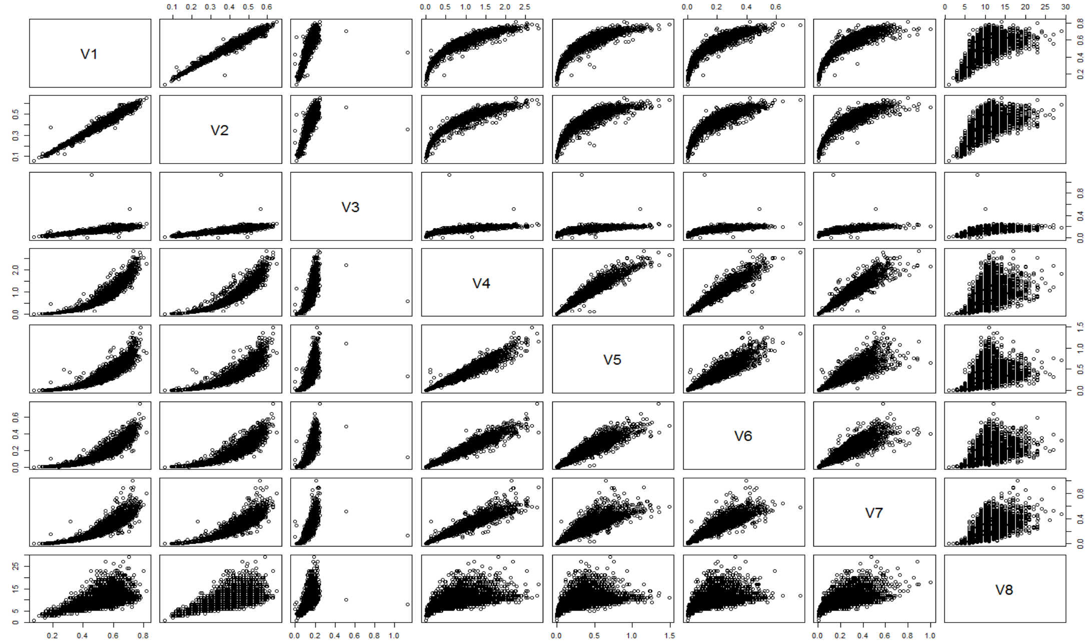


Figure 12: Scatter plot of the original Abalone data.

the shell through the cone, stains it, and counts the number of rings through a microscope. This is a highly time-consuming task. Hence, one would like to predict the age based on physical measurements such as weight and height. The Abalone data set was originally used for this purpose. It consists of 4,177 samples on the following 9 variables: Sex (V0), Length (V1), Diameter (V2), Height (V3), Whole weight (V4), Shucked weight (V5), Viscera weight (V6), Shell weight (V7) and Age measured by number of rings (V8).

We do not include the variable *Sex* in our study since it is a categorical variable. Note that the use of regular vines does not exclude discrete data; examples of discrete and mixed discrete vines may be found for instance in Panagiotelis et al. (2012) and Stöber et al. (2015). However, many of the methods become more complicated when discrete data are involved. Finally, since the variable *Rings* is an integer, we convert it to a continuous variable by adding Gaussian noise with expectation and standard deviation equal to 0 and 0.01, respectively.

Figure 12 shows scatter plots of the original data, while Figure 13 shows scatter plots of the data transformed with their empirical distribution functions. The variables are clearly far from Gaussian distributed marginally, and the pairwise dependencies are also distinctly non-Gaussian. Hence, a truncated regular vine with appropriate pair-copula families is expected to fit this data set much better than a Bayesian network based on an assumption of multivariate normality.

For this data set, we have not preprocessed the data to make them marginally normal before running the various algorithms. Hence, the effect of the margins is incorporated in the results. Figure 13 suggests that the data has lower, but not upper tail dependence. Hence, the Clayton copula seems to be a good candidate at the first level of the vine. Further, an inspection of the data transformed with the estimated conditional cdfs from the preceding level indicated that the Frank copula (that has neither upper nor lower tail dependence) would be reasonable for the remaining levels. We have therefore fitted an R-vine structure with Clayton and Frank copulae.

The results are shown in Table 6. The optimal vine was truncated after level 5 and no copula was pruned. As expected, the vine approach performs much better than the methods assuming a multivariate normal distribution, even with just one level. The third best method is the CBN with Gaussian copulae, which is due to its capability of modelling non-Gaussian margins. The performance of the PCBN is comparable to the one of the PC-algorithm. This is somewhat surprising since the PCBN, like the vine approach, is able to model both non-Gaussian margins and dependence structures, and moreover is not restricted to a subclass of chordal graphs. A closer look reveals that there are dependencies, such as between $V_2$ and $V_4$ and between $V_4$ and $V_7$, that are modelled as strong by the vine, and as conditional independence by the PCBN. This indicates that the PCBN, misses important edges. This is likely due to the use of the PC-algorithm, but the performance of the PCBN may also have been affected by the fact that we do not search for the optimal C- or D-vines in the computation of the Rosenblatt transforms needed in the independence tests.

Since the margins are far from Gaussian, we also applied the methods to the data transformed first using the marginal empirical distribution functions and then with the Gaussian quantile function. Table 7 shows the corresponding results. Since the CBN with normal margins is equivalent to Hill-climbing, it is not included in the table. The performance of the various methods is quite similar for the transformed as for the original data.

A further question is whether the improved fit also means better prediction results. Assume e.g. that we want to use the networks to study how the probability density of the number of rings of an abalone is affected by evidence about the values the other variables. Hence, for all approaches, we determine the probability density of $V_8$ given the seven other variables.

The probability density of the optimal vine is

$$\left[\prod_{i=1}^{8} f_i\right] c_{12}\, c_{24}\, c_{46}\, c_{45}\, c_{47}\, c_{73}\, c_{78}\, c_{56|4}\, c_{14|2}\, c_{27|4}\, c_{57|4}\, c_{34|7}\, c_{48|7}$$

$$\cdot c_{67|45}\, c_{17|24}\, c_{25|47}\, c_{58|47}\, c_{38|47}\, c_{26|457}\, c_{15|247}\, c_{28|457}\, c_{35|478}\, c_{16|2457}\, c_{18|2457}\, c_{23|4578},$$

and the conditional distribution of $V_8$ given all the others is given by

$$f_8 \frac{c_{78}\, c_{48|7}\, c_{58|47}\, c_{38|47}\, c_{28|457}\, c_{35|478}\, c_{18|2457}\, c_{23|4578}}{\int_0^1 c_{78}\, c_{48|7}\, c_{58|47}\, c_{38|47}\, c_{28|457}\, c_{35|478}\, c_{18|2457}\, c_{23|4578} du_8}.$$

The network determined by the Hill-Climbing algorithm has the pdf

$$\begin{aligned}&f(V_1)\,f(V_2|V_1)\,f(V_4|V_1,V_2)\,f(V_5|V_1,V_2,V_4)\,f(V_6|V_1,V_2,V_4,V_5)\\ &\cdot f(V_7|V_2,V_4,V_5,V_6)\,f(V_8|V_2,V_4,V_5,V_6,V_7)\,f(V_3|V_2,V_6,V_7,V_8),\end{aligned}$$

the one from Grow-Shrink is

$$\begin{aligned}&f(V_1)\,(V_4)\,f(V_7|V_4)\,f(V_6|V_1,V_4,V_7)\,f(V_3|V_6,V_7)\,f(V_6|V_1,V_4,V_6,V_7)\\ &\cdot f(V_2|V_1,V_3,V_7)\,f(V_8|V_3,V_4,V_5,V_7),\end{aligned}$$

the one from PC is

$$\begin{aligned}&f(V_2)\,f(V_5)\,f(V_8)\,f(V_1|V_2)\,f(V_6|V_1,V_5)\,f(V_3|V_2,V_8)\,f(V_4|V_2,V_5)\\ &\cdot f(V_7|V_2,V_3,V_4,V_5,V_6,V_8),\end{aligned}$$

the one from PCBN is

$$\left[\prod_{i=1}^{8} f_i\right]\, c_{12}\,c_{34}\,c_{46}\,c_{57}\,c_{38}\,c_{24|3}\,c_{47|5}\,c_{78|3}\,c_{45|23}\,c_{27|45} =$$

$$f(V_1)\,f(V_3)\,f(V_5)\,f(V_2|V_1)\,f(V_4|V_2,V_3,V_5)\,f(V_6|V_4)\,f(V_7|V_2,V_4,V_5)f(V_8|V_3,V_7),$$

and the one from CBN is

$$\begin{aligned}&f(V_4)\,f(V_7|V_4)\,f(V_1|V_4,V_7)\,f(V_3|V_4,V_7)\,f(V_6|V_1,V_3,V_4,V_7)\,f(V_8|V_3,V_5,V_6,V_7)\\ &\cdot f(V_2|V_1,V_3,V_4,V_7,V_8)\,f(V_5|V_1,V_2,V_4,V_7).\end{aligned}$$

Figure 14 shows the six versions of the probability density of the number of rings based on the other seven variables. In the upper row, we have conditioned on the 1%- and 5%-quantiles of the other variables, and in the lower row, on the medians and in the lower right corner on the 90%-quantiles. The figure shows that the densities from the vine approach, Hill-Climbing, Grow-Shrink, PCBN and CBN are quite similar if we condition on the medians of the other variables. However, for very large, and especially for very small quantiles, the vine and the PCBN densities are very similar, but they are significantly different from the other ones. A closer inspection of the subset of the raw data corresponding to values of the first 7 variables close to the corresponding empirical quantiles reveals that the vine-based conditional densities of $V_8$ are much closer to the truth than the others. Figure 15 shows histograms of the data points of $V_8$ corresponding to the other variables satisfying $q-h < V_i < q+h$, where $q$ is the quantile and $h$ is a box size that is appropriate for $q$, along with the estimated conditional densities using the optimal vine approach. This shows that the erroneous assumption of multivariate normality greatly affects the perception of the effect that the other variables have on the number of rings.

Judged by the AIC and BIC, the performance of the PCBN is not particularly good for this data set. However, the resulting conditional probability

Table 6: Abalone data: Log-likelihood, number of parameters, AIC and BIC for the different approaches.

| *Method* | *Log-lik* | *No. of par.* | *AIC* | *BIC* |
|---|---|---|---|---|
| Hill-climbing bic | 33899.00 | 21 | -67756.01 | -67622.92 |
| Max-Min HC bge | 33509.65 | 17 | -66985.31 | -66877.57 |
| Grow-shrink | 30069.67 | 17 | -60105.35 | -59997.61 |
| PC-algorithm | 27779.94 | 13 | -55533.89 | -55451.49 |
| CBN | 36228.13 | 22 | -72412.26 | -72272.84 |
| PCBN | 27118.17 | 10 | -54216.34 | -54152.97 |
| Optimal vine ($K = 5$) | 41876.51 | 25 | -83697.04 | -83519.57 |
| 1-level vine | 39494.52 | 7 | -78975.05 | -78930.68 |

Table 7: Abalone data: Log-likelihood, number of parameters, AIC and BIC for the different approaches applied on the data transformed to Gaussian margins.

| *Method* | *Log-lik* | *No. of par.* | *AIC* | *BIC* |
|---|---|---|---|---|
| Hill-climbing bic | -9320.246 | 38 | 18716.49 | 18957.31 |
| Max-Min HC bge | -57356.02 | 33 | 19978.91 | 20188.05 |
| Grow-shrink | -13118.79 | 33 | 26303.58 | 26303.58 |
| PC-algorithm | -23653.33 | 30 | 50722.11 | 47556.79 |
| PCBN | -20281.39 | 26 | 40614.78 | 40779.55 |
| Optimal vine ($K = 5$) | -5523.049 | 41 | 11134.08 | 11311.55 |
| 1-level vine | -7905.039 | 23 | 15856.07 | 15900.44 |

densities of the number of rings based on the other seven variables are close to the ones from the vine approach, despite the low number of parameters. Actually, $V_3$ and $V_7$ are the variables sharing the strongest dependence with $V_8$ in both the vine and the PCBN. This may explain why the latter captures the conditional distribution of $V_8$, given the others, well, but gives low AIC and BIC compared to the vine. Still, the PCBN required a CPU time that is more than $18,000$ higher than for the vine approach.

## 6. Conclusions

There are two main reasons for the success of graphical models. Firstly, graphs allow a powerful visual representation of relations between many variables, so that hierarchical, sequential, parallel or reinforcing effects can be identified and discussed. Secondly, graphical models allow a compact and coherent representation of the joint probability distribution, which is very convenient for inference on the model parameters and for knowledge propagation in the network. In this paper, we propose a new structure learning algorithm for Bayesian networks, based on pair-copula constructions.

With a few exceptions, existing structure learning algorithms for continuous variables involve either discretisation or the assumption of multivariate normality. The former strategy quickly becomes undoable in higher dimensions. On the other hand, the multivariate normal distribution fails to capture both nonlinear dependence and potential skewness and heavy tails in the margins. PCCs, which are hierarchical structures built merely from bivariate copulae, are able to portray all those characteristics, which are commonly found in multivariate data. Our structure learning algorithm uses a subclass of PCCs called regular vines. These are particularly appealing from an inferential and computational point of view. They have the limitation that they can represent only a subset of chordal graphs. However, we show in a number of applications that this disadvantage is more than compensated by the benefits of non-normality.

A potential weakness of regular vines is the number of parameters, which grows quickly with the dimension. In order to obtain more parsimonious models, we therefore employ two strategies; truncation and pruning. An R-vine is truncated by setting all copulae after a certain level to independence because they do not contribute significantly to the characterisation of the dependence in the data, while pruning consists in testing individual pairs of variables for conditional independence given some subset of the remaining variables. We prove that truncation always results in a certain kind of chordal graph. The pruning on the other hand must be performed in a specific order for the resulting graph to be triangulated. This is obtained by following the suggested algorithm.

To do the truncation, we use the method proposed by Brechmann et al. (2012). Combined with the structure selection of Dißmann et al. (2013), this constitutes a greedy and efficient algorithm, that produces reasonable graphical models, and can be characterised as a kind of score-based algorithm. When combined with pruning, it becomes a hybrid. In order to assess its performance, we have compared it to the score-based method Hill-Climbing, the hybrid method MMHC, the two constraint-based methods Grow-Shrink and PC and the two non-Gaussian methods CBN and PCBN on two synthetic and three real data sets. Our vine-based approach outperforms the other methods by far when the dependence structure of the data is clearly non-Gaussian. Moreover, it is competitive with the best-performing approach even when the true model is multivariate normal with a graph that cannot be exactly represented by an R-vine. In such cases, the estimated R-vine typically gives a graph with some extra edges, with the conditional independencies in question approximated by weak dependencies. Further, we have not taken the effect of potential non-normal margins into account, but only considered the dependence structure. The results could therefore be even more favourable for the vine approach.

Our method also outperforms the two non-Gaussian methods, that are not restricted to a certain subset of graphs. In particular, the PCBN, that has been shown to provide better fit than a regular vine when the true graph structure is known, seems to be inferior to the vine approach when the structure is unknown. This may be due to the fact that the PCBN uses a constraint-based structure finding algorithm. Moreover, it is computationally very expensive, and is therefore not an alternative to our method for high-dimensional problems.

## 7. Acknowledgements

We would like to thank Alex Lenkoski, Dorota Kurowicka, Gal Elidan, Markus Kalisch, Marco Scutari, Zongming Ma and Gianpaolo Scalia Tomba for contributing valuable information. We also thank the referees and Associate Editor for their help to improve this paper with their constructive comments and suggestions.

The work is sponsored by Statistics for Innovation, (sfi)$^2$. Kjersti Aas and Virginia Lacal also acknowledge The Norwegian Research Council (NFR) for financial support through the FINANSMARKEDSFONDET and FRINATEK programs.

Aas, K., Czado, C., Frigessi, A., Bakken, H., 2009. Pair-copula constructions of multiple dependence. Insurance, Mathematics and Economics 44, 182–198.

Acid, S., de Campos, L., Fernández-Luna, J., Rodríguez, S., Rodríguez, J., Salcedo, J., 2004. A comparison of learning algorithms for Bayesian networks. A case study based on an emergency medical service. Artificial Intelligence in Medicine 30, 215–232.

Banerjee, O., El Ghaoui, L., d'Aspremont, A., 2008. Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data. The Journal of Machine Learning Research 9, 485–516.

Bashar, A., Parr, G., McClean, S., Scotney, B., Nauck, D., 2010. Knowledge Discovery Using Bayesian Network Framework for Intelligent Telecommunication Network Management. In: Bi, Y., Williams, M.-A. (Eds.), Knowledge Science, Engineering and Management. Vol. 6291 of Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 518–529.

Bauer, A., Czado, C., 2015. Pair-copula bayesian networks. Journal of Computational and Graphical Statistics, DOI: DOI: 10.1080/10618600.2015.1086355.

Bauer, A., Czado, C., Klein, T., 2012. Pair-copula constructions for non-gaussian dag models. The Canadian Journal of Statistics 40, 86–109.

Bedford, T., Cooke, R., 2001. Probability density decomposition for conditionally dependent random variables modeled by vines. Annals of Mathematics and Artificial Intelligence 32, 245–268.

Bedford, T., Cooke, R., 2002. Vines - a new graphical model for dependent random variables. Annals of Statistics 30 (4), 1031–1068.

Beeri, C., Fagin, R., Maier, D., Yannakakis, M., 1983. On the desirability of acyclic database schemes. Journal of the ACM 30, 479–513.

Berg, D., Aas, K., 2009. Models for construction of higher-dimensional dependence: A comparison study. European Journal of Finance 15, 639–659.

Brechmann, E., Czado, C., Aas, K., 2012. Truncated regular vines in high dimensions with application to financial data. Canadian Journal of Statistics 40, 68–85.

Brechmann, E. C., 2013. Risk management with high-dimensional vine copulas: An analysis of the euro stoxx 50. Statistics & Risk Modeling 30, 307–342.

Brechmann, E. C., Joe, H., 2015. Truncation of vine copulas using fit indices. Journal of Multivariate Analysis 138, 19–33.

Chollete, L., Heinen, A., Valdesogo, A., 2009. Modeling international financial returns with a multivariate regime switching copula. Journal of Financial Econometrics 7, 437–480.

Chow, C. K., Liu, C. N., 1968. Approximating discrete probability distributions with dependence trees. IEEE Transactions on Information Theory 3, 462–467.

Cowell, R., Dawid, A., Lauritzen, S., Spiegelhalter, D., 1999. Probabilistic networks and experts systems. Springer, New York.

Czado, C., Schepsmeier, U., Min, A., 2012. Maximum likelihood estimation of mixed C-vines with application to exchange rates. Statistical Modelling 12, 229–255.

Deshpande, A., Garofalakis, M., Jordan, M. I., 2001. Efficient stepwise selection in decomposable models. In: Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence. UAI'01. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 128–135.

Dißmann, J., Brechmann, E. C., Czado, C., Kurowicka, D., 2013. Selecting and estimating regular vine copulae and application to financial returns. Computational Statistics and Data Analysis 59, 52–69.

Elidan, G., 2010a. Copula Bayesian Networks. In: Lafferty, J., Williams, C. K. I., Shawe-Taylor, J., Zemel, R., Culotta, A. (Eds.), Advances in Neural Information Processing Systems 23. pp. 559–567.

Elidan, G., 2010b. Inference-less Density Estimation using Copula Bayesian Networks. In: Grünwald, P., Spirtes, P. (Eds.), UAI. AUAI Press, pp. 151–159.

Genest, C., Favre, A.-C., 2007. Everything you always wanted to know about copula modeling but were afraid to ask. Journal of Hydrologic Engineering 12, 347–368.

Genest, C., Gerber, H. U., Goovaerts, M. J., Laeven, R. J., 2009. Editorial to the special issue on modeling and measurement of multivariate risk in insurance and finance. Insurance: Mathematics and Economics 44 (2), 143 – 145.

Genest, C., Ghoudi, K., Rivest, L., 1995. A semi-parametric estimation procedure of dependence parameters in multivariate families of distributions. Biometrika 82, 543–552.

Grønneberg, S., 2011. The copula information criterion and its implications for the maximum pseudo-likelihood estimator. In: Kurowicka, D., Joe, H. (Eds.), Dependence Modeling: Vine Copula Handbook. World Scientific Publishing Co.

Hanea, A., Kurowicka, D., Cooke, R., Ababei, D., 2010. Mining and visualizing ordinal data with non-parametric continuous bbn's. Computational Statistics and Data Analysis 54, 668–687.

Heinen, A., Valdesogo, A., 2009. Asymmetric CAPM dependence for large dimensions: the canonical vine autoregressive model. CORE discussion papers 2009069, Université catholique de Louvain, Center for Operations Research and Econometrics (CORE).

Hobæk Haff, I., 2012. Comparison of estimators for pair-copula constructions. Journal of Multivariate Analysis 110, 91–105.

Hobæk Haff, I., 2013. Estimating the parameters of a pair-copula construction. Bernoulli 19, 462–491.

Hobæk Haff, I., Aas, K., Frigessi, A., 2010. On the simplified pair-copula construction - simply useful or too simplistic? Journal of Multivariate Analysis 101 (5), 1296–1310.

Hobæk Haff, I., Frigessi, A., Maraun, D., 2015. How well do regional climate models simulate the spatial dependence of precipitation? an application of pair-copula constructions. Journal of Geophysical Research - Atmospheres 120, 2624–2646.

Hobæk Haff, I., Segers, J., 2015. Nonparametric estimation of pair-copula constructions with the empirical pair-copula. Computational Statistics & Data Analysis 84, 1–13.

Jensen, F., Jensen, F., 1994. Optimal junction trees. In: Uncertainty and Artificial Intelligence: Proceedings of the Tenth Conference. CA Morgan, San Mateo.

Joe, H., 1996. Families of m-variate distributions with given margins and m(m-1)/2 bivariate dependence parameters. In: L. Rüschendorf and B. Schweizer and M. D. Taylor (Ed.), Distributions with Fixed Marginals and Related Topics.

Joe, H., 1997. Multivariate Models and Dependence Concepts. Chapman & Hall, London.

Joe, H., 2005. Asymptotic effiency of the two stage estimation method for copula-based models. Journal of Multivariate Analysis 94, 401–419.

Kalisch, M., Mächler, M., Colombo, D., Maathuis, M. H., Bühlmann, P., 2012. Causal inference using graphical models with the R package pcalg. Journal of Statistical Software 47 (11), 1–26.

Koski, T., Noble, J., 2012. A review of bayesian networks and structure learning. Mathematica Applicanda 40, 53–103.

Kurowicka, D., 2011a. Optimal truncation of vines. In: Kurowicka, D., Joe, H. (Eds.), Dependence Modeling: Vine Copula Handbook. World Scientific Publishing Co.

Kurowicka, D., 2011b. Optimal truncation of vines. In: Kurowicka, D., Joe, H. (Eds.), Dependence Modeling: Vine Copula Handbook. World Scientific Publishing Co.

Kurowicka, D., Cooke, R., 2006a. Completion problem with partial correlation vines. Linear Algebra and its Applications 418, 188–200.

Kurowicka, D., Cooke, R., 2006b. Uncertainty Analysis with High Dimensional Dependence Modelling. Wiley, Chichester.

Lauritzen, S., 1996. Graphical models. Oxford University Press, Oxford.

Ma, J., Sun, Z.-Q., Chen, S., Liu, H.-H., 2012. Dependence tree structure estimation via copula. International Journal of Automation and Computing 9, 113–121.

Margaritis, D., 2003. Learning Bayesian Network Model Structure from Data. PhD Thesis, Carnegie-Mellon University, Pittsburgh, PA. Available as Technical Report CMU-CS-03-153.

Margaritis, D., 2005. Distribution-Free Learning of Bayesian Network Structure in Continuous Domains. In: Proceedings of the Twentieth National Conference on Artificial Intelligence (AAAI), Pittsburgh, PA.

Martinelli, G., Eidsvik, J., Hauge, R., 2013. Dynamic decision making for graphical models applied to oil exploration. European Journal of Operational Research, 688–702.

Min, A., Czado, C., 2010. Bayesian inference for multivariate copulas using pair-copula constructions. Journal of Financial Econometrics 8 (4), 511–546.

Min, A., Czado, C., 2011. Bayesian model selection for multivariate copulas using pair-copula constructions. Canadian Journal of Statistics 39, 239–258.

Morales-Napoles, O., 2011. Counting vines. In: Kurowicka, D., Joe, H. (Eds.), Dependence Modeling: Vine Copula Handbook. World Scientific Publishing Co.

Nelsen, R., 2006. An Introduction to Copulas, 2nd Edition. Springer, New York.

Panagiotelis, A., Czado, C., Joe, H., 2012. Pair copula constructions for multivariate discrete data. Journal of the American Statistical Association 107, 1063–1072.

Pearl, J., 1988. Probabilistic Reasoning in Intelligent Systems. Morgan Kaufmann, San Francisco, California.

Prim, R. C., 1957. Shortest connection networks and some generalizations. Bell System Technical Journal 36, 1389–1401.

Schwaighofer, A., Dejori, M., Tresp, V., Stetter, M., 2007. Structure learning with nonparametric decomposable models. In: Sa, J. M., Alexandre, L. A., Duch, W., Mandic, D. (Eds.), Artificial Neural Networks - ICANN 2007. Vol. 4668 of Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 119–128.

Scutari, M., 2010. Learning Bayesian Networks with the bnlearn R Package. Journal of Statistical Software 35, 1–22.

Shih, J., Louis, T., 1995. Inferences on the association parameter in copula models for survival data. Biometrics 51, 1384–1399.

Smith, M., Min, A., Czado, C., Almeida, C., 2010. Modeling longitudinal data using a pair-copula decomposition of serial dependence. Journal of the American Statistical Association 105, 1467–1479.

Spirtes, P., Glymour, C., Scheines, R., 2000. Causation, Prediction and Search, Second Edition (Adaptive Computation and Machine Learning). MIT Press, Cambridge.

Steck, H., 2008. Learning the bayesian network structure: Dirichlet prior versus data. In: Proceedings of the Twenty-Fourth Conference on Uncertainty in Artificial Intelligence (UAI2008).

Stöber, J., Hong, H., Czado, C. Ghosh, P., 2015. Comorbidity of chronic diseases in the elderly: Patterns identified by a copula design for mixed responses. Computational Statistics & Data Analysis 88, 28–39.

Thomas, A., Green, P., 2009. Enumerating the junction trees of a decomposable graph. Journal of Computational and Graphical Statistics 18, 930–940.

Tsamardinos, I., Brown, L., Aliferis, C., 2006. The max-min hill-climbing bayesian network structure learning algorithm. Machine Learning 65.

Uhler, C., Raskutti, G., Bühlmann, P., Yu, B., 2013. Geometry of faithfulness assumption in causal inference. Annals of Statistics 41, 436–463.

Vuong, Q. H., 1989. Likelihood ratio tests for model selection and non-nested hypotheses. Econometrica 57, 307–333.

Zhang, Q., Petrey, G., Deng, L., Qiang, L., Shi, Y., Thu, C. A., Bisikirska, B., Lefebvre, C., Accili, D., Hunter, T., Maniatis, T., Califano, A., Honig, B., 2012. Structure-based prediction of protein-protein interactions on a genome-wide scale. Nature 490, 556–560.

Zhu, M., Liu, S., Yang, Y., Liu, K., 2012. Using junction trees for structural learning of bayesian networks. Journal of Systems Engineering and Electronics 23, 286–292.

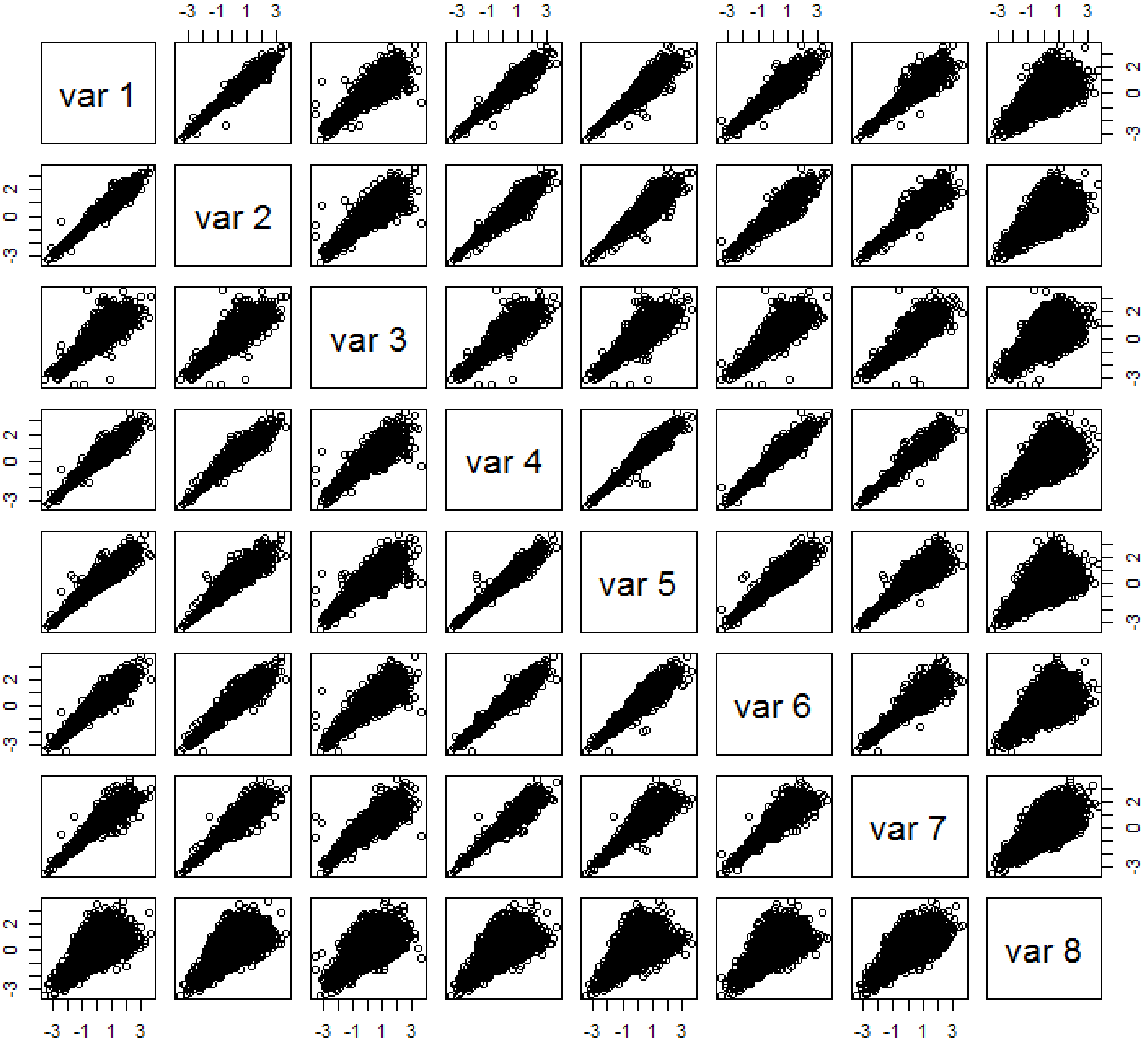


Figure 13: Scatter plot of the empirical copulae in the Abalone data set.

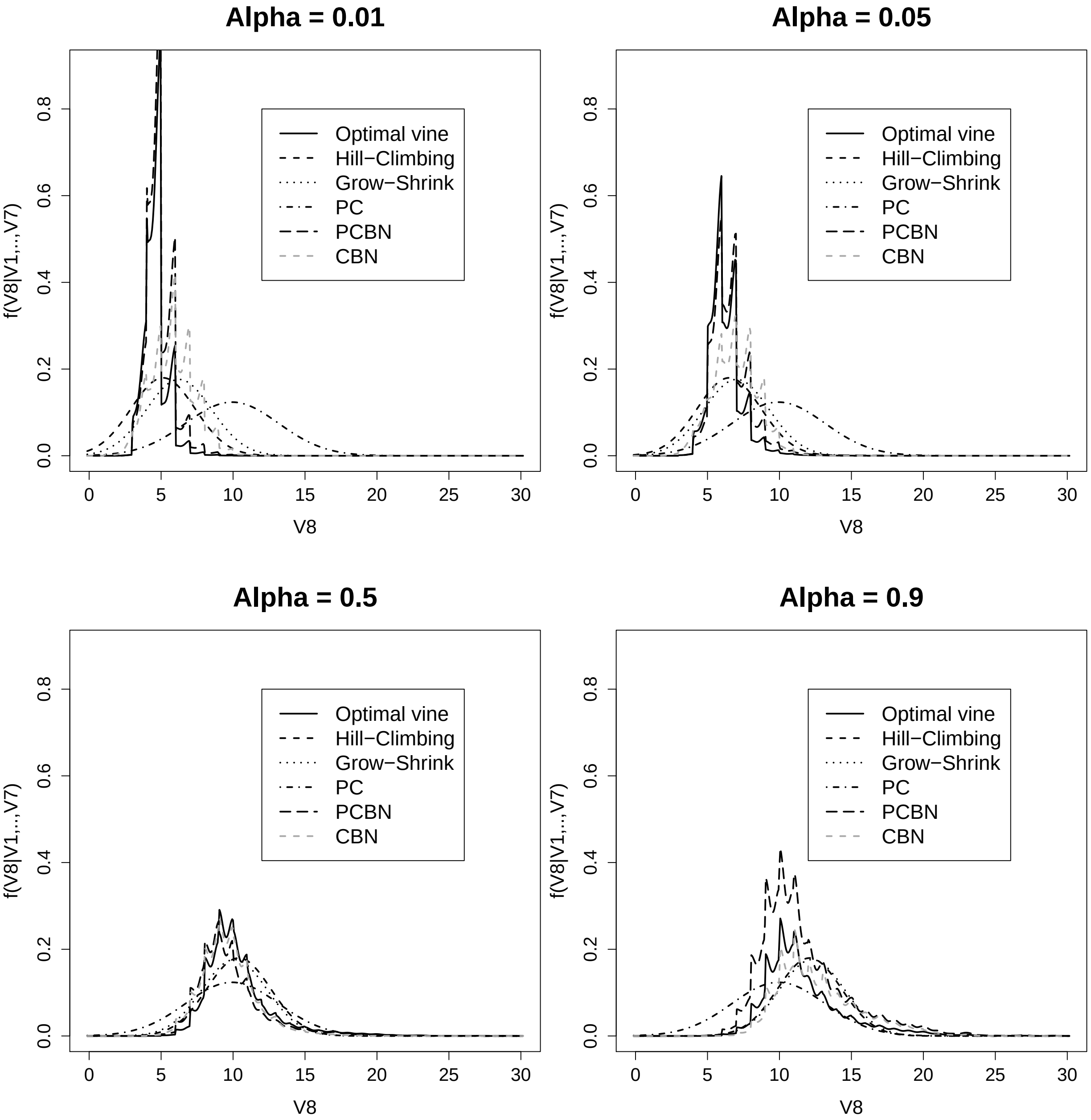


Figure 14: The probability density of the number of rings conditioned on the other variables. Upper row: Conditioned on the 1%- and 5%-quantiles of the other variables. Lower left panel: Conditioned on the medians. Lower right panel: Conditioned the 90%-quantiles.

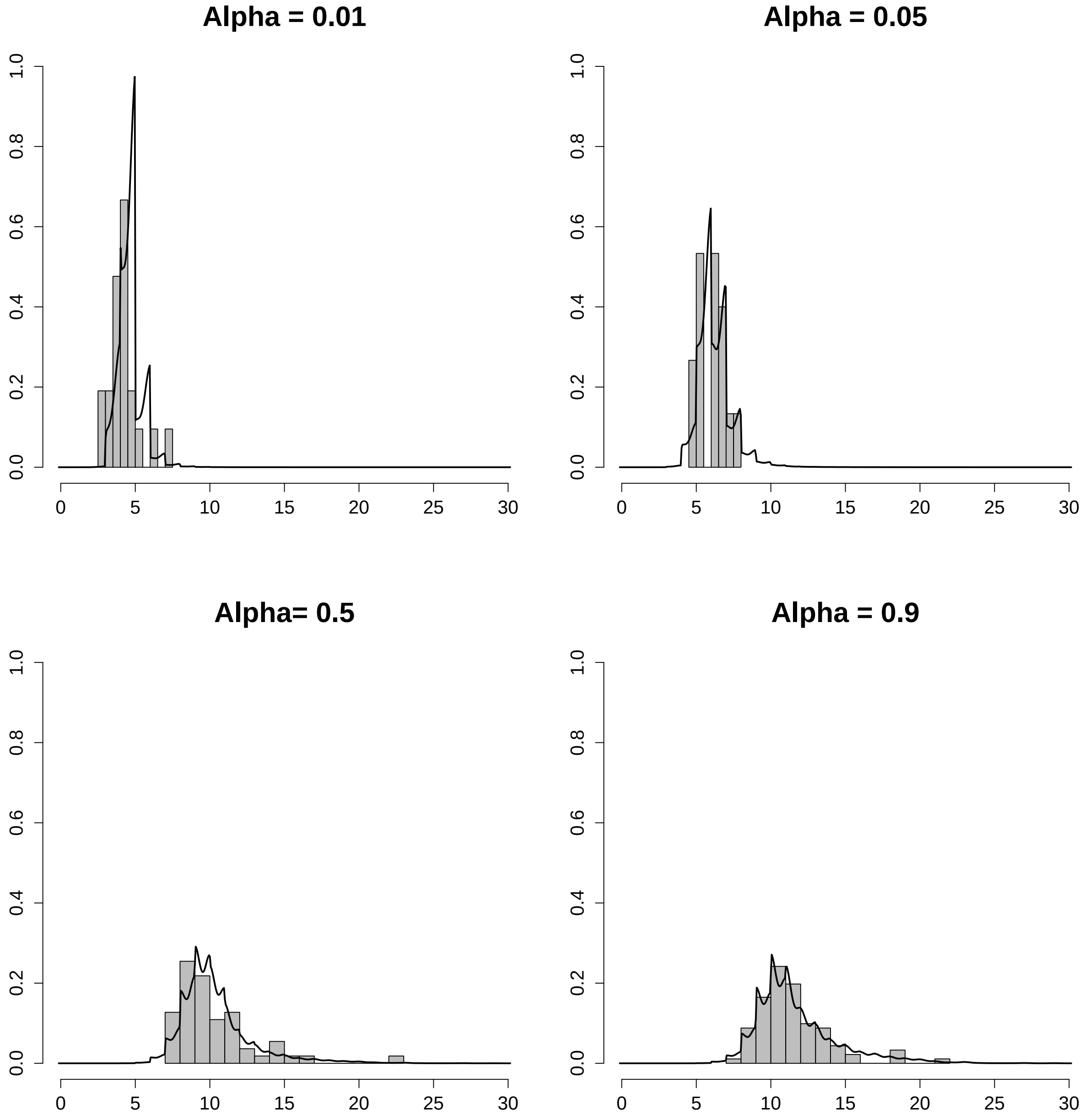


Figure 15: Histograms the number of the data points of $V_8$ corresponding to the other seven variables satisfying $q - h < V_i < q + h$ for the quantiles $q = 0.01, 0.05, 0.5, 0.9$ and an appropriate $h$ depending on $q$, along with the estimated conditional densities using the optimal vine approach. Upper row: Conditioned on the 1%- and 5%-quantiles of the other variables. Lower left panel: Conditioned on the medians. Lower right panel: Conditioned the 90%-quantiles.