mathematics

Article

Propagation Computation for Mixed Bayesian Networks Using Minimal Strong Triangulation

Yao Liu [1,†], Shuai Wang [1,†], Can Zhou [2] and Xiaofei Wang [1,*]

[1] Key Laboratory for Applied Statistics of MOE and School of Mathematics and Statistics, Northeast Normal University, Changchun 130024, China; liuy176@nenu.edu.cn (Y.L.); wangs343@nenu.edu.cn (S.W.)
[2] School of Statistics and Data Science, Nanjing Audit University, Nanjing 211815, China; zhouc@nau.edu.cn
[*] Correspondence: wangxf341@nenu.edu.cn
[†] These authors contributed equally to this work.

Abstract: In recent years, mixed Bayesian networks have received increasing attention across various fields for probabilistic reasoning. Though many studies have been devoted to propagation computation on strong junction trees for mixed Bayesian networks, few have addressed the construction of appropriate strong junction trees. In this work, we establish a connection between the minimal strong triangulation for marked graphs and the minimal triangulation for star graphs. We further propose a minimal strong triangulation method for the moral graph of mixed Bayesian networks and develop a polynomial-time algorithm to derive a strong junction tree from this minimal strong triangulation. Moreover, we also focus on the propagation computation of all posteriors on this derived strong junction tree. We conducted multiple numerical experiments to evaluate the performance of our proposed method, demonstrating significant improvements in computational efficiency compared to existing approaches. Experimental results indicate that our minimal strong triangulation approach provides a robust framework for efficient probabilistic inference in mixed Bayesian networks.

Keywords: Bayesian networks; strong junction trees; propagation computation

MSC: 62H22; 62F15

check for updates

Citation: Liu, Y.; Wang, S.; Zhou, C.; Wang, X. Propagation Computation for Mixed Bayesian Networks Using Minimal Strong Triangulation. *Mathematics* **2024**, *12*, 1925. https://doi.org/10.3390/math12131925

Academic Editor: Manuel Alberto M. Ferreira

Received: 13 May 2024
Revised: 16 June 2024
Accepted: 20 June 2024
Published: 21 June 2024

1. Introduction

Originating from artificial intelligence [1], Bayesian networks offer intuitive graphical representations for joint probability distributions and efficient algorithms for probabilistic reasoning. These networks are powerful tools for managing uncertainty and are applied across various fields, including bioinformatics [2,3], computational biology [4–6], and neuroscience [7], among others.

Over the past two decades, there has been a growing interest in Bayesian networks that incorporate both continuous and discrete variables [8]. This paper focuses on probabilistic models of such mixed networks, specifically those constrained to conditional linear Gaussian (CLG) distributions. Lauritzen [9] introduced a method for the exact local computation of means and variances in mixed Bayesian networks, but it suffered from numerical stability issues. Subsequently, Lauritzen and Jensen [10] developed an alternative, stable local computation scheme for these conditional Gaussian networks. However, this scheme is intricate, involving evaluations of matrix generalized inverses and recursive combinations of potentials. Cowell [11] proposed an alternative architecture utilizing exchange operations for variable elimination within an elimination tree. These exchange operations avoid complex matrix computations. Building upon the push and exchange operations, Madsen [12] extended lazy propagation, leveraging independence relations induced by evidence to enhance inference efficiency. The push operation is employed to obtain the posterior of continuous variables, while the exchange operation ensures the maintenance of a valid directed acyclic graph (DAG) structure during the belief update process.

Previous works have significantly improved the propagation computation for mixed Bayesian networks, primarily relying on strong junction trees. However, to our best knowledge, few works discuss how to construct appropriate strong junction trees. In this paper, we try to construct a strong junction tree under a minimal strong triangulation framework, which employs a polynomial-time minimal triangulation algorithm to minimize the number of fill-in edges. Additionally, we focus on the propagation computation of all posteriors. The process primarily involves three steps: finding triangulation and junction trees, performing inward and outward message passing, and computing posteriors for all non-evidence variables. This propagation computation based on junction trees facilitates the sharing of computations [13,14] among different posteriors. As suggested by Madsen [12], we applied exchange operations to eliminate both continuous and discrete variables during message passing. We conducted multiple simulation experiments, comparing our method with two existing methods: CGBayesNets [15] and BayesNetBP [16]. The results indicate that our method outperforms the existing methods in computational speed.

The primary contribution of this paper is the construction of a strong junction tree within a minimal strong triangulation framework. Specifically, we establish a connection between the minimal strong triangulation for marked graphs and the minimal triangulation for star graphs. We further propose a minimal strong triangulation method for the moral graph of mixed Bayesian networks and develop a polynomial-time algorithm to derive a strong junction tree from this minimal strong triangulation. Additionally, we present two theorems that guarantee our algorithm can successfully identify strong junction trees for mixed Bayesian networks.

This paper is organized as follows. In Section 2, the notation and definitions used in mixed Bayesian networks are reviewed. In Section 3, we discuss the minimal strong triangulation for marked graphs. Section 4 describes our method to find strong junction trees for mixed Bayesian networks by using minimal strong triangulations. In Section 5, we give a detailed analysis on exchange operation. Section 6 presents message passing on strong junction trees and posterior computation. Some numerical experimental results are shown in Section 7. Finally, Section 8 concludes this paper.

2. Basic Notions on Decomposition and Triangulation for Marked Graphs

Considering two graphs $G_1 = (V_1, E_1)$ and $G = (V, E)$, we call G_1 a subgraph of G if $V_1 \subseteq V$ and $E_1 \subseteq E$. Given a subset A of vertices in V, it induces a subgraph $G(A) = (A, E(A \times A))$ where $E(A \times A) = E \bigcap \{(a, b) | a \neq b \in A\}$. A graph is complete if every pair of distinct vertices is adjacent. A clique is a maximal vertex set inducing a complete graph. Given a graph $G = (V, E)$, a vertex subset S is called a separator for two disjoint vertex subsets A, B if every path in G between some $x \in A$ and $y \in B$ contains a vertex in S. We also say that S separates A and B in G. If $V = A \bigcup B \bigcup S$ and $A \bigcap B = \emptyset$, $A \bigcap S = \emptyset$, $B \bigcap S = \emptyset$, we call (A, B, S) as a partition of V.

Definition 1. *For a graph $G = (V, E)$, a partition (A, B, S) of V is a decomposition of G if A, B are separated by S in G, and S is complete.*

Let $G = (V = \Delta \bigcup \Gamma, E)$ be an undirected graph with two types of vertices, Δ and Γ, and this graph G is called a marked graph. Moral graphs of directed acyclic graphs (DAGs) of mixed Bayesian networks are marked graphs and those graph operations introduced below can be applied to moral graphs.

Definition 2. *A partition (A, B, S) of $V = \Delta \bigcup \Gamma$ is a strong decomposition of G if (A, B, S) is a decomposition of G and any of the three conditions $A \subseteq \Gamma$, $B \subseteq \Gamma$ or $S \subseteq \Delta$ holds.*

Leimer introduced the concept of marked graphs and their strong decomposition [17]. We specifically adopt strong decomposition in Definition 2 for marked graphs because it preserves the secondary structure [18] of the marked graph, as indicated by the closure [19] of marginal distributions during inward message passing.

A graph G is called triangulated if G is complete or there is a decomposition (A, B, S) of G such that $G(A \bigcup S)$ and $G(B \bigcup S)$ are triangulated. Similarly, a marked graph $G = (V = \Gamma \bigcup \Delta, E)$ is called strongly triangulated if G is complete or there is a strong decomposition (A, B, S) of G such that $G(A \bigcup S)$ and $G(B \bigcup S)$ are strongly triangulated. Given a marked graph G, we can construct a star graph from G by adding a vertex $\star$ to its vertex set and connecting $\star$ with every discrete variable. This star graph is denoted as $G^\star = (V \bigcup \{\star\}, E^\star)$, where $E^\star = \{(\delta, \star) | \delta \in \Delta\} \bigcup E$. For any subset $B \subseteq V$, the star graph $(G(B))^\star$ of the induced graph $G(B)$ is just the induced subgraph $G^\star(B \bigcup \{\star\})$ of the star graph $G^\star$. The following Proposition 1 [20] characterizes the equivalent relationship between the strongly triangulated marked graph and its star graph.

Proposition 1 ([20])**.** *A marked graph G is strongly triangulated if and only if $G^\star$ is triangulated.*

3. Minimal Strong Triangulation for Marked Graphs

A graph $H = (V, E \bigcup F)$ is called a triangulation of $G = (V, E)$ if H is triangulated. Any edge $uv \in F$ is referred to as a fill edge. A triangulation $H = (V, E \bigcup F)$ of $G = (V, E)$ is minimal if $(V, E \bigcup F')$ is not triangulated for any proper subset F' of F. For a marked graph $G = (V, E)$, a graph $H = (V, E \bigcup F)$ is called a strong triangulation of G if H is strongly triangulated. Let $\mathcal{H}_G$ be a set of all the strong triangulations of G. A strong triangulation $H = (V, E \bigcup F)$ of a marked graph $G = (V, E)$ is minimal if $(V, E \bigcup F')$ is not strongly triangulated for any proper subset F' of F. The following theorem establishes the relationship between minimal strong triangulations and minimal triangulations.

Theorem 1. *For a marked graph $G = (V = \Delta \bigcup \Gamma, E)$, $H = (V, E \bigcup F)$ is a minimal strong triangulation of G if and only if $H^\star$ is a minimal triangulation of $G^\star$.*

Proof. ($\Rightarrow$) If $H = (V, E \bigcup F)$ is a minimal strong triangulation of G, then H is strongly triangulated and $H^\star$ is triangulated. Denote E_1 as the edge set $\{(\delta, \star) | \delta \in \Delta\}$. Thus, $H^\star = (V \bigcup \{\star\}, E \bigcup E_1 \bigcup F)$ is a triangulation of $G^\star$. If $H_1^\star = (V \bigcup \{\star\}, E \bigcup E_1 \bigcup F_1)$ is also a triangulation of $G^\star$ and $F_1 \subsetneq F$, then $H_1 = (V, E \bigcup F_1)$ is a strong triangulation of G by Proposition 1. Since H is a minimal strong triangulation of G, it is a contradiction. Then $H^\star$ is a minimal triangulation of $G^\star$.

($\Leftarrow$) If $H^\star$ is a minimal triangulation of $G^\star$, then H is strongly triangulated and H is a strong triangulation of G. If $H_1 = (V, E \bigcup F_1)$ is also a strong triangulation of G and $F_1 \subsetneq F$, then $H_1^\star$ is triangulated by Proposition 1 and is a triangulation of $G^\star$. Since $H^\star$ is a minimal triangulation of $G^\star$, it is a contradiction. Thus, $H = (V, E)$ is a minimal strong triangulation of G. □

Theorem 1 indicates that a minimal strong triangulation for a marked graph G can be obtained from a minimal triangulation for the star graph G^*, which is an undirected graph. Specifically, we can apply the minimal triangulation algorithm MSC-M [21] to G^*, and subsequently obtain a minimal strong triangulation for G by dropping the star vertex.

Notice that the weight $w(G)$ of a marked graph G can be defined as the minimum sum of the weights of all the cliques over all possible strong triangulations of G, that is,

$$w(G) = \min_{H \in \mathcal{H}_G} \sum_{C \in \mathcal{K}_H} w(C),$$

where $\mathcal{K}_H$ is the set of all the cliques in the triangulation H. The weight $w(C)$ of a clique C is calculated as $s + sm(m+3)/2$ [22], where s represents the product of the number of states of the discrete nodes in C, and m denotes the number of continuous nodes in C. In $w(C)$, s has an exponential weight on the number of discrete nodes in C, while $m(m+3)/2$ acts as a polynomial weight on the number of continuous nodes. Unlike the optimal triangulation computation for $w(G)$ with an NP-hard computational complexity [23], our approach in this paper employs a polynomial-time minimal triangulation strategy to minimize the number of fill-in edges. Specifically, the time cost [21] of a minimal triangulation is $O(|V||E|)$, where $|V|$ and $|E|$ are the number of vertices and edges, respectively.

In this paper, we focus on directed acyclic graphs (DAGs) $G = (V, E)$, where V is marked into two groups, Δ and Γ. The vertices in Δ represent discrete variables, while those in Γ represent continuous variables. Since moral graphs of those DAGs with two types of variables are marked graphs, then we deduce the following corollary from Theorem 1:

Corollary 1. *Let G^m be the moral graph of a DAG G with two types of variables; H is a minimal strong triangulation of G^m if and only if $H^\star$ is a minimal triangulation of $(G^m)^\star$.*

4. Construction of Strong Junction Trees for Moral Graphs

Given a DAG G with two types of variables, its moral graph G^m is a marked graph. Utilizing the MCS-M algorithm [21], we can derive a minimal triangulation graph K of $(G^m)^\star$. In this algorithm, the star vertex $\star$ can be designated as the first numbered vertex. Consequently, the resulting graph K manifests as a star graph of some graph H, as $\star$ remains unconnected to any continuous vertex. Therefore, H is a minimal strong triangulation of G^m according to Corollary 1. The time complexity of the MCS-M algorithm for $(G^m)^\star$ is $O(|V_1||E_1|)$, where V_1 and E_1 denote the set of vertices and edges of $(G^m)^\star$, respectively.

The SMCS algorithm [17] shares a similar definition with the MCS algorithm [24], but always chooses discrete vertices if there is some discrete vertex with the maximum weight. Utilizing the SMCS algorithm enables us to acquire a D-numbering [25] denoted as $\{x_n, \cdots, x_1\}$ for H. Consequently, we obtain a D-ordered clique sequence $\{C_1, \cdots, C_m\}$ for H. Specifically, $V = C_1 \cup R_2 \cup \cdots \cup R_m$, where $C_1 = \{x_n, \cdots, x_{i_1}\}$ with $x_n \in \Delta$, $R_2 = \{x_{i_1-1}, \cdots, x_{i_2}\}$, and so forth, until $R_m = \{x_{i_{m-1}-1}, \cdots, x_{i_m} = x_1\}$. For any index k, $C_k = x_{i_k} \cup \text{madj}H(xi_k)$ forms a clique in H. The positions of $x_{i_1}, \cdots, x_{i_{m-1}}$ can be determined by the equivalent condition that $|\text{madj}_H(x_i)| + 1 \neq |\text{madj}_H(x_{i-1})|$.

Theorem 2. *The D-ordered clique sequence $\{C_1, \cdots, C_m\}$ obtained by the SMCS algorithm is also strong, i.e., $S_k \subseteq \Delta$ or $R_k \subseteq \Gamma$ for $2 \le k \le m$, where $S_k = C_k \bigcap (\bigcup_{i=1}^{k-1} C_i)$ and $R_k = C_k \setminus S_k$.*

Proof. Without loss of generalization, we assumes that $V \bigcap \Gamma \neq \emptyset$, $x_n \in \Delta$, and $m \geq 2$. Since H is strongly triangulated and the partition $(V \setminus C_m, R_m, S_m)$ is a decomposition of H, this partition is also a strong decomposition of H. Otherwise, there are $(V \setminus C_m) \bigcap \Delta \neq \emptyset$, $R_m \bigcap \Delta \neq \emptyset$, and $S_m \bigcap \Gamma \neq \emptyset$. Furthermore, there exists a path $[\alpha, \cdots, \beta]$ such that $\alpha \in \Delta \bigcap (V \setminus C_m)$, $\beta \in \Delta \bigcap R_m$ and other vertices are continuous. From Corollary 2.7 in [20], H is not a strong triangulated graph since α is not adjacent to β. It is a contradiction. So the partition $(V \setminus C_m, R_m, S_m)$ is also a strong decomposition of H.

If $R_m \subseteq \Gamma$, the result is trivial. If $R_m \bigcap \Delta \neq \emptyset$ and $S_m \bigcap \Gamma \neq \emptyset$, then $V \setminus C_m \subseteq \Gamma$ since $(V \setminus C_m, R_m, S_m)$ is a strong decomposition of H. Thus, $(V \setminus R_m) \bigcap \Delta \subseteq S_m$. Moreover, we have that $x_n \in S_m$. Otherwise, $x_n \in R_m$ means there is only one clique in H, and the result is trivial. Since x_n is the first vertex numbered by the SMCS algorithm, the cardinality of the discrete vertex is the same as that of the continuous vertex in C_m. Thus, the discrete vertex is always numbered earlier than the continuous vertex by the SMCS. The number of discrete vertex in R_m is always bigger than that of the continuous vertex in S_m. It is a contradiction. So, $S_m \subseteq \Delta$ if $R_m \bigcap \Delta \neq \emptyset$. Furthermore, since the subgraph $H(V \setminus R_m)$ is also strongly triangulated, the conclusion is obtained by recursive decompositions. □

As demonstrated above, the SMCS algorithm can be employed to generate a strong D-ordered clique sequence for a strong triangulation H, enabling the subsequent construction of a strong junction tree using this sequence. The time complexity for utilizing the SMCS algorithm [17] to find a strong D-ordered clique sequence and construct a strong junction tree is $O(|V_2| + |E_2|)$, where V_2 and E_2 denote the sets of vertices and edges of H, respectively.

From Corollary 1 and Theorem 2, we can devise the following Algorithm 1 with polynomial time complexity to find strong junction trees for mixed Bayesian networks:

Algorithm 1: Construct a Strong Junction Tree

```
Input  : A DAG $G$ with two types of variables
Output: A strong junction tree $T$ on $G$
1 Construct the Star Graph: Create the star graph $(G^m)^\star$ from the moral graph $G^m$
    of the DAG $G$;
2 Minimal Triangulation: Apply the MCS-M algorithm to the star graph $(G^m)^\star$
    with the star chosen as the first numbered vertex to derive its minimal
    triangulation graph $K$;
3 Obtain Strong Triangulation: Remove the star vertex $\star$ from $K$ to yield a
    minimal strong triangulation graph $H$ of $G^m$;
4 Construct Strong Junction Tree: Apply the SMCS algorithm to the minimal
  strong triangulation graph $H$ for generating a strong junction tree $T$.
```

5. The Exchange Operation for Mixed Bayesian Networks

In this paper, our focus lies on directed acyclic graphs (DAGs) denoted as $G = (V = \Delta \bigcup \Gamma, E)$. Here, the vertices of Δ represent discrete variables, while those of Γ represent continuous variables. To exploit conditional linear Gaussian (CLG) distributions for modeling, we make an additional assumption that no continuous variables have discrete children in G. A mixed Bayesian network $\mathcal{N} = (G, \mathcal{P}, \mathcal{F})$ with CLG distributions comprises a DAG G, a set of conditional probability distributions $\mathcal{P} = \{p(X|pa(X)) : X \in \Delta\}$, and a set of CLG density functions $\mathcal{F} = \{f(Y|pa(Y)) : Y \in \Gamma\}$, where $pa(Z)$ is the set of the parents of Z in G. A mixed Bayesian network $\mathcal{N}$ has a joint distribution of the form: $p(\Delta) \cdot f(\Gamma|\Delta) = \prod_{X\in\Delta} p(X|pa(X)) \cdot \prod_{Y\in\Gamma} f(Y|pa(Y))$. Let Y be a continuous variable and $I = pa(Y) \cap \Delta$ and $Z = pa(Y) \cap \Gamma$. Given variables $I = i$ and $Z = z$, the conditional distribution of Y is $N(\alpha(i) + \beta(i)^T z, \sigma^2(i))$, where $\alpha(i)$ is a table of real numbers, $\beta(i)$ is a table of $|Z|-$dimensional vectors, and $\sigma^2(i)$ is a table of non-negative values.

To compute all the marginal posteriors given evidence in mixed Besian networks, we employ propagation computation based on message passing within a strong junction tree structure. During message passing, barren variables [26] in the local structure can be directly eliminated without computation. And exchange operations [11] can avoid complex matrix operations, and make for finding barren variables [27]. In this section, we will discuss the usage of exchange operations in mixed Bayesian networks in detail.

Assume that $Z, Y, Y_1, \cdots, Y_n$ are continuous variables and there is a pair of conditional Gaussian distributions:

$$\mathcal{L}(Y|Y_1 = y_1, \cdots, Y_n = y_n) \sim N(a_Y + \sum_{i=1}^{n} b_Y^{(i)} y_i, \sigma_Y^2) \tag{1}$$

$$\mathcal{L}(Z|Y = y, Y_1 = y_1, \cdots, Y_n = y_n) \sim N(a_Z + \sum_{i=1}^{n} b_Z^{(i)} y_i + b_Z y, \sigma_Z^2) \tag{2}$$

The exchange operation, equivalent to Theorem 1 of the work [28], converts (1) and (2) into another pair of distributions:

$$\begin{aligned}&\mathcal{L}(Z|Y_1 = y_1, \cdots, Y_n = y_n)\\ &\sim N(a_Z + \sum_{i=1}^{n} b_Z^{(i)} y_i + b_Z(a_Y + \sum_{i=1}^{n} b_Y^{(i)} y_i), \sigma_Z^2 + b_Z^2 \sigma_Y^2)\end{aligned} \tag{3}$$

$$\mathcal{L}(Y|Z=z, Y_1=y_1,\cdots,Y_n=y_n) \sim N\left(\frac{h}{\sigma_Z^2+b_Z^2\sigma_Y^2}, \frac{\sigma_Z^2\sigma_Y^2}{\sigma_Z^2+b_Z^2\sigma_Y^2}\right) \tag{4}$$

where $h = b_Z^2\sigma_Y^2 z + \sigma_Z^2(a_Y + \sum_{i=1}^{n} b_Y^{(i)} y_i) - b_Z\sigma_Y^2(a_Z + \sum_{i=1}^{n} b_Z^{(i)} y_i)$. These formulae from Cowell [11] are essentially from the Bayes' theorem. (3) and (4) keep the same joint distribution as (1) and (2).

Let us consider exchange operations in mixed Bayesian networks $\mathcal{N}= (G,\mathcal{P},\mathcal{F})$. Assume that $Y, Z \in \Gamma$, $Y \in pa(Z)$ and there are no other directed paths from Y to Z in G. The parent set $pa(Y)$ of Y is partitioned into A_1 and A_2, where $A_1 = pa(Y)\bigcap\Delta = \{X_1',\cdots,X_s'\}$ and $A_2 = pa(Y)\bigcap\Gamma = \{U_1,\cdots,U_k\}$. Similarly, the parent set $pa(Z)$ of Z is partitioned into B_1 and B_2, where $B_1 = pa(Z)\bigcap\Delta = \{X_1'',\cdots,X_t''\}$ and $B_2 = pa(Z)\bigcap\Gamma = \{Y, W_1,\cdots,W_l\}$. The conditional density f_Y of Y and f_Z of Z have the following forms:

$$\begin{aligned} f_Y &= \mathcal{L}(Y|(X_1',\cdots,X_s') = i_{A_1}, U_1=u_1,\cdots,U_k=u_k) \\ &\sim N(a_Y(i_{A_1}) + \sum_{j=1}^{k} b_Y^{(j)}(i_{A_1})u_j, \sigma_Y^2(i_{A_1})) \end{aligned} \tag{5}$$

$$\begin{aligned} f_Z &= \mathcal{L}(Z|(X_1'',\cdots,X_t'') = i_{B_1}, Y=y, W_1=w_1,\cdots,W_l=w_l) \\ &\sim N(a_Z(i_{B_1}) + \sum_{j=1}^{l} b_Z^{(j)}(i_{B_1})w_j + b_Z(i_{B_1})y, \sigma_Z^2(i_{B_1})) \end{aligned} \tag{6}$$

Since there are no other directed paths from Y to Z except $Y \to Z$, variables in $pa(Z)\setminus\{Y\}$ are non-descendants of Y. By conditional independence relations in Bayesian networks, we have that

$$\mathcal{L}(Y|pa(Y), pa(Z)\setminus\{Y\}) = \mathcal{L}(Y|pa(Y)). \tag{7}$$

Similarly, we also have that

$$\mathcal{L}(Z|Y, pa(Z)\setminus\{Y\}, pa(Y)) = \mathcal{L}(Z|pa(Z)) \tag{8}$$

since any parent vertex of Y is non-descendant of Z. Denote $A_1\bigcup B_1$ as $C = \{X_1,\cdots,X_m\}$, and $A_2\bigcup B_2$ as $\{Y_1,\cdots,Y_n\}$. Thus from Equations (5)–(8), we have that:

$$\begin{aligned} &\mathcal{L}(Y|(X_1,\cdots,X_m) = i_C, Y_1=y_1,\cdots,Y_n=y_n) \\ &\sim N(a_Y(i_{A_1}) + \sum_{j=1}^{n} b_Y^{(j)}(i_{A_1})y_j, \sigma_Y^2(i_{A_1})) \end{aligned} \tag{9}$$

$$\begin{aligned} &\mathcal{L}(Z|(X_1,\cdots,X_m) = i_C, Y=y, Y_1=y_1,\cdots,Y_n=y_n) \\ &\sim N(a_Z(i_{B_1}) + \sum_{j=1}^{n} b_Z^{(j)}(i_{B_1})y_j + b_Z(i_{B_1})y, \sigma_Z^2(i_{B_1})). \end{aligned} \tag{10}$$

For any $1 \le t \le n$, thus $Y_t \in \{U_1,\cdots,U_k\}$ or $Y_t \in \{W_1,\cdots,W_l\}$. If $Y_m \notin \{U_1,\cdots,U_k\}$ for some $1 \le m \le n$, thus, $b_Y^{(m)} \equiv 0$. Similarly, if $Y_m \notin \{W_1,\cdots,W_l\}$ for some $1 \le m \le n$, thus $b_Z^{(m)} \equiv 0$. The exchange operation can convert Equations (5) and (6) into the following pair f_Y', f_Z' of conditional densities:

$$\begin{aligned} f_Z' &= \mathcal{L}(Z|(X_1,\cdots,X_m) = i_C, Y_1=y_1,\cdots,Y_n=y_n) \\ &\sim N(a_Z(i_{B_1}) + \sum_{j=1}^{n} b_Z^{(j)}(i_{B_1})y_j + b_Z(i_{B_1})(a_Y(i_{A_1}) \\ &\quad + \sum_{j=1}^{n} b_Y^{(j)}(i_{A_1})y_j), \sigma_Z^2(i_{B_1}) + b_Z^2(i_{B_1})\sigma_Y^2(i_{A_1})) \end{aligned} \tag{11}$$

$$\begin{aligned} f'_Y &= \mathcal{L}(Y|(X_1,\cdots,X_m)=i_C, Z=z, Y_1=y_1,\cdots,Y_n=y_n) \\ &\sim N(\frac{h'}{\sigma_Z^2(i_{B_1})+b_Z^2(i_{B_1})\sigma_Y^2(i_{A_1})}, \frac{\sigma_Z^2(i_{B_1})\sigma_Y^2(i_{A_1})}{\sigma_Z^2(i_{B_1})+b_Z^2(i_{B_1})\sigma_Y^2(i_{A_1})}) \end{aligned} \tag{12}$$

where $h' = b_Z^2(i_{B_1})\sigma_Y^2(i_{A_1})z + \sigma_Z^2(i_{B_1})(a_Y(i_{A_1}) + \sum\limits_{j=1}^{n} b_Y^{(j)}(i_{A_1})y_j) - b_Z(i_{B_1})\sigma_Y^2(i_{A_1})\ (a_Z(i_{B_1}) + \sum\limits_{j=1}^{n} b_Z^{(j)}(i_{B_1})y_j)$.

Following the exchange operation, we reconstruct the mixed Bayesian network G into G' based on the altered conditional distributions of Y and Z. Specifically, the new network G' is the same as the original G except that $Z \to Y$ and $X \to Z, X' \to Y$ in G' for any $X \in pa(Y)$ and $X' \in pa(Z) \setminus \{Y\}$. And since $Y \in pa(Z)$ and no other directed path exists from Y to Z in G, it can be verified that the new network G' is also a DAG. Thus, $\mathcal{N}' = (G', \mathcal{P}, \mathcal{F}')$ is also a mixed Bayesian network with the same joint distribution with $\mathcal{N}$, where $\mathcal{F}' = \mathcal{F} \bigcup \{f'_Y, f'_Z\} \setminus \{f_Y, f_Z\}$. The process from G to G' is called arc-reversal. By using arc-reversals step by step, any continuous variable can become a node with no children. Thus, the exchange operation provides us a method to eliminate continuous variables and avoids matrix operations [11].

Now, we consider handling discrete variables by using the exchange operation. Let $Q \in \Delta$ with parent set $pa(Q) = \{X_1, \cdots, X_n\} \subseteq \Delta$ and let $X \in \Delta$ with parent set $pa(X) = \{Q, X_1 \cdots, X_n\} \subseteq \Delta$. $p(Q|X_1, \cdots, X_n)$ and $p(X|Q, X_1, \cdots, X_n)$ are the corresponding probabilities of Q and X, respectively. The exchange operation converts the above pair of conditional probabilities into $p(X|X_1, \cdots, X_n)$ and $p(Q|X, X_1, \cdots, X_n)$, which maintain the same joint probability distribution of the original pair:

$$p(X|X_1,\cdots,X_n) = \sum_X p(X|Q,X_1,\cdots,X_n)p(Q|X_1,\cdots,X_n), \tag{13}$$

$$p(Q|X,X_1,\cdots,X_n) = \frac{p(X|Q,X_1,\cdots,X_n)p(Q|X_1,\cdots,X_n)}{p(X|X_1,\cdots,X_n)}. \tag{14}$$

From (13) and (14), the exchange operation converts p_Q and p_X into the following two conditional probabilities p'_Q, p'_X:

$$p'_X = p(X|X_1,\cdots,X_n) = \sum_Q p(X|pa(X))p(Q|pa(Q)), \tag{15}$$

$$p'_Q = p(Q|X,X_1,\cdots,X_n) = \frac{p(X|pa(X))p(Q|pa(Q))}{p(X|X_1,\cdots,X_n)}, \tag{16}$$

by the assumption that there are no other directed paths from Q to X in G.

After the exchange operation, similar to the case of continuous variables, the mixed Bayesian network G can be reconstructed into G' according to the changed conditional distributions of Q and X. G' is the same as G except that $X \to Q$ and $X_1 \to X, X_2 \to Q$ in G' for any $X_1 \in pa(Q)$ and $X_2 \in pa(X) \setminus \{Q\}$, and G' is also a DAG. Thus, $\mathcal{N}' = (G', \mathcal{P}', \mathcal{F})$ is also a mixed Bayesian network with the same joint distribution with $\mathcal{N}$, where $\mathcal{P}' = \mathcal{P} \bigcup \{p'_Y, p'_Z\} \setminus \{p_Y, p_Z\}$. The process from G to G' is also called arc-reversal.

In this paper, arc-reversals for discrete variables with no continuous children are only used in message passing for further finding barren variables, and the lazy propagation [12,27,29] is also utilized to improve the computational efficiency. Following message passing, direct variable eliminations [30] are used to eliminate discrete variables in the marginal posterior computation.

6. Message Passing and Posterior Computing

Let $H(p)$ denote the head variable of $p \in \mathcal{P}$ and $T(p)$ the tail variables of $p \in \mathcal{P}$. For instance, if $p(X|X_1, \cdots, X_n)$, then $H(p) = \{X\}$ and $T(p) = \{X_1, \cdots, X_n\}$. Let

$dom(p) = H(p) \cup T(p)$ denote the domain of p. $H(f), T(f)$ and $dom(f)$ are similarly defined for $f \in \mathcal{F}$. A potential π is represented as $(\mathcal{P}_\pi, \mathcal{F}_\pi)$, where $\mathcal{P}_\pi$ denotes the set of conditional probability and $\mathcal{F}_\pi$ denotes that related to conditional density. The combination of two potentials, $\pi_1 = (\mathcal{P}_{\pi_1}, \mathcal{F}_{\pi_1})$ and $\pi_2 = (\mathcal{P}_{\pi_2}, \mathcal{F}_{\pi_2})$, is defined as $\pi_1 \bigotimes \pi_2 = (\mathcal{P}_{\pi_1} \bigcup \mathcal{P}_{\pi_2}, \mathcal{F}_{\pi_1} \bigcup \mathcal{F}_{\pi_2})$.

Let $T = (\mathcal{C}, E_T)$ be a strong junction tree of H, which is a minimal strong triangulation of G^m. For any $p \in \mathcal{P}$, choose a clique $C \in \mathcal{C}$ such that $H(p) \bigcup T(p) \subseteq C$ and allocate p to C. For any $f \in \mathcal{F}$, choose a clique $C \in \mathcal{C}$ such that $H(f) \bigcup T(f) \subseteq C$ and allocate f to C. For any clique $A \in \mathcal{C}$, A has an original potential $\pi_A = (\mathcal{P}_{\pi_A}, \mathcal{F}_{\pi_A})$, where $\mathcal{P}_{\pi_A} = \{p \in \mathcal{P}|$ p is allocated to $A\}$ and $\mathcal{F}_{\pi_A} = \{f \in \mathcal{F}|\, f$ is allocated to $A\}$.

Assume that two cliques $A, B \in \mathcal{C}$ are adjacent on T and the current potential π'_A of A is $(\mathcal{P}_{\pi'_A}, \mathcal{F}_{\pi'_A})$. We define $D = (\bigcup_{p \in \mathcal{P}_{\pi'_A}} dom(p)) \bigcup (\bigcup_{f \in \mathcal{F}_{\pi'_A}} dom(f))$ as the union of domains of all variable in $\mathcal{P}_{\pi'_A}$ and $\mathcal{F}_{\pi'_A}$, and $R = D \setminus S$, where $S = A \bigcap B$. Let G' be a DAG induced by the current potential π'_A. The information potential $\pi_{A \to B}$ from A to B is computed by the following operations:

(1) Find barren variables N [12] with respect to set S, evidence ϵ and DAG G'. Remove conditional probabilities and densities from π'_A, whose head variable is contained in N, and the remaining potential is denoted by π_1. Remove N from G' and the remaining graph is denoted by $G'' = G'(D \setminus N)$. Let X be the set d-connected to S about $\epsilon \bigcap R$ in G''. Remove conditional probabilities and densities from π_1, whose domain has no variables in $S \bigcup X$. The resulting potential is denoted by π_2. The set X can be partitioned into X_d and X_c, which denote discrete and continuous non-evidence variables, respectively.

(2) Eliminate variables in X_c by exchange operations. Based on these operations, conditional densities of π_2 are changed, and those densities whose head variable is contained in X_c are removed from π_2. The remaining potential is denoted by π_3.

(3) Eliminate variables in X_d with no continuous children by exchange operations. Let $\Phi(x)$ denote $\{p \in \pi_3 | x \in dom(p)\}$. If $|\bigcup_{p \in \Phi(x)} dom(p)| > \delta$ for some proper δ, exchange operations for eliminating x are delayed. Based on these operations, conditional probabilities of π_3 are changed, and those probabilities whose head variable x is contained in X_d and $|\bigcup_{p \in \Phi(x)} dom(p)| \leq \delta$ are removed from π_3.

By these operations, the remaining potential is denoted by $(\mathcal{P}_{\pi'_A}, \mathcal{F}_{\pi'_A})^{\downarrow B}$. Thus, $\pi_{A \to B} = (\mathcal{P}_{\pi'_A}, \mathcal{F}_{\pi'_A})^{\downarrow B} = {\pi'_A}^{\downarrow B}$. When considering the information from A to B, the current potential π'_A is the combination of the original potential of A with all the information from the clique $C \in adj_T(A) \setminus \{B\}$ to A. Then, $\pi_{A \to B} = (\pi_A \bigotimes (\bigotimes_{C \in adj_T(A) \setminus \{B\}} \pi_{C \to A}))^{\downarrow B}$.

$\forall X \in V$, there is a clique A containing X. To compute the posterior distribution $P(X|\epsilon = e)$ of X,

$$\pi = \pi_A \bigotimes (\bigotimes_{C \in adj_T(A)} \pi_{C \to A}) \tag{17}$$

$$D = (\bigcup_{p \in \mathcal{P}_\pi} dom(p)) \bigcup (\bigcup_{f \in \mathcal{F}_\pi} dom(f)) \tag{18}$$

$$\phi = \prod_{p \in \mathcal{P}_\pi} p \cdot \prod_{f \in \mathcal{F}_\pi} f \tag{19}$$

$$P(X|\epsilon = e) = \frac{\sum_{D \cap \Delta \setminus \{\epsilon \bigcup \{x\}\}} \cdots \int_{D \cap \Gamma \setminus \{\epsilon \bigcup \{x\}\}} \phi}{\sum_{D \cap \Delta \setminus \epsilon} \cdots \int_{D \cap \Gamma \setminus \epsilon} \phi} \tag{20}$$

For Equation (20), exchange operations are performed on continuous variables, and the sum-out procedure [14] is used to eliminate discrete variables. Thus we can obtain the following Algorithm 2 for computing all the posteriors:

Algorithm 2: Propagation Computation for All the Posteriors.

```
Input  : A mixed Bayesian network $\mathcal{N}= (G, \mathcal{P}, \mathcal{F})$ with given evidence
         variable set $\epsilon = e$
Output: All the posteriors for non-evidence variables

Construct a strong junction tree $T$ on $G$ by Algorithm 1;
Choose a clique in the strong junction tree $T$ as the root $R$;
Perform inward message passing to the root such that root $R$ receives
  information from its neighbors;
Perform outward message passing from the root such that each clique in $T$
  receives information from its neighbors;
Compute the posterior $P(X|\epsilon = e)$ for each non-evidence variable $X$ by
  Equations (17)–(20).
```

7. Numerical Experiment

Our experiments were conducted on a collection of randomly generated CLG sparse Bayesian networks consisting of n variables and $\lfloor n \log(\log(n)) \rfloor$ edges. Referring to the work [12], we set the number of variables n to {50, 75, 100} and the ratio of continuous variables to {0, 0.25, 0.5, 0.75, 1}. With this simulation setting, we can evaluate the performances of our method across various sizes of mixed Bayesian networks and different proportions of continuous variables. For each network size, each proportion of continuous variables, and each size of the evidence set, one hundred networks were randomly generated, with each discrete variable having two states. These experiments were conducted using C++ on a PC equipped with an Intel Core i9-9900X CPU running at 3.5GHz. We mainly utilized three standard libraries in C++: <vector>, <random>, and <algorithm>. The <vector> library is a versatile container for dynamic arrays but often allocates more memory than necessary, leading to potential overhead. The <random> library offers a comprehensive set of tools for generating random numbers but may produce inconsistent sequences across different platforms due to varying implementations. The <algorithm> library provides a rich set of functions for operations on data sequences, like sorting and searching, but does not directly support complex data structures such as graphs and trees.

In our experiments, the average time required to find minimal strong triangulations and strong junction trees for networks with 50, 75, and 100 variables is approximately 0.0005, 0.0014, and 0.0031 s, respectively. These results demonstrate that the algorithm outlined in Section 4 is efficient in producing strong junction trees for mixed Bayesian networks. Subsequent belief updates based on these strong junction trees are also carried out in our experiments.

Figures 1–3 depict the average time costs of the propagation computation of our method, including both message passing and the posterior computation for all non-evidence variables, across networks with 50, 75, and 100 variables. Meanwhile, Figures 4–6 display the median time costs of the propagation computation. As shown in Figures 1–6, the speed of belief updates in networks only containing continuous variables remains consistently fast, whether considering the average or median time costs. Moreover, as shown in Figures 5 and 6, at low numbers of instantiations, a low proportion of continuous variables tends to increase the median time costs for belief updates in Bayesian networks. This is due to the state space size of the largest clique potentially being larger in networks with a lower proportion of continuous variables compared to networks with a higher proportion, under similar conditions. However, once the number of instantiations exceeds half of the variables, the discrepancy among networks with varying proportions of continuous variables becomes small. This reduction occurs because the average largest discrete domain size decreases as the size of the evidence set increases [12].

A notable observation when comparing the average time costs depicted in Figure 3 with the median time costs shown in Figure 6 is the significant disparity between the two measures. This disparity arises primarily from the considerable variance in time costs resulting from the random sampling of Bayesian networks. For instance, in networks with 100 variables, a continuous variable proportion of 0.5, and 10 pieces of evidence, the average time is 54.3024 s with a standard deviation of 538.1081, while the median time is 0.1355 s. It is evident that this huge average time is caused by certain outliers in the time cost for 100 randomly generated networks. The reason for this abnormal result may lie in the generation of large-scale cliques with discrete variables under certain random seeds when the size of the evidence set is ten.

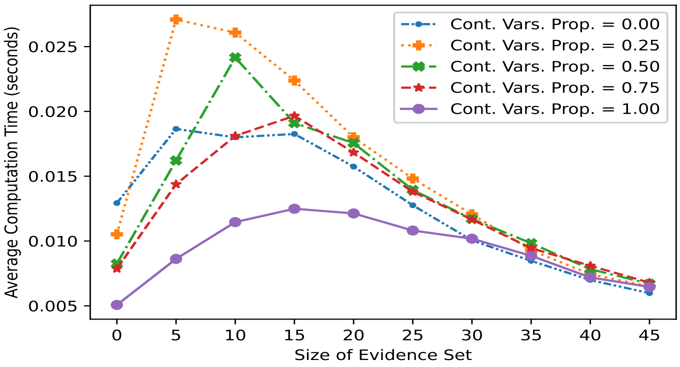

Figure 1. Average time in seconds for belief update in networks with 50 variables.

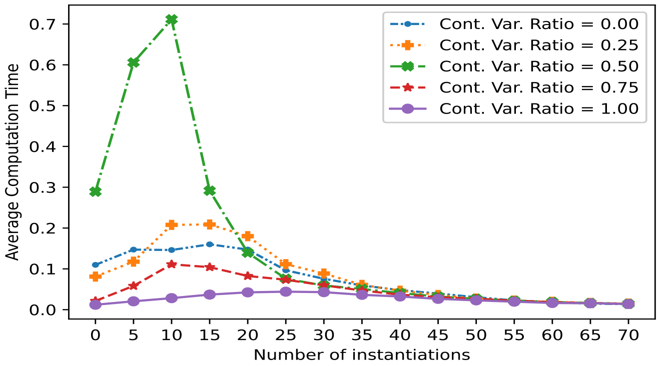

Figure 2. Average time in seconds for belief update in networks with 75 variables.

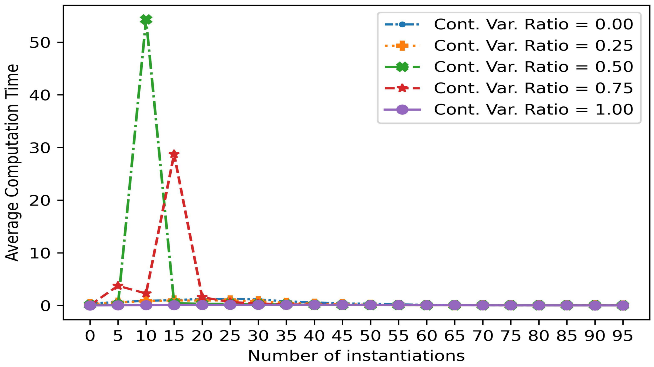

Figure 3. Average time in seconds for belief update in networks with 100 variables.

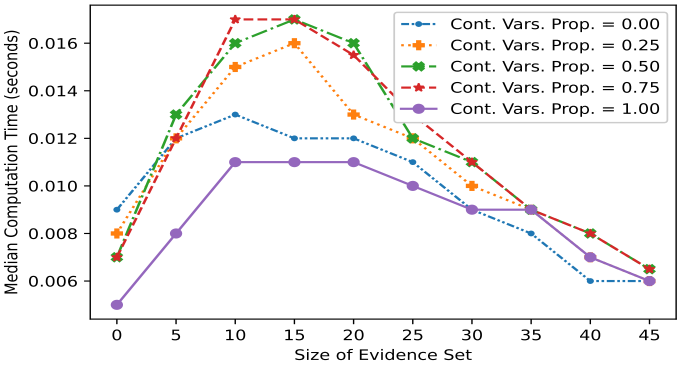

Figure 4. Median time in seconds for belief update in networks with 50 variables.

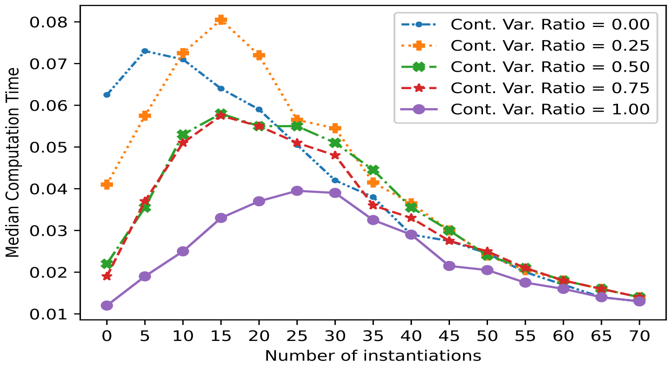

Figure 5. Median time in seconds for belief update in networks with 75 variables.

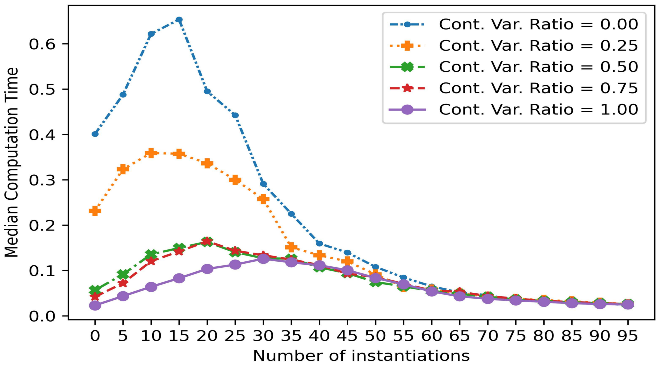

Figure 6. Median time in seconds for belief update in networks with 100 variables.

For a further analysis of the efficiency of our method, we conducted a comparative study with two existing packages: CGBayesNets (a MATLAB package [15]) and BayesNetBP (an R package [16]). Figure 7 illustrates the average time costs on propagation computation of three methods for 100 randomly generated networks, each comprising 50 variables. In the left panel, the size of the evidence set is fixed at five, while in the right panel, the proportion of continuous variables is set to 0.5. As shown in both panels, it is evident that our method outperforms the other two methods in terms of computational speed.

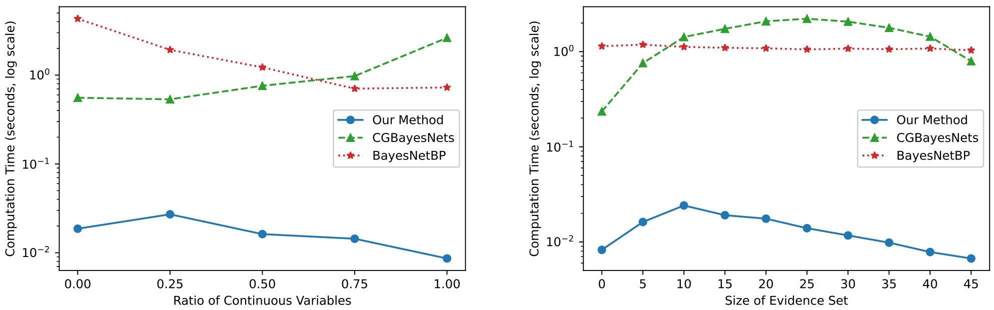

Figure 7. Time costs of propagation computation for our method, CGBayesNets, and BayesNetBP.

8. Conclusions

The Bayesian network is a prominent predictive modeling framework, where its nodes often represent genes or proteins interconnected by edges denoting relationships. These edges, informed by experimental data, impart a predictive capability to Bayesian computational methods [15,31], capturing both the strength and direction of connections. Despite over two decades of development in mixed Bayesian networks, combining continuous and discrete variables [8], the field still lacks efficient algorithms for identifying appropriate strong junction trees in their moral graphs. In our study, we address this gap by investigating the minimal strong triangulation of moral graphs and presenting a polynomial-time algorithm for deriving strong junction trees based on this approach. Moreover, we also consider the message passing computation of all posteriors on this derived strong junction tree. Our experimental findings affirm the superior efficiency of our algorithm compared to existing methods for message passing in mixed Bayesian networks. The proposed approach not only demonstrates remarkable improvements in computational speed but also provides a robust framework for efficient probabilistic inference in mixed Bayesian networks, making it a valuable tool for diverse applications in bioinformatics, computational biology, and neuroscience.

Author Contributions: Methodology, Y.L. and X.W.; Software, S.W. and C.Z.; Formal analysis, X.W. and S.W.; Writing—original draft, Y.L. and X.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Foundation of China (No. 12171076, 12226003, 11971097), the Natural Science Foundation of Jiangsu (No. BK20230439).

Data Availability Statement: All the codes can be found in https://github.com/KLASMOE/BPMBN.

Acknowledgments: Authors acknowledge valuable suggestions from the reviewers for this work.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Pearl, J. *Probabilistic Inference in Intelligent Sysetms*; Morgan Kaufmann: Burlington, MA, USA, 1988.
2. Laccourreye, P.; Bielza, C.; Larrañaga, P. Explainable Machine Learning for Longitudinal Multi-Omic Microbiome. *Mathematics* **2022**, *10*, 1994. [CrossRef]
3. Sazal, M.; Mathee, K.; Ruiz-Perez, D.; Cickovski, T.; Narasimhan, G. Inferring directional relationships in microbial communities using signed Bayesian networks. *BMC Genom.* **2020**, *21* (Suppl. S6), 663. [CrossRef] [PubMed]
4. Bernaola, N.; Michiels, M.; Larrañaga, P.; Bielza, C. Learning massive interpretable gene regulatory networks of the human brain by merging Bayesian networks. *PLoS Comput. Biol.* **2023**, *19*, e1011443. [CrossRef]
5. Joshi, A.; Rienks, M.; Theofilatos, K.; Mayr, M. Systems biology in cardiovascular disease: A multiomics approach. *Nat. Rev. Cardiol.* **2021**, *18*, 313–330. [CrossRef]
6. Liu, F.; Zhang, S.W.; Guo, W.F.; Wei, Z.G.; Chen, L. Inference of Gene Regulatory Network Based on Local Bayesian Networks. *PLoS Comput. Biol* **2016**, *12*, e1005024 [CrossRef]
7. Michiels, M.; Larrañaga, P.; Bielza, C. BayeSuites: An open web framework for massive Bayesian networks focused on neuroscience. *Neurocomputing* **2021**, *428*, 166–181. [CrossRef]

8. Salmerón, A.; Rumí, R.; Langseth, H.; Nielsen, T.D.; Madsen, A.L. A review of inference algorithms for hybrid Bayesian networks. *J. Artif. Intell. Res.* **2018**, *62*, 799–828 [CrossRef]
9. Lauritzen, S.L. Propagation of probabilities, means and variances in mixed graphical association models. *J. Am. Stat. Assoc. Stat.* **1992**, *87*, 1098–1108. [CrossRef]
10. Lauritzen, S.L.; Jensen, F. Stable local computation with mixed Gaussian distributions. *Stat. Computing* **2001**, *11*, 191–203. [CrossRef]
11. Cowell, R.G. Local propagation in conditional Gaussian Bayesian networks. *J. Mach. Learn. Res.* **2005**, *6*, 1517–1550.
12. Madsen, A.L. Belief update in CLG Bayesian networks with lazy propagation. *Int. J. Approx.* **2008**, *49*, 503–521. [CrossRef]
13. Lauritzen, S.L.; Spiegelhalter, D.J. Local computations with probabilities on graphical structures and their application to expert systems. *J. R. Stat. Soc. Ser.* **1988**, *50*, 157–224. [CrossRef]
14. Zhang, N.L. Computational properties of two exact algorithms for Bayesian networks. *J. Appl. Intell.* **1998**, *9*, 173–184. [CrossRef]
15. McGeachie, M.J.; Chang, H.H.; Weiss, S.T. CGBayesNets: Conditional Gaussian Bayesian network learning and inference with mixed discrete and continuous data. *PLoS Comput. Biol.* **2014**, *10*, e1003676. [CrossRef]
16. Yu, H.; Moharil, J.; Blair, R.H. BayesNetBP: An R Package for Probabilistic Reasoning in Bayesian Networks. *J. Stat. Softw.* **2020**, *94*, 1–31. [CrossRef]
17. Leimer, H.G. Triangulated graphs with marked vertices. *Ann. Discret. Math.* **1989**, *41*, 311–324.
18. Wang, X.F.; Guo, J.H. The tree structure of various graphs in statistics. *Stat. Its Interface* **2013**, *6*, 151–164.
19. Wang, X.F.; Guo, J.H.; He, X. Finding the minimal set for collapsible graphical models. *Proc. Am. Math. Soc.* **2011**, *139*, 361–373. [CrossRef]
20. Lauritzen, S.L. *Graphical Models*; Clarendon Press: Oxford, UK, 1996.
21. Berry, A.; Blair, J.R.S.; Heggernes, P.; Peyton, B.W. Maximum cardinality search for computing minimal triangulations of graphs. *Algorithmica* **2004**, *39*, 287–298. [CrossRef]
22. Jensen, F. HUGIN API Referene Manual. 2024. Available online: https://download.hugin.com/webdocs/manuals/api-manual.pdf (accessed on 19 June 2024).
23. Wen, W.X. Optimal decomposition of belief networks. In Proceedings of the Sixth Workshop on Uncertainty in Artificial Intelligence, Cambridge, MA, USA, 27–29 July 1990; pp. 245–256.
24. Blair, J.R.S.; Peyton, B.W. An introduction to chordal graphs and clique trees. In *Sparse Matrix Computations: Graph Theory Issues and Algorithms*; George, J.A., Liu, J.W.H., Eds.; IMA Volumes in Mathematics and its Applications; Springer: Berlin/Heidelberg, Germany, 1993; Volume 56, pp. 1–30.
25. Leimer, H.G. Optimal decomposition by clique separators. *Discret. Math.* **1993**, *113*, 99–123. [CrossRef]
26. Shachter, R.D. Evaluating Influence Diagrams. *Oper. Res.* **1986**, *34*, 871–882. [CrossRef]
27. Madsen, A.L. Improvements to message computation in lazy propagation. *Int. J. Approx.* **2010**, *51*, 499–514. [CrossRef]
28. Shachter, R.D.; Kenley, C.R. Gaussian influence diagrams. *Manag. Sci.* **1989**, *35*, 527–549. [CrossRef]
29. Madsen, A.L.; Jensen, F.V. Lazy propagation: A junction tree inference algorithm based on lazy evaluation. *Artif. Intell.* **1999**, *113*, 203–245. [CrossRef]
30. Butz, C.J.; Konkel, K.; Lingras, P. Join tree propagation utilizing both arc reversal and variable elimination. *Int. J. Approx. Reason.* **2011**, *52*, 948–959. [CrossRef]
31. Needham, C.J.; Bradford, J.R.; Bulpitt, A.J.; Westhead, D.R. A primer on learning in Bayesian networks for computational biology. *PLoS Comput. Biol.* **2007**, *3*, e129. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.