Knowledge-Based Systems 223 (2021) 107042

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

BNPA: An R package to learn path analysis input models from a data set semi-automatically using Bayesian networks

Elias Cesar Araujo de Carvalho [a,b,*], Joao Ricardo Nickenig Vissoci [c,d], Luciano de Andrade [a,c], Wagner de Lara Machado [e], Emerson Cabrera Paraiso [b], Julio Cesar Nievola [b]

[a] Department of Medicine, State University of Maringá, Av Mandacarú 1590-Anexo HUM-Maringá-PR, 87020-900, Brazil
[b] PPGIA, Pontifical Catholic University of PR, R Imaculada Conceição 1155, Bl 8, Curitiba-PR, 80215-901, Brazil
[c] Global Neurosurgery and NeuroScience Division, DUKE Global Health Institute, Duke University, 310 Trent Dr, Durham, NC 27710, USA
[d] Graduate Program in Health Sciences, State University of Maringá, Av Colombo, 5790-Maringá-PR, 87020-900, Brazil
[e] School of Health and Life Sciences, Pontifical Catholic University of Rio Grande do Sul, Av Ipiranga 6681, Predio 81-6°, and-Partenon-Porto Alegre-RS, 96619-000, Brazil

ARTICLE INFO

Article history:
Received 8 October 2020
Received in revised form 8 April 2021
Accepted 10 April 2021
Available online 18 April 2021

Keywords:
Bayesian networks
Path analysis
Causal inference
R-package

ABSTRACT

Epidemiologists constantly search for methodologies that help them better understand how diseases work. Populations urge these improvements to combat these diseases more effectively. The literature presents several authors defending the idea that epidemiologists should be able to develop causal models. In this area, the technique of structural equation models (SEM) has stood out in scientific research. Although SEM has been widely used in several research areas, it has been little explored by epidemiologists. Despite its evolution and efficiency, SEM has a gap in terms of discovering causalities. To fill this gap, this study developed an R package called BNPA, whose methodology joins the best of Bayesian network structural learning algorithms (BNSL) from data and path analysis (PA) a SEM subarea. The BNPA was built with pre-processing functions. Its main algorithm allows creating an input model to start the PA from a data set semi-automatically generating information to analyze the PA performance. An analysis of cardiovascular disease's main predictors was performed using the BNPA with data from the Canadian Community Health Survey (CCHS). Multiple linear regression (MR) was used as a gold standard methodology; the results of BNPA matched 85% of MR results. In conclusion, BNPA is efficient and can benefit researchers, mainly novices, by enabling them to build PA models from data. Furthermore, statisticians and PA experts will have more time to support these researchers instead of creating an initial model.

1. Introduction

Epidemiologists are continually looking for new knowledge to understand patterns better and fight diseases more efficiently. The description of these patterns and the illness's exposure are not enough to improve the population's health. For Petersen [1], there is a need to understand how these patterns evolve and intervene to combat diseases better. In this sense, epidemiologists must be able to ask causal questions and answer them; for this, it is necessary to develop causal models.

Although the methodology for creating causal models has existed for a long time, only in the last few decades have advances been made in creating these models, which have helped to disseminate the technique further. As an example, we can mention the unification of counterfactual languages [2,3], the possibility of creating causal graphs like Bayesian networks (BN), path analysis (PA) and structural equation models (SEM) [4–6]. The creation of computer programs that allow the implementation of a wide range of these models, from the simplest to the most sophisticated, such as AMOS, EQS, LISREL [7] and R packages like lavaan [8], sem [9] and openMX [10] also boosted the creation and use of causal models.

Among the various methodologies for creating causal models, SEM stood out for its increasing use in scientific research development in different areas of knowledge. SEM's popularity stems mainly from the fact that most traditional statistical methods apply only to a limited number of variables and can therefore fail to

The code (and data) in this article has been certified as Reproducible by Code Ocean: (https://codeocean.com/). More information on the Reproducibility Badge Initiative is available at https://www.elsevier.com/physical-sciences-and-engineering/computer-science/journals.

* Corresponding author at: Department of Medicine, State University of Maringá, Av Mandacarú 1590-Anexo HUM-Maringá-PR, 87020-900, Brazil.
E-mail address: ecacarva@uem.br (E.C.A.de Carvalho).

https://doi.org/10.1016/j.knosys.2021.107042
0950-7051/

deal with sophisticated emerging theories [11–13]. Sophisticated situations are understood in which the dynamic nature of real-life cases makes the same variable the outcome in one position and a predictor in another. SEM has advantages over other methods, such as correlation analysis and multivariate regression. The parameters of the causal relationship in SEM can be estimated simultaneously in a single multistage regression model (many dependent variables x many independent variables) [14].

SEM is widely used in economics, sociology and behavioral sciences (particularly psychology and econometrics). However, although there are already many studies using causal inference methods in epidemiology, SEM still has limited use [12,13]. A 2010 study [12] performed a search on PubMed using the keywords structural equation modeling, structural equation and trajectory analysis in six leading epidemiology journals (*Am J Epidemiol*, *Int J Epidemiol*, *Eur J Epidemiol*, *Ann Epidemiol*, *Lancet* and *Epidemiology*) from 2001 to 2008 and found only 24 articles that used SEM, 62.5% of which were published since 2006. This same search was performed again by this study (9 years later) from 2001 to 2019, and the result showed a total of 99 publications, 79.8% of which have been published since 2006, that is, few epidemiology studies use SEM. Some studies suggest that this limitation lies in the fact that most researchers cannot create their predictive models without the support of a statistical expert due to technical difficulties [5,6,11,13]. Corroborating the idea that there is a gap to deal with more complex models, Fox [15] states that the future of epidemiology lies in the methods that allow the creation of models of causality, as these models will enable us to answer a series of clinical questions that were previously intractable.

To develop this study, we used information about cardiovascular disease (CVD) provided by the Canadian Community Health Survey (CCHS) data set [16]. A literature review on different approaches was performed to identify studies about CVD using machine learning techniques such as BN, PA and SEM. The authors [17–24] developed models using Bayesian network techniques. PA models were created in the studies [25–27]. SEM was the technique used in the studies [28–30]. In these articles, the models were constructed with the help of experts, for instance, statisticians and specialists in BN, PA or SEM, which can be costly and time-consuming, and none of them used algorithms for learning predictive models from data. Besides, we believe that this process also generates problems in the final models caused by poor communication between knowledge engineers and human specialists as cited by Lacave [31], but in this case, during the manual development of a BN.

To overcome the limitations mentioned and further promote SEM use in epidemiology, it was developed an R-package called BNPA. BNPA is software that uses BN structure-learning (BNSL) algorithms to empower PA (a sub-area of SEM) with causal inference and learn the input PA model from a data set. When executing BNPA, it learns the structure of a BN (directed acyclic graph [DAG]) from a data set. This BN structure learning process is based on a combination of algorithms based on constraints and their respective tests of independence and algorithms based on scores and their network scores. This method allows creating different DAG structures if the researcher combines all the algorithms with all the tests and scores. These DAGs serve as the basis for the main BNPA algorithm to generate the input PA models to execute the PA inference. With these input models ready, the BNPA proceeds with the PA's execution and, at the end of the process, extracts tables of inference, goodness indices of adjustment of the model, residuals and the DAGs generated by the PA. The entire process is carried out semi-automatically, allowing researchers to create their PA models and only then discuss the model with the specialist. In this way, researchers will generate more PA models, and specialists will have more time to support researchers.

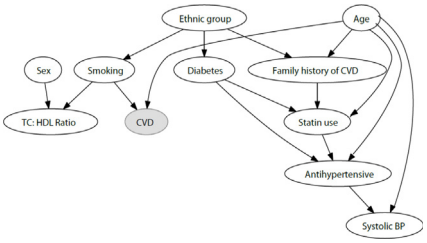

Fig. 1. DAG representing causal paths for cardiovascular disease—CVD: Cardiovascular disease. HDL: high-density lipoprotein cholesterol concentration. TC: total cholesterol concentration.

2. Baseline information

To carry out this study, it was necessary to understand graph theory, which has long been used as an auxiliary tool for causal analysis. Especially, the theory of DAGs has been used in conjunction with expert systems. Therefore, in the next sections important concepts will be described that will help explain the results generated by this study.

2.1. Directed acyclic graph

In graph theory, a graph is formed by vertices (nodes or variables) and edges connecting pairs of vertices. The vertices can be any type of object connected to the pairs by edges. In a DAG case, each edge that exits from one vertex to another vertex has an orientation. A DAG consists of a structure where each edge is directed from one vertex to another so that following these directions will never form a closed loop, that is, no vertex can reach itself through a nontrivial path [32]. An example of DAG for CVD is presented in Fig. 1 [33].

2.2. D-separation

BNs allows the creation of a causal model, whose objective is to represent the relationships between that model's variables. The way these variables connect determines the possible configurations between nodes and edges, generating the network structure, also known as fundamental connections (Fig. 2), and form the building blocks of BN's graph and probabilistic properties [34].

These connections reveal information about the nature of the variables and their dependence on other variables. Variables are dependent when the observation of one influences the other; in this case there must be a directed edge between them. The flow of this dependence must also be observed, since this is direct or indirect. When indirect, the dependency between two variables depends on a third variable. Conditional independence (a common term used in BNs) occurs when the value of one variable does not influence the other's result. The structures, which are essential for the characterization and learning of BNs, are classified into three distinct formations, described in the following paragraphs [34].

Serial connections contain structures where the flow of causal influence is of the type $S \rightarrow E \rightarrow R$ (first example of Fig. 2). In this case, both arcs have the same direction and follow one after the other, and any type of influence that S has will be replicated to E and consequently to R. However, if the value of E is informed, the causal flow between S and R is interrupted, and the variables will be considered as d-separated. In this case, S and R are d-separated by E or conditionally independent given E.

Diverging connections have the structure of the causal influence flow of type $R \leftarrow E \rightarrow O$ (according to the second example in Fig. 2). In this situation, the two arcs have divergent directions

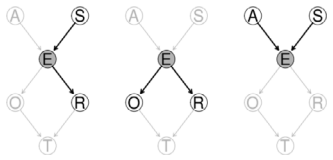

Fig. 2. Examples of fundamental connections.

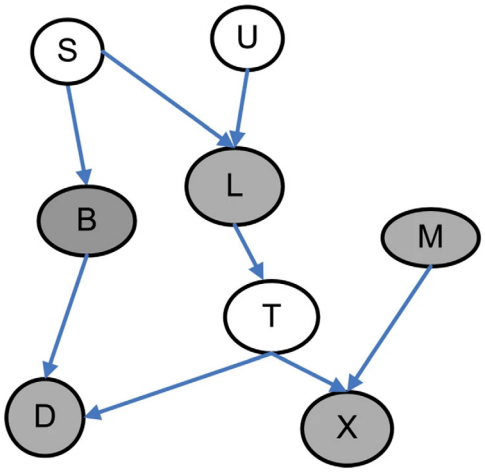

Fig. 3. The gray-filled nodes are the MB of node T.

from a central node, that is, variables O and R diverge from variable E. Considering this situation, variable E, as a parent node, will transmit the causal influence to all child nodes O and R, except when the state of E is known. As in the case of serial connections, O and R are d-separated by E or conditionally independent given E.

Converging connections are represented by structures of type $A \rightarrow E \leftarrow S$ (third example in Fig. 2). In this structure, the arcs converge to a central node, that is, the variables A and S converge to E, indicating that E will suffer a causal influence from A and S. In this case, A and S are not d-separated when the state of E is known; consequently, A and S are conditionally dependent on E.

The process of evaluating whether a variable is d-separated from another, commonly used in the construction of BNs, can be complex in models that represent the real world. Other solutions like the Markov blanket are also explored.

2.3. Markov blanket

Markov blanket (MB) is a method used in graph theory whose objective is to infer a random variable. When analyzing a graph, only a subset of variables can provide useful information about a given variable, and the other variables are irrelevant. The MB of a variable occurs when information is provided about the variables that represent your parents, your children, and your children's parents. In this case, this variable and/or node is independent of all other variables [35]. Fig. 3 represents the MB of variable T in a BN [36].

2.4. Bayesian networks

A BN is a graphical model with nodes representing random variables and edges representing the probabilistic dependencies between them [37]. A BN is defined by a network structure, represented as a DAG $G = (V, A)$, where each vertex or node $v_i \in V$ corresponds to a random variable X_i. Considering X a global probability distribution of a BN and its arcs $a_{ij} \in A$, X

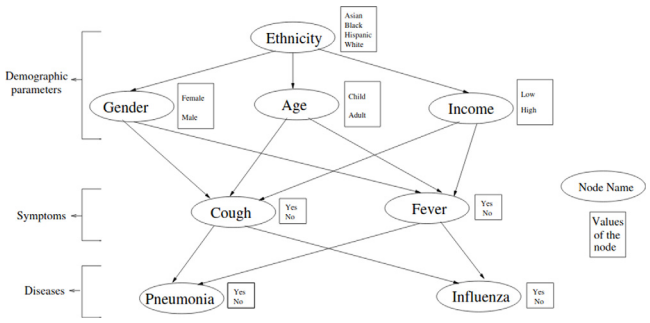

Fig. 4. Bayesian Network for demographic analysis of diseases.

can be factorized into smaller local probability distributions. The main objective of the network structure of a BN is to use the graphical d-separation to express the conditional independence relationships among the variables presented in the BN model. The factorization of the global distribution is specified by:

$$P(X_1, \ldots, X_v) = \prod_{i=1}^{v} P(X_i | \prod X_i) \quad \text{(for discrete variables)} \qquad (1)$$

$$f(X_1, \ldots, X_v) = \prod_{i=1}^{v} f(X_i | \prod X_i) \quad \text{(for continuous variables)} \qquad (2)$$

where $\prod X_i = \{\text{parents of } X_i\}$

As an example, the BN on Fig. 4 modeled the predictive reasoning of the demographics parameters (ethnicity, gender, age, and income) and symptoms (cough and fever) on the prevalence of two diseases: pneumonia and influenza [38].

To build a BN model two steps are needed: the first is to build the BN structure and the second is to estimate the conditional probability table (CPT) [39]. The BN structure can be built with help of experts or based on data using constrained-based, score-based or mixed BNSL algorithms. For this study we used the second option, which is detailed in the next section.

2.5. BN structure learning algorithms using constraint-based and score-based algorithms

To learn the structure of a BN based on a data set, there are two main approaches: constraint-based algorithms and score-based algorithms [40].

Constrained-based algorithms use Verma and Pearl's Inductive Causality algorithm [41] as a model, which provides a theoretical basis for learning causal model structures. The main idea is to analyze the probabilistic relationships resulting from MB by conditional independence (CI) tests creating a graph that satisfies the corresponding d-separation statements. It can be summarized in three steps:

- The network skeleton (non-directed graph) is learned. To avoid an exhaustive and computationally unviable search, the learning process restricts the search to the MB of each node as a way to optimize the process.
- Defines the directions of all edges that are part of a V structure, which are represented by three incident nodes in a convergent connection of type $X_j \rightarrow X_i \leftarrow X_k$.
- Define the directions of the other edges as needed to satisfy the acyclicity constraint.

For use in BNSL constrained-based algorithms, conditional independence tests are considered a key role in causality discovery [42]. In essence, if there are three variables X, Y and Z during the learning process of a BN structure, the objective

of the conditional independence test is to test the conditional independence hypothesis: H_0: X is dependent of Y given Z against the general alternative H_1: X is not dependent of Y given Z. The main objective of the CI test is to minimize the error type I (false rejection of the null hypothesis) and type II (false acceptance of the null hypothesis).

The bnlearn package [40] provides the following CI tests to work with BNSL algorithms [43–47]:

- For categorical variables are: mutual information—an information-theoretic distance measure, shrinkage estimator for the mutual information, Pearson's X^2—the classical Pearson's X^2 test for contingency tables;
- For categorical ordered variables are: Jonckheere–Terpstra—a trend test for ordinal variables;
- For continuous variables are: linear correlation—Pearson's linear correlation, Fisher's Z—a transformation of the linear correlation with asymptotic normal distribution, mutual information: an information-theoretic distance measure, shrinkage estimator for the mutual information—an improved asymptotic chi-square test based on the James–Stein estimator for the mutual information;
- For mixed categorical and continuous variables are: mutual information—an information-theoretic distance measure.

Score-based algorithms assign a score to each DAG candidate of BN structure learned, then, using some heuristic search algorithm, try to maximize that score [40]. Any search algorithm can be used, but the literature most often presents greedy search algorithms, such as hill-climbing or tabu search. The bnlearn package [40] implements the following network scores to work with BNSL algorithms [48–54]:

- For categorical variables are: the multinomial log-likelihood score, which is equivalent to the entropy measure used in Weka, the Akaike Information Criterion score, the Bayesian Information Criterion score, which is equivalent to the Minimum Description Length, the predictive log-likelihood computed on a separate test set, the logarithm of the Bayesian Dirichlet equivalent score, a score equivalent Dirichlet posterior density;
- For continuous variables are: multivariate Gaussian log-likelihood score, corresponding Akaike Information Criterion score, corresponding Bayesian Information Criterion score, predictive log-likelihood computed on a separate test set, a score equivalent Gaussian posterior density;
- For mixed categorical and continuous variables are: conditional linear Gaussian log-likelihood score, corresponding Akaike Information Criterion score, corresponding Bayesian Information Criterion score, the predictive log-likelihood computed on a separate test set.

This study used four constrained-based algorithms: grow-shrink (gs) [55], incremental association (iamb) [56], fast incremental association (fast.iamb) [57], interleaved incremental association (inter-iamb) [56] and two score-based algorithms, hill climbing (hc) [58] and tabu Search (tabu) [58]. A justification for the fact that there are only two score-based algorithms implemented in bnlearn [40] (according to the author) is that they are much more difficult to implement to have a good performance without using *C* language, which takes more time to write the R code.

All these algorithms were implemented in the bnlearn package [40] and were therefore used by BNPA.

2.6. Arch black listing and white listing

To create a BN using BNSL algorithms, the first step is to learn the structure of the BN by performing a data set that will determine which edges will be present in the graph that underlies the model. Ideally, this process would be purely data-driven, but real-world data generally does not allow this, and sometimes very little is known about the phenomenon studied. On the other hand, in some situations, there is prior knowledge of how the network structure should be. In this case, it is possible to incorporate this knowledge into the learning process of the BN structure through white lists and black lists. These lists work as follows: the edges present in the white list must be included in the BN structure and the edges present in the black list must be removed from the BN structure. This feature is implemented in the bnlearn package [40], and which was used in this study.

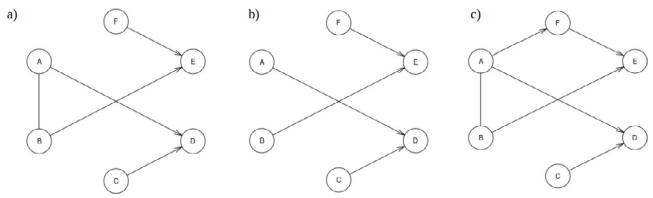

Fig. 5. An example of BN structure learned: (a) with an undirected edge between *A* and *B* and (b) without this undirected (in black list) and (c) with a new edge connecting *A* to *F* (in white).

Fig. 5a shows the structure learned from a set of data from a given BN. Note that an undirected edge from variable *A* to *B* was generated, which is not allowed in a DAG and consequently in a BN. To avoid this problem, a black list from *A* to *B* and from *B* to *A* is created, so a new BN structure will learn (Fig. 5b).

In an inverse to the black list, if you create a white list with the variable *A* pointing to variable *F*, this edge will appear in the BN structure (Fig. 5c); however, a black list with variable *F* pointing to *A* must be created, as this indicates the edge with the same nodes, but in the other direction must not be present in the graph.

2.7. Bootstrap resampling and model averaging

The techniques created to identify statistically significant characteristics in network structures learned from data had limitations. The cause of this is that the real probability distribution structure is unknown. In this context, a more efficient method was developed by Friedman [59] using bootstrap resampling [60] and model averaging [61].

Motivated by important statistical problems, Efron [60] originally proposed the method known as bootstrap, which has become an important statistical tool to deal with statistical biases. The central idea of bootstrap is to consider the sample as the population and obtain subsamples from it, by random resampling with replacement (in this case, non-parametric bootstrap) or by adjusting the model to which the data belong (parametric bootstrap).

Model averaging (MA) [61] is a technique that uses the result generated via bootstrap to select the best graph structure. This structure is built using statistical criteria that select the most relevant edges (for example, arcs that appear over a predefined limit of the good structures obtained). In this study, we used the "averaged.network" function from bnlearn, which receives as a parameter the result bootstrap resampling. A significance threshold, the calculation of which is described in [39], is computed automatically from the strength estimates. Then, all significant arcs from each BN structure were selected according to two criteria. First, the arc strength, calculated using the bootstrap resampling process, must be greater than the appropriate threshold value. Second, the direction parameter must be greater than 0.5, arcs with direction probability equal to 0.5 are score equivalent, and their direction cannot be identified. In contrast, values greater than 0.5 confirm that direction. Finally, the MA process dropped all arcs of BN that did not meet these two criteria.

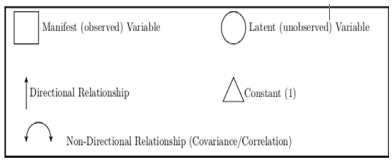

Fig. 6. Symbols used to construct PA diagrams.

2.8. Structural equation modeling and path analysis

SEM is an extension of several multivariate techniques created to overcome the limitation of techniques that can examine the relationship of just one dependent variable [62]. SEM is composed of a set of techniques and procedures that address an extension of other multivariate techniques, evaluating simultaneous relationships, that is, dependency and independence relationships between one or more variables. They are multivariate regression equations analyzed simultaneously, in which the outcome variable in one equation can appear as a predictor in another, and it is possible for the variables to influence each other reciprocally, directly or through other variables.

PA, the oldest member of the SEM family, is a method for measuring direct and indirect effects of cause variables on effect variables [6] developed by Sewall Wright, a geneticist, in 1919. A PA model is a pictorial representation (diagram) of the theory behind the relationship between variables. A special symbology is used to create the PA model [5] and can be seen in Fig. 6. In this figure, rectangles represent observed (or manifested) variables, circles or ellipses unobserved or latent variable, straight arrows with one end represent the direction of the relationship (cause to effect) and direct influence, curved double-headed arrows represent a covariance relationship or non-directional correlation, and the triangle represents a constant.

The naming of variables in PA, considering its more sophisticated counterpart SEM, is different from traditional statistics, precisely to avoid confusion. Instead of using the terms independent variable (IV) and dependent variable (DV), in PA, the terms exogenous variable (EXV) are used for those with straight arrows that emerge from them and none pointing to them, except when using error terms and endogenous variables (ENV) that must have at least one direct arrow pointing to them. These terms are justified by the fact that the causes or factors that influence EXVs are determined outside the model, while factors that influence ENVs are present in the model itself [5].

Fig. 7 shows an example of a PA model for a multiple regression, where X_1, X_2 and X_3 are EXVs and Y is ENV. In this case, variables X_1, X_2 and X_3 are considered to have a direct effect on Y and covariate with each other. The lower-case letters over each of the arrows represent the path coefficient. This coefficient can be positive indicating that an increase in the causal variable will result in an increase in the effect on the dependent variable if all other causal variables remain constant. If this coefficient is negative, an increase in the causal variable will cause a decrease in the effect on the dependent variable [4].

EVNs always have an error term, also known as a residual or disturbance term, represented by the circle associated with it. This term is similar to the error term inserted at the end of the regression equations. Similar to regression, they capture two occurrences: (a) inaccuracy in the measurement of ENVs, since all measurement tools suffer some degree of error, and (b) other factors that affect ENVs and have not been measured, either due to lack of time, lack of awareness of their importance or any other reason. This term represents the discrepancy between observed values and values predicted by the model.

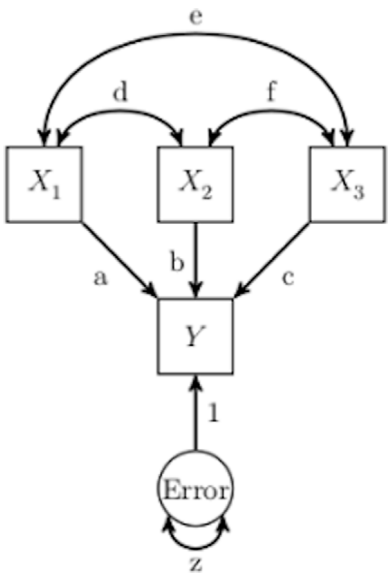

Fig. 7. PA model.

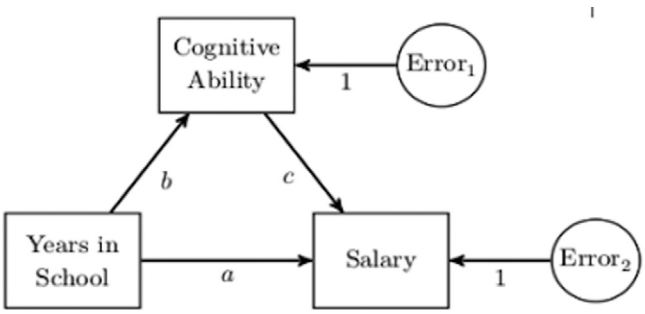

Fig. 8. Example of a PA model with an indirect effect.

Indirect effects occur when there is an influence of one variable over another through a third or fourth variable [5]. Fig. 8 presents an example of PA with indirect effect; in this case, it is postulated that there is a direct influence of the variable "*Years in School*" on the variable "*Salary*", represented by the path *a*. Additionally, it is also postulated that there is an indirect influence of "*Years at School*" on the variable "*Salary*", but passing through the variable "*Cognitive Capacity*", represented by the paths *b* and *c*.

Wright created the term known as "tracking rules" to estimate the covariance between two variables, that is, how to estimate the values for the coefficients of the model paths. This process takes place by scanning the paths within the model, that is, making an analysis of trajectories. During this process, the appropriate connection paths are added [5]. Thus, following the "tracking rules" the path from "*Years in School*" to "*Salary*" through "*Cognitive Ability*" is calculated by b^*c (indirect effect) and by a (direct effect), therefore the total impact of this relationship is given by $a+b^*c$.

3. Method

BNPA was developed primarily to fill the gap in the methods for creating PA models, which does not allow one to learn a PA model from a data set. It was also designed to help clinical researchers develop initial models without the help of an expert. Therefore, this software's development methodology involves the

```
# Set constrained-based algorithms to learn the structure BN
cb.algorithms = c("gs","iamb", "fast.iamb", "inter.iamb")

# Set score-based algorithms to learn the structure BN
sb.algorithms = c("hc","tabu")
```

Fig. 9. Definition of algorithms for executing BNPA.

study of pre-processing methods, which originated the creation of functions for this purpose in BNPA. It also involves Bayesian networks structure learning from data performed using the R package bnlearn [40] and estimating PA models using the R lavaan package [8]. The main BNPA algorithm creates an interface between these two packages.

3.1. The data set

This study used the data from CCHS 2012 [16], the result of a survey promoted in Canada by the Canadian Institute for Health Information, Statistics Canada and Health Canada. It was conducted every two years from 2001 to 2007, with approximately 130,000 respondents 12 years of age and older. For this study, we selected the 12 variables suggested by [63] and two variables proposed by [13].

3.2. Data preprocessing

The CCHS data set has some variables with the answers composed by "Not Applicable", "Do not Know", "Refusal", and "Not Stated". These answers are not significant for this study, so records with these answers were excluded. For all variables with values of "1 = Yes" and "2 = No", the second value was recorded as "0 = No".

The BNPA has support tools to help the researcher perform the data cleaning and pre-processing phase. Among them, there is the function "check.na" to show the amount and percentage of missing data for each variable; in this case, it is up to the researcher to treat the missing data. The function "check.outliers" and "preprocess.outliers" checks and eliminates, with the user's permission, continuous variables with outliers. We also analyze the correlation between the predictor variables (numeric and categorical) and generate a table with the possible pairs of correlated variables for the researcher to explore which is the best strategy to eliminate collinearity.

3.3. BNPA design

The BNPA was designed to require the researcher's minimum, so to create the PA model, the researcher must perform three steps from the data set. First, we must define which algorithms to use to learn the BN structure from the data set. In this case, BNPA uses the R bnlearn package [40]. Bnlearn uses constraint-based and score-based algorithms. Therefore, in this first step, the researcher must create two variables: cb.algorithms for the list of constrained-based algorithms and sb.algorithms for score-based algorithms (Fig. 9).

During the experiments, it was concluded that there are typically predictive and typically outcome variables in epidemiological studies. For a better understanding of this concept, suppose the variable "AGE" (AGE) causes CVD and other comorbidities such as 'high blood pressure', 'obesity', etc. If there is no variable in the study that influences age, such as 'time', for example, it can be considered as a typical predictor, that is, no other variable should point to it. The same applies to the variable "HHD" (Has heart disease) if the study does not evaluate the influence of "HHD" on other variables (case of this study). This variable will

```
# Create a black and white list (empty in the first momment)
black.list <- ""

# Set the outcome var(s)
type.var <- "o" # setting to outcome.predictor.var function to set a black list for outcome
var.name <- "HHD" # setting this variable as a typically outcome
black.list <- bnpa::outcome.predictor.var(data.to.work, var.name, type.var, black.list)

# Set the predictor var(s)
type.var <- "P" # setting to outcome.predictor.var function to set a black list for predictors
black.list <- bnpa::outcome.predictor.var(data.to.work, var.name, type.var, black.list)

type.var <- "P" # setting to outcome.predictor.var function to set a black list for predictors
var.name <- "SEX" # setting this variable as a typically predictor
black.list <- bnpa::outcome.predictor.var(data.to.work, var.name, type.var, black.list)
```

Fig. 10. Process of creating variables that are typically predictive and typically the outcome.

```
"HHD-AGE,HHD-SEX,HHD-HBP,HHD-DIA,HHD-SMK,HHD-ALC,HHD-BMI,HHD-WRK,HHD-STK,HHD-RUN,HHD-WLK,HHD-BYC,HHD-INC,
SEX-AGE,HBP-AGE,DIA-AGE,SMK-AGE,ALC-AGE,BMI-AGE,WRK-AGE,STK-AGE,RUN-AGE,WLK-AGE,BYC-AGE,INC-AGE,HHD-AGE"
```

Fig. 11. Black list generated by the function "outcome.predictor.var".

be considered an outcome typically, that is, it should not point to other variables, but the opposite is allowed.

In the case of BN structures, which are represented graphically by DAGs, any variable can point to any variable. This conflicts with what was exposed in the previous paragraph. In this case, the variable "AGE" can be pointed to by another variable, and the variable "HHD" can also point to any other. This situation generates relations called spurious (incorrect) and consequently, more processing of the process. For this reason, in the second stage, the researcher must inform which are the typical predictor variables and the typical outcome variables. This feature was implemented through a black list. This function is available in the R bnlearn package [40], but because the syntax for creating a black list in bnlearn is not user-friendly, especially for inexperienced users, the "outcome.predictor.var" function was created in BNPA, whose syntax is simpler (Fig. 10).

The result of the BNPA "outcome.predictor.var" function can be seen in Fig. 11. The variable "*HHD*", as it is typically an outcome, should not point to other variables such as "*AGE*", "*SEX*", etc. Because of this, it is positioned on the left side, representing the cause. The "*HHD-AGE*" parameter means that during the process of learning the structure of BN, the "*HHD*" variable cannot point to the variable "*AGE*". As for the variable "*AGE*", since it is typically a predictor, it cannot be indicated by "*SEX*", "*HBP*", etc., so it is on the right side, representing an effect. This combination represents a very flexible way of correcting any arbitrary set of assumptions about the data set. Regardless of how significant the result of the BN structure learning algorithms is, improper relationships may appear. It allows the use of prior knowledge, such as information from experts in the relevant area, to be integrated into the BN structure's learning process.

Although it was not used in this study, bnlearn also offers the resource to create a white list whose relations represent the BN structure's mandatory relations. In BNPA, both the black list and the white list can later be supplemented manually with other relationships as the process evolves and learns new BN structures.

As a third step (Fig. 12), the researcher defines the study's outcome variable by adjusting the variable "outcome.var", informs if he will execute the PA by adjusting the variable "build.pa = 1". The BNPA "build.pa" parameter, when equal to zero, allows only the process of learning the BN structure to be executed, in case the researcher wishes to analyze the BN DAG before executing the complete process. Finally, the user sets the "nreplicates" variable with the number of replications to be used during the execution of the resampling bootstrap process [47] for the validation of the BN learned from the data.

```
# Start generation of BN and PA models
outcome.var = "HHD"
build.pa = 1
nreplicates = 5000

# Learn the BN Structure and Build the PA Model and it fit indexes
gera.bn.structure(data.to.work.train, white.list, black.list, nreplicates,
                  cb.algorithms, sb.algorithms, outcome.var, build.pa)
```

Fig. 12. Preparation and execution of BNPA.

All of these parameters are optional, and the researcher can run the BNPA just by passing the data set as a parameter through the command: gera.bn.structure (data.to.work). Thus, BNPA will use as default, empty black and white lists, replication of the bootstrap = 1000, the combination of all BN structure learning algorithms. According to the authors of bnlearn [40], the default parameters of the bnlearn package were set by rules of thumb or suggestions in books and research papers.

During the execution of the process for learning the BN structure and consequently building the PA model, BNPA uses the function "check.types". This function identifies the type of variable that the data set has. According to the result of this function, it will be returned: 1 = integer, 2 = numeric, 3 = factor, 4 = integer and numeric, 5 = integer and factor, 6 = numeric and factor, 7 = integer, numeric and factor. If they are categorical variables, BNPA also uses the function "check.ordered.one.var" to classify a particular variable as ordinal categorical or nominal categorical, since these types of variables are treated differently. Based on the type of variable identified, BNPA defines which conditional independence tests (cb.tests) to use for constraint-based algorithms and which network scores (sb.tests) to use for score-based algorithms. Variables of type 4, 5, 6, 7, where there is a mixture of types (numeric + categorical), are not treated by BNPA yet.

Fig. 13 shows the BNPA workflow. This process starts with the processing of the black and white list (if any). This process automatically creates these lists in the syntax of the bnlearn package. The BNPA then receives the data, the BNSL algorithms, the white/black lists, and final parameters (outcome.var, build.pa, and nreplicates). In the next step, check if the BNSL algorithms are blank and, if so, use the BNPA default; if not, use what the researcher has defined and check if the algorithms are allowed. If the BNSL algorithms are not allowed, the process is interrupted with a message to review the algorithms; otherwise, BNPA then identifies the type of variable in the data set and adjusts the CI tests and network scores compatible with the identified type.

As a next step, BNPA executes the BNSL algorithms one by one, first the ones constrained-based and for its respective CI tests and then those score-based and their respective network scores. BN structures are learned from the data set, validated by the bootstrap resampling process [47] and model averaging [48], and their respective DAGs are exported. The BNPA reads these DAGs, and through its main algorithm, the PA input model is created. With this input model ready, the PA process is executed, generates the necessary inferences, goodness-of-fit indices, residuals and the PA graph.

As a final step, these results are exported by BNPA for the researcher to evaluate the PA models. If these results are satisfactory, the researcher ends the process; otherwise, they perform the necessary changes (data set, pre-processing, and parameters) and restart it.

For clarification: depending on the list of algorithms that the researcher passes to BNPA, the learning process of the BN structure will use algorithms based on constraint or algorithms based on the score; however, both types of algorithms can be used to create these structures (standard option of BNPA). Algorithm 1 is responsible for verifying if the parameter list for each type of algorithm is empty or contains some information, confirms if they are acceptable algorithms, and proceeds with the due processing.

3.4. Bayesian network structure

One way to learn a BN structure is by hand, using experts' knowledge, and another is to use the data to determine which arcs are present in the BN underlying the model [34]. To learn the structure of a Bayesian network from a data set, algorithms and their specific tests are used. The bnlearn implements constrained-based algorithms that use conditional independence tests and score-based algorithms that use network scores. BNPA allows the use of four constrained-based algorithms (through bnlearn): grow-shrink (gs) [42], incremental association (iamb) [43], fast incremental association (fast.iamb) [44], interleaved incremental association (inter.iamb) [43] and two score-based algorithms, hill-climbing (hc) [45] and tabu search (tabu) [45]. According to [40], as required by bnlearn, when using constrained-based algorithms, since our data are composed of categorical and ordinal variables, *BNPA* automatically transformed all binary variables into ordinal ones and used Jonckheere–Terpstra (JT) algorithm as CI test. For score-based algorithms, the Akaike information criterion (AIC), Bayesian information criterion (BIC), and the logarithm of the Bayesian Dirichlet equivalent (BDE) were used as network scores, and all variables were treated as categorical.

3.5. Validation of the learned Bayesian networks structure

To evaluate the degree of confidence of a DAG for a respective BN is a crucial problem in the inference in the network structure. Friedman, Goldszmidt and Wyner [46] introduced an effective method of quantifying this confidence. The process generates various network structures by applying non-parametric bootstrap resampling to the data and estimates each arc's strength and the probabilities of each arc's directions. We used this method to ensure the quality of the learned BNs on this study creating 100, 200, 300, 400, 500, 600, 700, 800, 900 and 1000 network structures. Since the process of bootstrap is of parallel nature, a parallel package embedded in R language to parallelize this process was used.

After this step, the BNPA executed the process of model averaging [48] to validate the learned BN structure.

3.6. Building the PA model

The first step to conduct a PA is to construct a PA input model representing the hypothesized relationships. The next step is to execute the statistical analyses and then build the output PA graph describing the relationships between the variables and generate the inference measures. Typically, the whole process to create a PA model is done by statisticians and/or SEM experts, a costly and time-consuming resource not always immediately available to all researchers. Of importance, according to [14], PA will not discover the causal relationship between variables. This gap will be filled by BNPA using its algorithm and BNSL algorithms available in bnlearn to learn the PA input model from the data set.

With the PA input model ready, BNPA will use the lavaan [8] to start all PA steps. In lavaan, if one has exogenous (independent) categorical dichotomous variables, these variables need to be recoded as a dummy (0/1); if they are ordinal, they need to be coded to reflect their order, as in the preprocessing step, and treated as any other (numeric) covariate. For endogenous (dependent)

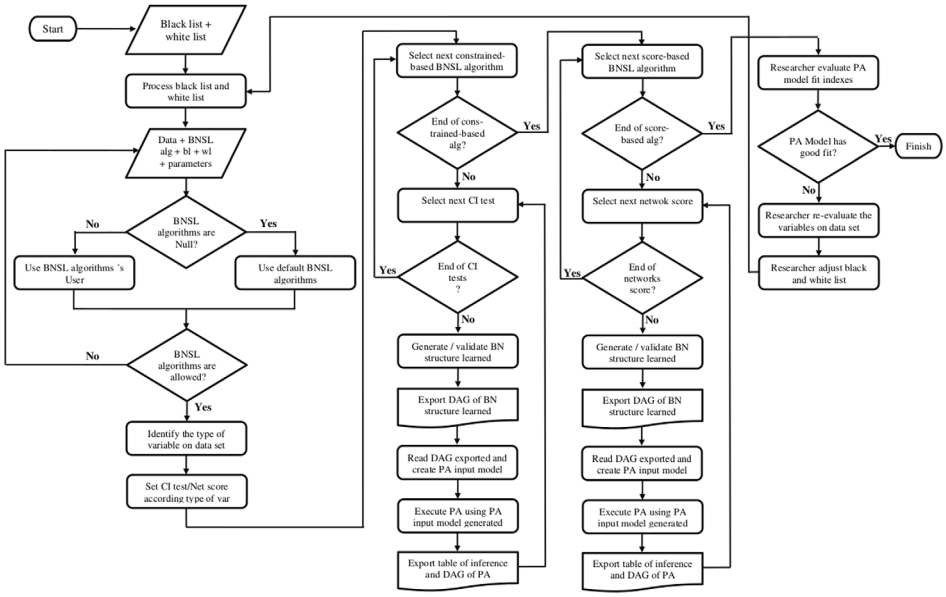

Fig. 13. Workflow of BNPA.

variables, if they are dichotomous or ordinal categorical variables, the 'ordered' argument of lavaan package should be used with the fitting functions. For example, with four binary/ordinal variables (*v1*, *v2*, *v3*, *v4*), they can fit the model using the syntax: pa. Model.fit = sem(input.pa.model, data = data.to.work, ordered=c("*v1*", "*v2*", "*v3*", "*v4*")); then lavaan will automatically use the most suitable algorithm to estimate the parameters. BNPA will do this.

To generate the PA model, algorithm 1 was developed as part of BNPA and automatically performed all the necessary steps according to the variable type, as mentioned in the previous paragraph. This algorithm receives as input a BN structure learned on the previous step, then also receives the data set name and the names to save the final PA diagram and PA statistics and residuals. In the first step, the PA input model is built using the BN structure learned from the data set. In the second step, after the PA input model is created, the BNPA verifies if endogenous variables are binary or ordered categorically. If they exist, mount a list to declare it. All variables not declared on this list are transformed into numeric by BNPA, and the PA model is built. In the fourth step, the algorithm verifies if there are endogenous variables to be declared as ordered and does it, then fits the PA model using "ordered" argument in lavaan. Otherwise, it uses commands without this argument. The fifth step computes a variety of fitness measures to assess the global fitness of the PA model, calculates the residuals and generates the PA graph.

3.7. The PA model evaluation

The evaluation of the PA model performance was made through the set of fitness statistics recommended by [5,6]: (a) root mean square error of approximation (RMSEA), which assesses if a specified model has a reasonable approximation over the data, (b) standardized root mean square residual (SRMR), a measure of the mean absolute correlation residual and represents the overall difference between the observed and predicted correlations. For a and b a score lower than 0.08 indicate better fit; (c) comparative fit index (CFI), which compares the fit of a target model to the fit of an independent, or null, model, where a score higher than 0.9 indicates a better fit; and (d) goodness of fit index (GFI), similar to R^2 in regression, which compares the fit of the proposed model to a saturated model that allows all the variables to covary. For c and d, scores higher than 0.9 indicate a better fit.

The residuals produced by the adjusted model were also evaluated. According to the literature [5,6], values above 0.10 deserve attention and review of the model, as they indicate that the model created did not have a good fit.

Finally, a multiple logistic regression (MR) was used as the gold-standard technique to identify the relationships among the variables and to be compared with PA results.

4. Results

The presence of missing values and collinearity was evaluated, but no events were identified. All variables were considered categorical, and those with more than two levels were deemed to be categorical ordinal.

4.1. The final data set

After the preprocessing phase, 24,632 patients were analyzed (Table 1), of which 714 (3.0%) had heart disease, and 23,901 (97.0%) did not have heart disease. The age more prevalent was adult (85.1%). The gender of the patients analyzed was similar in proportion (49.7% vs. 50.3%).

4.2. Bayesian networks structure learned

Ten BN structures learned from the data set were automatically generated by BNPA. Consequently, 10 PA models were built. Our CVD experts first analyzed the BN structures, and if at least one of the experts considered a structure incorrect, the BNSL process would be repeated with some adjustments using white-list and black-list through bnlearn. The main criteria for choosing the best BN structure were: (a) the BN structure presenting relations corroborated by the literature, (b) the BN structure showing the highest number of correct predictors for the CVD, and (c) minor incorrect relationship between the variables.

The BN structure that presented the best results for these criteria was created by the combination "hc/aic" (BNSL algorithm/network score). Fig. 14 partially shows the result of the bootstrap resampling process for this combination. The first column indicates the number of arcs created (182), the second

Algorithm 1: The PA model builder algorithm

```
Input: bn, ds, dn, gn
Output: PA model parameters, PA model graph
Initialize: pa.input.model, ordered.to.declare, cat.to.transform.into.numeric, fitted.model, fitted.measures
# Mount a PA input model
01. for each variable in ds # scan all variables of the data set
02.     for each variable in BN struct # scan all nodes of BN
04.         if the variable in DS is the same in BN Struct then
05.           if there exists a node parent of the variable in DS then
06.             extract nodes parent
07.             for each node parent
08.                 add 1 to the counter of path identifier
09.                 if is the first variable then
10.                   pa.input.model ← pa.input.model +  current variable in ds ~
11.                                    counter of path identifier (c1) *
12.                                    current node parent
13.                 else
14.                   pa.input.model ← pa.input.model+"+"current parent of a variable HBP in BN struct
15.                 end if
16.             end for each
17.           end if
18.         end if
19.     end for each
20. end for each
# Discover ordinal/dichotomous categorical variables.
21. for each variable in ds # scan all variables of data set
22.     if current variable in ds has parents in BN structure learned &
          is (dichotomous or ordinal categorical) then
23.           if is the first variable then
24.               ordered.to.declare ← ordered.to.declare (empty) + "ordered=c(" + currently variable in ds
25.             else
26.             ordered.to.declare ← ordered.to.declare + currently variable in ds
27.           end if
28.     else
29.      cat.to.transform.into.numeric←cat.to.transform.into.numeric+ current variable in ds
30.     end if
31. end for each
# Create a PA model
32. if there are ordinal/dichotomous categorical variables then
33.   transform other variables into numeric
34.   fitted.model ←fit the model (pa.input.model, ds, ordered option (ordered.to.declare))
35. else
36.   fitted.model ←fit the model (pa.input.model, ds)
37. end if
# Extract the fit measures/residuals and export parameters and the graph
38. fitted.measures ← extract the fit measures (fitted. Model)
39. residuals.matrix ← extract the residuals (fitted.model)
40. export PA model parameters (fitted.measures, dn)
41. export PA model graph (fitted. Model, fitted.measures, gn)
```

bn the BN structure learned from data set, *ds* the data set used to learn the BN structure, *dn* the document name to save the PA model parameters, *gn* the graph name to save the PA model graph

column, the source node, the third the destination node, the fourth the strength of this relationship, and the fifth the direction.

The result of the bootstrap resampling (Fig. 14) was passed to the model averaging process, which initially calculated the significance threshold of 0.487 (Fig. 15). All arcs with a strength less than this value and with direction less equal than 0.50 were excluded. In the "model" section, when there is only one variable in brackets, it means that it has no predictors, which is the case of the "*AGE*" and "*SEX*" variables. In the case of a variable separated by "|", it means that the variables after this sign are its predictors; according to Fig. 15 the variable "*BMI*" has as predictors the variables "*AGE*" and "*SEX*".

Fig. 16 represents the BN structure resulting from the model averaging process for hc/aic combination.

The red lines highlight the variables that have a direct influence on the *HHD* variable; the other variables have indirect effects.

4.3. PA models learned

The BNPA, using algorithm 1, receives the data from the BN structure (Fig. 16) and then creates the PA input model (Fig. 17). The variable on the left side represents the dependent variable, and the variables after the "∼" are the independent variables, the lower-case letters (*c1*, *c2*, *c3*, etc.) represent identifiers that will help identify and calculate the indirect effects on the generated indices. The PA model graph created by BNPA that corresponds to the chosen BN structure (Fig. 16) is presented in Fig. 18.

For a better understanding, the execution of algorithm 1 will be illustrated with examples. Suppose the *HBP* variable of the learned BN structure is shown in Fig. 16. Note that it has the *SEX* and *AGE* variables as parents. Algorithm 1 will receive the data set and the variables AGE, SEX, HBP, and BN structure with *SEX* → *HBP* ← *AGE* inside and the names to generate the fit indexes and figures.

Table 1
Patients with factors associated with heart disease and non-heart disease.

Variables	No Heart Disease (n=23918)	Heart Disease (n=714)	Total (n=24632)
AGE, N (%)			
Adult	20,342 (85.1%)	376 (51.4%)	20,718 (84.1%)
Elderly	3,559 (14.9%)	355 (48.6%)	3,914 (15.9%)
HBP, N (%)			
No HBP	20,686 (86.5%)	373 (51.0%)	21,059 (85.5%)
Has HBP	3,215 (13.5%)	358 (49.0%)	3,573 (14.5%)
STK, N (%)			
No Stroke	23,817 (99.6%)	711 (97.3%)	24,528 (99.6%)
Stroke	84 (0.4%)	20 (2.7%)	104 (0.4%)
SEX, N (%)			
Female	12,013 (50.3%)	244 (33.4%)	12,257 (49.8%)
Male	11,888 (49.7%)	487 (66.6%)	12,375 (50.2%)
DIA, N (%)			
Not Diabetic	22,897 (95.8%)	595 (81.4%)	23,492 (95.4%)
Diabetic	1,004 (4.2%)	136 (18.6%)	1,140 (4.6%)
SMK, N (%)			
Nonsmoker	8,466 (35.4%)	145 (19.8%)	8,611 (35.0%)
Smoker	15,435 (64.6%)	586 (80.2%)	16,021 (65.0%)
BMI, N (%)			
Normal	10,703 (44.8%)	206 (28.2%)	10,909 (44.3%)
Obese	8,182 (34.2%)	277 (37.9%)	8,459 (34.3%)
Overweight	5,016 (21.0%)	248 (33.9%)	5,264 (21.4%)
INC, N (%)			
< 20k	1,244 (5.2%)	54 (7.4%)	1,298 (5.3%)
20k–39k	3,382 (14.2%)	138 (18.9%)	3,520 (14.3%)
40k–59k	4,306 (18.0%)	154(21.1%)	4,460 (18.1%)
60k–79k	4,085 (17.1%)	132 (18.1%)	4,217 (17.1%)
80k+	10,884 (45.5%)	253 (34.6%)	11,137 (45.2%)
BYC, N (%)			
No bike	22,970 (96.1%)	717 (98.1%)	23,687 (96.2%)
Bike	931 (3.9%)	14 (1.9%)	945 (3.8%)
WLK, N (%)			
No walk	19,519 (81.7%)	642 (87.8%)	20,161 (81.8%)
Walk	4,382 (18.3%)	89 (12.2%)	4,471 (18.2%)
WRK, N (%)			
No work	916 (3.8%)	51 (7.0%)	967 (3.9%)
Work	22,985 (96.2%)	680 (93.0%)	23,665 (96.1%)
ALC, N (%)			
Not Alcoholic	3,242 (13.6%)	183 (25.0%)	3,425 (13.9%)
Alcoholic	20,659 (86.4%)	548 (75.0%)	21,207 (86.1%)
RUN, N (%)			
No run	17,890 (74.9%)	668 (91.4%)	18,558 (75.3%)
Run	6,011 (25.1%)	63 (8.6%)	6,074 (24.7%)

N number; **AGE** 1 = Adult, 2 = Elderly); *SEX* 0 = Female, 1 = Male; *HBP* 0 = No high blood pressure, 1 = high blood pressure; *DIA* 0 = No diabetes, 1 = diabetes; *SMK* 0 = No smoker, 1 = smoker; *ALC* 0 = No alcoholic, 1 = Yes alcoholic); *BMI* 1 = Normal weight, 2 = Overweight, 3 = Obese), WRK 0 = No work, 1 = Yes worked at job or business); STK 0 = No stroke, 1 = Yes suffers from the effects of a stroke; *RUN* 0 = No jogging, 1 = Yes jogging in the last 3 months; *WLK* 0 = No walking, 1 = Yes walking to go work or school); *BYC* 0 = No biking, 1 = Yes biking to and from work or school; *INC* for categories of total household income 1 = None or < 20K, 2 = 20–39K, 3 = 40–59K, 4 = 60–79K, 5 = 80k+) and *HHD* 0 = No heart disease , 1 = Yes has heart disease.

The first part (lines 1 to 20), the creation of the PA input model, is executed as follow:
for each variable in ds # scanning variable *HBP*
for each variable in BN structure
the variable in DS is the same variable in BN structure?
Suppose NO—The variable in DS is *HBP* and the variable in BN structure is *AGE*,
then read next variable in the BN structure
Suppose Yes—the variable in DS is *HBP* and the variable in BN structure is *HBP*
there exist a node parent in BN structure for variable *HBP*?
Suppose NO—Goto ". for each variable in ds" and read the next variable in DS

Suppose YES—Extract the node parents in BN structure of variable *HBP*
for each node parent extracted
add 1 to the counter of path identifier (c)
if is the first variable
pa.input.model ← pa.input.model (empty)+current variable in ds (*HBP*) ∼
counter of path identifier ($c1$) *
current parent of a variable HBP in BN (*AGE*)
pa.input.model now is equal to "*HBP* ∼ *c1* * *AGE*"
else
pa.input.model ← pa.input.model+"+"current parent of a variable *HBP* in BN (*SEX*)

	from	to	strength	direction
1	AGE	SEX	0.000000000	0.00000000
2	AGE	HBP	1.000000000	1.00000000
3	AGE	DIA	1.000000000	1.00000000
4	AGE	SMK	0.536926148	1.00000000
5	AGE	ALC	0.778443114	1.00000000
6	AGE	BMI	0.004990020	1.00000000
7	AGE	WRK	1.000000000	1.00000000
8	AGE	STK	0.375249501	1.00000000
9	AGE	RUN	1.000000000	1.00000000
10	AGE	WLK	0.744510978	1.00000000
11	AGE	BYC	0.160678643	1.00000000
...				
175	HHD	ALC	0.710578842	0.00000000
176	HHD	BMI	0.321357285	0.00000000
177	HHD	WRK	0.003992016	0.00000000
178	HHD	STK	0.344311377	0.00000000
179	HHD	RUN	0.689620758	0.00000000
180	HHD	WLK	0.005988024	0.00000000
181	HHD	BYC	0.007984032	0.00000000
182	HHD	INC	0.039920160	0.00000000

Fig. 14. Bootstrap resampling result.

```
model:
 [AGE][SEX][BMI|AGE:SEX][HBP|AGE:BMI][RUN|AGE:SEX:HBP:BMI]
 [WLK|AGE:SEX:HBP:BMI:RUN][BYC|AGE:SEX:BMI:RUN:WLK]
 [INC|AGE:SEX:BMI:RUN:WLK][SMK|AGE:SEX:HBP:BMI:RUN:WLK:INC]
 [STK|AGE:HBP:INC][WRK|AGE:SEX:SMK:WLK:INC]
 [ALC|AGE:SEX:SMK:WRK:RUN:INC][DIA|AGE:SEX:HBP:ALC:BMI:INC]
 [HHD|AGE:SEX:HBP:SMK:ALC]
nodes:                                14
arcs:                                 55
  undirected arcs:                    0
  directed arcs:                      55
average markov blanket size:          8.71
average neighbourhood size:           7.86
average branching factor:             3.93

generation algorithm:                 Model Averaging
significance threshold:               0.4870259
```

Fig. 15. Model averaging results.

pa.input.model now is equal to "*HBP* ∼ *c1* * *AGE* + *c2* * *SEX*"

end
end for each
end for each
end for each

The second part lines (21 to 31) is the discovering of categorical dichotomous variables and executes as follow:

Suppose the variables *ALC* and *SMK* presented in Fig. 16 both have parents and therefore are endogenous variables; in this case, algorithm 1 will perform a scan in the database variables, if they have parents in the BN structure and if they are of the dichotomous or ordinal categorical type, they should be placed on a list. This list will serve as a parameter for the command that will execute the PA. Variables without parents, even categorical ones, will be transformed into numeric ones. Both rules are stated in the lavaan package [8]. Below is an example of how the algorithm will behave when reading the *ALC* and *SMK* variables.

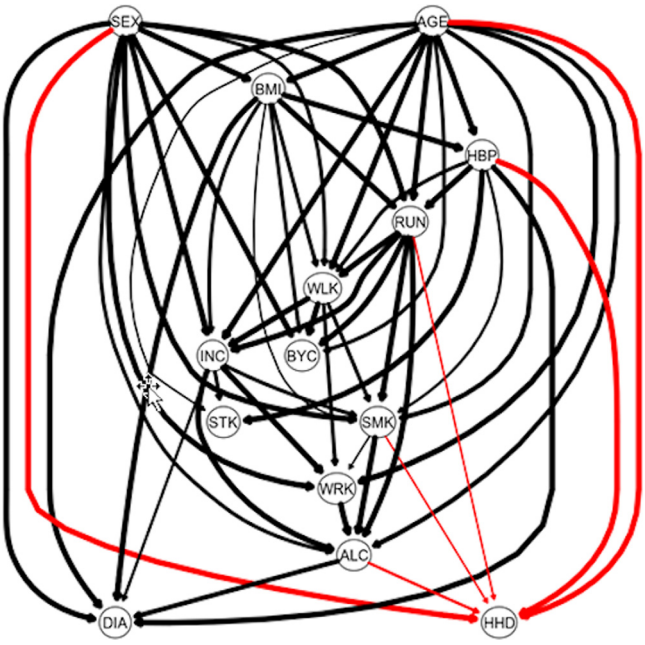

Fig. 16. The BN structure learned by *hc*/aic algorithm/network score.

```
HBP ~ c1 * AGE + c2 * BMI
DIA  ~ c3 * AGE + c4 * SEX + c5 * HBP + c6 * ALC  + c7 * BMI  + c8 * INC
SMK ~ c9 * AGE + c10 * SEX + c11 * HBP  + c12 * BMI  + c13  * RUN + c14 * WLK + c15 * INC
ALC  ~ c16 * AGE + c17 * SEX + c18 * SMK + c19 * WRK + c20 * RUN + c21 * INC
BMI  ~ c22 * AGE + c23 * SEX
WRK ~ c24 * AGE + c25 * SEX + c26 * SMK + c27 + WLK + c28 * INC
STK  ~ c29 * AGE + c30 * HBP + c31 * INC
RUN  ~ c32 * AGE + c33 * SEX + c34 * HBP + c35 * BMI
WLK ~ c36 * AGE + c37 * SEX + c38 * HBP + c39 * BMI + c40 * RUN
BYC  ~ c41 * AGE + c42 * SEX + c43 * BMI + c44 * RUN + c45 + WLK
INC  ~ c46 * AGE + c47 * SEX + c48 * BMI + c49 * RUN + c50 * WLK
HHD ~ c51 * AGE + c52 * SEX + c53 * HBP + c54 * SMK + c55 * ALC + c56 * RUN
```

Fig. 17. PA input model created by BNPA using the BN structure learned.

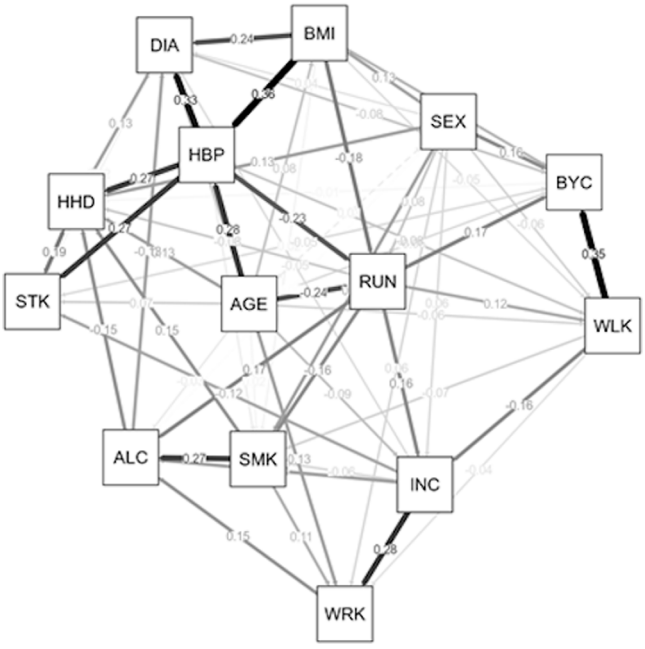

Fig. 18. The PA model built from the BN structure learned with hc/aic algorithm/network score.

for each variable in ds # scan *ALC*
current variable in ds has parents in BN structure learned (is endogenous) &
is (dichotomous or ordinal categorical)?
Suppose No–the variable in DS will be added to a list to be transformed into numeric
Suppose Yes–the variable in DS (*ALC*) will be added to a list to be declared as ordered
if is the first variable
ordered.to.declare ← ordered.to.declare (empty) + "ordered=c(" + currently variable in ds
ordered.to.declare now is equal to "ordered=c("*ALC*",
else
ordered.to.declare ← ordered.to.declare + currently variable in ds
end if
end for each
for each variable in ds # scan *SMK*
current variable in ds has parents in BN structure learned (is endogenous) &
is (dichotomous or ordinal categorical)?
Suppose Yes–the variable in DS (*SMK*) will be added to a list to be declared as ordered
if is the first variable
ordered.to.declare ← ordered.to.declare (empty) + "ordered=c(" + currently variable in ds
else
ordered.to.declare ← ordered.to.declare + currently variable in ds
ordered.to.declare now is equal to "ordered=c("*ALC*", "*SMK*")"
end if
end for each

The third part (lines 32 to 37) performs the procedure to create the PA model. Line 32, "if there are ordinal/dichotomous categorical variables then" will check the content of variable ordered.to.declare and if it is not empty, need to transform the other variables in cat.to.transform.into.numeric variable into numeric (line 33). Still considering variable ordered.to.declare is not empty, our examples on previous lines the algorithm 1 will build the command to execute PA and the result will be like **"pa.model ← sem(pa.input.model, data=ds,ordered=c("*ALC*", "*SMK*")**). Otherwise, without categorical endogenous variables, the line 37 will be executed and the result will be like "**pa.model ← sem(pa.input.model, data=ds)**".

4.4. The PA model evaluation

Table 2 shows residual values for each combination BNSL algorithm and test or score by the number of replications performed by the bootstrap (which ranged from 100 to 1000). As can be seen, the best results are from the combination "hc/aic" (3) and "tabu/aic" (10) with no residuals above 0.10, which requires a PA model revision.

Table 3 shows the number of Goodness-of-Fit Indexes whose scores indicated the best fit according to the literature's limits. The chi-square value is divided by the degree of freedom, which is also a Goodness-of-Fit Index whose value must be below 3. As in Table 3, the "hc/aic" combination (A/T/S=3), for the number of replicates 300, indicated the best result presenting the maximum acceptable number of fit indexes (5) and the acceptable X^2/df (second line) (2.3).

Table 4 presents in an analytic way the Goodness-of-Fit of the PA model learned for each combination BNSL algorithm, CI test, network score and 300 replications during the bootstrap process.

In addition to the Goodness-of-Fit Indexes and residuals, we also considered the MR (Table 5), whose results suggest that *age*, *high blood pressure*, *stroke*, *sex*, *diabetes* and *smoking* have a positive and significant influence on *heart disease* and *alcohol consumption*, *running* have a significant adverse effect. The results

Table 2
Quantity of residual values from PA models above 0.10 by the number of replicates on bootstrap.

A/T/S	100	200	300	400	500	600	700	800	900	1000	TOTAL
1	4	8	14	10	10	10	12	12	10	10	**100**
2	10	12	12	10	10	10	20	12	12	16	**124**
3	0	0	0	0	0	0	0	0	0	0	**0**
4	0	0	0	0	0	0	0	0	2	0	**2**
5	0	2	0	2	2	0	2	0	0	2	**10**
6	4	10	12	12	6	14	12	12	14	10	**106**
7	6	10	6	16	12	12	14	18	14	12	**120**
8	0	0	0	0	2	2	0	0	0	0	**4**
9	2	2	2	0	0	2	0	2	2	0	**12**
10	0	0	0	0	0	0	0	0	0	0	**0**

A/T/S Algorithm/Test/Score: 1 Fast-IAMB/JT; 2 GS/JT; 3 HC/AIC; 4 HC/BDE; 5 HC/BIC; 6 IAMB / JT; 7 Inter-IAMB/JT; 8 Tabu/AIC; 9 Tabu/BDE; 10 Tabu/BIC.

Table 3
Quantity of acceptable Goodness-of-Fit Indexes and X^2/df by number of replicates on bootstrap.

A/T/S	100	200	300	400	500	600	700	800	900	1000
1	4	2	2	3	3	3	2	2	1	3
	17.7	43.2	33.3	32.0	31.2	30.6	33.0	32.2	42.7	31.2
2	2	3	1	1	1	2	1	1	1	1
	59.4	40.5	39.7	76.0	64.5	58.3	68.4	41.2	39.7	42.0
3	4	4	5	4	4	5	4	4	4	5
	3.3	3.7	2.3	3.6	3.3	2.3	3.2	3.6	3.7	2.3
4	3	4	4	3	4	3	4	3	3	4
	23.8	20.5	20.1	23.8	20.1	23.8	20.1	23.8	23.8	20.1
5	4	4	4	4	4	4	4	4	4	4
	17.9	20.1	17.9	20.5	20.5	17.9	20.5	17.9	17.9	20.5
6	3	3	3	2	3	2	1	1	1	3
	34.6	39.5	38.5	37.7	20.5	44.7	39.3	39.7	48.2	35.4
7	3	2	3	1	2	1	2	1	2	3
	24.7	38.3	26.8	41.0	41.3	52.3	43.4	53.0	41.0	40.46
8	4	5	5	5	4	4	5	5	5	5
	3.4	2.6	2.5	2.4	4.4	3.4	2.6	2.5	2.5	2.5
9	3	4	3	4	4	3	4	3	3	4
	22.9	17.2	22.9	18.7	18.7	22.9	18.7	20.0	20.0	18.7
10	4	4	4	4	4	4	4	4	4	4
	18.7	18.7	18.7	16.1	18.7	18.7	18.7	18.7	18.7	18.7

A/T/S Algorithm/Test/Score: 1 Fast-IAMB/JT; 2 GS/JT; 3 HC/AIC; 4 HC/BDE; 5 HC/BIC; 6 IAMB / JT; 7 Inter-IAMB/JT; 8 Tabu/AIC; 9 Tabu/BDE; 10 Tabu/BIC.

Table 4
Summary of Goodness-of-fitness from the PA model.

A/T/S	CHISQ	DF	PV	RMSEA	SRMR	CFI	GFI	CHISQ/DF
1	1463.95	44	0.00	0.04	0.08	0.84	0.91	33.27
2	1788.06	45	0.00	0.04	0.08	0.80	0.89	39.73
3	64.99	28	0.00	0.01	0.03	1.00	1.00	2.32
4	886.17	44	0.00	0.03	0.05	0.91	0.97	20.14
5	768.05	43	0.00	0.03	0.05	0.92	0.97	17.86
6	1772.07	46	0.00	0.04	0.08	0.81	0.94	38.52
7	1233.84	46	0.00	0.03	0.06	0.87	0.96	26.82
8	72.03	29	0.00	0.01	0.03	1.00	1.00	2.48
9	1028.62	45	0.00	0.03	0.05	0.89	0.97	22.86
10	823.96	44	0.00	0.03	0.05	0.91	0.97	18.73

A/T/S Algorithm/Test/Score: 1 Fast-IAMB/JT; 2 GS/JT; 3 HC/AIC; 4 HC/BDE; 5 HC/BIC; 6 IAMB / JT; 7 Inter-IAMB/JT; 8 Tabu/AIC; 9 Tabu/BDE; 10 Tabu/BIC.
CHISQ Chi-Square; DF degree-of-freedom ; PV P-Value; RMSEA root mean square error of approximation; SRMR standardized root mean square residual; CFI comparative fit index ; GFI goodness of fit index.
Recommended fitness statistics: PV < 0.05; RMSEA <0.08; SRMR < 0.08; CFI > 0.90 ; GFI > 0.90; CHISQ/DF <3.00.

Table 5
Multiple Logistic Regression and BNPA results.

Path relationship	Variables	MR	PA		Supported	
		OR	Direct & Indirect	Total	MR	BNPA
AGE→HHD	Adult Elderly	Ref. 3.014 (0.936;1.270) *	0.130 * 0.094 **	0,224	Y	Y
HBP→HHD	No HBP Has HBP	Ref. 2.868 (0.881;1.225) *	0.273 * 0.005 **	0.278	Y	Y
STK→HHD	No Stroke Stroke	Ref. 2.387 (0.248;1.436) *	–	–	Y	N
SEX→HHD	Female Male	Ref. 2.051 (0.551;0.889) *	0.134 * 0.025 **	0.159	Y	Y
DIA→HHD	Not Diabetic Diabetic	Ref. 1.760(0.335;0.789) *	–	–	Y	N
SMK→HHD	Non Smoker Smoker	Ref. 1.779 (0.383;0.775) *	0.148* −0,042 **	0.106	Y	Y
BMI→HHD	Normal Obese Overweight	Ref. 1.082 (−0.116;0.276)_ 1.313 (0.064;0.480) _	–	–	N	N
INC→ HHD	<20k 20k–39k 40k–59k 60k–79k 80k+	Ref. 0.758 (−0.609;0.067) _ 0.773 (−0.584;0.082) _ 0.717 (−0.674;0.019) _ 0.698 (−0.671;−0.033) *	–	–	Y	Y
BYC→HHD	No bike Bike	Ref. 0.906 (−0.675;0.401) _	–	–	N	N
WLK→HHD	No walk Walk	Ref. 0.877 (−0.383;0.107) _	–	–	N	N
WRK→HHD	No work Work	Ref. 0.735 (−0.615;−0.017) _	–	–	N	N
ALC→HHD	Non alcoholic Alcoholic	Ref. 0.583 (−0.726;−0.348) *	−0.155*	−0,155	Y	Y
RUN→HHD	No run Run	Ref. 0.498 (−0.994; −0.418) *	−0.076 * −0.018**	−0.094	Y	Y

*Significant at P < 0.05 in the multivariate Model and PA model; **Indirect effect; MR—multiple logistic regression, OR—odds ratio.

of the PA, except for *stroke* and *diabetes*, were similar and suggest the same positive/negative considerable influence on *heart disease*.

5. Conclusion

This study proposes novel computational software using a hybrid BN–PA-based approach to help researchers to build a PA model based on data set in a semi-automatic way. The study was proposed to fill the existing gap in the PA model that is cited by [14]: "*PA will not discover the causal relationship, but it will combine the quantitative information given by correlations coefficients to give us a quantitative interpretation,*" also cited by [5,6]. The results of BNPA matched 85% of MR results (11 of 13).

An advantage of this approach is the possibility of novice researchers building PA models with some, but not total, help and opinions of statisticians and SEM/PA experts. It does not mean that these experts are not necessary. Instead, this software increases clinical researchers' ability, especially novices, to learn this concept in practice and create their initial predictive models for clinical studies. In this way, more models will be created, and experts in SEM/PA will have more time to assist in their evaluation. Another advantage is that the researchers can benefit from the flexibility provided by the bnlearn [40] to insert or remove edges between variables (using the black-list and white-list) before starting the BNSL and using previous knowledge. For this study, only categorical dichotomous and ordinal data were used. As a next step, executing experiments with continuous data only and with mixed (continuous, categorical dichotomous, and ordinal) data is planned. Also, other R packages that built BNs will be integrated into BNPA.

CRediT authorship contribution statement

Elias Cesar Araujo de Carvalho: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Project administration, Resources, Software, Validation, Visualization, Writing - original draft, Review & editing. **Joao Ricardo Nickenig Vissoci:** Conceptualization, Formal analysis, Supervision. **Luciano de Andrade:** Conceptualization, Formal analysis, Supervision. **Wagner de Lara Machado:** Conceptualization, Formal analysis, Supervision. **Emerson Cabrera Paraiso:** Conceptualization, Methodology, Supervision. **Julio Cesar Nievola:** Conceptualization, Methodology, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This analysis is based on the Statistics Canada Canadian Community Health Survey Microdata File which contains anonymized data collected in 2011–2012. Coordenaco de Aperfeioamento de Pessoal de Nível Superior (CAPES), a foundation linked to the Ministry of Education (MEC) of Brazil – Process CSF-PVE-99999.004179/2015-03, Pontificia Universidade Católica do Paraná, Brazil and Universidade Estadual de Maringá, Brazil funded this study.

References

[1] M.L. Petersen, M.J. van der Laan, Causal models and learning from data: integrating causal modeling and statistical estimation, Epidemiology 25 (3) (2014) 418, http://dx.doi.org/10.1097/ede.0000000000000078.

[2] D.B. Rubin, Estimating causal effects of treatments in randomized and nonrandomized studies, J. Educ. Psychol. 66 (5) (1974) 688, http://dx.doi.org/10.1037/h0037350.

[3] J. Pearl, Causality: Models, Reasoning and Inference, vol. 9, Cambridge University Press, Cambridge, MA, USA, 2000, pp. 10–11, http://dx.doi.org/10.3923/rjmsci.2015.272.278.

[4] S. Greenland, J. Pearl, J.M. Robins, Causal diagrams for epidemiologic research, Epidemiology 3 (1999) 7–48.

[5] A.A. Beaujean, Latent variable modeling using R: A step-by-step guide, Routledge, 2014, http://dx.doi.org/10.4324/9781315869780.

[6] R.B. Kline, Principles and Practice of Structural Equation Modeling, Guilford publications, 2012, http://dx.doi.org/10.1080/10705511.2012.687667.

[7] B.M. Byrne, Structural equation modeling with AMOS, EQS, and LISREL: Comparative approaches to testing for the factorial validity of a measuring instrument, Int. J. Test. 1 (1) (2001) 55–86, http://dx.doi.org/10.1207/S15327574IJT0101_4.

[8] Y. Rosseel, Lavaan: An R package for structural equation modeling and more. Version 0.5–12 (BETA), J. Stat. Softw. 48 (2) (2012) 1–36, http://dx.doi.org/10.18637/jss.v048.i02.

[9] J. Fox, Teacher's corner: structural equation modeling with the sem package in R, Struct. Equ. Model. 13 (3) (2006) 465–486, https://www.tandfonline.com/doi/abs/10.1207/s15328007sem1303_7.

[10] M.C. Neale, M.D. Hunter, J.N. Pritikin, M. Zahery, T.R. Brick, R.M. Kirkpatrick, …, S.M. Boker, OpenMx 2.0: Extended structural equation and statistical modeling, Psychometrika 81 (2) (2016) 535–549, http://dx.doi.org/10.1007/s11336-014-9435-8.

[11] E. Kupek, Beyond logistic regression: structural equations modelling for binary variables and its application to investigating unobserved confounders, BMC Med. Res. Methodol. 6 (1) (2006) 13, http://dx.doi.org/10.1186/1471-2288-6-13.

[12] L.D.A.F. Amorim, R.L. Fiaccone, C.A.S. Santos, T.N.D. Santos, L.T.L. de Moraes, N.F. Oliveira, M.L. Barreto, Structural equation modeling in epidemiology, Cadernos de Saúde Pública 26 (12) (2010) 2251–2262, http://dx.doi.org/10.1590/S0102-311X2010001200004.

[13] Z. Zhang, Structural equation modeling in the context of clinical research, Ann. Transl. Med. 5 (5) (2017) http://dx.doi.org/10.21037/atm.2016.09.25.

[14] S. Wright, Path coefficients and path regressions: Alternative or complementary concepts, Causal Model. Soc. Sci. 10 (1960) 1–114, http://dx.doi.org/10.2307/2527551.

[15] M.P. Fox, J.K. Edwards, R. Platt, L.B. Balzer, The critical importance of asking good questions: The role of epidemiology doctoral training programs, Am. J. Epidemiol. 189 (4) (2020) 261–264, http://dx.doi.org/10.1093/aje/kwz233.

[16] Statistics. Canada, Statistics Canada Canadian Community Health Survey (CCHS) Annual component User guide 2012 and 2011-2012 Microdata files, 2013, -eng, https://doi.org/10.25318/1310044701.

[17] M.J. Flores, A. Nicholson, A. Brunskillc, K. Korbb, S. Mascarod, Incorporating expert knowledge when learning Bayesian network structure: Heart Failure as a Case Study$, Technical Report 2010/3, Bayesian Intelligence, 2010, http://dx.doi.org/10.1016/j.artmed.2011.08.004.

[18] E. Gatti, D. Luciani, F. Stella, A continuous time Bayesian network model for cardiogenic heart failure, Flex. Serv. Manuf. J. 24 (4) (2012) 496–515, http://dx.doi.org/10.1007/s10696-011-9131-2.

[19] M.S.A. Sawa, A. Naba, H.S. Dachlan, BayesIan network expert system for early diagnosis of heart diseases, J. EECCIS 7 (2) (2014) 171–178, ISSN (Online): 2460-8122.

[20] K. Orphanou, A. Stassopoulou, E. Keravnou, DBN-Extended: a dynamic Bayesian network model extended with temporal abstractions for coronary heart disease prognosis, IEEE J. Biomed. Health Inform. 20 (3) (2015) 944–952, http://dx.doi.org/10.1109/JBHI.2015.2420534.

[21] B. Al-Hamadani, An emergency unit support system to diagnose chronic heart failure embedded with SWRL and Bayesian network, Int. J. Adv. Comput. Sci. Appl. 7 (7) (2016) 446–453, http://dx.doi.org/10.14569/IJACSA.2016.070761.

[22] M. Singh, L.M. Martins, P. Joanis, V.K. Mago, Building a cardiovascular disease predictive model using structural equation model & fuzzy cognitive map, in: 2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), IEEE, 2016, pp. 1377–1382, http://dx.doi.org/10.1109/FUZZ-IEEE.20116.7737850.

[23] L.S.C. de Oliveira, R.V. Andreao, M. Sarcinelli Filho, BayesIan network with decision threshold for heart beat classification, IEEE Lat. Am. Trans. 14 (3) (2016) 1103–1108, http://dx.doi.org/10.1109/TLA.2016.7459585.

[24] Z. Wei, X.L. Zhang, H.X. Rao, H.F. Wang, X. Wang, L.X. Qiu, Using the Tabu-search-algorithm-based Bayesian network to analyze the risk factors of coronary heart diseases, Zhonghua liu xing bing xue za zhi= Zhonghua liuxingbingxue zazhi 37 (6) (2016) 895–899, http://dx.doi.org/10.3760/cma.j.issn.0254-6450.2016.06.031.

[25] Y.S. Kim, A path analysis model of health-related quality of life in patients with heart failure, Korean J. Adult Nurs. 19 (4) (2007) 547–555.

[26] W. Chen, S.R. Srinivasan, G.S. Berenson, Path analysis of metabolic syndrome components in black versus white children, adolescents, and adults: the Bogalusa Heart Study, Ann. Epidemiology 18 (2) (2008) 85–91, http://dx.doi.org/10.1016/j.annepidem.2007.07.090.

[27] S.S. Ghazavi Shariat Panahi, R. Ansari, F. Shamnaz, M. Kahouei, A path analysis model of ischemic heart disease patient's preferences in obtaining health information and factors affecting them, Res. J. Med. Sci. 27 (2015) 2–278, http://dx.doi.org/10.3923/rjmsci.2015.272.278.

[28] H.D. de Heer, H.G. Balcazar, F. Castro, L. Schulz, A path analysis of a randomized promotora de salud cardiovascular disease–prevention trial among at-risk Hispanic adults, Health Educ. Behavior 39 (1) (2012) 77–86, http://dx.doi.org/10.1177/1090198111408720.

[29] E. Vellone, B. Riegel, F. D'Agostino, R. Fida, G. Rocco, A. Cocchieri, R. Alvaro, Structural equation model testing the situation-specific theory of heart failure self-care, J. Adv. Nurs. 69 (11) (2013) 2481–2492, http://dx.doi.org/10.1111/jan.12126.

[30] M.A. Castro, V. Baltar, D.M. Marchioni, R.M. Fisberg, CO034. Overall and central obesity indicators are different predictors of metabolic cardiovascular disease risk factors: A structural equation model approach, Archivos Latinoamericanos de Nutrición 65 (Suplemento 2) (2015).

[31] C. Lacave, F.J. Díez, A review of explanation methods for Bayesian networks, Knowl. Eng. Rev. 17 (2) (2002) 107–127, http://dx.doi.org/10.1017/S026988890200019X.

[32] K. Thulasiraman, M.N. Swamy, Graphs: Theory and Algorithms, John Wiley & Sons, 2011, http://dx.doi.org/10.1002/9781118033104.

[33] S. Thornley, R.J. Marshall, S. Wells, R. Jackson, Using directed acyclic graphs for investigating causal paths for cardiovascular disease, J. Biom. Biostat. 4 (2013) 182, http://dx.doi.org/10.4172/2155-6180.1000182.

[34] M. Scutari, J.B. Denis, Bayesian Networks: With Examples in R, CRC press, 2014, http://dx.doi.org/10.1111/anzs.12220.

[35] J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Elsevier, 2014.

[36] B. Han, M. Park, X.W. Chen, A Markov blanket-based method for detecting causal SNPs in GWAS, in: BMC Bioinformatics (Vol. 11, No. S3, P. S5), BioMed Central, 2010, http://dx.doi.org/10.1186/1471-2105-11-S3-S5.

[37] K.B. Korb, A.E. Nicholson, Bayesian Artificial Intelligence, CRC press, 2010, http://dx.doi.org/10.1201/b10391.

[38] K. Abbas, A.R. Mikler, A. Ramezani, S. Menezes, Computational epidemiology: Bayesian disease surveillance, in: Advances in Bioinformatics and Its Applications, 2005, pp. 95–106.

[39] R. Nagarajan, M. Scutari, S. Lèbre, BayesIan networks in R, Springer 122 (2013) 125–127.

[40] M. Scutari, Learning Bayesian networks with the bnlearn R package, 2009, http://dx.doi.org/10.18637/jss.v035.i03, arXiv preprint arXiv:0908.3817.

[41] T. Verma, J. Pearl, Equivalence and Synthesis of Causal Models, UCLA, Computer Science Department, 1991, pp. 220–227.

[42] P. Spirtes, C.N. Glymour, R. Scheines, D. Heckerman, Causation, Prediction, and Search, MIT press, 2000.

[43] D. Edwards, Introduction To Graphical Modelling, Springer Science & Business Media, 2012.

[44] P. Legendre, Comparison of permutation methods for the partial correlation and partial Mantel tests, J. Stat. Comput. Simul. 67 (1) (2000) 37–73, http://dx.doi.org/10.1080/00949650008812035.

[45] I. Tsamardinos, G. Borboudakis, Permutation testing improves Bayesian network learning, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, Berlin, Heidelberg, 2010, pp. 322–337.

[46] J. Hausser, K. Strimmer, Entropy inference and the James–Stein estimator, with application to nonlinear gene association networks, J. Mach. Learn. Res. 10 (7) (2009).

[47] O. Ledoit, M. Wolf, Improved estimation of the covariance matrix of stock returns with an application to portfolio selection, J. Empir. Finance 10 (5) (2003) 603–621, http://dx.doi.org/10.1016/S0927-5398(03)00007-0.

[48] D.M. Chickering, A transformational characterization of equivalent bayesian network structures, in: Proceedings of the Eleventh Annual Conference on Uncertainty in Artificial Intelligence, 1995, pp. 87–98.

[49] D.M. Chickering, D. Heckerman, A comparison of scientific and engineering criteria for Bayesian model selection, Stat. Comput. 10 (2000) 55–62, http://dx.doi.org/10.1023/A:1008936501289.

[50] M. Scutari, C. Vitolo, A. Tucker, Learning Bayesian networks from big data with greedy search: Computational complexity and efficient implementation, Stat. Comput. (2019) online first DOI: 10.1007/s11222-019-09857-1.

[51] D. Heckerman, D. Geiger, D.M. Chickering, Learning Bayesian networks: The combination of knowledge and statistical data, Mach. Learn. 20 (3) (1995) 197–243, http://dx.doi.org/10.1023/A:1022623210503.
[52] R. Castelo, A. Siebes, Priors on network structures. Biasing the search for Bayesian networks, Internat. J. Approx. Reason. 24 (1) (2000) 39–57, http://dx.doi.org/10.1016/S0888-613X(99)00041-9.
[53] D. Geiger, D. Heckerman, Learning gaussian networks, in: Proceedings of the Tenth Annual Conference on Uncertainty in Artificial Intelligence, 1994, pp. 235–243, https://doi.org/10.1016/B978-1-55860-332-5.50035-3.
[54] J. Kuipers, G. Moffa, D. Heckerman, Addendum on the scoring of Gaussian directed acyclic graphical models, Ann. Statist. 42 (4) (2014) 1689–1691, http://dx.doi.org/10.1214/14-AOS1217.
[55] D. Margaritis, Learning Bayesian Network Model Structure from Data, Carnegie-Mellon Univ Pittsburgh Pa School of Computer Science, 2003.
[56] I. Tsamardinos, C.F. Aliferis, A.R. Statnikov, E. Statnikov, Algorithms for large scale Markov blanket discovery, in: FLAIRS conference, vol. 2, 2003, pp. 376-380.
[57] S. Yaramakala, D. Margaritis, Speculative Markov blanket discovery for optimal feature selection, in: Fifth IEEE International Conference on Data Mining (ICDM'05), IEEE, 2005, p. 4, http://dx.doi.org/10.1109/ICDM.2005.134.
[58] S.J. Russell, P. Norvig, Artificial Intelligence: A Modern Approach, Prentice Hall, 2009.
[59] N. Friedman, M. Goldszmidt, A. Wyner, Data analysis with Bayesian networks: A bootstrap approach, 2013, arXiv preprint arXiv:1301.6695.
[60] B. Efron, R.J. Tibshirani, An Introduction To the Bootstrap, CRC press, 1994, http://dx.doi.org/10.1201/9780429246593.
[61] G. Claeskens, N.L. Hjort, Model Selection and Model Averaging, Cambridge Books, 2008, http://dx.doi.org/10.1017/CBO9780511790485.
[62] J.F. Hair, Multivariate Data Analysis, 2005, Pearson.
[63] L.A. Cole, P.R. Kramer, Human Physiology, Biochemistry and Basic Medicine, Academic Press, 2015, http://dx.doi.org/10.1016/C2014-0-04282-7.