Large-scale inference of conjunctive Bayesian networks

Hesam Montazeri1,2, Jack Kuipers1,2, Roger Kouyos3,4, Jürg Böni5, Sabine Yerly6, Thomas Klimkait7, Vincent Aubert8, Huldrych F. Günthard3,4, Niko Beerenwinkel1,2,* and The Swiss HIV Cohort Study

1Department of Biosystems Science and Engineering, ETH Zurich, Basel, Switzerland, 2SIB Swiss Institute of Bioinformatics, Basel, Switzerland, 3Division of Infectious Diseases and Hospital Epidemiology, University Hospital Zurich, University of Zurich, Zurich, Switzerland, 4Institute of Medical Virology, 5Swiss National Center for Retroviruses, Institute of Medical Virology, University of Zurich, Zurich 8057, Switzerland, 6Laboratory of Virology, Division of Infectious Diseases, Geneva University Hospital, Geneva, Switzerland, 7Department of Biomedicine-Petersplatz, University of Basel, Basel, Switzerland and 8Division of Immunology and Allergy, University Hospital Lausanne, Lausanne, Switzerland

*To whom correspondence should be addressed

Abstract

The continuous time conjunctive Bayesian network (CT-CBN) is a graphical model for analyzing the waiting time process of the accumulation of genetic changes (mutations). CT-CBN models have been successfully used in several biological applications such as HIV drug resistance development and genetic progression of cancer. However, current approaches for parameter estimation and network structure learning of CBNs can only deal with a small number of mutations (<20). Here, we address this limitation by presenting an efficient and accurate approximate inference algorithm using a Monte Carlo expectation-maximization algorithm based on importance sampling. The new method can now be used for a large number of mutations, up to one thousand, an increase by two orders of magnitude. In simulation studies, we present the accuracy as well as the running time efficiency of the new inference method and compare it with a MLE method, expectation-maximization, and discrete time CBN model, i.e. a first-order approximation of the CT-CBN model. We also study the application of the new model on HIV drug resistance datasets for the combination therapy with zidovudine plus lamivudine (AZT + 3TC) as well as under no treatment, both extracted from the Swiss HIV Cohort Study database.

Availability and implementation: The proposed method is implemented as an R package available at https://github.com/cbg-ethz/MC-CBN.

Contact: niko.beerenwinkel@bsse.ethz.ch

Supplementary information: Supplementary data are available at Bioinformatics online.

1 Introduction

In many biological systems, there are constraints on the order in which genetic alterations (mutations) fixate in a population of biological entities such as viruses and cancer cells (Lozovsky et al., 2009; Poelwijk et al., 2007; Weinreich et al., 2006). In HIV infection, drug resistance development is due to the accumulation of resistance mutations, i.e. point mutations in the viral genome that increase viral fitness under the selective pressure of antiretroviral drugs (Seifert et al., 2015). Since resistance mutations give a strong selective advantage under a fixed drug pressure, their fixation is almost an irreversible process. More than 70 mutations in the reverse transcriptase, protease, envelope and integrase genes of viral genome are known to be associated with HIV drug resistance (Wensing et al., 2014). Models of the accumulation of mutations have been proposed and extensively used in HIV treatment optimization (Altmann et al., 2009; Beerenwinkel et al., 2005; Deforche et al., 2008; Prosperi et al., 2009; Beerenwinkel et al., 2013; Montazeri et al., 2013). Similarly, genetic progression of cancer is caused by the accumulation of mutations such as changes in single nucleotides or copy numbers (Hanahan and Weinberg, 2011; Merlo et al., 2006). Around 140 mutations are known to have effect on cancer progression (Vogelstein et al., 2013). Mathematical and statistical models
have also been used to describe the genetic progression of cancer (Desper et al., 1999; Gerstung et al., 2011; Heydembrecht et al., 2004; Hjelm et al., 2006; Jang et al., 2000; Mattias, 2004; Rahnenführer et al., 2005).

In the present paper, we study the continuous time conjunctive Bayesian network (CT-CBN) model for describing the accumulation of mutations (Beerenwinkel and Sullivant, 2009). The CT-CBN is a continuous time Markov chain defined on a partially ordered set (poset) of mutations. The poset encodes temporal ordering among mutations by assuming that the waiting time of each mutation begins only after all its predecessor mutations have already occurred. To each mutation, a rate of evolution is assigned, which includes generation of the mutation and its fixation in the population. In large populations, mutations will be generated almost immediately and the waiting time is effectively dominated by the fixation time. Hence, we use the terms “rate of evolution” and “fixation rate” interchangeably throughout this paper. The fixation rates characterize the waiting time process of mutations.

Maximum likelihood estimation (MLE) is mainly used for inference of CBN models from censored cross-sectional genotypes. Observed genotypes are censored since explicit occurrence times of mutations are not known and it is only possible to measure which mutations are not directly observable. Hence, the expectation–maximization (EM) algorithm, which can properly handle unobserved latent variables, has been extensively used for inference of CBN models (Beerenwinkel and Sullivant, 2009; Gerstung et al., 2009; Beerenwinkel et al., 2007, 2011, Montazeri et al., 2015). In (Sakoparnig and Beerenwinkel, 2012), a Bayesian inference approach is used for network learning and parameter estimation of the discrete time CBN. However, all the above-mentioned inference methods for CBN models are only feasible for small sets of mutations of less than around 20 mutations.

The main goal of the present paper is to address this limitation and to develop inference schemes for CBN models that scale to several hundreds of mutations. This model is particularly useful in modeling HIV evolution. Modeling the dynamics and dependencies among hundreds of HIV mutations makes it possible to better quantify drug resistance development and consequently improve the prediction of therapy response. In the present paper, we propose two novel inference methods for parameter estimation of the CBN models. The first one is an exact MLE method that uses general-purpose optimization methods to directly maximize the observed likelihood. For this method, a new formulation of the likelihood and its gradient function are given based on the properties of continuous-time Markov chains. This method is feasible for up to about 30 mutations. However, for some classes of posets, e.g. empty posets, it is feasible for much larger posets with several hundreds of mutations. The second method is an approximate inference scheme, namely a Monte Carlo expectation–maximization (MC–EM) algorithm (Wei and Tanner, 1990) with importance sampling. We demonstrate that the MC–EM method is almost as accurate as the exact MLE method in simulation studies. The method is applicable for parameter estimation of large posets with around 1000 mutations, an increase by two orders of magnitude. For network learning of the CT-CBN model, we adapted the mixture-model approach outlined in (Beerenwinkel and Sullivant, 2009; Montazeri et al., 2015) and address some of its limitations in order to make it applicable for large numbers of mutations.

The rest of this paper is organized as follows. In Section 2, after giving a brief introduction to the CT-CBN model, we present a new MLE method based on continuous time Markov chain for the CBN model. In addition, we propose an approximate large-scale inference method using MC–EM with importance sampling for the CBN model. We close this section by discussing how to reconstruct the underlying network topology from observed data. Section 3 reports the performance of the MC–EM method in comparison to other inference methods. In addition, we analyze thoroughly the HIV drug resistance development in a clinical dataset of Swiss HIV infected patients. We close with conclusions in Section 4.

2 Methods

The CT-CBN is defined on a set of genetic events (mutations) \( P \) and a partial order \( \preceq \) among the mutations. A relation \( e_1 \prec e_2 \) in \( (P, \preceq) \) indicates that mutation \( e_2 \) can only happen after the occurrence of \( e_1 \). The relation \( e_1 \prec e_2 \) is called a cover relation if it exists in the transitive reduction of \( (P, \prec) \), i.e. if \( \exists e' \in P \setminus \{e_1, e_2\} \) with \( e_1 \prec e' \prec e_2 \). A genotype \( g \) is a subset of \( P \). The set of all genotypes compatible with the order constraints of the partially ordered set (poset) \( P \) is denoted by \( J(P) \). For example, the poset shown in Figure 1(a) consists of four mutations \( \{1, 2, 3, 4\} \) subject to the relations \( 1 \prec 3, 2 \prec 3, \) and \( 2 \prec 4 \). Its corresponding genotype lattice, shown in Figure 1(b), is \( J(P) = \{\emptyset, \{1\}, \{2\}, \{1, 2\}, \{2, 4\}, \{1, 2, 3\}, \{1, 2, 4\}, \{1, 2, 3, 4\}\} \). The set Exit(\( g \)) is defined as \( \{e \in P | e \not\in g, g \cup e \in J(P)\} \), i.e. the subset of events in \( P \setminus g \) that can happen next. In the continuous time CBN, for each mutation \( i \) in \( P \) the waiting time to its occurrence, denoted as \( T_a \), is defined as

\[
T_a = \max_{j \in \text{pa}(i)} T_j + Z_i ,
\]

where \( \text{pa}(i) \) is the set of parents of mutation \( i \) for the poset \( P \). The expression \( \max_{j \in \text{pa}(i)} T_j \) indicates the parent mutations happen first and then the waiting process for the mutation \( i \) itself begins. The time needed exclusively for the mutation \( i \) is represented by \( Z_i \) and is assumed to be an exponentially distributed random variable \( Z_i \sim \text{Exp}(\lambda_i) \). The joint density function of \( T = (T_1, \ldots, T_p) \) is

\[
f(t) = \prod_{i=1}^{p} f_i(t_i | \text{pa}(i)) = \prod_{i=1}^{p} f_i(t_i - \max_{j \in \text{pa}(i)} T_j)
\]

where \( p = |P| \) and the density function \( f_i \) is the univariate exponential probability density function with rate \( \lambda_i \). A genotype with mutation times \( t = (t_1, \ldots, t_p) \) is not compatible with the poset \( P \) if the density is zero, or equivalently, if there exists an event \( i \in P \) such that \( t_i < \min_{j \in \text{pa}(i)} t_j \). In real world applications, the random vector \( T \) is not observed and the mutations are only observable at a certain sampling (sequencing) time, denoted by \( t_o \), which itself might not be observable in all settings. Formally, the observed genotype \( g \) at time \( t_o \) is defined as \( g = \{e | t_e < t_o\} \). The reader can refer to (Beerenwinkel and Sullivant, 2009; Gerstung et al., 2009; Montazeri et al., 2015) for a more detailed introduction to CBN models. The inference of the model consists of two parts namely (i) parameter estimation: estimation of the exponential rates \( \lambda_i \) for \( i = 1, \ldots, p \) for a given poset and (ii) network learning. For the parameter estimation, we first propose a new MLE method that directly maximizes the observed likelihood of the CBN model. In this approach, the likelihood and its gradient functions are formulated using some properties of continuous time Markov chains. In addition, we propose a large-scale efficient approximate algorithm using Monte Carlo expectation...
maximization based on importance sampling for the parameter estimation. Finally, a mixture-model approach outlined in Section 2.3 is used to find the maximum likelihood estimate of true poset.

2.1 Exact inference using continuous time Markov chain
The CT-CBN is a continuous time Markov chain with transition rate matrix \( S \) defined as (Beerenwinkel and Sullivant, 2009),

\[
S_{i(g)j(h)} = \begin{cases} 
\lambda_{i(h)} & \text{if } g = h \\
\lambda_{i(g)} & \text{if } g < h, |g| = 1 \\
0 & \text{otherwise}
\end{cases}
\]

where \( i(g) \) is the index of the corresponding row or column to genotype \( g \) in the transition matrix.

Example 1 For the poset shown in Figure 1(a), the transition rate matrix \( S \) is

\[
S = \begin{pmatrix}
0 & 1 & 2 & 2.4 & 1.2 & 1.2.3 & 1.2.4 & 1.2.3.4 \\
1 & 0 & -\lambda_2 & \lambda_1 & 0 & 0 & 0 & 0 \\
2 & 0 & -\lambda_3 & \lambda_2 & 0 & 0 & 0 & 0 \\
1.2 & 0 & 0 & 0 & -\lambda_3 & \lambda_2 & 0 & 0 \\
1.2.4 & 0 & 0 & 0 & 0 & -\lambda_4 & \lambda_3 & 0 \\
1.2.3 & 0 & 0 & 0 & 0 & 0 & -\lambda_4 & \lambda_3 \\
1.2.3.4 & 0 & 0 & 0 & 0 & 0 & 0 & -\lambda_4
\end{pmatrix}
\]

the non-zero off-diagonal elements of the transition matrix are the transition rates from each genotype to its successive genotypes in the genotype lattice, also shown in Figure 1(b).

The transition probability from genotype \( g \) to \( h \) in time \( t \) is denoted by \( p_{gh}(t) \) and is equal to the element \((i(g), i(h))\) of the matrix exponential \( e^{\lambda t} \). Consequently, the probability that genotype \( g \) will be observed at time \( t \), starting from the wild-type, \( \theta \), at time zero is simply \( p_{g\theta}(t) \). In terms of hidden random vector \( T \) and observed sampling time \( t_{\text{obs}} \), this is equivalent to the probability that events in \( g \) happen before the sampling time \( t \) and other events happen after \( t_{\text{obs}} \), \( \Pr(\max_{g \in P} T_e < t, \min_{g \in P} T_e > t) \). The log-likelihood of the exponential rates of the fixed poset \( P \) for given observations \( D \) (pairs of genotypes and sampling times) is then

\[
L(\lambda) = \sum_{(g, t) \in D} \log p_{g\theta}(t)
\]

and the gradient of the log-likelihood function, \( \nabla L(\lambda) = (\partial L(\lambda)/\partial \lambda_1, \ldots, \partial L(\lambda)/\partial \lambda_d) \), is given by

\[
\frac{\partial L(\lambda)}{\partial \lambda_i} = \sum_{(g, t) \in D} \frac{\partial p_{g\theta}(t)}{\partial \lambda_i} p_{g\theta}(t)
\]

The term \( \partial p_{g\theta}(t) / \partial \lambda_i \) can be computed using the matrix exponential of an augmented matrix (Fung, 2004). Here, the \( 2M \times 2M \) augmented matrix is

\[
A_S = \begin{bmatrix}
S & 0 \\
0 & S
\end{bmatrix}
\]

where \( M \) is the size of transition matrix \( S \). The matrix exponential of the submatrix \( (A_S)_{(M+1:2M,1:M)} \) contains the derivative of transition probabilities with respect to parameter \( \lambda_i \). In particular, \( \partial p_{g\theta}(t) / \partial \lambda_i \) can be calculated by \((M + 1, i(g))^\text{th} \) element of the matrix exponential of the augmented matrix.

Example 2 For the poset shown in Figure 1(a), the matrix \( \partial S / \partial \lambda_1 \) is

\[
\frac{\partial S}{\partial \lambda_1} = \begin{pmatrix}
0 & 1 & 2 & 2.4 & 1.2 & 1.2.3 & 1.2.4 & 1.2.3.4 \\
1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\
2 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\
1.2 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\
1.2.4 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\
1.2.3 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\
1.2.3.4 & 0 & 0 & 0 & 0 & 0 & 0 & -1
\end{pmatrix}
\]

The ML estimation can be performed by standard gradient ascent optimization methods such as the L-BFGS method (Liu and Nocedal, 1989) or by derivative-free optimization algorithms using a quadratic approximation (Powell, 2006) (package minqa in R).

Since there are more genotypes compatible with sparser posets, the size of the transition matrix is larger for sparser posets. For a given number of mutations \( p \), the transition matrix has the maximum size \( 2^p \times 2^p \) for the empty poset. Hence, the likelihood computation that involves computing matrix exponential is not feasible for large posets. However, when the poset consists of \( m \) independent components \( C_i, i = 1, \ldots, m \), the probability of observing genotype \( g \) can be written as

\[
p_{g\theta}(t) = \prod_{i=1}^{m} p_{g\theta_i}(t)
\]

In particular, for the empty poset with \( p \) mutations, we have \( p \) different components and transition matrices. Each component consists of two genotypes: the wild-type and a mutated genotype. Consequently, we need only to compute \( p \) matrix exponentials of size \( 2 \times 2 \). In general, the complexity of computing the genotype probabilities is determined by the size of the largest component of the poset. In addition, because we are only interested in the first row of \( e^{\lambda t} \), the matrix exponential of the whole matrix \( S \) is not required.
A faster method is to compute the action of matrix exponential $ue^{st}$ where $u = [1, 0, \ldots, 0]$ with size of $1 \times p$ (method `expm` in the package `expm` in R). With these improvements, the likelihood computation and consequently the parameter estimation is possible for larger posets up to around 30 mutations.

In most biological applications, the sampling times are not available. For example, in cancer progression, the start of the tumor evolutionary process is not known. Hence, the time elapsed from the beginning of the process till the sampling time point is not known. In these applications, the sampling time is integrated out for likelihood computation

$$p_{bg} = \int_{t=0}^{\infty} p_{bg}(t) dt. \quad (5)$$

A reasonable assumption for the sampling time is $T_s \sim \text{Exp}(\lambda_s)$, where $\lambda_s$ is the rate of sampling process (Beerenwinkel and Sullivant, 2009). This quantity has been computed in Beerenwinkel and Sullivant, 2009, Theorem 4.1]. A more elegant solution to compute this quantity is given in Appendix A. Since the sampling time is not observed, different connected components are not independent of each other anymore and the probability $p_{bg}$ is not decomposable over the components of the poset. Hence in this case, the likelihood computation is not possible when the number of genotypes compatible with a poset is very large, irrespective of the sparseness of the poset. Finally, one can learn the CT-CBN model when some sampling times are missing and some of them are observed by maximizing the following log-likelihood function

$$\sum_{g, t \in O} \log p_{bg}(t) + \sum_{g \in M} \log p_{bg}$$

where $O$ is the set of observed genotype-sampling time pairs and $M$ is the set of genotypes for which the corresponding sampling times are missing.

### 2.2 Approximate parameter estimation

Since mutation occurrence times, $t_i = 1, \ldots, p$, are not directly observable, the direct optimization of the density (2) to find the rate parameters is not possible. The EM algorithm has previously been used in (Beerenwinkel and Sullivant, 2009; Gerstung et al., 2011; Montazeri et al., 2015) for parameter estimation of a given poset. It is easy to see that time differences, $t_i - \max_{p \in p(i)}$, are the sufficient statistics for estimating $\lambda_i$. The maximum likelihood estimate of $\lambda_i$ is

$$N/\sum_{i=1}^{N} (t_i - \max_{p \in p(i)} f(t_i)) \quad \text{(Beerenwinkel and Sullivant, 2009),}$$

where $f(t_i)$ is the occurrence time of mutation $i$ for the $j$th observation. In the E-step of the EM algorithm, expected values of the time differences are computed for all mutations given the observation $(g, t_i)$ and the estimate of the parameters from the previous iteration. This expectation is defined as $e_i(g, t_i) = E[T_i - \max_{p \in p(i)} f(t_i)]$ and is computed analytically in Theorem 2.5 of (Montazeri et al., 2015). A similar expectation when the sampling time is not observed is computed in (Beerenwinkel and Sullivant, 2009).

$$e_i(g, t_i) = \frac{1}{P_r(g, t_i, P, \lambda)} \int_{t_i = 1} f(t_i) dt$$

$$= \int_{t_i = 1} f(t_i) dt - \max_{p \in p(i)} f(t_i)$$

$$\int_{t_i = 1} f(t_i) dt$$

where $t_i = (g, t_i)$ denotes that the occurrence time vector $t_i$ is in concordance with the observation $(g, t_i)$. Computing these integrals is complicated due to the fact that the density function (2) contains the maximum function in the exponent. In the previous approaches, the above-mentioned integrals were decomposed into simpler integrals over all possible maximal extensions of a given poset. Since the total number of maximal chains is factorial in the number of mutations in the worst case, these methods are not feasible in general for large posets. In this paper, we use Monte Carlo integration by importance sampling to compute $e_i(g, t_i)$. In particular, we draw $L$ samples from a proposal distribution $q(t)$ and $e_i(g, t_i)$ is approximated as

$$e_i(g, t_i) \approx \frac{1}{L} \sum_{l=1}^{L} \frac{f(t_i)}{q(t_i)} = \frac{1}{L} \sum_{l=1}^{L} \frac{f(t_i)}{q(t_i)} - \max_{p \in p(i)} f(t_i)$$

$$= \frac{\sum_i w_k f(t_i) - \max_{p \in p(i)} f(t_i)}{\sum_i w_k} \quad (6)$$

The quantities $w_k = f(t_i)/q(t_i)$ are called importance weights. The Monte Carlo EM (Wei and Tanner, 1990) is called the stochastic EM for the case $L = 1$ (Bishop, 2006; Nielsen 2000). The choice of the proposal distribution plays an important role in efficiency and accuracy of the estimation. We use the following proposal distribution and the equation $T_i = \max_{p \in p(i)} T_i + Z_i$ to generate mutation occurrence times $t_i$ for genotype $g$ and sampling time $t_s$.

$$Z_i \sim \text{TExp}(\lambda_i, 0, \max_{p \in p(i)} t_i), \quad \text{for } i \in g$$

$$Z_i \sim \text{Exp}(\lambda_i) \quad \text{otherwise}$$

where TExp$(\lambda, a, b)$ is an exponential truncated to the intervals $a$ and $b$. Since we use the ancestral sampling method (Bishop, 2006) to sample from the proposal distribution, the maximum of parent occurrence times, i.e. $\max_{p \in p(i)} t_i$ is known before we sample the occurrence time of mutation $i$ using the distribution of $Z_i$. This proposal distribution is a good choice for this problem because all the generated samples are consistent with the observation of interest, $g$ and $t_s$. In addition, due to the memorylessness of exponentials, the proposal distribution of mutation $i \notin g$ is the same as the true conditional distribution, $Z_i | g, t_s$. In the M-step, the new estimate of the mutation rate $i \notin g$ is computed as $\lambda_i^{\text{new}} = N / \sum_{i=1}^{N} e_i(g^t_i, f(t_i))$. It has been shown that averaging the Markov chain improves the estimation (Nielsen, 2000). Hence, we average the last $\times$ maxIter iterations of the Markov chain where $\xi \in (0, 1]$. The MC-EM algorithm is given in Algorithm 1.

### 2.3 Network learning

Now we explain how we perform the network learning. Under the strong assumption that all observed genotypes are perfect realizations of the CBN model, it has been shown that the largest poset compatible with all the observations is the MLE poset (Beerenwinkel and Sullivant, 2009). However, due to the imperfection of the model and the fact that observations are subject to noise, the ML poset is often very sparse for most real-world applications. In (Beerenwinkel and Sullivant, 2009; Montazeri et al., 2015), a mixture model approach was employed to address the problem of noisy observations. We follow a similar approach here and modify the method such that it will be tractable for large posets as well. In this method, the CBN model is extended to allow some degree of violations of the poset relations by observed genotypes. In the extended model, $P_r$ represents the maximal poset in which each relation is violated by at most a...
fraction $\gamma$ of the genotypes. In fact, the extended model is a mixture model with two components. The main component, which is the CBN model, is responsible for generating genotypes compatible to under the CBN component is $P(G|I, P_\gamma, \lambda)$. The second component is a noise component defined as a generative model for the genotypes that are incompatible with the poset $P_\gamma$. The probability of a genotype in the noise component is denoted by $q_\gamma$. In this mixture model, the probability that an observation belongs to the CBN component is denoted by $\pi$, the mixing proportion. The maximum likelihood estimate of $\pi$ is the fraction of genotypes compatible with the poset $P_\gamma$. It is noteworthy that $\gamma$ itself is subject to optimization. In this paper, we define the noise component in two different ways. The first approach is a uniform noise model, in which the probability of observing a genotype incompatible with the poset follows the uniform distribution $q_\gamma = 1/(2^{k_s} - |J(P)|)$ (Beerenwinkel and Sullivant, 2009; Montazeri et al., 2015). In order to compute $q_\gamma$, we need to compute the number of genotypes compatible with the poset, $|J(P)|$. An algorithm that simply calculates the number of compatible genotypes by enumeration is not feasible for large posets. Hence, we use the following efficient divide and conquer algorithm to calculate $|J(P)|$ (Davey and Priestley, 2002),

$$|J(P)| = |J(P) \setminus \{e\}| + |J(P) \setminus \{e \cup \{\}\}|$$

(8) where $e \uparrow = \{y \in P|y \geq e\}$ and $e \uparrow = \{y \in P|y \leq e\}$. The recursion holds for every $e \in P$, and is reasonably fast for posets with up to 50 mutations as well as large sparse posets. For dense large posets (more than 50 mutations and more than 10 edges), we have $q_\gamma \approx 1/(2^{P})$ and it is not necessary to compute $|J(P)|$. However, a limitation of the uniform noise model is that the contribution of the noise component to the likelihood is much smaller than the contribution of the CBN component particularly for larger posets. Consequently, this approach will result in very sparse posets when dealing with large numbers of mutations. To address this limitation, we assume in a second alternative approach that the noise component is the independence model. The independence model is the CBN model with empty poset (i.e. no edges), and it can explain all genotypes incompatible with $P_\gamma$. Its mutation rates are estimated from the incompatible genotypes. A similar noise model has been used in (Beerenwinkel et al., 2005) with an additional latent variable specifying the CBN or noise component that each genotype belongs to. To avoid an additional hidden layer that would require, for example, a nested EM algorithm, here we employ an approximate solution and estimate the mutation rates directly from all incompatible genotypes. This approach has a superior performance in comparison to the uniform noise model for larger posets while still being computationally efficient. We use this approach as the main noise model in this paper.

### 3 Results and discussion

In this section, we assess the performance of the MC–EM algorithm, the MLE method explained in Section 2.1, and the discrete time CBN (D-CBN) in different simulation experiments. In addition, we analyze the application of the CT-CBN model on two HIV drug resistance datasets extracted from the SHCS.

#### 3.1 Simulation study

First, we analyzed the performance of the MC–EM algorithm in rate estimation for different simulation experiments. We investigated different posets with 2, 4, 8, 16, ..., 1024 mutations. For each poset size, we drew 100 random posets from the space of CBNs. To generate a new CBN sample, we first drew a random directed acyclic graph (DAG) by generating a random upper triangular matrix, representing the edges of the DAG. By computing transitive reduction of the generated DAG, we got a new CBN sample. The sampling time distribution was assumed to be exponential with rate $\lambda_0$, $T_i \sim \text{Exp}(\lambda_0)$. However, the EM method has been shown to work well with other sampling time distributions (Montazeri et al., 2015). Mutation rates were drawn uniformly between $\lambda_0$ and $\lambda_0/2$.

We drew $N$ observations, pairs of genotypes and sampling times, for each parameter setting. Since we need more observations for a larger poset, we chose $N$ equal to max(50$p$, 1000), where $p$ is the number of mutations in the poset. In the first simulation experiment, we compared the MC–EM, the D-CBN (Beerenwinkel et al., 2007), and the MLE method in their parameter estimation for a given poset. It has been shown that the D-CBN method is a first-order approximation of the CT-CBN model (Beerenwinkel and Sullivant, 2009). In this experiment, we used the following parameters for the MC–EM method $L = 5$, maxIter = 100 and $\zeta = 0.2$. However, the

---

**Algorithm 1** Parameter estimation of the CBN model using the Monte Carlo EM algorithm based on importance sampling.

**INPUT:** $N$ genotype-sampling time pairs $(g(k), t(k))$, $\lambda_0$: initial mutation rates, $P$: an input poset, maxIter: maximum number of iterations of the EM, $L$: sample size of the importance sampling integration

**OUTPUT:** $\hat{\lambda}_{\text{final}}$: mutation rate estimates.

\[
\begin{align*}
\hat{\lambda}_{\text{new}} & = \hat{\gamma}, \text{ iter } = 1 \\
\text{repeat} & \\
\hat{\lambda}_{\text{old}} & = \hat{\lambda}_{\text{new}} \\
\text{E-step:} & \\
\text{for all } (g(k), t(k)) \text{ in } D \text{ do} & \\
& \text{compute } e_i(g(k), t(k)) \text{ for } i = 1, \ldots, p \text{ using Equation (7) from the } L \text{ samples generated in the previous step.} \\
\text{end for} & \\
M-step: & \\
\hat{\lambda}_{\text{new}} & = N/\sum_{k=1}^{N} e_i(g(k), t(k)) \text{ for } i = 1, \ldots, p. \\
\text{iter} & = \text{iter} + 1 \\
\text{until iter > maxIter} & \\
\hat{\lambda}_{\text{final}} & = \text{the average of the estimated rates over the last } \zeta \times \text{maxIter iterations.}
\end{align*}
\]
choice of parameters does not have a huge effect on the performance of the MC–EM method. We have shown that the MC–EM method converges for a broad range of parameters using the same simulated genotypes (Supplementary Figs S2–S4).

Due to the fact that the transition rate matrix grows exponentially with the number of mutations, in general the MLE method is only feasible for posets with up to 32 mutations. It was not possible to run the MLE method in 22 out 100 posets with size 32 (due to memory constraints). The comparison of log-likelihoods of the MLE, the MC–EM, and the D-CBN is depicted in Figure 2(a), along with the log-likelihoods of true rates. The MC–EM method performs as accurately as the MLE method for small posets. Furthermore, the log-likelihood of the MC–EM is almost equal to the log-likelihood of the true rates for all poset sizes. The similarity of the log-likelihood of the true rates to those from the MC–EM and the MLE method indicates that the estimation process did not get stuck in a low-quality local mode of the likelihood function. The D-CBN method has the worst performance for all poset sizes. Figure 2(b) shows the relative absolute errors between estimated rates and true rates for different parameter estimation methods. For each method, we compute the absolute error, $|\hat{\lambda} - \lambda|$, for each event $e$ of a given poset. The relative absolute error of all mutations is summarized as $\text{median}(|\hat{\lambda} - \lambda|)/\text{median}(\lambda)$. The performance of the MLE and the MC–EM are very similar for small posets. In addition, MC–EM is better than the D-CBN method for all poset sizes. Figures 2(a) and (b) indicate that MC–EM is an accurate method for parameter estimation of small and large posets. A similar result to that shown in Figure 2(b) is obtained if the relative absolute error, defined as $\text{median}(|\hat{\lambda} - \lambda|)/\text{median}(\lambda)$ (Supplementary Fig. S1), is used.

Supplementary Figure S5 shows the comparison of running times of the MC–EM and the MLE method. As mentioned above, the MLE method is not possible for large posets (due to memory and running time constraints), hence its running times are only shown for small posets. The MC–EM running time on a single computer took between a few seconds and an hour for posets from 2 to 1024 mutations. The complexity of the MC–EM is $O(NLp)$ for each iteration of the EM algorithm. In the next experiment, since MC–EM has a stochastic behavior in the parameter estimation, we assessed the sensitivity of the MC–EM against the parameter $L$ and compare it with the EM method in (Montazeri et al., 2015) for the example poset shown at Supplementary Figure S6(a). The results are illustrated for Mutations 2 and 6, as examples of early and late events, in Supplementary Figure S6(b) and (c), respectively. The MC–EM estimates converge to the EM estimates, shown by the dashed lines, for larger $L$.

In the next experiments, we assessed the performance of the network learning method described in Section 2.3 for posets from 2 to 1024 mutations. For each candidate poset $P$, (see Section 2.3 for the definition), we used MC–EM for computing the approximate MLE of rate parameters. In addition to the specifications that were used earlier for drawing genotypes for each poset, we perturbed genotypes by adding observational errors with a per-locus error rate of $\epsilon$ to make the generated genotypes more similar to real-world applications. The error rates were selected as 0, 0.001, 0.005, 0.01 and 0.05 in agreement with the reported values for sequencing error rates in (Hoff, 2009). First, we performed a sensitivity analysis to see the impact of the parameter $L$ in the network learning performance. We observed that the poset learning is not sensitive to this parameter. Estimated posets for $L = 5$ and 100 were the same for 2314 out of 2500 considered posets and only in 66 cases posets were different by more than two edges. Hence, we choose the parameter $L$ equal to 5 in the subsequent experiments.

Figure 3 shows the performance of the network learning method in terms of true positive rate (TPR) and true negative rate (TNR) for different number of mutations $p$ and observational error rates $\epsilon$. The TPR is defined as the proportion of correctly recovered edges of the estimated poset to the total number of edges of the true poset. TNR is defined as $1 -$false positive rate (FPR) where FPR is the number of edges that are falsely estimated to the total number of absent relations in the true poset. Figure 3(a) and (b) shows TPRs and TNRs computed based on transitive closures of estimated posets and true posets, respectively. Supplementary Figure S7 illustrates the same quantities for transitive reductions of these posets. The network learning method performs well for small error rates for all poset sizes. The TPR values of the estimated posets decrease for larger $\epsilon$ and $p$. The algorithm tends to have better performance in terms of TNR particularly for $p \geq 256$ at the expense of decreasing performance in TPR. The increase of TNR values for $p \geq 256$ is due to fact that larger posets are much sparser in comparison to smaller posets. Therefore, it is easier for the network
learning method to estimate a large fraction of true negatives (absent edges) correctly. As shown in Supplementary Figure S8, the network learning running time takes from a minute for small posets to at most an hour for larger posets. The presented methods are available as an R package at https://github.com/cbg-ethz/MC-CBN.

3.2 HIV drug resistance data

In this section, we analyze two HIV drug resistance datasets from Swiss HIV Cohort Study database (SHCS). In particular, we study the accumulation of mutations in the reverse transcriptase (RT) gene of the HIV genome under the drug pressure of the combination therapy zidovudine, 3TC + AZT, as well as under no treatment, with 264 and 615 observations, respectively. The mutation 41L, for example, denotes that the amino acid Leucine (L) is observed at position 41 of the reverse transcriptase gene of HIV genome. We required that the genotype was measured at least 90 days after the onset of treatment and no more than 30 days after treatment end. For all genotypes, the time difference between the treatment start and genotyping is available. Sampling time in HIV can be approximated by the time difference between the start of the treatment (as a proxy for the start of viral progression) to the genotyping time point. Average treatment time is around 700 days for all datasets. We are interested in modeling the accumulation of mutations in the reverse transcriptase for different datasets. For each dataset, we considered all the RT mutations that happen more than 10 times. This helps to get rid of spurious edges in the estimated posets. In total, we obtained 107 and 155 mutations for 3TC + AZT and no-treatment datasets, respectively, and CBN models were estimated using these datasets. According to the simulation studies in the previous section, we estimate the expected TPR of the estimated posets is roughly between 62 and 87% while expected TNR is between 92 and 94%. We obtained these estimates based on the expected TPR and TNR for $p = 128$, which is the closest poset size in the considered simulations to the numbers of mutations in both datasets (Fig. 3).

In this analysis, we mainly focused on the thymidine-analog mutations (TAMs) that arise under selective pressure of zidovudine. In particular, we are interested in two well-known pathways TAM1 (41L, 215Y and 210W mutations) and TAM2 (67N, 70R and 219Q mutations) (Yahi et al., 1999). Figure 4(a) shows the learned network for 3TC + AZT dataset. The estimated CBN model for 3TC + AZT recovered successfully both TAM1 and TAM2 clusters (Fig. 4(b)). The learned network for the dataset 3TC + AZT can explain 69% of the observations. Similarly for no-treatment dataset, the learned poset and the subset of the poset for TAM mutations are shown in Supplementary Figure S9. The corresponding poset as well as inferred temporal relation between TAM mutations for no-treatment dataset are much sparser than those for 3TC + AZT, which indicate that temporal dependencies among mutations are more likely to exist under the selective drug pressure. Quantitatively, the density of the poset for
3TC + AZT is 0.034 as opposed to 0.008 for no-treatment poset. The density of a poset is defined as the ratio of the number of edges of the poset to the number of possible edges. In addition, analysis of estimated average waiting times for both datasets, obtained from the learned CBN models, reveals that mutations tend to happen much faster under the selective drug pressure of 3TC + AZT in comparison to the no-treatment case (Supplementary Table S1).

4 Conclusion

CT-CBN models have been used for modeling the waiting time process of the accumulation of mutations under temporal ordering constraints. In these models, a waiting time process for a mutation only begins after the occurrence of its predecessor mutations. The waiting time of a mutation is assumed to be exponentially distributed in the CT-CBN model. In addition, temporal ordering constraints of the CT-CBN model are encoded by a partially ordered set. Inference of CT-CBN models consists of parameter estimation and network learning. For the parameter estimation, the EM algorithm (Beerenwinkel and Sullivant, 2009; Montazeri et al, 2015) and the MCMC method (Sakoparnig and Beerenwinkel, 2012) have been used. Both approaches are limited to at most 20 mutations.

In this paper, we introduced a Monte Carlo EM algorithm with importance sampling for the parameter estimation of CT-CBN models. We demonstrated that this efficient method can be used for accurate parameter estimation of large networks. For the network learning, we modified a mixture-model approach that have been previously used for CBN models (Beerenwinkel and Sullivant, 2009; Beerenwinkel et al., 2007; Montazeri et al, 2015) and made it computationally feasible for large posets. In future works, we aim to work on more sophisticated network learning algorithms that can handle high sequencing rates, particularly when dealing with high number of mutations. A possible approach is to use search algorithms. However, the search space for large number of mutations is huge and algorithms such as simulated annealing or MCMC do not have any chance to find the optimal network. One possibility is to use the PC algorithm (Spires et al., 2000) to first reduce the search space significantly so that search algorithms can then be employed.

In summary, in this paper we show the MC–EM with importance sampling is an accurate and efficient parameter estimation method for the CBN models and in future works we aim to use this inference algorithm for more complex extensions of the CBN models such as taking into account patient-specific covariates in the model.

Acknowledgements

We thank the patients who participate in the SHCS; the physicians and study nurses, for excellent patient care; the resistance laboratories, for high-quality genotypic drug resistance testing; SmartGene (Zug, Switzerland), for technical support; Brigitte Remy, RN, Martin Rickenbach, MD, Franziska Schöni-Affolter, MD, Alexandra Scherrer PhD and Yannick Vallet, MSc, from the SHCS Data Center (Lausanne, Switzerland), for data management; and Daniele Perraudin and Mirjam Minichiello, for administrative assistance.

Study Group Members


Funding

This work was supported by the Swiss National Science Foundation (grant 33C303_148522, grant 320030_159886 to H. F. G., and grant PZ00P3-142411 to R. K.); the Swiss HIV Cohort Study (SHCS projects 470, 528, 569, 683 and 759); the SHCS Research Foundation; the Yvonne-Jacob Foundation; Gilead, Switzerland (unrestricted grant to the SHCS Research Foundation); and the University of Zurich’s Clinical Research Priority Program (Viral infectious diseases: Zurich Primary HIV Infection Study; to H. F. G.). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Conflict of Interest: H. F. G. has been an adviser and/or consultant for GlaxoSmithKline, Abbott, Gilead, Novartis, Boehringer Ingelheim, Merck, Roche, Tibotec, Pfizer, and Bristol-Myers Squibb and has received unrestricted research and educational grants from Roche, Abbott, Bristol-Myers Squibb, Gilead, Astra-Zeneca, GlaxoSmithKline, and Merck Sharp and Dohme. S. Y. has been consultant for BMS and has received unrestricted research and educational grants from Roche, ViV, and Gilead. T. K. served as an advisor for Bristol-Myers Squibb and Pfizer and has received travel grants from Abbott and Pfizer. All other authors report no potential conflicts.

References


