Modeling Variability in Populations of Cells using Approximated Multivariate Distributions

Matthieu Pichené, Sucheendra K. Palaniappan, Eric Fabre, Blaise Genest

Modeling Variability in Populations of Cells using Approximated Multivariate Distributions

Matthieu Pichené, Sucheendra K. Palaniappan, Eric Fabre, and Blaise Genest

Abstract—We are interested in studying the evolution of large homogeneous populations of cells, where each cell is assumed to be composed of a group of biological players (species) whose dynamics is governed by a complex biological pathway, identical for all cells. Modeling the inherent variability of the species concentrations in different cells is crucial to understand the dynamics of the population. In this work, we focus on handling this variability by modeling each species by a random variable that evolves over time. This appealing approach runs into the curse of dimensionality since *exactly* representing a joint probability distribution involving a large set of random variables quickly becomes intractable as the number of variables grows. To make this approach amenable to biopathways, we explore different techniques to (i) *approximate* the exact joint distribution at a given time point, and (ii) to track its evolution as time elapses.

We start with the problem of *approximating* the probability distribution of biological species in a population of cells at some given time point. Data come from different *fine-grained* models of biological pathways of increasing complexities, such as (perturbed) Ordinary Differential Equations (ODEs). Classical approximations rely on the strong and unrealistic assumption that variables/species are independent, or that they can be grouped into small independent clusters. We propose instead to use the Chow-Liu tree representation, based on overlapping clusters of two variables, which better captures correlations between variables. Our experiments show that the proposed approximation scheme is more accurate than existing ones to model probability distributions deriving from biopathways.

Then we address the problem of tracking the *dynamics* of a population of cells, that is computing from an initial distribution the evolution of the (approximate) joint distribution of species over time, called the inference problem. We evaluate several approximate inference algorithms (e.g. [14], [17]) for *coarse-grained* abstractions [12], [16] of biological pathways. Using the Chow-Liu tree approximation, we develop a new inference algorithm which is very accurate according to the experiments we report, for a minimal computation overhead. Our implementation is available at https://codeocean.com/capsule/6491669/tree.

Index Terms—Biological pathways, population of cells, multivariate distributions.

1 Introduction

Multi-scale biological systems are challenging to model and analyze (see [10] for an overview). In this paper, we are interested in the dynamics of a population of tens of thousands of cells, where each cell is characterized by the concentration of biological species, and governed by the dynamics of a given biological pathway. In this context, capturing the variability among the cells w.r.t to the concentrations of species in the pathway is crucial to understand how the population will evolve [8]. For instance, consider the process of apoptosis [1] affecting a population of cells: Assume that half of the cells have high concentration of anti-apoptotic molecules and the other half have low concentration of anti-apoptotic molecules. Under an apoptotic drug, this population will not behave in the same way as a population where every cell is assumed to have an average concentration of anti-apoptotic molecules: in one case, almost all cells die (because average concentration of anti-apoptotic molecules is usually not enough to make the cell survive), while in the other case, around half of the cells survive. A naive approach to study such populations would be to track the concentrations of species in every single cell, but this obviously leads to extremely large models, and hence to very intensive computations.

- *Matthieu Pichené (mpichene@inria.fr) and Eric Fabre (eric.fabre@inria.fr) are with Univ Rennes, Inria, Team SUMO, Rennes, France*
- *Sucheendra K. Palaniappan (sucheendra@sbi.jp) is with The Systems Biology Institute, Tokyo, Japan*
- *Blaise Genest (bgenest@irisa.fr) is with Univ Rennes, CNRS, France*

In order to model a population of cells in a tractable manner, we propose to use multivariate probability distributions, where random variables represent the concentrations of species in the pathway, one variable per species. Handling such a joint probability distribution exactly is usually intractable, due to the curse of dimensionality: biological pathways usually have tens of variables. To make this approach tractable, we explore different techniques to *approximate* the original joint distribution by meaningful and tractable ones. The idea is to consider families of joint probability distributions on large sets of random variables that admit a compact representation, and then select within this family the one that best approximates the desired intractable one.

In this paper, we consider several *approximate representations* of multivariate distributions, and explore their interest in the particular context of populations of cells governed by a biological pathway. Our approximation scheme consists in imposing *conditional* independence properties to the multivariate distribution: the more conditional independence, the more compact is the representation of this joint distribution. For instance, in the extreme *fully factored (FF)* case where one assumes that all variables are unconditionally independent, the joint distribution is simply the product of marginals on every single variable. The key to our approach is thus to select the most relevant correlations between variables and assume the rest to be conditionally independent. In order to measure correlation strength, we use *Mutual information (MI)* which naturally measures correlations between pairs of vari-

ables. This orients us towards Information Theory to measure the approximation accuracy on probability distributions.

Classical approximations such as the fully factored approximation can be used to model the dynamics of a biopathway, but it leads to unsatisfactory results as it drops every correlation between species, in particular meaningful ones such as those that result from invariants of chemical reactions. Similarly, we can use *disjoint* clusters representations which partitions variables into clusters: within each cluster, variable correlations are preserved, but clusters are assumed independent. One can think of it as an FF approach by blocks.

In this paper, we go further and drop the assumption that clusters should be disjoint (and thus independent). Specifically, we consider clusters of variables organized into a tree-shape structure. In the case where clusters are pairs of variables, this tree is obtained by taking random variables as nodes, and placing an edge between any pair of variables that form a cluster. A tractable algorithm [5] allows one to compute the optimal approximation of any distribution by a tree of clusters of two variables, called the *tree-clustered approximation (TCA)*. The approximated joint distribution is fully determined by the marginals over each selected cluster of two variables, which offers a very compact encoding of this joint distribution (less than 800 values in our experiments).

The expression of the joint probability distribution of a TCA is more complex than classical approximations, because clusters are not disjoint. Nevertheless, we show that computations involving the TCA can be performed directly and efficiently over its compact representation. In particular, we show that any marginal over k out of n total variables can be computed with time complexity $O(n \cdot v^k)$, where v is the number of values each variable can assume.

We experiment on populations of cells modeled by several *fine-grained population* models of pathways. Population models are not so frequent, as usually a generic "average cell" model is used, assuming that every cell in the population follows this model. However, there are cases where the population does not behave as a copy of a cell. For instance, in the TRAIL-induced apoptosis pathway [1], some cells die and others do not: no average cell model could be used. In this work, we use three biological models of populations: we consider the hybrid Stochastic-Deterministic model [2] of the TRAIL-induced apoptosis pathway, and perturbed ODE models [12] of the EGF-NGF pathway [4] and of a simple catalytic reaction. In our experiments, TCA succeeds in capturing most correlations between pairs of variables involved in these biopathways, unlike with *disjoint* clusters.

Beyond modeling a population of cells at a given time point by the (approximate) distribution of species concentrations in cells, we also explore the interest of such approximate distributions for analyzing how a population will *evolve* when driven by a certain pathway. One solution would be to randomly generate tens of thousands of configurations according to the initial distribution, numerically integrate the ODEs from these configurations, compute the clusters of interest from the obtained configurations, and statistically compute the probabilities to be in each cluster configuration.

Instead, we use *coarse-grained* abstractions of the biological pathways [9], [12], [16], and more precisely *Dynamic Bayesian Networks (DBNs)*. Compared to ODEs that require a fine time scale, DBNs allow us to focus on relevant (subsampled) time points. In [16], it has been shown that simulating a DBN is much faster than simulating the fine-grained model it abstracts, for comparable prediction performances. DBNs are attractive since computing distributions reached at time t from an initial distribution at time 0 can be done efficiently through *inference* algorithms. While it is intractable to perform inference *exactly*, one can resort to *approximate* inference algorithms, such as Factored Frontier (FF[1]) [14] and Hybrid Factored Frontier (HFF) [17]. In short, FF represents the probability distribution in a fully factored form in order to perform very fast computations. HFF preserves a small number of joint probabilities of high value (called spikes), plus an FF representation of the remaining of the distribution. The more spikes, the more accurate the approximation, and the slower the HFF inference.

Our last contribution is an approximated inference algorithm, using TCA to represent the joint distribution at each time point. We developed a very efficient version of this algorithm, and we provide an error analysis for it. Over the biological pathways we considered, inference using TCA is very accurate, while HFF generates sizable errors, even with a considerable number of spikes (32k). Further, inference using TCA is faster than HFF, even with few spikes (3k). Both FF and disjoint-clusters are fast but quite inaccurate.

The paper is organized as follows. The next section recalls basic results about the Chow-Liu approximation of multivariate distributions. Section 4 introduces our model of dynamic Bayesian networks to model the evolution of species concentrations over time. It explains how the true multivariate distribution at each time step can be recursively approximated, in the Chow-Liu formalism. Section 5 presents the biological pathway considered in our experimental setting, and Section 6 reports the performances of our approach and compares them to alternate approaches previously proposed in the literature.

2 Representing Multivariate Distributions

In this section, we consider a joint probability distribution over a set $X = \{X_1, ..., X_N\}$ of N random variables (for instance concentrations of N molecules at some given time point). We assume that these variables can assume discrete values in the same set V (for instance, $V = \{\text{low,medium,high}\}$). In our case, the size $|V|$ of V will be small, typically around 5, while the size $N = |X|$ of X would be larger, typically around 30. We assume that continuous random variables (e.g. $V = \mathbb{R}^+$) are already discretized: there exist many quantization schemes to discretize continuous random variables with minimal distortion, for instance the Lloyd-Max algorithm (see [16] for a discussion).

Notation. Let $I = 1, 2, ..., N$ be the index set of the variables in X, and let $J \subseteq I$. We denote by X_J the tuple of variables $(X_i, i \in J)$, and with a slight abuse of notation we identify X with X_I. For partition $I = J \uplus K$, we also write $X = X_I = (X_J, X_K)$. We denote by $x = x_I = (x_i, i \in I)$ a tuple of values in V^I, that is a possible value for random vector X. Similarly, we denote by $x_J = (x_i, i \in J) \in V^J$ a possible value for X_J. Let P be a joint distribution on X.

1. Factored Frontier uses the Fully Factored approximation. Both have the same acronym FF

It is fully defined by the set of (joint) likelihoods $P(x)$ for $x \in V^I$. The marginal distribution over X_J is defined by the likelihoods $P(x_J) = \sum_{x_K \in V^K} P(x_J, x_K)$ for all values x_J assumed by the random variable X_J. We use notation $P(x)$ instead of $P(X = x)$ when it is unambiguous.

Encoding a probability distribution on X by the list of its joint likelihoods requires $|V|^N$ values. Throughout the paper, we use the distribution $P = \Delta_0$ for $X = \{X_1, X_2, X_3\}$ and $V = \{0, 1\}$, with the following joint likelihoods:

- $P(X_1 = 0, X_2 = 0, X_3 = 0) = 0.1$
- $P(X_1 = 0, X_2 = 0, X_3 = 1) = 0.03$
- $P(X_1 = 0, X_2 = 1, X_3 = 0) = 0.3$
- $P(X_1 = 0, X_2 = 1, X_3 = 1) = 0.17$
- $P(X_1 = 1, X_2 = 0, X_3 = 0) = 0.1$
- $P(X_1 = 1, X_2 = 0, X_3 = 1) = 0.1$
- $P(X_1 = 1, X_2 = 1, X_3 = 0) = 0.1$
- $P(X_1 = 1, X_2 = 1, X_3 = 1) = 0.1$

Clearly, the joint likelihoods representation quickly becomes intractable as the number of variables grows. We explore below approximations of such distributions that admit more compact representations.

2.1 Approximate Representations

There exist numerous approaches to approximate a probability distribution P over a large set of random variables $X = \{X_1, ..., X_N\}$. The first obvious one consists in assuming that all random variables are independent. In this family of distributions, the closest to the original P is simply the product of the marginal distributions of each single variable, also called the *fully factored* approximation P_{FF}:

$$P_{\text{FF}}(x) \quad = \quad \prod_{i=1}^{N} P(x_i) \tag{1}$$

P_{FF} is naturally represented by the list of individual likelihoods for each variable, resulting in $N \cdot |V|$ values to store. For our running example $P = \Delta_0$, this yields the representation:

- $P(X_1 = 0) = 0.6$
- $P(X_1 = 1) = 0.4$
- $P(X_2 = 0) = 0.33$
- $P(X_2 = 1) = 0.67$
- $P(X_3 = 0) = 0.6$
- $P(X_3 = 1) = 0.4$
- $P_{FF}(X_1 = X_2 = X_3 = 0) = 0.6 \cdot 0.33 \cdot 0.6 = 0.1188$

The main drawback of the P_{FF} approximation is to discard potentially important correlations between variables, such as those deriving from mass preservation principles in the chemical reactions of a biological pathway. To circumvent this difficulty, some authors have suggested to group correlated random variables into clusters that would be considered as a single random variable. This amounts to requesting the P_{FF} factorization by blocks. Formally, given a partition $I = K_1 \uplus ... \uplus K_c$ of the indices, one defines the disjoint clusters approximation as:

$$P_{\text{cluster}}(x) \quad = \quad \prod_{j=1}^{c} P(x_{K_j}) \tag{2}$$

If each cluster is of size m (with $c = \frac{N}{m}$), P_{cluster} needs $\frac{|X|}{m} \cdot |V|^m$ values to be fully defined. On our running example, assuming two clusters $K_1 = \{1, 2\}$ and $K_2 = \{3\}$, this yields the representation:

- $P(X_1 = 0, X_2 = 0) = 0.13$
- $P(X_1 = 0, X_2 = 1) = 0.47$
- $P(X_1 = 1, X_2 = 0) = 0.2$
- $P(X_1 = 1, X_2 = 1) = 0.2$
- $P(X_3 = 0) = 0.6$
- $P(X_3 = 1) = 0.4$

In general, however, any pair of random variables (species) involved in a biological pathway exhibits meaningful correlation (see the experimental work in Section 6). The independence between variables or between clusters of variables imposed by the two approximations above is thus inappropriate, as it discards some important information. We propose an alternate approach that relies on *non-disjoint clusters* $I = K_1 \cup ... \cup K_c$, which will better approximate the correlations between species. This approximation requires $c \cdot |V|^m$ values, that is $|V|^m$ values for each of the c clusters of m variables. Notice that the number c of clusters will now be larger than $\frac{N}{m}$ as clusters are non-disjoint. For our running example again, with clusters $K_1 = \{1, 2\}$ and $K_2 = \{2, 3\}$, this yields the list

- $P(X_1 = 0, X_2 = 0) = 0.13$
- $P(X_1 = 0, X_2 = 1) = 0.47$
- $P(X_1 = 1, X_2 = 0) = 0.2$
- $P(X_1 = 1, X_2 = 1) = 0.2$
- $P(X_2 = 0, X_3 = 0) = 0.2$
- $P(X_2 = 0, X_3 = 1) = 0.13$
- $P(X_2 = 1, X_3 = 0) = 0.4$
- $P(X_2 = 1, X_3 = 1) = 0.27$

In such a setting, the closed-form expression of the approximate distribution P_{NDC} (for non-disjoint clusters) differs much from (2), as some variables appear in several clusters, but must only be accounted once for their contribution. For our running example, as variable X_2 appears in two clusters, one has:

$$P_{\text{NDC}}(x) \quad = \quad \frac{P(x_1, x_2)\, P(x_2, x_3)}{P(x_2)} \tag{3}$$

or equivalently using conditional probabilities:

$$P_{\text{NDC}}(x) \quad = \quad P(x_1, x_2) P(x_3 | x_2) \tag{4}$$

This amounts to requiring that X_1 and X_3 are conditionally independent given X_2, but it does not cancel out the correlation between X_1 and X_3. We show below how this expression generalizes.

2.2 Cluster Tree Distributions

The distributions we consider in this paper rely on a choice of (non-disjoint) clusters $I = K_1 \cup ... \cup K_c$ that cover the index set I of random variables $(X_i, i \in I)$. It is further required that these clusters organize into a tree. For simplicity, from now on we will only consider clusters of size at most 2. An overview of the general case can be found in Appendix 1.

Formally, let $T = (V, E)$ be the undirected graph with:

- $V = I$, the set of indices of variables, and
- $E = \{K_1, ..., K_c\}$, the set of clusters of size at most 2,

where each cluster K_i is a pair $\{u, v\} \subseteq I$. We require that T is a tree: there is no cycle, that is no sequence $i_1, ..., i_k \in I$ with $k \geq 3$ s.t. $\forall j \leq k$, $\{i_j, i_{j+1}\} \in E$ and $\{i_k, i_1\} \in E$.

For such trees, the approximated probability distribution, denoted P_{NDC} (for non-disjoint clusters), is the following:

$$P_{\text{NDC}}(x) \quad = \quad \frac{\prod_{j=1}^{c} P(x_{K_j})}{\prod_{i=1}^{N} P(x_i)^{C_i - 1}} \tag{5}$$

where C_i is defined as the number of clusters K_j containing index i: Each variable X_i separates the tree T into C_i branches which are conditionally independent given X_i for distribution P_{NDC}. This expression directly generalizes (3). Also, it is easy to check that this expression defines a probability distributions, by considering an asymmetric formula using conditional probability, generalizing (4), which considers edges of the tree in postorder.

Another essential observation is that P_{NDC} coincides with P on each cluster K_j. Specifically, one has $P_{\text{NDC}}(X_{K_j}) = P(X_{K_j})$, for all cluster K_j. This is clearly visible in (4): one gets $P_{\text{NDC}}(x_1, x_2) = P(x_1, x_2)$ by marginalizing out x_3.

2.3 Obtaining Optimal Clusters

A central issue in building clustered approximations of P (disjoint or not) is to select optimal clusters, that is clusters for which P_{NDC} will best approximate the real joint probability distribution P. In this paper, we use the simple Chow-Liu algorithm [5] given in Algorithm 1 to select non disjoint clusters of size 2 forming a tree. Put informally, it quantifies the strength of correlations using Mutual Information (MI) between two variables X_1, X_2, defined as

$$\text{MI}(X_1, X_2) \quad = \quad \sum_{x_1, x_2} P(x_1, x_2) \log \frac{P(x_1, x_2)}{P(x_1)P(x_2)}$$

Intuitively, it selects the most strongly correlated pairs of variables first, and drops a pair when it would create a cycle. For our running example, one has $\text{MI}(X_1, X_2) \approx 0.0625$, $\text{MI}(X_2, X_3) \approx 5.44 * 10^{-5}$ and $\text{MI}(X_1, X_3) \approx 0.02$. The Chow-Liu procedure would thus select edges $K_1 = \{1, 2\}$ and $K_2 = \{1, 3\}$, creating a tree with root 1 and two children 2 and 3. The direct correlation between X_2 and X_3 is thus dropped, as it would create cycle $(1, 2, 3, 1)$. Notice however that an indirect correlation between X_2 and X_3 does exist, through variable X_1. Indeed, variables X_2 and X_3 are not independent for P_{NDC}: $P_{\text{NDC}}(X_2, X_3) = \sum_{x_1} P(X_2|x_1)P(X_3|x_1)P(x_1) \neq P(X_2)P(X_3)$. We will actually discuss in detail the efficient derivation of all such pairwise marginals $P_{\text{NDC}}(X_2, X_3)$ in Section 3.

One may wonder how good the choice of non-disjoint clusters given by the Chow-Liu algorithm is. In Section 6, we will see that it provides twice as much information as disjoint clusters on several biological pathways. Theoretically, one can show that this choice is optimal over trees of non-disjoint clusters of size 2, when considering the Kullback-Leibler divergence of two distributions P, Q, defined as:

$$KL(P, Q) \quad = \quad \sum_{x \in V^X} P(x) \log \frac{P(x)}{Q(x)} \tag{6}$$

$KL(P, Q)$ is always positive and vanishes iff $Q = P$.

Proposition 1. Let P^T_{NDC} the probability distribution derived by (5) and associated with clusters from a tree $T = (I, E)$. Let T be the Chow-Liu tree. It satisfies:

$$KL(P, P^T_{\text{NDC}}) \quad = \quad \min_{T'} \; KL(P, P^{T'}_{\text{NDC}}) \tag{7}$$

Proof: Given criterion (6), and for clusters $I = K_1 \cup ... \cup K_c$ defining a cluster tree, one can readily observe that the optimal Q must satisfy $Q(x_{K_i}) = P(x_{K_i})$ for every value x_{K_i} and any cluster K_i. Which corresponds to the implicit choice made in all approximations of Section 2.1. Moreover, one has:

$$KL(P, P_{\text{FF}}) \quad = \quad KL(P, P^{T'}_{\text{NDC}}) + KL(P^{T'}_{\text{NDC}}, P_{\text{FF}}) \tag{8}$$

The proof of the first point can be found in [5]. The second point derives from [6]. Observe that it holds for all cluster tree approximations of P, which proves that $P_{\text{NDC}^{T'}}$ is always a better approximation of P than P_{FF}. □

Notice that it is possible to define P_{NDC} for structures more complex than trees (e.g. triangulated graphs with clusters of size 3, see Appendix 1). Starting from the Chow-Liu tree and adding edges can only improve the resulting approximation of P. However, there is no simple procedure that would give the optimal triangulated graph: this problem was proved to be NP-complete. Nevertheless, generalization of the Chow-Liu algorithm can perform well [7].

Algorithm 1: Chow-Liu Algorithm.
Computes an optimal tree of clusters.

For each pair $\{X_i, X_j\}$ in X, compute $MI(X_i, X_j)$.
Sort edges $\{i, j\}$ by decreasing value of $MI(X_i, X_j)$.
Starting with an empty graph as tree T, repeat:
- consider the next edge $\{i, j\}$ in the list
- if $\{i, j\}$ does not close a cycle in T, add it to T

3 Handling P_{NDC} with low complexity

Cluster-tree distributions enjoy a compact encoding as they are fully determined by their marginals on clusters $(K_i)_{1 \leq i \leq c}$ (see Sec. 2.2). However, to fully benefit from the closed-form expression (5) and reduce the complexity of standard computations, on needs carefully designed algorithms.

In this section, we examine the derivation of the marginal $P_{\text{NDC}}(X_J)$ on a subset of variables X_J, for $J \subseteq I$. This operation will be instrumental in the sequel. If each variable X_j can assume $|V|$ values, there are $|V|^{|J|}$ values to compute. However, brute forcing computing them using the full distribution $P_{\text{NDC}}(X)$ over all variables would results to $|V|^{|I|}$ intermediate computations, which is not affordable in practice and would kill any interest for (5). We show here that a careful use of (5) can actually avoid this complexity explosion.

Let P be a cluster tree distribution on $X = (X_i, i \in I)$, where clusters have size 2. In other words, the graph $T = (I, E)$ associated to these clusters is a tree, and P is fully determined by its marginals $P(X_i, X_j)$ on the edges $\{i, j\}$ of this tree, see (5).

Consider a partition $I = J \uplus K$ of indices. We are interested in computing the marginalization of P on X_J, that is

$$P(x_J) = \sum_{x_K} P(x_J, x_K)$$

for all values $x_J \in V^{|J|}$ of X_J. Computing the $P(x)$ for all x and deriving $P(X_J)$ as a marginal is clearly not an option. One should rather perform clever marginalization in the product form (5). This takes the form of a classical message passing algorithm along the edges of T, that we describe below.

Let r be a node of J, that we will fix as the root of tree T. From there, one can easily define a parent relation (r has no parent, and every other node i has exactly one parent j with $\{i, j\}$ a cluster). For a node i, we denote by $S(i)$ its set of children. Leaves are nodes i such that $S(i) = \emptyset$. We also denote by $D(i)$ the set of descendants of node i, that is the set of children of i union the children of children of i, etc. We will proceed in a bottom-up fashion on the tree.

We define the *message* from i to j on edge $\{i, j\}$ as

$$M_{i,j}(X_j) = P(X_{J\cap D(i)}|X_j) \tag{9}$$

As a conditional distribution, this is a function of $(X_{J\cap D(i)}, X_j)$, but we only materialize it as a function of X_j for simplicity of notations.

We now define a bottom-up message passing algorithm. It is initiated on the leaves of tree T and progresses towards r, thanks to the following forwarding rule. Let $i \in I$ be a vertex in T that has received a message from each of its children in $S(i)$. Let j be the parent of i. If $i \in J$, then

$$M_{i,j}(X_j) \quad = \quad P(X_i|X_j) \cdot \prod_{k\in S(i)} M_{k,i}(X_i) \tag{10}$$

otherwise

$$M_{i,j}(X_j) \quad = \quad \sum_{x_i} P(x_i|X_j) \cdot \prod_{k\in S(i)} M_{k,i}(x_i) \tag{11}$$

In other words, if $i \in J$, X_i is preserved as one of the free variables in the message, otherwise X_i is marginalized out. In both cases, the operation introduces the new variable X_j. These expressions rely on the fact that variables sitting in different subtrees around i are conditionally independent given X_i for distribution P. Observe also that the message propagation rule (10,11) only requires $P(X_i, X_j)$, i.e. it operates on the compact representation (5) of P by its marginals on clusters (edges of T).

The algorithm terminates when the root r has received messages from all its children, thanks to the following merge rule:

$$P(X_J) \quad = \quad P(X_r) \cdot \prod_{k\in S(r)} M_{k,r}(X_r) \tag{12}$$

As all nodes in I must be visited once and messages have size bounded by $|V|^{|J|}$ (as $r \in J$), this yields the following result.

Proposition 2. Let $P(X_I)$ be a cluster tree distribution where all clusters have size 2. Let $J \subseteq I$. One can compute the marginal distribution $P(X_J)$, i.e. $|V|^{|J|}$ values, in time $O(|I| \cdot |V|^{|J|})$.

4 Inference for Stochastic Discrete Abstractions of Biological Pathways

This section describes a specific stochastic abstraction of a biological pathway, under the form of a *Dynamic Bayesian Network (DBN)* which can model the dynamics of biological species along time (see next section). We then demonstrate how the approximate representation of distributions developed in Section 2 can be used to perform approximate inference on these DBN models. Comparing the results of the different approximations for inference is a good way to quantify how accurate these approximations are for conveying important information about the population over time, beyond raw mutual information numbers.

4.1 Dynamic Bayesian Networks

Our objective is to model the evolution of variables $(X_i, i \in I)$ along time, for time ranging over discrete values $\{0, 1, ..., T\}$. We denote by $X^t = (X_i^t, i \in I)$ the (vector of) random variables at time t.

$$P(X^t|X^0 \dots X^{t-1}) \quad = \quad P(X^t|X^{t-1}) \tag{13}$$

The transition probability is then requested to factorize as

$$P(X^t|X^{t-1}) \quad = \quad \prod_{i\in I} P(X_i^t|X^{t-1}) \tag{14}$$

which captures the fact that the variables $(X_i^t, i \in I)$ at time t are independent given X^{t-1}. Notice that these individual transition probabilities $P(X_i^t|X^{t-1})$ may depend on time t.

Terms $P(X_i^t|X^{t-1})$ are still involving too many variables ($|I| + 1$) for practical computations ($|V|^{|I|+1}$ entries). To mitigate that, DBNs further require that each X_i^t only depends on a small subset of variables $X_{\hat{\imath}}^{t-1}$ from time point $t - 1$, that is:

$$P(X_i^t|X^{t-1}) \quad = \quad P(X_i^t|X_{\hat{\imath}}^{t-1}) \tag{15}$$

The index set $\hat{\imath} \subseteq I$ is called the *parent set* of each index i. By extension, $X_{\hat{\imath}}^{t-1} = (X_j^{t-1}, j \in \hat{\imath})$ are the *parents* of X_i^t. Notation $\hat{\imath}$ in (15) implicitly captures the assumption that parent sets do not depend on time t. In the sequel, we either use notation $P(X_i^t|X_{\hat{\imath}}^{t-1})$ or $P^t(X_i|X_{\hat{\imath}})$ for (time-varying) transition probabilities. In practice, values $P^t(x_i|x_{\hat{\imath}})$ are stored in so-called Conditional Probability Tables (CPT).

The initial distribution $P^0(X) = P(X^0)$ assume independent variables, i.e.

$$P^0(X) \quad = \quad \prod_{i\in I} P^0(X_i) \tag{16}$$

The distribution $P^t(X) = P(X^t)$ of variables at time t then satisfies the following recursion:

$$P^t(X) \quad = \quad \sum_x P^{t-1}(x) \cdot \prod_{i\in I} P^t(X_i|x_{\hat{\imath}}) \tag{17}$$

The *inference problem* consists in computing marginal distributions $P(X_i^t) = P^t(X_i)$ for each variable index $i \in I$ and each time instant $1 \le t \le T$. The main difficulty is that the independence assumed at time $t = 0$ in (16) no longer holds at successive time instants. Even for small parent sets in (15), correlations propagate in space and variables in X^t quickly become all correlated to one another. Consequently, recursion (17) can not be performed exactly, and one needs to replace all $P^{t-1}(X)$ by approximations. While the *Factored Frontier* (FF) method imposes factorization (16) to hold at any time step, we rather elaborate on the results of Section 2 to better preserve meaningful correlations.

4.2 A Generic Approximate Inference Algorithm

The inference problem relies on recursion (17), which requires an integration over values of X^{t-1}. For complexity reasons, this integral can not be directly performed on any distribution P^{t-1}, just like P^t can not be fully determined. We therefore replace each P^t by an approximation B^t, called the *belief state* at time t.

Let B be a general probability distribution on X. We denote by $App(B)$ an approximation of B under a simpler and more manageable form. The *Factored Frontier (FF)* algorithm [14] makes the choice $App(B) = B_{\text{FF}}$, where B_{FF} is fully determined by the $B(X_i)$, $i \in I$. In this paper, we refine this approach using $App(B) = B_{\text{NDC}}$ assuming a cluster tree defined by clusters $K_1 \cup ... \cup K_c = I$; so B_{NDC} is fully determined by the $B(X_{K_i})$, $1 \leq i \leq c$, see (5). Recall that these clusters are pairs $\{i, j\}$ and form a tree $G^t = (I, E^t)$, so $c = |I| - 1$. The tree G^t is computed off-line before the inference algorithm, by using the Chow-Liu algorithm from data obtained from some simulations of the DBN over time using [15]. Observe that the tree $G^t = (I, E^t)$ may change at each time point t. Each B^t is fully determined by the marginals $B^t(X_i, X_j)$ for $\{i, j\} \in E^t$, see Sec. 2.2.

Recursion (17) then becomes

$$Q^t(X) = \sum_x B^{t-1}(x) \cdot \prod_{i \in I} P^t(X_i|x_{\hat{\imath}}) \quad (18)$$

$$B^t(X) = App(Q^t(X)) \quad (19)$$

These relations can now be further simplified thanks to the structural properties of the B^t. First, observe that B^t defined by (19) is fully determined by the $B^t(X_i, X_j) = Q^t(X_i, X_j)$ for $\{i, j\} \in E^t$, therefore (18) should actually aim at deriving these pairwise marginals instead of the full $Q^t(X)$. Further, by specializing (18) one gets:

$$\begin{aligned} & B^t(X_i, X_j) \\ = & \sum_x B^{t-1}(x) \cdot P^t(X_i|x_{\hat{\imath}}) \cdot P^t(X_j|x_{\hat{\jmath}}) \\ = & \sum_{x_{\hat{\imath}\cup\hat{\jmath}}, x_J} B^{t-1}(x_{\hat{\imath}\cup\hat{\jmath}}, x_J) \cdot P^t(X_i|x_{\hat{\imath}}) \cdot P^t(X_j|x_{\hat{\jmath}}) \\ = & \sum_{x_{\hat{\imath}\cup\hat{\jmath}}} B^{t-1}(x_{\hat{\imath}\cup\hat{\jmath}}) \cdot P^t(X_i|x_{\hat{\imath}}) \cdot P^t(X_j|x_{\hat{\jmath}}) \quad (20) \end{aligned}$$

where $J = I \setminus (\hat{\imath} \cup \hat{\jmath})$, therefore the integration space reduces to the values of parents of X_i and X_j. Finally, the marginal distribution $B^{t-1}(X_{\hat{\imath}\cup\hat{\jmath}})$ required by (20) can be derived with reasonable complexity thanks to Prop. 2.

All these operations are summarized in Algorithm 2, that approximately solves the inference problem.

Theorem 1. For App the approximation based on successive trees $G^t = (I, E^t)$, Algo. 2 inductively computes B^T from B^0 in time $O(T \cdot |I| \cdot |V|^p \cdot (|I| + |V|^2))$, where $\ell = max_{t,\{i,j\}\in E^t} |\hat{\imath} \cup \hat{\jmath}|$.

Proof: The maximum in the definition of ℓ is taken over all clusters $\{i, j\}$ that appear in Algo. 2, therefore over edges of all trees $G^t = (I, E^t)$. The correctness of Algo. 2 comes directly from (20). The complexity follows from:

- Factor T comes from the induction over T time steps.
- The number of clusters (pairs of variables) at each time step is upper bounded by $|I|$, which gives the first $|I|$.

Algorithm 2: Clustered Factored Frontier (CFF)

Input: Trees $G^t = (I, E^t)$, for each time point $t \leq T$
Input: Parents $X_{\hat{\imath}}$ for each variable X_i, $i \in I$
Input: Local transition probabilities $P^t(X_i|X_{\hat{\imath}})_{i,t}$
Input: Initial distributions $P^0(X_i, X_j)$ for $\{i, j\} \in E^0$
Init : $B^0(X_i, X_j) = P^0(X_i, X_j)$ for $\{i, j\} \in E^0$
for $t \in [1..T]$ **do**
  for $\{i, j\} \in E^t$ **do**
    compute $B^{t-1}(X_{\hat{\imath}\cup\hat{\jmath}})$ by the message passing algorithm on G^{t-1} of Section 3,
    set $B^t(X_i, X_j) = \sum_{x_{\hat{\imath}\cup\hat{\jmath}}} B^{t-1}(x_{\hat{\imath}\cup\hat{\jmath}}) \cdot P^t(X_i|x_{\hat{\imath}}) \cdot P^t(X_j|x_{\hat{\jmath}})$

- For each cluster $\{i, j\}$, (20) must compute $|V|^2$ values $B^t(x_i, x_j)$, and each one requires an integration over at most $|V|^\ell$ values $B^{t-1}(x_{\hat{\imath}\cup\hat{\jmath}})$, whence the complexity $|V|^{\ell+2}$. Then the computation of $B^{t-1}(X_{\hat{\imath}\cup\hat{\jmath}})$ has complexity upper bounded by $|V|^\ell \cdot |I|$, thanks to Prop. 2.

The overal cost for cluster $\{i, j\}$ is thus $|V|^\ell(|V|^2 + |I|)$. □

We show in the supplementary material (Appendix 2) how to optimize the algorithm and obtain a complexity exponential in $\max(|\hat{\imath}|, |\hat{\jmath}|)$ instead of exponyential in $|\hat{\imath} \cup \hat{\jmath}|$ for Theorem 1.

Practical considerations

In practice, the transition probabilities $P^t(X_i|X_{\hat{\imath}})$ are matrices of dimension $|V| \times |V|^{|\hat{\imath}|}$. They derive from data produced by fine grain models, and thus may exhibit singularities, such as zero rows in the case where the conditioning value $x_{\hat{\imath}}$ was never observed in the data (see [15] for a discussion). In addition, an expression like

$$B^t(X_i) = \sum_{x_i} P^t(X_i|x_{\hat{\imath}}) B^{t-1}(x_{\hat{\imath}}) \quad (21)$$

and *a fortiori* (20) involves a large number of small values, and may thus suffer from rounding errors. We thus introduce a renormalization in (21) to ensure that it yields a proper probability distribution on variable X_i, which takes the form

$$\tilde{B}^t(X_i) \propto \sum_{x_i} P^t(X_i|x_{\hat{\imath}}) B^{t-1}(x_{\hat{\imath}}) \quad (22)$$

In the same way, the central relation of Algo. 2

$$B^t(X_i, X_j) = \sum_{x_{\hat{\imath}\cup\hat{\jmath}}} B^{t-1}(x_{\hat{\imath}\cup\hat{\jmath}}) \cdot P^t(X_i|x_{\hat{\imath}}) \cdot P^t(X_j|x_{\hat{\jmath}}) \quad (23)$$

may not sum to one and furthermore, when marginalizing out X_j in $B^t(X_i, X_j)$ and X_k in $B^t(X_i, X_k)$, one may not get the same marginal $B^t(X_i)$. We therefore rely on (22) (renormalized) to compute the expected marginals $\tilde{B}^t(X_i)$ and $\tilde{B}^t(X_j)$ at time t, and then require from the term $B^t(X_i, X_j)$ to satisfy both of these marginals. This is performed by the standard Iterative Proportional Fitting Procedure (IPFP). In our experiments, convergence (up to numerical noise) took place in 5 to 6 iterations of IPFP.

5 Populations governed by Biopathways

In this section, we present the pathways on which we will perform our experiments, both for representing the distribution (see Section 2) and for tracking the dynamics of the system (see Section 4). We consider two types of population models: We use one mathematical model [2] specifically developed for populations of biological cells, involving the generation of native proteins, which thus gives different results on different cells because of stochasticity of gene activation. We also use perturbed ODE models from [12], using slightly different reaction speeds and initial concentrations in each cell, around nominal values. We start by explaining how to obtain a DBN abstraction from a population model, independent on its exact formalization.

5.1 DBN Models as Abstractions of Biological Systems

Liu et al. developed a DBNs abstraction method [11], [12] that we extended in [16], from models describing biopathway dynamics. Its main features are illustrated by a simple enzyme kinetics system shown in Fig. 1.

We consider a generic model for the dynamics of a pathway using a system of equations $x^{t+dt} = f(x^t)$ (e.g. ODEs), where f can exhibit stochastic behaviors to model randomness. Usually, the concentration x_i^{t+dt} of molecular species i at time $t+1$ only depends on few other molecular species at time t, the ones producing or reacting directly with i. These molecular species are called the parents $\hat{\imath}$ of i, and we have $x_i^{t+dt} = f_i(x_{\hat{\imath}}^t)$. Different cells of the population can use (slightly) different functions f to model (slightly) different behaviors (e.g. perturbed ODEs), but the parent relation is assumed to be fixed over the population of cells.

The dynamics of the system is assumed to be of interest only for discrete time points up to a maximal time point. Let us assume that time points are denoted as $\{0, 1, \ldots, T\}$. There is random variable X_i corresponding to the concentration of every molecular species. The range of each variable X_i is quantized into a set of intervals, with $|V|$ the number of intervals (discretized values) for variable X_i (typically $|V| = 5$). The quantized dynamics is intrinsically stochastic, as even for deterministic dynamics (e.g. of an ODE system), it is possible that two distinct deterministic configurations correspond to the same quantized configuration, but their successors are in distinct quantized configurations.

Initial concentrations of the population follow a distribution, to model the variability of cells in the population - usually in a product form if the native species evolve independently before the pathway starts kicking in. The distribution is obtained by sampling initial configurations of the population models.

The parent relations of the DBN can be obtained in different ways. In [11], [12], the parents of a species are simply the variables producing or reacting directly with it. In [16], information theoretic tools can be used to automatically infer a more adequate set of parents.

CPT entries are evaluated through classical enumeration over many simulated trajectories of the model. To simulate a trajectory, an initial configuration is sampled from the distribution of initial concentrations, and function f describing the model dynamics is applied iteratively to cover all time points. For instance, among the generated trajectories, the number of simulations where the value of Y_j falls in the interval u_j at time $t-1$ for each $j \in \hat{\imath}$ is recorded, say $J_{\vec{u}}$. Next, among these $J_{\vec{u}}$ trajectories, the number of these where the value of Y_i falls in the interval x at time t is recorded. If this number is J_x then the empirical probability p is set to be $\frac{J_x}{J_{\vec{u}}}$. We refer interested readers to Liu et al.'s work [11], [12] for the details.

5.2 Hybrid Stochastic Deterministic Model

To model populations, we will first consider the Hybrid Stochastic Deterministic (HSD) model of [2] specifically developed to model populations of biological cells governed by the apoptosis pathway. This HSD model matches experimental data on populations of cells from [1]. It involves the generation of native proteins, and thus gives different results on different cells because of stochasticity of gene activation.

More precisely, the apoptosis pathway it models is the one triggered by TNF-related apoptosis-inducing ligand (TRAIL) for HeLa cells. TRAIL is an apoptosis inducing protein in cancer cells, considered as a target for anti-cancer therapeutic strategies. Biological observations on HeLa cells suggest that in a population of cells, TRAIL application only leads to fractional killing of cells. Further, there is a time dependent evolution of cell resistance to TRAIL. These phenomena are modeled in the HSD model of [2], with a stochastic part linked to gene activation triggering the production of native proteins, and a deterministic part describing the effect of TRAIL on the cells. In particular, whether apoptosis is triggered or not, depending on the initial concentrations of pro- and anti-apoptotic proteins and their productions in

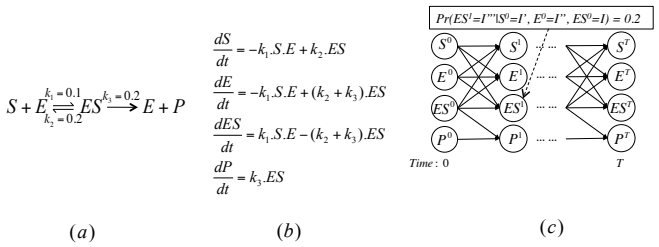


Fig. 1. a) The enzyme catalytic reaction network (b) The (perturbed) ODEs model (c) The DBN approximation for 2 successive time points, that reproduces the dependencies within the ODEs.

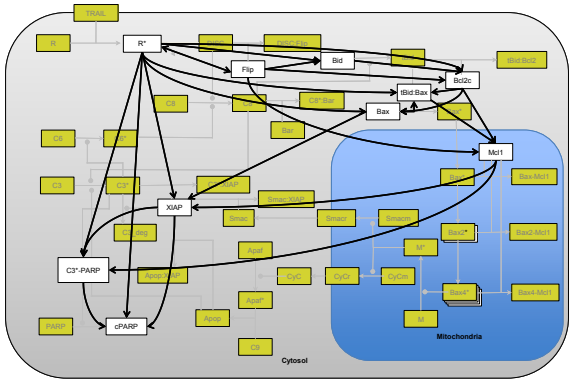


Fig. 2. Apoptosis pathway

the early stage of the pathway. This model explains well experimental data from [1] both on the de-correlation with time of the death of sister cells and on the raw percentage of cells of the population dying after several repeated treatment.

Out of the hundred variables of the system, we focus on the abstraction from [16], consisting of the 10 most important protein variables (see Figure 4). To produce the initial distribution modeling the variability between cells, the stochastic model explaining the production of native proteins is executed for few hours [2]. A DBN is built on a 10 variable system, with 4 parents per variable. The time horizon is the first 90 minutes period after injection of TRAIL, which was divided into 22 time points. Both stochastic and deterministic part are simulated from initial configurations, generating 10^5 trajectories used to fill up the CPTs entries.

5.3 Perturbed ODE models:

We test our algorithms with two other pathways of varying size: one smaller, a simple enzyme catalytic pathway with 4 variables, and one bigger, the EGF-NGF pathway with 52 variables. In order to obtain models of populations, we use the perturbed ODE method of [11], [12]: each cell is associated with the same ODE model, but with slightly different reaction speeds and initial configurations, taken randomly in a small interval around nominal values from the BioModels database.

Enzyme Catalysis: The simple enzyme catalytic system is shown in Fig. 1 (a). It describes a typical mass action based kinetics of the binding (ES) of enzyme (E) with substrate (S) and its subsequent catalysis to form the product (P). The value space of each species (variable) is divided into 5 equal intervals. The time scale of the system is 10 minutes which was divided into 100 time points. The parents relations for the DBN are obtained using [11], [12]. Conditional probability tables were populated by drawing 10^5 simulations from the underlying ODE model.

EGF-NGF Pathway: The EGF-NGF pathway describes the behavior of cells to EGF or NGF stimulation [4]. The ODE model of this pathway is available in the BioModels database and consists of 32 differential equations. The value domains of the 32 variables were divided into 5 equal intervals. The time horizon of each model was assumed to be 10 minutes which was divided into 100 time points. The parents relations for the DBN are obtained using [11], [12]. To fill up the CPTs, 10^5 ODE trajectories were generated.

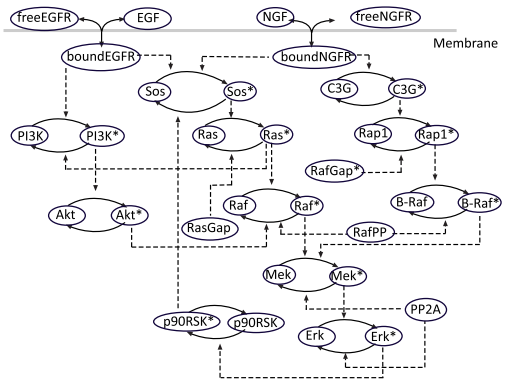


Fig. 3. EGF-NGF pathway

6 Experimental results

We developed implementations of all algorithms in Python, freely available at https://codeocean.com/capsule/6491669/tree. All experiments were performed on an Intel i7-4980HQ processor (2,8 GHz quad core Haswell with SMT) with 16 GB of memory. For each of the pathway case study discussed in the previous section, we consider the following:

- the exact and approximated probability distributions at a arbitrarily chosen time point. As one cannot compute the exact joint probability for large systems, we evaluate them considering the mutual information between any pair of variables (computed from 10.000 simulations of the system), to understand where correlations are lost. We use Prop. 2 to compute MI values for the Tree Cluster representation. Results can be found in Fig. 6.
- the approximated inference algorithm, compared with statistical simulations of the DBNs using the algorithm from [15]. Results can be found in Fig. 7. We report mean error over marginals normalized to FF (with $FF = 100\%$), as the raw numbers are not meaningful - most marginals being irrelevant and thus diluting the raw error tremendously).

We explain these numbers in more detail in the following:

6.1 How correlated are species in biopathways?

We first discuss the correlations between species in populations governed by biopathways. The question is to evaluate these correlations, using MI to quantify them.

First, consider the bottom left diagram (labeled "Real") in Figure 5. Around 40 species (the ones on the top right part) show some kind of correlations with one another after 5 minutes of the start of the pathway.

More generally, over the 3 pathways we consider, we also find correlations between many species (Figure 6 a),b),c)). Consider the line "Exact" and column "mean MI": the average MI between two species is relevant. Notice that the mean MI number decreases with the number of species (0.278 with 4 species, 0.12 with 10 species, 0.026 with 52 species), which is to be expected as species far away in a pathway are not as correlated as species next to each other. One can also consider line "FF" and column "max MI error" to see the maximal correlations between two different species, which is very large in all 3 cases: 0.27 for 4 species, 0.32 for 10 species, 0.6 for 52 species. Indeed, "FF" assume that the species are not correlated, hence the MI error of "FF" wrt "Exact" is exactly this maximal correlation. Considering line "Disjoint Cluster" and column "max MI error", we can also understand that there are chains of correlations which cannot be broken in disjoint clusters, with correlations of $MI = 0.11, , 0.2, 0.2$ lost in the 3 pathways respectively.

6.2 Do these correlation matter?

While the correlations have tangible MI numbers, those numbers are not always very large. One important question is then how these raw MI numbers translates in practice.

To understand that, let us consider the three different inference approximation techniques, which only differ in the correlations between species they take into account. We consider the effect fogetting some correlations have on the

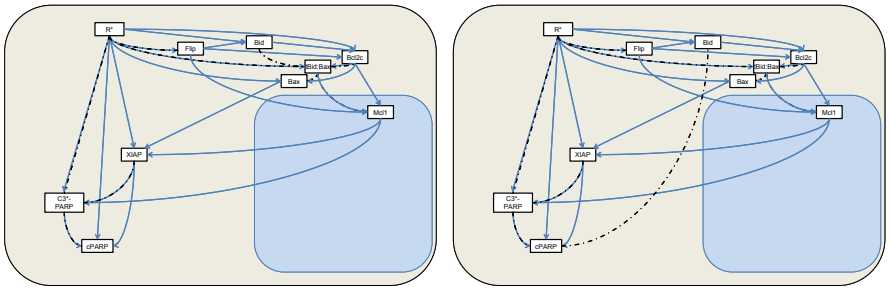


Fig. 4. Trees build from the abstracted Apoptosis pathway. At 20 minutes on the left, and 90 minutes on the right.

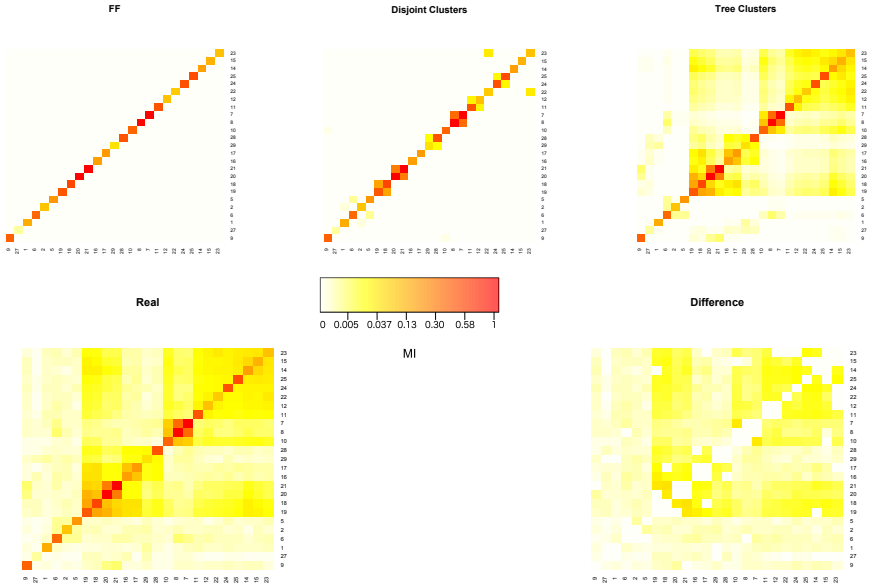


Fig. 5. Comparison of Mutual Information between the 3 different probability distribution approximations at minute 5 (top). The bottom 2 diagrams show the exact mutual information and the difference between exact and the approximations using clusters associated with the tree.

(a) Enzyme catalytic reaction, probability distribution at 2 minutes:

Representation	Mean MI	max MI Error	Max P error	KL diverg.
FF	0.22	0.27	0.22	0.31
Disjoint Cluster	0.26	0.11	0.05	0.12
Tree Cluster	0.277	0.04	0.005	0.001
Exact	0.278	0	0	0

(b) Apoptosis pathway, probability distribution at 30 minutes:

Representation	mean MI	max MI Error	Size of representation
FF	0.06	0.32	50
Disjoint Cluster	0.08	0.2	125
Tree Cluster	0.1	0.12	225
Exact	0.12	0	10^7

(c) EGF-NGF pathway, probability distribution at 5 minutes:

Representation	mean MI	max MI Error	Size of representation
FF	0.016	0.6	160
Disjoint Cluster	0.019	0.2	400
Tree Cluster	0.023	0.07	775
Exact	0.026	0	10^{22}

Fig. 6. Tables representing the error of the approximations w.r.t. the real distribution for various pathways.

concentrations of each molecules (effect on joint distribution is expected to be much worse - but we would evaluate it using MI again, which we put in question here).

Figure 8 is the most visual example: on the left, we display the evolution of species "cleaved PARP" (marker of cell death) in the apoptosis pathway, while on the right, we display the evolution of protein ErkAct in the EGF-NGF pathway, which shows the most obvious mistakes due to forgetting correlations. We can see that the evolutions of both species is quantitatively altered when forgetting all correlations (FF) or most correlations (Disjoint Clusters).

More generally over all species, we can consider Figure 7 a),b),c), line "FF" and column "Nb. Error > 0.1", which shows that there are many time points and species where forgetting the correlations between species eventually leads to a tangible error in the concentrations of the species.

We now discuss results for each of the example pathways:

6.3 Enzyme Catalysis

The system is very simple with only 4 variables. The tree obtained using the Chow-Liu algorithm is the same over all time points, with $\{\{E, S\}, \{E, P\}, \{E, ES\}\}$ as set of non-disjoint clusters. To compare with a disjoint cluster representation, we chose the set of disjoint clusters with highest mutual information, that is $\{\{E, S\}, \{ES, P\}\}$. On this example, in addition to computing the largest difference in MI, we provide the maximum difference of the probability of joints and the Kullback-Leibler divergence as the system is small enough to compute them.

Fig.6 (a) shows the measures at an arbitrarily chosen time point (corresponding to 2 minutes) of the system. It can be seen that our Tree Cluster representation manages to preserve most of the mutual information (of the original distribution, 0.277 of 0.278) between variables, which translates to minimal error on computed probabilities (< 0.005). It is important to note that the case of Disjoint Clusters while better than FF, is still short of capturing all the dependencies faithfully in the distributions: it considers independent a pair (out of only 16) variables with $MI = 0.11$ (the maximum correlations between two different variables have $MI = 0.27$). This results in a probability error 10 times higher (0.05) than using Tree Cluster, which is significant for a small system.

In terms of inference, as evident in Fig.7(a), our method is the most accurate: 20 times less errors overall than Disjoint clusters, and 30 times less maximal error, while being only 20% slower. On the other hand, even though Disjoint Clusters capture most correlations for each distribution (Fig.6 (a) 0.26 out of 0.278), it produces sizable errors (0.12).

6.4 Abstracted Apoptosis Pathway

Fig.6 (b) shows the statistics for an arbitrary time point (corresponding to 30 minutes). Our Tree Cluster approach captures most of the mutual information between variables (0.1 out of 0.12), and the maximum error on the MI is the least (0.12) compared to FF or Disjoint Clusters ($0.2, 0.32$). Also, the size of the representation does not increase too much (225 values vs 125 or 50).

Fig. 4 shows the two sets of clusters computed by algorithm 1 [5] at the arbitrarily chosen time of 20 and 90 minutes. Most links of the tree follow direct correlations, except for the link Bid-cPARP at 90 minutes. Our interpretation is that at 90 minutes, Bid does not play much of a role anymore, and its correlation is not meaningful. Further, Bax, Bcl2c and Mcl1, which are highly correlated, and which transduce or inhibit the signal are connected towards the downstream only through R^*. The reason is that the correlation with R^* is higher than the direct correlations, and the direct correlations are removed by the algorithm. Notice that this interaction graph can change in time (compare at time 20 and 90 when Bid swaps from one side of the tree to the other side).

In terms of inference, Fig.7(b) shows that our algorithm based on Tree Cluster makes minimal error (≤ 0.06). In terms of trade off, it makes half the errors compared with Disjoint Clusters and takes only 1.5 times longer to compute. Compared with FF, it improves accuracy by 7-8 times (FF is very inaccurate on some variable), while being 6.3 times slower. Fig. 8 (left) shows the dynamic of the marginal for RAct over time as computed by the different algorithms: Tree Cluster is extremely close to the simulative curve, while Disjoint Cluster is considerably off and FF makes larger errors.

6.5 EGF-NGF Pathway

In case of the EGF-NGF pathway, we consider a biologically reasonable set of disjoint clusters, grouping a species with its activated form, as their concentrations are very correlated. We display on Fig. 5 the approximated correlations obtained using the different approximations at time 5 minute of the EGF-NGF pathway. Tree Cluster manages to keep some correlations among almost every pair of variables, which is not the case for FF or Disjoint clusters, assuming independence of almost every pair of variables. The loss of information is also minimal, as confirmed by Fig.6 c) (MI error ≤ 0.07).

We now consider inference. This pathway allows us to compare the inference algorithms with another approximated algorithm, called *HFF (Hybrid FF)* [17]. In short, HFF keeps a small number of joint probabilities of high value (called spikes), plus an FF representation of the remaining of the distribution. The more spikes, the more accurate and the slower the algorithm. As HFF has been implemented in another language (C++) on a different data structure, we report the error with FF as the baseline in order to draw a fair comparison, in terms of errors (FF=100%) and time (FF=$1x$). The superiority of our approach for inference is even more evident in this case. Fig.7(b) shows that our method produces 3 times less errors overall than the most accurate method considered before. The maximal errors and number of errors greater than 0.1 are also substantially reduced. FF and disjoint clusters can be 2 to 4 times faster, but with very large errors (see Fig. 8 right for an example of large error), while HFF proposes worse results both for time and accuracy.

6.6 Discussion on the different approximations

We now compare the different representations of the distributions. We report statistics on all pathways on Fig. 6. FF captures only the auto-correlations of each variable with itself, while it does not keep any correlation of 2 different variables. This helps understand how much of correlations between variables are lost by the approximations (doing MI real - MI FF): as said above, by definition, all is lost by FF, while

(a) Enzyme catalytic reaction:

Method	Max. Error	Mean Error (normalized)	Nb. Error > 0.1	Comput. Time
FF	0.17	100%	49	0.2s
Disj. Cluster	0.12	65%	16	0.5s
Tree Cluster	0.004	3%	0	0.6s

(b) Apoptosis pathway:

Method	Max. Error	Mean Error (normalized)	Nb. Error > 0.1	Comput. Time
FF	0.44	100%	124	2.2s
Disj. Cluster	0.12	24%	2	9.8s
Tree Cluster	0.06	14%	0	13.8s

(c) EGF-NGF pathway (normalized wrt FF for comparison with HFF):

Method	Max. Error	Mean Error	Nb. Error > 0.1	Comput. Time
FF	100%	100%	100%	1x
HFF (3k)	62%	60%	50%	10x
HFF (32k)	49%	38%	35%	1100x
Disjoint Cluster	84%	79%	84%	1.9x
Tree Cluster	32%	14%	16%	4.2x

Fig. 7. Table representing the errors of the different inference algorithms.

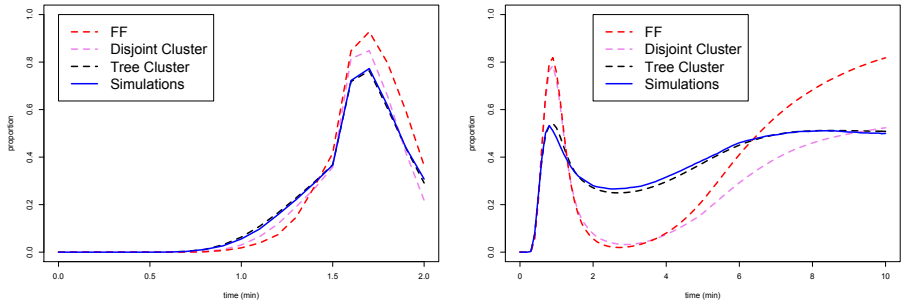


Fig. 8. Evolution of $P(cPARP = 2)$ in the apoptosis pathway (left) and of $P(ErkAct = 2)$ in the EGF-NGF pathway(right) as computed by the inference based either on FF or Tree cluster approximations (broken lines), compared with the real value (solid line).

disjoint cluster keeps less than one third of correlations. On the other hand, the Tree Cluster approximation keeps more than two third of the correlations, while having almost the same representation size (< 800 values) as disjoint clusters. Further, there are pairs of variables highly correlated that FF (resp. disjoint cluster) considers independent ($MI > 0.3$, resp. $MI = 0.2$), which is not the case for the Tree Clustered representation ($MI < 0.12$).

6.7 Discussion on Inference Algorithms

We compare the evolution of concentrations of each molecule using the different inference algorithms (Fig. 7). Overall, performing inference based on the tree clustered representation is fast (less than 15 seconds), while being the most accurate of all the inference algorithms we tested, included HFF with a lot of spikes (32k). To visualize the errors incurred by different approximations, we draw in Fig.8 the probability that Erk^* takes a medium concentration in the EGF-NGF pathway and the probability that R^* takes a medium concentration in the apoptosis pathways. The tree cluster approximation follows very closely the simulative curve (in this examples as well as in every examples we considered), while other algorithms are further away, sometimes being far from what is computed by simulations.

It was not clear before the experiments whether using Chow Liu algorithm would produce accurate results for biopathways, and if yes, for which types of biopathways. On Figure 8 with (a) 4, (b) 10 and (c) 52 variables, all showed great accuracy while keeping a good computational time.

FF is the most widely used procedure in Inference so far. While Disjoint clusters is more precise while not being much slower, its main drawback is that there is no general method to choose a good disjoint clustering, although it has been used with ad hoc expert clustering. HFF is more confidential, as accuracy improvement is obtained at a very large computational price, and it cannot be used in practice. Notice that MI is not used in any of these methods.

We use the Chow-Liu algorithm and MI in order to obtain non-disjoint clusters: From few pre-computed simulations of the systems, optimal trees of clusters of size 2 are computed for each time point. Thus it can be used easily, without input from an expert. Its accuracy is the best, and its computational price very limited.We still expect FF to be used for fast screening however as it is very efficient. When accuracy matters, Tree Cluster inference should be used.

7 Conclusion and Perspectives

In this paper, we reviewed several approximated representations of probability distributions. We also discussed how these representations can be applied to perform inference in discrete stochastic models. With different case studies, we

show that the approximation based on non-disjoint clusters of size two forming a tree structure offers a very good trade-off between accuracy and tractability.

In future work, we aim at modeling and studying a tissue, made of tens of thousands of cells, each subject to the dynamics of a pathway. In this context, capturing the inherent variability of the population of cells is crucial. First, we would abstract the low level model of the pathway of a single cell into a stochastic discrete abstraction, e.g. using our tool *DBNizer* [16]. Secondly, we would use a model of the tissue, which does not explicitly represent every cell but qualitatively explains how the *population* evolves. We plan to obtain such populations model from more common agents based model of tissues (e.g. [18]). We will finally use approximate representations of the distributions, as discussed in this paper, to handle multilevel biological systems.

ACKNOWLEDGMENTS

ANR-13-BS02-0011-01 Stoch-MC funded parts of this works.

REFERENCES

[1] Albeck, J. G., Burke, J. M., Spencer, S. L., Lauffenburger, D. A., and Sorger, P. K. (2008). Modeling a snap-action, variable-delay switch controlling extrinsic cell death. *PLoS Biology*, **6**(12), 2831–2852.

[2] Bertaux, F., Stoma, S., Drasdo, D., and Batt, G. (2014). Modeling Dynamics of Cell-to-Cell Variability in TRAIL-Induced Apoptosis Explains Fractional Killing and Predicts Reversible Resistance. *PLoS Comput Biol*, **10**(10), 14.

[3] Boyen, X. and Koller, D. (1998). Tractable inference for complex stochastic processes. In *Proc.of Uncertainty in Artificial Intelligence*, pages 33–42. Morgan Kaufmann.

[4] K. S. Brown, C. C. Hill, G. A. Calero, K. H. Lee, J. P. Sethna, and R. A. Cerione. The statistical mechanics of complex signaling networks: nerve growth factor signaling. *Physical Biology 1*, pages 184–195, 2004.

[5] C. K. Chow, C. N. Liu. Approximating discrete probability distributions with dependence tree. In *IEEE Trans. on Information Theory,* Vol. 14(3): 462–467, 1968.

[6] I. Csiszar. I-divergence geometry of probability distributions and minimization problems. In *The Annals of Probability,* Vol. 3(1): 146–158, 1975.

[7] F. Malvestuto. Approximating Discrete Probability Distributions with Decomposable Models. In *Trans. on Systems, Man and Cybernetics,* Vol. 21(5): 1287–1294, 1991.

[8] X. Gao, C. Arpin, J. Marvel, S. Prokopiou, O. Gandrillon, F. Crauste (2016). IL-2 sensitivity and exogenous IL-2 concentration gradient tune the productive contact duration of CD8+ T cell-APC: a multiscale modeling study. In *BMC Systems Biology*, 10, 77.

[9] Feret, J., Danos, V., Krivine, J., Harmer, R., and Fontana, W. (2009). Internal coarse-graining of molecular systems. *PNAS*, **106**(16), 6453–8.

[10] Gilbert, D., Heiner, M., Takahashi, K., Uhrmacher, A. Multiscale Spatial Computational Systems Biology. In *Dagstuhl Reports, Seminar 14481*, (4) 11, 2015.

[11] Liu, B., Zhang, J., Tan, P. Y., Hsu, D., Blom, A. M., Leong, B., Sethi, S., Ho, B., Ding, J. L., and Thiagarajan, P. (2011a). A computational and experimental study of the regulatory mechanisms of the complement system. *PLoS Comput Biol*, **7**(1), e1001059.

[12] Liu, B., Hsu, D., and Thiagarajan, P. (2011b). Probabilistic approximations of odes based bio-pathway dynamics. *Theoretical Computer Science*, **412**(21), 2188–2206.

[13] Munsky, B. and Khammash, M. (2006). The finite state projection algorithm for the solution of the chemical master equation. *J. Chem. Phys.*, **124**(4), 044–104.

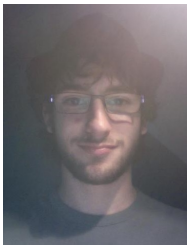
[14] Murphy, K. and Weiss, Y. (2001). The factored frontier algorithm for approximate inference in DBNs. In *Proc. of Uncertainty in Artificial Intelligence*, pages 378–385. Morgan Kaufmann.

[15] Palaniappan, S. K., Pichené, M., Batt, G., Fabre, E., and Genest, B. (2016). A look-ahead simulation algorithm for dbn models of biochemical pathways. In *HSB*. Lecture Notes in Bioinformatics.

[16] Palaniappan, S. K., Bertaux, F., Pichené, M., Fabre, E., Batt, G., and Genest, B. (2017). Stochastic Abstraction of Biological Pathway Dynamics: A case study of the Apoptosis Pathway. In *BIOINFORMATICS*, btx095, Oxford University Press.

[17] Palaniappan, S. K., Akshay, S., Liu, B., Genest, B., Thiagarajan, P.S (2012). A Hybrid Factored Frontier Algorithm for Dynamic Bayesian Networks with a Biopathways Application. In *TCBB 9(5)*:1352-1365, IEEE/ACM.

[18] Waclaw, B., Bozic, I., Pittman, M., Hruban, M., Vogelstein, B., Nowak, M. (2015). A spatial model predicts that dispersal and cell turnover limit intratumour heterogeneity. *Nature 525*, 261-264.



Matthieu Pichené received his M.Sc degree from University Paris 10 Orsay in 2014, and his Ph.D from University of Rennes 1 in 2018. His main research interest focuses on computational systems biology.

Sucheendra K. Palaniappan received his Ph.D in computer science from School of Computing, National University of Singapore, Singapore (2013). Currently he is working as a Scientist with the The Systems Biology Institute, Tokyo. His main research interests include computational systems biology and machine learning.

Eric Fabre graduated from the Ecole Nationale Superieure des Telecommunications, Paris, France, and received the M.Sc. degree in probability and statistics and the Ph.D. degree in electrical engineering from the University of Rennes, Rennes, France, in 1990, 1993, and 1994, resoectively. In 1995, he was a Postdoctoral Researcher at the Laboratory of System Theory and Bioengineering (LADSEB-CNR), Padua, Italy. Since 1996, he has been with the Institut National de Recherche en Informatique et Automatique (INRIA), Rennes, France. His main research interests are related to graphical models/Bayesian networks and their applications to multiresolution signal and image processing, and to digital communications. Recently, he has extended these concepts to distributed discrete-event systems

Blaise Genest is a former student of ENS Cachan (Mathematics and Computer Science, 1999-2003). He received his Ph.D. in Computer Science from University Paris 7, France (2004). He is a full time permanent researcher at French CNRS in Computer Science. His current research interests are computational systems biology and model checking.

APPENDIX 1: PROBABILITY DISTRIBUTION DEFINED BY A CLUSTER TREE

The main developments of the present paper focus on a specific subclass of cluster tree approximations for a given multivariate distribution, namely the class where clusters have a maximum size of two. For the use-case addressed here, the modeling of the dynamics of a population of cells, this simple class already provides significant improvements compared to the classical FF (fully factored) approach, for a minimal computation overhead. In this appendix, we present the general case, which enables even better approximations, still for a reasonable overhead. All developments of the paper can be extended to this more general setting. The main idea is to allow for (overlapping) clusters of more than two variables.

We start by defining the appropriate choices of clusters to which our approximation scheme applies. Let $I = K_1 \cup ... \cup K_c$ be a decomposition of the index set I into c clusters. Let us associate a graph $G = (I, E)$ to this cluster covering in the following manner: vertices are defined by I, and the edge $\{j, k\}$ is present in E iff there exists some cluster K_i such that $j, k \in K_i$. In other words, each cluster K_i defines a clique (complete subgraph) in G. We say that graph G is a *cluster tree* (see Figure 9) iff

1) G is a triangulated graph, i.e. any cycle $(v_0, v_1, v_2, ..., v_k = v_0)$ in G of length $k > 3$ contains a *chord*, that is an edge $\{v_i, v_j\}$ with $j \geq i + 2$,
2) each maximal clique in G coincides with some cluster K_i.

Without loss of generality, one can assume that clusters $K_1, ..., K_c$ are precisely the maximal cliques in G. When clusters are limited to two variables, G is a cluster tree iff it is a tree.

In a cluster tree, it is well known that maximal cliques organize into a tree, hence the name (see Figure 9, where maximal cliques have size 3). Such a tree of clusters can be obtained by connecting clusters one at a time: when cluster K_i is introduced, it can be attached to cluster K_j only if the variables that K_i shares with previously attached clusters all lie inside K_j. The sequence $K_1, ..., K_c$ forms an *adequate ordering* of clusters iff it corresponds to an ordering in which clusters can be attached to form a tree. The tree of clusters is not unique, and for a given tree, there exist numerous adequate orderings.

Our objective is now to approximate P by some distribution P_{NDC} (for non-disjoint clusters) in such a way that P and P_{NDC} coincide on all clusters, i.e. $P(x_{K_i}) = P_{\text{NDC}}(x_{K_i})$ for all values x_{K_i} and $1 \leq i \leq c$. It turns out that on a cluster tree, knowing the marginals of P_{NDC} on each cluster fully determines the complete joint distribution P_{NDC}. Assuming $K_1, ..., K_c$ is an adequate ordering of clusters, P_{NDC} can be derived as follows

$$P_{\text{NDC}}(x) \quad = \quad \prod_{i=1}^{c} \frac{P(x_{K_i})}{P(x_{K_i^{old}})} \tag{24}$$

where $K_i^{old} = K_i \cap (\cup_{j<i} K_j)$ represents variables of K_i already present in $K_1 \cup ... \cup K_{i-1}$. The generic term in the product thus corresponds to $P(x_{K_i}|x_{K_i^{old}})$. This generic term also coincides with $P(x_{K_i}|x_{K_i \cap K_j})$, where K_j is the cluster to which K_i is attached in a cluster tree.

Clearly, (24) generalizes the disjoint cluster equation, which in turn generalizes the fully factored equation.

Proposition 3. P_{NDC} defined in (24) for a cluster tree is a proper probability distribution over X, i.e. $\sum_{x \in V^X} P_{\text{NDC}}(x) = 1$. Moreover, P_{NDC} does not depend on the adequate ordering of clusters used in (24). It does not depend either on the cluster tree matching clusters $(K_i)_{1 \leq i \leq c}$. P_{NDC} coincides with P on each cluster $(K_i)_{1 \leq i \leq c}$.

Proof: To prove the first point, simply observe that each generic term in (24) is the conditional distribution of X_{K_i} given $X_{K_i^{old}}$, $P(X_{K_i}|X_{K_i^{old}})$. By marginalizing out variables in the reverse order of their introduction, one obtains the desired result. We only sketch the proof of the second point. For a given cluster tree, observe that two leaves can be attached to the current tree in any order without changing the final P_{NDC}. Moreover, one can also invert the first and second clusters in (24) and get the same result. So, by recursion, this allows one to start with any cluster as the root of the tree, i.e. with any cluster as K_1. This immediately yields the fourth and last point, as one has $P_{\text{NDC}}(X_{K_1}) = P_(X_{K_1})$ from (24). Finally, for the third point, we rely again on the fact that the generic term in (24) is the conditional distribution $P(X_{K_i}|X_{K_i^{old}})$, which does not depend on the cluster K_j to which the new cluster K_i is attached when building the cluster tree. □

For the specific case where cluster size is limited to two variables, as it is the case in the main part of the paper, (24) can be reformulated with a more symmetrical shape. Specifically, one has

$$P_{\text{NDC}}(x) \quad = \quad \frac{\prod_{i=1}^{c} P(x_{K_i})}{\prod_{i=1}^{N} P(x_i)^{C_i - 1}} \tag{25}$$

where C_i is the number of clusters that contain variable X_i. This expression can be easily derived by recursion from (24), and it coincides with the equation that we adopted in the paper. Notice that Proposition 3 becomes more obvious with this new formulation.

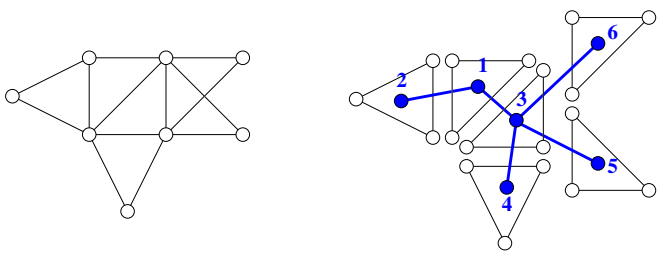


Fig. 9. In a triangulated graph (left), maximal cliques can be organized into a tree (right). Numbers indicate one possible adequate ordering.

APPENDIX 2: IMPROVING COMPLEXITY OF ALGORITHM 2

We now refine Algo. 2, that performs approximated inference, in order to lower the exponential factor of $|V|$ from $|\hat{\imath} \cup \hat{\jmath}| + 2$ to $\max(|\hat{\imath}|, |\hat{\jmath}|) + 2$. The improvement can thus be significant.

***Theorem* 2.** Given $B^0 = P^0$, one can compute $B^1, \ldots, B^T$ in time $O(T \cdot |I| \cdot (|I| + |V|) \cdot \lambda \cdot |V|^{\ell+1})$, where $\ell = max_{t,\{i,j\} \in E^t} \, max(|\hat{\imath}|, |\hat{\jmath}|)$.

Proof: In the original algorithm, the complexity factor in $|\hat{\imath} \cup \hat{\jmath}|$ comes from the necessity of integrating over values of $B^{t-1}(x_{\hat{\imath} \cup \hat{\jmath}})$. So we propose a technique to reduce the complexity of this integration. Let us first simplify notations for clarity (dropping in particular the time index): the aim is to compute

$$B(y_i, y_j) \quad = \quad \sum_{x_{\hat{\imath} \cup \hat{\jmath}}} B(x_{\hat{\imath} \cup \hat{\jmath}}) \, P(y_i|x_{\hat{\imath}}) \, P(y_j|x_{\hat{\jmath}}) \qquad (26)$$

where distribution $B(X)$ is a tree distribution on $G = (I, E)$, which is thus defined by the tuple $[\, B(X_u, X_v) \,]_{\{u,v\} \in E}$.

As a first step of the proof, let us consider a *separating set* $K \subseteq I$ of nodes that separates nodes of $\hat{\imath}$ from those of $\hat{\jmath}$ on tree $G = (I, E)$, i.e. any path from $\hat{\imath}$ to $\hat{\jmath}$ on G crosses K (see Fig.10). Notice that one has $\hat{\imath} \cap \hat{\jmath} \subseteq K$, and that taking $K = \hat{\imath}$ or $K = \hat{\jmath}$ satisfies this separation property. So the size of K can be upper bounded by $min(|\hat{\imath}|, |\hat{\jmath}|)$. The fact that K separates $\hat{\imath}$ from $\hat{\jmath}$ entails that $X_{\hat{\imath}}$ and $X_{\hat{\jmath}}$ are conditionally independent given X_K for distribution B. Therefore

$$B(x_{\hat{\imath} \cup \hat{\jmath}}) \quad = \quad \sum_{x_K} B(x_{\hat{\imath}}|x_K) B(x_{\hat{\jmath}}|x_K) B(x_K) \qquad (27)$$

By plugging this expression into (26) one gets

$$B(y_i, y_j) \quad = \quad \sum_{x_K} P(y_i|x_K) \, P(y_j|x_K) \, B(x_K) \qquad (28)$$

where

$$P(y_i|x_K) \quad = \quad \sum_{x_{\hat{\imath}}} P(y_i|x_{\hat{\imath}}) \, B(x_{\hat{\imath}}|x_K) \qquad (29)$$

$$P(y_j|x_K) \quad = \quad \sum_{x_{\hat{\jmath}}} P(y_j|x_{\hat{\jmath}}) \, B(x_{\hat{\jmath}}|x_K) \qquad (30)$$

Notice that in (29) – and similarly for (30) – the summation over values $x_{\hat{\imath}}$ is actually a summation over values $x_{\hat{\imath} \setminus K}$ as the values of $x_{\hat{\imath} \cap K}$ are imposed by the conditional distribution $P(x_{\hat{\imath}}|x_K)$. Compared to (26), computing (28) for all x_K now has time complexity $|V|^{|K|}$ and the complexity of deriving $P(X_K)$ also matches this exponent. This concludes the proof if one can guarantee that $|K| \leq max(|\hat{\imath}|, |\hat{\jmath}|)$ and if the complexities of (29)-(30) are upper bounded in the same manner. This is actually the delicate part, as a term like $B(X_{\hat{\imath}}|X_K)$ for example has $|V|^{|\hat{\imath}|+|K|}$ many entries, so we cannot compute this term directly.

The second step of the proof considers a particular choice for set K which guarantees that (29)-(30) can be computed efficiently. First, for $k \in K$ not a leaf of G, we define the *k-section of G* as follows: it is the minimal subtree of G rooted at k, containing at least 2 nodes, and such that its leaves are either in K or are leaves of G. For instance, on Fig. 10, we have $r \in K$, and the r-section has 6 nodes, including r and s. It has three leaves, two being the children of s, plus another

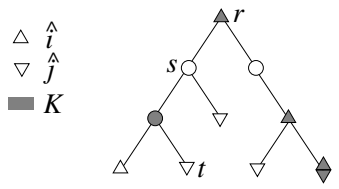


Fig. 10. A separating set K (nodes in gray) between $\hat{\imath}$ and $\hat{\jmath}$.

leaf in K. For $k, k' \in K$ not leaves of G, the intersection between the k-section and the k'-section have either no node in common, or only one, which is either k or k'. For $k \in K$ not a leaf of G, we define the *strict k-section* of G as the k-section minus the leaves of the k-section which are in K.

We now define a separating set K such that every strict k-section contains at least one node of $\hat{\imath}$ and one node of $\hat{\jmath}$. The set K is built bottom-up, that is recursively starting from leaves and progressing up towards the root of G. Each node will be tagged by a number in $\{0, 1, 2, 3\}$. Nodes tagged 3 will be nodes of K. We first tag a leaf by:

- 0 if it is not in $\hat{\imath} \cup \hat{\jmath}$,
- 1 if it is in $\hat{\imath} \setminus \hat{\jmath}$,
- 2 if it is in $\hat{\jmath} \setminus \hat{\imath}$,
- 3 if it is in $\hat{\imath} \cap \hat{\jmath}$.

We then inductively tag a node by:

- 3 if it has a child tagged by 1 (or itself is in $\hat{\imath}$) and if it has a child tagged by 2 (or itself is in $\hat{\jmath}$). Else:
- 1 if it is in $\hat{\imath} \setminus \hat{\jmath}$ or as at least one child tagged as 1,
- 2 if it is in $\hat{\jmath} \setminus \hat{\imath}$, or has one child tagged by 2
- by 0 otherwise.

At the end of the procedure, we set K as the set of nodes tagged by 3. That is, a node k where at least one branch of type i and one of type j meet is declared to belong to K. The separating set K in Fig. 10 has been built using this algorithm. Node s is tagged 2 as it has one child in $\hat{\jmath}$. Then r is tagged 3 as it is itselfss in $\hat{\imath}$ and it has has a child (node s) tagged by 2. It is easy to see that K is a separating set, and that for all $k \in K$ not a leaf of G, the strict k-section contains at least one node of $\hat{\imath}$ and one node of $\hat{\jmath}$ (possibly, this is the same node if it is in $\hat{\imath} \cap \hat{\jmath}$, and possibly, this is k itself). In particular, $|K| \leq min(|\hat{\imath}|, |\hat{\jmath}|)$.

The third step of the proof examines an efficient algorithm that compute $P(y_i|x_K)$ from $P(y_i|x_{\hat{\imath}})$. It inductively eliminates nodes in $\hat{\imath} \setminus K$, section by section, in a bottom-up fashion, removing nodes in $\hat{\imath} \setminus K$ and adding nodes from K. We consider the properties of this procedure with respect to $\hat{\imath}$, but the same holds with respect to $\hat{\jmath}$.

Let us define a set of nodes that will evolve in our procedure :

- $A \subseteq K \cup \hat{\imath}$ progressively introduces nodes of K in replacement of nodes of $\hat{\imath}$, initialized to $A = \hat{\imath}$. At the end of the procedure, $A = K$.

Let us start from deepest k-sections in the tree, for $k \in K$. First, if k a leaf, there is nothing to do as $k \in K \cap \hat{\imath}$. The set A stays the same. Otherwise, for all k' below k in the tree, k' has already been considered by induction, that is $k' \in A$. Let C be the subset of nodes in the strict-k-section that are in $\hat{\imath}$. We will explain how to perform an operation amounting to:

$$A \quad := \quad (A \setminus C) \cup \{k\}$$

At every step, $|A|$ does not increase, as $|C| \geq 1$ (in the case where $C = \{k\}$, nothing happens). At the begining of the procedure, we have $P(y_i|x_{\hat{\imath}})$, that is $P(y_i|x_A)$ as A is initialized to $\hat{\imath}$. It is easy to inductively compute $P(y_i|x_{(A\setminus C)\cup\{k\}})$ from $P(y_i|x_A)$. Adapting (29), we obtain:

$$P(y_i|x_{(A\setminus C)\cup\{k\}}) \quad = \quad \sum_{x_{C\setminus\{k\}}} P(y_i|x_A)\, B(x_A|x_{(A\setminus C)\cup\{k\}})$$

Let D be the set of nodes in K that are in the k-section, and let $E = A \setminus (C \cup D)$. We have $A \cup \{k\} = C \cup D \cup E$ and $(A \setminus C) \cup \{k\} = D \cup E$. Using (24), as E is separated from C, D by D, we obtain:

$$B(x_A|x_{(A\setminus C)\cup\{k\}}) = \frac{B(x_{C\cup D\cup E})}{B(x_{D\cup E})} = \frac{B(x_{C\cup D})}{B(x_D)}$$

It thus suffices to set:

$$P(y_i|x_{(A\setminus C)\cup\{k\}}) := \sum_{x_{C\setminus\{k\}}} P(y_i|x_A) \cdot \frac{B(x_{C\cup D})}{B(x_D)}$$

Let us now analyse the complexity of the algorithm. The recursion step requires first to compute $B(x_{C\cup D})$ and $B(x_D)$. This can be done in time $O(|I| \cdot V^{|C\cup D|})$. We now show that $|C \cup D| \leq |\hat{\imath}| + 1$. Let us partition $\hat{\imath}$ into $J_1 \uplus J_2 \uplus J_3$, with J_1 the set of nodes of $\hat{\imath}$ which are *not* below k, $J_2 = C$ and J_3 the rest, that is nodes below the strict k-section. By construction, each strict k-section contains at least a node of $\hat{\imath}$. It means that $|D \setminus \{k\}| \leq J_3$. Thus $|C \cup D| \leq 1 + |J_2| + |J_3| \leq 1 + |\hat{\imath}|$. Then we need to perform the summation $\sum_{x_{C\setminus\{k\}}} P(y_i|x_B) \cdot \frac{B(x_{C\cup D})}{B(x_D)}$. For that, we need to consider a table with $1 + |A \cup C \cup D| = 1 + |A \cup \{k\}|$ variables. As $|A| \leq |\hat{\imath}|$, it gives a tables with at most $|V|^{|\hat{\imath}|+2}$ entries. Computing the sum is linear in the number of entries. We then need to repeat this process for all $k \in K$, which is less than $|\hat{\imath}|$ times. Hence one can compute $P(y_i \mid x_K)$ for all y_i, x_K in time $O(|\hat{\imath}|(|I| + |V|)|V|^{1+|\hat{\imath}|})$.

To conclude the proof, let us gather all elements. We have T time points and $|I|$ clusters (as they form a tree). For each pair of values (y_i, y_j), one must derive $P(y_i|x_K), P(y_j|x_K)$ and then perform (28). The latter has complexity $O(|V|^K)$ with $|K| \leq min(|\hat{\imath}|, |\hat{\jmath}|)$, which is clearly dominated by the computation of $P(y_i|x_K), P(y_j|x_K)$. This results in a total complexity of $O(T \cdot |I| \cdot (|I| + |V|) \cdot \ell \cdot |V|^{\ell+1})$. □

Notice that it is a crude upper bound on the worst case complexity, and the actual complexity will be almost always better than that by a factor $|V|$ to $|V|^2$. If further there is a node in $|\hat{\imath} \cap \hat{\jmath}|$ for all $(i, j) \in E^t$, it suffices to set it as the root of G to obtain an immediate improvement of factor $|V|$. Also, when removing nodes from $\hat{\imath}$, one can remove branches of k-sections which does not contain nodes in $\hat{\imath}$. We obtained improvement of an order of magnitude using it, and no slowdown.

APPENDIX 3: ERROR ANALYSIS

We can analyze the error $\Delta^t = |P^t - B^t|$ obtained at time t, w.r.t. the one step error $\epsilon_0 = \max_t |Q^t - B^t|$, when using Algo. 2.

Following [3], [17], this scheme ensures that, denoting by $\beta \leq 1$ the contraction factor associated with the DBN:

Proposition 4. $\Delta^t \leq \epsilon_0 \sum_{j=0}^{t} \beta^j$. Further, if $\beta < 1$, we have $\Delta^t \leq \frac{\epsilon_0}{1-\beta}$.

Proof: By definition, we have that after applying the CPTs to two distributions P, P', the results $\tilde{P}, \tilde{P}'$ will be at distance at most $|\tilde{P} - \tilde{P}'| \leq \beta|P - P'|$. In particular, we have that $|P^t - Q^t| \leq \beta|P^{t-1} - B^{t-1}|$.

Now, we shall show that Δ^t can be bounded by $\epsilon_0(\sum_{j=0}^{t} \beta^j)$. By definitions and triangular inequality, we have:

$$\begin{aligned}\Delta^t &= |B^t - P^t| \\ &\leq |B^t - Q^t| + |Q^t - P^t| \\ &\leq \epsilon_0 + \beta\Delta^{t-1}\end{aligned}$$

Then by recursively computing the second factor, we obtain,

$$\begin{aligned}\Delta^t &\leq \epsilon_0 + \beta_t\epsilon_0 + \beta\beta\epsilon_0 + \ldots + (\beta\beta\cdots\beta)\epsilon_0 \\ &\leq \epsilon_0(\sum_{j=0}^{t} \beta^j)\end{aligned}$$

Further if $\beta < 1$, we have:

$$\Delta^t \leq \epsilon_0(\sum_{j=0}^{t} \beta^j) \leq \epsilon_0(\sum_{j=0}^{\infty} \beta^j) = \frac{\epsilon_0}{1-\beta}$$

□

We can further analyze on the fly the one step error ϵ_0 made at each step. For that, it suffices to consider the result of [5] for the Chow Liu approximation: we have that the one step error at step k is $\varepsilon^k = |B^t - Q^t| = \sum_i H^t(x_i) - H^t(X) - \sum_{(i,j)\text{clusters}} H^t(x_i, x_j)$, where H^t stands for the entropy (at time t), defined as follows:

$$H^t(X) = -\sum_{x_X \in V^X} Q^t(x_X) \log Q^t(x_X)$$

Now, $H(x_i, x_j)$ and $H(x_i)$ are already computed by our algorithm for all i and all clusters (i, j). Computing $H^t(X)$ exactly is however more complex, as Q^t is a multivariate distribution over tens of variables. Nevertheless, it suffices to under-approximate it in order to over-approximate the one step error ε^k.

Under-approximating the entropy: One easy under-approximation is $H(X) \geq 0$. To improve it, one can compute better values by computing a subset S of tuples for which $Q(x)$ is large, and under-approximate $Q(x)$ for these tuples. This can be done in a way very similar to the computation of spikes in [17]:

It suffices to use $B^t(x_i, x_j)$ and $B^{t+1}(y_i, y_j)$ for clusters in order to select thousands of tuples x at time t and y at time $t + 1$ with potentially large $B(x)$ and $Q(y)$ (ones which have the largest projection on clusters). Let S^t and S^{t+1} be these two sets of tuples. For $x \in S^t$, the probability $B^t(x)$ is computed exactly from values of $B^t(x_i, x_j)$ for the clusters. For $y \in S^{t+1}$, we under-approximate $Q^{t+1}(y) \geq \sum_{x\in S^t} B^t(x) \prod_i CPT_i^t(y_i \mid x_{\hat{\imath}})$.

We then use the following to under-approximate $H^{t+1}(X)$:

$$H^{t+1}(X) \geq -\sum_{y\in S^{t+1}} Q^{t+1}(y) \log(Q^{t+1}(y))$$