Bioinformatics, 2017, 1–7
doi: 10.1093/bioinformatics/btx095
Advance Access Publication Date: 14 February 2017
Original Paper

OXFORD

Systems biology

Abstracting the dynamics of biological pathways using information theory: a case study of apoptosis pathway

Sucheendra K. Palaniappan[1,2,3], François Bertaux[2], Matthieu Pichené[1], Eric Fabre[1], Gregory Batt[2,*,†] and Blaise Genest[4,*,†]

[1]INRIA, Rennes, France, [2]INRIA, Paris-Saclay, France, [3]The Systems Biology Institute, Tokyo, Japan and [4]CNRS, IRISA, Rennes, France

*To whom correspondence should be addressed.
†The authors wish it to be known that, in their opinion, the last two authors should be regarded as Joint Last Authors.
Associate Editor: Jonathan Wren

Received on August 25, 2016; revised on January 19, 2017; editorial decision on February 7, 2017; accepted on February 10, 2017

Abstract

Motivation: Quantitative models are increasingly used in systems biology. Usually, these quantitative models involve many molecular species and their associated reactions. When simulating a tissue with thousands of cells, using these large models becomes computationally and time limiting.
Results: In this paper, we propose to construct abstractions using information theory notions. Entropy is used to discretize the state space and mutual information is used to select a subset of all original variables and their mutual dependencies. We apply our method to an hybrid model of TRAIL-induced apoptosis in HeLa cell. Our abstraction, represented as a Dynamic Bayesian Network (DBN), reduces the number of variables from 92 to 10, and accelerates numerical simulation by an order of magnitude, yet preserving essential features of cell death time distributions.
Availability and Implementation: This approach is implemented in the tool DBNizer, freely available at http://perso.crans.org/genest/DBNizer.
Contact: gregory.batt@inria.fr or bgenest@irisa.fr
Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Quantitative models play an ever increasing role in systems and synthetic biology. Various modeling frameworks can be employed depending on the specificities of the studied system. One can mention ordinary differential equation (ODE) models for deterministic systems, continuous time Markov chain (CTMC) models for stochastic systems and hybrid stochastic/deterministic (HSD) models. Obtaining analytical solutions of these models is almost always impossible. Moreover, because of the ever-increasing size and complexity of these models, even numerical simulation can become challenging. When a very large number of simulations are necessary this becomes all the more difficult. This is notably the case for multi-scale simulations in which the dynamics of a tissue or an organ, made of hundreds of thousands of cells, each described by a complex model, has to be analyzed. We propose to build an abstracted model using thousands of simulations of the original model and then use this abstracted model to as a place needed millions of simulations, speeding up the simulation of the tissue.

A range of different approaches for computing model abstractions has been proposed in the past. For example, finite state projection (Munsky and Khammash, 2006), moment closure (Gillespie, 2009) methods, coarse-graining (Feret *et al.*, 2009) and trajectory-based state coarsening (Michaelides *et al.*, 2016) enable efficient (approximate) simulation of CTMC models. ODE models can be abstracted using methods exploiting time scale separation (Gunawardena, 2014), tropical analysis (Radulescu *et al.*, 2015), or stochastic discrete abstraction (Liu *et al.*, 2011b). The latter study is the most related to the work in this paper.

The approach proposed by Liu and colleagues has been shown to approximate well the quantitative dynamics of a large set of ODE models of signal transduction pathways (Liu *et al.*, 2011a,b). Moreover, it has lead to novel findings in immune system regulation (Liu *et al.*, 2011a). The approach is based on Dynamic Bayesian Networks (DBNs), a class of probabilistic graphical models. In this approach, each dimension is partitioned into a few intervals, leading to a hyper-rectangular partition of the state space. Ideally, one would like to know the probabilities of the transitions of the system state from any rectangular region to any other region at regular time instants, which would then provide a coarse but very simple means to simulate approximately the behavior of the system. For non-toy models, storing and reusing this information for simulation is computationally intractable. Instead one can compute and store—in the so-called conditional probability tables (CPTs)—the probabilities that a given variable remains in its interval or switches to another one given the values of the variables that influence the variable of interest at the previous time instant ('parent' variables in the DBN terminology). In its standard implementation, parents are the set of variables that directly appear in the ODE governing the temporal evolution of the variable of interest. We will show that this choice of parents is not always appropriate: the quantitative information they convey decreases with the increase in duration between two successive instants in the DBN, for which concentrations of ancestors may be more correlated to the variable of interest than the concentration of its direct parents. This choice of parents may lead to significant differences between the original and the abstracted dynamics.

In this paper, we propose to use elements from information theory such as entropy and mutual information to accurately abstract the dynamics of pathway models. Rather than using a syntactic criterion to define parents, parents are selected so as to maximize the mutual information between their previous values and the current value of the variable of interest. Therefore, the temporal evolution of the individual variables is defined with respect to the past values of the maximally informative variables. Going further, one can wonder whether the values of variables are all equally important to predict the outcome of the pathway. Using again the notion of mutual information, we propose a method to identify a highly informative subset of model variables and define a low-dimensional DBN abstraction of the original system. The techniques we propose, implemented in the freely available tool DBNizer, are valid for general classes of models, including ODE, CTMC and HSD models.

To illustrate the effectiveness of our methods and algorithms, we will focus on a particularly challenging problem. Specifically, we will study a model of the TRAIL-induced apoptosis pathway for HeLa cells (Bertaux *et al.*, 2014). This system presents a stiff behavior. Indeed, the gradual activation of upstream proteins (initiator caspases) may result in a sudden activation of downstream proteins (executioner caspases) leading to cell death (Albeck *et al.*, 2008). Also cell-to-cell variability plays an important role. Even large quantities of TRAIL do not induce apoptosis in all cells (e.g. only 70% die after 8 hours for a 250 ng/ml TRAIL treatment) (Spencer *et al.*, 2009). Moreover, survivors are transiently more resistant to TRAIL (Flusberg *et al.*, 2013). Several models have been proposed to explain these observations. Sorger and colleagues proposed an ODE model focusing on signal transduction able to explain fractional killing (Spencer *et al.*, 2009). The initial state of the cell (i.e. the initial level of the signalling proteins) played a fundamental role in the death/survival outcome. This model was further extended so as to account for stochastic protein turnover, resulting in an HSD model (Bertaux *et al.*, 2014). This model was additionally able to explain reversible resistance. The HSD model is made of 58 ODEs, coupled with 17 2-variable stochastic models for protein turnover. Using optimized implementations, the simulation of the behavior of few cells (<1000) over short time durations (<1 day) is bearable. However, simulating the behavior of tissues or of small spheroids exposed over extended durations (e.g. several weeks) to repeated TRAIL treatments becomes a serious issue.

Using our strategy we obtained a low-dimensional DBN model (10 variables instead of 92) that presented a very good agreement with the original HSD model (< 1% difference in cell death probability) and whose numerical simulation is faster by an order of magnitude.

2 TRAIL-induced apoptosis in HeLa cells

TNF-Related Apoptosis Inducing-Ligand (TRAIL) is a protein that is known to induce apoptosis in cancer cells and hence has been considered as a promising choice for anti-cancer therapeutic strategies. The molecular events leading HeLa cells to die following TRAIL application are well known (see Fig. 1). It is observed that surviving cells develop a temporary resistance to TRAIL treatments over time (Flusberg *et al.*, 2013). Understanding the mechanism by which isogenic populations of cells acquire resistance to TRAIL treatment will be crucial to designing effective therapeutic strategies. Because a quantitative understanding of these processes necessitates to account for cell-to-cell variability, modeling and model analysis tools are expected to play an essential role.

In Bertaux *et al.* (2014), we proposed a model of TRAIL-induced apoptosis combining a deterministic model for signal transduction, as in the original model of (Spencer *et al.*, 2009), and stochastic models for protein turnover that capture cell-to-cell variability and its dynamics. Using this low level biochemical model, *in silico* experiments matched biological observations of fractional killing, correlated sister cell fate, and the time-dependent evolution of cell resistance induced by a TRAIL treatment. While this detailed model has been extremely useful for analyzing TRAIL induced apoptosis its analysis was tedious since single-cell simulations needed to be repeated many times. This can become challenging when one wants to

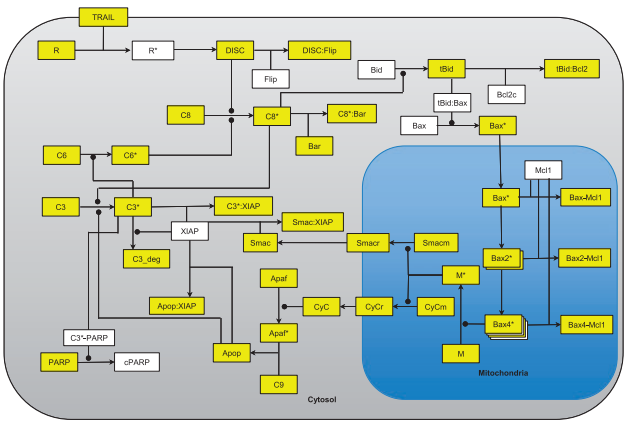

Fig. 1. Main protein species and elementary reactions of the TRAIL-induced apoptosis pathway modeled in Spencer and colleagues Spencer *et al.* (2009), which were used as the basis for signal transduction part of the HSD model (Bertaux *et al.*, 2014). Signal transduction is initiated by the binding of TRAIL to its receptor, and may subsequently lead to pores created in the mitochondria outer membrane and eventually cell death. The white nodes are those identified by our tool DBNizer as most important for cell death

analyze the system in multi-scale settings, say modeling a tumor of hundreds of thousands of cells over long time horizons for repeated treatments. Instead, we propose here to abstract this model into a (series of) high-level behavioral model(s). More efficient simulations can then be performed using the abstract model(s).

3 Information theory-based abstraction

In this section, we describe our strategy to obtain a discrete probabilistic structure that represents a system dynamics. In practice, our algorithms will take as inputs a large set of trajectories sampled at discrete time points. We will assume that this set represents all the relevant dynamics of the original system.

The key steps of our approach include the identification of the most important variables and the inference of the most informative local dependencies between them. To do so, we will rely on well-established information theory notions. As a first step, given that we are interested in a discrete abstraction of the underlying system, we will describe how to discretize the values of model variables using entropy. After this, we will explain how using mutual information our algorithm chooses a small subset of the most relevant variables of the original system and then infers a directed graph of the most important influences between them.

3.1 Entropy-based discretization

A simple and common strategy to discretize a variable is to get an estimate of the minimal (usually 0) and maximum value it can take, and to partition this range into equal sized intervals, henceforth called *uniform discretization* (see e.g. Liu *et al.*, 2011a). In our case, each variable describes the concentration level of a biochemical species, ranging from very low to very high. This scheme has therefore the advantage of being easily interpretable biologically.

The issue with using the uniform discretization is that for some variables (e.g. polymerized molecules with non-linear dynamics), most of the concentration values are condensed (e.g. with extremely small values), with rare outliers in the extremes. In such cases, the uniform discretization would bundle almost all the values together in a single discretized interval, and be almost useless.

We propose to first use *entropy* to analyze the quality of the uniform discretization. The entropy $H(X)$ of a K-valued discrete random variable X is $H(X) = -\sum_{x\in X} p(x) log_K(p(x))$.

For instance, for a variable X, if the discretization scheme perfectly splits the data such that each of the K valuations has equal probability, then its entropy is $-K * (1/K) * log_K(1/K) = 1$. In the worst case though when all the data point are concentrated in a single interval of the random variable, the entropy is 0.

We use entropy to check the effectiveness of the uniform discretization. Only for variables X with an acceptable level of entropy (we chose $H(X) \geq 0.4$ for the HSD model), we stick with the uniform discretization. Choosing other reasonable values (in $[0.25, 0.55]$) does not impact results much, see Supplementary Table S4.

For variables where uniform discretization had a low entropy (< 0.4), we resort to an alternate quantization algorithm which automatically discretizes these variables with the goal of maximizing the entropy. For this, we first sample enough simulations of the HSD model to obtain a histogram of values for each variable, over all time points. Based on these histograms, we partition the values to have an equal number of samples in each interval. Further, we use the Lloyd-Max (Lloyd, 1982; Max, 1960) discretization algorithm which minimizes the distortion (quadratic distance of the samples and their discrete values). The downside of this method is that the biological interpretation can be lost. Additionally, this increases the size of the internal representation of transition probabilities in the abstract model. These are the reasons we use it only for variables where uniform discretization results in a low entropy.

3.2 Mutual information

Mutual Information (*MI* for short), commonly used in information and probability theory, gives a quantitative measure of the correlation between two variables. Mathematically, mutual information evaluates how similar the joint distribution between two random variables compared to the product of the marginal probabilities of the individual variables. It is a convenient metric in our case for finding the set of variables most relevant globally for the dynamics, and also understanding the effective local dependencies between the variables. In formal terms, given three discrete random variables X,Y and Z, the mutual information $MI(X; Y|Z)$ between X and Y conditioned on Z is given by:

$$MI(X; Y|Z) = \sum_{z\in V_Z} \sum_{y\in V_Y} \sum_{x\in V_X} p(x, y, z)\, log\left(\frac{p(x, y, z)p(z)}{p(x, z)p(y, z)}\right)$$

where V_R is the set of values of variable $R \in \{X, Y, Z\}$, and $p(x, y, z)$ is the joint probability that $X = x$, $Y = y$ and $Z = z$.

3.3 Inferring key random variables

In high-dimensional dynamical systems, not all variables convey the same quantity of information on the dynamics. In order to define a set of important variables, we propose to use mutual information between variables and the signal of interest. In the apoptosis pathway, we are interested in the cell fate (dynamics of death). The latter is modeled as a binary variable D: it represents if the cell was alive or dead 8 hours after exposing the cell to 250 ng/ml TRAIL (death is defined by a cPARP concentration threshold of 100 000 units in the HSD model (Bertaux *et al.*, 2014)).

We first compute for each variable the mutual information between its initial configuration and the cell fate. A simple way of choosing relevant variables is to select the k variables having the highest mutual information w.r.t the signal of interest (here D). However, doing so is not necessarily a good choice. For instance, assume that variables X and Y have very similar dynamics. Hence $MI(D; X)$ and $MI(D; Y)$ are very similar (say high). However, selecting both variables $\{X, Y\}$ is not efficient as the value for (X, Y) does not bring much more information than say X alone. This can be automatically detected by considering $MI(D; Y|X)$, which would be much lower than $MI(D; Y)$. In the extreme case where $X = Y$, we have $MI(D; Y|X) = 0$.

Consequently, we adopted the following scheme. First we select the variable, say X_1, having the highest mutual information with D, $MI(D; X_1)$, and hence being the most important variable for its effect on death. Then we select the second variable, say X_2, as the one that, conditioned on X_1, has the highest mutual information with D, $MI(D; X_2|X_1)$. We continue this for k steps until the mutual information, given by $MI(D; X_k|X_1 \cdots X_{k-1})$, is sufficiently low. At this stage we stop our computation.

While this identifies variables (which have an initial value) that have a considerable impact on death, it may miss some key intermediate variables important for conveying the signal of death. We will explain in Section 4 how to find such variables.

3.4 Inferring local dependencies between key variables

A crucial step towards constructing an accurate abstraction is to find the set of important local dependencies between the identified variables. Our strategy to find those is again to rely on mutual information. Let $V = \{v_1 \ldots . v_k\}$ be the k identified variables, and $T = \{0, 1 \ldots . t-1\}$ be the discretized set of t time points.

3.4.1 Direction of local dependencies

We first build the directed graph based on the reaction network, $G_{RN} = (V_{RN}, E_{RN})$. The set of vertices of G_{RN} is the set of variables of the original system. Edges of G_{RN} are defined with respect to the reaction network of the biological system: $(X, Y) \in E_{RN}$ if the concentration of X influences the concentration of Y in some reaction, that is, if X appears in the differential equation of Y. For instance, if we have a one-way reaction that produces Z from X and Y, then X, Y and Z will be vertices, and (X, Z), (Y, Z) and (X, Y) and (Y, X) will be edges of G_{RN}. Once we have constructed the reaction network graph, we build the graph G_V^+ over the set of vertices V made of the selected variables $\{v_1 \ldots . v_k\}$. In G_V^+, we have an edge (v_i, v_j) iff v_j is reachable from v_i in G_{RN}. Stated differently, G_V^+ is the vertex-induced subgraph of the transitive closure of G_{RN}.

3.4.2 Selecting important variables

Graph G_V^+ reflects potential dependencies between selected variables. However, not all are useful (either because they are negligeable or redundant). To obtain a smaller, but still sufficiently informative set of local dependencies between variables, we define $G_{MI} = (V, E_{MI})$ by refining G_V^+ using mutual information. Denoting E_v the predecessors of $v \in V$ in G_V^+, and $Z_1^m, \ldots, Z_k^m, Y^m, V^{m+1}$ the random variables of $z_1, \ldots, z_k, y$ at time m and v at time $m+1$, we define:

$$M(Y; V|Z_1, \ldots, Z_k) = \max_{m \in T} MI(Y^m; V^{m+1}|Z_1^m, \ldots, Z_k^m)$$

For each key variable v, we select the variables influencing v iteratively, as in Section 3.3. First, we select the variable, $z_1 \in E_v$, having the highest value for $M(Z; V)$. Then we select $z_2 \in E_v$ as the variable having the highest $M(Z; V|Z_1)$, and iterate.

4 DBN formalism

We now present the mathematical model of *Dynamic Bayesian Networks* (DBNs for short), which allow us to encode the abstraction, and can be simulated in an efficient way (Liu *et al.*, 2011b; Palaniappan *et al.*, 2016).

In the DBN abstraction, first, the time domain is discretized i.e. the dynamics is assumed to be of interest only for a finite set of time points. Each node of the DBN represents the state (concentration) of a molecular species at a time point. Edges between nodes are defined by an underlying graph (G_{RN} or G_{MI} in our case; see Fig. 2). The value range of each variable V_i are discretized into a set of intervals $\mathbf{I}_i$. Entries in the *Conditional Probability Tables* (CPTs for short) are of the form $C_i^t(I|\mathbf{I}_i) = p$, saying that p is the probability of the value of V_i falling in the interval I at time t, given that the value of Z was in $\mathbf{I}_Z$ at time $t-1$ for each (Z, V_i) an edge of the underlying graph.

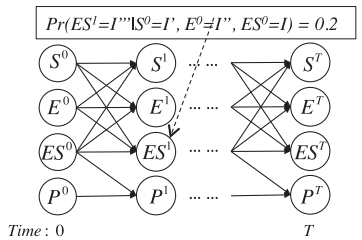

Fig. 2. A Dynamic Bayesian Network. Nodes represent discrete random variables at a time point and edges represent the local dependencies between the nodes. Each node is also associated with a conditional probability table

The size of the CPTs strongly influences performance. Indeed, the efficiency of the DBN approach (both in terms of space complexity and simulation time) scales down exponentially with the number *pa* of parents of a variable. Formally, the complexity of simulating the DBN is $O(k * |V|^{pa})$, where $|V|$ is the number of values for each variable, and k the number of variables. In order to have CPTs with not too many entries, we limit the number of parents to 4 per variable. Increasing the number of parents does not improve the accuracy much while slowing down the simulations, while decreasing it reduces the accuracy of results (Supplementary Table S1).

Our tool **DBNizer** automatically constructs the full DBN structure. First it generates a large number of trajectories of the underlying biochemical model by numerical integration of the original model. Second, it selects the suitable subset of variables to be used for DBN construction and infers their edge relationships. Third, it calculates probabilities for each entry in the CPTs through simple counting (as in Liu *et al.*, 2011a). It also automatically considers model refinement through iterative improvements (see next section). The information flow is schematically represented in Supplementary Figure S1.

There may be additional variables, not identified in Section 3.3, that are important for the transduction of signal. Adding these variables in the DBN can further improve its accuracy. For this, we iteratively find additional variables v and build an associated DBN to assess how important they are. We rank a DBN according to the weighted mean difference between the simulation outputs of the original biochemical model and the DBN. More precisely, to select an additional variable on top of a set V of variables, we rank the DBN $MIDBN_{V \cup \{v\}}$ automatically built with set of variables $V \cup \{v\}$, for every $v \notin V$. We select v optimizing the rank of $MIDBN_{V \cup \{v\}}$. We iterate from $V' = V \cup \{v\}$, and stop when no additional variable really improves the accuracy. The exact subset of variables chosen by this iterative discovery is sensitive to parameters of our tool. This is because several variables carry similar information. Indeed, different choices do not impact the results much (see Supplementary Table S5).

5 Computational results

In this section we will outline our key experimental results to compare the different models according to different metrics (time per simulation, accuracy, etc.). Unless stated otherwise, we consider treatments with 250 ng/ml of TRAIL and simulate cell behaviors for 8 hours after treatment. The concentration of *cPARP* was used as an indicator of cell death. 'Observations' (i.e. time points) were available every 2 min for the first 30 min, and every 15 min for the subsequent 7.5 hours. We used 100 000 simulations of the original HSD model to populate the CPT entries of the DBNs. Using more simulations does not improve the accuracy of the DBNs (Supplementary Table S3). All experiments were carried on a quad core 2.8 Ghz Intel Xeon E5−1603 CPU with 8 GB RAM.

We will consider DBNs obtained by our tool DBNizer, as described in the next subsection. For comparison, we consider DBN *RNDBN*, defined using the technique advocated in Liu *et al.* (2011b), where the local dependency between nodes is chosen from

Table 1. Conditional mutual information of species with respect to death decision computed in two different conditions

TRAIL = species	250 ng/ml MI	TRAIL = species	10 ng/ml MI
Bcl2c	0.33	Bcl2c	0.456
XIAP	0.023	XIAP	0.014
Flip	0.023	Bax	0.008
Bax	0.021	Mcl1	0.01
Mcl1	0.025	Smacm	0.011
Bid	0.020	Flip	0.009

Note: The identified set of most important variables does not vary much in the two conditions.

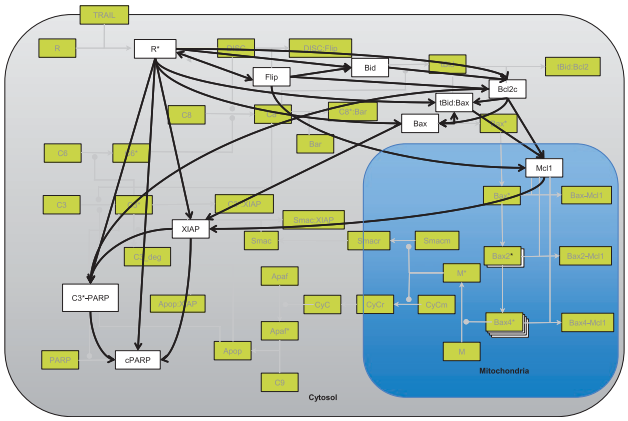

Fig. 3. Inferred connectivity network for $MIDBN_{10}$ (self connections not represented). The nodes of the DBN are *Flip*, *Bid*, *Bcl2c*, $tBid-Bax$, *Bax*, *Mcl*1, $C3-PARP$, *XIAP*, *cPARP* and activated form of *R*. Selected variables correspond either to the upper part of the pathway or to its lower part, with only one representative of mitochondrial processes (*Mcl*1)

the underlying reaction network G_{RN}. For analysis purpose, we also consider DBN $MIDBN_{58}$ with the same 58 variables as *RNDBN*, but which differs by the parent relation, defined using our mutual information based procedure. DBN simulations are performed using look-ahead simulations (Palaniappan *et al.*, 2016).

5.1 Abstractions produced by DBNizer

Using the approach described in Section 3.3, we select variables having maximal effect on cell fate. Setting a cut-off of 0.005 for conditional mutual information, we obtained 6 variables, namely by order of importance *Bcl*2*c*, *XIAP*, *Flip*, *Bax*, *Mcl*1 and *Bid* (Table 1). The species that are not considered, have an initial concentration with almost no impact on the cell fate. In addition to these 6 variables, we also added *cPARP* as it is the marker for cell death. This set of variables is used to define and compute a mutual information based DBN, $MIDBN_7$.

To test the robustness of our variable selection scheme, we reiterated the computations for a significantly different amount of TRAIL, namely 10 ng/ml. The key native variables did not vary much: only the least informative variable, *Bid*, is replaced by *Smacm* (Table 1).

Following the procedure described at the end of Section 4, the complexes $tBid-Bax$, $C8-Bid$ and activated-C3 were iteratively added, resulting in DBNs of increasing size, $MIDBN_8$, $MIDBN_9$ and $MIDBN_{10}$. The procedure stopped at 10 variables since adding any other variable to $MIDBN_{10}$ did not improve significantly the discerning power of the DBN.

Table 2. Quality and efficiency of the abstractions

Model	Cell death (HSD: 69.9%)	Discerning power (HSD: 100%)	Time/1000 simulations (HSD: 56s)
$MIDBN_7$	70.43%	96.14%	2.13s (26.3X)
$MIDBN_8$	69.57%	96.31%	2.64s (21.21X)
$MIDBN_9$	69.33%	96.37%	2.98s (18.8X)
$MIDBN_{10}$	69.03%	96.84%	3.30s (17X)
$MIDBN_{58}$	66.85%	94.12%	73.05s
RNDBN	92.29%	85.53%	299s

Note: All mutual information based DBNs show good results for quality as measured by percentage of cell death and discerning power (see text for their definition), with $MIDBN_{10}$ providing the best results. All compact DBNs show good performance as measured by simulation time with at least a ten times speedup with respect to the reference HSD model.

The variables considered by $MIDBN_{10}$ are depicted in white in Figure 1. The associated network of causalities (parent relation) computed automatically for $MIDBN_{10}$ is represented on Figure 3. This network has several interesting features. First, one could be surprised by the fact that the activated initiator Caspase8 ($C8^*$) does not appear. The DBN does not need the concentration of $C8^*$ explicitly as it can be fairly well evaluated using *Flip* and R^*. The same goes for Bax^*, generally considered a critical player for apoptosis decision, which can be fairly well evaluated using $tBid-Bax$, *Bax* and *Mcl*1. The level of free *Mcl*1 is therefore more informative on the cell fate than the level of Bax^*. One can hypothesize that this comes from the fact that free *Mcl*1 is able to efficiently sequester recently produced active *Bax*, making its level a measure of the cell's resistance to apoptosis induction. More generally, the set of reactions taking place into the mitochondrion is barely represented. The direct inhibitor *XIAP* of the executioner caspase $C3^*$ is in our abstraction directly influenced by $tBid-Bax$ and *Mcl*1 (in addition to $R*$; Fig. 3). This strongly suggests that the mitochondrion acts as a black box with a fast (given the timescale of DBNs) and relatively simple input/output function.

5.2 Quality and efficiency of the abstractions

In this section, we evaluate the different abstractions produced. Ideally, behaviors predicted using the original or an abstract model should match. However, because the original model is stochastic (and the abstract one too), such a direct comparison is not possible. A first, global measure of quality is given by the comparison of the predicted percentage of cell death. The original HSD model predicts that nearly 70% of cells die. Abstract models should predict similar values (see Table 2, second column). Moreover, the timing of death, that is, the distribution of death times, should be similar (see Fig. 4).

A more refined measure of abstraction quality is provided by the discerning power. The probability that a cell dies depends on its initial state, that is, the initial concentrations of the proteins involved in signal transduction. Indeed, it has been observed that applying TRAIL to two sister cells just after division results in highly correlated fates (dead or alive) of the two cells (Spencer *et al.*, 2009). We say that a model has a good discerning power if for many different initial conditions, it is able to predict the death probability obtained with the original model for the same initial conditions (see Fig. 5 for an illustration). Note that it is a more stringent criterion than the overall death percentage. In practice, we ran 200 000 simulations of the HSD model, and record in each case the fate of the cell (dead or alive) together with its initial configuration defined as the discrete value of the initial concentrations of the 6 selected key proteins whose initial

configurations have an impact on the cell fate. For instance, the configuration *XIAP* : *low*, *Bid* : *high*, *Bcl2c* : *verylow*, *Bax* : *high*, *Mcl*1 : *low*, *Flip* : *verylow* (configuration 120210) was highly represented (2628 samples) and highly associated to cell death (99% probability), whereas the configuration *XIAP* : *high*, *Bid* : *low*, *Bcl2c* : *veryhigh*, *Bax* : *low*, *Mcl*1 : *high*, *Flip* : *verylow* (configuration 213 120; 809 samples) leads to cell death in only 9% of the simulations. For each initial configuration, we compute the difference between the predictions by the HSD model and the DBN abstraction weighted by the percentage of occurrence of this profile in the HSD simulations. For statistical reasons, we focused on the 60 most frequent configurations that together represent 50% of the 200 000 simulations. Any single such configuration is represented in at least 700 simulations. The discerning power is then defined as 100% minus this weighted error (see Table 2, second column).

Regarding the efficiency of the simulation, we assess the time needed to run 1000 simulations of the original HSD model and of its various abstractions, $MIDBN_7$, $MIDBN_8$, $MIDBN_9$, $MIDBN_{10}$, and for reference $MIDBN_{58}$ and *RNDBN*. All these information are provided in Table 2 (last column).

As represented in Figure 4, Supplementary Figure S2, and summarized in Table 2 (first column), all DBNs using mutual information provide good to very good descriptions of the dynamics of cell death. This is in sharp contrast to the reaction network based DBN, *RNDBN*. The comparison of *RNDBN* and $MIDBN_{58}$ (Supplementary Fig. S2), having both 58 nodes, clearly shows that the critical feature is to have a proper parent relation. As expected, the comparison of DBNs with 7–10 variables (Table 2 and Supplementary Fig. S2) shows that adding more variables improves accuracy. However, the performance of $MIDBN_{58}$ is slightly worse than that of $MIDBN_7$, indicating that in probabilistic representations there might be a trade off between the capacity to store information (favoring large DBNs) and to reuse it (favoring small ones).

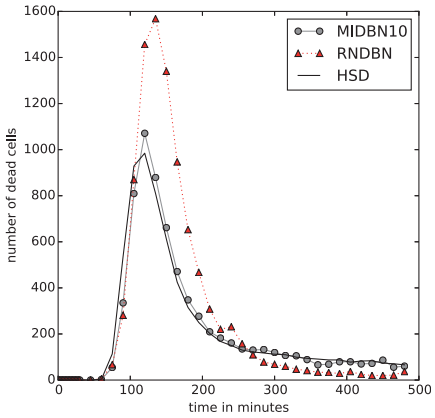

Fig. 4. Distribution of the time of death during a TRAIL treatment as predicted by the reference *HSD* model and two abstractions, *RNDBN* and $MIDBN_{10}$. The death distribution of $MIDBN_{10}$ very closely follows that of the *HSD* model with only a marginal error at the peak value. *RNDBN* on the other hand significantly overestimates the number of cell death during the period 100–200 min and slightly underestimates it later on. Results for all DBNs are provided in Supplementary Figure S2

Similar results are found for the discerning power (Table 2 (second column)). All MI based DBNs have a $> 94\%$ discerning power, in sharp contrast to *RNDBN* ($< 86\%$). To better analyze the low performance of *RNDBN*, we represent the predicted percentage of cell death in different initial configurations (Fig. 5). We observe that irrespective of the initial configuration, *RNDBN* predicts a constant high death rate ($> 80\%$): Influences between variables are not well captured by *RNDBN* for this challenging dynamical system (high dimension, strong non-linearities).

The analysis of simulation times (Table 2, last column) shows that the performance of abstraction depends strongly on their size. Large DBNs, *RNDBN* and $MIDBN_{58}$, are actually slower to simulate than (an optimized implementation of) the original HSD model. All compact DBNs however show good performance, being at least 17 times faster than the HSD model (for $MIDBN_{10}$) and up to 26 times faster (for $MIDBN_7$). Experiments run for treatment with 10 ng/ml TRAIL display very similar results (see Supplementary Figs S7 and S8).

In summary, using MI based DBNs is essential to obtain abstractions of good quality; and using low-dimensional DBNs is essential to obtain efficient abstractions. $MIDBN_7$ to $MIDBN_{10}$ present both advantages, with slightly different trade-offs.

5.3 Importance of MI-based abstractions

In the previous section, we found that the approach used to define local dependencies between variables has a critical impact on the abstraction quality. To better understand why this choice is so important, we focus in this section on how the parent relation is represented in *RNDBN* and in $MIDBN_{58}$ for one particular complex, namely $M^* - Smacm$.

An excerpt of the network is represented in more detail in Figure 6. Because of the reaction forming $M^* - Smacm$, the parents of $M^* - Smacm$ in *RNDBN* are M^*, *Smacm* and $M^* - Smacm$. In

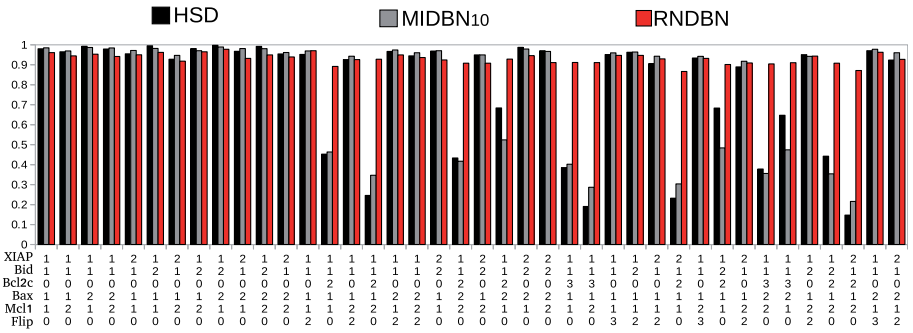

Fig. 5. Dead cell percentage for various initial configurations as predicted by *HSD*, $MIDBN_{10}$ and *RNDBN*. As expected, death probability is maximal when pro-apoptotic proteins *Bid* and *Bax* are high and anti-apoptotic proteins, *XIAP*, *Bcl*2*c*, *Mcl*1 and *Flip*, are low (*e.g.* 100% in initial configuration 120210). The converse holds as well (*e.g.* 15% in initial configuration 212120) $MIDBN_{10}$ accurately follows the dynamics of the original model, in accordance with its good discerning power. In contrast the cell death percentage for *RNDBN* does not vary much over all initial configurations. Results for all DBNs are provided in Supplementary Figure S3

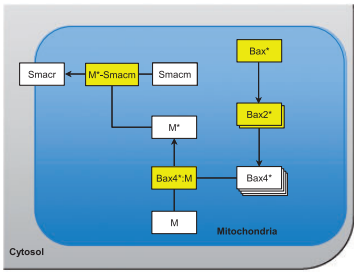

Fig. 6. Excerpt of the apoptosis reaction network focusing on the complex $M^* - Smacm$ and mitochondrial reactions. For the structure-based *RNDBN*, parents of $M^* - Smacm$ are $M^* - Smacm$, $Smacm$ and M^*. For the MI-based $MIDBN_{58}$, parents are $M^* - Smacm$, Bax^*, $Bax2^*$ and $Bax4^* - M$, depicted in yellow

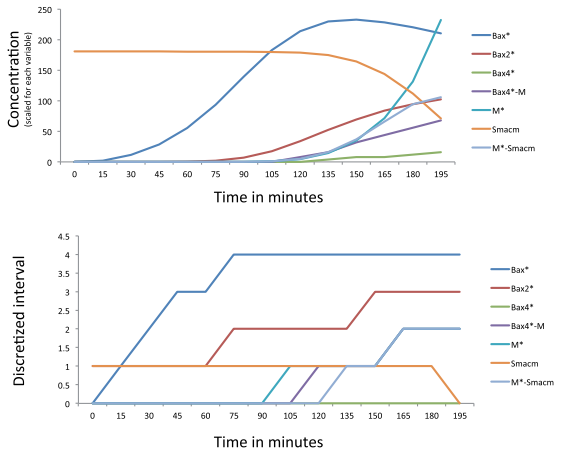

Fig. 7. Temporal evolution of the concentration (top) and its discretized value (bottom) for selected variables of one simulation of the apoptosis pathway. During the time interval 150-165, $Bax4^* - M$, M^* and $Smacm - M^*$ go simultaneously from level 1 to level 2. Using these fast evolving variables is not as informative as using variables Bax^*, $Bax2^*$ evolving more slowly

$MIDBN_{58}$, Bax^*, $Bax2^*$ and $Bax4^* - M$ have been chosen as parents by our MI-based approach, in addition to $M^* - Smacm$ itself (depicted in white in Fig. 6). We verified that these parents allow to predict the level of $M^* - Smacm$ (15 min later) better than M^*, $Smacm$ and $M^* - Smacm$ (see Supplementary Fig. S5).

We conjecture that the speed of reactions from $Bax4^*$ to $M^* - Smacm$ is faster than the timestep of the DBN, while Bax^* and $Bax2^*$ have a more gradual evolution. This seems confirmed considering a trajectory of the system, as depicted in Figure 7. Because of causality, one would expect that a significant increase of reactants in a reaction would cause a significant increase of the products. After discretization, this should typically lead to two successive threshold crossing events, reactants being followed by products. However, if reactions are fast it may often be the case that the two threshold crossings happen during the same time interval and therefore appear simultaneously after time discretization, thereby loosing causality and mutual dependence between the current values of parents and the future value of the variable to predict. This can be observed in Figure 7 where both M^* and $M^* - Smacm$ cross their threshold during the interval 150–165 min. Defining parent relations based on the reaction network might therefore not be appropriate for systems showing fast and slow dynamics (referred to as 'snap-action' in Albeck *et al.*, 2008) as is the case for apoptosis.

6 Conclusions and perspectives

In this paper we have discussed how we can abstract the dynamics of a biological pathway into a discrete-time stochastic model, namely a DBN, using information theory measures. Specifically, we proposed a new strategy to automatically infer the structure of small DBNs and demonstrate their accuracy and efficiency. On the first aspect, our abstractions are able to represent the fraction of dead cells and the distribution of death times as described by the original HSD model (the mismatch is of the order of 1%). On the second aspect, the DBN abstractions enjoy an order of magnitude faster simulations in comparison with the original model. Besides pure simulation, the inferred structure is also informative regarding effective dependencies between variables and could typically be of use when selecting variables for experimental measurements. Lastly, our abstraction procedure is general, provided that a sufficient diversity of input profiles and initial conditions have been used to train the DBNs. We are working on a multi-scale model where cells affect their environment in a physical way exclusively (altering diffusion).

Funding

This work was partially supported by ANR projects STOCH-MC (ANR-13-BS02-0011-01) and Iceberg (ANR-IABI-3096).

Conflict of Interest: none declared.

References

Albeck,J.G. *et al.* (2008) Modeling a snap-action, variable-delay switch controlling extrinsic cell death. *PLoS Biol.*, **6**, 2831–2852.

Bertaux,F. *et al.* (2014) Modeling dynamics of cell-to-cell variability in TRAIL-induced apoptosis explains fractional killing and predicts reversible resistance. *PLoS Comput. Biol.*, **10**, 14.

Feret,J. *et al.* (2009) Internal coarse-graining of molecular systems. *PNAS*, **106**, 6453–6458.

Flusberg,D.A. *et al.* (2013) Cells surviving fractional killing by trail exhibit transient but sustainable resistance and inflammatory phenotypes. *Mol. Biol. Cell*, **24**, 2186–2200.

Gillespie,C.S. (2009) Moment-closure approximations for mass-action models. *IET Syst. Biol.*, **3**, 52–58.

Gunawardena,J. (2014) Time-scale separation – Michaelis and Menten's old idea, still bearing fruit. *FEBS J.*, **281**, 473–488.

Liu,B. *et al.* (2011a) A computational and experimental study of the regulatory mechanisms of the complement system. *PLoS Comput. Biol.*, **7**.

Liu,B. *et al.* (2011b) Probabilistic approximations of odes based bio-pathway dynamics. *Theor. Comput. Sci.*, **412**, 2188–2206.

Lloyd,S. (1982) Least squares quantization in PCM. *IEEE T. Inf. Theory*, **28**, 129–137.

Max,J. (1960) Quantizing for minimum distortion. *IEEE T. Inf. Theory*, **6**, 7–12.

Michaelides,M. *et al.* (2016) Property-driven state-space coarsening for continuous time Markov chains. In: *QEST* **9826**, 3–18.

Munsky,B. and Khammash,M. (2006) The finite state projection algorithm for the solution of the chemical master equation. *J. Chem. Phys.*, **124**, 44–104.

Palaniappan,S.K. *et al.* (2016) A look-ahead simulation algorithm for dbn models of biochemical pathways. In: *HSB*, **9957**, 3–19.

Radulescu,O. *et al.* (2015) Model reduction of biochemical reactions networks by tropical analysis methods. *Math. Model Nat. Phenom.*, **10**, 124–138.

Spencer,S.L. *et al.* (2009) Non-genetic origins of cell-to-cell variability in trail-induced apoptosis. *Nature*, **459**, 428–432.