Hajiramezanali *et al. BMC Genomics* 2019, **20**(Suppl 6):435
https://doi.org/10.1186/s12864-019-5720-3

BMC Genomics

RESEARCH Open Access

Scalable optimal Bayesian classification of single-cell trajectories under regulatory model uncertainty

Ehsan Hajiramezanali, Mahdi Imani, Ulisses Braga-Neto, Xiaoning Qian* and Edward R. Dougherty

From Selected original research articles from the Fifth International Workshop on Computational Network Biology: Modeling, Analysis and Control (CNB-MAC 2018): Genomics
Washington, D.C., USA. 29 August 2018

Abstract

Background: Single-cell gene expression measurements offer opportunities in deriving mechanistic understanding of complex diseases, including cancer. However, due to the complex regulatory machinery of the cell, gene regulatory network (GRN) model inference based on such data still manifests significant uncertainty.

Results: The goal of this paper is to develop optimal classification of single-cell trajectories accounting for potential model uncertainty. Partially-observed Boolean dynamical systems (POBDS) are used for modeling gene regulatory networks observed through noisy gene-expression data. We derive the exact optimal Bayesian classifier (OBC) for binary classification of single-cell trajectories. The application of the OBC becomes impractical for large GRNs, due to computational and memory requirements. To address this, we introduce a particle-based single-cell classification method that is highly scalable for large GRNs with much lower complexity than the optimal solution.

Conclusion: The performance of the proposed particle-based method is demonstrated through numerical experiments using a POBDS model of the well-known T-cell large granular lymphocyte (T-LGL) leukemia network with noisy time-series gene-expression data.

Keywords: Optimal Bayesian classification, Single-cell trajectory classification, Particle filter, Probabilistic Boolean networks

Background

A key issue in genomic signal processing is to classify normal versus cancerous cells, different stages of tumor development, or different prospective drug response. Previous gene-expression technologies, such as microarray and RNA-Seq, typically measure the average behavior of tens of thousands of cells [1, 2]. By contrast, the recent advances in next-generation sequencing technologies have allowed in-depth investigation of the transcriptome at a single-cell resolution [3, 4].

Gene regulatory networks (GRNs) govern the functioning of key cellular processes, such as stress response, DNA repair, and other mechanisms involved in complex diseases such as cancer. Often, the relationship among genes can be described by logical rules updated at discrete time intervals with each gene have Boolean states: 0 (OFF) or 1 (ON) [5]. The Partially-Observed Boolean dynamical system (POBDS) model [6–8] is a rich framework for modeling the behavior of GRNs observed through contemporary gene-expression technologies, as it allows indirect and incomplete observation of gene states. Several tools for the POBDS model have been developed in recent years, such as the optimal filter and smoother based on the Minimum Mean Square Error (MMSE) criterion, termed as the Boolean Kalman filter (BKF) and Boolean Kalman smoother (BKS) [6], respectively.

*Correspondence: xqian@ece.tamu.edu
Department of Electrical and Computer Engineering, Texas A&M University, MS3128 TAMU, 77843 College Station, TX, USA

In [9] and [10], the maximum-likelihood (ML) based classification of single-cell trajectories has been developed. The method uses the ML-adaptive filter proposed in [6] for estimation of the unknown parameters, followed by the Bayes classifier tuned to the ML parameter estimates. The drawback of this method is its inability to use prior knowledge in deriving the classifier. In [11], the intrinsically Bayesian robust (IBR) classifier for the trajectories is developed. This IBR classifier is optimal relative to the prior distribution of unknown parameters.

In this paper, assuming that there are two classes, healthy ($c = 0$) and cancerous ($c = 1$), we derive the optimal Bayesian classifier (OBC) [12, 13] for classification of single-cell trajectories. The difference between the OBC and IBR classifiers is that in the OBC the expectation of the class-conditional densities is taken over the posterior distribution of the unknown parameters [14, 15], whereas in the IBR classifier the expectation is taken over the priors.

Despite the optimality of the developed OBC for single-cell trajectories, its exact computation for large GRNs becomes intractable, due to the large size of the matrices involved. In this paper, we develop a particle-based OBC to scale up the classification of single-cell trajectories. The proposed method contains a bank of Auxiliary Particle-Filter implementations of the Boolean Kalman Filter (APF-BKF) proposed in [16], for both training and test processes.

Our contributions are twofold: 1) Optimal Bayesian Classification: we derive the optimal Bayesian classifiers (OBC) for both single-cell gene expression trajectories and multiple-cell averaged gene expression with uncertain regulatory network prior; and 2) Scalability: the parallel particle filters together with the Monte-Carlo inference have been efficiently used to estimate the likelihood and stationary distributions, making the derived OBC scalable to larger gene regulatory networks.

We apply the APF-BKF-based OBC to classify trajectories of the blood cancer T-cell large granular lymphocyte (T-LGL) leukemia. T-LGL leukemia is a chronic disease characterized by a clonal proliferation of cytotoxic T cells [17]. A Boolean network model of T cell survival signaling in the context of T-LGL leukemia has been constructed by [18] through performing extensive literature search. Then the T-LGL network has been simplified by [17], which constructs the minimum network that preserves the attractor structure of that system. The reduced network contains 18 genes, which has an optimal solution with a transition matrix with $2^{2\times 18} = 68,719,476,736$ elements. By contrast, as we will show in our numerical experiments, the proposed APF-BKF-based method captures T-cell dynamics with only 1000 particles.

Methods

Gene regulatory network model

Gene regulatory networks are modeled as partially-observed Boolean dynamical systems (POBDS). The two components of the POBDS model are a state space model that describes the evolution of the dynamics of the GRN, and an observation model for the measurements. These two components are described below.

GRN state space model

The state process $\{\mathbf{X}_k; k = 0, 1, \ldots\}$, where $\mathbf{X}_k \in \{0,1\}^n$, represents the activation (ON)/inactivation (OFF) state for the corresponding gene across time. The state at each discrete time is assumed to be updated through the nonlinear signal model

$$\mathbf{X}_k = \mathbf{f}\left(\mathbf{X}_{k-1}\right) \oplus \mathbf{n}_k, \tag{1}$$

for $k = 1, 2, \ldots$, where $\mathbf{f} : \{0,1\}^n \to \{0,1\}^n$ is a Boolean function called the network function, "$\oplus$" indicates component-wise modulo-2 addition, and $\mathbf{n}_k \in \{0,1\}^n$ is Boolean transition noise at time k. The modulo-2 addition means that if a bit in the noise $\mathbf{n}_k$ is 1, the value of the corresponding gene in the Boolean state $\mathbf{X}_k$ is flipped. In this paper, the noise process $\mathbf{n}_k$ is assumed to have independent components distributed as Bernoulli(p), where the parameter $p > 0$ models the noise "intensity" — the closer p is to 0.5, the more chaotic the system will be, while a value of p close to zero means that the state trajectories are nearly deterministic, being governed tightly by the network function and perturbation process. The network possesses a steady-state distribution π^∞ describing its long-run behavior as:

$$\pi^\infty = \lim_{k\to\infty} \left[P\left(\mathbf{X}_k = \mathbf{x}^i\right), \ldots, P\left(\mathbf{X}_k = \mathbf{x}^{2^n}\right)\right]^T, \tag{2}$$

where $\left\{\mathbf{x}^1, \ldots, \mathbf{x}^{2^n}\right\}$ denotes the set of the corresponding network states in the Boolean vector representation.

Observation model

The data available to the experimenter is described by the observation process $\{\mathbf{Y}_k; k = 1, 2, \ldots\}$, where $\mathbf{Y}_k = (\mathbf{Y}_k(1), \ldots, \mathbf{Y}_k(n))$ is a vector containing the transcript abundance measurements at time k, for $k = 1, 2, \ldots$. We consider a Gaussian linear model

$$\mathbf{Y}_k = \boldsymbol{\lambda} + D\mathbf{X}_k + \mathbf{v}_k,\ k = 1, 2, \ldots, \tag{3}$$

where $\mathbf{v}_k \sim \mathcal{N}(0, \sigma^2 I_n)$ is an uncorrelated zero-mean Gaussian noise vector, $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_n]^T$ is a vector of baseline gene expressions corresponding to the "zero" state for each gene, and $D = \mathrm{Diag}(\delta_1, \ldots, \delta_n)$ is a diagonal matrix containing differential expression values for each gene along the diagonal (these indicate by how much the "one" state of each gene is overexpressed over the "zero" state). Such a Gaussian linear model is an appropriate model for single-cell gene-expression data [19, 20].

Optimal Bayesian classifier (OBC) for single-cell trajectories
Assume there are two POBDSs corresponding to the healthy and cancerous (mutated) classes, each having n genes. The difference between the healthy and mutated classes could be the over-expression or disruption of a value of single or multiple genes in the mutated case. Let $\mathbb{Y}^c = \left\{\mathcal{Y}_c^{(1)}, \mathcal{Y}_c^{(2)}, \ldots, \mathcal{Y}_c^{(D_c)}\right\}$ be the set of D_c observed trajectories from class c, $c = 0, 1$. Let $\Theta = (\theta_1, \ldots, \theta_M)$ be the uncertainty set of M network functions containing the unknown true network functions in (1), indicating M possible Boolean functions as $\left\{\mathbf{f}_{\theta_1}^c, \ldots, \mathbf{f}_{\theta_M}^c\right\}$ considering the regulatory model uncertainty for the class c. The prior probability of the model θ for the class c is represented by $\pi(\theta \mid c)$, where $\sum_{i=1}^{M} \pi(\theta_i \mid c) = 1$, for $c = 0, 1$. This uncertainty could arise due to some unknown regulations (i.e. interactions) between some genes in the pathway diagram (more information in "Results and discussion" section). We wish to derive the optimal Bayesian classifier (OBC) under uncertainty using all available data and prior knowledge.

If the feature-label distribution is unknown but belongs to an uncertainty class Θ of feature-label distributions, then we desire a classifier to minimize the expected error over the uncertainty class. This expected error is equivalent to the Bayesian minimum mean-square-error estimate [21] given by $\hat{\epsilon}(\psi) = E_{\theta|\mathbb{Y}^c,c}[\epsilon(\psi, \theta)]$, where $\epsilon(\psi, \theta)$ is the error of ψ on the feature-label distribution parameterized by θ and the expectation is taken over the posterior distribution of parameters $\pi(\theta \mid \mathbb{Y}^c, c)$, $c = 0, 1$.

For a given test trajectory $\mathcal{Y}$, the OBC, minimizing the Bayesian minimum mean-square error estimate, can be obtained as

$$\psi_{\mathrm{OBC}}(\mathcal{Y}) = \begin{cases} 0 \text{ if } p^0 E_{\theta|\mathbb{Y}^0,c=0}[p_\theta(\mathcal{Y} \mid c=0)] \geq \\ \qquad (1-p^0)\, E_{\theta|\mathbb{Y}^1,c=1}\left[p_\theta(\mathcal{Y} \mid c=1)\right]. \\ 1 \text{ otherwise} \end{cases} \tag{4}$$

where $p_\theta(.)$ denotes a probability density function corresponding to parameter θ, p^0 is the prior probability of class 0 and

$$E_{\theta|\mathbb{Y}^c,c}[p_\theta(\mathcal{Y} \mid c)] = \sum_{\theta\in\Theta} p_\theta(\mathcal{Y} \mid c)\, \pi(\theta \mid \mathbb{Y}^c, c), \tag{5}$$

for $c = 0, 1$.

Derivation of (5) requires computing the posterior distribution

$$\pi(\theta \mid \mathbb{Y}^c, c) = \frac{p_\theta(\mathbb{Y}^c \mid c)\pi(\theta \mid c)}{\sum_{\theta'\in\Theta} p_{\theta'}(\mathbb{Y}^c \mid c)\pi(\theta' \mid c)}, \tag{6}$$

where $\pi(\theta \mid c)$ denotes the prior probability of the corresponding network model θ for class c. For an arbitrary trajectory $\tilde{\mathcal{Y}}$, we define the log-likelihood function associated to model θ and class c by

$$L_c^\theta(\tilde{\mathcal{Y}}) := \log p_\theta(\tilde{\mathcal{Y}} \mid c). \tag{7}$$

Now, using the above definition in (4)-(6) leads to the following exact OBC solution:

$$\psi_{\mathrm{OBC}}(\mathcal{Y}) = \begin{cases} 0 \text{ if } p^0 \tau_\Theta^0(\mathcal{Y}) \geq (1-p^0)\tau_\Theta^1(\mathcal{Y}) \\ 1 \text{ otherwise} \end{cases}, \tag{8}$$

where

$$\begin{aligned} \tau_\Theta^c(\mathcal{Y}) &= E_{\theta|\mathbb{Y}^c,c}\left[p_\theta(\mathcal{Y} \mid c)\right] \\ &= \sum_{\theta\in\Theta} \pi(\theta \mid \mathbb{Y}^c, c)\, p_\theta(\mathcal{Y} \mid c) \\ &= \sum_{\theta\in\Theta} \frac{p_\theta(\mathbb{Y}^c|c)\pi(\theta|c)}{\sum_{\theta'\in\Theta} p_{\theta'}(\mathbb{Y}^c|c)\pi(\theta'|c)} p_\theta(\mathcal{Y} \mid c) \\ &= \sum_{\theta\in\Theta} \frac{\exp\left(\sum_{d=1}^{D_c} L_c^\theta\left(\mathcal{Y}_c^{(d)}\right)\right)\pi(\theta|c)}{\sum_{\theta'\in\Theta} \exp\left(\sum_{d=1}^{D_c} L_c^{\theta'}\left(\mathcal{Y}_c^{(d)}\right)\right)\pi(\theta'|c)} \\ &\qquad \times \exp(L_c^\theta(\mathcal{Y})). \end{aligned} \tag{9}$$

The expectation in (9) is taken with respect to the posterior distribution of θ, i.e. $\pi(\theta|\mathbb{Y}^c, c)$, as opposed to IBR classifier [11] that considers the prior distribution $\pi(\theta|c)$. Furthermore, $\log p_\theta(\mathbb{Y}^c \mid c) = \sum_{d=1}^{D_c} \log p\left(\mathcal{Y}_c^{(d)} \mid \theta, c\right)$ is used in (9) due to the independency of the training trajectories.

As shown in Eq. (8), the OBC requires computing the log-likelihood functions of all training trajectories and test trajectory for both classes and $\theta \in \Theta$. Let $\left\{\mathbf{x}^1, \ldots, \mathbf{x}^{2^n}\right\}$ denote the corresponding network states in the Boolean vector representation and $\mathbf{Y}_{1:T} = (\mathbf{Y}_1, \ldots, \mathbf{Y}_T)$ be a single trajectory of length T. The log-likelihood function can be computed as

$$\begin{aligned} L_c^\theta(\mathbf{Y}_{1:T}) &= \log p_\theta(\mathbf{Y}_{1:T} \mid c) \\ &= \log p_\theta(\mathbf{Y}_1 \mid c) + \sum_{k=2}^{T} \log p_\theta(\mathbf{Y}_k \mid \mathbf{Y}_{1:k-1}, c), \end{aligned} \tag{10}$$

where based on the POBDS model,

$$\begin{aligned} &p_\theta(\mathbf{Y}_k \mid \mathbf{Y}_{k-1}, c) \\ &\quad = \sum_{i=1}^{2^n} p_\theta\left(\mathbf{Y}_k \mid \mathbf{X}_k = \mathbf{x}^i, c\right) P_\theta\left(\mathbf{X}_k = \mathbf{x}^i \mid \mathbf{Y}_{1:k-1}, c\right) \\ &\quad = \sum_{i=1}^{2^n} p_\theta\left(\mathbf{Y}_k \mid \mathbf{X}_k = \mathbf{x}^i, c\right) \\ &\qquad \times \sum_{j=1}^{2^n} P_\theta(\mathbf{X}_k = \mathbf{x}^i \mid \mathbf{X}_{k-1} = \mathbf{x}^j, c) \\ &\qquad \times P_\theta(\mathbf{X}_{k-1} = \mathbf{x}^j \mid \mathbf{Y}_{1:k-1}, c). \end{aligned} \tag{11}$$

Define the conditional probability of any network state at time step k for the model θ and class c as

$$\mathbf{\Pi}_{k|k}^{\theta,c} = \left[P_\theta\left(\mathbf{X}_k = \mathbf{x}^1 \mid \mathbf{Y}_{1:k}, c\right), \ldots, P_\theta\left(\mathbf{X}_k = \mathbf{x}^{2^n} \mid \mathbf{Y}_{1:k}, c\right)\right]^T. \tag{12}$$

Using (11) and (12), the log-likelihood function in (10) can be written in a compact form as [22]

$$L_c^{\theta}(\mathbf{Y}_{1:T}) = \sum_{k=1}^{T} \log \left\| T_k^{\theta,c}(\mathbf{Y}_k) M_k^{\theta,c} \Pi^{\theta,c} \right\|_1, \tag{13}$$

where $||.||_1$ denotes L_1 norm, indicating the summation in (10). $M_k^{\theta,c}$ is the transition matrix of the Markov state process corresponding to model $\theta \in \Theta$, with entries given by:

$$\begin{aligned} \left(M_k^{\theta,c}\right)_{ij} &= P_\theta\left(\mathbf{X}_k = \mathbf{x}^i \mid \mathbf{X}_{k-1} = \mathbf{x}^j, c\right) \\ &= p^{||\mathbf{f}_\theta^c(\mathbf{x}^j) \oplus \mathbf{x}^i||_1} (1-p)^{n-||\mathbf{f}_\theta^c(\mathbf{x}^j) \oplus \mathbf{x}^i||_1}, \\ &\qquad i,j = 1,\ldots,2^n, \end{aligned} \tag{14}$$

with p denoting the Bernoulli noise parameter. $T_k^{\theta,c}(\mathbf{Y}_k)$ is a diagonal matrix, called the update matrix, with the ith diagonal element given by

$$\begin{aligned} &\left(T_k^{\theta,c}(\mathbf{Y}_k)\right)_{ii} = p_\theta(\mathbf{Y}_k|\mathbf{X}_k = \mathbf{x}^i, c) \\ &= \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^n \exp\left(-\sum_{j=1}^{n} \frac{\left(\mathbf{Y}_k(j) - \lambda_j - \delta_j \mathbf{x}^i(j)\right)^2}{2\sigma^2}\right), \end{aligned} \tag{15}$$

for $i = 1,\ldots,2^n$, where δ_j and λ_j are gene-expression parameters associated to the jth gene in class c as defined in (3). Notice that the initial distribution $\Pi_{0|0}^{\theta,c} = \pi_{\theta,c}^{\infty}$ is the steady-state distribution associated to model θ and the class c defined as

$$\pi_{\theta,c}^{\infty} = \lim_{k\to\infty} \left[P_\theta\left(\mathbf{X}_k = \mathbf{x}^1|c\right), \ldots, P_\theta\left(\mathbf{X}_k = \mathbf{x}^{2^n}|c\right)\right]^T. \tag{16}$$

This vector can be either computed exactly as introduced in [9] or approximated by creating multiple Monte-Carlo trajectories with relatively long horizons.

The posterior distribution can also be recursively computed according to the transition and update matrices as described in [6]:

$$\Pi^{\theta,c}k|k = \frac{T_k^{\theta,c}(\mathbf{Y}_k)\, M_k^{\theta,c} \Pi^{\theta,c}}{\left\| T_k^{\theta,c}(\mathbf{Y}_k) M_k^{\theta,c} \Pi^{\theta,c} \right\|_1}, \quad k = 1,2,\ldots. \tag{17}$$

The complexity of computing the log-likelihood function for a single trajectory of length T is of order $O(2^{2n} \times T)$ due to the transition matrix involved in its computation. The whole process of the proposed OBC is presented in Algorithm 1.

Scalable classification of single-cell trajectories

In the previous section, the exact solution for the optimal Bayesian classifier is introduced. However, for large systems with a large number of state variables, the exact computation of Algorithm 1 becomes impractical. This is due to the large transition matrix with 2^{2n} elements required to compute the log-likelihoods, leading to exponential computational and memory complexity. Thus, the key here is to scale up the OBC for single-cell trajectories by reducing both computational and memory complexity when computing (10).

We adopt the Sequential Monte-Carlo (SMC) techniques [23–27] for estimating nonlinear state-space models, such as our POBDS here. These techniques approximate the target distribution using sample points ("particles") drawn from a proposal distribution, taking advantage of the fact that sampling from the proposal distribution is easier than from the target. This helps alleviate the high computation of the exact filter by using a finite set of Monte-Carlo samples. In this paper, we use the Auxiliary Particle Filter implementation of the Boolean Kalman Filter (APF-BKF) proposed in [16] to deal with large GRNs.

Let $\mathbf{Y}_{1:T}$ be a given trajectory and the goal is to approximate the likelihood function in Eq. (10) (i.e., $L_c^{\theta}(\mathbf{Y}_{1:T})$ for class $c \in \{0,1\}$ and $\theta \in \Theta$). Let there be N total particles $\{\mathbf{x}_{k-1,i}\}_{i=1}^N$, with their associated weights $\{w_{k-1,i}\}_{i=1}^N$. The particle filter allows us to produce an approximation for the elements in $\mathbf{\Pi}_{k|k}^{\theta,c}$ of (12), simply by using the discrete support of the particles

$$\begin{aligned} &P_\theta(\mathbf{X}_k \,|\, \mathbf{Y}_{1:k}, c) \\ &\quad \propto\; p_\theta(\mathbf{Y}_k \,|\, \mathbf{X}_k, c) \sum_{i=1}^{N} P_\theta(\mathbf{X}_k \,|\, \mathbf{x}_{k-1,i}, c)\, w_{k-1,i}. \end{aligned} \tag{18}$$

Usually only a few particles have significant weights after a few iterations of the algorithm and most particles have negligible weights. APF-BKF is a look-ahead method that predicts the location of particles with a high probability at time k based on the observations at time step $k-1$, with the purpose of making the subsequent resampling step more efficient. Without the look-ahead, the basic algorithm blindly propagates all particles, even those in low probability regimes.

The APF-BKF algorithm defines:

$$\begin{aligned} &P_\theta(\mathbf{X}_k, \zeta_k \,|\, \mathbf{Y}_{1:k}, c) \\ &\quad \propto\; p_\theta(\mathbf{Y}_k \,|\, \mathbf{X}_k, c)\, P_\theta(\mathbf{X}_k \,|\, \mathbf{x}_{k-1,\zeta_k}, c)\, w_{k-1,\zeta_k}, \end{aligned} \tag{19}$$

where $\zeta_k = 1,\ldots,N$, is an index of the mixture in (18). If we draw from the joint density and then discard the index, then we will have a sample from (18) as required. This procedure carries out the prediction and update steps of the optimal filter in (17). We can approximate (19) by

$$\begin{aligned} &P_\theta(\mathbf{X}_k, \zeta_k \,|\, \mathbf{Y}_{1:k}, c) \\ &\quad \propto\; p_\theta(\mathbf{Y}_k \,|\, \mu_{k,\zeta_k}, c)\, P_\theta(\mathbf{X}_k \,|\, \mathbf{x}_{k-1,\zeta_k}, c)\, w_{k-1,\zeta_k}, \end{aligned} \tag{20}$$

Algorithm 1 Optimal Bayesian Classification of Single-Cell Trajectories

Training Process

Compute the steady-state distribution $\mathbf{\Pi}_{0|0}^{\theta,c} = \pi_\theta^\infty$, $\theta \in \Theta, c = 0, 1$ [9].
for $d \in \{1, \ldots, D_c\}$ **do**
  for $c \in \{0, 1\}$ **do**
    for $\theta \in \Theta$ **do**
      $L_c^\theta\left(\mathcal{Y}_c^{(d)}\right) \leftarrow$ **Log-Lik** $\left(\theta, c, \mathcal{Y}_c^{(d)}, \pi_\theta^\infty\right)$
    end for
  end for
end for

Test Process

for $c \in \{0, 1\}$ **do**
  for $\theta \in \Theta$ **do**
    $L_c^\theta(\mathcal{Y}) \leftarrow$ **Log-Lik** $\left(\theta, c, \mathcal{Y}, \Pi_\theta^\infty\right)$
  end for
  $\tau_\Theta^c(\mathcal{Y}) = \sum_{\theta\in\Theta} \frac{\exp\left(\sum_{d=1}^{D_c} L_c^\theta\left(\mathcal{Y}_c^{(d)}\right)\right) \pi(\theta|c) \exp\left(L_c^\theta(\mathcal{Y})\right)}{\sum_{\theta'\in\Theta} \exp\left(\sum_{d=1}^{D_c} L_c^{\theta'}\left(\mathcal{Y}_c^{(d)}\right)\right) \pi(\theta'|c)}$
end for
$\psi_{\mathrm{OBC}}(\mathcal{Y}) = \begin{cases} 0 \text{ if } p^0\, \tau_\Theta^0(\mathcal{Y}) \geq (1-p^0)\, \tau_\Theta^1(\mathcal{Y}) \\ 1 \text{ otherwise} \end{cases}$.

Log-Lik $(\theta, c, \mathbf{Y}_{1:T}, \pi_{0|0})$

$L_c^\theta = 0$.
for $k = 1, 2, \ldots,$ **do**
  $\boldsymbol{\beta}_k^{\theta,c} = T_k^{\theta,c}(\mathbf{Y}_k)\, M_k^{\theta,c}\, \mathbf{\Pi}_{k-1|k-1}^{\theta,c}$.
  $\mathbf{\Pi}_{k|k} = \boldsymbol{\beta}_k^{\theta,c} / ||\boldsymbol{\beta}_k^{\theta,c}||_1$.
  $L_c^\theta = L_c^\theta + \log||\boldsymbol{\beta}_k^{\theta,c}||_1$.
end for
Return (L_c^θ)

where $\mu_{k,i}$ is the mode associated with the density of $P_\theta(\mathbf{X}_k \mid \mathbf{x}_{k-1,i}, c)$, given by [16]:

$$\mu_{k,i} = \mathrm{Mode}[\mathbf{X}_k \mid \mathbf{x}_{k-1,i}, \theta, c] = \mathbf{f}_\theta^c(\mathbf{x}_{k-1,i}), \qquad (21)$$

for $i = 1, \ldots, N$, where we have used (1) and the fact that the noise is zero-mode (i.e., the Bernoulli noise intensity p is smaller than 0.5).

By simulating the index with the probability $\nu_{k,i} = p_\theta(\mathbf{Y}_k \mid \mu_{k,i}, c)\, w_{k-1,i}$, we can sample from $P_\theta(\mathbf{X}_k, \zeta_k \mid \mathbf{Y}_{1:k}, c)$ and then sample from the transition density given the mixture, $P_\theta(\mathbf{X}_k \mid \mathbf{x}_{k-1,i}, c)$.

Actually, we simulate only from particles associated with large predictive likelihoods. We first sample N times from the joint density of $P_\theta(\mathbf{X}_k, \zeta_k \mid \mathbf{Y}_{1:k}, c)$, and then obtain the new particles $\{\mathbf{x}_{k,i}\}_{i=1}^N$ and their associated weights $\{w_{k,i}\}_{i=1}^N$ by

$$\begin{aligned} \mathbf{x}_{k,i} &= \mu_{k,\zeta_{k,i}} \oplus \mathbf{n}_{k,i} \sim P_\theta(\mathbf{X}_k \mid \mathbf{x}_{k-1,\zeta_{k,i}}, c), \\ w_{k,i} &= \frac{p_\theta(\mathbf{Y}_k \mid \mathbf{x}_{k,i}, c)}{p_\theta(\mathbf{Y}_k \mid \mu_{k,\zeta_{k,i}}, c)}, \quad i = 1, \ldots, N. \end{aligned} \qquad (22)$$

The auxiliary variables $\{\zeta_{k,i}\}_{i=1}^N$ are obtained by sampling from a discrete distribution:

$$\{\zeta_{k,i}\}_{i=1}^N \sim \mathrm{Cat}\left(\{\nu_{k,i}\}_{i=1}^N\right), \qquad (23)$$

where $\mathrm{Cat}(a_1, \ldots, a_N)$ represents a categorical distribution with the probability mass function $f(\zeta = i) = a_i / \sum_{j=1}^N a_j$.

It is shown in [16, 28] that the log-likelihood function in (10) can be approximated by:

$$\begin{aligned} L_c^\theta(\mathbf{Y}_{1:T}) &= p_\theta(\mathbf{Y}_1 \mid c) + \sum_{k=2}^T \log p_\theta(\mathbf{Y}_k \mid \mathbf{Y}_{1:k-1}, c) \\ &\approx \sum_{k=1}^T \log\left[\left(\frac{1}{N}\sum_{i=1}^N \nu_{k,i}\right)\left(\frac{1}{N}\sum_{i=1}^N w_{k,i}\right)\right] \\ &\triangleq \hat{L}_c^\theta(\mathbf{Y}_{1:T}), \end{aligned} \qquad (24)$$

where $\hat{L}_c^\theta(\mathbf{Y}_{1:T})$ denotes the approximation of $L_c^\theta(\mathbf{Y}_{1:T})$. Note that the computational complexity of this algorithm is of order $O(NT)$ which can be much smaller than $O(2^{2n}T)$ that is the complexity of computing the exact log-likelihood function in (10).

The whole process and the schematic diagram of the proposed classifier are presented in Algorithm 2 and Fig. 1, respectively. During the training process, $2MD_0D_1$ particle filters need to be run for computing the log-likelihood functions of trajectories from all network models in the uncertainty class for two classes. The output values of the particle filters can be used for efficient approximation of the posterior distribution in (9). Then, during the test process, for a given test trajectory, $2M$ particle filters need to be performed for log-likelihood approximation of all network models ($\theta \in \Theta$) and classes ($c = 0, 1$). These log-likelihood values and posterior probability approximated during the training process can be used to derive the approximate OBC in (8).

The training process of the proposed method has the computational complexity $O(2NMTD_0D_1)$, whereas the exact solution has the complexity $O(2^{2n+1}MTD_0\, D_1)$. The exponential growth of the complexity with the size of network (i.e., number of genes) for the exact solution precludes its application to large GRNs. However, the number of particles, N, by the proposed method can be chosen relatively small according to the attractor structure of

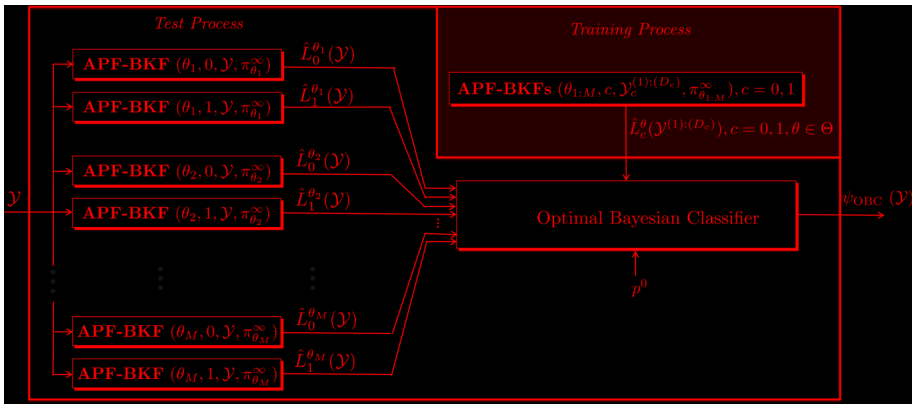

Fig. 1 The schematic diagram of the proposed method

the system (i.e., $N << 2^{2n}$) [29], allowing the classification of large-scale single-cell trajectories (see "Results and discussion" section). The complexity of the test process for the proposed method is $O(2NMT)$, as opposed to $O(2^{2n+1}MT)$ for the optimal solution.

Optimal Bayesian classifier for multiple-cell scenarios

In the previous sections, the classification of single-cell trajectories is discussed. Here, we consider common scenarios in molecular biology research where gene-expression data are often based on the average expression from multiple-cells at different time with different states. Since the trajectories are assumed to be independent and drawn based on the dynamics of the true network, its steady-state distribution $\pi^{\infty}_{\theta^*}$ characterizes the probability of the system being at different states. It can be shown that the multiple-cell data are independent samples from the following measurement model [10]:

$$\mathbf{Y} \sim \mathcal{N}\left(\boldsymbol{\lambda} + \sum_{i=1}^{2^n} D\mathbf{x}^i \pi^{\infty}_{\theta^*,c}(i), \sigma^2\right), \tag{25}$$

where $\{\mathbf{x}^1, \ldots, \mathbf{x}^{2^n}\}$ denotes all the network states in the Boolean vector representation. However, we assume that the true network model θ^*, is unknown, and is represented by a finite set of M possible network models $\{\theta_1, \ldots, \theta_M\}$ with prior probability $\pi(\theta \mid c)$. Let $\mathbf{Y}_c^{(1)}, \ldots, \mathbf{Y}_c^{(N_c)}$ be the multiple-cell training measurements available for class c. The optimal Bayesian classifier for a given test sample $\mathbf{Y}$ can be represented by

$$\psi_{\mathrm{OBC}}(\mathbf{Y}) = \begin{cases} 0 \text{ if } p^0 E_{\theta|\mathbf{Y}_0^{(1)},\ldots,\mathbf{Y}_0^{(N_0)},c=0}[p_\theta(\mathbf{Y} \mid c=0)] \\ \quad \geq (1-p^0) \\ \quad \times E_{\theta|\mathbf{Y}_1^{(1)},\ldots,\mathbf{Y}_1^{(N_1)},c=1}\left[p_\theta(\mathbf{Y} \mid c=1)\right]. \\ 1 \text{ otherwise} \end{cases} \tag{26}$$

The posterior probability of the parameter θ can be computed as

$$\begin{aligned} &P\left(\theta \mid \mathbf{Y}_c^{(1)}, \ldots, \mathbf{Y}_c^{(N_c)}\right) \\ &\quad = \frac{p_\theta(\mathbf{Y}_c^{(1)}, \ldots, \mathbf{Y}_c^{(N_c)} \mid c)\,\pi(\theta \mid c)}{\sum_{\theta' \in \Theta} p_{\theta'}(\mathbf{Y}_c^{(1)}, \ldots, \mathbf{Y}_c^{(N_c)} \mid c)\,\pi(\theta' \mid c)} \\ &\quad = \frac{\left(\prod_{d=1}^{N_c} p_\theta\left(\mathbf{Y}_c^{(d)} \mid c\right)\right)\pi(\theta \mid c)}{\sum_{\theta' \in \Theta}\left(\prod_{d=1}^{N_c} p_{\theta'}\left(\mathbf{Y}_c^{(d)} \mid c\right)\right)\pi(\theta' \mid c)}. \end{aligned} \tag{27}$$

Computation of (26) requires computing the conditional probability of training and test samples given all $\theta \in \Theta$ and $c \in \{0, 1\}$. Let $\mathbf{A}$ be an $n \times 2^n$ matrix containing all Boolean states of the system (i.e., $\mathbf{A} = [\mathbf{x}^1, \ldots, \mathbf{x}^{2^n}]$). According to (25), the conditional distribution of an arbitrary sample $\tilde{\mathbf{Y}}$ given class c and network model θ can be written as

$$\begin{aligned} \tilde{\mathbf{Y}} \mid c, \theta \sim &(1 - \mathbf{A}\,\pi^{\infty}_{\theta,c}) \circ \mathcal{N}(\boldsymbol{\lambda}, \sigma^2 \mathbf{I}_n) \\ &+ (\mathbf{A}\,\pi^{\infty}_{\theta,c}) \circ \mathcal{N}(\boldsymbol{\lambda} + D\mathbf{1}_n, \sigma^2 \mathbf{I}_n). \end{aligned} \tag{28}$$

where "$\circ$" is the Hadamard product. Thus, the Boolean nature of the state vector suggests that each element of the multiple-cell measurement is distributed as a mixture of two Gaussian distributions. Replacing (27) and (28) into (26) leads to the OBC for multiple-cell scenarios. Comprehensive comparison results between the OBC in single-cell trajectories and multiple-cells are provided in the next section.

Results and discussion

We evaluate the proposed single-cell trajectory classifier and compare its performance with the OBC based on multiple-cell average expression on the T-LGL leukemia Boolean network, whose GRN [17] is shown in Fig. 2. This GRN has 18 genes. The regulating functions are defined in Table 1 [17, 18]. The main node is "Apoptosis", which denotes programmed cell death. According to [17],

Algorithm 2 Scalable Classification of Single-Cell Trajectories

Training Process

Approximate the steady-state distribution, $\pi_\theta^\infty, \theta \in \Theta$ by Monte-Carlo simulations.
for $c \in \{0,1\}$ **do**
  for $d \in \{1,\dots,D_c\}$ **do**
    for $\theta \in \Theta$ **do**
      $\hat{L}_c^\theta(\mathcal{Y}_c^{(d)}) \leftarrow$ **APF-BKF** $\left(\theta, c, \mathcal{Y}_c^{(d)}, \pi_\theta^\infty\right)$
    end for
  end for
end for

Test Process

for $c \in \{0,1\}$ **do**
  for $\theta \in \Theta$ **do**
    $\hat{L}_c^\theta(\mathcal{Y}) \leftarrow$ **APF-BKF** $(\theta, c, \mathcal{Y}, \pi_\theta^\infty)$
  end for
  $\tau_\Theta^c(\mathcal{Y}) = \sum_{\theta\in\Theta} \frac{\exp\left(\sum_{d=1}^{D_c} \hat{L}_c^\theta\left(\mathcal{Y}_c^{(d)}\right)\right)\pi(\theta|c)\exp(\hat{L}_c^\theta(\mathcal{Y}))}{\sum_{\theta'\in\Theta}\exp\left(\sum_{d=1}^{D_c}\hat{L}_c^{\theta'}\left(\mathcal{Y}_c^{(d)}\right)\right)\pi(\theta'|c)}$
end for
$\psi_{\mathrm{OBC}}(\mathcal{Y}) = \begin{cases} 0 \text{ if } p^0\,\tau_\Theta^0(\mathcal{Y}) \geq (1-p^0)\,\tau_\Theta^1(\mathcal{Y}) \\ 1 \text{ otherwise} \end{cases}$.

APF-BKF $(\theta, c, \mathbf{Y}_{1:T}, \pi_{0|0})$ [16]

$\hat{L}_c^\theta = 0, \mathbf{x}_{0,i} \sim \pi_{0|0}, w_{0,i} = 1/N$, for $i = 1,\dots,N$.
for $k = 1, 2, \dots,$ **do**
  $\mu_{k,i} = \mathbf{f}_\theta^c(\mathbf{x}_{k-1,i}), i = 1,\dots,N.$
  $\nu_{k,i} = p_\theta(\mathbf{Y}_k \mid \mu_{k,i}, c)\, w_{k-1,i}, i = 1,\dots,N.$
  $\{\zeta_{k,i}\}_{i=1}^N \sim \mathrm{Cat}\left(\{\nu_{k,i}\}_{i=1}^N\right).$
  $\mathbf{x}_{k,i} = \mu_{k,\zeta_{k,i}} \oplus \mathbf{n}_{k,i}, i = 1,\dots,N.$
  $w_{k,i} = \frac{p_\theta(\mathbf{Y}_k|\mathbf{x}_{k,i},c)}{p_\theta(\mathbf{Y}_k|\mu_{k,\zeta_{k,i}},c)}, i = 1,\dots,N.$
  $\hat{L}_c^\theta = \hat{L}_c^\theta + \log\left[\left(\frac{1}{N}\sum_{i=1}^N \nu_{k,i}\right)\left(\frac{1}{N}\sum_{i=1}^N w_{k,i}\right)\right].$
  $w_{k,i} = w_{k,i}/\sum_{j=1}^N w_{k,j},\ i = 1,\dots,N.$
end for
Return $(\hat{L}_c^\theta)$

in the mutated case, the node Apoptosis is stuck at OFF state and cannot be activated. As a result, we derive the healthy Boolean network from Table 1 and for the mutated Boolean network we put the value of Apoptosis in Table 1 to zero. This means that in the cancerous scenario, the value of Apoptosis does not obey the regulating functions and is always zero.

As we may not know the true network function, we consider four candidate network functions for each of the healthy and mutated networks as the uncertainty class of possible GRN models. In addition to the true network, we remove the operation ¬ of Apoptosis for the genes sFas and GPCR, which are intermediate nodes. Therefore, this network is very close to the true network. For the third network, we remove ¬ of Apoptosis from two other nodes IAP and P2. In the fourth network of this uncertainty class, we change the operation AND to OR for the gene BID. In this study, we use the observation models described in Eqs. (3) and (25) for the single-cell trajectory and multiple-cell averaging, respectively.

Figure 3a and b show the classification errors for the simulated data based on the T-LGL leukemia Boolean network versus the number of time points T for two values of state perturbation probability $p = 0.05$ and 0.1, respectively. For the sake of simplicity, we assume the gene-expression parameters in Eqs. (3) and (25) to be the same for all genes, that are $\boldsymbol{\lambda} = [\lambda,\dots,\lambda]^T$ and $D = \mathrm{Diag}(\delta,\dots,\delta)$, and set $\lambda = 10$, and $\delta = 30$. Two different values are considered for the observation noise level: $\sigma = 20$ (low noise), and $\sigma = 25$ (high noise). While σ corresponds to both within subject and between subject variations in the single-cell, it shows between subject variation in the multiple-cell because multiple-cells would allow to average out the within subject variance. We set the number of training trajectories $D = D^{c=0} + D^{c=1} = 4$. In both figures, the error curves are monotonically decreasing in terms of the trajectory length T for the single-cell classifier. There is a special case in which the error gets fixed after some T. This may be explained by the effect from the steady-state distributions depending on the lengths of attractor cycles of the networks under study. When the perturbation noise p and the observation noise σ are small, the sufficient T to achieve the least possible error is $L + 1$, where L is the minimum attractor length in the two networks. More precisely, the BNps tend to the deterministic BNs when the perturbation noise is small, meaning that the observations occur only in the attractor states and circulate inside the attractor cycles. In such a case, $L + 1$ is the maximum length of a trajectory that can help distinguish the two networks. But there is a nonnegligible probability of jumping states in the considerable perturbation scenarios, so that longer trajectories can be helpful. In all figures for every value of T and p, the error increases with increasing observation noise. While the proposed classifier works in both low- and high-noise scenarios, the classifier based on the multiple-cell expression data only works well in the low-noise scenarios and is very sensitive to σ. In low observation noise scenarios and when $p = 0.05$, multiple-cell classifier can easily

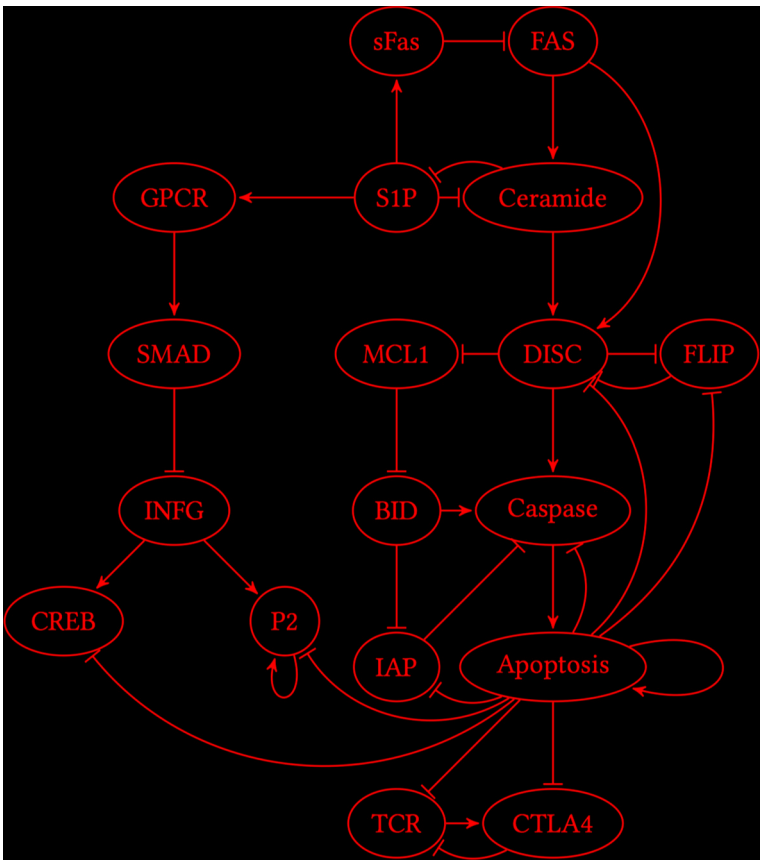

Fig. 2 T-LGL leukemia gene regulatory network

classify, while the trajectory-based with two time-points will have a little bit higher error due to the common short segments of the trajectories between two classes. When there are at least four time-points, which is the attractor cycle size in the uncertainty class of the networks, the performance of the classifiers based on single-cell trajectories is better compared to the ones using multiple-cell even in the low-noise scenarios. Increasing the number of time-points help better decipher the difference in single-cell trajectories between two classes (healthy vs. cancerous) with improved classification accuracy. Compared to the multiple-cell classifier based on averaged gene expression over cells at different states, the trajectory based classifier can clearly improve the classification performance. Even using four time points, the classification accuracy can be improved up to 8%. Using longer trajectories, the improvement can be up to 18% as indicated in Fig. 3.

In addition to the effect of trajectory length, we would like to investigate how the number of training trajectories affect the performance of the proposed method, especially with a low number of training samples. Figure 4 shows the effect of the number of training trajectories D on the

Table 1 Definitions of Boolean functions for the T-LGL leukemia Boolean network with 18 nodes [17, 18]

Node	Regulating function
CTLA4	TCR $\wedge$ $\neg$ Apoptosis
TCR	$\neg$ (CTLA4 $\vee$ Apoptosis)
CREB	IFNG $\wedge$ $\neg$ Apoptosis
IFNG	$\neg$ (SMAD $\vee$ P2 $\vee$ Apoptosis)
P2	(IFNG $\vee$ P2) $\wedge$ $\neg$ Apoptosis
GPCR	S1P $\wedge$ $\neg$ Apoptosis
SMAD	GPCR $\wedge$ $\neg$ Apoptosis
Fas	$\neg$ (sFas $\vee$ Apoptosis)
sFas	S1P $\wedge$ $\neg$ Apoptosis
Ceramide	Fas $\wedge$ $\neg$ (S1P or Apoptosis)
DISC	(Ceramide $\vee$ (Fas $\wedge$ $\neg$ FLIP)) $\wedge$ $\neg$ Apoptosis
Caspase	((BID $\wedge$ $\neg$ IAP) $\vee$ DISC) $\wedge$ $\neg$ Apoptosis
FLIP	$\neg$ (DISC $\vee$ Apoptosis)
BID	$\neg$ (MCL1 $\vee$ Apoptosis)
IAP	$\neg$ (BID $\vee$ Apoptosis)
MCL1	$\neg$ (DISC $\vee$ Apoptosis)
S1P	$\neg$ (Ceramide $\vee$ Apoptosis)
Apoptosis	Caspase $\vee$ Apoptosis

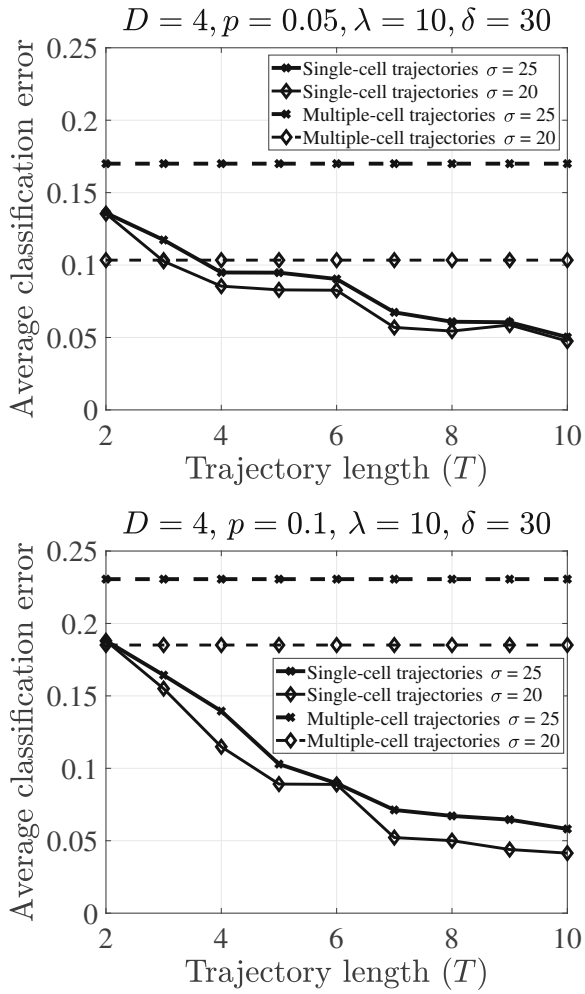

Fig. 3 Classification errors using the trajectory and multiple-cell classifiers in the T-LGL leukemia Boolean network. **a** The parameter is $p = 0.05$, **b** The parameter is $p = 0.1$

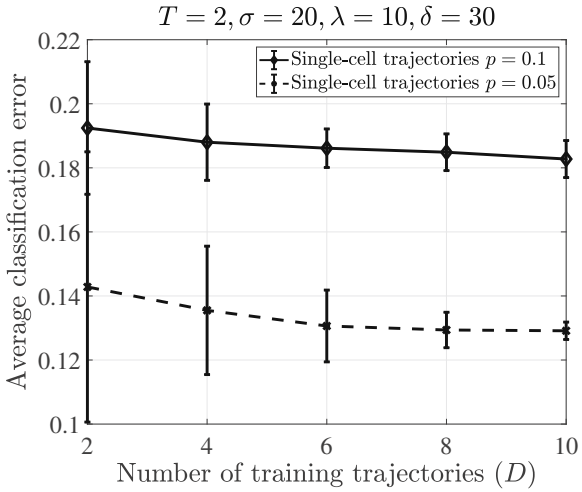

Fig. 4 Classification error of the single-cell classifier versus training sample size for the T-LGL leukemia Boolean network

classification performance. In the particle filter point of view, increasing D by 1 means increasing the available data as T. Therefore, we set $T = 2$, that is smallest T, to better see the trend of classification error. Both the average error and its standard deviation decrease with more training trajectories and the classification error converges to a fixed value when D becomes large enough. The value of D required for a converged error rate depends on the parameters T, p, and M. In real-world scenarios that may have significant uncertainty and the perturbation probability is high, we need more training data to improve the performance.

Figure 5 illustrates how the model uncertainty may affect the classification performance. In our setup, the uncertainty is manifested as the size of the network uncertainty class – the number of uncertain networks for each class. In both perturbation probabilities, the error increases with increasing M. To demonstrate how the proposed method can reduce the computational cost of the Boolean network classification, we check the change of the average classification error with respect to the number of particles. Figure 6 shows that increasing the number of particles monotonically decreases the classification error and the average error converges with only 1000 particles. This can be compared with the traditional method [9], which needs computation based on the $2^{18} \times 2^{18}$ transition

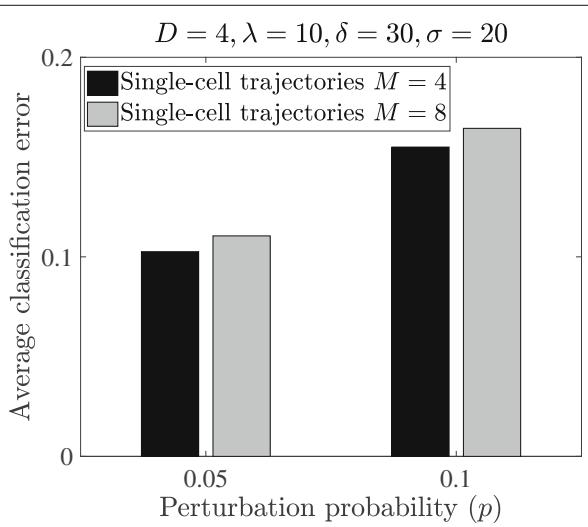

Fig. 5 Classification errors using the single trajectory classifier in the T-LGL leukemia Boolean network versus the number of uncertain networks

probability matrix. Such a dimension is too large for the direct application of the OBC.

To more comprehensively evaluate the proposed method, we have compared it with two other classification methods, i.e. IRB [11] and Plug-In [10]. The performance comparisons under four different scenarios are provided in Table 2. The OBC stands out as the best performing method in different scenarios. Using OBC improves the accuracy by 8.5% and 1.5% with $T = 3$ for $p = 0.1$ and $\sigma = 25$ compared to Plug-In and IRB, respectively.

The proposed method does not have any restriction on the noise distribution assumptions due to the generalizability of particle filters. To show this, we also test our method with different noise distributions. While the noise of GRNs is usually Gaussian, sometimes due to up regulation, noise can be Poisson or Negative Binomial (NB). We have simulated Gaussian and NB noise distributions with the same mean and variance while for Poisson noise, the variance is equal to its mean (Details can be found in Additional file 1, the additional experimental results). Additional file 1: Figure S1 shows that our method performs consistently well for all observation noises. The Poisson results are superior to the other models as its variance is smaller.

We also compare the performance of APF with the plain Sequential Importance Resampling (SIR) in high process noise. As Additional file 1: Figure S2 shows, the performance is similar especially when there is enough time points. When the number of time points is low, the SIR-based particle filter performs worse. The superior performance of APF in real-world GRNs is due to the fact that the size of attractors is usually small and the predicted mode values by APF can be a good approximation for the next prediction. Moreover, the initial distribution is assumed to be from the stationary distribution. This makes APF a more desirable approximation solutions due to the lower diversity in the particles.

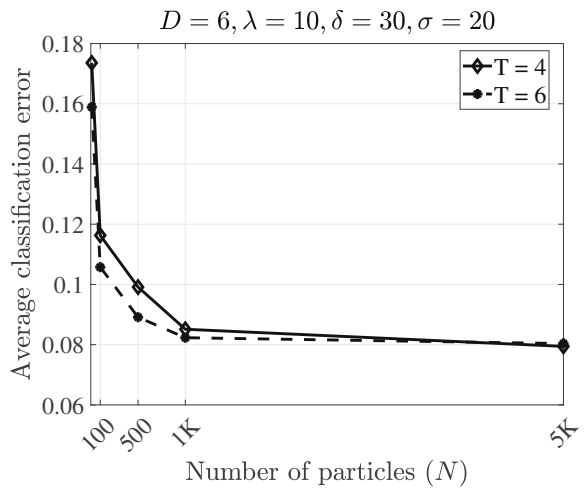

Fig. 6 Classification errors using the trajectory based classifier in the T-LGL leukemia Boolean network versus the number of particles

Table 2 Trajectory based classification results for high-noise scenario ($\sigma = 25$)

	($p = 0.05$)		($p = 0.1$)	
Method	$T = 3$	$T = 7$	$T = 3$	$T = 7$
Plug-In	0.1777	0.0922	0.2524	0.1774
IBR	0.1384	0.0723	0.1800	0.0750
OBC	**0.1173**	**0.0674**	**0.1643**	**0.0646**

The achieved best accuracy is highlighted in boldface

Conclusions

In this paper, we have developed the optimal Bayesian classifier for binary classification of single-cell trajectories under regulatory model uncertainty. The partially-observed Boolean dynamical system is used for modeling the dynamical behavior of gene regulatory networks. Due to the intractability of the OBC for large GRNs, we have proposed a particle filtering technique for approximating the OBC. This particle-based solution reduces the computational and memory complexity of the optimal solution significantly. The performance of the proposed particle-based method is demonstrated through numerical experiments using a POBDS model of the well-known T-cell large granular lymphocyte (T-LGL) leukemia network based on noisy time-series gene-expression data.

Additional file

Additional file 1: This additional file contains the additional experiment results. (PDF 229 kb)

Abbreviations

APF-BKF: Auxiliary particle-filter implementations of the Boolean Kalman filter; BKF: Boolean Kalman filter; BKS: Boolean Kalman smoother; BNp: Boolean network with perturbation; GRN: Gene regulatory network; IRB: Intrinsically Bayesian robust; MMSE: Minimum mean square error; OBC: Optimal Bayesian classifier; POBDS: Partially observed Boolean dynamical system; SMC: Sequential Monte-Carlo; T-LGL: T-cell large granular lymphocyte

Acknowledgment

We thank Texas A&M High Performance Research Computing and Texas Advanced Computing Center for providing computational resources to perform experiments in this work.

Funding

This research has been supported in part by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Mathematical Multifaceted Integrated Capability Centers (MMICCS) program, under award number DE-SC0019393, NSF Awards CCF-1553281, CCF-1718513 and CCF-1718924, and a grant from the Brookhaven National Laboratory (BNL). Publication costs are funded by DE-SC0019393.

Availability of data and materials

Not applicable.

About this supplement
This article has been published as part of *BMC Genomics Volume 20 Supplement 6, 2019: Selected original research articles from the Fifth International Workshop on Computational Network Biology: Modeling, Analysis and Control (CNB-MAC 2018): Genomics*. The full contents of the supplement are available online at https://bmcgenomics.biomedcentral.com/articles/supplements/volume-20-supplement-6.

Authors' contributions
E. H. developed OBC for the single-cell trajectory classification, developed the scalable OBC, performed the experiments, and wrote the manuscript. M. I. wrote part of the code and the first draft, and in conjunction with U. B. N. structured the APF-BKF by integrating their previous partially-observed Boolean dynamical system into this new framework. E. R. D. and X.Q. oversaw the project, proposed the new scalable OBC, and wrote the manuscript. All authors have read and approved the final manuscript.

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Publisher's Note

Published: 13 June 2019

References

1. Hajiramezanali E., Dadaneh S. Z., de Figueiredo P., Sze S.-H., Zhou M., Qian X. Differential expression analysis of dynamical sequencing count data with a gamma markov chain. 2018. arXiv preprint arXiv:1803.02527.
2. Hajiramezanali E., Dadaneh S. Z., Karbalayghareh A., Zhou M., Qian X. Bayesian multi-domain learning for cancer subtype discovery from next-generation sequencing count data. In Advances in Neural Information Processing Systems 31. 20189132–41.
3. Lake B. B., Chen S., Sos B. C., Fan J., Kaeser G. E., Yung Y. C., Duong T. E., Gao D., Chun J., Kharchenko P. V., et al. Integrative single-cell analysis of transcriptional and epigenetic states in the human adult brain. Nat Biotechnol. 2018;36(1):70.
4. Fiers M. W., Minnoye L., Aibar S., Bravo González-Blas C., Kalender Atak Z., Aerts S. Mapping gene regulatory networks from single-cell omics data. Brief Funct Genom. 2018.
5. Shmulevich I., Dougherty E. R. Genomic Signal Processing vol. 50. New Jersey: Princeton University Press; 2014.
6. Imani M., Braga-Neto U. Maximum-likelihood adaptive filter for partially-observed Boolean dynamical systems. IEEE Trans Sig Process. 2017;65(2):359–71.
7. Imani M., Braga-Neto U. M. Point-based methodology to monitor and control gene regulatory networks via noisy measurements. IEEE Trans Control Syst Technol. 2018. https://doi.org/10.1109/TCST.2017.2789191.
8. Imani M., Braga-Neto U. Finite-horizon LQR controller for partially-observed Boolean dynamical systems. Automatica. 2018;95:172–9.
9. Karbalayghareh A., Braga-Neto U., Hua J., Dougherty E. R. Classification of state trajectories in gene regulatory networks. IEEE/ACM IEEE Trans Control Syst Technol Biol Bioinforma. 2016;15:68–82.
10. Karbalayghareh A., Braga-Neto U., Dougherty E. R. Classification of single-cell gene expression trajectories from incomplete and noisy data. IEEE/ACM IEEE Trans Control Syst Technol Biol Bioinforma. 2017;16:193–207.
11. Karbalayghareh A., Braga-Neto U., Dougherty E. R. Intrinsically Bayesian robust classifier for single-cell gene expression trajectories in gene regulatory networks. BMC Syst Biol. 2018;12(3):23.
12. Dalton L. A., Dougherty E. R. Optimal classifiers with minimum expected error within a Bayesian framework—Part I: Discrete and Gaussian models. Pattern Recog. 2013;46(5):1301–14.
13. Dalton L. A., Dougherty E. R. Intrinsically optimal Bayesian robust filtering. IEEE Trans Sig Process. 2014;62(3):657–70.
14. Boluki S., Esfahani M. S., Qian X., Dougherty E. R. Incorporating biological prior knowledge for Bayesian learning via maximal knowledge-driven information priors. BMC Bioinformatics. 2017;18(14):552.
15. Boluki S., Esfahani M. S., Qian X., Dougherty E. R. Constructing pathway-based priors within a gaussian mixture model for Bayesian regression and classification. IEEE/ACM Trans Comput Biol Bioinforma. 2017;16:524–537.
16. Imani M., Braga-Neto U. Particle filters for partially-observed Boolean dynamical systems. Automatica. 2018;87:238–50.
17. Saadatpour A., Wang R.-S., Liao A., Liu X., Loughran T. P., Albert I., Albert R. Dynamical and structural analysis of a T cell survival network identifies novel candidate therapeutic targets for large granular lymphocyte leukemia. PLoS Comput Biol. 2011;7(11):1002267.
18. Zhang R., Shah M. V., Yang J., Nyland S. B., Liu X., Yun J. K., Albert R., Loughran T. P. Network model of survival signaling in large granular lymphocyte leukemia. Proc Natl Acad Sci. 2008;105(42):16308–13.
19. Wu A. R., Neff N. F., Kalisky T., Dalerba P., Treutlein B., Rothenberg M. E., Mburu F. M., Mantalas G. L., Sim S., Clarke M. F., et al. Quantitative assessment of single-cell RNA-sequencing methods. Nat Methods. 2014;11(1):41.
20. Ståhlberg A., Kubista M. The workflow of single-cell expression profiling using quantitative real-time PCR. Expert Rev Mol Diagn. 2014;14(3):323–31.
21. Dalton L. A., Dougherty E. R. Bayesian minimum mean-square error estimation for classification error—Part I: Definition and the Bayesian MMSE error estimator for discrete classification. IEEE Trans Sig Process. 2011;59(1):115–29.
22. Imani M., Braga-Neto U. Multiple model adaptive controller for partially-observed Boolean dynamical systems. In: 2017 American Control Conference (ACC). IEEE; 2017. p. 1103–8. https://doi.org/10.23919/ACC.2017.7963100.
23. Kantas N., Doucet A., Singh S. S., Maciejowski J., Chopin N., et al. On particle methods for parameter estimation in state-space models. Stat Sci. 2015;30(3):328–51.
24. Hajiramezanali M., Amindavar H. Maneuvering target tracking based on SDE driven by GARCH volatility. In: 2012 IEEE Statistical Signal Processing Workshop (SSP). IEEE; 2012. p. 764–7. https://doi.org/10.1109/SSP.2012.6319816.
25. Imani M., Ghoreishi S. F., Allaire D., Braga-Neto U. MFBO-SSM: Multi-fidelity Bayesian optimization for fast inference in state-space models. In: AAAI; 2019.
26. Hajiramezanali M., Amindavar H. Maneuvering target tracking based on combined stochastic differential equations and GARCH process. In: 2012 11th International Conference on Information Science, Signal Processing and their Applications (ISSPA). IEEE; 2012. p. 1293–7. https://doi.org/10.1109/ISSPA.2012.6310492.
27. Hajiramezanali M., Fouladi S. H., Ritcey J. A., Amindavar H. Stochastic differential equations for modeling of high maneuvering target tracking. ETRI J. 2013;35(5):849–58.
28. Pitt M. K., Shephard N. Filtering via simulation: Auxiliary particle filters. J American Stat Assoc. 1999;94(446):590–9.
29. Qian X., Ghaffari N., Ivanov I., Dougherty E. R. State reduction for network intervention in probabilistic Boolean networks. Bioinformatics. 2010;26(24):3098–104.