2018 IEEE INTERNATIONAL WORKSHOP ON MACHINE LEARNING FOR SIGNAL PROCESSING, SEPT. 17–20, 2018, AALBORG, DENMARK

INFERENCE OF GENE REGULATORY NETWORKS BY MAXIMUM-LIKELIHOOD ADAPTIVE FILTERING AND DISCRETE FISH SCHOOL SEARCH

Yukun Tan, Fernando B. Lima Neto and Ulisses Braga Neto

Department of Electrical and Computer Engineering
Texas A&M University, College Station, TX, USA 77843-3128

ABSTRACT

We propose a new algorithm for inference of gene regulatory networks (GRN) from noisy gene expression data based on maximum-likelihood (ML) adaptive filtering and the discrete fish school search algorithm (DFSS). The approach is based on the general partially-observed Boolean dynamical system (POBDS) model, and as such can be used for simultaneous state and parameter estimation for any Boolean dynamical system observed in noise. The proposed DFSS-ML-BKF algorithm combines the ML adaptive Boolean Kalman Filter (ML-BKF) with DFSS, a version of the Fish School Search algorithm tailored for discrete parameter spaces. Results based on synthetic gene expression time-series data using the well-known p53-MDM2 negative-feedback loop GRN demonstrate that DFSS-ML-BKF can infer the network topology accurately and efficiently.

Index Terms— Discrete fish school search, gene regulatory network, partially observed Boolean dynamical system, maximum-likelihood estimation, Boolean Kalman filter

1. INTRODUCTION

Gene regulatory networks (GRN) govern the functioning of key cellular processes, such as the cell cycle, stress response, and DNA repair. Inference of GRNs from gene expression time-series data is a problem of critical importance in computational biology [1, 2, 3].

The GRN inference problem includes the determination of the topology of the network as well as the estimation of expression and noise parameters. Towards this goal, many mathematical models have been proposed in the literature, including linear models [4], Bayesian networks [5, 6], and neural networks [7]. In particular, the Boolean network model, first introduced by Kauffman and collaborators [8], has proved to be an effective model for GRNs consisting of bi-stable genes, which can be either in an activated or inhibited transcriptional state [9, 10, 11].

This work was supported by the U.S. National Science Foundation, through NSF award CCF-1718924.

However, the Boolean network model is deterministic and assumes that the system Boolean states are directly observed without noise. The partially-observed Boolean dynamical system (POBDS) [12, 13] is a signal model that addresses those difficulties by postulating stochastic Boolean state and general observation processes, as well as state transition and observation noise processes; the time-series data in the experiment arise from the observation process, while the Boolean states are hidden. The optimal minimum mean-square error (MMSE) state estimator for this model can be computed exactly and efficiently through a recursive algorithm known as the Boolean Kalman Filter (BKF).

In [13], a framework was proposed for the simultaneous estimation of state and parameters for POBDS using a maximum-likelihood approach. In the case where the parameter space is discrete, the resulting ML-BKF estimator consists of a bank of BKFs running in parallel, each tuned to a different value of the parameter. This corresponds to an exhaustive search over the parameter space. However, for a system with a large number of unknown parameters, the computation of all candidate solutions becomes impractical.

In this paper, we propose to combine maximum-likelihood adaptive filtering and nature-inspired swarm intelligence techniques to perform the search over large discrete parameter space and thus make GRN inference possible in low-information cases where most of the topology of the network is unknown. The proposed framework combines the ML-BKF with the discrete Fish School Search (DFSS) algorithm. DFSS is a contribution of this paper, which combines elements of the original fish school search (FSS) [14, 15] and binary fish school search (BFSS) algorithms [16, 17]. The performance of the proposed approach is assessed by numerical experiments based on the well-known p53-MDM2 negative-feedback loop gene regulatory model.

2. PARTIALLY-OBSERVABLE BOOLEAN DYNAMICAL SYSTEMS

The POBDS state model is

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

978-1-5386-5477-4/18/$31.00 ©2018 IEEE

for $k = 1, 2, \ldots$ where $\mathbf{f} : \{0,1\}^d \longrightarrow \{0,1\}^d$ is called the *network function*, $\mathbf{X}_k \in \{0,1\}^d$ is the Boolean system state at time k, $\mathbf{u}_k \in \{0,1\}^d$ is an input at time k, $\mathbf{n}_k \in \{0,1\}^d$ is Boolean process noise at time k, and "$\oplus$" indicates component-wise modulo-2 addition.

The POBDS observation model is entirely general:

$$\mathbf{Y}_k \;=\; \mathbf{h}\left(\mathbf{X}_k, \mathbf{v}_k\right) \tag{2}$$

for $k = 1, 2, \ldots$ where $\mathbf{Y}_k$ is the data at time k, $\mathbf{h}$ is a general function of the state vector, and $\mathbf{v}_k$ is measurement noise. The process and observation noise processes $\{\mathbf{n}_k, \mathbf{v}_k; k = 1, 2, \ldots\}$ are assumed to be white noise, independent of each other, and independent of the initial state $\mathbf{X}_0$.

Given observations $\mathbf{Y}_{1:k} = \{\mathbf{Y}_1, \ldots, \mathbf{Y}_k\}$, we would like to find an estimator $\hat{\mathbf{X}}_k = \mathbf{g}(\mathbf{Y}_{1:k})$ of the state $\mathbf{X}_k$ that minimizes the conditional MSE:

$$\hat{\mathbf{X}}_k^{\mathrm{MS}} \;=\; \underset{\hat{\mathbf{X}}_k \in\, \Psi}{\operatorname{argmin}}\; E[\,||\hat{\mathbf{X}}_k - \mathbf{X}_k||^2 \mid \mathbf{Y}_{1:k}], \tag{3}$$

where Ψ is the space of all Boolean estimators.

For a vector $\mathbf{v} \in [0,1]^d$, define the thresholding operator $\overline{\mathbf{v}} \in \{0,1\}^d$ as $\overline{\mathbf{v}}(i) = 1$ if $\mathbf{v}(i) > 1/2$ and 0 otherwise, for $i = 1, \ldots, d$, respectively. It was shown in [12, 13] that

$$\hat{\mathbf{X}}_k^{\mathrm{MS}} \;=\; \overline{E\left[\mathbf{X}_k \mid \mathbf{Y}_{1:k}\right]}. \tag{4}$$

The optimal MMSE filter in (4) can be calculated exactly by a recursive procedure called the Boolean Kalman filter (BKF) [12, 13], which is described briefly next. Let $(\mathbf{x}^1, \ldots, \mathbf{x}^{2^d})$ be an arbitrary enumeration of the possible state vectors. Define the state conditional probability distribution vector as:

$$\mathbf{\Pi}_{k|k}(i) \;=\; P\left(\mathbf{X}_k = \mathbf{x}^i \mid \mathbf{Y}_{1:k}\right),\;\; i = 1, \ldots, 2^d, \tag{5}$$

for $k = 0, 1, \ldots$ According to equation (4),

$$\hat{\mathbf{X}}_k^{\mathrm{MS}} \;=\; \overline{E\left[\mathbf{X}_k \mid \mathbf{Y}_{1:k}\right]} \;=\; \overline{A\mathbf{\Pi}_{k|k}},\;\; k = 1, 2, \ldots \tag{6}$$

where $A \;=\; \left[\mathbf{x}^1 \cdots \mathbf{x}^{2^d}\right]$ is a matrix of size $d \times 2^d$.

The computation of $\mathbf{\Pi}_{k|k}$ can be performed recursively. First, notice that

$$\mathbf{\Pi}_{k|k-1} \;=\; M_k \mathbf{\Pi}_{k-1|k-1},\;\; k = 1, 2, \ldots \tag{7}$$

where M_k is the transition matrix of the Markov state process, with entries given by:

$$(M_k)_{ij} \;=\; P(\mathbf{X}_k = \mathbf{x}^i \mid \mathbf{X}_{k-1} = \mathbf{x}^j),\;\; i, j = 1, \ldots, 2^d. \tag{8}$$

On the other hand,

$$\mathbf{\Pi}_{k|k} \;\propto\; T(\mathbf{Y}_k)\, \mathbf{\Pi}_{k|k-1},\;\; k = 1, 2, \ldots \tag{9}$$

where "$\propto$" means that the result must be normalized to add up to 1, and $T(\mathbf{Y}_k)$ is the *update matrix*, which is a diagonal matrix of size $2^d \times 2^d$ with diagonal elements:

$$\left(T_k(\mathbf{Y}_k)\right)_{ii} \;=\; p\left(\mathbf{Y}_k \mid \mathbf{X}_k = \mathbf{x}^i\right),\;\; i = 1, \ldots, 2^d. \tag{10}$$

3. MAXIMUM-LIKELIHOOD ADAPTIVE BOOLEAN KALMAN FILTER

The previous BKF algorithm can only be applied to compute the optimal MMSE state estimator in case the system is fully known. If the network function or the statistics of the noise processes are unknown or only partially known, then a suboptimal adaptive approach must be employed, where the unknown components of the system are estimated simultaneously with the state.

In this paper, we will focus on the case where there are a finite number M of possible models in one-to-one correspondence to a parameter space $\Theta = \{\theta_1, \theta_2, \cdots, \theta_M\}$. For example, this would be the case when some of the discrete connections between genes in a GRN are unknown and need to be inferred from the data.

Next we review briefly the ML-BKF algorithm, proposed as a solution to the adaptive filtering problem in [13]. Given the data $\mathbf{Y}_{1:k}$ up to time k, the log-likelihood function $L_k(\theta)$ can be computed recursively via

$$\begin{aligned} L_k(\theta) &= \log\; p_\theta(\mathbf{Y}_{1:k}) \\ &= \log\; p_\theta(\mathbf{Y}_k \mid \mathbf{Y}_{1:k-1}) \;+\; log\, p_\theta(\mathbf{Y}_{1:k-1}) \\ &= \log\; p_\theta(\mathbf{Y}_k \mid \mathbf{Y}_{1:k-1}) \;+\; L_{k-1}(\theta)\,, \end{aligned} \tag{11}$$

where $L_0(\theta) = 0$ and

$$\begin{aligned} &p_\theta(\mathbf{Y}_k \mid \mathbf{Y}_{1:k-1}) \\ &= \sum_{i=1}^{2^d} p_\theta(\mathbf{Y}_k \mid \mathbf{X}_k = \mathbf{x}^i,\, \mathbf{Y}_{1:k-1})\; P_\theta(\mathbf{X}_k = \mathbf{x}^i \mid \mathbf{Y}_{1:k-1}) \\ &= \sum_{i=1}^{2^d} p_\theta(\mathbf{Y}_k \mid \mathbf{X}_k = \mathbf{x}^i)\; P_\theta(\mathbf{X}_k = \mathbf{x}^i \mid \mathbf{Y}_{1:k-1}) \\ &= ||\, T_k^\theta(\mathbf{Y}_k)\, \mathbf{\Pi}_{k|k-1}^\theta||_1\,, \end{aligned} \tag{12}$$

where $||\mathbf{w}||_1$ is the sum of the elements of vector $\mathbf{w}$, while T_k^θ and $\mathbf{\Pi}_{k|k-1}^\theta$ are respectively the update matrix and the one-step predicted state distribution at time k, defined in the previous section, for the system corresponding to parameter θ.

With $\beta_k^\theta = T_k^\theta(\mathbf{Y}_k)\, \mathbf{\Pi}_{k|k-1}^\theta$, (11) and (12) lead to

$$L_k(\theta) = L_{k-1}(\theta) + \log ||\beta_k^\theta||_1\,. \tag{13}$$

Notice that the quantity $||\beta_k^\theta||_1$ can be readily computed by a BKF tuned to parameter θ. It follows that ML estimates of the parameter and state

$$\begin{aligned} \hat{\theta}_k^{ML} &= \underset{\theta \in \{\theta_1, \cdots, \theta_M\}}{\operatorname{argmax}}\; L_k(\theta), \\ \hat{\mathbf{X}}_k^{ML} &= \hat{\mathbf{X}}_k(\hat{\theta}_k^{ML}), \end{aligned} \tag{14}$$

for $k = 1, 2, \ldots$ can be obtained by running M BKFs in parallel, each one tuned to a different parameter θ_j, for $j = 1, \ldots, M$.

4. FISH SCHOOL SEARCH ALGORITHM

Fish school search (FSS) is a population-based continuous optimization algorithm proposed in [14], which is inspired by the collective behavior of natural fish schools that expand and contract while searching for food. We describe here the original FSS algorithm for optimization over continuous spaces, and then propose in the next section a modification of the algorithm for the discrete spaces arising in GRN inference.

There are N "fish" in the "school" corresponding to possible solution of the optimization problem. Each fish is represented by an M-dimensional vector $\mathbf{z_i}$ with associated weight w_i, for $i = 1, \dots N$, where the weight reflects the quality of the solution. After randomly initializing the positions and weights of all fish, those are updated at each iteration by means of four main operations, which are described next.

Individual movement operator: Each fish $\mathbf{z}_i$ moves in the direction of a unit vector $\Delta\mathbf{z}_{\mathrm{ind},i}$ generated randomly and independently if it improves its fitness, otherwise it stays put:

$$\mathbf{z}_i(t{+}1) = \begin{cases} \mathbf{z}_i(t) + \xi \times step_{\mathrm{ind}} \times \Delta\mathbf{z}_{\mathrm{ind},i} \\ \quad \text{if } f(\mathbf{z}_i(t) + \xi \times step_{\mathrm{ind}} \times \Delta\mathbf{z}_{\mathrm{ind},i}) > f(\mathbf{z}_i(t)), \\ \mathbf{z}_i(t)\,, \qquad \text{otherwise.} \end{cases} \tag{15}$$

where f is the fitness function, $\xi \sim \mathrm{Unif}([0,1])$, and $step_{\mathrm{ind}}$ is the step size, which is linearly reduced after each iteration to promote exploitation over exploration later in execution.

Feeding operator: The weights of all fish are updated based on the fitness improvement from individual movement:

$$w_i(t{+}1) \;=\; w_i(t){+}\frac{f[\mathbf{z}_i(t+1)] - f[\mathbf{z}_i(t)]}{\max_i\{|f[\mathbf{z}_i(t+1)] - f[\mathbf{z}_i(t)]|\}}\,. \tag{16}$$

Collective instinctive movement operator: This operator makes the fish that had successful individual movements influence the collective direction of movement of the school:

$$\mathbf{z}_i(t+1) = \mathbf{z}_i(t) + \frac{\sum_{i=1}^{N} \Delta\mathbf{z}_{\mathrm{ind},i}\{f[\mathbf{z}_i(t+1)] - f[\mathbf{z}_i(t)]\}}{\sum_{i=1}^{N}\{f[\mathbf{z}_i(t+1)] - f[\mathbf{z}_i(t)]\}}, \tag{17}$$

Collective volitive movement operator: Let

$$\mathbf{B}(t) = \frac{\sum_{i=1}^{N} \mathbf{x}_i w_i(t)}{\sum_{i=1}^{N} w_i(t)}, \tag{18}$$

be the Barycenter of the fish school. If the school is successful, i.e., its total weight increases, its radius contracts:

$$\mathbf{z}_i(t+1) \;=\; \mathbf{z}_i(t) - \tau \times step_{\mathrm{vol}} \times [\,\mathbf{z}_i(t) - \mathbf{B}(t)], \tag{19}$$

otherwise, it expands to escape a bad region or local optimum:

$$\mathbf{z}_i(t+1) \;=\; \mathbf{z}_i(t) + \tau \times step_{\mathrm{vol}} \times [\,\mathbf{z}_i(t) - \mathbf{B}(t)], \tag{20}$$

where $\tau \sim \mathrm{Unif}([0,1])$, $\mathbf{B}(t)$ is the barycenter of the fish school at time t, and $step_{\mathrm{vol}}$ is the volitive step, which is set to twice the size of $step_{\mathrm{ind}}$. The procedure is summarized in the following algorithm, where the stopping criterion adopted here is to exit the algorithm after a pre-specified number of iterations $t_{\max}$.

Algorithm 1: Fish School Search Algorithm.

```
Initialize randomly all fish positions and weights.
while stopping criterion is not met do
  for each fish do
    Execute individual movement using (15)
    Evaluate fitness and feed the fish using (16)
  end for
  for each fish do
    Execute instinctive movement using (17)
  end for
  Calculate barycenter using (18)
  for each fish do
    Execute volitive movement using (19) or (20)
  end for
  Update step_ind and step_vol
end while
```

5. DFSS-ML-BKF ALGORITHM FOR GENE REGULATORY NETWORK INFERENCE

GRNs in the biomedical literature are typically represented by gene activation/inhibition pathway diagrams, such as the one in Figure 1(a). If $\mathbf{f} = (f_1, \dots, f_d)$ is the network function in (1), the activation/inhibition of gene i can be modeled as:

$$f_i(\mathbf{x}) \;=\; \begin{cases} 1, & \sum_{j=1}^{d} a_{ij}\mathbf{x}(j) + b_i + \mathbf{u}(i) > 0\,, \\ 0, & \sum_{j=1}^{d} a_{ij}\mathbf{x}(j) + b_i + \mathbf{u}(i) \le 0\,, \end{cases} \tag{21}$$

where a_{ij} can take three values: $a_{ij} = +1$ if gene j activates gene i, $a_{ij} = -1$ if gene j represses gene i, and $a_{ij} = 0$ if gene j is not an input to gene i; b_i can take two values: $b_i = +1/2$ or $b_i = -1/2$, according to whether gene i is activated or repressed, respectively, when its inputs contain the same number of activation and inhibition signals.

In this paper, we assume that all system parameters are known, except for some of the interactions between genes, i.e., some of the discrete parameters a_{ij}. Therefore, the FSS algorithm described in the previous section needs to be modified to deal with a discrete parameter space. The algorithm proposed has similarities with the Binary FSS algorithm in [16], which deals with binary parameter vectors (while we deal here with ternary parameters a_{ij}). The modifications to the original FSS algorithm are as follows.

Discrete initialization: The initial position $\mathbf{z}_i(0) = [z_{01}, \dots, z_{0M}]$ of each fish is set to:

$$z_{0j} \;=\; \begin{cases} 1, & \text{if } \rho \ge 0.5, \\ -1, & \text{if } \rho \le -0.5, \\ 0, & \text{otherwise,} \end{cases} \tag{22}$$

for $j = 1, \dots, M$, where $\rho \sim \mathrm{Unif}([0,1])$. The initialization

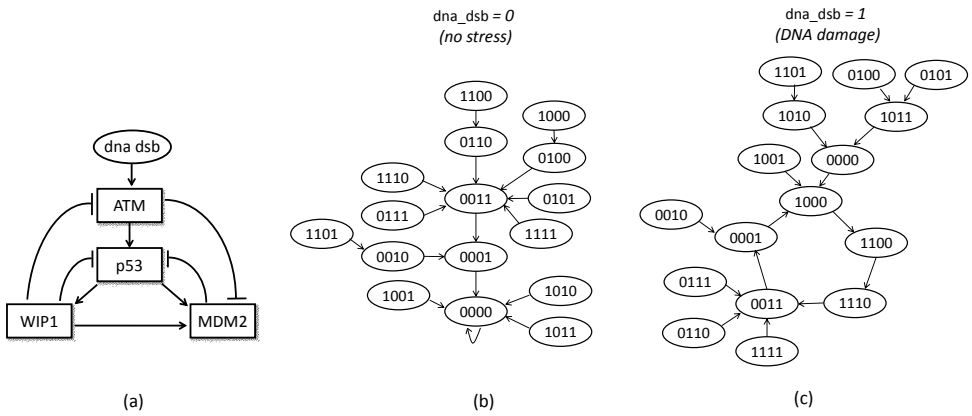

Fig. 1. Activation/inhibition pathway diagram and state transition diagrams corresponding to a constant input dna_dsb = 0 (no-stress) and dna_dsb = 1 (DNA-damage) for the p53-MDM2 negative feedback loop gene regulatory network with negative regulation biases.

of each parameter is biased towards 0 over 1 and -1. This reflects the biological fact that GRNs are sparsely connected.

Discrete moves: Let $\mathbf{s} = [s_1, s_2, \ldots, s_M]$ be the displacement vector calculated by the individual, instinctive, or volitive movement operators of the original FSS algorithm. The values s_i are continuous, with $-1 \leq s_i \leq 1$, for $i = 1, \ldots, M$, and must be discretized to be applicable over a ternary parameter space. We generalize the scheme for binary parameters in [16] by considering two adaptive thresholds:

$$\begin{aligned} thr_{\text{pos}} &= \max_j\{\text{pos}(s_j)\} \times \tfrac{t}{t_{\max}} \\ thr_{\text{neg}} &= \min_j\{\text{neg}(s_j)\} \times \tfrac{t}{t_{\max}} \end{aligned} \tag{23}$$

where the positive and negative parts of s_j are defined as $\text{pos}(s_j) = (s_j + |s_j|)/2$ and $\text{neg}(s_j) = (s_j - |s_j|)/2$, respectively. Then the discrete move is defined as

$$\mathbf{z}_i(t+1) = \mathbf{z}_i(t) + T[\text{pos}(\mathbf{s}) > thr_{\text{pos}}] - T[\text{neg}(\mathbf{s}) < thr_{\text{neg}}], \tag{24}$$

where the pos, neg, and T threshold operators are applied componentwise. Therefore, a positive continuous displacement s_j produces a discrete displacement of +1 if it is larger than thr_{pos}; a discrete displacement of -1 if it is smaller than thr_{neg}; and no displacement, otherwise. An "absorbing" boundary condition is adopted, whereby if a component of $\mathbf{z}_i(t+1)$ would be larger than +1 or smaller than -1 after a move, then it stays unchanged. The relaxation factor $t/t_{\max}$ in (23) ensures convergence.

The fitness function for the DFSS search is the likelihood function $L_k(\theta)$ in (11). By virtue of (13), $L_k(\theta)$ by recomputed recursively by running a BKF tuned to parameter θ. Furthermore, this computation can be picked up from the last time point at which the specific value of θ was visited, so the BKF does not have to be restarted from time zero (unless this value of θ has not been visited previously). Notice that, in contrast to the ML-BKF approach in [13], which runs 3^M filters in parallel and thus has exponential complexity $O(3^M)$ in the number of parameters M, the DFSS-ML-BKF algorithm has complexity $O(MN)$, where N is the number of fish.

6. NUMERICAL EXPERIMENTS

In this section, we examine the performance of DFSS-ML-BKF using the well-known p53-MDM2 negative-feedback gene regulatory network [18], which is displayed in Figure 1. The gene interaction parameters a_{ij} for this GRN are:

$$\begin{aligned} &a_{11} = 0, \quad a_{12} = 0, \quad a_{13} = -1, \quad a_{14} = 0 \\ &a_{21} = +1, \quad a_{22} = 0, \quad a_{23} = -1, \quad a_{24} = -1 \\ &a_{31} = 0, \quad a_{32} = +1, \quad a_{33} = 0, \quad a_{34} = 0 \\ &a_{41} = -1, \quad a_{42} = +1, \quad a_{43} = +1, \quad a_{44} = 0 \end{aligned}$$

with all biases $b_i = -1/2$, $i = 1, 2, 3, 4$. The input signal $\mathbf{u}_k$ = (dna_dsb, 0, 0, 0) is constant, with dna_dsb=0 (no stress) or dna_dsb=1 (DNA damage). These two cases lead to the state transition diagrams displayed in Figure 1(b) and (c), respectively.

In our experiments, the process noise $\mathbf{n}_k$ vector in (1) consists of independent Bernoulli(p) random variables. Parameter p is the probability that any component of the noise vector is 1 which, according to (1), flips the state of the corresponding gene. On the other hand, synthetic gene expression data is generated by adding zero-mean Gaussian noise of variance σ_v^2 to each gene state; i.e., (2) here takes the form ¡

$$\mathbf{Y}_k = \mathbf{X}_k + \mathbf{v}_k, \text{ where } \mathbf{v}_k \sim \mathcal{N}_d(0, \sigma_v^2 I_d). \tag{25}$$

Average accuracy rates computed over 500 independently-generated time series of different length n, process noise intensity p, observation noise standard deviation σ_v, and the no-stress and DNA-damage conditions are displayed in Table 1. In each run, four of the interaction parameters are

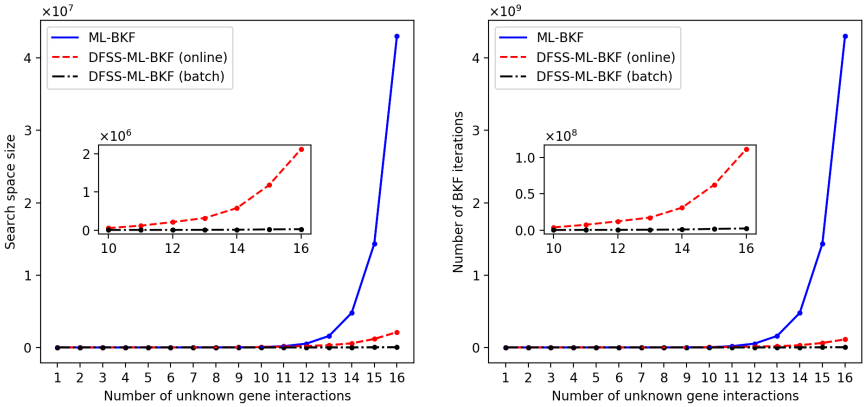

Fig. 2. Comparison in computational effort among the various methods.

selected as unknown randomly, but the process and observation noises and regulation biases are always assumed to be known. The accuracy rates correspond to the proportion of time all four gene interaction parameters are correctly identified at the time-series endpoint (i.e., an error occurs if at least one parameter is incorrectly identified). We can observe that performance increases monotonically with an increasing time-series length and decreasing observation noise intensity, as expected. The behavior with respect to the process noise is more interesting: under no stress, performance exhibits peaking, whereby accuracy rates initially increases with increasing process noise but eventually decreases. The reason for this is that at low process noise levels, the system cannot escape its singleton attractor easily, visiting fewer states and decreasing performance. This is not an issue under DNA damage, which contains a large cyclic attractor. On the other hand, large process noise intensity renders the system too chaotic, decreasing performance in all cases. Finally, we can see that accuracy rates are better under DNA damage than no stress, for a similar reason moderate process noise helps the inference process: under DNA damage the system contains a large cyclic attractor and thus, for a fixed time series length, tends to visit a larger portion of the state space than under no stress, when the system contains a singleton attractor. In fact, performance can be quite poor under no stress, large process and observation noise and small time series length, while the opposite happens under DNA damage and small process and observation noise levels.

Next we compare the performance of the ML-BKF and the DFSS-ML-BKF approaches. Since the former corresponds to an exhaustive search, it is expected to uniformly dominate in terms of accuracy. The question we would like to ask instead is how they compare in terms of computational effort at a high level of accuracy for the DFSS-ML-BKF, as the number of unknown parameters (i.e., the number of unknown gene interactions) increases. The parameters of the

Table 1. Average accuracy rates for estimation of the gene interaction parameters.

		No-stress			DNA-damage		
n	p	$\sigma_v = 0.1$	$\sigma_v = 0.3$	$\sigma_v = 0.5$	$\sigma_v = 0.1$	$\sigma_v = 0.3$	$\sigma_v = 0.5$
20	0.05	0.378	0.338	0.194	0.830	0.762	0.624
	0.1	0.446	0.388	0.208	0.738	0.616	0.462
	0.2	0.426	0.290	0.156	0.516	0.374	0.202
	0.3	0.230	0.192	0.086	0.238	0.138	0.074
50	0.05	0.528	0.426	0.312	0.954	0.908	0.838
	0.1	0.728	0.610	0.400	0.948	0.898	0.766
	0.2	0.808	0.628	0.322	0.838	0.666	0.490
	0.3	0.538	0.392	0.170	0.518	0.334	0.152
100	0.05	0.692	0.596	0.444	0.986	0.956	0.914
	0.1	0.900	0.786	0.528	0.996	0.976	0.896
	0.2	0.932	0.854	0.518	0.972	0.898	0.698
	0.3	0.780	0.630	0.296	0.756	0.634	0.324
200	0.05	0.902	0.732	0.486	1.000	0.992	0.964
	0.1	0.982	0.882	0.688	1.000	1.000	0.966
	0.2	0.996	0.958	0.742	1.000	0.980	0.902
	0.3	0.964	0.858	0.522	0.944	0.828	0.566

simulation are set to $n = 100$, $p = 0.1$, and $\sigma_v = 0.1$ under DNA damage. Two settings for the DFSS-ML-BKF are considered: "online," when the observations are presented one by one and "batch," when all 100 data points are presented at once. This does not make a difference for the ML-BKF algorithm, since it runs 3^M BKFs in parallel for a total of $3^M \times 100$ BKF iterations in either case. We increased the size of the fish school N and the maximum number of iterations $t_{\max}$ to make the DFSS-ML-BKF accuracy rate at least 97% throughout. The number of visited parameters in the search space and the number of BKF iterations against

the number of unknown gene interactions are plotted in Figure 2. We can see that the two methods are very similar in computational effort for a small number of parameters, but DFSS-ML-BKF is much more efficient for a number of unknown parameters exceeding 11. We can also observe that the batch method is more efficient than the online method, since in the former case DFSS is only run once.

7. CONCLUSION

We proposed in this paper the DFSS-ML-BKF algorithm for inference of GRNs. The algorithm is based on the POBDS model and combines the ML-BKF and a new variant of fish school search for discrete parameter spaces, called here DFSS. After a brief review of the POBDS model, the ML-BKF, and the original FSS algorithm, we introduced the DFSS algorithm, which allows us to replace the exhaustive search in the ML-BKF by an efficient search based on the FSS heuristic. Numerical experiments with the p53-MDM2 negative feedback loop GRN demonstrated the accuracy and efficiency of the proposed method.

8. REFERENCES

[1] Riet De Smet and Kathleen Marchal, "Advantages and limitations of current network inference methods," *Nature Reviews Microbiology*, vol. 8, no. 10, pp. 717, 2010.

[2] Ilya Shmulevich, Edward R Dougherty, and Wei Zhang, "From boolean to probabilistic boolean networks as models of genetic regulatory networks," *Proceedings of the IEEE*, vol. 90, no. 11, pp. 1778–1792, 2002.

[3] Shoudan Liang, Stefanie Fuhrman, and Roland Somogyi, "Reveal, a general reverse engineering algorithm for inference of genetic network architectures," 1998.

[4] Eugene P van Someren, Lodewyk FA Wessels, and Marcel JT Reinders, "Linear modeling of genetic networks from experimental data.," in *Ismb*, 2000, pp. 355–366.

[5] Kevin Murphy, Saira Mian, et al., "Modelling gene expression data using dynamic bayesian networks," Tech. Rep., Technical report, Computer Science Division, University of California, Berkeley, CA, 1999.

[6] Nir Friedman, Michal Linial, Iftach Nachman, and Dana Pe'er, "Using bayesian networks to analyze expression data," *Journal of computational biology*, vol. 7, no. 3-4, pp. 601–620, 2000.

[7] Daniel C Weaver, Christopher T Workman, and Gary D Stormo, "Modeling regulatory networks with weight matrices," in *Biocomputing'99*, pp. 112–123. World Scientific, 1999.

[8] Stuart A Kauffman, "Metabolic stability and epigenesis in randomly constructed genetic nets," *Journal of theoretical biology*, vol. 22, no. 3, pp. 437–467, 1969.

[9] Réka Albert and Hans G Othmer, "The topology of the regulatory interactions predicts the expression pattern of the segment polarity genes in drosophila melanogaster," *Journal of theoretical biology*, vol. 223, no. 1, pp. 1–18, 2003.

[10] Fangting Li, Tao Long, Ying Lu, Qi Ouyang, and Chao Tang, "The yeast cell-cycle network is robustly designed," *Proceedings of the National Academy of Sciences*, vol. 101, no. 14, pp. 4781–4786, 2004.

[11] A. Faure, A. Naldi, C. Chaouiya, and D. Thieffry, "Dynamical analysis of a generic boolean model for the control of the mammalian cell cycle," *"Bionformatics"*, vol. 22, no. 14, pp. e124–e131, 2006.

[12] U.M. Braga-Neto, "Optimal state estimation for Boolean dynamical systems," 2011, Proceedings of 45th Annual Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, CA.

[13] M. Imani and U.M. Braga-Neto, "Maximum-likelihood adaptive filter for partially-observed Boolean dynamical systems," *IEEE Transactions on Signal Processing*, vol. 65, no. 2, pp. 359–371, 2017.

[14] Carmelo JA Bastos Filho, Fernando B de Lima Neto, Anthony JCC Lins, Antonio IS Nascimento, and Marilia P Lima, "A novel search algorithm based on fish school behavior," in *Systems, Man and Cybernetics, 2008. SMC 2008. IEEE International Conference on*. IEEE, 2008, pp. 2646–2651.

[15] CJA Bastos-Filho and DO Nascimento, "An enhanced fish school search algorithm," in *Computational Intelligence and 11th Brazilian Congress on Computational Intelligence (BRICS-CCI & CBIC), 2013 BRICS Congress on*. IEEE, 2013, pp. 152–157.

[16] João AG Sargo, Susana M Vieira, João MC Sousa, and Carmelo JA Bastos Filho, "Binary fish school search applied to feature selection: Application to icu readmissions," in *Fuzzy Systems (FUZZ-IEEE), 2014 IEEE International Conference on*. IEEE, 2014, pp. 1366–1373.

[17] Raphael F Carneiro and Carmelo JA Bastos-Filho, "Improving the binary fish school search algorithm for feature selection," in *Computational Intelligence (LA-CCI), 2016 IEEE Latin American Conference on*. IEEE, 2016, pp. 1–6.

[18] Eric Batchelor, Alexander Loewer, and Galit Lahav, "The ups and downs of p53: understanding protein dynamics in single cells," *Nature Reviews Cancer*, vol. 9, no. 5, pp. 371, 2009.