2020 IEEE INTERNATIONAL WORKSHOP ON MACHINE LEARNING FOR SIGNAL PROCESSING, SEPT. 21–24, 2020, ESPOO, FINLAND

INFERENCE OF PROTEIN-PROTEIN INTERACTION NETWORKS FROM LIQUID-CHROMATOGRAPHY MASS-SPECTROMETRY DATA BY APPROXIMATE BAYESIAN COMPUTATION-SEQUENTIAL MONTE CARLO SAMPLING

Yukun Tan[†], *Fernando B. Lima Neto*[‡], *Ulisses Braga Neto*[†]

[†] Department of Electrical and Computer Engineering
Texas A&M University, College Station, TX, USA, 77843-3128
[‡] Department of Computer Engineering
Polytechnic University of Pernambuco, Recife, Brazil, 50720

ABSTRACT

We propose a new algorithm for inference of protein-protein interaction (PPI) networks from noisy time series of Liquid-Chromatography Mass-Spectrometry (LC-MS) proteomic expression data based on Approximate Bayesian Computation - Sequential Monte Carlo sampling (ABC-SMC). The algorithm is an extension of our previous framework PALLAS. The proposed algorithm can be easily modified to handle other complex models of expression data, such as LC-MS data, for which the likelihood function is intractable. Results based on synthetic time series of cytokine LC-MS measurements corresponding to a prototype immunomic network demonstrate that our algorithm is capable of inferring the network topology accurately.

Index Terms— Protein-Protein Interaction Network, Liquid-Chromatography Mass-Spectrometry, Approximate Bayesian Computation, Sequential Monte Carlo, PALLAS

1. INTRODUCTION

Reconstructing the molecular networks underlying the functioning of a living cell is one of the main goals of biology and medicine. In this respect, protein-protein interaction (PPI) networks play a major role in most cellular processes. Inference of PPI networks from protein expression data is essential for understanding the structure, function, and dynamics of the cell [1]. With the advancement of high-throughput experimental technologies, such as Liquid-Chromatography Mass-Spectrometry (LC-MS), massive amounts of proteomics data make the data-driven reconstruction of PPI networks possible.

Many methods for inference of PPI networks have been developed, including experimental [2, 3] and computational techniques [1, 4, 5, 6]. The Boolean network (BN) model [7] is an effective model which is widely used for inference gene regulatory networks (GRN) due to its ability to describe temporal patterns of gene activation and inactivation and its comparatively small data requirement.

The Penalized mAximum LikeLihood and pArticle Swarms (PALLAS) algorithm for GRN inference from gene-expression data was introduced in [8]. PALLAS is a practical method for gene network inference based on the Partially-Observable Boolean Dynamical System (POBDS) model [9, 10], using penalized maximum likelihood (PML) and particle swarms for optimization. However, PALLAS assumes that the likelihood function describing the expression data is available. In the present paper, we extended PALLAS to the the inference of PPI networks from proteomics data, for which the likelihood function is intractable. This is achieved by applying Approximate Bayesian Computation based on Sequential Monte Carlo sampling (ABC-SMC) method [11]. The ABC-SMC method is combined with mixed fish school search (MFSS) algorithm [8], to infer the PPI and estimate the parameters simultaneously, which allows it to handle the the absence of any prior knowledge. The performance of the proposed approach is assessed by numerical experiments based on a prototype immunomic network.

2. PARTIALLY-OBSERVABLE BOOLEAN DYNAMICAL SYSTEMS

The POBDS system is a nonlinear statistical model [9] that allows for uncertainty in Boolean state transitions and partial observation of the Boolean state variables through noise.

2.1. State model

Consider a state process $\{\mathbf{X}_k; k = 0, 1, \ldots\}$, where $\mathbf{X}_k \in \{0, 1\}^d$ is a Boolean vector of size d. In the POBDS model, the system state $\mathbf{X}_k$ evolves according to:

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

The authors acknowledge the support of the National Science Foundation, through NSF award CCF-1718924.

pdfelement
The Trial Version

978-1-7281-6662-9/20/$31.00 ©2020 IEEE

Authorized licensed use limited to: Texas A M University. Downloaded on October 24,2020 at 21:29:19 UTC from IEEE Xplore. Restrictions apply.

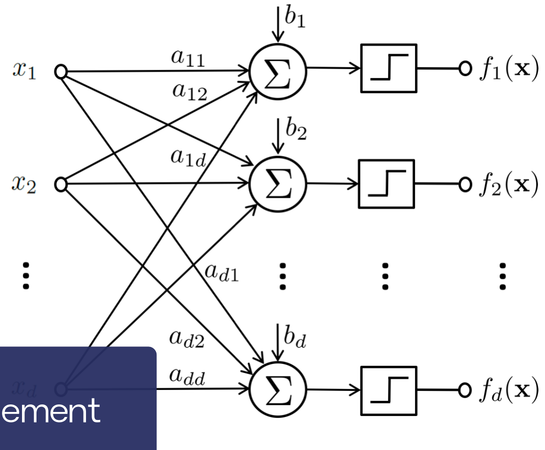
for $k = 1, 2, \ldots$ where $\mathbf{f} : \{0,1\}^d \longrightarrow \{0,1\}^d$ is called the *network function*, $\mathbf{n}_k \in \{0,1\}^d$ is additive noise at time k, and "$\oplus$" indicates component-wise modulo-2 addition. The state and noise processes are assumed to be independent. The state model (1) can be suitably modified to include external inputs, if desired.

The process noise vector $\mathbf{n}_k$ is assumed to have independent components distributed as Bernoulli(p), which models uncertainty in the state transition. We assume $p = 0.05$ as a default value in our framework, but a different value $0 \leq p \leq 0.5$ can be selected by the user; however, the closer it is to $p = 0.5$, the more chaotic the system will be.

In addition, we assume a specific model for the network function. Let a sample state vector $\mathbf{x} \in \{0,1\}^d$ and the network function $\mathbf{f}$ be expressed in component form as $\mathbf{x} = (x_1, \ldots, x_d)$ and $\mathbf{f} = (f_1, \ldots, f_d)$, respectively. Each component $f_i : \{0,1\}^d \to \{0,1\}$ is given by

$$f_i(\mathbf{x}) = \begin{cases} 1, & \sum_{j=1}^{d} a_{ij} x_j + b_i > 0\,, \\ 0, & \text{otherwise}, \end{cases} \tag{2}$$

where a_{ij} and b_i are system parameters. The former can take three values: $a_{ij} = +1$ if there is positive regulation (activation) from protein j to protein i; $a_{ij} = -1$ if there is negative regulation (inhibition) from protein j to protein i; and $a_{ij} = 0$ if protein j is not an input to protein i. The latter specifies regulation *biases* and can take two values: $b_i = +1/2$ if protein i is positively biased in the sense that an equal number of activation and inhibition inputs will produce activation; the reverse being the case if $b_i = -1/2$. The network model is depicted in Figure 1, where the threshold units are step functions that output 1 if the input is positive, and 0, otherwise. This model constraint reduces the number of parameters needed to specify $\mathbf{f}$ from 2^d to $d^2 + d$.



pdfelement
The Trial Version

Fig. 1. Schematic representation of the network function.

2.2. Observation Model

Let $\mathbf{Y}_k$ be the observation corresponding to the state $\mathbf{X}_k$ at time k. The sequence of states is observed indirectly through the process $\{\mathbf{Y}_k; k = 0, 1, \ldots\}$, where the measurement vector Y_k is a general nonlinear function of the state and observation noise:

$$\mathbf{Y}_k = \mathbf{h}(\mathbf{X}_k, \mathbf{v}_k) \tag{3}$$

for $k = 1, 2, \ldots$, where the noise vector $\mathbf{v}_k$ is assumed to independent of the state process and state transition noise process.

In what follows, we consider the observational model for liquid chromatography-mass spectrometry (LC-MS) data proposed in [12], which we describe briefly next. The protein concentration can be modeled as a Gamma distribution [13],

$$\gamma_i = \Gamma(\mathscr{k}, \vartheta),\; i = 1, 2, \ldots, d, \tag{4}$$

where the shape $\mathscr{k}$ and scale ϑ parameters are assumed to be uniform random variables, such that $\mathscr{k} \sim \text{Unif}(\mathscr{k}_{low}, \mathscr{k}_{high})$ and $\vartheta \sim \text{Unif}(\vartheta_{low}, \vartheta_{high})$. The multivariate Gaussian distribution is recommended as the model for protein concentration variations. In this paper, we assume that protein concentrations are mutually independent, so that

$$y_i = \gamma_i + \gamma_i(\zeta_i - 1)\, x_i + v_i, \tag{5}$$

for $i = 1, \ldots, d$, where γ_i is the baseline of protein concentration expression levels, ζ_i is the fold change when protein i is overexpressed, and v_i is an uncorrelated zero-mean Gaussian noise for protein i which $\boldsymbol{v} \sim \mathcal{N}(0, \Sigma)$, where $\Sigma = \text{diag}(\sigma_1^2, \ldots, \sigma_d^2)$ and $\sigma_i^2 = \varphi \times \gamma_i^2$. The coefficient of variation φ is calibrated based on the observed data.

3. EXTENDED PALLAS ALGORITHM

In this section, we describe the extended PALLAS algorithm for inference of Boolean PPI networks from noisy time series of LC-MS data. The framework uses the ABC-SMC algorithm to perform an efficient computation of a penalized log-likelihood cost function.

Let $\theta = (\theta_{\text{disc}}, \theta_{\text{cont}}) \in \Theta$, with $\theta_{\text{disc}} \in \Theta_{\text{disc}}$ and $\theta_{\text{cont}} \in \Theta_{\text{cont}}$, be the discrete and continuous unknown model parameters, where Θ, Θ_{disc} and Θ_{cont} are the corresponding parameter spaces, with $\Theta = \Theta_{\text{disc}} \times \Theta_{\text{cont}}$. Here, θ_{disc} contains the parameters of the network function in (2), namely the edge parameters $a_{ij} \in \{-1, 0, 1\}$, for $i, j = 1, \ldots, d$, and the regulation bias parameters $b_i \in \{-1/2, 1/2\}$, for $i = 1, \ldots, d$. Hence, $\Theta_{\text{disc}} = \{-1, 0, 1\}^{d^2} \times \{-1/2, 1/2\}^d$. This is a finite space, but its cardinality $|\Theta_{\text{disc}}| = 3^{d^2} \times 2^d$ increases extremely fast with the number of proteins d. On the other hand, θ_{cont} contains the observational parameters: the shape $\mathscr{k}$ and scale ϑ of the baseline of protein concentration expression level, the fold change ζ, and the coefficient of variation φ.

Next we describe main steps of the extended algorithm.

Authorized licensed use limited to: Texas A M University. Downloaded on October 24,2020 at 21:29:19 UTC from IEEE Xplore. Restrictions apply.

3.1. ABC-SMC Algorithm

In our previous work [8], we use an auxiliary particle filter algorithm to estimate the likelihood and obtain the unknown parameters by maximum the estimation. However, this method cannot be used for complex models, in which the conditional density $\mathbf{g}(\mathbf{Y}|\mathbf{X})$ is intractable or computationally expensive. Instead, we will assume that one may still able to obtain samples from this conditional likelihood for different values of the parameter θ, which leads to the Approximate Bayesian Computation (ABC) technique [11]. ABC replaces the calculation of the likelihood with a comparison between the observed and sampled data to approximate the likelihood, namely, we generate M samples from $\mathbf{g}_\theta(\mathbf{Y}_k|\mathbf{X}_k)$ and the estimated likelihood can be calculated as

$$l_k = \frac{\sum_{j=1}^{M} \mathbb{1}(d(\hat{\mathbf{Y}}_k^j, \mathbf{Y}_k) \leq \epsilon_k)}{M} \tag{6}$$

for $k = 1, \ldots, T$, where ϵ_k is the precision tolerance, and $d(\cdot, \cdot)$ is the distance function between the observed and sampled data. Theoretically, the approximation obtained by ABC filtering is matched to true one when $\epsilon_k \approx 0$ and $M = \infty$.

However, a drawback of the ABC method is the low acceptance rate when stuck in a bad region. In order to improve the ABC performance, the use of Sequential Monte Carlo (SMC) sampling has been suggested [14, 15, 16]. In the SMC algorithm, $\frac{1}{N}\sum_{i=1}^{N} \mathbf{g}(\mathbf{Y}_k \mid \mathbf{X}_{k,i})$ is an approximation to the conditional likelihood $p(\mathbf{Y}_k \mid \mathbf{Y}_{1:k-1})$. Thus, with the estimated likelihood from the ABC algorithm, the full likelihood approximation $p(\mathbf{Y}_{1:k})$ can be generated.

In Algorithm 1 we present the ABC-SMC algorithm based on [14, 16]. The basic design elements are the number of particles N, the number of auxiliary observation samples M and the ABC precision tolerance ϵ. The vector $\Pi_{0|0}$ is the initial (prior) distribution of the states at time zero. The vector $\mathbf{W}$ gives the weight of the particles which is initialized to $1/N$ for all particles. The resampling step is necessary when the effective sample size (ESS) is low. The resampling threshold E is commonly taken to be $N/2$ [17].

3.2. Penalized Likelihood Computation

Suppose that the sample data consist of n independent time series $\mathbf{Y}_{1:k}^j = \{\mathbf{Y}_1^j, \ldots, \mathbf{Y}_k^j\}$ up to time k, for $j = 1, \ldots, n$. The penalized log-likelihood of model θ at time k is defined as

$$\begin{aligned} L_k(\theta) &= \frac{1}{kn} \log p_\theta(\mathbf{Y}_{1:k}^{(1)}, \ldots, \mathbf{Y}_{1:k}^{(n)}) - \eta \sum_{i,j=1}^{2^d} |a_{ij}| \\ &= \frac{1}{kn} \sum_{j=1}^{n} \log p_\theta(\mathbf{Y}_{1:k}^j) - \eta \sum_{i,j=1}^{2^d} |a_{ij}|, \end{aligned} \tag{7}$$

pdfelement
The Trial Version

where $\eta > 0$ is a regularization parameter, which has a default value of $\eta = 0.01$ in our implementation. Hence, the penalized log-likelihood in (7) is the sum of the average log-likelihood per time series and a negative value times the number of edges in the model. Maximization of (7) thus encourages the model to both fit the data and be sparse, i.e., contain a small number of edges between proteins, which is in agreement with biological knowledge. The value of η can be adjusted by the user to obtain a desired level of sparsity.

Algorithm 1 ABC-SMC

```
Initialize $\epsilon_1 > \epsilon_2 > \cdots > \epsilon_T > 0$ and
$\mathbf{X}_{0,i} \sim \Pi_{0|0}, \mathbf{W}_{0,i} = 1/N$, for $i = 1, 2, \ldots, N$
for $k = 1$ to $T$ do:
    for $i = 1$ to $N$, do:
        $\mathbf{X}_{k,i} = \mathbf{f}(\mathbf{X}_{k-1,i}) \oplus \mathbf{n}_{k,i}$
        for $j = 1$ to $M$ do:
            Generate $\hat{\mathbf{Y}}_{k,i}^j \sim \mathbf{g}_\theta(\cdot|\mathbf{X}_{k,i})$
        end for
        $\tilde{\mathbf{W}}_{k,i} = \frac{\sum_{j=1}^{M} \mathbb{1}(d(\hat{\mathbf{Y}}_{k,i}^j, \mathbf{Y}_{k,i}) \leq \epsilon_k)}{M}$
        $\mathbf{W}_{k,i} \propto \mathbf{W}_{k-1,i}\tilde{\mathbf{W}}_{k,i}$
    end for
    $\mathbf{W}_{k,i} = \mathbf{W}_{k,i} / \sum_{i=1}^{N} \mathbf{W}_{k,i}$
    If $ESS = [\sum_{i=1}^{N} (\mathbf{W}_{k,i})^2]^{-1} \leq E$:
        Resample $\mathbf{X}_{k,i}$ with weights $\mathbf{W}_{k,i}$
        Set $\mathbf{W}_{k,i} = 1/N$
end for
```

Notice that

$$\begin{aligned} \log\, p_\theta(\mathbf{Y}_{1:k}^j) &= \log \Big[p_\theta(\mathbf{Y}_k^j \mid \mathbf{Y}_{1:k-1}^j) p_\theta(\mathbf{Y}_{k-1}^j \mid \mathbf{Y}_{1:k-2}^j) \\ &\qquad \cdots\, p_\theta(\mathbf{Y}_2^j \mid \mathbf{Y}_1^j) p_\theta(\mathbf{Y}_1^j) \Big] \\ &= \sum_{m=1}^{k} \log\, p_\theta(\mathbf{Y}_m^j \mid \mathbf{Y}_{1:m-1}^j), \end{aligned} \tag{8}$$

and based on [14, 16],

$$p_\theta(\mathbf{Y}_k^j \mid \mathbf{Y}_{1:k-1}^j) = \frac{1}{N} \sum_{i=1}^{N} \tilde{\mathbf{W}}_{m,i}^{\theta,j}, \tag{9}$$

where (7) can finally be written as

$$L_k(\theta) = \frac{1}{kn} \sum_{j=1}^{n} \sum_{m=1}^{k} \log\big(\frac{1}{N} \sum_{i=1}^{N} \tilde{\mathbf{W}}_{k,i}^{\theta,j}\big) - \eta \sum_{i,j=1}^{2^d} |a_{ij}|. \tag{10}$$

The maximum-likelihood estimator of parameter θ at time k is then given by

$$\hat{\theta}_k^{\text{ML}} = \arg\max_{\theta \in \Theta} L_k(\theta). \tag{11}$$

Authorized licensed use limited to: Texas A M University. Downloaded on October 24,2020 at 21:29:19 UTC from IEEE Xplore. Restrictions apply.

3.3. Mixed Fish School Search Algorithm

Maximization of the penalized likelihood function is performed by the mixed fish school search (MFSS) algorithm described in [8, 18]. In the MFSS algorithm, the objective is to find a model that maximizes a given score or fitness, in this case, the penalizes likelihood of the model. Each candidate model, i.e., each candidate parameter vector $\theta = (\theta_{\text{disc}}, \theta_{\text{cont}})$, corresponds to a particle or "fish," which has therefore a discrete and a continuous part (hence, the name "mixed" FSS). An ensemble of candidate solutions is a "school," which is iteratively updated by a series of biologically-inspired moves, namely, (1) an individual movement operator for each fish (a small random perturbation), (2) a feeding operator that updates the weight of of all fish based on the fitness improvement from the previous step, (3) a collective instinctive movement operator that makes the fish that had successful individual movements influence the collective direction of movement of the school, and (4) a collective volitive movement operator when the fish move in concert, depending on whether the fish school is successful after the previous steps, i.e., its total weight increases, or not. If the fish school is successful, then it should contract, changing from exploration to exploitation mode. Otherwise, it should expand in order to explore the space more. This process is repeated for either a fixed number of iterations or until there is no significant improvement in the score. Please see [8] for a detailed description of the MFSS algorithm.

4. NUMERICAL EXPERIMENTS

4.1. Performance Criteria

Here we consider two classes of metrics, one based on the difference between the network functions and the other based on edge-calling accuracy rates [19].

4.1.1. Edge-Calling Accuracy Rates

An *edge* in the groundtruth network represents a relationship between two proteins. Here we consider directionality (an edge from protein i to protein j is distinct from an edge from protein j to protein i), but disregard activation/inhibition relationships. Let TP and FN be the total number of directional edges that are correctly detected (irrespective of inhibition/activation) and incorrectly missed by the inference algorithm, respectively. Similarly, let FP and TN be the total number of directional edges that are incorrectly found and correctly missed, respectively. We define the following edge-calling accuracy rates:

pdfelement
The Trial Version

(i) Sensitivity/True Positive Rate (TPR):

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}} \,. \tag{12}$$

(ii) Specificity/True Negative Rate (SPC):

$$\text{SPC} = \frac{\text{TN}}{\text{FP} + \text{TN}} \,. \tag{13}$$

(iii) Precision/Positive Predictive Value (PPV):

$$\text{PPV} = \frac{\text{TP}}{\text{TP} + \text{FP}} \,. \tag{14}$$

4.1.2. Network Function Distance

Let $\mathbf{f} = (f_1, \ldots, f_d)$ and $\hat{\mathbf{f}} = (\hat{f}_1, \ldots, \hat{f}_d)$ be the network functions of the groundtruth and inferred networks, where the component functions f_i and $\hat{f}_i$ are Boolean functions on d variables, for $i = 1, \ldots, d$; see (1). The performance criterion is the average number of disagreeing Boolean functions between the two networks

$$\varphi(\mathbf{f}, \hat{\mathbf{f}}) = \frac{1}{d \times 2^d} \sum_{i=1}^{d} \sum_{j=1}^{2^d} [f_i(\mathbf{x}^j) \oplus \hat{f}_i(\mathbf{x}^j)] \,. \tag{15}$$

This distance is related to the dynamical behavior of the networks, since it has to do with how the Boolean functions differ.

4.2. Immune System PPI Network Experiment

We investigate the performance of the extended PALLAS algorithm using a prototype immunomic network during infection [20]. The model consists of three state variables, which represent immune activation of three distinct T-cell populations. We assume that the dynamic activity of the various T-cell populations on the model are measure through time series of LC-MS measurements of the corresponding cytokines (interferon-gamma specific to CD4+ T helper cells, interferon-gamma specific to CD8+ cytotoxic T cells, and interleukin-10 specific to the CD4+ regulator T cells).

Figure 2(a) depicts the model, which consists of a Boolean network with three nodes, labeled "A", "B", and "C". The interaction parameters $a_{i,j}$ can be read from the figure. For example, node "C" is activated by node "A" and inhibited by node "B". These interactions can be represented as $a_{31} = 1, a_{32} = -1, a_{33} = 0$. Figure 2(b) depicts the resulting state-space. From the state space, we can see that these states are partitioned into two basins of attraction: the first one corresponds to a single attractor, whereas the second one consists of an attractor cycle. However, the two behaviors of the system is only depends on the state of node "A". If it is not expressed (there is no "helper T-cell" response), then the system will always tend to the resting single-state attractor 000. If "A" is expressed (there is help), then the activity of the system corresponds to that of a attractor cycle with the effector reponse being turned on and off cyclically.

In this experiment, We assume negative regulation biases, $b_i = -1/2$, for $i = 1, 2, 3$ and the synthetic data is generated based on LC-MS model with parameters $\mathscr{k} = 4, \vartheta = 100, \zeta =$

Authorized licensed use limited to: Texas A M University. Downloaded on October 24,2020 at 21:29:19 UTC from IEEE Xplore. Restrictions apply.

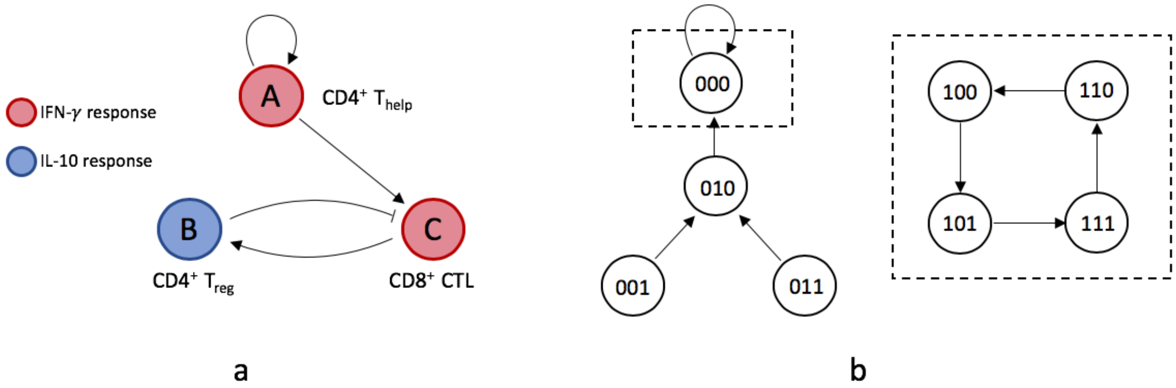


Fig. 2. Example of a simple Boolean network model of immunomic interactions during response to infection, consisting of three nodes A, B, and C; node A is a promoter, B is a suppressor, while node C produces the effector response, while also promoting suppression of B (negative feedback). a. Network wiring diagram and transition rules. b. Basins of attraction in state-space, with attractors indicated by dashed rectangles.

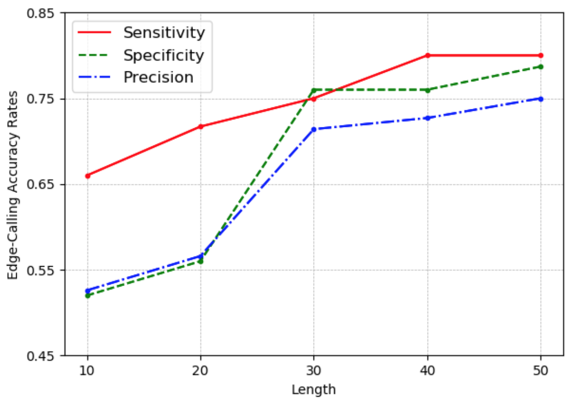


Fig. 3. Average edge-calling accuracy rates.

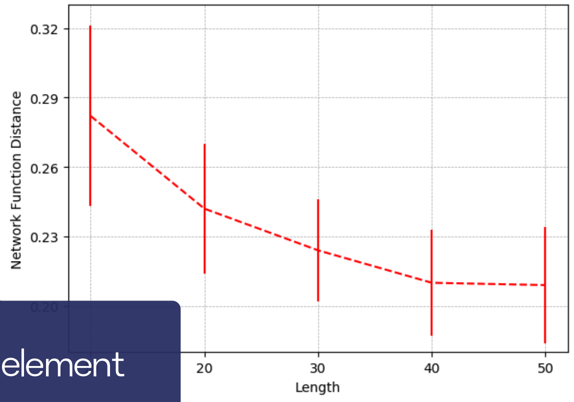


Fig. 4. Average network function distance.

pdfelement
The Trial Version

$3, and\varphi = 0.01$. Predefined interval ranges for estimation are $\Bbbk \in [2,6], \vartheta \in [80,120], \zeta \in [1,5], \varphi \in [0.01,0.2]$. Average edge-calling accuracy rates and network function distances obtained over 10 repetitions of the experiment are displayed in Figure 3 and 4. In Figure 3, we can see that as the time series length increases, the algorithm is both sensitive and specific, with high precision, capturing well the topology of the network. In figure 4 we can see that, in addition to capturing the network edges well, the proposed algorithm can correctly identify the regulatory functions, which controls the system dynamics, as time series length increases.

5. CONCLUSION

In this paper, we describe an extension of the the PALLAS algorithm by introducing a novel Approximate Bayesian Computation - Sequential Monte Carlo sampling (ABC-SMC) algorithm to handle the intractability of observational models of complex data, such as Liquid-Chromatography Mass-Spectrometry (LC-MS) proteomic expression data. The algorithm combines ABC-SMC and particle swarms intelligence to infer the underlying Boolean network. In this paper, we consider the specific case of LC-MS time series data, but the methodology can be modified to accommodate other data modalities. Numerical experiments with synthetic LC-MS cytokine measurements of a prototype immunomic network demonstrate that the proposed algorithm can capture both topological and dynamical system properties accurately.

6. REFERENCES

[1] Ariel Jaimovich, Gal Elidan, Hanah Margalit, and Nir Friedman, “Towards an integrated protein–protein inter-

Authorized licensed use limited to: Texas A M University. Downloaded on October 24,2020 at 21:29:19 UTC from IEEE Xplore. Restrictions apply.

action network: A relational markov network approach," *Journal of Computational Biology*, vol. 13, no. 2, pp. 145–164, 2006.

[2] Takashi Ito, Tomoko Chiba, Ritsuko Ozawa, Mikio Yoshida, Masahira Hattori, and Yoshiyuki Sakaki, "A comprehensive two-hybrid analysis to explore the yeast protein interactome," *Proceedings of the National Academy of Sciences*, vol. 98, no. 8, pp. 4569–4574, 2001.

[3] Peter Uetz, Loic Giot, Gerard Cagney, Traci A Mansfield, Richard S Judson, James R Knight, Daniel Lockshon, Vaibhav Narayan, Maithreyan Srinivasan, Pascale Pochart, et al., "A comprehensive analysis of protein–protein interactions in saccharomyces cerevisiae," *Nature*, vol. 403, no. 6770, pp. 623–627, 2000.

[4] Karthik Raman, "Construction and analysis of protein–protein interaction networks," *Automated experimentation*, vol. 2, no. 1, pp. 2, 2010.

[5] Saad Haider and Ranadip Pal, "Boolean network inference from time series data incorporating prior biological knowledge," *BMC genomics*, vol. 13, no. S6, pp. S9, 2012.

[6] Yang Hu, Ying Zhang, Jun Ren, Yadong Wang, Zhenzhen Wang, and Jun Zhang, "Statistical approaches for the construction and interpretation of human protein-protein interaction network," *BioMed research international*, vol. 2016, 2016.

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

[8] Yukun Tan, Fernando B Lima Neto, and Ulisses Braga-Neto, "Pallas: Penalized maximum likelihood and particle swarms for inference of gene regulatory networks from time series data," *bioRxiv*, 2020.

[9] Ulisses Braga-Neto, "Optimal state estimation for boolean dynamical systems," in *2011 Conference Record of the Forty Fifth Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*. IEEE, 2011, pp. 1050–1054.

[10] 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.

[11] Tina Toni, David Welch, Natalja Strelkowa, Andreas Ipsen, and Michael PH Stumpf, "Approximate bayesian computation scheme for parameter inference and model selection in dynamical systems," *Journal of the Royal Society Interface*, vol. 6, no. 31, pp. 187–202, 2009.

[12] Youting Sun, Ulisses Braga-Neto, and Edward R Dougherty, "A systematic model of the lc-ms proteomics pipeline," *BMC genomics*, vol. 13, no. S6, pp. S2, 2012.

[13] Yuichi Taniguchi, Paul J Choi, Gene-Wei Li, Huiyi Chen, Mohan Babu, Jeremy Hearn, Andrew Emili, and X Sunney Xie, "Quantifying e. coli proteome and transcriptome with single-molecule sensitivity in single cells," *science*, vol. 329, no. 5991, pp. 533–538, 2010.

[14] Ajay Jasra, Sumeetpal S Singh, James S Martin, and Emma McCoy, "Filtering via approximate bayesian computation," *Statistics and Computing*, vol. 22, no. 6, pp. 1223–1237, 2012.

[15] Pierre Del Moral, Arnaud Doucet, and Ajay Jasra, "An adaptive sequential monte carlo method for approximate bayesian computation," *Statistics and Computing*, vol. 22, no. 5, pp. 1009–1020, 2012.

[16] Thomas A Dean, Sumeetpal S Singh, Ajay Jasra, and Gareth W Peters, "Parameter estimation for hidden markov models with intractable likelihoods," *Scandinavian Journal of Statistics*, vol. 41, no. 4, pp. 970–987, 2014.

[17] Scott A Sisson, Yanan Fan, and Mark M Tanaka, "Sequential monte carlo without likelihoods," *Proceedings of the National Academy of Sciences*, vol. 104, no. 6, pp. 1760–1765, 2007.

[18] Yukun Tan, Fernando B Lima Neto, and Ulisses Braga Neto, "Inference of gene regulatory networks by maximum-likelihood adaptive filtering and discrete fish school search," in *2018 IEEE 28th International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, 2018, pp. 1–6.

[19] Edward R Dougherty, "Validation of inference procedures for gene regulatory networks," *Current genomics*, vol. 8, no. 6, pp. 351–359, 2007.

[20] Ulisses M Braga-Neto and Ernesto TA Marques Jr, "From functional genomics to functional immunomics: new challenges, old problems, big rewards," *PLoS computational biology*, vol. 2, no. 7, 2006.

pdfelement
The Trial Version

Authorized licensed use limited to: Texas A M University. Downloaded on October 24,2020 at 21:29:19 UTC from IEEE Xplore. Restrictions apply.