Ditzler *et al. BioData Mining* (2019) 12:5
https://doi.org/10.1186/s13040-019-0192-1

BioData Mining

RESEARCH Open Access

Approximate kernel reconstruction for time-varying networks

Gregory Ditzler[1]*, Nidhal Bouaynaya[2], Roman Shterenberg[3] and Hassan M. Fathallah-Shaykh[4]

*Correspondence:
ditzler@email.arizona.edu
[1]Department of Electrical and Computer Engineering, University of Arizona, Tucson, AZ, USA
Full list of author information is available at the end of the article

Abstract

Background: Most existing algorithms for modeling and analyzing molecular networks assume a static or time-invariant network topology. Such view, however, does not render the temporal evolution of the underlying biological process as molecular networks are typically "re-wired" over time in response to cellular development and environmental changes. In our previous work, we formulated the inference of time-varying or dynamic networks as a tracking problem, where the target state is the ensemble of edges in the network. We used the Kalman filter to track the network topology over time. Unfortunately, the output of the Kalman filter does not reflect known properties of molecular networks, such as sparsity.

Results: To address the problem of inferring sparse time-varying networks from a set of under-sampled measurements, we propose the Approximate Kernel RecONstruction (AKRON) Kalman filter. AKRON supersedes the Lasso regularization by starting from the Lasso-Kalman inferred network and judiciously searching the space for a sparser solution. We derive theoretical bounds for the optimality of AKRON. We evaluate our approach against the Lasso-Kalman filter on synthetic data. The results show that not only does AKRON-Kalman provide better reconstruction errors, but it is also better at identifying if edges exist within a network. Furthermore, we perform a real-world benchmark on the lifecycle (embryonic, larval, pupal, and adult stages) of the *Drosophila Melanogaster*.

Conclusions: We show that the networks inferred by the AKRON-Kalman filter are sparse and can detect more known gene-to-gene interactions for the *Drosophila melanogaster* than the Lasso-Kalman filter. Finally, all of the code reported in this contribution will be publicly available.

Keywords: Time-varying network, Compressive sensing, Gene regulatory, Gene regulatory networks

Background

Understanding the dynamical behavior of living cells from their complex genomic regulatory networks is a challenge posed in systems biology; yet it is one of critical importance (i.e., morphogenesis). Gene expression data can be used to infer, or reverse-engineer, the underlying genomic network to analyze the interactions between the molecules. Unfortunately, most of the existing work on reverse-engineering genomic regulatory networks estimates one single static network from all available data, which is often collected during different cellular functions or developmental epochs. The idea that molecular networks are remodeled as a function of time and stage is well understood; this conclusion is supported by the developmental networks of sea urchin embryos [1]. Throughout a cellular process, such as cancer progression or anticancer therapy, there may exist multiple underlying "themes" that determine the functionalities of each molecule and their relationships to others, and such themes are dynamic. In signal processing terms, summarizing gene expression data, that comes from different cellular stages, into one network would be similar to characterizing a non-stationary signal by its Fourier spectrum. Biologically, static networks cannot reveal regime-specific or key transient interactions that lead to biological changes.

One of the challenges of inferring a time-varying network is that there are only a few observations available at each time point. This small sample size is amplified by the high dimension of every sample, leading to a small n large p problem (i.e., more variables than observations). In particular, the system is under-determined. However, exploiting the fact that molecular networks are sparse, one can use compressive sensing to find a solution. Compressive sensing is concerned with the optimal reconstruction of a sparse signal from an under-determined linear system [2, 3]. Under-determined systems are quite common in computational biology/ecology, and the application of compressive sensing to solve these under-determined systems in nature has been a popular solution [4–6]. Compressive sensing theory states that, under the restricted isometry property (RIP), the optimal sparsest solution of a linear system is equivalent to the minimum l_1-norm solution [2, 3]. Unfortunately, it is almost impossible to check whether a linear system satisfies the RIP condition. In general, the minimum l_1-norm solution can be far from the optimal sparse solution.

In our previous work [4], we addressed the problem of under-sampled sparse systems by proposing a new energy-weighted likelihood function that ensures the convergence of the likelihood function for under-determined systems with unknown covariance. The approach was coined Small sample MUltivariate Regression with Covariance estimation (SMURC) and was applied to infer the wing-muscle gene regulatory networks of the Drosophila Melanogaster during the four phases of its development [4]. However, the estimated networks at every epoch used only the data in the corresponding epoch. In particular, the larval network ignored all the measurements in the previous embryonic phase, and so was the case for the subsequent stages. Other research efforts have been proposed to address the problem of recovering time-varying gene regulatory networks by using dynamic Bayesian models [7], non-parametric Bayesian regression [8], and random graph models [9].

In this contribution, we introduce a new approach to modeling sparse time-varying networks and their applications to gene regulatory networks that are based on our recent work [10, 11]. We start by projecting the Kalman solution onto an "approximately sparse" space by using l_1-regularization. We further expand upon our previous work by using a Kalman smoother. We then explore the l_1-neighborhood for sparser solutions by leveraging our recent compressive sensing technique known as *Kernel RecONstruction* (KRON) [12]. KRON recovers the optimal sparsest solution whether the RIP condition is satisfied or not. However, KRON's computational complexity is still exponential in the number of parameters p. We, therefore, advance Approximate KRON (AKRON) [11], which builds growing neighborhoods of the l_1 solution that moves towards the optimal sparsest solution and eventually reaches it. The size of the neighborhood is tunable depending on the computational resources available. We derive theoretical bounds of optimality. The AKRON Kalman filter is validated on synthetic and real-world data sets.

The state-space model and Kalman filter

Following the works in [4, 13], we model the network dynamics using a state space model. The system equation is given by a random walk model, which reflects a lack of prior knowledge of the network topological changes. The observation equation is given by a first-order differential equation, whose parameters reflect the strength and sign of interactions (positive and negative are activating and repressing, respectively) [14]. The state space model of the incoming edges, $\mathbf{a}_i \in \mathbb{R}^p$, for gene i can be shown to be [13]

$$\mathbf{a}_i(k+1) = \mathbf{a}_i(k) + \boldsymbol{w}_i(k). \tag{1}$$

$$\mathbf{y}_i(k) = \mathbf{X}^{\mathrm{T}}(k)\, \mathbf{a}_i(k) + \boldsymbol{v}_i(k), \tag{2}$$

where $i = 1, \cdots, p$ and p is the number of genes. $\boldsymbol{X}(k) \in \mathbb{R}^{p\times n}$ is the gene expression matrix at time k. $\mathbf{y}_i(k)$ is the rate of change of expression of gene i at time k. $\boldsymbol{w}_i(k)$ and $\boldsymbol{v}_i(k)$ are the process and observation noise, respectively. These noise processes are assumed to be zero mean Gaussian noise processes with the known covariances $\boldsymbol{Q}_k$ and $\boldsymbol{R}_k$, respectively, and uncorrelated to the state vector $\boldsymbol{a}_i(k)$. The full connectivity matrix, $\boldsymbol{A}(k)$, can be recovered by simultaneous parallel recovery of its rows $\boldsymbol{a}_i^t(k)$ at every time instant k. Thus, we can process each gene in parallel. The Kalman filter can be used to track $\boldsymbol{a}(k)$ [13, 15]; however, this is only if the system is observable. The problem with using a Kalman filter in our setting is that the system is under-determined (i.e., more variables than equations, $p > n$). This problem, however, can be circumvented by taking into account the sparsity of the vector $\boldsymbol{a}_i(k)$. Since each gene in the genomic regulatory network is governed by only a small number of other genes, these networks are known to be sparse. Furthermore, we have also experimented with a Kalman Smoother that is applied after the Kalman filter. Note that the Kalman Smoother is optional. A Kalman smoother can reduce the covariance of the optimal estimate. We implemented Rauch et al.'s Kalman smoothing algorithm [16] and we compared AKRON-Kalman both with and without smoothing.

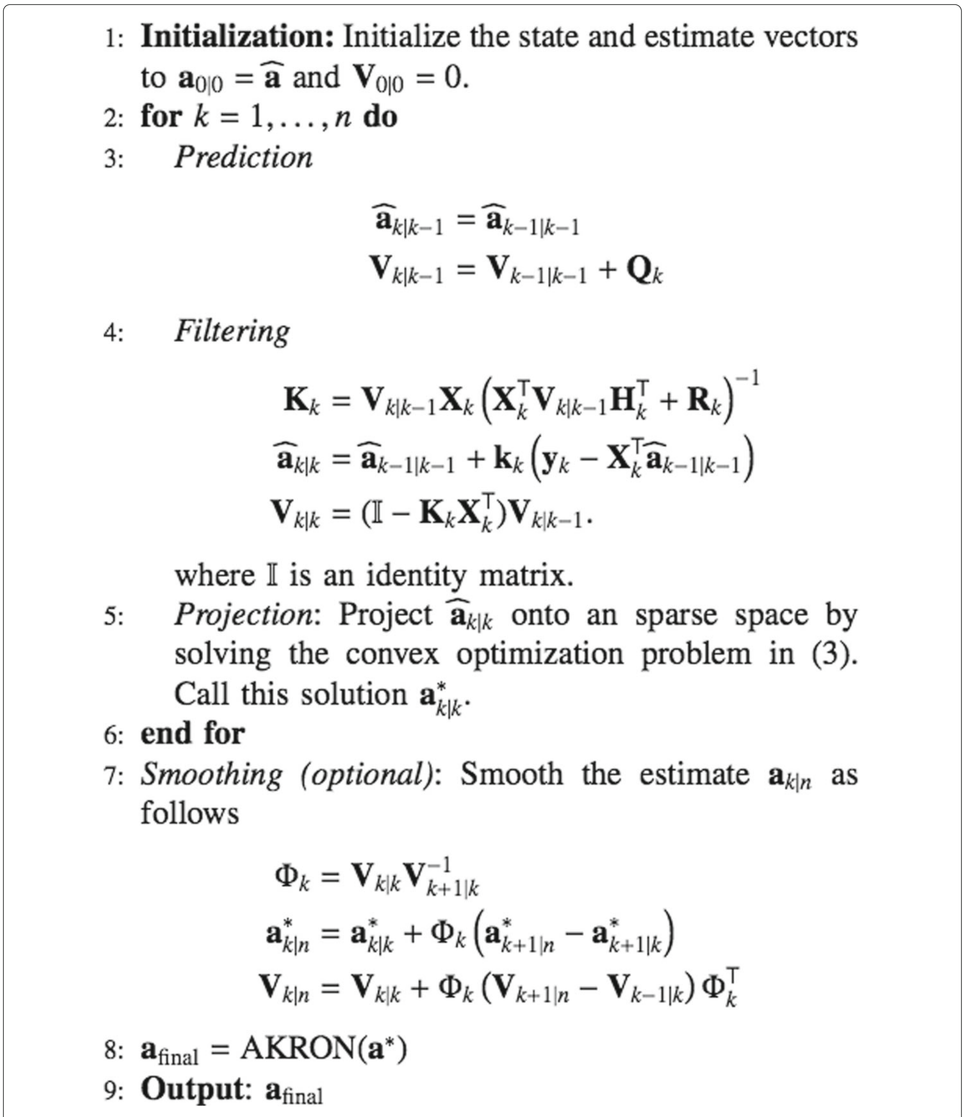
The pseudo code for the proposed Kalman filtering approach is shown in Fig. 1.

Constrained Kalman filtering

It is known that the connectivity of the edges in gene regulatory networks [13] is sparse. Unfortunately, the output of the Kalman filter is likely not going to be sparse. We first start by projecting the Kalman solution at time k, $\boldsymbol{a}_{k|k}$, onto the set of "approximately sparse" vectors by solving the following Lasso problem [17]:

$$\boldsymbol{a}^*_{k|k} = \arg\min_{\boldsymbol{a}\in\mathbb{R}^p} \left\{ (1-\alpha)\, \left\|\boldsymbol{a}_{k|k} - \boldsymbol{a}\right\|_2^2 + \alpha \|\boldsymbol{a}\|_1 \right\}, \tag{3}$$

where $\alpha \in [0, 1]$ controls the tradeoff between the Kalman estimate and sparsity. An α close to zero will result in a solution that is close to the Kalman estimate, but that may not be sparse. The opposite happens when α is close 1, which will produce a sparser solution, but may be far from the Kalman estimate. This is achieved by minimizing the reconstruction error (i.e., the first term in (3)).



```
Initialization: Initialize the state and estimate vectors
    to $\mathbf{a}_{0|0} = \widehat{\mathbf{a}}$ and $\mathbf{V}_{0|0} = 0$.
for $k = 1, \dots, n$ do
    Prediction
        $\widehat{\mathbf{a}}_{k|k-1} = \widehat{\mathbf{a}}_{k-1|k-1}$
        $\mathbf{V}_{k|k-1} = \mathbf{V}_{k-1|k-1} + \mathbf{Q}_k$
    Filtering
        $\mathbf{K}_k = \mathbf{V}_{k|k-1}\mathbf{X}_k \left(\mathbf{X}_k^\top \mathbf{V}_{k|k-1}\mathbf{H}_k^\top + \mathbf{R}_k\right)^{-1}$
        $\widehat{\mathbf{a}}_{k|k} = \widehat{\mathbf{a}}_{k-1|k-1} + \mathbf{k}_k \left(\mathbf{y}_k - \mathbf{X}_k^\top \widehat{\mathbf{a}}_{k-1|k-1}\right)$
        $\mathbf{V}_{k|k} = (\mathbb{I} - \mathbf{K}_k\mathbf{X}_k^\top)\mathbf{V}_{k|k-1}$.
    where $\mathbb{I}$ is an identity matrix.
    Projection: Project $\widehat{\mathbf{a}}_{k|k}$ onto an sparse space by
    solving the convex optimization problem in (3).
    Call this solution $\mathbf{a}^*_{k|k}$.
end for
Smoothing (optional): Smooth the estimate $\mathbf{a}_{k|n}$ as
    follows
        $\Phi_k = \mathbf{V}_{k|k}\mathbf{V}_{k+1|k}^{-1}$
        $\mathbf{a}^*_{k|n} = \mathbf{a}^*_{k|k} + \Phi_k \left(\mathbf{a}^*_{k+1|n} - \mathbf{a}^*_{k+1|k}\right)$
        $\mathbf{V}_{k|n} = \mathbf{V}_{k|k} + \Phi_k \left(\mathbf{V}_{k+1|n} - \mathbf{V}_{k-1|k}\right) \Phi_k^\top$
$\mathbf{a}_{\text{final}} = \text{AKRON}(\mathbf{a}^*)$
Output: $\mathbf{a}_{\text{final}}$
```

Fig. 1 AKRON-Kalman Tracker+Smoother

AKRON: a search for a sparser solution

Consider the following l_0-optimization problem, which finds the optimal sparsest solution in a linear under-determined system.

$$\begin{aligned} \boldsymbol{x}^* = & \arg\min_{\boldsymbol{x}\in\mathbb{R}^p} \|\boldsymbol{x}\|_0 \\ & \text{s.t. } \boldsymbol{\Phi x} = \boldsymbol{y} \end{aligned} \tag{4}$$

where $\|\cdot\|_0$ is the l_0-norm, which is defined as the support of the vector, $\boldsymbol{x} \in \mathbb{R}^p$, $\boldsymbol{y} \in \mathbb{R}^n$, and $\boldsymbol{\Phi} \in \mathbb{R}^{n\times p}$. We consider the scenario where $p \gg n$ and denote in the sequel $s = p - n$. Without loss of generality, $\boldsymbol{\Phi}$ is assumed to be full-rank. Compressive sensing theory [3] shows that, under the Restricted Isometry Property (RIP) condition on the matrix $\boldsymbol{\Phi}$, the l_1-norm solution is equivalent to the l_0-norm solution. Unfortunately, it is impossible to check if the RIP condition is satisfied for a given matrix. Despite this strict condition, l_1 has been routinely used to find a sparse solution in systems of the form (4).

The proposed Approximate Kernel RecONstruction (AKRON) is an approximation to computationally complex Kernel RecONstruction (KRON) problems [12]. KRON is able to achieve an exact solution to (4), but the algorithm becomes computationally expensive for typically $p > 15$. AKRON, detailed below, is introduced to balance the trade-off between the computational resources that are available and the accuracy of the reconstruction.

The Kalman filter estimate is first sparsified by incorporating l_1 regularization in (3) (line 5 of Fig. 1). However, the l_1 projection is not guaranteed to be the optimal sparsest solution. AKRON-Kalman filter (AKRON-KF) starts off from the l_1-regularized Kalman estimate in (3). Then, the $s = p - n$ smallest elements of the l_1 projection in (3) are set to zero. The logic behind this strategy is to use the l_1-projection to guess the position of the zeros in the optimal solution. Given that the kernel of the system matrix $\mathbf{\Phi}$ in (3) has dimension s, we know that if s zero locations are correctly set, then the optimal sparsest solution can be exactly found by solving the linear system in (4) [12].

Following this reasoning, AKRON finds a sparser solution by exploring δ-neighborhoods of the l_1-projection. The central idea behind AKRON's δ-neighborhoods is as follows: (i) find the indices with the $(s + \delta)$ smallest magnitudes of the l_1 solution, (ii) set exactly s of these indices to zero, (iii) re-solve the system $\mathbf{\Phi x} = \mathbf{y}$. All the possible $\binom{s+\delta}{s}$ combinations of the smallest elements in the solution of (3) are evaluated. This idea can also be viewed as a "perturbation" of the l_1 approximation to make it closer to the l_0-norm. The size of the neighborhood δ is tunable depending on the computational power available, and vary from 0 (l_1-approximation) to n (KRON, i.e., perfect reconstruction).

Example: To understand the AKRON algorithm and illustrate the importance of the δ-neighborhoods, we present a simple numerical example. Consider the following randomly generated noiseless system as in (4):

$$\mathbf{\Phi} = \begin{pmatrix} -0.4588 & 1.5977 & -0.8724 & -0.1121 & -1.3068 \\ 0.2942 & 3.0954 & -1.0530 & 0.3454 & 1.5257 \\ -0.1948 & -0.7558 & -0.9756 & 0.1549 & 0.9586 \end{pmatrix};$$

$$\mathbf{y} = \begin{pmatrix} -1.2316 & 1.1739 & 0.8135 \end{pmatrix}^T$$

The optimal ℓ_0-norm sparsest solution is given by

$$\mathbf{x}^* = \begin{pmatrix} 0 & 0 & 0 & -1.2372 & 1.04858 \end{pmatrix}^T \tag{5}$$

The ℓ_1 solution, which solves (4), is given by

$$\widehat{\mathbf{x}}_1 = \begin{pmatrix} 0.0 & -0.034 & 0.047 & 0.0 & 0.870 \end{pmatrix}^T$$

Clearly, the l_1-solution is not as sparse as the optimal solution and has incorrect zero locations. We have $n = 3, p = 5$ and thus $s = 2$. If we choose $\delta = 1$ the AKRON considers the $s + \delta = 3$-smallest magnitudes of $\widehat{\mathbf{x}}_1$, which are located at indices 1, 2 and 4. We set $s = 2$ locations to zero among these 3 indices. We consider all $\binom{s+\delta}{s} = \binom{3}{2} = 3$ combinations of two zeros in indices 1, 2 and 4 of $\widehat{\mathbf{x}}_1$. The combination of indices 1 and 2 set to zero leads to the sparsest optimal solution $\mathbf{x}^*$ in (5). Thus, in this case, the ℓ_1-norm solution is sub-optimal; but by considering a $\delta = 1$-neighborhood of this approximation, AKRON is able to exactly recover the sparsest optimal l_0-solution.

In the noisy case, where the constraint in (4) is replaced by $\|\boldsymbol{\Phi x} - \boldsymbol{y}\| \leq \epsilon$, with ϵ being a given noise threshold level, the neighborhood δ is chosen adaptively as follows: set the s smallest magnitudes of the l_1 solution to zero; compute the observation error $\|\boldsymbol{\Phi x} - \boldsymbol{y}\|$. If this error is smaller than the energy of the noise, we adopt this solution. Otherwise, the next smallest element is set to zero and the error is recalculated.

In the following propositions, we investigate under which assumptions on the entries of the l_1 solution and its closeness to the l_0 solution, will AKRON yield the optimal l_0 solution.

Proposition 1 *Consider the system in (4) with the optimal l_0-solution $\boldsymbol{x}^*$ having $k > s$ zeros. Consider the l_1-solution, $\boldsymbol{x}_1$, and assume that $\|\boldsymbol{x}_1 - \boldsymbol{x}^*\|_2 \leq \epsilon$. Let J denote the number of indices j such that*

$$|(\boldsymbol{x}_1)_j| \leq \frac{\epsilon}{\sqrt{k-s+1}}. \tag{6}$$

Then, by choosing $\delta \leq J - s$, AKRON yields the optimal l_0-solution.

Proof Let Θ and $\overline{\Theta}$ be the index sets of zero and non-zero entries in the l_0-solution $\boldsymbol{x}^*$, respectively. We have $|\Theta| = k$. We need to show that at least s indices in Θ are where (6) holds. To prove this fact, assume the opposite, i.e., at least $(k-s+1)$ indices in Θ are such that

$$|(\boldsymbol{x}_1)_j| > \frac{\epsilon}{\sqrt{k-s+1}}.$$

However, in this case, we have

$$\|\boldsymbol{x}_1 - \boldsymbol{x}^*\|_2 > \sqrt{k-s+1}\frac{\epsilon}{\sqrt{k-s+1}} = \epsilon,$$

which contradicts the assumption $\|\boldsymbol{x}_1 - \boldsymbol{x}^*\|_2 \leq \epsilon$. □

The following proposition derives an upper bound for $\delta \in \mathbb{N}_+$ when the non-zero elements of the optimal l_0-solution are bounded from below. We first need the following Lemma.

Lemma 1 *Consider the system in (4) with the optimal l_0-solution $\boldsymbol{x}^*$ and approximate l_1-solution $\boldsymbol{x}_1$. Denote by Θ and $\overline{\Theta}$ the index sets of zero and non-zero entries in $\boldsymbol{x}^*$, respectively. Assume that $\|\boldsymbol{x}_1 - \boldsymbol{x}^*\|_2 \leq \epsilon$. Let R be the number of indices $j \in \overline{\Theta}$ such that $|(\mathrm{x}_1)_j| \leq \epsilon$. If $\delta = R$, then AKRON yields the optimal solution.*

Proof To obtain the sparsest l_0-solution, it is sufficient to choose s zeros in "correct places", i.e., with indices in Θ. Recall that AKRON sets s out of the smallest-magnitude $(s + \delta)$ entries in $\boldsymbol{x}_1$ to zero. Therefore, AKRON will yield the optimal solution if out of these $(s+\delta)$ smallest-magnitude entries, there are at least s entries from Θ. But all entries from Θ have $|(\boldsymbol{x}_1)_j| \leq \epsilon$, for otherwise the assumption $\|\boldsymbol{x}_1 - \boldsymbol{x}^*\|_2 \leq \epsilon$ would be violated.

This means that only those entries from $\overline{\Theta}$, which also satisfy $|(\boldsymbol{x}_1)_j| \leq \epsilon$, could be chosen by AKRON, and there are only R of them. Hence $\delta = R$ will yield the optimal l_0-solution.

Lemma 1 provides a sufficient condition on δ for optimality of AKRON, namely, if $\delta = R$, then we are guaranteed the optimal l_0-solution. Since this condition is not necessary, we could reach optimality with $\delta \leq R$. □

Proposition 2 *Consider the system in (4) with the optimal l_0-solution $\boldsymbol{x}^*$ and approximate l_1-solution $\boldsymbol{x}_1$. Let $\overline{\Theta}$ be the set of indices of non-zero elements in $\boldsymbol{x}^*$. Assume that the non-zero entries of* x* *are bounded from below, i.e.,*

$$\left|\left(\boldsymbol{x}^*\right)_j\right| \geq \eta, \textit{ for all } j \in \overline{\Theta} \textit{ and for some } \eta > 0.$$

Assume further that $\|\boldsymbol{x}_1 - \boldsymbol{x}^*\|_2 \leq \epsilon$. If $\epsilon < \eta$, then $\delta \leq \left(\frac{\epsilon}{\eta-\epsilon}\right)^2$ yields the optimal l_0-solution. In particular, if $\epsilon < \frac{\eta}{2}$ then $\delta = 0$ suffices.

Proof Since $|(\boldsymbol{x}^*)_j| \geq \eta$ for $j \in \overline{\Theta}$, we have

$$\left\|\boldsymbol{x}_1 - \boldsymbol{x}^*\right\|_2 \geq (\eta - \epsilon)\sqrt{R} \geq (\eta - \epsilon)\sqrt{\delta},$$

where we used the fact that $\delta \leq R$ from Lemma 1. Thus, $\epsilon \geq \|\boldsymbol{x}_1 - \boldsymbol{x}^*\|_2 \geq (\eta - \epsilon)\sqrt{\delta}$, which completes the proof. □

Although Propositions 1 and 2 derive theoretical bounds for the choice of the neighborhood radius δ to recover the optimal sparsest solution, we found in our experiments below that relatively small values of δ are sufficient to achieve a balance between desired accuracy and computational complexity.

Results

In this section, we present an empirical analysis of the AKRON-KF and its smoother, including comparisons to other approaches proposed for detecting the relationships between different genes in a molecular network. The experiments include a number of carefully designed synthetic data sets, as well as a real-world data set, namely the fruit fly.

Overview of experimental protocols

Our experiments are conducted on real-world and synthetic data. The advantage of the synthetic data are that the ground truth networks are known; therefore, we can calculate different statistics about the reconstruction error of the network. Unfortunately, we do not have a clear view of the "ground truth" for real-world data. Therefore, we use findings from the life sciences that have studied these networks and were able to infer gene-to-gene relationships that are well established [18].

Our experiments make use of the following algorithms for a sparse reconstruction of a time-varying network:

- l_1-KF(S): This algorithm is the output of the Kalman filter with the l_1 projection applied to the state vector. The (S) indicates whether the smoother was implemented.
- AKRON-KF(S): This is the proposed approach using the output of l_1-KF(S) to seed AKRON. It is also implemented with and without the smoother.

Both of the above algorithms can reconstruct a network that represents the interactions between genes. We compute the true positive (TP), true negative (TN), false positive (FP) and false negative (FN) rates. These rates are summarized through accuracy (acc), sensitivity (sen), specificity (spe), and Matthew's correlation coefficient (mcc), which are defined below.

$$
\begin{aligned}
\text{acc} &= \frac{TP + TN}{TP + TN + FP + FN}, \\
\text{sen} &= \frac{TP}{TP + FN}, \\
\text{spe} &= \frac{TN}{TN + FP}, \\
\text{mcc} &= \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP) \cdot (TP + FN) \cdot (TN + FP) \cdot (TN + FN)}}.
\end{aligned}
$$

Matthew's correlation coefficient provides a more balanced statistic for examining the overall trade-offs between the different rates (i.e., TP, TN, FP, and FN).

Results on synthetic data

Synthetic time-varying networks are simulated to evaluate the efficacy of the proposed AKRON-KF(S) on data that we have complete control over. All results in this section are presented as the average over 25 monte carlo simulations. Averaging is performed because there could be a large degree of variation in the time-varying networks that are randomly generated.

First, we evaluate the impact of α in (3) on the estimation of $\boldsymbol{a}_{k|k}$ and the reconstruction errors of l_1-KF and AKRON-KF. The experiment is configured as follows: a 25-gene network evolves over four-time points by simulating a random walk; all networks being 85% sparse. At each time step, there are nine observations that are available. Figure 2 shows the effect of α on the reconstruction errors of AKRON-KF and l_1-KF. Clearly,

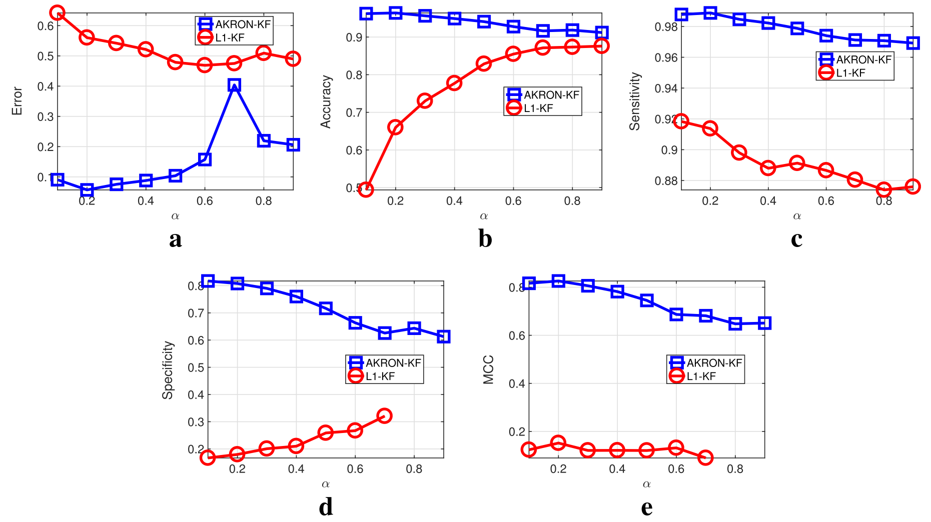


Fig. 2 Comparison of AKRON-KF and l_1-KF on a synthetic data with 25 genes over four time points and with 85% sparsity. All results are averaged over 25 monte carlo simulations. **a** Error. **b** Accuracy. **c** Sensitivity. **d** Specificity. **e** MCC

AKRON-KF is the better performer across the statistics that we collected. Thus, AKRON significantly improves the previous implementation of sparse Kalman filters for time-varying networks. Furthermore, AKRON-KF detects the location of the edges in the simulated networks (see Fig. 2b). Simply using the solution from l_1-KF for a small α is not enough to find the location of the zeros. In fact, α needs to be close to one to achieve a high accuracy at edge detection (i.e., (3) will place a large weight on the l_1 penalty and a small weigh on the error). Given these results, we choose $\alpha = 0.2$ for the remainder of the experiments since this value provides a reasonable trade-off between the different statistics that were assessed. Figure 2e shows that AKRON-KF is also superior to Lasso-KF as assessed by Matthew's correlation coefficient, which provides a balanced measure.

Second, we expand the synthetic experiments to evaluate the impact of δ in AKRON and the dimensionality of the kernel of $\boldsymbol{X}$, and we also evaluate the impact that the smoothing has on the reconstruction of the networks. We simulated three different kernel dimensions and values for δ. Table 1 shows the outcome of these experiments. The entries in the table are presented as A/B, where A and B are the results from AKRON-KF and l_1-KF, respectively. The table is also divided in half to separate the results for the Kalman filter and Kalman smoother. Similar to the first experiment, we observe that AKRON-KF typically outperforms l_1-KF in nearly every statistic and the results can be quite significant. Furthermore, systems with a large kernel dimension (i.e., large p small n) benefit significantly from a larger value of δ. For example, consider a network that is $9 \times 50 \times 4$. $\delta = 1$ provides a little reduction in error; however, increasing δ to 3, significantly reduces the error and increases the other statistics. Finally, we observe that smoothing improves the error of the system; however, we should note that this improvement comes at a computational cost.

Table 1 Results on simulated networks

		AKRON-KF			AKRON-KFS		
		$\delta = 1$	$\delta = 2$	$\delta = 3$	$\delta = 1$	$\delta = 2$	$\delta = 3$
err	$9 \times 11 \times 4$	2.27/131.42	1.83/119.78	2.74/128.54	0.68/133.65	1.38/122.99	1.21/128.05
	$9 \times 25 \times 4$	7.86/62.42	25.49/58.88	16.84/63.87	1.72/64.72	8.99/69.45	4.85/68
	$9 \times 50 \times 4$	17.06/18.06	10.92/18.25	4.94/18.81	7.11/22.02	12.59/21.35	2.74/22.47
acc	$9 \times 11 \times 4$	97.52/57.6	98.57/54.87	98.59/56.06	98.23/58.28	98.53/57.98	99.33/57.04
	$9 \times 25 \times 4$	96.31/63.25	97.49/65.17	98.17/65.43	97.82/77.12	98.84/77.23	99.46/78.54
	$9 \times 50 \times 4$	97.79/85.74	98.87/85.35	99.22/85.05	98.31/93.99	99.05/93.99	99.52/93.9
sen	$9 \times 11 \times 4$	98.52/67.25	98.92/65.69	98.62/66.22	99.1/65.89	98.68/65.29	99.34/64.98
	$9 \times 25 \times 4$	98.69/89.51	98.7/90.61	98.95/91.67	99.28/89.48	99.37/89.41	99.65/90.18
	$9 \times 50 \times 4$	99.32/98.73	99.59/98.78	99.67/98.72	99.48/98.27	99.63/98.24	99.78/98.23
spe	$9 \times 11 \times 4$	96/45.58	97.96/41.55	98.55/43.94	96.83/43.19	98.29/42.53	99.3/42.17
	$9 \times 25 \times 4$	82.55/17.32	89.28/17.79	92.33/19.42	88.46/20.71	94.97/20.01	98.05/22.18
	$9 \times 50 \times 4$	48.16/7.17	68.95/6.95	78.78/6.31	57.77/9.37	73.93/8.17	87.62/7.72
mcc	$9 \times 11 \times 4$	94.6/6.1	95.23/11.33	97.17/10.56	96.7/12.95	97.46/8.63	97.52/8.61
	$9 \times 25 \times 4$	84.74/12.08	89.1/13.45	92.88/11.88	90.76/7.92	97.76/8.76	95.47/10.86
	$9 \times 50 \times 4$	54.44/13.33	65.12/13.99	81.71/13.3	58.54/7.05	79.02/9.81	86.51/8.11

Each entry in the table reports two numbers separated by "/". The number on the left is the result of AKRON and the number on the right is the result of Lasso-KF. The simulation is performed with a 15% network density. The dimensions $A \times B \times C$ can be interpreted as A samples with B genes across C timestamps

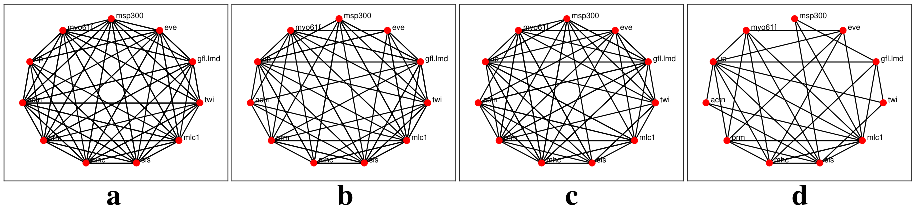


Fig. 3 Reconstructed networks for the l_1-KF across the four time stages. Edges in the network represent either the suppression of a gene or excitation of a gene, or one gene excites while the other suppresses. **a** Embryonic. **b** Larval. **c** Pupal. **d** Adult

Results on Flybase

The application of interest is the inference of the time-varying wing-muscle genomic network of the Drosophila Melanogaster (fruit fly). The Drosophila's microarray dataset originally consists of 4028 genes taken over 66 different time points [18]. The data includes 4 stages of the Drosophila's life: embryonic (samples 1 through 30), larval (samples 31 through 40), pupal (samples 41 through 58), and adulthood (samples 59 through 66). Flybase hosts a list of undirected gene interactions [19]. We set $\alpha = 0.2$ based on the experiments in the previous section for l_1-KF and AKRON-KF.

In this application, we considered a list of 11 genes that are responsible for the wing muscle development, which has been considered by many researchers before [7–9, 20]. The embryonic, pupal, and larval stages are undersampled to 9 observations in each stage that were used in the reconstruction of the 11-gene network in each developmental epoch. All 8 time points were used in the adulthood period. To summarize, the reconstruction of the connectivity matrix uses 9 samples in the embryonic, pupal, and larval developmental stages and 8 samples in the adulthood developmental stage. The 11 gene network was reconstructed throughout each of the four developmental stages using AKRON-KF and AKRON-KFS.

The networks reconstruction using the l_1-KF and AKRON-KF are shown in Figs. 3a-d and 4a-d, respectively. For clarity, the displayed networks do not show the strength of the interaction, only that there is an interaction detected by one of the algorithms. The AKRON-KF tracker results in clearly sparser networks than the l_1-KF. AKRON-KF was able to find all the connections reported in Flybase: (Actn,prm) appears in the embryonic, larval, and pupal stages, (Actn,up) appears in all four stages, (up,mhc) appears in the embryonic, larval, and pupal stages, (up,sls) appears in the embryonic, larval, and pupal stages and (sls,mhc) appears in the embryonic and larval stages.

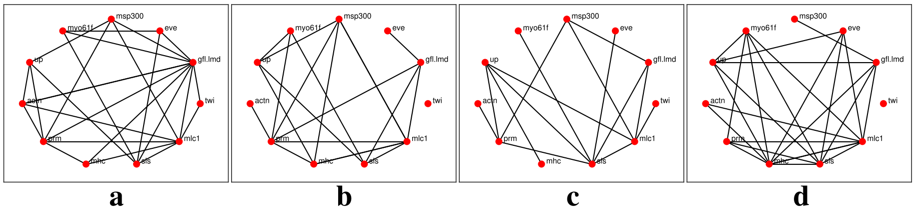


Fig. 4 Reconstructed networks for the AKRON-KF across the four time stages. Edges in the network represent either the suppression of a gene or excitation of a gene, or one gene excites while the other suppresses. **a** Embryonic. **b** Larval. **c** Pupal. **d** Adult

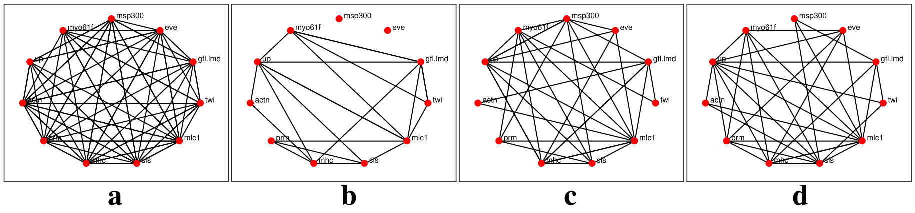


Fig. 5 Reconstructed networks for the l_1-KF with a smoother across the four time stages. Edges in the network represent either the suppression of a gene or excitation of a gene, or one gene excites while the other suppresses. **a** Embryonic. **b** Larval. **c** Pupal. **d** Adult

The other two connections appear through one medium gene (Actn,sls) appears in the embryonic, pupal and adulthood phases through one additional gene and (twin, eve) appears through one or more additional genes only in the embryonic and larval phases.

The networks reconstruction using the l_1-KF and AKRON-KF with smoothers are shown in Figs. 5a-d and 6a-d, respectively. The smoothing provides very similar network topologies to the ones without the smoother; however, we did observe that the networks were sparser in the larval stage (see Figs. 4b and 6b). Note that while Fig. 7 is considered the ground truth, the could exist relationships that have not yet been discovered. These statistics are shown in Table 2, which again shows the benefit of using the AKRON-KF over the l_1-KF.

Table 3 lists all previous algorithms that were applied to this genetic network. Only the AKRON-KF, l_1-KF [13], SMURC [4] and Dynamic Bayesian networks [7] considered time-varying networks; and, hence, were able to distinguish the different phases in the network. The other algorithms (minimum description length [20], random graph model [9], and nonparametric Bayesian regression [8]) assumed a stationary network, and hence it is not clear at which stage the detected connections develop. The AKRON-KF along with the l_1-KF are the only algorithms able to recover all known interactions and specify the developmental stage where these interactions occur. Although the l_1-KF also finds all reported interactions, the networks are denser (less sparse) than the AKRON-KF. With regard to the MDL results in Table 2, we have not reported the interations. The MDL authors in [20] inferred a single network, using all 66 time points, that characterizes the entire Drosophila's life cycle. In particular, the MDL approach is stationary and does not differentiate between the phases or time-varying epochs of the data. We wanted to report

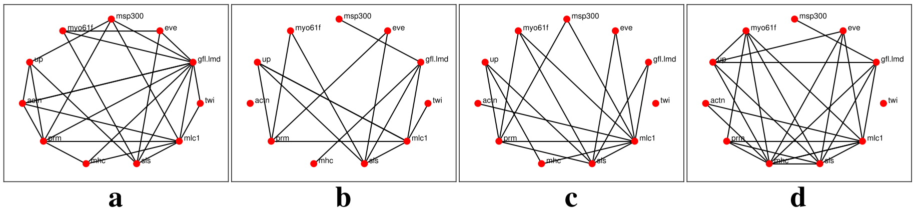


Fig. 6 Reconstructed networks for the AKRON-KF with a smoother across the four time stages. Edges in the network represent either the suppression of a gene or excitation of a gene, or one gene excites while the other suppresses. **a** Embryonic. **b** Larval. **c** Pupal. **d** Adult

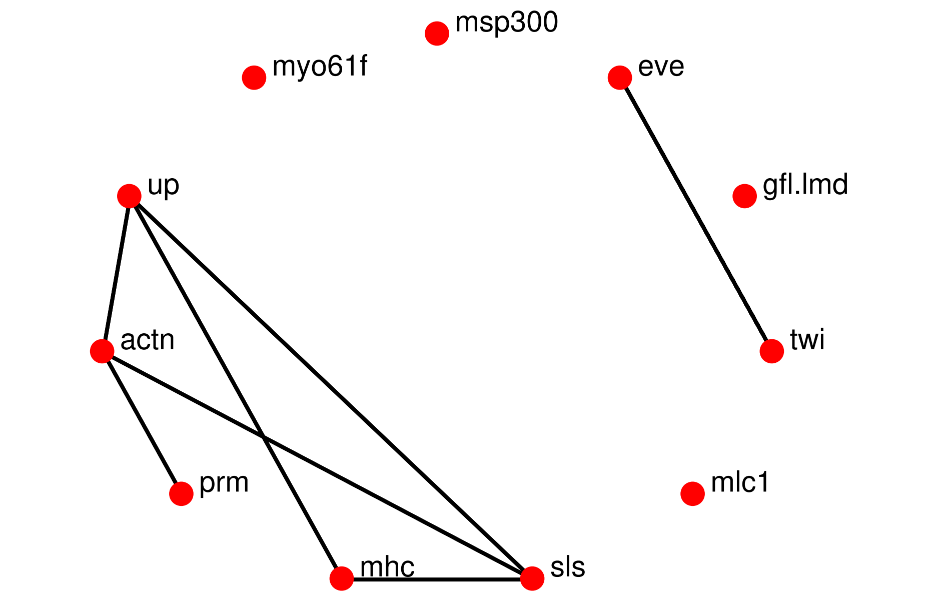


Fig. 7 The known undirected gene interactions in the Drosophila's 11-gene wing muscle network as determined from Flybase [18]

the MDL findings as they are in the literature. Notice that AKRON used less points and found more correct interactions.

Conclusion

In this work, we addressed the problem of inferring time-varying molecular networks as a tracking problem that can be solved using the Kalman filter. The major difficulty, however, is that there is not a sufficient number of observations at each time point, which makes the state-space model unobservable and the tracking senseless. Fortunately, molecular networks are known to be sparse because the dynamics of every gene are governed by only a small number of genes. By incorporating the sparsity condition, we show that the tracking problem becomes feasible.

We presented the AKRON Kalman filter, which builds on our previous work on the Lasso-Kalman filter (l_1-KF). Our proposed approach leverages the AKRON algorithm to find a sparser solution that is more representative of the ground truth. The proposed tracker/smoother first computes the output of l_1-KF; then explores growing neighborhoods of the l_1-projection to look for sparser solutions, eventually reaching the optimal sparsest estimate. The size of these neighborhoods is a tunable parameter that depends on

Table 2 Results of the AKRON-KF and l_1-KF on the real-world Drosophila Melanogaster data set

	Algorithm	acc	sens	spec	mcc
AKRON	$\delta = 1$	35.54	93.94	13.64	10.55
	$\delta = 2$	36.16	94.07	13.75	10.97
	$\delta = 3$	38.02	95.71	14.53	14.53
	$\epsilon = 0.0005$	49.59	90.35	13.28	5.67
	L_1-KF	38.43	88.69	11.71	0.59

Table 3 Detection of the known gene interactions in Flybase (E: embryonic, L: larval, P: pupal and A: adulthood)

	(*prm,Actn*)	(*sls,mhc*)	(*mhc,up*)	(*sls,Actn*)	(*sls,up*)	(*twi,eve*)	(*up,Actn*)
AKRON-KFS	✓ (E,L,A)	✓ (A)	✓ (L,P,A)	✓ (E,P,A)	✓ (E,L,P,A)	✓ (E,L)	✓ (E)
AKRON-KF	✓ (E,L,A)	✓ (A)	✓ (L,P,A)	✓ (E,P,A)	✓ (E,L,P,A)	✓ (E,L)	✓ (E)
LASSO-Kalman [13]	✓ (E,L,P)	✓ (E,L)	✓ (E,L,P)	✓ (E,L,P)	✓ (E,L,P)	✓ (E,L,P,A)	✓ (E,L,P,A)
SMURC [4]	✓ (A)	✓ (A)	✓ (L)	✓ (L)	✓ (E)	✓ (P)	×
MDL [20]	✓	✓	×	×	×	✓	×
Random graph model [9]	×	×	✓ (E,L,P,A)	✓ (P,A)	✓ (E,L,P,A)	×	×
Dyn. Bayes. netw. [7]	×	✓ (E,L,P,A)	×	×	×	×	×
Nonpar. Bayes. [8]	×	×	×	×	×	✓ (E)	×

the computational power available. AKRON-KF was benchmarked on synthetic and real-world data against l_1-KF. The results demonstrate that the proposed approach is better at recovering sparse time-varying networks than l_1-KF. Not only was the reconstruction error of the proposed approach lower than l_1-KF, but it was also better at detecting whether an edge exists in a network. AKRON-KF tracker was applied to infer the wing muscle gene regulatory network of the *Drosophila Melanogaster* during four developmental phases of its life cycle, and successfully identified all seven known interactions reported in Flybase. We should also note that our proposed approach will work for time-series networks that have more than four times steps and sparsity levels

Our future work includes applying the AKRON-KF to other types of data, particularly data related to different types of cancers to create a predictive network biomarker for clinical outcome. These ideas have applicability in translational clinical cancer research, basic cancer research, and in network-based drug discovery.

Acknowledgements
This work was supported by the National Science Foundation under Award Numbers NSF CCF-1527822 and NSF DUE-1610911.

Funding
This work was supported by the National Science Foundation (NSF) under CCF-1527822 and DUE-1610911. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the funding agencies.

Availability of data and materials
Code and reproducibles will be made available on Github if accepted for publication.

Authors' contributions
GCD, NCB and RS developed the AKRON Kalman filter and the theory of AKRON. GCD, NCB, RS and HMF-S participated in the design of the study. GCD and NCB implemented the study. GCD, NCB, RS and HMF-S analyzed and interpreted the data. GCD and NCB prepared the manuscript and revised it. All authors 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

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Author details

[1]Department of Electrical and Computer Engineering, University of Arizona, Tucson, AZ, USA. [2]Department of Electrical and Computer Engineering, Rowan University, Glassboro, NJ, USA. [3]Department of Mathematics, University of Alabama at Birmingham, Birmingham, AL, USA. [4]School of Medicine, University of Alabama at Birmingham, Birmingham, AL, USA.

Received: 10 August 2018 Accepted: 14 January 2019
Published online: 06 February 2019

References

1. Sethi AJ, Wikramanayake RM, Angerer RC, Range RC, Angerer LM. Crosstalk regulates endomesoderm segregation in sea urchin embryos. Science. 2012;335(6068):590–3.
2. Foucart S, Rauhut H. A mathematical introduction to compressive sensing. Birkhäuser. 2013.
3. Fornasier M, Rauhut H. Compressive sensing. In: Handbook of Mathematical Methods in Imaging; 2011.
4. Bayar B, Bouaynaya N, Shterenberg R. SMURC: High-dimension small-sample multivariate regression with covariance estimation. IEEE J Biomed Health Inform. 2016;21(2):573–81.
5. Koslicki D, Foucart S, Rosen G. WGSQuikr: fast whole-genome shotgun metagenomic classification. PLoS ONE. 2014;9(3):91784.
6. Koslicki D, Rosen G, Foucart S. Quikr: a method for rapid reconstruction of bacterial communities via compressive sensing. Bioinformatics. 2013;29(17):2096–102.
7. Robinson JW, Hartemink AJ. Learning non-stationary dynamic bayesian networks. J Mach Learn Res. 2010;11: 3647–80.
8. Miyashita H, Nakamura T, Y Ida TM, Kaburagi T. Nonparametic Bayes-based Heterogenious Drosophila Melanogaster Gene Regulatory Network Inference: T-process Regression. In: International Conference on Artificial Intelligence and Applications; 2013. p. 51–8.
9. Go F, Hanneke S, Fu W, Xing EP. Recovering Temporally Rewiring Networks: A Model Based Approach. In: International Conference on Machine Learning; 2007. p. 321–8.
10. Carluccio V, Bouaynaya N, Ditzler G, Fathallah Shaykh HM. The AKRON-Kalman Filter for Tracking Time-Varying Networks. In: IEEE International Conference on Biomedical and Health Informatics; 2017.
11. Ditzler G, Bouaynaya N, Shterenberg R. AKRON: an Algorithm for Approximating Sparse Kernel Reconstruction. Signal Processing. 2018;144:265–270.
12. Bayar B, Bouaynaya N, Shterenberg R. Kernel Reconstruction: an Exact Greedy Algorithm for Compressive Sensing. In: International Workshop on Genomic Signal Processing and Statistics; 2014.
13. Khan J, Bouaynaya N, Fathallah-Shaykh H. Tracking of time-varying genomic regulatory networks with a lasso-kalman smoother. EURASIP J Bioinforma Syst Biol. 2014;1. https://link.springer.com/article/10.1186/1687-4153-2014-3.
14. Fathallah-Shaykh HM, Bona JL, Kadener S. Mathematical model of the drosophila circadian clock: Loop regulation and transcriptional integration. Biophys J. 2009;97(9):2399–40.
15. Xiong J, Zhou T. A kalman-filter based approach to identification of time-varying gene regulatory networks. PLoS ONE. 2013;8(10). https://www.ncbi.nlm.nih.gov/pubmed/24116005.
16. Rauch HE, Tung F, Striebel CT. Maximum likelihood estimates of linear dynamic systems. AIAA J. 1965;3(8):1445–50.
17. Tibshirani R. Regression shrinkage and selection via the lasso. J R Stat Soc. 1996;58(1):267–88.
18. Arbeitman M, Furlong E, Imam F, Johnson E, Null B, Baker B, Krasnow M, Scott M, Davis R, White K. Gene expression during the life cycle of drosophila melanogaster. Science. 2002;297(5590):2270–5.
19. Marygold S, Leyland PC, Seal RL, Goodman JL, Thurmond J, Strelets VB, Wilson RJ. Flybase: improvements to the bibliography. Nucleic Acids Res. 2013;41:751–7.
20. Zhao W, Serpedin E, Dougherty E. Inferring gene regulatory networks form time series data using minimum description length principle. Bioinformatics. 2006;22(17):2129–35.