Supplementary material 1:

In this part, we infer the GRN model under the other two non-Gaussian noises: the flicker noise and the noise with indeterminate covariance.

*1) Flicker Noise*

The probability density function of flicker noise can be written as:

 (1)

It is generated by Gaussian noise *fG*(*x*) and Laplace noise *fL*(*x*). In this experiment, the covariance matrix of internal noise is *Q* = 0.01 \**I*, and the means of Gaussian noise and Laplace noise are *μ*G*=*0*, μ*L*=*0, the variances of two noises arerespectively.

We infer the GRN model under flicker noise by four joint algorithms based on the E-CELL synthetic data. The GRN model inferred by GP and PF under flicker noise is:

 (2)

The identified parameters of GRN model using four joint algorithms under flicker noise are summarized in Table I. The models inferred by the four joint algorithms are solved by a fourth-order Runge-Kutta method, and the obtained time-series curves are shown in Fig.1.

The red solid line in the figure represents raw time series generated by E-cell, the blue dotted line represents the time-series data from the model inferred by GP and PF, the black line represents the time-series data from the model inferred by GP and RKF, the green line represents the time-series data from the model inferred by GP and KF, the pink line represents the time-series data from the model inferred by GP and RLS. The time series generated from the model inferred by GP and PF are closer to the raw values, and this algorithm obtains a more accurate GRN model.

TABLE I

Identified Parameters by Four Joint Algorithm under Flicker Noise

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | true parameter | GP+PF | GP+RKF | GP+KF | GP+RLS |
|  | -10.32 | -10.37 | -10.47 | -10.21 | -10.56 |
|  | 9.72 | 9.57 | 9.11 | 8.63 | 8.76 |
|  | -17.5 | -17.49 | -14.03 | -15.26 | -13.10 |
|  | -9.72 | -9.70 | -9.10 | -9.13 | -8.50 |
|  | 17.5 | 17.07 | 14.84 | 14.83 | 14.16 |



Fig.1 Time series of the inferred models under flicker noise

*2)* *The Noise with Indeterminate Covariance*

Suppose the covariance of the noise is written as

, (3)

where *r* is a random number, and , *R* is the covariance of Gaussian noise.

We infer the GRN model under the noise with indeterminate covariance by four joint algorithms based on the E-CELL synthetic data. The GRN model inferred by GP and PF under the noise with indeterminate covariance is:

 (4)

The identified parameters of GRN model using four joint algorithms under the noise with indeterminate covariance are summarized in Table II. The models inferred by the four joint algorithms are solved by a fourth-order Runge-Kutta method, and the obtained time-series curves are shown in Fig.2.

The red solid line in the figure represents raw time series generated by E-cell, the blue dotted line represents the time-series data from the model inferred by GP and PF, the black line represents the time-series data from the model inferred by GP and RKF, the green line represents the time-series data from the model inferred by GP and KF, the pink line represents the time-series data from the model inferred by GP and RLS. The time series generated from the model inferred by GP and PF are closer to the raw values, and this algorithm obtains a more accurate GRN model.

TABLE II

Identified Parameters by Four Joint Algorithm under the Noise with Indeterminate Covariance

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | true parameter | GP+PF | GP+RKF | GP+KF | GP+RLS |
|  | -10.32 | -10.36 | -10.17 | -10.34 | -9.96 |
|  | 9.72 | 9.79 | 8.34 | 9.14 | 8.39 |
|  | -17.5 | -17.68 | -14.13 | -15.47 | -12.95 |
|  | -9.72 | -9.70 | -9.04 | -9.28 | -8.17 |
|  | 17.5 | 17.93 | 14.76 | 16.90 | 14.43 |



Fig.2 Time series of the inferred models under the noise with indeterminate covariance

Supplementary material 2:

The complexity of an algorithm is usually evaluated with the running time and memory space. This part mainly discusses the time complexity of the proposed method.

Generally, the basic operations associated with time complexity are addition, subtraction, multiplication, division, sorting and so on. The time complexity *T* of an algorithm can be measured in terms of the number of real floating point operations:

 (5)

Where ，，，，and  are the number of real floating point operations for addition, subtraction, multiplication, division, comparison and exchange operations respectively, the parameters *A*, *S*, *M*, *D*, *C* and *E* represent the numbers of above six operations, respectively.

The complexity of genetic programming depends on the generations, the population size, the tree-depth and the fitness value. We assume that the number of generation is *H*, population size is *P*, maximum tree-depth of individual is *K*, and the worst case complexity of applying the fitness function to one individual is *X*, in this case, the complexity of genetic programming is.

*A. The Complexity of GP and RLS*

For RLS, suppose *m* is the number of weight coefficients, the steps for inferring parameters by RLS are as follows:

1) Calculate the gain vector 

 (6)

2) Calculate the prior estimation error 

 (7)

3) Update the inverse correlation matrix 

 (8)

4) Update the parameter matrix 

 (9)

The computational complexity for calculating the gain vector is, the computational complexity for calculating the prior estimation error is, the computational complexity for updating the inverse correlation matrix is, the computational complexity for updating the parameter matrix is, based on the above analyses, the computational complexity of RLS algorithm is:

 (10)

So the complexity of RLS algorithm is. The RLS algorithm needs to repeatedly execute *L* times to identify the parameters in the model after GP deduces the GRN model structure on each population, therefore, the complexity of GP and RLS is.

*B. The Complexity of GP and KF*

We need to calculate the inverse matrix for amatrix when the parameters are identified by KF. The complexity for calculating the inverse matrix is.

Suppose *m* is the number of weight coefficients, and *b* is the size of the estimation values. The steps for identifying the parameters by KF are as follows:

1) Calculate the prior estimates of state:

 (11)

2) Calculate the variance for estimated value and true value, and assess the estimated error.

 (12)

3) Calculate the gain matrix：

 (13)

4) Update the estimation value:

 (14)

5) Update the estimation value of variance:

 (15)

The computational complexity for calculating the prior estimate of state is, the computational complexity for calculating the prior estimate of variance is. In order to get the gain matrix, we need to calculate the inverse matrix, so the complexity for calculating the gain matrix is.The computational complexity for updating estimation value is. Considering that the multiplication  in equation (12) might be sparse matrix, so the complexity for updating the estimation value of variance is at least. Therefore, the complexity of KF algorithm is. The KF needs to repeatedly execute *L* times to identify the parameters in the model after GP deduces the GRN model structure on each population, so the time complexity of GP and KF is.

*C. The Complexity Analysis of GP and RKF*

Suppose m is the number of weight coefficients, and b is the size of estimated values. The steps for identifying the parameters by RKF are as follows:

1) Find the gain matrix of standard Kalman filter in the steady state.

2) Generate all the robust Kalman filter gain in the range of , (). Calculate the performance index of Robust Kalman filter gain. Then generate the next generation population by reproduction, crossover and mutation.

3) If cannot be improved or the loops reach the maximum iteration number, stop it.

The RKF is the algorithm to optimize the KF by the genetic algorithm (GA). The gain of KF is obtained at first in each generation, and the complexity is  at this time. For GA, suppose *i* is the number of iteration, *c* is the size of individual, *d* is the worst case for applying the fitness function to each individual, and RKF needs to repeatedly execute *L* times to identify the parameters in the model, so the time complexity of GP and RKF is .

The complexity of the four joint algorithms and the running time in the E-cell simulation experiment are shown in Table III.

TABLE III

Complexity of Joint Algorithm

|  |  |  |
| --- | --- | --- |
| Joint algorithms | Complexity | Run time(/s) |
| GP+RLS |  | 2.2 |
| GP+KF |  | 2.3 |
| GP+RKF |  | 10034.2 |
| GP+PF |  | 43.5 |