

# Constructing backbone network by using tinker algorithm

Zhiwei He<sup>1,†</sup>, Meng Zhan<sup>2</sup>, Jianxiong Wang<sup>3</sup>, Chenggui Yao<sup>1</sup>

<sup>1</sup>Department of Mathematics, Shaoxing University, Shaoxing 312000, China

<sup>2</sup>State Key Laboratory of Advanced Electromagnetic Engineering and Technology, School of Electrical and Electronic Engineering, Huazhong University of Science and Technology, Wuhan 430074, China

<sup>3</sup>College of Science, Hubei University of Technology, Wuhan 430068, China

Corresponding author. E-mail: <sup>†</sup>hezhiwei10@mails.ucas.ac.cn

Received October 8, 2016; accepted December 2, 2016

Revealing how a biological network is organized to realize its function is one of the main topics in systems biology. The functional backbone network, defined as the primary structure of the biological network, is of great importance in maintaining the main function of the biological network. We propose a new algorithm, the tinker algorithm, to determine this core structure and apply it in the cell-cycle system. With this algorithm, the backbone network of the cell-cycle network can be determined accurately and efficiently in various models such as the Boolean model, stochastic model, and ordinary differential equation model. Results show that our algorithm is more efficient than that used in the previous research. We hope this method can be put into practical use in relevant future studies.

**Keywords** biological network, backbone network, tinker algorithm, mathematical model

**PACS numbers** 07.05.Kf, 07.05.Tp

## 1 Introduction

One of the main problems in systems biology is revealing the relationship between functions and components (proteins, nucleic acids, polysaccharides) as well as their interactions in biological networks [1–4]. With the contribution of experimental physiologists, the precise details of components, interactions, and full map of the biological regulatory network have been presented. To analyze the function of the network, many mathematical methods have been introduced, such as master equations model, the Monte Carlo method [5], the stochastic model [6–8], ordinary differential equation (ODE) model [9–13], and the Boolean network method [14–19]. These mathematical models not only help us understand how the biological components work, but also provide many useful and precise predictions [20, 21].

An important application of the mathematical models is to investigate how the biological network is robustly designed for realizing its function. Creatures should be well adapted to the environment after the long process

of evolution. Sometimes, the adaptability is expressed as the high robustness of the function and structure against perturbations [22–24]. One of the well-studied examples is the research on the cell-cycle network that describes the metabolic process [25–27]. Li *et al.* have investigated the global dynamical property and stability of the cell-cycle network of budding yeast in the Boolean network model [15] and the ODE model [28], and revealed that the cell-cycle pathway is an attracting trajectory of the dynamics and rather stable against perturbations. They have also studied the robustness of the network in the stochastic model from the potential landscape perspective [8]. Wang *et al.* have also carried out many works from the potential landscape perspective in the study of the cell-cycle network using the stochastic model and the ODE model [17–19]. Their results have demonstrated that the network has the least dissipation cost design for its robustness. These studies that use various models help us understand the biological network's robustness further.

The biological network is robustly designed as a whole, but the functional role of local edges would be diverse. Several works of Wang *et al.*, on the basis of the Boolean network model, have revealed that the functional backbone motif structures of the cell-cycle network of bud-

\*Special Topic: Soft-Matter Physics and Complex Systems (Ed. Zhi-Gang Zheng). arXiv: 1612.05749.

ding and fission yeast, where the functional backbone motif is the minimal subnetwork (with the fewest edges) that can maintain the main function of the full network [17–19, 29]. Afterward, Yang *et al.* constructed a simple cell-cycle network of cancer cells and found that the backbone motif structure also exists in this network [30]. From this perspective, a biological network can be viewed as the combination of two components: the functional backbone motif (backbone network), which carries out the main function of the network; and the remaining edges (supplementary edges), which can enhance the robustness of the network. This classification could not only help us understand why a network still works normally after some regulatory edges are removed, but also provide us considerable information to control the network or even design a new artificial network in further experiments. Therefore, finding the backbone network is a very important task.

The method used by Wang *et al.* is rather good in finding the backbone network, but it has some limitations. The method is process based: firstly, the Boolean network equations can be established and simplified with the known biological process; then, the simplified equations are solved by enumeration, and all the solutions are called the possible networks of the biological process; and finally, the backbone network is selected from the possible networks with the known biological network. However, this method can be applied only to the Boolean network model, and is not adaptable to the other models previously mentioned. Moreover, the process of solving equations by enumeration may cost considerable time in many situations. Therefore, just as in the case of studies on the robustness of the cell-cycle network, an efficient method that can be applied to determine the backbone network in different models is considered to be very necessary.

In this paper, we suggest a new method to determine the backbone network of the biological network based on the idea of tinkering. The idea of tinkering was firstly used in evolution by François Jacob, who considered the process of evolution as a tinker's tinkering [31]. Then, Alon *et al.* proposed that the tinker is a good engineer for some biological networks' motifs, which are well designed to enhance the network robustness [2, 32–34]. Here, based on the idea of tinkering, we propose a simple algorithm, the tinker algorithm, with which the backbone network of a given network can be determined by tinkering the edges of the network one by one. We apply it to different models such as the Boolean network model, stochastic model, and ODE method. Simulation results show that the algorithm works satisfactorily.

The paper is organized as follows. In Section 2, the description of the tinker algorithm is provided. In Section 3, the simulation results of the tinker algorithm ap-

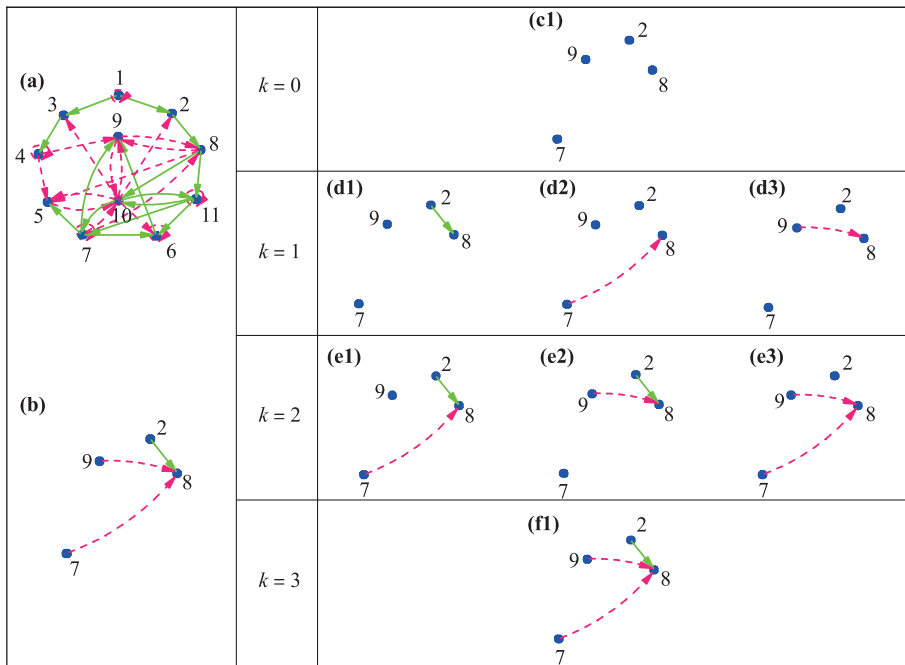
plied in the Boolean network model, stochastic model, and ODE method are presented in detail. Finally, our conclusion and discussions are given.

## 2 Methods

A biological network with  $N$  interacting molecules can be expressed as an adjacency matrix  $A = \{a_{ij}, i, j = 1, 2, \dots, N\}$ , where each node denotes a molecule. The value of  $a_{ij}$  ( $j \neq i$ ) can be positive, negative, or zero, depending on whether node  $j$  activates, inhibits, or does not interact with node  $i$ , respectively. The value of  $a_{ii}$  can also be positive, negative, or zero, representing that node  $i$  has self-degradation, self-stimulation, or no action back on itself, respectively. We use  $S_i(t)$  to denote the concentration or state of the  $i$ th molecule at the moment  $t$ , where  $S(t) = \{S_1(t), S_2(t), \dots, S_N(t)\}$  for all the molecules. Here, the function of a network is expressed as a special time series of  $S(t)$ , say  $S^0(t) = \{S_1^0(t), S_2^0(t), \dots, S_N^0(t)\}$ ; we call it the functional sequences. To determine the backbone network, two basic assumptions are necessary: the full network and functional sequences should be known from experiments or database in advance; and every node is indispensable in maintaining the function of the network, and cannot be removed.

The basic idea of the tinker algorithm is to identify the functional role by tinkering each edge. We check whether the functional sequences are still maintained under the remaining network after removing an edge. If this is true, this edge is a supplementary edge and can be removed; otherwise, the edge is a backbone edge, and should be retained. For discrete models, such as the Boolean network model and the stochastic model, the function being maintained means that the new time series (after an edge is removed) is completely in accordance with the functional sequences, namely,  $S(t) = S^0(t)$ . Obviously, each node's functional sequence is influenced only by its neighbors. Therefore, we can construct a local network for each node: the target node and its entry regulations. In addition, there must be a minimal subnetwork of the local network under which the functional sequence of the node is maintained, and we call it the subbackbone network. Then, the backbone network of the full network can be viewed as the combination of subbackbone networks of all nodes. The schematic showing how to determine the subbackbone network of each node is in Fig. 1. For the full network [Fig. 1(a)], we arbitrarily select a node and construct its local network [Fig. 1(b)]. Then, we check its subnetworks in the order of the subnetwork containing  $0, 1, 2, \dots, d_i$  edges [Figs. 1(c), (d), (e), (f)] to identify which subnetwork is the subbackbone network. With this idea, the code of the tinker algorithm can be written as follows:

```
// the C pseudo-code of the tinker algorithm
tinker(A, I){ // I denotes the set of nodes.
  Iremain=I;
  do(select node i ∈ Iremain){
    // construct the local network of node i, eij denotes the edge from j to i.
    select Ei0 = {eij|aij ≠ 0, j = 1, ..., N};
    //find the subnetwork of Ei0 that has the fewest entry regulations of
    //node i and under which the functional sequence can still be maintained.
    mark=false;
    for(k=0; k ≤ di; k++){
      for(l=1; l ≤ Cdik; l++){
        // update Ei with k edges of Ei0 retained, each k corresponds to
        // Cdik cases, and l denotes ergodicity for different cases.
        Ei = update(Ei0, k, l);
        // the subroutine check(S, E) is different for different models.
        if(check(S0, Ei)==true)mark=true, save Ei, break;
      }
      if (mark==true) break;
    }
    remove node i from Iremain;
  }while(Iremain != Φ);
  // obtain and check the result of the backbone subnetwork.
  Eresult = ∪i=1N Ei; check(S0, Eresult);
}
```



**Fig. 1** Steps to determine the subbackbone network of a node. (a) The full network. (b) The local network of a node, for example, node  $i_0 = 8$  with entry degree  $d_{i_0} = 3$ . All the subnetworks of the local network are shown in: (c1) one case for  $k = 0$ ; (d1–d3) three cases for  $k = 1$ ; (e1–e3) three cases for  $k = 2$ ; (f1) one case for  $k = 3$ . Each  $k$  indicates that the subnetwork of the local network contains  $k$  edges. Throughout this paper, the green arrows and pink dashed arrows respectively represent the positive regulations and “deactivation” (inhibition, repression, degradation) regulations, unless otherwise noted.

Next, we analyze the computational complexity of this algorithm. For a node  $i$  with entry degree  $d_i$ , the sub-networks of its local network can be organized as shown in Fig. 1. For each case of  $k$  and  $l$ ,  $k = 0, 1, 2, \dots$ , or  $d_i$  and  $l = 1, 2, \dots$ , or  $C_{d_i}^k$ , we update the network and use the subroutine *check* to check whether the functional sequences can be maintained under the new network. This process will cost most time in running the program since the number of cases that we have to check is

$$m_i = C_{d_i}^0 + C_{d_i}^1 + C_{d_i}^2 + \dots + C_{d_i}^{d_i-1} + C_{d_i}^{d_i} = 2^{d_i}. \quad (1)$$

Actually, the program may stop in the middle of the loop (if the check is true); thus, less than  $m_i$  cases are needed to be considered in practical conditions. Moreover, for a network with degree distribution  $d_1, d_2, \dots, d_N$ , we consider  $T_m = \sum_{i=1}^N m_i = \sum_{i=1}^N 2^{d_i}$  cases at most. Since  $T_m < N2^{d_{max}}$ , the computational time in which we determine the backbone network of a network with  $N$  nodes is less than the order of  $\vartheta(N2^{d_{max}})$ , where  $d_{max}$  is the largest degree of the nodes.

We also compare the efficiency of the tinker algorithm with that of the algorithm in the previous reference. To find all the possible networks of the functional sequences, most of the time for the algorithm used in reference [17] is spent in solving the Boolean network equations by enumerating the types of regulation between any two nodes. Because the types of regulation can be activation, inhibition, or no regulation, the time needed to solve the equations (not simplified) is of the order of  $\vartheta(N3^N)$ . If the equations are simplified, the time would be less. For the tinker algorithm, the running time of finding the backbone network is determined by  $d_{max}$ . For many biological networks, the average degree of each network is approximately between 2 and 4, where  $d_{max} \ll N$  usually [15, 35–38]. Obviously,  $\vartheta(N2^{d_{max}})$  is much smaller than  $\vartheta(N3^N)$ , indicating that the tinker algorithm is more efficient.

The tinker algorithm is more efficient mostly because it makes full use of the given network. Once the full network is given, in order to identify the functional role of an edge, only two cases (keeping or removing the edge) are needed to be checked. On the other hand, for the algorithm in reference [17], the full network is not required at first. To identify the type of regulation between any two nodes, three cases (activation, inhibition, or no regulation) should be checked. This leads to the computational complexity of  $\vartheta(N2^{d_{max}})$  versus  $\vartheta(N3^N)$  for the tinker algorithm and the previous algorithm. However, we can do nothing with the tinker algorithm if the full network is not given in advance. Therefore, it should be noted that the tinker algorithm is preferred only for the task of finding the backbone network of a given full network. We will focus our attention on this task in the following sections.

### 3 Simulation results

In this paper, we take the cell-cycle networks as our examples [15]. The full process of the cell cycle consists of four phases: G1, S, G2, and M. In the G1 phase, the cell grows, and undergoes division under appropriate conditions. Then, in the S phase, DNA is synthesized and chromosomes are replicated. The G2 phase is a “gap” phase between S and M. The final phase M corresponds to mitosis, in which chromosomes are separated and the cell is divided into two. Eventually, after the M stage, the cell enters G1, thereby completing a “cycle.” The cell-cycle networks of budding yeast and fission yeast are shown in Fig. 2(a) and Fig. 2(c), respectively [15, 20]. The network for budding yeast consists of eleven ( $N = 11$ ) proteins: Cln3, MBF, SBF, Cln1,-2, Cdh1, Swi5, Cdc20/Cdc14, Clb5,-6, Sic1, Clb1,-2, and Mcm1/SFF, whereas the network for fission yeast includes nine ( $N = 9$ ) proteins: SK, Cdc2/Cdc13, Ste9, Rum1, Slp1, Cdc2/Cdc13\*, Wee1/Mik1, Cdc25, and PP. We use  $S^0(t) = \{S_i^0(t), i = 1, 2, \dots, N\}$  to express the biological process (the functional sequences) of the cell cycle. The functional sequences  $S^0(t)$  for these two types of yeast are shown in Table 1.

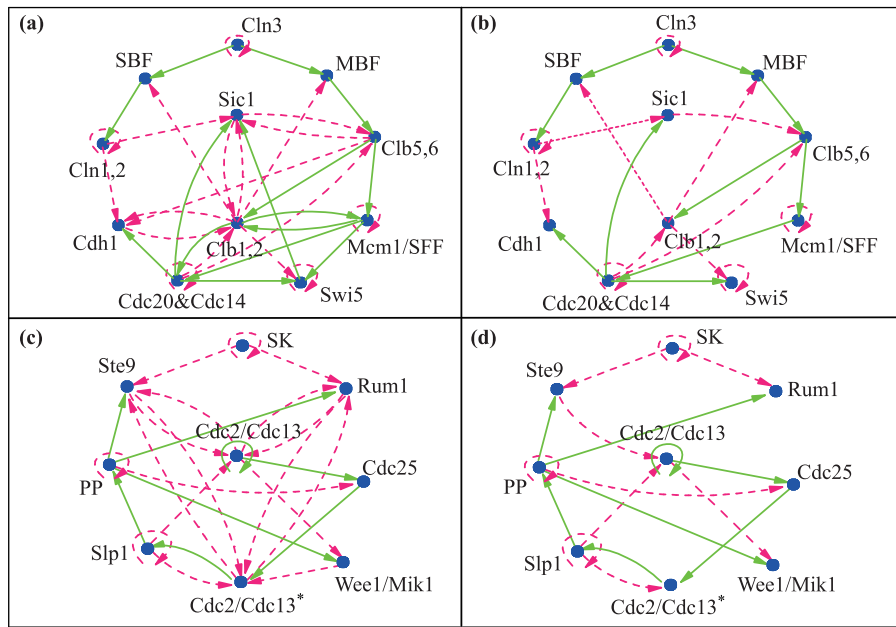
#### 3.1 Tinker algorithm applied in the Boolean network model

In the Boolean network model, each node  $i$  has only two states,  $S_i(t) = 1$  or  $0$ , representing the “on” (active) or “off” (inactive) state of the molecule, respectively. Therefore, there are  $2^N$  possible states of the networked system. The general dynamical rule, based on which the molecular states in the next time step can be determined by the molecular states in the current time step, is described as follows [15]:

$$S_i(t+1) = \begin{cases} 1, & \text{if } \sum_{j=1}^N a_{ij} S_j(t) + h_i > 0, \\ 0, & \text{if } \sum_{j=1}^N a_{ij} S_j(t) + h_i < 0, \\ S_i(t), & \text{if } \sum_{j=1}^N a_{ij} S_j(t) + h_i = 0, \end{cases} \quad (2)$$

where  $h_i$  is the threshold. Here,  $h_i = 0$ ,  $a_{ii}$  take the value  $-1, 1$ , or  $0$ , and  $a_{ij}$  take the value  $-\gamma, 1$ , or  $0$  for  $j \neq i$ . In the dominant inhibition model, where inhibition is dominant over stimulation for other types of interactions, namely,  $\gamma = \infty$ , Eq. (2) can be simplified into a logical equation [17]:

$$S_i(t+1) = \left( \sum_{j \neq i} (S_j(t) g_{ij}) + S_i(t) \bar{r}_{ii} + \bar{S}_i(t) g_{ii} \right) \prod_{j \neq i} (\bar{S}_j(t) \bar{r}_{ij}), \quad (3)$$



**Fig. 2** (a) and (c) The full cell-cycle networks of budding yeast and fission yeast, respectively. (b) and (d) The backbone networks of budding yeast and fission yeast in the Boolean network model, respectively. There are 34 edges for the full cell-cycle network of budding yeast, while only 23 edges are retained in its backbone network. There are 26 edges in the full cell-cycle network of fission yeast, while only 18 edges are retained in its backbone network. The backbone networks in the stochastic model are the same as those in the Boolean network model.

**Table 1** The time course of the cell-cycle process of budding yeast (a) and fission yeast (b). This time course is also called the functional sequences of the cell-cycle network.

(a)											
Time	Cln3	MBF	SBF	Cln1, 2	Cdh1	Swi5	Cdc20/14	Clb5, 6	Sic1	Clb1, 2	Mcm1/SFF
$t$	$S_1^0$	$S_2^0$	$S_3^0$	$S_4^0$	$S_5^0$	$S_6^0$	$S_7^0$	$S_8^0$	$S_9^0$	$S_{10}^0$	$S_{11}^0$
0	1	0	0	0	1	0	0	0	1	0	0
1	0	1	1	0	1	0	0	0	1	0	0
2	0	1	1	1	1	0	0	0	1	0	0
3	0	1	1	1	0	0	0	0	0	0	0
4	0	1	1	1	0	0	0	1	0	0	0
5	0	1	1	1	0	0	0	1	0	1	1
6	0	0	0	1	0	0	1	1	0	1	1
7	0	0	0	0	0	0	1	0	0	0	1
8	0	0	0	0	1	1	1	0	1	0	0
9	0	0	0	0	1	1	0	0	1	0	0
10	0	0	0	0	1	0	0	0	1	0	0
11	0	0	0	0	1	0	0	0	1	0	0

(b)										
Time	Sk	Cdc2/13	Ste9	Rum1	Slp1	Cdc2/13*	Wee1	Cdc25	PP	
$t$	$S_1^0$	$S_2^0$	$S_3^0$	$S_4^0$	$S_5^0$	$S_6^0$	$S_7^0$	$S_8^0$	$S_9^0$	
0	1	0	1	1	0	0	1	0	0	0
1	0	0	0	0	0	0	1	0	0	0
2	0	1	0	0	0	0	1	0	0	0
3	0	1	0	0	0	0	0	1	0	0
4	0	1	0	0	0	1	0	1	0	0
5	0	1	0	0	1	1	0	1	0	0
6	0	0	0	0	1	0	0	1	1	1
7	0	0	1	1	0	0	1	0	0	1
8	0	0	1	1	0	0	1	0	0	0
9	0	0	1	1	0	0	1	0	0	0

where  $r_{ij}$  represents a putative inhibitory edge and  $g_{ij}$  stands for a putative stimulatory edge from node  $j$  to  $i$ . The addition, multiplication, and bar in the equation

represent the Boolean operators OR, AND, and NOT, respectively. In this model, the subroutine *check* is described as follows:

```
// the subroutine check for the Boolean model
check( $S^0, E_i$ ){
   $l_s = \text{length of } S^0(t)$ ;
  flag=true;
  for( $t=0; t < l_s; t++$ ){
     $S(t) = S^0(t)$ ;
     $S_i(t+1) = eq(S(t), i)$ ; // here,  $eq()$  is equation (2) or (3)
    if(  $S_i^0(t+1) \neq S_i(t+1)$ ) flag=false, break;
  }
  return flag;
}
```

Based on the tinker algorithm and the subroutine of the dominant inhibition Boolean network model, the backbone networks of the cell-cycle networks of budding and fission yeast are determined and shown in Fig. 2(b) and Fig. 2(d), respectively. Through careful comparison, we find that they are identical to those shown in reference [17]. This verifies that the tinker algorithm is effective.

Furthermore, we apply the tinker algorithm to find the backbone network of a larger network. As shown in Fig. 3(a), the regulatory network that controls the differentiation process of the T helper cell contains 22 nodes: IL-18, IL-12, IFN $\gamma$ , IL-4, IL-10, NFAT, IL-18R, IL-12R, IFN- $\gamma$ R, IL-4R, IL-10R, IFN- $\beta$ , IRAK, STAT4, JAK1, STAT6, STAT3, IFN- $\beta$ R, SOCS1, T-bet, STAT1, and GATA3 [39]. In addition, the time course shown in Table 2 is chosen to be the functional sequences [ $S^0(t)$ ] of the nodes. As one can see, the largest degree of the nodes is 7 ( $d_{max} = 7$ ), which is much smaller than the network size. According to the theoretical result in Section 2, very short time would be enough to determine the backbone network with the tinker algorithm. In fact, our simulation result shows that we can obtain the backbone network [Fig. 3(b)] of the regulatory network in  $10^{-4}$  seconds. This demonstrates that the tinker algorithm is really efficient.

Although the backbone network of a full network can be found accurately with the tinker algorithm, it should be mentioned that the obtained backbone network may not be unique because of the existence of structure symmetry. From Fig. 2(c) and Table 1(b), one can see that the role of Ste9 and Rum1 in the cell-cycle network of fission yeast share high similarities. In Fig. 2(d), if the edge from Ste9 to Cdc2/Cdc13 is substituted by the edge from Rum1 to Cdc2/Cdc13, the new network is also a backbone network. As our algorithm is based on determining

the subbackbone network of each node just according to the main function sequence, structure symmetry is a global property and it may not be expressed. For the network with high structure symmetry, it is not easy to find all the backbone networks for it may greatly increase the computational complexity. Hence, we do not expand this topic further.

### 3.2 Tinker algorithm applied in the stochastic model

In the stochastic model, which is based on the Boolean network model, it is assumed that the transition of the states of the molecules is a stochastic process with the Markov property [8]. In the Boolean network model, the next state  $S(t+1)$  from the present state  $S(t)$  is exactly determined. However, in the stochastic model, the next state  $S(t+1)$  can be an arbitrary state of the  $2^N$  states, but with a different transition probability to each state. The transition probability is described as follows:

$$Pr(S(t+1)|S(t)) = \prod_{i=1}^N Pr(S_i(t+1)|S(t)), \quad (4)$$

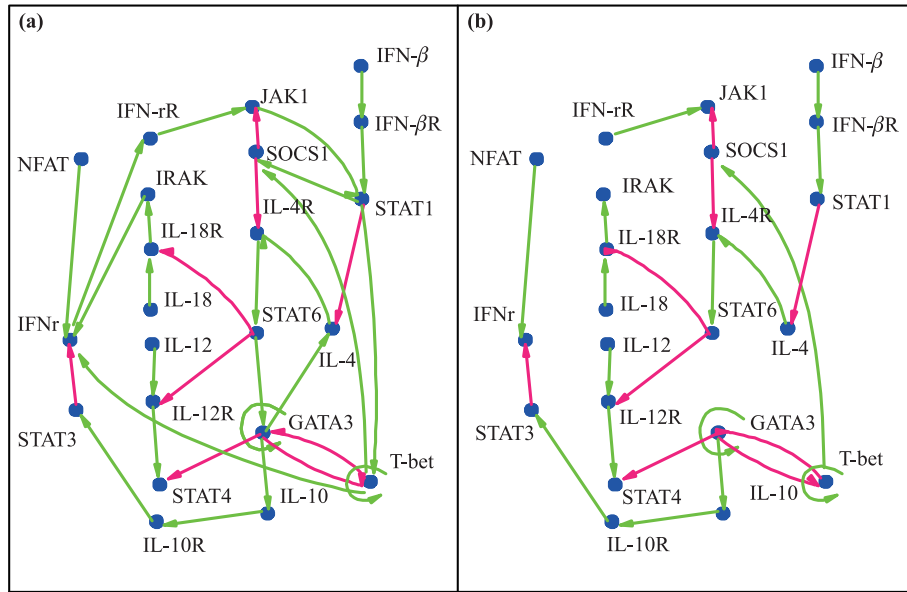
where  $Pr(S_i(t+1)|S(t))$  is the conditional probability that the state of the  $i$ th node is  $S_i(t+1)$  at moment  $t+1$ , and it is expressed as

$$Pr(S_i(t+1) = \sigma_i | S(t)) = \frac{e^{(2\sigma_i - 1)\beta T}}{e^{\beta T} + e^{-\beta T}}, \quad (5)$$

if  $T = \sum_{j=1}^N a_{ij} S_j(t) \neq 0$ ,  $\sigma_i \in \{0, 1\}$ ; if  $T = \sum_{j=1}^N a_{ij} S_j(t) = 0$ , then

$$Pr(S_i(t+1) = S_i(t) | S(t)) = \frac{1}{1 + e^{-\alpha}}, \quad \text{if } a_{ii} \neq 1, \quad (6)$$

$$Pr(S_i(t+1) = 1 | S(t)) = \frac{1}{1 + e^{-\alpha}}, \quad \text{if } a_{ii} = 1. \quad (7)$$



**Fig. 3** (a) The regulatory network that controls the differentiation process of T helper cell. It is almost identical to the network in reference [39] except that the node *TCR* is removed here. (b) The backbone network of the T helper cell network. There are 34 edges in the full T-helper cell network, while only 24 edges are retained in its backbone network.

**Table 2** The time course of the T-helper cell network.

Time	$S_1^0$	$S_2^0$	$S_3^0$	$S_4^0$	$S_5^0$	$S_6^0$	$S_7^0$	$S_8^0$	$S_9^0$	$S_{10}^0$	$S_{11}^0$	$S_{12}^0$	$S_{13}^0$	$S_{14}^0$	$S_{15}^0$	$S_{16}^0$	$S_{17}^0$	$S_{18}^0$	$S_{19}^0$	$S_{20}^0$	$S_{21}^0$	$S_{22}^0$	
0	1	1	0	1	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	0	1	1	1	1	1	0	1	0	0	1	0	0	1	0	1	0	1	0
2	1	1	1	1	1	1	1	1	1	1	0	1	1	0	1	1	0	1	1	0	1	0	0
3	1	1	1	0	1	1	0	0	1	0	1	1	1	1	0	1	0	1	1	1	1	1	1
4	1	1	1	0	1	1	0	0	1	0	1	1	1	0	0	1	1	1	1	0	1	0	0
5	1	1	0	0	1	1	0	0	1	0	1	1	1	0	0	1	1	1	1	1	1	1	1
6	1	1	0	0	1	1	0	0	1	0	1	1	1	0	0	1	1	1	1	0	1	0	0
7	1	1	0	0	1	1	0	0	1	0	1	1	1	0	0	1	1	1	1	1	1	1	1

The values of  $a_{ii}$  and  $a_{ij}$  are identical to those used in the Boolean network model.  $\alpha$  and  $\beta$  are two parameters, whose robustness has been analyzed in reference [8]. Here, we set  $\alpha = 5$ ,  $\beta = 6$ , and  $\gamma = 10$ .

Next, we will show the details of the subroutine *check* in this model. Although the next state  $S(t+1)$  can be an arbitrary state of the  $2^N$  states, there is a state  $S^L(t+1)$  such that

$$Pr(S^L(t+1)|S(t)) = \max\{Pr(S_1(t+1), \dots, S_N(t+1)|S(t)), S_i(t+1) \in \{0, 1\}\}, \tag{8}$$

and we call it the most probable transition state. Naturally, the state  $S_i^L(t+1)$  for each  $i$  satisfies  $Pr(S_i^L(t+1)|S(t)) = \max\{Pr(S_i(t+1)|S(t), S_i(t+1)=0, 1)\}$ , namely,

$$S_i^L(t+1) = \{S_i(t+1)|Pr(S_i(t+1)|S(t))$$

$$= \max\{Pr(S_i(t+1)|S(t))\}. \tag{9}$$

In this model, the function of the network being maintained means that  $S^L = S^0$ . Then, the subroutine *check* is described as follows:

```

// the subroutine check for the stochastic model
check( $S^0, E_i$ ){
   $l_s = \text{length of } S^0(t)$ ;
  flag=true;
  for( $t=0; t < l_s; t++$ ){
     $S(t) = S^0(t)$ ;
     $S_i^L(t+1) = \text{pr-eq}(S(t), i)$ ; // here, pr-eq( ) is equation (9)
    if(  $S_i^L(t+1) \neq S_i^0(t+1)$  ) flag=false, break;
  }
  return flag;
}

```

We still take the cell-cycle network of budding and fission yeast as examples. Their backbone networks can be determined with this algorithm, and it turns out that they are the same as those found in the dominant Boolean network model [Figs. 2(b) and (d)].

We also compare the importance of backbone network edges with supplementary edges (the edges belonging to the full network but not in the backbone network). For the budding yeast, there are 34 edges in the full network and 23 edges in the backbone network. We define the trajectory transition probability as

$$Prt = \prod_{t=0}^{t_{end}} Pr\{S^0(t+1)|S^0(t)\}. \quad (10)$$

$Prt$  is calculated under the following networks: the backbone network with one edge removed (23 cases), the backbone network, the backbone network with one of the supplementary edges added (11 cases), and the full network [Fig. 4(a)]. The probability  $Prt$  is nearly zero for the backbone network with one edge removed, indicating that the network can barely carry out the function if even one edge is lost. For the other cases,  $Prt$  is much larger, and it is the largest for the full network case. It demonstrates that the supplementary edges really can enhance the robustness of the network. We carry out the same process in the cell-cycle network of fission yeast and find similar phenomena [Fig. 4(b)].

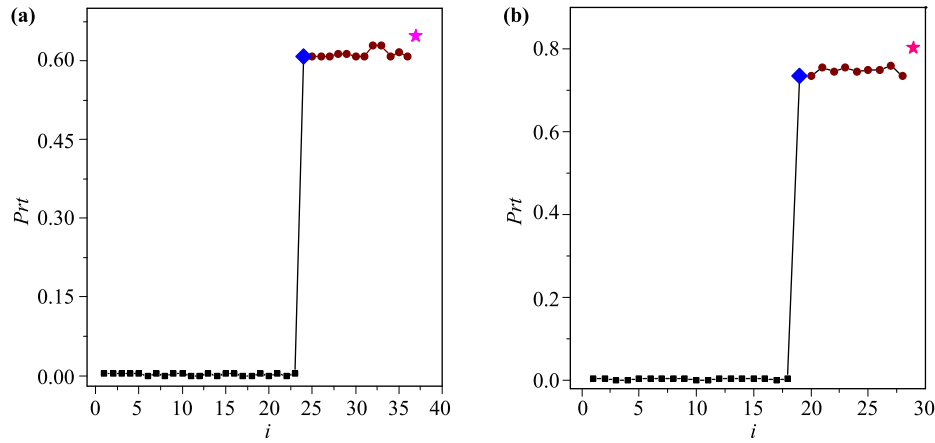
### 3.3 Tinker algorithm applied in the ordinary differential equation (ODE) model

The ordinary differential equation (ODE) model is a time continuous model. It is one of the most useful and effective models, as it can exactly give the concentration of each molecule, which can be compared with experimental results. Therefore, we expect to determine the functional backbone network in the ODE model with the tinker algorithm.

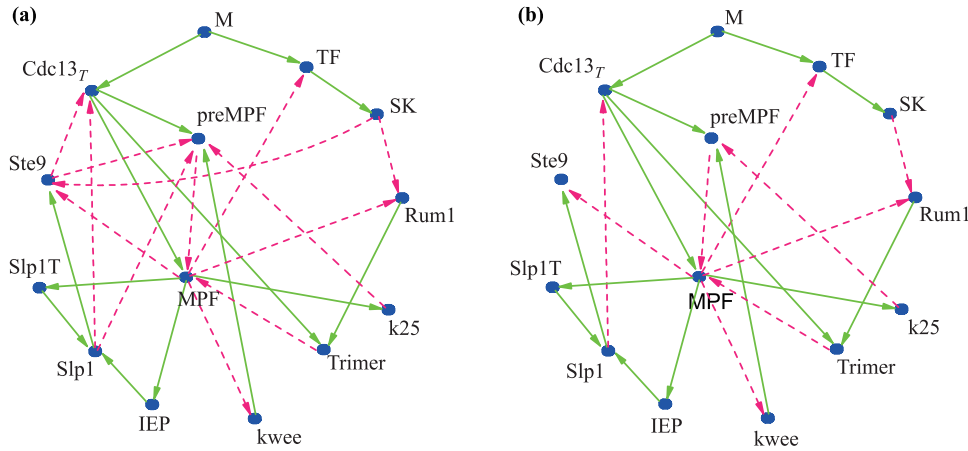
Compared to the previous two models, determining the backbone network using the tinker algorithm is

slightly different in the ODE model. In the Boolean network model, to determine whether the function is maintained, we check whether the new  $S(t)$  is equal to  $S^0(t)$  after removing some edges of the network. If we still apply this standard to the ODE model, the results of the check will always be false. Because for the ODE model, the functional sequence  $[S^0(t)]$  is the continuous solution of the ODE equations under the full network. If an arbitrary edge of the full network is removed, the new solution  $S(t)$  under the new network would always be different from  $S^0(t)$ . However, a very small difference between  $S(t)$  and  $S^0(t)$  is actually inconsequential for the function. Thus, the original standard is not appropriate for the ODE model. Therefore, we give a more reasonable standard: (i) the Pearson correlation coefficient ( $R_i$ ) between  $S_i^0(t)$  and the new  $S_i(t - t_0)$  is larger than  $R_0$ , namely,  $R_i > R_0$ , for  $i = 1, 2, \dots, N$ ; (ii) the difference between the period of  $S_i^0(t)$  and  $S_i(t)$  is less than  $T_0$ , namely,  $|T_i^0 - T_i| < T_0$ , for  $i = 1, 2, \dots, N$ , where  $S_i(t - t_0)$  is a proper time translation of  $S_i(t)$ . If the new standard is achieved with the proper thresholds  $R_0$  and  $T_0$ , the basic trends of  $S(t)$  and  $S^0(t)$  would be identical except for small deviations. Then we can say that the function of the network is maintained. Under the new standard, the network should be treated as a whole and each node cannot be treated locally anymore, necessitating some modifications of the tinker algorithm. The details of the tinker algorithm, as well as the subroutine *check*, for the ODE model can be seen in Appendix A.

Here, we take the cell-cycle network of fission yeast as an example to show the results. Many ODE models for the cell-cycle network of fission yeast have been introduced [11, 40–42]; here we use the model in Refs. [10] and [43] as it can well explain the variation in concentration of the proteins during the process of the cell cycle. The network for this ODE model is shown in Fig. 5(a). We use  $S_1(t), S_2(t), \dots, S_{14}(t)$  to represent the concentrations or values of Cdc13<sub>T1</sub>, preMPF<sub>1</sub>, Ste9<sub>1</sub>, Slp1<sub>T1</sub>, Slp1<sub>1</sub>, IEP<sub>1</sub>, Rum1<sub>T1</sub>, SK<sub>1</sub>, M, TF<sub>1</sub>,  $k_{wee1}$ ,  $k_{251}$ , MPF<sub>1</sub>, and Trimer<sub>1</sub>, respectively; the subscript 1 marks the rescaled variables with maximum 1. The equations for



**Fig. 4** (a) The trajectory probability  $P_{rt}$  in Eq. (10) for the backbone network with one edge removed (first 23 squares), for the backbone network (blue diamond), for the backbone network with one of the supplementary edges added (11 dots), and for the full cell-cycle network (pink star) of budding yeast. (b) The same as (a) for fission yeast.



**Fig. 5** (a, b) The full cell-cycle network and the backbone network of fission yeast in the ODE model, respectively.

$S(t)$  are described as follows:

$$\frac{dS_1}{dt} = a_{1,9}k_1S_9 - (k'_2 + a_{1,3}k''_2S_3 + a_{1,5}k'''_2S_5)S_1,$$

$$\frac{dS_2}{dt} = a_{2,11}k'_0S_{11}(a_{2,1}k''_0S_1 - S_2) - a_{2,12}k'''_0S_{12}S_2 - (k'_2 + a_{2,3}k''_2S_3 + a_{2,5}k'''_2S_5)S_2,$$

$$\frac{dS_3}{dt} = a_{3,5}(k'_3 + k''_3S_5) \frac{1 - S_3}{J_3 + 1 - S_3} - (a_{3,8}k'_4S_8 + a_{3,13}k_4S_{13}) \frac{S_3}{J_4 + S_3},$$

$$\frac{dS_4}{dt} = k'_5 + a_{4,13}k''_5 \frac{S_{13}^4}{J_5^4 + S_{13}^4} - k_6S_4,$$

$$\frac{dS_5}{dt} = a_{5,6}k_7S_6 \frac{a_{5,4}(S_4 - S_5)}{J_7 + S_4 - S_5} - k_8 \frac{S_5}{J_8 + S_5} - k_6S_5,$$

$$\frac{dS_6}{dt} = a_{6,13}k_9S_{13} \frac{1 - k'_9S_6}{J_9 + 1 - k'_9S_6} - k_{10} \frac{k'_9S_6}{J_{10} + k'_9S_6},$$

$$\frac{dS_7}{dt} = k_{11} - (k_{12} + a_{7,8}k'_{12}S_8 + a_{7,13}k''_{12}S_{13})S_7,$$

$$\frac{dS_8}{dt} = a_{8,10}k_{13}S_{10} - k_{14}S_8,$$

$$\frac{dS_9}{dt} = \mu S_9,$$

$$S_{10} = G(a_{10,9}k_{15}S_9, k'_{16} + a_{10,13}k''_{16}S_{13}, J_{15}, J_{16}),$$

$$S_{11} = k'_{wee} + (k''_{wee} - k'_{wee})$$

$$\cdot G(V_{awee}, a_{11,13}V_{iwee}S_{13}, J_{awee}, J_{iwee}),$$

$$S_{12} = k'_{25} + (k''_{25} - k'_{25})G(a_{12,13}V_{a25}S_{13}, V_{i25}, J_{a25}, J_{i25}),$$

$$S_{13} = \frac{(a_{13,1}k_{17}S_1 - a_{13,2}k'_{17}S_2)(a_{13,1}k_{17}S_1 - a_{13,14}k''_{17}S_{14})}{k'''_{17}S_1},$$

$$S_{14} = \frac{a_{14,1}a_{14,7}k_{18}S_1S_7}{\sigma + \sqrt{\sigma^2 - k'_{18}S_1S_7}},$$

where

$$\sigma = k'_{19}S_1 + k''_{19}S_7 + K_{diss},$$

$$G(a, b, c, d) = \frac{2ad}{b-a+bc+ad + \sqrt{(b-a+bc+ad)^2 - 4ad(b-a)}}.$$

$a_{i,j}$  is the matrix element of the regulatory network [Fig. 5(a)],  $A = \{a_{i,j}, i, j = 1, \dots, N\}$ , takes the value 0 or 1 and denotes the presence or absence of interaction from protein  $j$  to  $i$ . The values of the parameters in the equations can be seen in Table 3. Moreover, the cell mass  $M$  represented by  $S_9$ , increases exponentially before cell division. We divide the cell mass by two ( $S_9 = S_9/2$ ) at the end of mitosis; the time  $S_{13}$  (MPF) decreases through 0.1 when a cell splits itself into two daughter cells.

Based on the modified tinker algorithm and the new standard, the backbone network can be found. During our simulation, the functional sequence of the network,  $S^0(t) = \{S_1^0(t), \dots, S_{14}^0(t)\}$ , is defined as the solution of the ODE equations under the full cell-cycle network. Since the values of  $R_0$  and  $T_0$  should not be too critical or too weak, we set  $R_0 = 0.8$  and  $T_0 = 2 \ll T_i^0$ , where  $T_i^0 \approx 138$ . Accordingly, we determine the backbone network and show it in Fig. 5(b). The solutions of the ODE

equations under the full network and the backbone network are compared in Fig. 6. One can see that they are well matched, indicating that the backbone network can really maintain the function of the full cell-cycle network, further demonstrating that the tinker algorithm is adoptable to determine the backbone network in the ODE model.

## 4 Conclusion and discussion

We have proposed a new algorithm, the tinker algorithm, with which the backbone network can be determined accurately and efficiently in different models. In the dominant inhibition Boolean network model, the functional backbone networks determined in the cell-cycle networks of the two yeasts match well the results in previous works, verifying that the tinker algorithm is effective. In the stochastic model, the functional backbone networks could also be determined easily. At the same time, the importance of the backbone network edges and the supplementary edges has been tested. In addition, we found that the network loses its function if an arbitrary backbone network edge is removed, while the supplementary edges enhance the robustness of the network. Furthermore, this method has also been applied to the ODE model, and the backbone network of the cell-cycle network of fission yeast was determined.

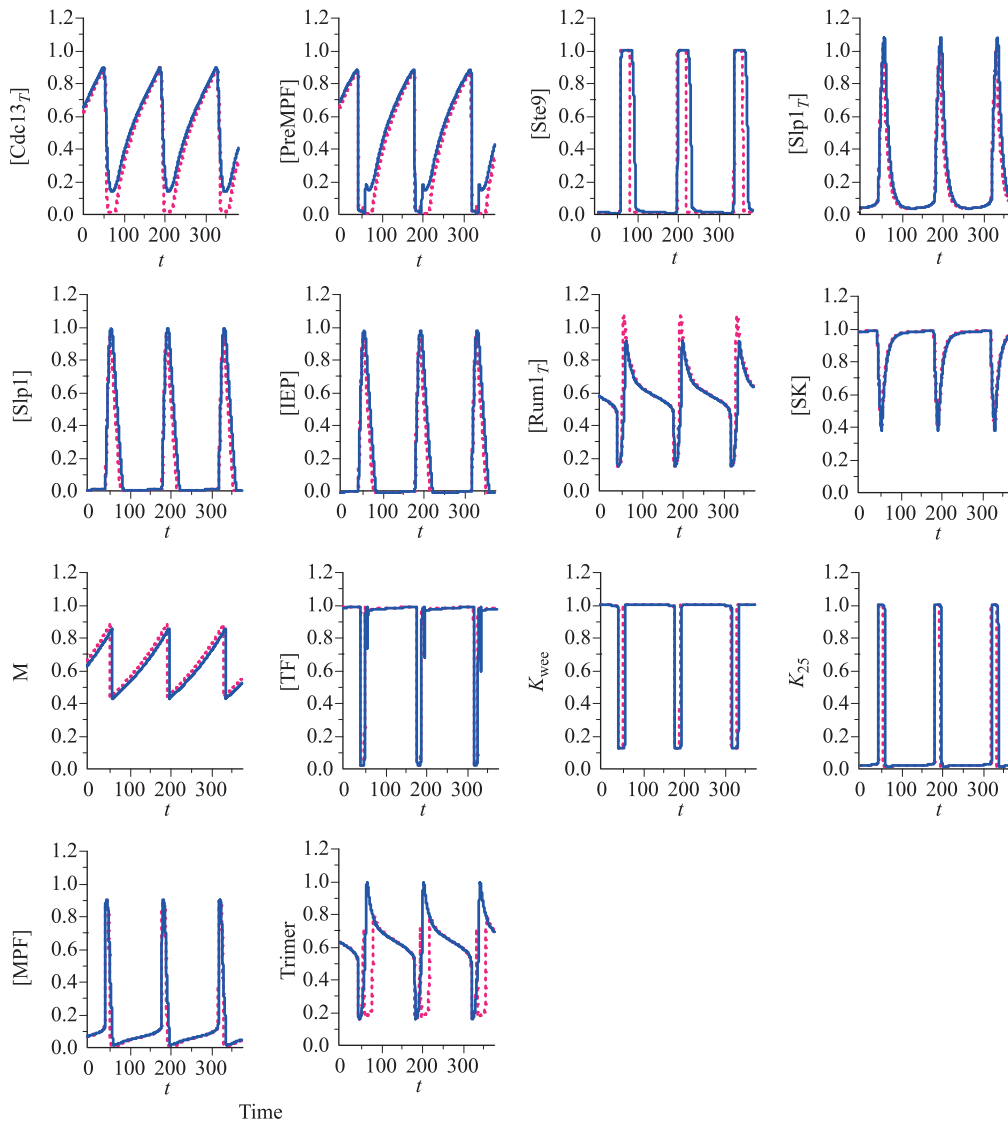
**Table 3** Parameter values for the ODE model of the cell-cycle network of fission yeast. They are almost identical to those in Ref. [43] except for some modifications.

$S_1$	$k_1 = 0.04,$	$k'_2 = 0.03,$	$k''_2 = 1.0,$	$k'''_2 = 0.21$		
$S_2$	$k'_0 = 1.0,$	$k''_0 = 1.17,$	$k'''_0 = 5.0$			
$S_3$	$k'_3 = 1.0,$	$k''_3 = 21.0,$	$J_3 = 0.01,$	$k_4 = 50.75,$	$k'_4 = 1.98,$	$J_4 = 0.01$
$S_4$	$k'_5 = 0.002,$	$k''_5 = 0.143,$	$J_5 = 0.20689,$	$k_6 = 0.1$		
$S_5$	$k_7 = 0.429,$	$J_7 = 0.0005,$	$k_8 = 0.119,$	$J_8 = 0.0005$		
$S_6$	$k_9 = 0.16,$	$k'_9 = 0.91,$	$J_9 = 0.01,$	$k_{10} = 0.01,$	$J_{10} = 0.01$	
$S_7$	$k_{11} = 0.698,$	$k_{12} = 0.01,$	$k'_{12} = 0.99,$	$k''_{12} = 4.35$		
$S_8$	$k_{13} = 0.1,$	$k_{14} = 0.1$				
$S_9$	$\mu = 0.005$					
$S_{10}$	$k_{15} = 3.0,$	$k'_{16} = 1.0,$	$k''_{16} = 2.9,$	$J_{15} = 0.01,$	$J_{16} = 0.01$	
$S_{11}$	$k'_{wee} = 0.115,$	$k''_{wee} = 1.0,$	$V_{awe} = 0.25,$	$V_{iwee} = 1.45,$	$J_{awe} = 0.01,$	$J_{iwee} = 0.01$
$S_{12}$	$k'_{25} = 0.01,$	$k''_{25} = 1.0,$	$V_{a25} = 1.45,$	$V_{i25} = 0.25,$	$J_{a25} = 0.01,$	$J_{i25} = 0.01$
$S_{13}$	$k_{17} = 1.5,$	$k'_{17} = 1.3,$	$k''_{17} = 1.0,$	$k'''_{17} = 2.175$		
$S_{14}$	$k_{18} = 0.441,$	$k'_{18} = 0.882$				
$\sigma$	$k'_{19} = 1.5,$	$k''_{19} = 0.147,$	$K_{diss} = 0.001$			

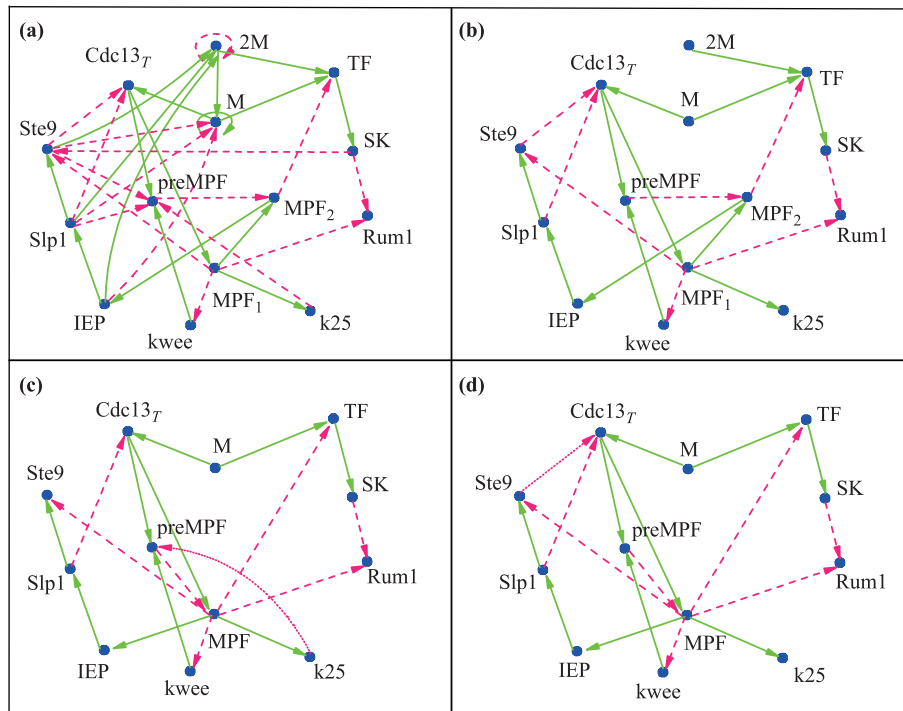
The tinker algorithm is quite different from the method used in previous research papers. In reference [17], the method is mainly based on the idea of reverse engineering and makes full use of the message of the biological process; the goal is to determine all possible networks just from the biological process according to the basic rules of the dominant inhibition Boolean network model. Nevertheless, the requirement of basic rules restricts it to only determine the backbone networks in the dominant inhibition Boolean network model. In contrast, the tinker algorithm is based on the idea of tinkering and makes use of the messages of both the network structure and the biological process. The purpose is to remove the supplementary edges that have little impact

on the biological process. This method is more general and can be applied in various mathematical models as shown in our simulation results.

With this advantage, the backbone networks obtained from different models can be compared with each other. From previous results, we know that the backbone networks in the dominant inhibition Boolean network model and the stochastic model are identical. Next, we discuss the difference between the backbone networks in the Boolean network model and the ODE model. In the previous section, the backbone network of the fission yeast cell-cycle network has been found in the ODE model. As a special coarse-grained limit of this ODE model, a Boolean model can be formulated [43], whose details can



**Fig. 6** Time series of the solutions of the ODE model under the full network in Fig. 5(a) (pink dashed line) and the backbone network in Fig. 5(b) (blue solid line).



**Fig. 7** (a) The full network of fission yeast obtained according to the transition from the ODE model to the Boolean network model. (b) The backbone network of (a) in the Boolean network model. (c) The simplified network of the backbone network in the ODE model [Fig. 5(b)], where the nodes Slp1 $T$  and Trimer are removed. (d) The simplified network of (b), where the node 2M is removed, and MPF1 and MPF2 are merged. The difference between networks in (c) and (d) is marked with dotted arrows.

be seen in Appendix B. We determine the backbone network in this Boolean model and compare it with the backbone network found in the ODE model (Fig. 7). Interestingly, we find that they are almost identical except that one edge is mismatched: the regulation from k25 to preMPF in the ODE model is replaced by the regulation from Ste9 to Cdc13 $T$  in the Boolean network model [Figs. 7(c) and (d)]. By analyzing the interaction coefficients and time series associated with these four proteins, we find that the mismatch may be attributed to the following fact: the regulations between any two proteins can be strong or weak in the ODE model, while they are all the same in the Boolean model. This indicates that the interaction strength affects the importance of the interaction (edge), and hence influences the backbone network structure.

Our aim of proposing this algorithm is to find the most important interactions in the complex biological processes. It is easy to find that the perturbation on the backbone structure rather than the supplementary edges can make a much more important effect on the dynamic process. As the evolutionary process goes on, the mutations on the supplementary structure may cause an effective process and help the creatures to adapt to

the change of environment, whereas the mutation on the backbone structure usually causes the process to be terminated and adversely affects the survival and evolution. The tinker algorithm helps us distinguish the backbone network and supplementary edges, and it may give us a preliminary conclusion about the most important interactions of the network. In addition to determining the backbone network, we notice that many researchers are working to detect the core or the skeleton network of a complex network, which is defined as a network that has only a few nodes and its dynamic trajectory qualitatively approaches the full network [44–46]. This skeleton or core network has much in common with the backbone network for they both maintain the main function of the full network. Since the tinker algorithm is a general method, we look forward to it being put into practical use in future studies.

## Appendix A The details of the tinker algorithm for the ODE model

The modified tinker algorithm and subroutine *check* for the ODE model are organized as follows:

```
// the C pseudo-code of the tinker algorithm for the ODE model

tinker(A, I){
  I_remain=I, E_remain = {eij|aij ≠ 0, i, j = 1, ..., N};
  do(select node i ∈ I_remain) {
    select Ei0 = {eij|aij ≠ 0, j = 1, ..., N};
    mark=false;
    for(k=0; k ≤ di; k++) {
      for(l=1; l ≤ Cdik; l++){
        Ei=E_remain - Ei0+update(Ei0, k, l); //different from previous case
        if(check(S0, Ei)==true)mark=true, E_remain = Ei, break;
      }
    }
    if(mark==true)break;
  }
  remove node i from I_remain;
}while(I_remain != Φ);
E_result = E_remain; check(S0, E_result);
}
```

```
// The subroutine check for the ODE model
check(S0, Ei){
  flag=true;
  // Use the ODE equations to obtain the time series with
  // initial condition S0(0) and network Ei.
  S(t)=ode-eq(Ei, S0(0));
  //shift S(t) with proper time t0;
  S'(t) = S(t - t0);
  // Calculate the Pearson correlation R between S' and S0 and the period of S' and S0.
  [Ri, Ti, Ti0, i = 1, ... N]=pc-period(S', S0);
  for(i=1; i ≤ N; i++){
    if(Ri < R0 OR |Ti - Ti0| > T0) flag=false, break;
  }
  return flag;
}
```

## Appendix B Boolean network model obtained from the transition of the ODE model

The Boolean network model of the cell-cycle network of fission yeast is obtained from the transition of the ODE model [43]. The regulatory network ( $A'$ ) is shown in Fig. 7(a). Comparing this network with the original regulatory network ( $A$ ) used in the ODE model, we can get the following messages: the protein MPF is divided into two parts MPF1 and MPF2; M is divided into M and 2M, where M works at the beginning of the cell cycle and 2M plays the role of an indicator for the end of the cell cycle; Slp1<sub>T</sub> and Trimer are removed.

We use  $S_1(t), S_2(t), \dots, S_{14}(t)$  to represent the protein states of Cdc13<sub>T</sub>, preMPF, MPF1, MPF2, k<sub>25</sub>, k<sub>wee</sub>, M, Slp1, Ste9, TF, SK, 2M, IEP, and Rum1, respectively. Here, the Boolean network model is almost the same with Eq. (2) except for a small difference and is described as follows:

$$S_i(t+1) = \begin{cases} 1, & \text{if } \sum_{j=1}^N a'_{ij} S_j(t) + h_i > 0, \\ 0, & \text{if } \sum_{j=1}^N a'_{ij} S_j(t) + h_i \leq 0, \end{cases}$$

where

$$H = \{h_1, h_2, \dots, h_{14}\} \\ = \{0, -1, 0, 0, -0.5, 0.5, 3, 0, 0, 0, 0, -3, 0, 0.5\}.$$

**Table S4** The functional sequences of the cell-cycle process of fission yeast in the Boolean network model obtained from the transition of the ODE model [43].

Time	Cdc13 <sub>T</sub>	preMBF	MBF1	MBF2	k <sub>25</sub>	k <sub>wee</sub>	M	Slp1	Ste9	TF	SK	2M	IEP	Rum1
$t$	$S_1^0$	$S_2^0$	$S_3^0$	$S_4^0$	$S_5^0$	$S_6^0$	$S_7^0$	$S_8^0$	$S_9^0$	$S_{10}^0$	$S_{11}^0$	$S_{12}^0$	$S_{13}^0$	$S_{14}^0$
0	0	0	0	0	0	1	0	1	1	0	0	1	1	1
1	0	0	0	0	0	1	1	1	1	1	0	0	0	1
2	0	0	0	0	0	1	1	0	1	1	1	0	0	1
3	0	0	0	0	0	1	1	0	0	1	1	0	0	0
4	1	0	0	0	0	1	1	0	0	1	1	0	0	0
5	1	1	1	0	0	1	1	0	0	1	1	0	0	0
6	1	1	1	0	1	0	1	0	0	1	1	0	0	0
7	1	0	1	0	1	0	1	0	0	1	1	0	0	0
8	1	0	1	1	1	0	1	0	0	1	1	0	0	0
9	1	0	1	1	1	0	1	0	0	0	1	0	1	0
10	1	0	1	1	1	0	1	1	0	0	0	0	1	0
11	0	0	1	1	1	0	1	1	0	0	0	0	1	0
12	0	0	0	1	1	0	1	1	0	0	0	0	1	0
13	0	0	0	0	0	1	1	1	1	0	0	0	1	1

$a'_{ij} = 1$  for a green arrow from protein  $j$  to  $i$  and  $a'_{ij} = -1$  for a pink dashed arrow from protein  $j$  to  $i$ . The functional sequence  $S^0(t)$  is shown in Table S4.

**Acknowledgements** This study was supported partially by the National Science Foundation of China (Grant Nos. 11475253, 11405263, and 11675112) and the Natural Science Foundation of Zhejiang Province (Grant No. LY16A050001).

## References

1. S. A. Kauffman, *The Origins of Order: Self-organization and Selection in Evolution*, New York: Oxford University Press, 1993
2. U. Alon, *An Introduction to System Biology: Design Principles of Biological Circuits*, Boca Raton: Chapman & Hall, 2006
3. I. Rigoutsos and G. Stephanopoulos, *Systems Biology: Networks, Models, and Applications*, New York: Oxford University Press, 2007
4. X. F. Pang, Influence of structure disorders and temperatures of systems on the bio-energy transport in protein molecules (II), *Front. Phys.* 3(4), 457 (2008)
5. D. T. Gillespie, A general method for numerically simulating the stochastic time evolution of coupled chemical reactions, *J. Comput. Phys.* 22(4), 403 (1976)
6. H. de Jong, Modeling and simulation of genetic regulatory systems: A literature review, *J. Comput. Biol.* 9(1), 67 (2002)
7. X. Cai and Z. M. Yuan, Stochastic modeling and simulation of the p53-MDM2/MDMX loop, *J. Comput. Biol.* 16(7), 917 (2009)
8. Y. P. Zhang, M. P. Qian, Q. Ouyang, M. H. Deng, F. T. Li, and C. Tang, Stochastic model of yeast cell-cycle network, *Physica D* 219(1), 35 (2006)
9. K. C. Chen, A. Csikász-Nagy, B. Gyorfy, J. Val, B. Novak, and J. J. Tyson, Kinetic analysis of a molecular model of the budding yeast cell cycle, *Mol. Biol. Cell* 11(1), 369 (2000)
10. B. Novak, Z. Pataki, A. Ciliberto, and J. J. Tyson, Mathematical model of the cell division cycle of fission yeast., *Chaos* 11(1), 277 (2001)
11. J. J. Tyson, K. Chen, and B. Novak, Network dynamics and cell physiology, *Nat. Rev. Mol. Cell Biol.* 2(12), 908 (2001)
12. J. J. Tyson, K. C. Chen, and B. Novak, Sniffers, buzzers, toggles and blinkers: dynamics of regulatory and signaling pathways in the cell, *Curr. Opin. Cell Biol.* 15(2), 221 (2003)
13. A. Csikász-Nagy, Computational systems biology of the cell cycle, *Brief. Bioinformatics* 10(4), 424 (2009)
14. S. A. Kauffman, Metabolic stability and epigenesis in randomly constructed genetic nets, *J. Theor. Biol.* 22(3), 437 (1969)
15. F. Li, T. Long, Y. Lu, Q. Ouyang, and C. Tang, The yeast cell-cycle network is robustly designed, *Proc. Natl. Acad. Sci. USA* 101(14), 4781 (2004)
16. S. Bornholdt, Systems biology. Less is more in modeling large genetic networks, *Science* 310(5747), 449 (2005)
17. G. Wang, C. Du, H. Chen, R. Simha, Y. Rong, Y. Xiao, and C. Zeng, Process-based network decomposition reveals backbone motif structure, *Proc. Natl. Acad. Sci. USA* 107(23), 10478 (2010)

18. Q. Z. Xia, L. L. Liu, W. M. Ye, and G. Hu, Inference of gene regulatory networks with the strong-inhibition Boolean model, *New J. Phys.* 13(8), 083002 (2011)
19. G. Wang, Y. Rong, H. Chen, C. Pearson, C. Du, R. Simha, and C. Zeng, Process-driven inference of biological network structure: Feasibility, minimality, and multiplicity, *PLoS ONE* 7(7), e40330 (2012)
20. M. I. Davidich and S. Bornholdt, Boolean network model predicts cell cycle sequence of fission yeast, *PLoS ONE* 3(2), e1672 (2008)
21. M. I. Davidich and S. Bornholdt, Boolean network model predicts knockout mutant phenotypes of fission yeast, *PLoS ONE* 8(9), e71786 (2013)
22. H. Kitano, Biological robustness, *Nat. Rev. Genet.* 5(11), 826 (2004)
23. J. Stelling, U. Sauer, Z. Szallasi, F. III. Doyle, and J. Doyle, Robustness of cellular functions, *Cell* 118(6), 675 (2004)
24. A. Wagner, Robustness and Evolvability in Living Systems, Princeton Studies in Complexity, Princeton: Princeton University Press, 2005
25. G. M. Cooper, The Cell: A Molecular Approach, Chapter 14, Washington: ASM Press, 2000
26. M. Morohashi, A. E. Winn, M. T. Borisuk, H. Bolouri, J. Doyle, and H. Kitano, Robustness as a measure of plausibility in models of biochemical networks, *J. Theor. Biol.* 216(1), 19 (2002)
27. U. de Lichtenberg, L. J. Jensen, A. Fausbøll, T. S. Jensen, P. Bork, and S. Brunak, Comparison of computational methods for the identification of cell cycle-regulated genes, *Bioinformatics* 21(7), 1164 (2005)
28. F. T. Li, M. Y. Hu, B. Zhao, H. Yan, B. Wu, and Q. Ouyang, A globally attractive cycle driven by sequential bifurcations containing ghost effects in a 3-node yeast cell cycle model, arXiv: 1312.5204 (2014)
29. X. M. Zhang, B. Shao, Y. L. Wu, and Q. Ouyang, A reverse engineering approach to optimize experiments for the construction of biological regulatory networks, *PLoS ONE* 8(9), e75931 (2013)
30. L. Yang, Y. Meng, C. Bao, W. Liu, C. Ma, A. Li, Z. Xuan, G. Shan, and Y. Jia, Robustness and backbone motif of a cancer network regulated by miR-17-92 cluster during the G1/S transition, *PLoS ONE* 8(3), e57009 (2013)
31. F. Jacob, Evolution and tinkering, *Science* 196(4295), 1161 (1977)
32. U. Alon, Biological networks: The tinkerer as an engineer, *Science* 301(5641), 1866 (2003)
33. U. Alon, Network motifs: Theory and experimental approaches, *Nat. Rev. Genet.* 8(6), 450 (2007)
34. R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon, Network motifs: Simple building blocks of complex networks., *Science* 298(5594), 824 (2002)
35. C. Oosawa and M. A. Savageau, Effects of alternative connectivity on behavior of randomly constructed Boolean networks, *Physica D* 170(2), 143 (2002)
36. S. Bornholdt and T. Rohlf, Topological evolution of dynamical networks: Global criticality from local dynamics, *Phys. Rev. Lett.* 84, 6114 (2000)
37. M. C. Costanzo, M. E. Crawford, J. E. Hirschman, J. E. Kranz, P. Olsen, L. S. Robertson, M. S. Skrzypek, B. R. Braun, K. L. Hopkins, P. Kondu, C. Lengieza, J. E. Lew-Smith, M. Tillberg, and J. I. Garrels, YPD<sup>TM</sup>, PombePD<sup>TM</sup> and WormPD<sup>TM</sup>: model organism volumes of the BioKnowledge<sup>TM</sup> Library, an integrated resource for protein information, *Nucleic Acids Res.* 29(1), 75 (2001)
38. T. I. Lee, N. J. Rinaldi, F. Robert, D. T. Odom, Z. Bar-Joseph, G. K. Gerber, N. M. Hannett, C. T. Harbison, C. M. Thompson, I. Simon, J. Zeitlinger, E. G. Jennings, H. L. Murray, D. B. Gordon, B. Ren, J. J. Wyrick, J. B. Tagne, T. L. Volkert, E. Fraenkel, D. K. Gifford, and R. A. Young, Transcriptional regulatory networks in *Saccharomyces cerevisiae*, *Science* 298(5594), 799 (2002)
39. L. Mendoza and I. Xenarios, A method for the generation of standardized qualitative dynamical systems of regulatory networks, *Theor. Biol. Med. Model* 3(1), 13 (2006)
40. B. Novak and J. J. Tyson, Modeling the control of DNA replication in fission yeast, *Proc. Natl. Acad. Sci. USA* 94(17), 9147 (1997)
41. A. Sveiczler, A. Csikász-Nagy, B. Gyorffy, J. J. Tyson, and B. Novak, Modeling the fission yeast cell cycle: Quantized cycle times in *wee1- cdc25Delta* mutant cells, *Proc. Natl. Acad. Sci. USA* 97(14), 7865 (2000)
42. B. Li, B. Shao, C. Yu, Q. Ouyang, and H. Wang, A mathematical model for cell size control in fission yeast, *J. Theor. Biol.* 264(3), 771 (2010)
43. M. Davidich and S. Bornholdt, The transition from differential equations to Boolean networks: A case study in simplifying a regulatory network model, *J. Theor. Biol.* 255(3), 269 (2008)
44. Z. Y. Zhang, Z. Y. Li, G. Hu, and Z. G. Zheng, Exploring cores and skeletons in oscillatory gene regulatory networks by a functional weight approach, *EPL* 105(1), 18003 (2014)
45. X. Liao, Q. Xia, Y. Qian, L. Zhang, G. Hu, and Y. Mi, Pattern formation in oscillatory complex networks consisting of excitable nodes, *Phys Rev E* 83, 056204 (2011)
46. B. Novak, A. Csikász-Nagy, B. Gyorffy, K. Nasmyth, and J. J. Tyson, Model scenarios for evolution of the eukaryotic cell cycle, *Philos. Trans. R. Soc. Lond. B* 353(1378), 2063 (1998)