From graph topology to ODE models for gene regulatory networks

Xiaohan Kang[1*], Bruce Hajek[1], and Yoshie Hanzawa[2]

1 Department of Electrical and Computer Engineering and Coordinated Science Laboratory, University of Illinois at Urbana–Champaign, Urbana, Illinois, United States of America
2 Department of Biology, California State University, Northridge, Northridge, California, United States of America

* xiaohan.kang1@gmail.com

Abstract

A gene regulatory network can be described at a high level by a directed graph with signed edges, and at a more detailed level by a system of ordinary differential equations (ODEs). The former qualitatively models the causal regulatory interactions between ordered pairs of genes, while the latter quantitatively models the time-varying concentrations of mRNA and proteins. A property, called the constant sign property, is proposed for a general class of ODE models. The constant sign property characterizes which models are consistent with a signed, directed graph. The ODE models that capture the effect of *cis*-regulatory elements involving protein complex binding, based on the model in the GeneNetWeaver source code, are described in detail and shown to satisfy the constant sign property. Such ODE models are shown to have great complexity due to many continuous parameters and combinatorial module configurations. Finally the question of how closely data generated by one ODE model can be fit by another ODE model is explored. It is observed that the fit is better if the two models come from the same graph.

1 Introduction

A goal of systems biology is to understand real world gene regulatory networks. A directed graph with vertices representing genes and signed edges representing gene-to-gene interactions, also known as a circuit model [1] or a logical model [2], is a model with a high level of abstraction (see Appendix S1). The ordinary differential equation (ODE) models are far more detailed: they quantitatively describe the dynamics of the time-varying mRNA and protein concentrations of the genes, and can be used to capture complex effects, including protein–protein interaction, post-translational modification, environmental signals, diffusion of proteins in different parts of the cell, and various time constants. As a result, ascribing a directed graph to a gene regulatory network can miss important biological details because of the abstraction. However, it is significantly more challenging to ascribe a particular ODE model to a gene regulatory network than to ascribe a directed graph with signed edges because the former requires much finer classification. As one example, the work [3] is notable for proposing an ODE model for a particular gene regulatory network. The ODE model in [3] is based on previous studies, and it is shown that parameters for the model can be

selected to give a good match to data. In general, however, the relation between the graph models and the ODE models is unclear. The purpose of this paper is to explore the connections between the two types of models.

We propose a property of the ODE models, called the constant sign property (CSP), such that an ODE model corresponds to a single graph model if and only if the ODE model satisfies CSP. It turns out that many ODE models satisfying CSP can correspond to the same graph model. Therefore, while the amount of data may not be sufficient to reconstruct an ODE model, it could still be sufficient to reconstruct a graph model (see Fig. 1).

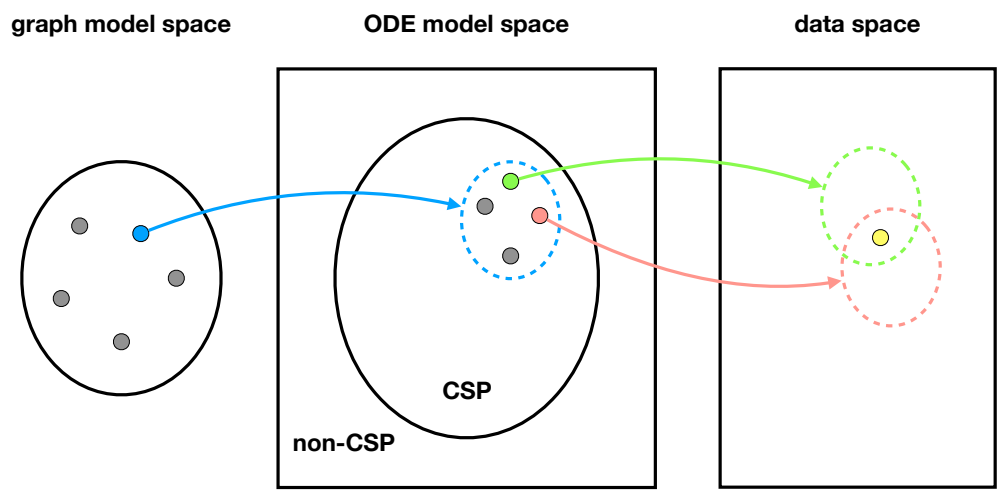

Fig 1. Relation between graph models, ODE models, and data. One graph model may correspond to multiple ODE models that satisfy CSP. Each ODE model may correspond to multiple possible datasets in the data space depending on the initial conditions, external signals, and model parameters. As an example, a dataset (the solid yellow circle) may be generated from two ODE models (the solid green and red circles), both of which correspond to the same graph model (the solid blue circle).

One particularly rich class of ODE models that satisfy CSP are based on GeneNetWeaver [4, 5], the software used to generate expression data in DREAM challenges 3–5. In these ODE models a layer of intermediate elements called *modules* are constructed with transcription factors (TFs) as their input and target genes their output. The activity level of a module depends on its input and its type, and determines the production rate of its output. The modules model the binding of protein complexes to DNA in transcriptional regulation. Assuming for each TF and each target gene there is only one module that takes the TF as an input and the target gene as an output, we show below that CSP is satisfied, so each GeneNetWeaver ODE model has a well-defined graph model associated with it. The combinatorial nature of the number of possible module configurations (i.e., the number of the modules and their input and output) and the continuous value parameters make the GeneNetWeaver ODE models extremely rich.

The organization of this paper is as follows. In Section 2.1 we describe the ODE models and the graph models, and propose CSP. In Section 2.2, we describe ODE models based on GeneNetWeaver. In Section 3.1, the GeneNetWeaver ODE models are shown to satisfy the constant sign property, and their complexity is investigated. In Section 3.2, a case study of a core soybean flowering network based on the literature is presented to demonstrate the use of the GeneNetWeaver ODE models. First it is illustrated that a single signed, directed graph model has a large space of consistent ODE models. Second, to study how different the GeneNetWeaver ODE models are, we

explore the problem of numerically fitting parameters of one ODE model to synthetic expression data generated from another.

2 Materials and methods

2.1 ODE model and constant sign property

In this section we define the constant sign property, a property under which ODE models are consistent with signed directed graphs. As we will see, CSP holds when unilaterally increasing the expression level of one gene always causes the change rate of another gene to move in one direction. In other words, the effect of one regulator gene has a constant sign on a target gene, regardless of the expression levels of the target gene or any other variables or the particular parameter values of the ODE model.

Let $x_1(t)$, $x_2(t)$, $\ldots$, $x_n(t)$ be the mRNA abundances for the n genes (the observables) at time t. Let $x_{n+1}(t)$, $x_{n+2}(t)$, $\ldots$, $x_{n+m}(t)$ be the protein concentrations (the unobservables) at time t, which may include derived (protein complexes and modifications like protein phosphorylation) and localized (e.g., cytoplasmic and nuclear) proteins. Let $x_{n+m+1}(t)$, $x_{n+m+2}(t)$, $\ldots$, $x_{n+m+l}(t)$ be the strengths of the chemical and environmental signals (the controllables, e.g., temperature and illumination) at time t. Let $x(t) = (x_i(t) \colon i \in [n+m+l])$ be the system state at time t, where $[n]$ denotes the set of integers $\{1, 2, \ldots, n\}$. Let $p \in \mathbb{R}^s$ be the parameters of the ODE model and let $f_i \colon \mathbb{R}^{n+m+l} \times \mathbb{R}^s \to \mathbb{R}$ be the time derivative of x_i as a function of the $n+m+l$ dimensional system state and the parameters for $i \in [n+m]$.

An ODE model is characterized by the tuple (n, m, l, f), where $f = (f_i \colon i \in [n+m])$. In addition, the detailed trajectories of the mRNA and protein concentrations evolving with time depend on $(x^0, \tilde{x}, p)$, where $x^0 = \left(x_i^0 \colon i \in [n+m]\right)$ are the initial conditions of the mRNAs and proteins, $\tilde{x} = (\tilde{x}_i(t) \colon n+m+1 \le i \le n+m+l, t \in [0, T])$ are the external signal strengths over the time period $[0, T]$, and $p \in \mathbb{R}^s$ are the parameters. The trajectories can be obtained by solving the following initial value problem.

$$x_i(0) = x_i^0, \quad i \in [n+m],$$

$$x_i(t) = \tilde{x}_i(t), \quad n+m+1 \le i \le n+m+l, t \in [0, T],$$

$$\frac{\mathrm{d}x_i(t)}{\mathrm{d}t} = f_i(x(t), p), \quad i \in [n+m].$$

Note the signals $(x_i \colon n+m+1 \le i \le n+m+l)$ are exogenously controlled. Given the ODE model characterized by (n, m, l, f), the *molecular graph* $([n+m+l], \mathcal{E}, w)$ is a weighted directed graph with vertices $[n+m+l]$, edges $\mathcal{E}$, and weights w, defined as follows. For any $i \in [m+n+l]$ and any $j \in [m+n]$ with $i \neq j$, there is an edge from i to j if f_j is non-constant in x_i for some p. Formally,

$$\mathcal{E} = \{(i, j) \in [m+n+l] \times [m+n] \colon \exists p \in \mathbb{R}^s \exists x \in \mathbb{R}^{n+m+l} \exists x_i' \in \mathbb{R} \\ \text{s.t. } f_j(x, p) \neq f_j((x_i', x_{-i}), p)\},$$

where $(x_i', x_{-i}) = (x_1, x_2, \ldots, x_{i-1}, x_i', x_{i+1}, \ldots, x_{n+m+l})$. Furthermore, the weight on the edge (i, j) is determined by

$$w_{ij} = \begin{cases} 1 & \text{if } f_j \text{ is non-decreasing in } x_i \text{ for any } p, \\ -1 & \text{if } f_j \text{ is non-increasing in } x_i \text{ for any } p, \\ 0 & \text{otherwise (i.e., } f_j \text{ is non-monotone in } x_i). \end{cases}$$

See Figure 2 for an illustration. Then $(i, j) \notin \mathcal{E}$ if f_j is constant in x_i. For $(i, j) \in \mathcal{E}$, $w_{ij} = 1$ if the molecular species i increases the change rate of the molecular species j,

$w_{ij} = -1$ if i decreases the change rate of j, and $w_{ij} = 0$ if the effect of i on the change rate of j is indeterminate (can both increase and decrease depending on the vectors x and p).

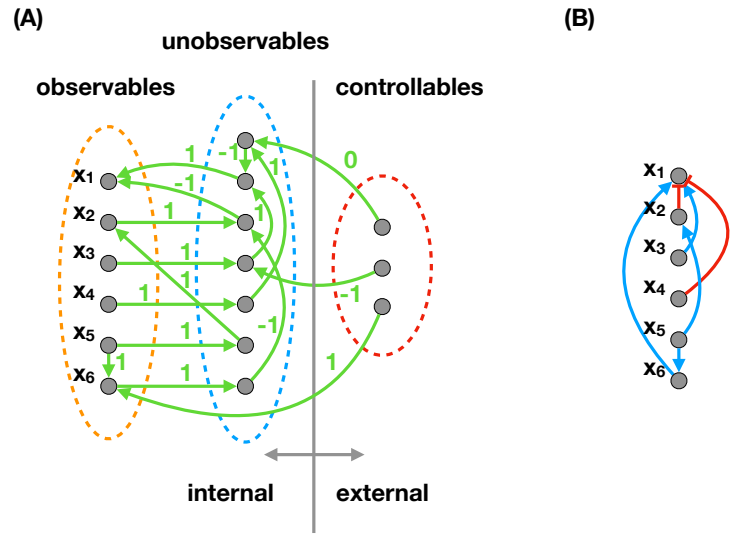

Fig 2. A molecular graph and its corresponding gene regulatory graph. (A) The three columns of vertices represent the observables (mRNA abundances), the unobservables (protein or protein complex concentrations), and the controllables (external signals). (B) The corresponding gene regulatory graph for (A), with blue edges indicating a weight of $b = 1$ and red edges indicating a weight of $b = -1$.

One can think of the molecular graph as the graph over the molecular species embedded in the information of the derivatives of the concentrations of the molecular species. Note no edges point to the controllables because they are controlled exogenously instead of internally by the dynamics. However, usually only the mRNA abundances are measured; the proteins and their derived products are not measured, making the molecular graph only partially observed. As a result, one often seeks an induced graph on the mRNA species, which requires the following definitions.

Definition 1 (CSP for an ordered pair of genes). For a molecular graph $([n+m+l], \mathcal{E}, w)$, the effect of gene i on j ($i, j \in [n]$, $i \neq j$) satisfies the constant sign property if there exists $b_{ij} \in \{1, -1\}$ such that for any integer $q \geq 2$ and any $i_k \in [n+m] \backslash [n]$, $2 \leq k \leq q-1$ with $i_1 = i$, $i_q = j$, $i_k \neq i_{k'}$ for $k \neq k'$, and $(i_k, i_{k+1}) \in \mathcal{E}$ for all $k \in [q-1]$, we have

$$\prod_{k=1}^{q-1} w_{i_k i_{k+1}} = b_{ij}.$$

In other words, the effect of i on j satisfies CSP if and only if all paths from i to j (not passing through another mRNA) give the same monotone relation ($b_{ij} = 1$ for increasing, and $b_{ij} = -1$ for decreasing). By convention if no paths exist from i to j then we still say CSP is satisfied, but with an indeterminate b_{ij}.

Definition 2 (CSP for an ODE model). An ODE model characterized by (n, m, l, f) satisfies the constant sign property if (i, j) satisfies the CSP for the molecular graph of the ODE model for all $i, j \in [n]$, $i \neq j$. Furthermore, the signed directed graph $([n], \mathcal{D}, b)$, with $(i, j) \in \mathcal{D}$ if and only if there exists a path from i to j in the molecular graph and b_{ij} determined in the definition above, is called the *gene regulatory graph*.

Remark 1. The controllables are external and do not affect the gene regulatory graph.

Definition 3 (CSP for an ensemble of ODE models). An ensemble of ODE models is said to satisfy the constant sign property if they all satisfy the CSP and have the same gene regulatory graph.

Remark 2. If multiple ODE models satisfy CSP with the same gene regulatory graph, then they can be combined into a single ODE model with different parameterization so that the combined ODE model still satisfies CSP with the same gene regulatory graph.

Remark 3. The effect of a gene on itself can be both autoregulation and degradation. Autoregulation can be included in the gene regulatory graph by removing all the self loops in the molecular graph, but this paper does not deal with this complexity and simply ignores autoregulation in the gene regulatory graphs.

Remark 4. In the rest of this paper we reserve symbol x for mRNA abundances, y for protein concentrations, and z for exogenous signals.

The following is an example of a non-CSP ODE model that is similar to the interactions among *FT*, *TFL1*, *FD*, and *LFY* genes in [6].

Example 1. Consider a four-gene ODE model where genes 1 and 3 form a protein complex that activates gene 4, while genes 2 and 3 form a protein complex that represses gene 4. For example, the dynamics of x_4 can be

$$\dot{x}_4 = \frac{x_1 x_3}{k_1 + x_1 x_3} \frac{k_2}{k_2 + x_2 x_3},$$

where we use x for both the mRNA and protein concentrations. Then it can be checked that the effect of gene 3 on gene 4 does not satisfy the CSP. Indeed, suppose $k_1 = k_2 = x_1 = x_2 = 1$. Then $\dot{x}_4 = \frac{x_3}{(1+x_3)^2}$, which increases with x_3 when $x_3 \leq 1$ and decreases with x_3 when $x_3 > 1$. Thus the effect of gene 3 on gene 4 violates the CSP.

2.2 GeneNetWeaver ODE model

We consider a differential equation model such that transcription factors participate in modules which bind to the promoter regions of a given target gene. This model is based on the GeneNetWeaver software version 3 [4]. Part of the model of the popular simulator is described in [7] and [5], but there is no good reference that precisely describes the model. So in this section we describe the generative model in GeneNetWeaver based on a given directed graph, and show in the next section that the CSP is satisfied.

The model in GeneNetWeaver is based on standard modeling assumptions (see [8]) including statistical thermodynamics, as described in [9]. The activity level of the promoter of a gene is controlled by one or more *cis*-regulatory modules, which for brevity we refer to as modules. A module can be either an enhancer or a silencer. Each module has one or more transcription factors as activators, and possibly one or more TFs as deactivators. For each target gene, a number of modules are associated with its TFs such that each TF is an input of one of the modules. For simplicity assume that each module regulates only a single target gene.

Let there be n genes indexed by $i \in [n] = \{1, 2, \ldots, n\}$. For target gene i, let $N_i \subset [n]$ be the set of its TFs and let $\mathcal{S}_i \subset \mathcal{P}(N_i)$ be a partition of N_i according to the input of the modules. Then the modules for target gene i can be indexed by the tuple (K, i) (denoted by $K : i$ in the subscripts), where $K \in \mathcal{S}_i$. Note each TF regulates the target gene i only through one module. The random model for assignment of the TFs to modules and of the parameters in GeneNetWeaver is summarized in Appendix S2. Let the sets of activators and deactivators for module $K : i$ be $A_{K:i}$ and $D_{K:i}$ with $A_{K:i} \cup D_{K:i} = N_i$ and $A_{K:i} \cap D_{K:i} = \emptyset$. For a module $K : i$, let $c_{K:i}$ be the type (1 for

enhancer and -1 for silencer), $r_{K:i}$ the mode (1 for synergistic binding and 0 for independent binding). Note $r_{K:i}$ only matters for multi-input modules (i.e., those with $|K| > 1$). Let $\beta_{K:i} \geq 0$ be the absolute effect of module $K : i$ on gene i in mRNA production rate.

Let $x_i(t)$ and $y_i(t)$ be the mRNA and protein concentrations for gene i at time t. We ignore t in the remainder of the paper for simplicity. The dynamics are given by

$$\frac{\mathrm{d}x_i}{\mathrm{d}t} = f_i(y) - \delta_i^{(\mathrm{m})} x_i$$

and

$$\frac{\mathrm{d}y_i}{\mathrm{d}t} = \lambda_i x_i - \delta_i^{(\mathrm{p})} y_i,$$

where $f_i(y)$ is the relative activation rate for gene i (i.e. the mRNA production rate for gene i for the normalized variables) discussed in the next two subsections, λ_i is the translation rate of protein i, and δ_i^{m} and δ_i^{p} are the degradation rates of the mRNA and the protein. Because only x is observed in RNA-seq experiments, without loss of generality the unit of the unobserved protein concentrations can be chosen such that $\lambda_i = \delta_i^{(\mathrm{p})}$ for all i (see nondimensionalization in [7]).

2.2.1 Activity level of a single module

For edge (i, j), the normalized expression level of gene i, ν_{ij}, is defined by

$$\nu_{ij} = \left(\frac{y_i}{k_{ij}}\right)^{h_{ij}},$$

where k_{ij} is the Michaelis–Menten normalizing constant and h_{ij} is a small positive integer, the Hill constant, representing the number of copies of the TF i that need to bind to the promoter region of gene j to activate the gene. (If gene i is not bound to the promoter region of gene j, it is like taking the Hill constant equal to zero and thus normalized expression level equal to one.) The activity level of module $K : j$ denoted by $M_{K:j}$, which is the probability that module $K : j$ is active, is given in the following three cases.

Type 1 modules: Input TFs bind to module independently In this case, $r_{K:j} = 0$, and we have

$$M_{K:j} = \left(\prod_{i \in A_{K:j}} \frac{\nu_{ij}}{1 + \nu_{ij}}\right) \left(\prod_{i \in D_{K:j}} \frac{1}{1 + \nu_{ij}}\right).$$

Interpreting each fraction as the probability that an activator is actively bound (or a deactivator is not bound), the activation $M_{K:j}$ is the probability that all the inputs of module $K : j$ are working together to activate the module, i.e., the probability that the module is active. It is assumed that for a module to be active, all the activators must be bound and all the deactivators must be unbound, and all the bindings happen independently.

One can think of module $K : j$ as a system with $2^{|A_{K:j}|+|D_{K:j}|}$ possible states of the inputs. Suppose each input j binds with rate ν_{ij} and unbinds with rate 1 independently. Then the stationary probability of the state that all the activators are bound and none of the deactivators is bound is $M_{K:j}$.

Alternatively, one can assign additive energy of

$$\begin{aligned} E_{ij} &= -\log \nu_{ij} \\ &= -h_{ij} \log \frac{y_i}{k_{ij}} \end{aligned}$$

to each bound input gene i and energy zero to each unbound gene. Then $M_{K:j}$ is the probability that all activators are bound and none of the deactivators is bound in the Gibbs measure.

Type 2 modules: TFs are all activators and bind to module as a complex
In this case, $D_{K:j} = \emptyset$, $r_{K:j} = 1$, and we have

$$M_{K:j} = \frac{\prod_{i \in A_{K:j}} \nu_{ij}}{1 + \prod_{i \in A_{K:j}} \nu_{ij}}.$$

One can think of such a module as a system with only two states: bound by the activator complex, or unbound. The transition rate from unbound to bound is $\prod_{i \in A_{K:j}} \nu_{ij}$, and that from bound to unbound is 1. Then the activation of the module is the probability of the bound state in the stationary distribution, given by $M_{K:j}$.

Alternatively, this corresponds to the Gibbs measure as in the previous case, except all the states other than fully unbound and fully bound are unstable (i.e. have infinite energy).

Type 3 modules: Some TFs are deactivators and bind to module as a complex
In this case, $D_{K:j} \neq \emptyset$ and $r_{K:j} = 1$, and we have

$$M_{K:j} = \frac{\prod_{i \in A_{K:j}} \nu_{ij}}{1 + \prod_{i \in A_{K:j}} \nu_{ij} + \left(\prod_{i \in A_{K:j}} \nu_{ij}\right)\left(\prod_{i \in D_{K:j}} \nu_{ij}\right)}.$$

In this case the system can be in one of three states: unbound, bound by the activator complex, and bound by the deactivated (activator) complex. The Gibbs measure such that states other than the three mentioned are unstable (i.e. have infinite energy) assigns probability $M_{K:j}$ to the activated state.

Note the third case does not reduce to the second case when $D_{K:j} = \emptyset$ because $\prod_{i \in D_{K:j}} \nu_{ij}$ is usually understood as 1 rather than 0. Missing the second case was considered a bug that originated from early versions of GeneNetWeaver.

Remark 5. Presumably it is possible for there to be more than three stable states for a module, so additional types of modules could arise, but for simplicity, following GeneNetWeaver, we assume at least one of the three cases above holds.

Remark 6. If a module $K : j$ has only one input i (i.e. $K = \{i\}$) then the module is type 1 and $M_{K:j} = \frac{\nu_{ij}}{1+\nu_{ij}}$ or $M_{K:j} = \frac{1}{1+\nu_{ij}}$. We will see later in the random model of GeneNetWeaver that only the former (single activator) is allowed.

GeneNetWeaver software uses the 3 types of modules derived above. In all three cases the activation $M_{K:j}$ is monotonically increasing in y_i for activators $i \in A_{K:j}$, and monotonically decreasing in y_i for deactivators $i \in D_{K:j}$.

2.2.2 Production rate as a function of multiple module activations

The relative activation of gene i as a function of the protein concentrations y is

$$f_i(y) = \sum_{s \in \{0,1\}^{\mathcal{S}_i}} \alpha_{i,s} \left(\prod_{K \in \mathcal{S}_i :\, s_K = 1} M_{K:i} \right) \left(\prod_{K \in \mathcal{S}_i :\, s_K = 0} (1 - M_{K:i}) \right), \tag{1}$$

where $\alpha_{i,s}$ is the relative activation of the promoter under the module configuration s. Note that α in (1) gives $2^{|\mathcal{S}_i|}$ degrees of freedom, one for every possible subset of the modules being active. However, following the GeneNetWeaver computer code [4], we assume that the interaction among the modules is linear, meaning that for some choice of $\alpha_{i,\text{basal}}$, $(c_{K:i} \colon K \in \mathcal{S}_i)$, and $(\beta_{K:i} \colon K \in \mathcal{S}_i)$, we have for any configuration $s \in \{0,1\}^{\mathcal{S}_i}$,

$$\alpha_{i,s} = \alpha_{i,\text{basal}} + \sum_{K \in \mathcal{S}_i \colon s_K = 1} c_{K:i}\beta_{K:i}, \tag{2}$$

This reduces the number of degrees of freedom for α to $|\mathcal{S}_i| + 1$. Then, combining (1) and (2) yields

$$\begin{aligned} f_i(y) &= \mathbb{E}\,\alpha_{i,S} \\ &= \alpha_{i,\text{basal}} + \sum_{K \in \mathcal{S}_i} c_{K:i}\beta_{K:i}\,\mathbb{E}\,S_K \\ &= \alpha_{i,\text{basal}} + \sum_{K \in \mathcal{S}_i} c_{K:i}\beta_{K:i}M_{K:i}, \end{aligned} \tag{3}$$

where S is distributed by the product distribution of the Bernoulli distributions with means $(M_{K:i} \colon K \in \mathcal{S}_i)$. So the relative activation, or the mRNA production rate, of a gene is given by the basal activation plus the inner product of the module effects and the module activation. We also note that the effect of the modules is not assumed to be statistically independent: all we need to know to compute the relative activation of a gene are the marginal probability of activation of the single modules.

Taking into account the three different types of modules described in Section 2.2.1, (3) yields the following expression for the relative activation of gene i :

$$\begin{aligned} f_i(y) = \alpha_{i,\text{basal}} &+ \sum_{K \colon r_{K:i}=0} c_{K:i}\beta_{K:i} \left(\prod_{j \in A_{K:i}} \frac{\nu_{ji}}{1+\nu_{ji}}\right)\left(\prod_{j \in D_{K:i}} \frac{1}{1+\nu_{ji}}\right) \\ &+ \sum_{\substack{K \colon r_{K:i}=1 \\ D_{K:i}=\emptyset}} c_{K:i}\beta_{K:i} \frac{\prod_{j \in A_{K:i}} \nu_{ji}}{1+\prod_{j \in A_{K:i}} \nu_{ji}} \\ &+ \sum_{\substack{K \colon r_{K:i}=1 \\ D_{K:i}\neq\emptyset}} c_{K:i}\beta_{K:i} \frac{\prod_{j \in A_{K:i}} \nu_{ji}}{1+\prod_{j \in A_{K:i}} \nu_{ji} + \left(\prod_{j \in A_{K:i}} \nu_{ji}\right)\left(\prod_{j \in D_{K:i}} \nu_{ji}\right)}. \end{aligned} \tag{4}$$

We see in (3) that f_i is monotone in $M_{K:i}$ (increasing if $c_{K:i} > 0$, and decreasing if $c_{K:i} < 0$), which is in turn monotone in its inputs (increasing in activators, and decreasing in deactivators). As a result, with each pair of regulator and target genes (j,i) correspond to a single module K, f_i is monotonically increasing in y_j if the module type and the input type are the same ($c_{K:i} = 1$ and $j \in A_{K:i}$, or $c_{K:i} = -1$ and $j \in D_{K:i}$). So the ODE model with production function (4) satisfies the CSP.

Note that in the actual GeneNetWeaver source code every $\alpha_{i,s}$ is truncated to the interval $[0,1]$:

$$\alpha_{i,s} = \left[\alpha_{i,\text{basal}} + \sum_{K \in \mathcal{S}_i \colon s_K = 1} c_{K:i}\beta_{K:i}\right]_0^1,$$

where $[x]_0^1 = \max\{\min\{x,1\},0\}$ is the projection of x to the $[0,1]$ interval. Then the relative activation in each state may not be linear in the individual module effects. In that case one has to resort to (1) instead of (4) for computing the mRNA production rate. The resulting truncated model does not necessarily satisfy the CSP because f_i may not be monotone in $M_{K:i}$ in (1).

3 Results and Discussion

3.1 GeneNetWeaver: CSP and complexity

As we saw in Section 2.2, the computer simulation data in the DREAM challenges are based on signed directed graph inputs, and the randomly generated ODE models (without the truncation of the α terms in the implementation) are always consistent with the signed directed graphs. Thus, the network models satisfy the constant sign property. Moreover, when data is generated through multifactorial perturbation for the DREAM challenge (primarily for generation of stationary expression levels, rather than trajectories), each ensemble of networks produced is also associated with the same directed signed graph, and thus the ensembles satisfy the constant sign property.

We now discuss the complexity of GeneNetWeaver ODE models for a given gene regulatory graph. The complexity comes from both the large number of parameters and the combinatorial nature of the module configurations. The complexity indicates that ODE models are both much more detailed and considerably harder to infer compared to the graphical models.

For each gene i there are 5 non-negative real parameters ($\alpha_{i,\text{basal}}$, $x_i(0)$, $y_i(0)$, $\delta_i^{(\text{m})}$, $\delta_i^{(p)}$). For each edge (i, j) there is a non-negative real parameter (k_{ij}) and an integer parameter (h_{ij}). For each module $K : i$ there is a positive real parameter ($\beta_{K:i}$) and two binary parameters ($c_{K:i}$ and $r_{K:i}$).

The module configuration encodes great combinatorial complexity. Given a gene has $K \geq 1$ input genes, the number of ways to partition the genes into modules is the Kth Bell number. The first ten Bell numbers are 1, 2, 5, 15, 52, 203, 877, 4140, 21147, and 115975. In addition, each input to a given module needs to be classified as an activator or deactivator.

3.2 Case study: soybean flowering networks

In this section the similarities of the ODE models corresponding to three different graph models are studied. First the classes of ODE models are listed for the three graph models. Then, to investigate their similarities, we generate expression data from one ODE model, and fit another model to the data by optimizing the parameters. The level of fitness of one class of ODE model to the data generated from another is used as a metric of similarity. As we will see, ODE models corresponding to the same graph model tend to have a higher similarity, while those from different graph models tend to have a lower similarity, as long as the least-squares problem is sufficiently overdetermined. The result implies that the graph model corresponding to the ODE model may be recovered with moderate amount of data, while the amount of data required for ODE model recovery may be of a much higher order. The simulation code for the data fitting results is available at [10].

3.2.1 Five-gene graph and ODE models

In this section we explicitly write out the classes of GeneNetWeaver ODE models of three graph models. The first two graph models are compiled from the literature, with only the sign of one edge different between them (the difference is discovered in [11]). The third graph model is an arbitrary five-gene repressilator [12] for comparison purpose.

Flowering network with *COL1a* activating *E1* A graph model of a five-gene soybean flowering network is shown in Figure 3. The network is based on the flowering network for *Arabidopsis* and homologs of *Arabidopsis* genes found in soybean (see

references in Table 1). The corresponding gene IDs are shown in Table 2. The mRNA

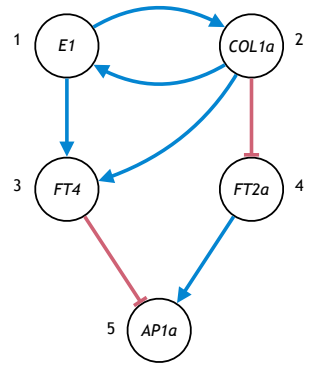

Fig 3. A graph model of the core flowering network for soybean.

regulatory interaction	reference
E1 activates *COL1a*	[13]
E1 activates *FT4*	[14]
COL1a activates *E1*	[11]
COL1a represses *E1*	[13]
COL1a activates *FT4*	[13], [11]
COL1a represses *FT2a*	[13], [11]
FT4 represses *AP1a*	[14]*
FT2a activates *AP1a*	[15]

Table 1. Core flowering genes.
* For *FT4* only, not for the interaction with *AP1a*.

index	gene ID	gene name
1	Glyma.06G207800	*E1*
2	Glyma.08G255200	*COL1a*
3	Glyma.08G363100	*FT4*
4	Glyma.16G150700	*FT2a*
5	Glyma.16G091300	*AP1a*

Table 2. Core flowering genes.

and proteins concentrations of the soybean genes *E1*, *COL1a*, *FT4*, *FT2a*, and *AP1a* are denoted by $(x_i)_{1\leq i\leq 5}$ and $(y_i)_{1\leq i\leq 5}$. The differential equations based on the GeneNetWeaver model are

$$\dot{x}_1 = \alpha_{1,\text{basal}} + \frac{(y_2/k_{21})^{h_{21}}}{1+(y_2/k_{21})^{h_{21}}}\beta_{2:1} - \delta_1^{(\text{m})} x_1. \tag{5}$$

$$\dot{x}_2 = \alpha_{2,\text{basal}} + \frac{(y_1/k_{12})^{h_{12}}}{1+(y_1/k_{12})^{h_{12}}}\beta_{1:2} - \delta_2^{(\text{m})} x_2. \tag{6}$$

$$\begin{cases} \dot{x}_3 = \alpha_{3,\text{basal}} + \frac{(y_1/k_{13})^{h_{13}}}{1+(y_1/k_{13})^{h_{13}}} \frac{(y_2/k_{23})^{h_{23}}}{1+(y_2/k_{23})^{h_{23}}} \beta_{12:3} - \delta_3^{(\text{m})} x_3 \\ \qquad\qquad \text{(independent binding), or} \\ \dot{x}_3 = \alpha_{3,\text{basal}} + \frac{(y_1/k_{13})^{h_{13}} (y_2/k_{23})^{h_{23}}}{1+(y_1/k_{13})^{h_{13}} (y_2/k_{23})^{h_{23}}} \beta_{12:3} - \delta_3^{(\text{m})} x_3 \\ \qquad\qquad \text{(synergistic binding), or} \\ \dot{x}_3 = \alpha_{3,\text{basal}} + \frac{(y_1/k_{13})^{h_{13}}}{1+(y_1/k_{13})^{h_{13}}} \beta_{1:3} + \frac{(y_2/k_{23})^{h_{23}}}{1+(y_2/k_{23})^{h_{23}}} \beta_{2:3} - \delta_3^{(\text{m})} x_3 \\ \qquad\qquad \text{(two modules).} \end{cases} \tag{7}$$

$$\dot{x}_4 = \left(\alpha_{4,\text{basal}} - \frac{(y_2/k_{24})^{h_{24}}}{1+(y_2/k_{24})^{h_{24}}} \beta_{2:4} \right)^+ - \delta_4^{(\text{m})} x_4. \tag{8}$$

$$\begin{cases} \dot{x}_5 = \alpha_{5,\text{basal}} + \frac{1}{1+(y_3/k_{35})^{h_{35}}} \frac{(y_4/k_{45})^{h_{45}}}{1+(y_4/k_{45})^{h_{45}}} \beta_{34:5} - \delta_5^{(\text{m})} x_5 \\ \qquad\qquad \text{(independent binding enhancer), or} \\ \dot{x}_5 = \left(\alpha_{5,\text{basal}} - \frac{(y_3/k_{35})^{h_{35}}}{1+(y_3/k_{35})^{h_{35}}} \frac{1}{1+(y_4/k_{45})^{h_{45}}} \beta_{34:5} \right)^+ - \delta_5^{(\text{m})} x_5 \\ \qquad\qquad \text{(independent binding silencer), or} \\ \dot{x}_5 = \alpha_{5,\text{basal}} + \frac{(y_4/k_{45})^{h_{45}}}{1+(y_4/k_{45})^{h_{45}} + (y_3/k_{35})^{h_{35}} (y_4/k_{45})^{h_{45}}} \beta_{34:5} - \delta_5^{(\text{m})} x_5 \\ \qquad\qquad \text{(synergistic binding enhancer), or} \\ \dot{x}_5 = \left(\alpha_{5,\text{basal}} - \frac{(y_3/k_{35})^{h_{35}}}{1+(y_3/k_{35})^{h_{35}} + (y_3/k_{35})^{h_{35}} (y_4/k_{45})^{h_{45}}} \beta_{34:5} \right)^+ - \delta_5^{(\text{m})} x_5 \\ \qquad\qquad \text{(synergistic binding silencer), or} \\ \dot{x}_5 = \left(\alpha_{5,\text{basal}} - \frac{(y_3/k_{35})^{h_{35}}}{1+(y_3/k_{35})^{h_{35}}} \beta_{3:5} + \frac{(y_4/k_{45})^{h_{45}}}{1+(y_4/k_{45})^{h_{45}}} \beta_{4:5} \right)^+ - \delta_5^{(\text{m})} x_5 \\ \qquad\qquad \text{(two modules).} \end{cases} \tag{9}$$

$$\dot{y}_1 = \lambda_1 (x_1 - y_1). \tag{10}$$

$$\dot{y}_2 = \lambda_2 (x_2 - y_2). \tag{11}$$

$$\dot{y}_3 = \lambda_3 (x_3 - y_3). \tag{12}$$

$$\dot{y}_4 = \lambda_4 (x_4 - y_4). \tag{13}$$

$$\dot{y}_5 = \lambda_5 (x_5 - y_5). \tag{14}$$

Here $(x)^+ = \max\{x, 0\}$. We apply non-dimensionalization by setting $\delta_i = \alpha_{i,\text{basal}} + \sum_j \beta_{j:i}$, so that the steady state expression levels are between 0 and 1. We can see that given the graph, there are 15 configurations of the ODEs (3 for x_3 times 5 for x_5). We use $[i, j]$ with $1 \le i \le 3$ and $1 \le j \le 5$ to denote the configuration using the ith equation for x_3 and the jth equation for x_5, and use the symbol $\mathsf{F}_{[i,j],+}$ to denote the class of flowering network ODE models with configurations $[i, j]$ (the plus sign signifies the activation regulation of *COL1a* on *E1*). The initial conditions, namely the 5 mRNA abundances $x(0)$'s and the 5 protein concentrations $y(0)$'s, are 10-dimensional. In addition, there are 24–26 positive real parameters (depending on the configuration) and 7 discrete parameters (the Hill coefficients) for the dynamics. For example, for configuration [1, 1], the parameters for the dynamics consist of the basal activations α's (5), the Michaelis–Menten constants k's (7), the absolute effect of modules β's (7), the translation rate λ's (5), summing up to 24 parameters.

Flowering network with *COL1a* repressing *E1* A slight variant of the soybean flowering graph model in Figure 3 is shown in Figure 4. Note the only difference is the sign of the edge from *COL1a* to *E1*. The symbol $\mathsf{F}_{[i,j],-}$ denotes the class of ODE models (5)–(14) with the ith and the jth configurations in (7) and (9), but with (5) replaced by

$$\dot{x}_1 = \left(\alpha_{1,\text{basal}} - \frac{(y_2/k_{21})^{h_{21}}}{1+(y_2/k_{21})^{h_{21}}} \beta_{2:1} \right)^+ - \delta_1^{(\text{m})} x_1. \tag{15}$$

Here the negative sign in $\mathsf{F}_{[i,j],-}$ signifies the repression regulation of *COL1a* on *E1*. The number of parameters is the same as the network in Figure 3.

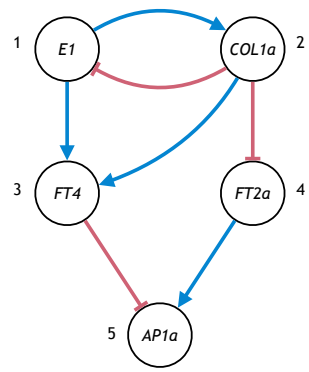

Fig 4. A variant of the graph model of the core flowering network for soybean.

Repressilator An arbitrary repressilator network is shown in Figure 5. The symbol

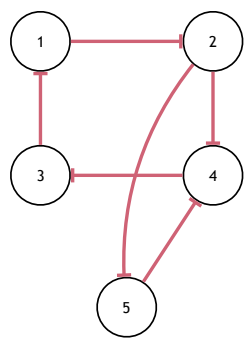

Fig 5. A five-gene repressilator graph model.

R denotes the class of ODE models for the repressilator, given below.

$$\dot{x}_1 = \left(\alpha_{1,\text{basal}} - \frac{(y_3/k_{31})^{h_{31}}}{1+(y_3/k_{31})^{h_{31}}}\beta_{3:1}\right)^+ - \delta_1^{(\text{m})} x_1. \tag{16}$$

$$\dot{x}_2 = \left(\alpha_{2,\text{basal}} - \frac{(y_1/k_{12})^{h_{12}}}{1+(y_1/k_{12})^{h_{12}}}\beta_{1:2}\right)^+ - \delta_2^{(\text{m})} x_2. \tag{17}$$

$$\dot{x}_3 = \left(\alpha_{3,\text{basal}} - \frac{(y_4/k_{43})^{h_{43}}}{1+(y_4/k_{43})^{h_{43}}}\beta_{4:3}\right)^+ - \delta_3^{(\text{m})} x_3. \tag{18}$$

$$\dot{x}_4 = \left(\alpha_{4,\text{basal}} - \frac{(y_5/k_{54})^{h_{54}}}{1+(y_5/k_{54})^{h_{54}}}\beta_{5:4}\right)^+ - \delta_4^{(\text{m})} x_4. \tag{19}$$

$$\dot{x}_5 = \left(\alpha_{5,\text{basal}} - \frac{(y_2/k_{25})^{h_{25}}}{1+(y_2/k_{25})^{h_{25}}}\beta_{2:5}\right)^+ - \delta_5^{(\text{m})} x_5. \tag{20}$$

There is only one possible configuration for each target gene. The dynamics involve 20 parameters.

3.2.2 Data generation

The synthetic expression dataset is generated as follows. For the generated data, we use $\mathsf{F}_{[1,1],+}$ (the flowering network with configuration [1, 1] and *COL1a* activating *E1*) with a fixed set of parameters for the dynamics. For a single set of trajectories (i.e., for a single plant), we use a set of initial values $x(0)$'s and $y(0)$'s generated uniformly at random between 0 and 1. The entire dataset may consist of only a single set of trajectories, corresponding to a single plant; or the dataset may consist of multiple sets of trajectories, corresponding to multiple plants. If multiple sets of trajectories are used, the initial conditions for each set of trajectories are generated independently, while the parameters for the dynamics are the same across all sets of trajectories. In other words, we model distinct plants by assuming distinct initial conditions, while using common parameters for the dynamics. To produce the data, the x variables are sampled at time points 0, 1, 2, 3, 4, 5, 6, so that each set of trajectories (i.e., each plant) produces 35 data points. Because each set of trajectories is sampled at different times from the system with one initial condition representing different stages of a single plant, the synthetic datasets are of multi-shot sampling, as opposed to one-shot sampling in practice where each individual is only sampled once [16]. We also generate random expression datasets with reflected Brownian motions with covariance 0.05, and denote such a stochastic model by B.

3.2.3 Fitting results

The counts for data points and parameters are summarized in Table 3. Note that with a single set of trajectories, the number of parameters is close to the number of data points. As the number of sets of trajectories increases, the number of data points outgrows the number of parameters because each additional set provides 35 new data points while only allowing 10 more parameters from the initial conditions (because the dynamic parameters are shared across all sets of trajectories).

S (number of sets of trajectories)	1	2	5	10
STn (number of data points)	35	70	175	350
$\mathsf{F}_{[1,1],+}$	34	44	74	124
$\mathsf{F}_{[3,5],+}$	36	46	76	126
$\mathsf{F}_{[1,1],-}$	34	44	74	124
R	30	40	70	120

Table 3. Number of parameters in different ODE models.

The data fitting results are shown in Table 4. The fitting loss function for two $S \times T \times n$ tensors x and $\hat{x}$ is defined by

$$l(x, \hat{x}) = \left(\frac{1}{STn} \sum_{i=1}^{S} \sum_{j=1}^{T} \sum_{k=1}^{n} (x_{ijk} - \hat{x}_{ijk})^2 \right)^{1/2},$$

where S is the number of sets of trajectories in the dataset, T the number of time points, and n the number of genes.

We make the following observations from Table 4.

1. The implemented optimization algorithm failed to find the optimal parameters in row 1 (the best fit should be a perfect fit with zero loss), but the relative loss compared to the average non-dimensionalized expression level 0.5 is very small (less than 0.5%), indicating a near-optimal fit.

S (number of sets of trajectories)	1	2	5	10
fit $\mathsf{F}_{[1,1],+}$ model to $\mathsf{F}_{[1,1],+}$ data	0.0015	0.0015	0.0010	0.0009
fit $\mathsf{F}_{[3,5],+}$ model to $\mathsf{F}_{[1,1],+}$ data	0.0016	0.0021	0.0019	0.0021
fit $\mathsf{F}_{[1,1],-}$ model to $\mathsf{F}_{[1,1],+}$ data	0.0032	0.0036	0.0165	0.0208
fit R model to $\mathsf{F}_{[1,1],+}$ data	0.0030	0.0037	0.0148	0.0204
fit $\mathsf{F}_{[1,1],+}$ model to B data	0.1269	0.1125	0.1307	0.1390

Table 4. Fitting losses using different classes of ODE models on different synthetic datasets.

2. ODE models from all three graph models (rows 1, 2, 3, and 4) fit the synthetic flowering network data well when there are only one or two sets of trajectories (columns 1 and 2). The relative losses are less than 1%. We can see from Table 3 that the number of data points is close to the number of parameters in the $S = 1$ setting, and only moderately larger in the $S = 2$ setting. Thus one may not be able to infer the graph structure with very limited data.

3. When fitting the models to 5 or 10 sets of trajectories simultaneously, i.e., when the system is sufficiently overdetermined, only the models from the correct graph (rows 1 and 2) fit well. The models from incorrect graphs (rows 3 and 4) suffer a roughly 4% relative loss after fitting for 10 sets of trajectories. Note that $\mathsf{F}_{[1,1],-}$ differs from the ground truth of the data $\mathsf{F}_{[1,1],+}$ only by the sign of one edge, while the model R shares no edges in common with the ground truth at all. Yet the fitness of the slight variant of the ground truth graph is as bad as the completely different repressilator graph.

4. Both $\mathsf{F}_{[1,1],+}$ and $\mathsf{F}_{[3,5],+}$ fit the $\mathsf{F}_{[1,1],+}$ data very well for all numbers of sets of trajectories (rows 1 and 2). This indicates the classes of ODE models with different configurations of the same graph model are similar in terms of data fitting. Consequently, even with data sufficient to infer the correct graph model, it may be impossible to infer the specific ODE model.

5. The models from the flowering network cannot fit the random dataset (reflected Brownian motions with covariance 0.05) well. It turns out that the ODE models with 34 parameters have trouble following the highly variable 35 data points from the reflected Brownian motions. The low fitness level to the random dataset shows great redundancy in the parameters in terms of generating data points. It also indicates the fitting results to the synthetic ODE data are significant compared to fitting a random dataset.

4 Conclusion

Gene regulatory networks are modeled at different abstraction levels with tradeoff between accuracy and tractability. Graph models with signed directed edges provide circuit-like characterization of gene regulation, while ODE models quantify detailed dynamics for various molecular species. The constant sign property proposed in this paper allows ODE models to be mapped uniquely to graph models, and thus gives a natural classification of general ODE models. While ODE models that do not satisfy the constant sign property do exist in the literature, a class of ODE models for a given graph model based on the source code of a popular software package GeneNetWeaver is described in detail and shown to satisfy the constant sign property. Exploration of data fitting of one ODE model to the data generated from another shows better fit when two models have the same graph model.

Appendix

S1 Basic model of gene interaction

In molecular biology, DNA is transcribed into mRNAs, which in turn are translated into proteins. A gene is a region of the DNA that encodes the corresponding mRNAs and proteins. We say a gene is expressed if it is producing the corresponding proteins, and its expression level refers to the concentration of those proteins.

We consider the transcriptional regulatory networks, which involve the interaction of DNA, mRNA and proteins. In this type of regulatory networks, some proteins are transcription factors (TFs) regulating the transcription of other genes by turning them on or off. For example, gene X produces protein X, which may be a TF that activates

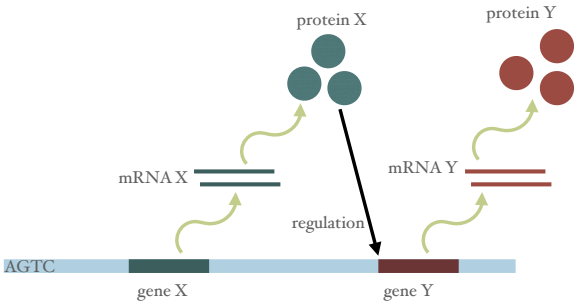

Fig 6. Direct binding of TFs to the promoter region of the target gene.

gene Y by increasing the transcription rate of gene Y (turning it on). Some TFs are repressors, whose presence decreases the production rate of the genes they regulate (turning them off). See Figures 6 and 7 for an illustration. This model suggests summarizing a gene regulatory network by a directed graph with signed edges, such that the vertices represent genes and the sign on an edge indicates activation or repression.

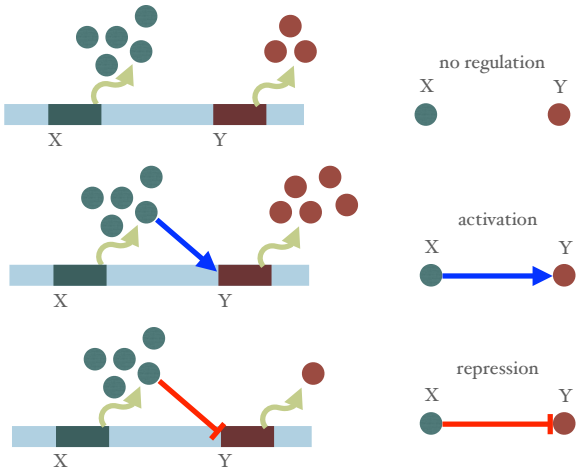

Fig 7. Types of transcriptional regulation.

The gene expression in the transcriptional regulatory networks can be modeled with various levels of granularity. Some use multiple variables for each gene (see, e.g., [17] and [7]), while others use a single variable to represent both the mRNA and the protein concentrations of a single gene (see, e.g., [8]). Some model the interaction between TFs

and the promoter region of the target genes directly [8], while others consider intermediate elements called *cis*-regulatory modules (CRMs) between the TFs and the promoters (see [7] and the GeneNetWeaver code [4]).

Furthermore, the transcriptional gene expression can also be affected by protein–protein interaction. One type of such interaction is signal transduction, which most commonly consists of the process of protein phosphorylation catalyzed by protein kinases. The phosphorylated proteins are active TFs, which bind the DNA and regulate transcription (see Chapter 6.3 in [8] and [18]). Another type of such interaction is protein complex forming, where multiple proteins can bind and form a protein complex that act as an active TF.

S2 Random model for production functions used in GeneNetWeaver

The materials and methods in the main body describes the family of ODE modules for gene regulatory networks, involving *cis*-regulatory modules, used in the GeneNetWeaver code. A main purpose of GeneNetWeaver is to generate random instances of such networks, which includes selecting how to partition the inputs to a given gene into modules and selection of parameter values. The probabilistic model of GeneNetWeaver is summarized in this section.

S2.1 Random module structure

An input to the GeneNetWeaver simulator is a signed, directed graph that specifies for any target gene i, which other genes are inputs that directly influence it, and the signs of the influence. The production function for a target gene i is randomly generated as follows. First, the inputs are partitioned to a random number of modules according to the following process.

1. Inputs are randomly reordered.
2. Module set A is initialized empty.
3. Each input in order is uniformly added to one of the modules in A or a new module, all with probability $\frac{1}{|A|+1}$. The module set A is updated if new modules are created.

The parameter $c_{K:i}$ determines the type of the module $K : i$ (enhancer or silencer). A given input gene in module $K : i$ is an activator if the sign of the gene (in the signed, directed graph) is the same as $c_{K:i}$, and otherwise the given input gene is a deactivator. The software ensures that each module has at least one activator for signed graph input, which means that if for module $K : i$, all the input genes have the same sign, then the $c_{K:i}$ should be the same as the sign of the input genes.

S2.2 Random parameter initialization

This section describes the random initialization of the parameters. Note the truncation is implemented by regenerating i.i.d. random variable until it falls within the desired region, which effectively scales up the probability density restricted to the region to form a proper probability distribution.

- The Michaelis–Menten coefficient is generated by $k_{ij} \sim \mathsf{Unif}([0.01, 1])$.
- The Hill coefficient is generated by $h_{ij} \sim \mathcal{N}(2, 2^2)$ truncated to $[1, 10]$.

- The mRNA degradation rate is $\delta_i^{(\mathrm{m})} = \log(2)/T_{1/2}$, where $T_{1/2} \sim \mathcal{N}(27.5, (7.5)^2)$ truncated to $[5, 50]$. As a result, $\delta_i^{(\mathrm{m})} \in [0.014, 0.14]$.
- The maximum translation rate and the protein degradation rate are both given by the same $\lambda_i = \delta_i^{(\mathrm{p})} = \log(2)/T_{1/2}$, where $T_{1/2} \sim \mathcal{N}(27.5, (7.5)^2)$ truncated to $[5, 50]$ is a realization independent with that for the mRNA degradation rate. Again $\delta_i^{(\mathrm{p})} \in [0.014, 0.14]$.

The basal activation $\alpha_{i,\mathrm{basal}}$ and the vector of absolute module effect parameters $\beta_{K:i}$ are generated using the following procedure. For target gene i:

1. If $N_i = \emptyset$, set $\alpha_{i,\mathrm{basal}} = 1$.
2. Otherwise set $\beta_{K:i} \sim \mathcal{N}(5/8, (1/8)^2)$ truncated to $[1/4, 1]$ for all $K \in \mathcal{S}_i$.
3. If $c_{K:i} = -1$ for all $K \in \mathcal{S}_i$, set $\alpha_{i,\mathrm{basal}} = 1$.
4. If $c_{K:i} = 1$ for all $K \in \mathcal{S}_i$, set $\alpha_{i,\mathrm{basal}} \sim \mathcal{N}(0, (0.05)^2)$ truncated to $[0, 1/4]$.
5. Otherwise $\alpha_{i,\mathrm{basal}} \sim \mathcal{N}(1/2, (1/12)^2)$ truncated to $[1/4, 3/4]$.
6. If $\alpha_{i,\mathrm{basal}} + \sum_{K:\, c_{K:i}=1} \beta_{K:i} < 1$, increase the smallest β therein to reach a maximum activation of 1.
7. If $\alpha_{i,\mathrm{basal}} + \sum_{K:\, c_{K:i}=-1} \beta_{K:i} > 0.25$, increase the smallest β therein to reach a random minimum activation of $\mathcal{N}(0, (0.05)^2)$ truncated to $[0, 1/4]$.
8. Finally, truncate all $\alpha_{i,s}$ to $[0, 1]$.

The resulting $\alpha_{i,s}$'s have a good coverage of the interval $[0, 1]$.

References

1. Kim HD, Shay T, O'Shea EK, Regev A. Transcriptional regulatory circuits: Predicting numbers from alphabets. Science. 2009;325(5939):429–432. doi:10.1126/science.1171347.
2. Karlebach G, Shamir R. Modelling and analysis of gene regulatory networks. Nat Rev Mol Cell Biol. 2008;9(10):770–780. doi:10.1038/nrm2503.
3. Seaton DD, Smith RW, Song YH, MacGregor DR, Stewart K, Steel G, et al. Linked circadian outputs control elongation growth and flowering in response to photoperiod and temperature. Mol Syst Biol. 2015;11(1). doi:10.15252/msb.20145766.
4. Schaffter T, Marbach D. GeneNetWeaver; 2012. Available from: `https://github.com/tschaffter/gnw`.
5. Schaffter T, Marbach D, Floreano D. GeneNetWeaver: *in silico* benchmark generation and performance profiling of network inference methods. Bioinformatics. 2011;27(16):2263–2270. doi:10.1093/bioinformatics/btr373.
6. Jaeger KE, Pullen N, Lamzin S, Morris RJ, Wigge PA. Interlocking feedback loops govern the dynamic behavior of the floral transition in *Arabidopsis*. Plant Cell. 2013;25(3):820–833.

7. Marbach D, Prill RJ, Schaffter T, Mattiussi C, Floreano D, Stolovitzky G. Revealing strengths and weaknesses of methods for gene network inference. Proc Natl Acad Sci USA. 2010;107(14):6286–6291. doi:10.1073/pnas.0913357107.

8. Alon U. An introduction to systems biology: Design principles of biological circuits. CRC press; 2006.

9. Ackers GK, Johnson AD, Shea MA. Quantitative model for gene regulation by lambda phage repressor. Proc Natl Acad Sci USA. 1982;79(4):1129–1133. doi:10.1073/pnas.79.4.1129.

10. Kang X. Graph and ODE models simulations; 2020. Available from: `https://github.com/Veggente/graph-ode`.

11. Wu F, Kang X, Wang M, Haider W, Price WB, Hajek B, et al. Transcriptome-enabled network inference revealed the *GmCOL1* feed-forward loop and its roles in photoperiodic flowering of soybean. Front Plant Sci. 2019;10. doi:10.3389/fpls.2019.01221.

12. Pokhilko A, Fernández AP, Edwards KD, Southern MM, Halliday KJ, Millar AJ. The clock gene circuit in *Arabidopsis* includes a repressilator with additional feedback loops. Mol Syst Biol. 2012;8(1):574. doi:10.1038/msb.2012.6.

13. Cao D, Li Y, Lu S, Wang J, Nan H, Li X, et al. *GmCOL1a* and *GmCOL1b* function as flowering repressors in soybean under long-day conditions. Plant Cell Physiol. 2015;56(12):2409–2422. doi:10.1093/pcp/pcv152.

14. Zhai H, Lü S, Liang S, Wu H, Zhang X, Liu B, et al. *GmFT4*, a homolog of *FLOWERING LOCUS T*, is positively regulated by *E1* and functions as a flowering repressor in soybean. PLOS ONE. 2014;9(2):e89030. doi:10.1371/journal.pone.0089030.

15. Nan H, Cao D, Zhang D, Li Y, Lu S, Tang L, et al. *GmFT2a* and *GmFT5a* redundantly and differentially regulate flowering through interaction with and upregulation of the bZIP transcription factor *GmFDL19* in soybean. PLOS ONE. 2014;9(5):e97669. doi:10.1371/journal.pone.0097669.

16. Kang X, Hajek B, Wu F, Hanzawa Y. Time series experimental design under one-shot sampling: The importance of condition diversity. PLOS ONE. 2019;14(10):e0224577. doi:10.1371/journal.pone.0224577.

17. Locke JCW, Millar AJ, Turner MS. Modelling genetic networks with noisy and varied experimental data: The circadian clock in *Arabidopsis thaliana*. J Theor Biol. 2005;234(3):383–393. doi:10.1016/j.jtbi.2004.11.038.

18. Tyson JJ, Chen KC, Novak B. Sniffers, buzzers, toggles and blinkers: Dynamics of regulatory and signaling pathways in the cell. Curr Opin Cell Biol. 2003;15(2):221–231. doi:10.1016/S0955-0674(03)00017-6.