Citation Mustafa Ozen et al 2022 Phys. Biol. 19 066004
DOI 10.1088/1478-3975/ac920d

Learning Feedback Molecular Network Models Using Integer Linear Programming

Mustafa Ozen[a], Effat S. Emamian[b], and Ali Abdi*[c]

Abstract: Analysis of intracellular molecular networks has many applications in understanding of the molecular bases of some complex diseases and finding effective therapeutic targets for drug development. To perform such analyses, the molecular networks need to be converted into computational models. In general, network models constructed using literature and pathway databases may not accurately predict experimental network data. This can be due to the incompleteness of literature on molecular pathways, the resources used to construct the networks, or some conflicting information in the resources. In this paper, we propose a network learning approach via an integer linear programming formulation that can systematically incorporate biological dynamics and regulatory mechanisms of molecular networks in the learning process. Moreover, we present a method to properly consider the feedback paths, while learning the network from data. Examples are also provided to show how one can apply the proposed learning approach to a network of interest. In particular, we apply the framework to the ERBB signaling network, to learn it from some experimental data. Overall, the proposed methods are useful for reducing the gap between the curated networks and experimental data, and result in calibrated networks that are more reliable for making biologically meaningful predictions.

Keywords: Feedbacks, Integer linear programming, Learning network models, Machine learning, Molecular networks, Network modeling

1 Introduction

Molecular networks are the networks of biochemical interactions between the molecules. They can be portrayed as directed graphs in which nodes represent biological molecules, i.e., proteins, genes etc., and edges represent biochemical interactions between the molecules[1-4]. Research and development of such networks has application in target discovery and drugs development, and analyzing the role of the molecular component in disease pathogeneses[5,6], understanding cellular decision making processes[7,8], understanding cell development and cell differentiation[9], developing molecular fault diagnosis and signaling capacity analysis methods[10-13], identifying disease subtypes and their regulators[14], and many other applications for better understanding of human diseases. Hence, constructing and analyzing molecular network models have emerged particularly over the past decade as an important area of systems biology research.

To study molecular networks, one needs to convert the molecular network graphs into computational models so that they can be analyzed to obtain novel and biologically relevant results. Continuous and discrete models have been widely studied so far. One way to model molecular networks is to convert them into a mathematical form, by building a system of differential equations that can capture temporal and spatial behaviors of molecules within a complex network. A main challenge in this approach is that the mechanistic details and kinetic parameters of the molecular interactions are not available for continuous models in large molecular networks. In such scenarios, Boolean modeling has been useful as it does not need detailed kinetic information and still can provide biologically relevant results, as discussed in [10-13], and in several review articles [15-24] that are summarizing many other original research contributions.

Typically, models for literature-curated molecular networks do not adequately match experimental data. This is due to the incompleteness and species heterogeneity of resources, databases, and the literature used to construct the networks. In such networks, for some individual interactions, generally there is more than one publication, and sometimes some studies suggest contradicting results. Consequently, models constructed for molecular networks using only the literature may poorly perform in terms of fitting experimental data. Thus, one needs to develop algorithms to learn and refine the models, so that the learned networks can mimic the experimental observations[5,25-27]. Herein, we propose a method for fitting a network model to data. The method transforms the model into an integer linear programming (ILP) formulation, allowing us to learn a subnetwork of the initial network that exhibits an optimal fit to the experimental data. ILP is basically a mathematical optimization formulation, where the involved variables and parameters take integer values. This makes ILP particularly suitable for Boolean networks where all the variables and parameters take integer (binary) values. As discussed in what follows, the method explicitly incorporates the role of network regulatory feedback mechanisms, not considered[5,26,27] or removed[28,29] in prior studies.

[a] Department of Biochemistry, Vanderbilt University Nashville, TN 37205. mustafa.ozen@vanderbilt.edu

[b] Advanced Technologies for Novel Therapeutics Millburn, NJ 07041. emame@atnt-usa.com

*[c] Department of Electrical and Computer Engineering and Department of Biological Sciences, New Jersey Institute of Technology, Newark, NJ 07102. *e-mail: ali.abdi@njit.edu, phone: +1 973-596-5621*

Modeling and analysis of molecular networks become more challenging if there are positive or negative feedback paths in the network. Due to the feedback mechanisms, network responses may change over time because of some compensatory or regulatory mechanisms[30,31]. Feedbacks can cause delays in propagation of signals to the network outputs, while passing through the feedback paths. Therefore, incorporating the delays caused by the feedbacks, which may result in different network responses over time, is essential when developing network learning algorithms. The goal of this paper is to introduce new network learning ILP formulations for different Boolean models, when the network of interest has some feedback paths.

To systematically handle the feedbacks and fit the model to multi-time point data, here we are inspired by a technique called combinational iterative array model of synchronous sequential digital circuits[32] in which different time point responses of a sequential digital circuit are mapped to the space domain, by expanding the circuit and connecting the expansions via memory units - that model the feedbacks in our molecular network models. In these circuits, a memory unit stores 1-bit data appearing at its input, and therefore introduces a delay in the propagation of the signal to its output. Since feedback loops in a molecular network act like memory units, by causing signal propagation delays from downstream to upstream molecules, they are usually modeled using memory units in Boolean models. This enables us to apply the combinational iterative array modeling approach to the considered network models, that systematically incorporates the effects of feedbacks during the training of the network models against multi-time point data.

Given a network with feedback edges and also T-time point measured data, we start with the subnetwork of the original network that excludes the feedback edges, and call it the early event (EE) network. To incorporate the delays caused by the feedbacks, similar to the combinational iterative array model of synchronous sequential circuits, we copy the EE network T times and connect them sequentially using the feedback edges, where each copy's response represents a specific time point. Afterward, the proposed ILP formulation can be written down for the expanded network and then solved to find the optimal network that provides the best fit to the T-time point data. For simplicity, we present this idea considering T = 2 time point data in our examples, but it can be implemented for other T values as well, as elaborated here.

The rest of this paper is organized as follows: In Section 2, we present two Boolean models, along with their truth tables and examples. In Section 3, we present an ILP formulation for each model, then demonstrate them in Section 4 using some numerical examples, followed by a real biological network example, i.e., the ERBB signaling network, to show how networks with feedbacks can be learned from data. Finally, we conclude the paper with some remarks given in Section 5.

2 Molecular Network Models

Figure 1 illustrates a toy molecular network having 7 nodes and 10 edges. Each edge represents an interaction between the molecules. An arrow edge "→" rep-resents an activatory interaction and a blunt edge "—|" reflects an inhibitory

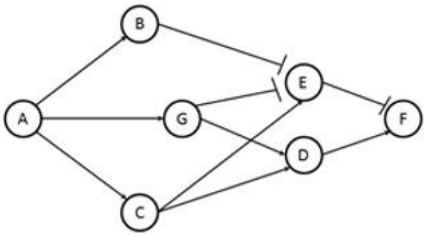

Figure 1. Toy example of a molecular network.

interaction. A node at the beginning of an edge represents an input molecule, whereas a node at the end of an edge stands for a product (output). A set of input nodes and a product node together constitute an interaction set. For instance, in Figure 1, the nodes B, G, C and E together represent an interaction set in which B, G and C are the input molecules and E is the output molecule (product). Overall, it can be said that molecular networks consist of interaction sets, each set comprising one or more inputs and one output. A molecule is defined as active if its abundance or activity level, e.g., phosphorylation level, is above a biologically significant threshold, and inactive otherwise.

To model how the output molecule of each interaction set is controlled by its input molecules, 0 and 1 values and logic operations are used in the Boolean framework[10,13,15,16]. A key advantage of this framework is that it does not need the detailed mechanistic information about the molecular interactions in various interaction sets, i.e., it does not have hundreds of unknown parameters to be estimated, and yet provides certain biologically meaningful results and predictions. In the rest of this section, two such models, Model I and Model II, are presented, and then are used in subsequent sections, for the network modeling and learning.

2.1 Model I: 1 for Increased Activity and 0 for Decreased or Not Changed Activity

In a typical biological interaction set, the activity level of the output, the product of the interaction set, can increase, decrease, or remain unchanged, compared to its basal level and depending on its input molecules. In Model I, *increase* in the activity level of a molecule is represented by a 1, whereas *decrease* or *no change* in the activity level of a molecule is represented by a 0. Assume there exists an interaction set with multiple activators and inhibitors. Model I incorporates the following two rules to specify the output molecule's activity level: Let $x_1,\ldots,x_i,x_{i+1},\ldots,x_n$ be n input molecules and w be the product of an interaction set such that x_i is an activator for $i=1,\ldots,j$ and x_i is an inhibitor for $i=j+1,\ldots,n.$ Then,

(a) If at least one of the activators and none of the inhibitors are 1, i.e., active, then the output is 1. This means if $\exists\, i\in\{1,\ldots,j\}$ such that $x_i=1$ and $x_i=0$ for all $i=j+1,\ldots,n,$ then $w=1.$

(b) If at least one of the inhibitors is 1, then the output is 0, i.e., if $\exists\, i\in j+1,\ldots,n$ such that $x_i=1,$ then $w=0.$

Figure 2 is an illustration of the model and its truth table, based on its rules (a) and (b) given above.

2.2 Model II: 1 for Changed Activity and 0 for Not Changed Activity

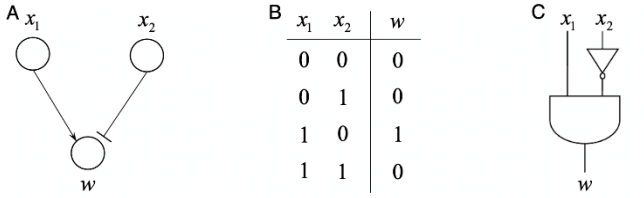

Figure 2. An example for Model I. (A) A two-input one-output interaction set. (B) Truth table of the interaction set based on the model rules. (C) Logic circuit representation of the interaction set using NOT and AND gates.

In Model II, *change* and *no change* in the activity are considered, and are labeled as 1 and 0, respectively. In response to an input signal, we declare a *change* if the activity of a molecule increases or decreases, compared to its basal level. On the other hand, we declare a *no change* if the activity of a molecule does not change with respect to its basal level, when an input signal is applied.

Recall that $x_1,\ldots,x_i,x_{i+1},\ldots,x_n$ are the *n* input molecules and w is the product of an interaction set, such that x_i is an activator for $i=1,\ldots,j$ and x_i is an inhibitor for $i=j+1,\ldots,n$. In Model II, each interaction set in the network follows these two rules:

(a) The output is 1, meaning that there is a change in the output's activity, if at least one of the input molecules is 1, i.e., if $\exists\, i\in\{1,\ldots,n\}$ such that $x_i=1$, then $w=1$.

(b) The output is 0, meaning that there is no change in the output's activity, if all the inputs are 0, i.e., $w=0$ if $x_i=0$ for all $i=1,\ldots,n$.

The model and its truth table are exemplified in Figure 3, using its two rules given above. Overall, this model incorporates and reflects any changes in the input of an interaction set as a change at its product.

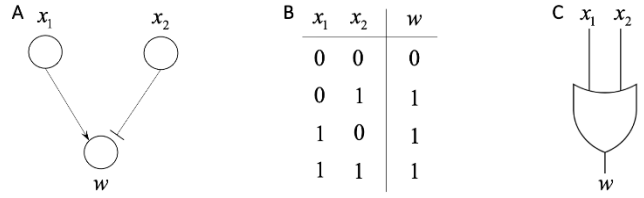

Figure 3. An example for Model II. (A) A two-input one-output interaction set. (B) Truth table of the interaction set based on the model rules. (C) Logic circuit representation of the interaction set using an OR gate.

3 Learning Molecular Network Models

As discussed in Section 1, model predictions of literature-based networks may not agree with experimental data. In order to fit the network to the experimental data, some of the edges (interactions) in the network may need to be removed (spurious interactions), or some new edges may need to be added (missed interactions), so that the resulting network can reflect actual collective behaviors of the molecules, i.e., models that fit the experimental data. Herein, we focus on removing edges and finding a subnetwork of the initial network, since adding new edges requires having access to additional original publications or performing many experiments that are costly and time consuming to acquire. One way to do this is to conduct an optimization to remove edges one by one and check the number of mismatches between model predictions and experimental data. However, for large networks, this does not help as removing one edge at a time most often does not change model predictions. For this reason, we convert this problem into an ILP problem so that multiple edges can be removed systematically, and finally a subnetwork of the initial network can be found as the optimal solution that fits the data. A similar approach is studied in [5,26,27] on the networks that do not have feedbacks and in [28,29], where they removed positive feedback loops in the solutions. In this paper, however, we present a method in Section 4 that incorporates feedback interactions in network learning, after presenting our ILP formulations for Models I and II in this section.

Our goal is to minimize the number of mismatches between model responses and the experimental data. The experimental data set is typically obtained by treating cells with selective agonist of the input molecules and then measuring the activity, i.e., the protein or phosphoprotein levels, of some of the intermediate and output molecules by running western blot analysis.

Let n_E be the number of experiments and each experiment be indexed by the superscript $k=1,\ldots n_E$. In the network, assume there exist n_R interaction sets that are indexed by the subscript $i=1,\ldots,n_R$. Each interaction set i has the corresponding index set $I_i=A_i\cup H_i$ for its input molecules, in which A_i and H_i are the index sets of activators and inhibitors, respectively. Lastly, let M be the index set of molecules for which we have experimental data. Then, in the general form of the proposed ILP formulation, we define all the other variables as shown below.

- $x_\ell^{k,m}$: experimental value of the ℓ^{th} node in the k^{th} experiment, for all $\ell\in M$. Here, the superscript *m* indicates that the ℓ^{th} node has experimental measurement in the k^{th} experiment.

Then, in each interaction set *i*, we have:

- x_j^k: model's predicted value of the j^{th} input node of the i^{th} interaction set in the k^{th} experiment, for all $j\in I_i$. To simplify the notation, the interaction set label *i* is not included in the x_j^k variable.
- y_j: decision variable, for all $j\in I_i$. $y_j=1$ means that j^{th} edge in the interaction set i should be preserved in the network whereas $y_j=0$ means that j^{th} edge in the interaction set i should be removed from the network.
- z_j^k: transition variable, for all $j\in I_i$. It transits the input value x_j^k associated with the j^{th} edge to the output of interaction set i, i.e., $z_j^k=x_j^k$, if $y_j=1$. Otherwise, $z_j^k=0$.
- w_i^k: output (product) of the interaction set i in the experiment k.

The objective function to be minimized in the learning phase is the summation of the mismatches - absolute differences - between the experimental data and model's predictions over all experiments. Thereby, the objective function is:

$$\sum_{k=1}^{n_E}\sum_{\ell\in M}\left|x_\ell^k-x_\ell^{k,m}\right|. \tag{1}$$

For binary x_ℓ^k and $x_\ell^{k,m}$ values, (1) can be linearized as:

$$\sum_{k=1}^{n_E}\sum_{\ell\in M}x_\ell^{k,m}+\left(1-2x_\ell^{k,m}\right)x_\ell^k. \tag{2}$$

3.1 ILP Formulation for Model I

Using all the definitions given above, the constrained ILP formulation for Model I introduced in Section 2.1 can be written as follows:

$$\min_{\boldsymbol{y}} \sum_{k=1}^{n_E} \sum_{\ell \in M} x_\ell^{k,m} + \left(1 - 2x_\ell^{k,m}\right) x_\ell^k$$

Subject to $\forall i = 1, \ldots, n_R, \ \forall k = 1, \ldots, n_E$

(i) $z_j^k \geq y_j + x_j^k - 1, \quad \forall j \in I_i$

(ii) $z_j^k \leq x_j^k, \ \forall j \in I_i$

(iii) $z_j^k \leq y_j, \ \forall j \in I_i$

(iv) $w_i^k \leq 1 - \sum_{j \in H_i} \frac{z_j^k}{|H_i| + 1},$

(v) $w_i^k \geq \sum_{j \in A_i} \frac{z_j^k}{|A_i| + 1} - \sum_{j \in H_i} z_j^k,$

(vi) $w_i^k \leq \sum_{j \in I_i} z_j^k,$ (3)

(vii) $0 \leq x_j^k, y_j, z_j^k, w_i^k \leq 1, \quad x_j^k, y_j, z_j^k, w_i^k \in \mathbb{Z}.$

In (3), $\boldsymbol{y} = [y_j]$ is the vector of indices of edges in the network, and the constraints (i), (ii), and (iii) are introduced for edge removal. More precisely, these three constraints assure that if the j^{th} interaction in interaction set i is removed, i.e., $y_j = 0$, then the transition variable is 0, i.e., $z_j^k = 0$, so that the input molecule associated with the j^{th} interaction does not affect the value of the interaction set output w_i^k. If the j^{th} interaction needs to stay, i.e., $y_j = 1$, then these constraints guarantee for the transition variable z_j^k that $z_j^k = x_j^k$. The constraints (iv), (v), and (vi) implement the two rules of Model I. To elaborate, depending on the constraints (i), (ii), and (iii), the transition variable z_j^k becomes either 0 or x_j^k. Then, if none of the inhibitors and at least one of the activators is 1, the constraints (iv) and (v) guarantee that the interaction set output $w_i^k = 1$ (Section 2.1, rule (a)). On the other hand, if at least one of the inhibitors is 1, i.e., $\exists j \in H_i$ such that $z_j^k = 1$, then, the constraints (iv) and (v) make sure that the interaction set output $w_i^k = 0$ (Section 2.1, rule (b)). The constraint (vi) is necessary to guarantee that the interaction set output $w_i^k = 0$, if all the incoming edges are removed or the input values of the remaining edges are 0. Lastly, the constraint (vii) is needed to guarantee that all variables are integers, and they are either 0 or 1.

3.2 ILP Formulation for Model II

A similar formulation can be developed for learning Model II introduced in Section 2.2. This can be done by discarding the constraint (iv) of (3) and replacing the constraint (v) of (3) by $w_i^k \geq z_j^k$ for all $j \in I_i$, which result in the following constrained ILP formulation for Model II:

$$\min_{\boldsymbol{y}} \sum_{k=1}^{n_E} \sum_{\ell \in M} x_\ell^{k,m} + \left(1 - 2x_\ell^{k,m}\right) x_\ell^k \qquad (4)$$

Subject to $\forall i = 1, \ldots, n_R, \ \forall k = 1, \ldots, n_E$

(i) $z_j^k \geq y_j + x_j^k - 1, \quad \forall j \in I_i$

(ii) $z_j^k \leq x_j^k, \ \forall j \in I_i$

(iii) $z_j^k \leq y_j, \ \forall j \in I_i$

(iv) $w_i^k \geq z_j^k \ \ \forall j \in I_i$

(v) $w_i^k \leq \sum_{j \in I_i} z_j^k,$

(vi) $0 \leq x_j^k, y_j, z_j^k, w_i^k \leq 1, \quad x_j^k, y_j, z_j^k, w_i^k \in \mathbb{Z}.$

Similarly to (3), $\boldsymbol{y} = [y_j]$ in (4) is the vector of indices of edges in the network, and the constraints (i), (ii), and (iii) in (4) are introduced for edge removal, as elaborated in the previous subsection. The constraints (iv) and (v) in (4) implement the two rules of Model II. More precisely, if at least one of the input values whose associated edge is not removed is 1, then we have $w_i^k \geq 1$ from (iv) and (v), which guarantees $w_i^k = 1$, because of the constraint (vi). Otherwise, $w_i^k = 0$. Finally, the constraint (vi) is needed to guarantee that all variables are integers, and they are either 0 or 1.

4 Numerical Results

The ILP formulations in (3) and (4) search for a vector $\boldsymbol{y} = [y_j]$, the vector of indices of edges in the network, to minimize the number of mismatches between predictions and the data. To be more precise, a network can be represented by the vector y that is a vector of 1s whose length is equal to the total number of interactions in the network. Thus, a subnetwork of the initial network can be represented by the same length y vector where some of the 1s there are changed to 0 (if the j^{th} entry of y is 0, then this means that the j^{th} interaction is not present in the subnetwork). As a result, by solving the ILP formulations, one can find the best y vectors, i.e., the subnetworks, that have the optimal fit to the data.

4.1 An Exemplary Network

Now we apply the ILP formulation in (3) to the exemplary network in Figure 4A. The equations - based on the two rules of Model I - for each node can be written as shown in Figure 4B (for Model II, the ILP formulation in (4) has to be used). Because of the presence of feedback interactions, the blue edges in Figure 4A, this network has two early event (EE) and late event (LE) representations, as shown in Figure 5A and 5D. This is because when there are feedbacks in the network, the network response can be different at different time instances, due to the signal propagation delays caused by the feedback paths. Using the equations in Figure 5B and 5E for EE and LE networks, respectively, EE and LE truth tables can be created, as given in Figure 5C and 5F.

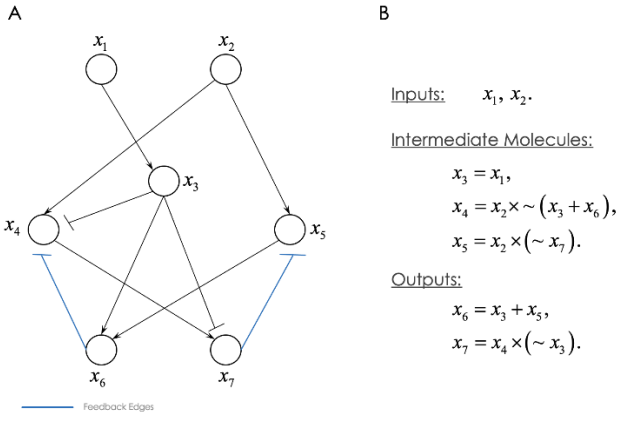

Figure 4. An exemplary network and its equations. (A) The network with two input nodes and two output nodes, where feedforward and feedback edges are shown in black and blue, respectively. (B) The equations for the nodes obtained using Model I.

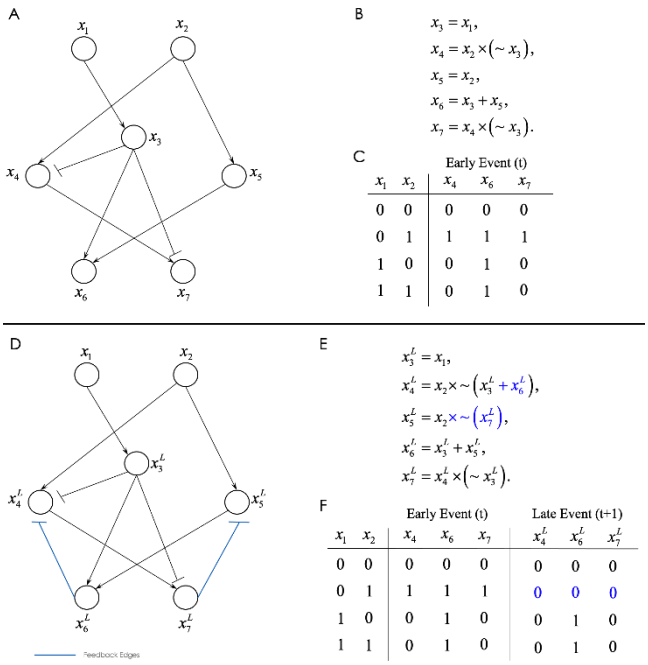

Figure 5. The early event (EE) and the late event (LE) representations of the exemplary network. (A) The EE network. (B) Equations of the nodes in the EE network. (C) Truth table of the EE network. (D) The LE network. (E) Equations of the nodes in the LE network. (F) Truth table of the LE network (truth table of the EE network is also in-cluded for convenience).

4.2 Network Learning from Data

Assume that the network in Figure 5D is the ground truth network and Figure 5F is its truth table that is obtained via hypothetical lab experiments, i.e., measuring early and late output activities in response to different input activities. To test the ILP formulation in (3), we alter this network by adding some spurious edges and generate a new network shown in Figure 6A, that has a new truth table given in Figure 6B.

Suppose that the new network with the spurious interactions (Figure 6A) is the initial network constructed from literature, and therefore our initial truth table has some mismatches, the red values in Figure 6B, compared to the experimental truth table in Figure 5F. Our goal is to learn this network from the experimental data of Figure 5F, by developing and solving the ILP formulation in (3). In other words, the goal is to find a subnetwork of the initial curated network that exhibits the optimal fit to the data. The expectation after solving the ILP for the network in Figure 6A is that the network in Figure 5D must be obtained as the optimal solution.

4.3 Incorporating Feedback Paths in Network Learning from Data

In the learning phase, care should be taken while considering the feedback paths. Since the network may present different responses at different time instances - which is the case as seen in Figure 5F - implementing the constraints in (3) is not trivial for the nodes receiving incoming feedback inputs. In fact, it is very challenging to mathematically formulate such nodes in one step because such nodes need to be initialized and then updated when the

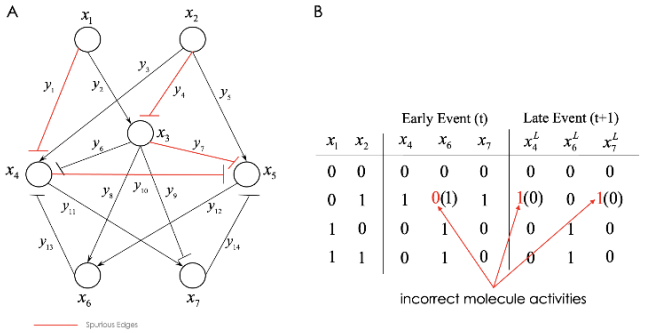

Figure 6. The exemplary network with some spurious edges and its incorrect truth table. (A) Red edges in the network represent spurious interactions that are – unknowingly – included during the network curation from the literature, in addition to the the correct interactions shown by black edges. (B) Red entries in the truth table represent incorrect activity levels, caused by the spurious interactions, whereas the corresponding correct activity levels are given in parentheses.

LE data is considered. To solve this problem, we propose to duplicate the EE network, as shown in Figure 7, and then connect these two identical networks using the feedback edges. Furthermore, we treat the intermediate and output nodes in the duplicate network as new nodes, as shown in Figure 7 using the superscript ^{L} that refers to the late event. For instance, x_4 and x_4^L represent EE and LE activity variables, respectively, where x_4^L receives the feedback input initiated from x_6. Note that if a node in the original network does not receive any feedback input, then its LE variable is equal to its EE variable, e.g., $x_3^L = x_3$. Moreover, the edges in the two identical network copies are labeled by the same decision variable y_j, $j \in I_i$, $i = 1,\ldots,n_R$, so that if $y_j = 0$, then both edges are removed from the two network copies. For instance, the edges $x_1 \dashv x_4$ and $x_1 \dashv x_4^L$ are both labeled by y_1 in Figure 7, so that $y_1 = 0$ means that both edges are removed.

The ILP formulation for Figure 7 is implemented using OPL (Optimization Programming Language), a high-level programming language, and is solved using the IBM ILOG

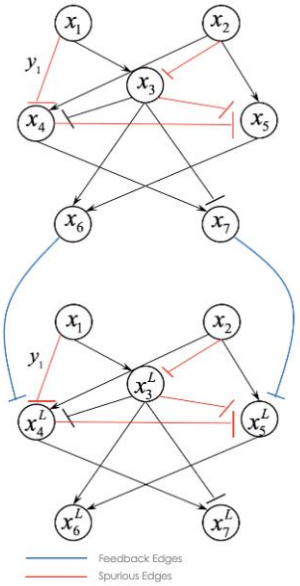

Figure 7. The duplicated network (bottom) of the original network (top), to incorporate the feedbacks during the network learning via ILP.

CPLEX optimization studio[33], a commercial software that solves optimization problems. CPLEX found twelve optimal solutions, i.e., twelve y vectors, such that the numerical values of their learning objective function - computed using (2) - are equal to 0, which means 100% fitness. One of these optimal solutions is $y = [0\ 1\ 1\ 0\ 1\ 1\ 0\ 1\ 1\ 0\ 1\ 1\ 1]$, which represents the network in Figure 5D. This demonstrates the ability of the proposed approach in finding a subnetwork with the best fit to the data, while preserving the rules of the model of interest. How to handle other solutions is discussed in the next subsection.

One can similarly implement and solve the ILP formulation in (4) for network learning using the Model II given in Section 2.2, which is omitted here to avoid repetition.

4.4 Variations and Modifications

The proposed learning approach via ILP formulation and the introduced strategy to handle feedbacks are applicable to very large networks and capable of finding the exact optimal subnetworks with the best fitness percentage to the data. However, solutions of the ILP formulations may not be unique and multiple solutions may be obtained. On the other hand, we have observed that the results are usually highly correlated, indicating that the solutions are very similar. Therefore, one can examine the solutions and choose the one that meets a specific criterion, for example, being the closest to the published information in the literature.

It is also foreseeable that the resulting subnetwork solutions may be missing several interactions that existed in the initial network. To control the number of removed edges, one can add a penalty term to the learning objective function, for example, the one in (2), as follows:

$$\sum_{k=1}^{n_E} \sum_{\ell \in M} x_\ell^{k,m} + \left(1 - 2x_\ell^{k,m}\right) x_\ell^k + \beta \sum_i^{n_R} \sum_j^{I_i} \left(1 - y_j\right) \qquad (5)$$

where $\beta > 0$ is a tunable penalty parameter that penalizes the objective function for each removed edge. Higher values of β may result in worse fitness to the data, while keeping more edges in the final learned subnetwork. Therefore, there can be a tradeoff between the number of removed edges and the data fitness percentage that we should keep in mind.

In addition, some of the interactions that are removed to obtain the optimal solution, may be well-known interactions that are experimentally confirmed by several groups of researchers. To prevent this from happening, one can add some additional constraints to force well-known edges of interest to remain in the network.

4.5 A Molecular Network Example: Learning the ERBB Signaling Network via ILP

ERBB network is a signaling network that regulates the transmembrane tyrosine kinase ERBB[34], which is a therapeutic target in breast cancer. The ERBB network has one input, one output, 18 intermediate molecules, and 51 interactions, as presented in Figure 8A. The input molecule is the Epidermal Growth Factor (EGF), whereas the output molecule is the Retinoblastoma Protein (pRB). Moreover, it contains five feedback loops. This network has been studied in the context of breast cancer and understanding the mechanisms of actions of few drug molecules. Herein, we employ this network together with the provided experimental data in [34], to learn the network from the data via the proposed framework, using the ILP formulation given in (3). Further details on the network and the experimental data are provided in [34].

The network of [34] provides 76.5% data fitness. To see if this result can be improved, we learn the ERBB network using the ILP formulation introduced in (3). Since the network contains feedback interactions, we first duplicate the EE network and connect the two identical networks via the feedback edges (Figure 8B), as previously exemplified in Figure 7. Then, we implement the ILP formulation in (3) with an extra constraint that ensures at least one incoming edge for each node is preserved in the learned network, so that none of the nodes are removed from the network. This is mathematically done by adding new inequalities. For instance, suppose y_1 and y_2 label two incoming edges to a node. Then, the constraint $y_1 + y_2 \geq 1$ makes sure that at least one of the y_1 and y_2 will be 1, meaning that at least one of the edges labeled by y_1 and y_2 will remain in the learned network.

Upon implementing the resulting ILP using OPL and solving it via CPLEX, we achieve 82.4% data prediction accuracy, with Figure 8C depicting a learned network. This network improves the data fitness accuracy by removing only two edges, i.e., the two edges previously connecting ERBB1 to AKT1 and CDK6 to pRB (gray dashed edges in Figure 8C), while preserving the majority of the initial network. This finding and approach help biologists by generating hypotheses for them to further investigate the interactions between ERBB1 to AKT1 and CDK6 to pRB, to see if the interactions can be replicated in different cell types. If the interactions are reproducible, it is possible that other intermediate molecules are involved in between ERBB1 and AKT1, or CDK6 and pRB that were not reported, which can be the subject of additional research in molecular biology.

If the above constraint is removed and learning is performed with more relaxed ILP, the data prediction accuracy further increases to 88.2%. However, about 15 edges are removed after learning, which exemplifies the tradeoff mentioned in Section 4.4. Overall, network learning from experimental data is essential for conducting research and analysis on molecular networks that faithfully represent the data, so that the research findings will be biologically plausible and relevant.

5 Conclusion

Transforming molecular networks into mathematically analyzable yet experimentally verifiable models is a major challenge in systems biology. The network models usually do not agree with the experimental measurements initially, specifically for literature-curated networks. The disagreement between model predictions and the experimental data can be due to the incompleteness of information resources, databases, and the literature used to construct the networks. Developing tools to learn the network models from empirical data is of high importance, since it improves the reliability of the models, and consequently increases the likelihood of confirming computational predictions in laboratory experiments. In this paper, we have presented two network models (Section 2) and have shown how the networks can be learned and calibrated

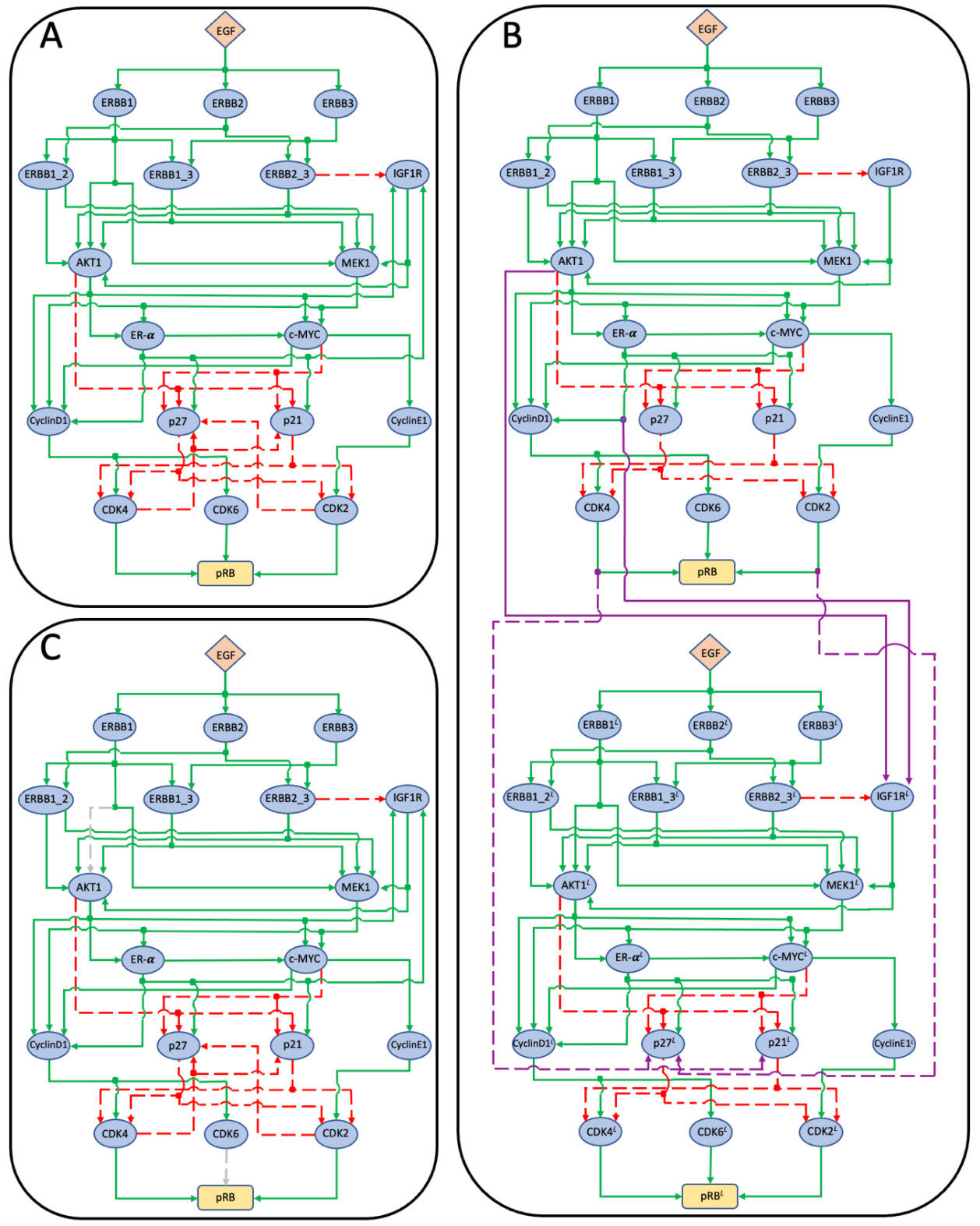

Figure 8. Learning the ERBB signaling network of [34] using its experimental data and via the proposed ILP formulation in (3). (A) The redrawn ERBB signaling network of [34]. The green solid edges represent activatory interactions and the red dashed edges represent inhibitory interactions. Each green or red square marker pinpoints the beginning of at least one branching-out edge. The input and output nodes are EGF and pRB, respectively. (B) The duplicated network (bottom) of the original network (top), to incorporate the feedbacks during the proposed network learning algorithm via ILP. These two networks are connected by the feedback edges represented by the purple edges. (C) An example of a learned network having an improved 82.4% data prediction accuracy. The improvement is achieved after the two edges previously connecting ERBB1 to AKT1 and CDK6 to pRB in the network of panel (A) are removed upon ILP network learning (for easy viewing, the two removed edges are shown using gray dashed edges).

using experimental data and via integer linear programming (Section 3), by minimizing the number of mismatches between the model predictions and the experimental data (Section 4).

Due to the feedback paths, modeling and analysis of molecular networks become more challenging. Because of the signal propagation delays introduced by the feedback mechanisms, network responses may change over time. Thus, the complex compensatory and regulatory mechanisms of feedbacks should be considered, while learning network models from data. Here, we have presented a method that is able to systematically handle feedbacks in networks – key components of nearly all molecular networks – which is lacking in previous published works to the best of our knowledge, inspired by a technique called combinational iterative array model of synchronous sequential circuits. As tested on an exemplary network (Figure 6 and Figure 7), the ILP formulation can effectively find the correct subnetwork from an initial network that has some spurious interactions. Furthermore, once it is applied to the ERBB signaling network (Figure 8), studied in the context of breast cancer, the data prediction accuracy is increased after learning. Different aspects of the proposed algorithm and relevant modifications are also discussed (Section 4.4).

Overall, the proposed network learning approach has promising potentials to reduce the gap between the literature-curated networks and the experimental data of the network, while incorporating complex interactions and biological mechanisms within the network such as feedbacks. This study is particularly important if computational analysis is going to be performed on the molecular network models associated with complex disorders, when some unknown molecular mechanisms are contributing to the development of the pathology. By providing more reliable networks with more accurate data prediction capabilities, the proposed approach of fitting the disease-associated molecular networks to the experimental data can assist in building better systems biology models for understanding the pathology, and eventually finding the best molecules in the network to target with novel therapeutic.

Acknowledgements

Authors declare no conflict of interests exist.

References

[1] I. Shmulevich, E. R. Dougherty, W. Zhang, From Boolean to Probabilistic Boolean Networks as Models of Genetic Regulatory Networks, *Proc.IEEE*, **2002,** *90*(11), 1778–1792.

[2] A. Saadatpour, R. Albert, A Comparative Study of Qualitative and Quantitative Dynamic Models of Biological Regulatory Networks, *EPJ Nonlinear. Biomed. Phys.*, **2016**, *4*(5).

[3] J. Saez-Rodriguez, A. MacNamara, S. Cook, Modeling Signaling Networks to Advance New Cancer Therapies, *Annu. Rev. Biomed. Eng.,* **2015**, *17*, 143–163.

[4] R. S. Wang, R. Albert, Elementary Signaling Modes Predict the Essentiality of Signal Transduction Network Components, *BMC Syst. Biol.*, **2011**, *5*, article 44.

[5] A. Mitsos, I. N. Melas, P. Siminelakis, A. D. Chairakaki, J. Saez-Rodriguez, L. G. Alexopoulos, Identifying Drug Effects via Pathway Alterations using an Integer Linear Programming Optimization Formulation on Phosphoproteomic Data, *PLoS Comput Biol*, **2009**, 5(12).

[6] F. Eduati, P. Jaaks, J. Wappler, T. Cramer, C. A. Marten, M. J. Garnett, J. Saez-Rodriguez, Patient-specific Logic Models of Signaling Pathways from Screenings on Cancer Biopsies to Prioritize Personalized Combination Therapies, *Mol. Syst. Biol.*, **2020**, *16*.

[7] T. Helikar, J. Konvalina, J. Heidel, J. A. Rogers. Emergent Decision-making in Biological Signal Transduction Networks, *Proc. Natl. Acad. Sci.*, **2008**, *105*(6), 1913-1918.

[8] M. Ozen, T. Lipniacki, A. Levchenko, E. S. Emamian, A. Abdi, Modeling and Measurement of Signaling Outcomes Affecting Decision Making in Noisy Intracellular Networks Using Machine Learning Methods, *Integr. Biol*., **2020**, *12*(5), 122–138.

[9] B. Offermann, S. Knauer, A. Singh, M. L. Fernandez-Cachon, M. Klose, S. Kowar, H. Busch, M. Boerries, Boolean Modeling Reveals the Necessity of Transcriptional Regulation for Bistability in PC12 Cell Differentiation, *Front. Genet.,* **2016**, *14*.

[10] A. Abdi, M. B. Tahoori, E. S. Emamian, Fault Diagnosis Engineering of Digital Circuits can Identify Vulnerable Molecules in Complex Cellular Pathways, *Sci. Signaling*, **2008**, *1*(42), 48-61.

[11] I. Habibi, E. S. Emamian, A. Abdi, Quantitative Analysis of Intracellular Communication and Signaling Errors in Signaling Networks, *BMC Syst. Biol*., **2014**, *8*(89).

[12] I. Habibi, E. S. Emamian, A. Abdi, Advanced Fault Diagnosis Methods in Molecular Networks, *PLoS One*, **2014**, *9*.

[13] I. Habibi, E. S. Emamian, O. Simeone, A. Abdi, Computation Capacities of a Broad Class of Signaling Networks are Higher Than Their Communication Capacities, *Phys. Biol.*, **2019**, *16*(6).

[14] D. J. Wooten, S. M. Groves, D. R. Tyson, Q. Liu, J. S. Lim, R. Albert, C. F., Lopez, J. Sage, V. Quaranta, Systems-level network modeling of Small Cell Lung Cancer subtypes identifies master regulators and destabilizers, *PLoS Comput. Biol.*, **2019**, *15*(10).

[15] R. S. Wang, A. Saadatpour, R. Albert, Boolean Modeling in Systems Biology: An Overview of Methodology and Applications, *Phys. Biol.*, **2012**, *9*.

[16] A. Saadatpour, R. Albert, Boolean Modeling of Biological Regulatory Networks: A Methodology Tutorial, *Methods*, **2013**, *62*, 3-12.

[17] T. Helikar, N. Kochi, J. Konvalina, J. A. Rogers, Boolean Modeling of Biochemical Networks, *The Open Bioinf. Journal*, **2011**, *5*, 16–25.

[18] M. K. Morris, J. Saez-Rodriguez, P. K. Sorger, D. A. Lauffenburger, Logic-based Models for the Analysis of Cell Signaling Networks, *Biochemistry*, **2010**, *49*, 3216-3224.

[19] A. Saadatpour, R. Albert, Discrete Dynamic Modeling of Signal Transduction Networks, *Methods Mol. Biol*, **2012**, *880*, 255-272.

[20] R. Samaga, S. Klamt, Modeling Approaches for Qualitative and Semi-quantitative Analysis of Cellular Signaling Networks, *Cell Commun. Signaling*, **2013**, *11*(43).

[21] G. Stoll, E. Viara, E. Barillot, L. Calzone, Continuous Time Boolean Modeling for Biological Signaling: Application of Gillespie Algorithm, *BMC Syst. Biol.*, **2012**, *6*(116).

[22] T. Handorf, E. Klipp, Modeling mechanistic biological networks: An Advanced Boolean Approach, *Bioinformatics*, **2012**, *28*, 557-563.

[23] C. Chaouiya, E. Remy, Logical Modelling of Regulatory Networks, Methods and Applications, *Bull. Math. Biol*, **2013**, *75*, 891-895.

[24] C. Chaouiya, A. Naldi, D. Thieffry, Logical Modelling of Gene Regulatory Networks with GINsim, *Methods Mol. Biol*, **2012**, *804*, 463-479.

[25] J. Saez-Rodriguez, L. G. Alexopoulos, J. Epperlein, R. Samaga, D. A. Lauffenburger, S. Klamt, P. K. Sorger, Discrete Logic Modelling as a Means to Link Protein Signalling Networks with Functional Analysis of Mammalian Signal Transduction, *Mol. Sys. Biol*., **2009**, *5*(331).

[26] S. Videla, C. Guziolowski, F. Eduati, S. Thiele, N. Grabe, J. Saez-Rodriguez, A. Siegel, Revisiting the Training of Logic Models of Protein Signaling Networks with ASP, *Comput. Methods in Syst. Biol.*, D. Gilbert D and M. Heiner, eds,

CMSB Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, **2012**, *7605*.
[27] R. Sharan, R. M. Karp, Reconstructing Boolean Models of Signaling, *J. Comput. Biol.*, **2013**, *20*(3), 249-257.
[28] I. N. Melas, T. Sakellaropoulos, F. Iorio, L. G. Alexopoulos, W-Y. Loh, D. A. Lauffenburger, J. Saez-Rodriguez, J. P. F. Bai, Identification of Drug-specific Pathways Based on Gene Expression Data: Application to Drug Induced Lung Injury, *Integr. Biol.,* **2015**, *7*, 904-920.
[29] E. Gjerga, A. Dugourd, L. Tobalina, A. Sousa, J. Saez-Rodriguez, PHONEMeS: Efficient Modeling of Signaling Networks Derived from Large-Scale Mass Spectrometry Data, *J. Proteome Res.,* **2021**, *20*, 2138-2144.
[30] E. Azpeitia, S. Muñoz, D. González-Tokman, M. E. Martinez-Sanchez, N. Weinstein, A. Naldi, E. R. Alvarez-Buylla, D. A. Rosenblueth, L. Mendoza, The Combination of the Functionalities of Feedback Circuits is Determinant for the Attractors' Number and Size in Pathway-like Boolean Networks, *Sci. Rep.*, **2017**, *7*.
[31] R. Somogyi, L. D. Greller, The Dynamics of Molecular Networks: Applications to Therapeutic Discovery, *Drug Discovery Today*, **2001**, *6*(24), 1267-1277.
[32] M. Abramovici, M.A. Breuer and A.D. Friedman, Digital Systems Testing and Testable Design, Wiley-IEEE Press, New York, 1994.
[33] IBM. (n.d.), IBM ILOG CPLEX Optimization Studio, https://www.ibm.com/products/ilog-cplex-optimization-studio, Retrieved April 18, 2022.
[34] O. Sahin, H. Fröhlich, C. Löbke et al., Modeling ERBB Receptor-regulated G1/S Transition to Find Novel Targets for de novo Trastuzumab Resistance, *BMC Syst. Biol.*, **2009**, *3*(1).