SIAM J. DISCRETE MATH.
Vol. 20, No. 3, pp. 727–740

RECONSTRUCTING CHAIN FUNCTIONS IN GENETIC NETWORKS*

IRIT GAT-VIKS†, RICHARD M. KARP‡, RON SHAMIR†, AND RODED SHARAN†

Abstract. The following problems arise in the analysis of biological networks: We have a boolean function of n variables, each of which has some default value. An *experiment* fixes the values of any subset of the variables, the remaining variables assume their default values, and the function value is the result of the experiment. How many experiments are needed to determine (reconstruct) the function? How many experiments that involve fixing at most q values are needed? What are the answers to these questions when an unknown subset of the variables are actually involved in the function? In the biological context, the variables are genes and the values are gene expression intensities. An experiment measures the gene levels under conditions that perturb the values of a subset of the genes. The goal is to reconstruct the particular logic (regulation function) by which a subset of the genes together regulate one target gene, using few experiments that involve minor perturbations. We study these questions under the assumption that all functions belong to a biologically motivated set of so-called chain functions. We give optimal reconstruction schemes for several scenarios and show their application in reconstructing the regulation of galactose utilization in yeast.

Key words. network reconstruction, experimental design

AMS subject classifications. 90B10, 62K99, 06E30

DOI. 10.1137/S089548010444376X

1. Introduction.

In this paper we study the problem of function reconstruction. We have a set of N boolean variables. Each variable has a default value, and an *experiment* can change (fix to 0 or 1) its value. The *order* of an experiment is the number of variables fixed during the experiment. The value of one variable of interest (the output) is determined by a boolean function of n other variables. The *output* of an experiment is the value of the function, where all fixed variables attain their respective values and the rest attain their default values. The problem of *function reconstruction* is to determine this function using a minimum number of experiments of the smallest possible order.

The motivation to studying the problem arises in molecular biology: The regulation of biological entities is key to cellular function. The genes are expressed (transcribed) into mRNAs, which are translated into proteins. The regulatory factors which control (regulate) gene expression are themselves protein products of other genes. The result is a complex network of regulatory relations among genes. A *genetic network* consists of a set of variables that correspond to *genes*, attaining real values, called *states*. The state of a gene indicates the discretized expression level of the gene. A gene may be *regulated* by several other genes, implying that its state

*Received by the editors May 16, 2004; accepted for publication (in revised form) January 24, 2006; published electronically October 4, 2006. The work of the first author was supported by a Colton fellowship. The second and third authors were supported by a grant from the US-Israel Binational Science Foundation (BSF). The fourth author was supported by a Fulbright grant and by NSF ITR grant CCR-0121555. A preliminary version of this paper appeared in *Proceedings of the Ninth Pacific Symposium on Biocomputing* [10].

http://www.siam.org/journals/sidma/20-3/44376.html

†School of Computer Science, Tel-Aviv University, Tel-Aviv 69978, Israel (iritg@tau.ac.il, rshamir@tau.ac.il, roded@tau.ac.il).

‡International Computer Science Institute, 1947 Center St., Berkeley, CA 94704 (karp@icsi.berkeley.edu).

is a function of the states of its regulating genes, or its *regulators*. An experiment involves *perturbations* such as knocking out certain genes (fixing their states to some low value) or overexpressing them (fixing their states to some high value) and measuring the expression levels of all other genes. The measurement of gene expression levels is facilitated by high throughput technologies, such as DNA microarrays (e.g., [6]). The order of an experiment is the number of genes that are perturbed. In order to reconstruct the regulatory relations among genes, we need to infer the set of genes that cooperate in the regulation of a given gene and the particular logical function by which this regulation is determined. This paper studies the number and order of experiments that are needed in order to infer the regulatory function that governs a specific gene.

A key obstacle in the inference of regulation relations is the large number of possible solutions and, consequently, the unrealistically large amount of data needed to identify the right one. A common and simple model for genetic networks is the *boolean* model, in which the state of a gene is 0 (off) or 1 (on). The boolean assumption is a drastic simplification of real biology, yet it captures important features of biological systems and was frequently used in previous studies [16].

There is a large body of previous work on learning boolean functions from a random sample of their output values (see [3] for a review). Those studies focus on devising efficient probably approximately correct (PAC) learning algorithms for subclasses of boolean functions using a polynomial-size sample. Another body of work is devoted to exact learning of certain classes of boolean functions using a polynomial number of queries (see, e.g., [4] and references thereof). For the specific problem of exact boolean function reconstruction in a genetic network, Akutsu et al. [1] have shown that the number of experiments (or queries) that are needed for reconstructing a function of N genes is prohibitive: The lower and upper bounds on the number of experiments of order $N-1$ that are needed are $\Omega(2^{N-1})$ and $O(N \cdot 2^{N-1})$, respectively. When the function involves only d regulators, the number of required experiments of order d is still $\Omega(N^d)$ and $O(N^{2d})$, respectively [1].

The inherent complexity of this problem led researchers to seek ways around this problem. Ideker, Thorson, and Karp [16] studied how to dynamically design experiments so as to maximize the amount of information extracted. Friedman et al. [8] used Bayesian networks to reveal parts of the genetic network that are strongly supported by the data. Tanay and Shamir [24] suggested a method of expanding a known network core using expression data. Several studies used prior knowledge about the network structure, or restrictive models of the structure, in order to identify relevant processes in gene expression data [12, 15, 23, 22].

Recently, a biologically motivated, boolean model of regulation relations based on *chain functions* was suggested in order to cope with the problem of function reconstruction in biological context [9]. In a chain function, the state of the regulated gene depends on the influence of its direct regulator, whose activity may in turn depend on the influence of another regulator, and so on, in a chain of dependencies (we defer formal definitions to the next section). The class of chain functions has several important advantages [9]: These functions reflect common biological regulation behavior, so many real biological regulatory relations can be elucidated using them (examples include the SOS response mechanism in *E. coli* [21] and galactose utilization in yeast [18]). Moreover, by restricting consideration to chain functions, the number of candidate functions drops from double exponential to single exponential only.

In this paper we study several computational problems arising when wishing to

reconstruct chain functions using a minimum number of experiments of the smallest possible order. We address both the question of finding the set of regulators of a chain function, which is typically much smaller than the entire set of genes, and the question of reconstructing the function given its regulators. We give optimal reconstruction schemes for several scenarios and show their application on real data. Our analysis focuses on the theoretical complexity of reconstructing regulation relations (number and order of experiments), assuming that experiments provide accurate results and that the target function can be studied in isolation from the rest of the genetic network.

The paper is organized as follows: Section 2 contains basic definitions related to chain functions. In section 3 we give worst-case and average-case analyses of the number of experiments needed in order to reconstruct a chain function. Both low-order and high-order experimental settings are considered. In section 4 we study the reconstruction of composite regulation functions that combine several chains. Finally, in section 5 we describe a biological application of our analysis to reconstruct the regulation mechanism of galactose utilization in yeast.

2. Chain functions. Chain functions were introduced by Gat-Viks and Shamir [9]. In the following we define these functions and describe their main properties. Our presentation differs from the original one to allow succinct description of the reconstruction schemes in later sections.

Variables, regulators and states. Let U denote the set of all variables in a network, where $|U| = N + 1$. These variables correspond to genes, mRNAs, proteins, or metabolites. Each variable may attain one of two *states*: 1 or 0. The state of gene g, denoted by $state(g)$, indicates the discretized expression level of the gene. A variable normally attains its *wild-type* state, but perturbations such as gene knockouts may change its state. We say that a variable $g_0 \in U$ is *regulated by* a set $S = \{g_1, \ldots, g_n\} \subset U$ if $state(g_0) = f^{g_0}(state(g_n), \ldots, state(g_1))$ and S is a minimal set with that property. In that case we say that S is the *regulator set* of g_0, and g_0 is called the *regulatee*. Associated with each regulator g_i is a binary constant y_i which dictates the *control* property of g_i. If $y_i = 0$ then g_i is an *activator*; otherwise g_i is a *repressor*. This is an intrinsic property of the regulator and is not subject to change. The *control pattern* of f^{g_0} is the binary vector $(y_n, \ldots, y_1)$.

Given a certain order $g_n, \ldots, g_1$ of the regulators, we call g_i a *predecessor* of g_j for $i > j$ and a *successor* of g_k for $i < k$. We also say that g_i is to the *left* of g_j and to the *right* of g_k. Each regulator transmits a signal to its immediate successor, and this chain of events enables a signal to propagate from g_n to g_0 in a manner defined by a chain function (see Figure 1, top part).

Chain function definition. The chain function model assumes that the functional relations are deterministic. The chain function f^{g_0} on the regulators $g_n, \ldots, g_1$ determines the state of the regulatee g_0.

The function f^{g_0} can be defined using two n-long boolean vectors attributing *activity* and *influence* to each g_i. Let $a(g_i)$ denote the activity of g_i, and let $infl(g_i)$ denote the influence signal from g_i to g_{i-1}. The definitions of activity and influence on the other regulators are recursive: The influence on g_n is always 1. g_i is active ($a(g_i) = 1$) iff it exists ($state(g_i) = 1$) and it receives a positive influence from its predecessor ($infl(g_{i+1}) = 1$). The influence $infl(g_i)$ transmitted from g_i to g_{i-1} is a xor ($\oplus$) of $a(g_i)$ and y_i: $infl(g_i)$ is 1 if g_i is an activator and is itself activated or if g_i is a repressor and is not activated (so that it fails to repress g_{i-1}). Formally,

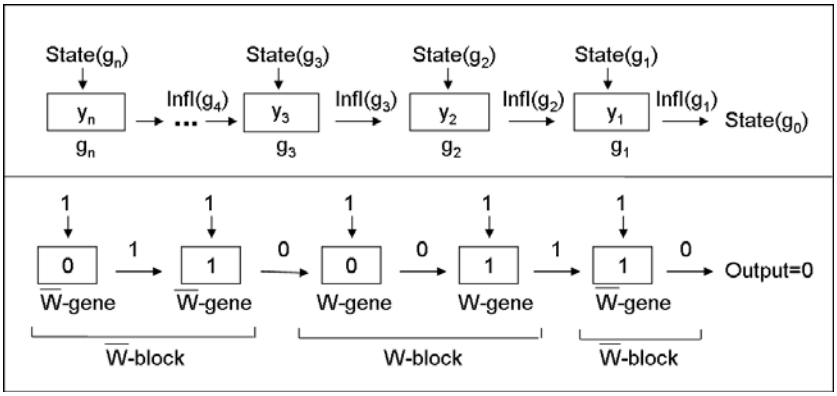


FIG. 1. *The chain function model. Top: A chain function model. Bottom: An illustration of a chain function with five regulators.* g_1, g_2, g_4 *are repressors, and* g_3, g_5 *are activators. The state of all regulators is* 1. *Influences are indicated on the horizontal arrows. Regulator types and blocks are indicated below.*

$$a(g_i) \;=\; infl(g_{i+1}) \;\wedge\; state(g_i), \tag{1}$$

$$infl(g_i) \;=\; y_i \oplus a(g_i). \tag{2}$$

Finally, the state of the regulatee g_0 is simply the influence of g_1. We define the *output* of f^{g_0} to be $state(g_0)$.

A chain function is uniquely determined by its set of regulators, their order, and the control pattern. For example, if g_0 is regulated by (g_3, g_2, g_1) via a chain function with control pattern 010, then $f(1,1,1) = 0$ and $f(0,1,1) = 1$.

3. Reconstruction of chain functions. In this section we study the question of uniquely determining the chain function which operates on a known regulatee, using a minimum number of experiments. We assume throughout that all variable states in wild type are known. We further assume that all regulator states in wild type are 1, except possibly g_n. The latter assumption is motivated by the observation that in many biological examples, all regulators are expressed in wild type, and the state of the regulatee is determined by the presence or absence of a metabolite g_n. (Examples include the Trp, lac, and araBAD operons in *E. coli* [21], the regulation of galactose utilization [18] in yeast, and human MAPK cascades [17]).

An *experiment* is defined by a set of variables that are externally perturbed (knocked-out or overexpressed). The states of the perturbed variables are thus fixed, and the states of all nonperturbed regulators are assumed to remain at the wild-type values. The state of the regulatee is determined by the chain function. The *order* of an experiment is the number of externally perturbed variables in it.

Our reconstruction algorithms are based on performing various experiments and observing their effect on the state of the regulatee. The algorithms implicitly assume that the regulation function is indeed a chain function and do not explicitly test this property.

We now devise a simple set of equations that characterize the output of a chain function as a function of the control pattern and the states of the regulators, both

in the wild-type state and in states produced by perturbing some regulators. These equations are the foundation of all the subsequent reconstruction schemes:

PROPOSITION 1. *Let f be a chain function on $g_n, \ldots, g_1$. If $state(g_i) = 1$ for $1 \le i < n$, then $state(g_0) = state(g_n) \oplus (\oplus_{i=1}^{n} y_i)$. For any other state vector, if the least index of a state-0 regulator is $j \le n$, then $f^{g_0}(g_n, \ldots, g_1) = \oplus_{i=1}^{j} y_i$.*

Proof. By definition, $a(g_n) = state(g_n)$. For $i < n$, $state(g_i) = 1$ implies that $a(g_i) = a(g_{i+1}) \oplus y_{i+1}$. It follows by induction that $state(g_0) = state(g_n) \oplus (\oplus_{i=1}^{n} y_i)$. Similarly, if $state(g_j) = 0$ and $state(g_i) = 1$ for all $i < j$, it follows by induction that $f^{g_0}(g_n, \ldots, g_1) = \oplus_{i=1}^{j} y_i$. ∎

Under the above assumptions on regulator states, a chain function can be viewed as a series of inversion and identity gates, whose input is the state of g_n. Each identity gate corresponds to an activator, whose output is equal to its input. Each inversion gate corresponds to a repressor, whose output is opposite to its input. The output of the last gate in the chain is the state of the regulatee.

3.1. Types and blocks. A *perturbation* is an experiment that changes the state of a variable to the opposite of its state in wild type. By our assumption on the regulator states in wild type (all regulator states in wild type are 1, except possibly g_n), the perturbation of a regulator in $\{g_{n-1}, \ldots, g_1\}$ is a knockout. For $S \subseteq U$, an *S-perturbation* is an experiment in which the states of all the variables in S are perturbed.

Let w be $state(g_0)$ in wild type. Let $\bar{w}$ be the opposite state. For the reconstruction, we first classify the variables in U into two *types*: W and $\bar{W}$ (see Figure 1, bottom part). A variable is in W ($\bar{W}$) if its perturbation produces output w ($\bar{w}$). Typically, the majority of the genes have type W, since in particular all the genes that are not part of the chain function are such. By Proposition 1 we have $g_n \in \bar{W}$, and $g_{n-1} \in W$ iff $state(g_n) \oplus y_n = 0$. We call a gene that belongs to W ($\bar{W}$) a *W-gene* (*$\bar{W}$-gene*). Similarly, we call a regulator of type W ($\bar{W}$) a *W-regulator* (*$\bar{W}$-regulator*). For a given gene, we call a successor of type W ($\bar{W}$) of that gene a *W-successor* (*$\bar{W}$-successor*).

The type of a gene can be determined by a single perturbation of the gene. Such an experiment will be referred to as a *typing experiment* throughout.

COROLLARY 2. *Given an ordered set of regulators $g_n, \ldots, g_1$, their control pattern can be reconstructed using $n - 1$ typing experiments.*

Proof. Perform typing experiments for $g_1, \ldots, g_{n-1}$ (by definition $g_n \in \bar{W}$). By Proposition 1, for every $1 < i < n$, $y_i = 1$ iff the types of g_i and g_{i-1} differ. Also, $y_n = 1$ iff either $state(g_n) = 0$ and the types of g_n, g_{n-1} are equal, or $state(g_n) = 1$ and the two types differ. Finally, we can use Proposition 1 to deduce y_1. ∎

Any control pattern $(y_n, \ldots, y_1)$ may be separated into *blocks* of consecutive regulators by truncating the control pattern after each 1. The first block (rightmost, ending at g_1) has two possible forms: $0 \ldots 0$ or $0 \ldots 01$. All other blocks are of the form $0 \ldots 01$, so the right boundary of a block corresponds to a regulator g_j with $y_j = 1$, and any other regulator g_i in the block has $y_i = 0$.

LEMMA 3. *Each block contains regulators of a single type, and two adjacent blocks contain regulators of opposite types.*

The proof follows from the fact that the type of g_i, $i < n$ differs from the type of g_{i-1} iff $y_i = 1$. Thus, we can refer to a block as either a *W-block* or a *$\bar{W}$-block*, and the two types of blocks alternate. For convenience, we shall refer to g_n as forming a $\bar{W}$-block of its own.

3.2. Reconstructing the regulator set and the function. Consider a chain function with control pattern $(y_n, \ldots, y_1)$ and let $g_j, \ldots, g_i$ be a block. Then $infl(g_i) = [infl(g_{j+1}) \wedge (\bigwedge_{h=i}^{j} state(g_h))] \oplus y_i$. Thus, the effect of the block on the function is determined by the boolean variable $infl(g_{j+1})$, by the control pattern, and by the conjunction of the states of its regulators. Since this conjunction is independent of the order of occurrence of these genes, no experiment based on perturbing the states of the genes can determine the order of the genes within the block. In view of this limitation, we shall aim to find the equivalence class of chain functions as detectable by perturbation experiments, i.e., our goal is to reconstruct the control pattern, the set of genes within each block (but not their order), and the ordering of the blocks. Correspondingly, in the following we will use the term *successor* of a gene to denote a regulator that succeeds that gene in the chain and is not a member of its block. For convenience, we shall refer to gene (in fact, W-genes) that are not regulators of g_0 as predecessors of g_n.

The above discussion implies that once we have typed each gene, it remains to determine, for each pair consisting of a W-gene and a $\bar{W}$-gene, which one precedes the other in the chain. Let k_W and $k_{\bar{W}}$ denote the number of regulators of type W and $\bar{W}$, respectively. Note that $k_W + k_{\bar{W}} = n \leq N$, and in fact, typically, $n \ll N$ as $k_W \ll |W|$.

Suppose we perform a $\{i, k\}$-perturbation with $g_i \in W$ and $g_k \in \bar{W}$. If the result is w, then g_k precedes g_i. Otherwise, g_i precedes g_k. A 2-order experiment for determining the relative order of a W-gene and a $\bar{W}$-gene will be called a *comparison* throughout.

PROPOSITION 4. *Given the set of regulators of a chain function and their types, $k_W k_{\bar{W}}$ comparisons are necessary and sufficient to reconstruct the function.*

Proof. The upper bound follows by comparing every W-regulator with every $\bar{W}$-regulator. The lower bound follows from the fact that, in the special case where every $\bar{W}$-regulator precedes every W-regulator, no set of comparisons can determine the relative order of a given pair consisting of a W-regulator and a $\bar{W}$-regulator, unless it includes a direct comparison between the pair. Therefore, all such comparisons must be performed. □

Note that the problem of reconstructing a chain function by comparisons, once the regulators have been typed, can be viewed as a sorting problem: The input is a list of n elements of two types, such that the set of elements of each type consists of several equivalence classes, and there is a linear order of all these classes. The objective is to find the equivalence classes and their order, using only queries that compare two elements of distinct types. In the special case that each equivalence class consists of one element, the problem is related[1] to the well-studied problem of matching nuts and bolts [2] and has an optimal $\Theta(n \log n)$ deterministic solution [19].

We now turn to the question of reconstructing a chain function without prior knowledge of the identity of its regulators. The discussion above suggests a way to solve the problem: First, we find the gene types using N typing experiments. Next, we reconstruct the block structure by performing all possible comparisons between a W-gene and a $\bar{W}$-gene.

A more efficient reconstruction is possible when g_n is known. This is often the case when the chain function models a signal transduction pathway, where g_n represents

[1] The difference between the problem of matching nuts and bolts and our problem is that in our case we have strict linear order among all the elements and there is no notion of matching between W-regulators and $\bar{W}$-regulators.

a known stimulator of the corresponding biological response. If g_n is known, then since $g_n \in \bar{W}$, all W-regulators can be identified by comparing every W-gene with g_n, using a total of $N - k_{\bar{W}}$ comparisons. Since every $\bar{W}$-gene is a regulator, these experiments are sufficient to identify all the regulators, and we can apply Proposition 4 to complete the reconstruction in $N - k_{\bar{W}} + k_W(k_{\bar{W}} - 1)$ comparisons. In summary, we have the following proposition.

PROPOSITION 5. *A chain function can be reconstructed using at most N typing experiments and $k_{\bar{W}}(N - k_{\bar{W}})$ comparisons. Given g_n, a chain function can be reconstructed using at most $N - 1$ typing experiments and $N - n + k_W k_{\bar{W}}$ comparisons.*

We can prove a matching lower bound by generalizing the argument in Proposition 4.

PROPOSITION 6. *At least $k_{\bar{W}}(N - k_{\bar{W}})$ comparisons are necessary to reconstruct a chain function.*

Proof. Consider the case where all $\bar{W}$-regulators precede the W-regulators. In this case, no set of comparisons can determine the relative order of a given pair consisting of a W-gene and a $\bar{W}$-gene unless it includes a direct comparison between the pair. Therefore, all such comparisons must be performed. □

Propositions 4 and 5 provide a worst-case analysis. Next, we describe another reconstruction algorithm, whose *expected* number of required experiments is lower. The analysis of the running time is similar to that of quick-sort (cf. [5]) and assumes that the chain to be reconstructed has $\bar{W}$-blocks of bounded size. Denote by D_g the set of W-successors of $g \in \bar{W}$ in f.

PROPOSITION 7. *A chain function with $\bar{W}$-blocks of size bounded by d can be reconstructed using N typing experiments and an expected number of $O(Nd \log k_{\bar{W}} + k_W k_{\bar{W}})$ comparisons.*

Proof.

Algorithm: First, we perform N typing experiments. Next, we apply a randomized scheme to reconstruct the chain: Each time we pick a gene $g \in \bar{W}$ at random, find its successors and their order, and remove g and all its successors from further consideration. We stop when no $\bar{W}$ genes are left. In order to find the successors of g, we first identify the members of D_g using at most $N - k_{\bar{W}}$ comparisons. Using D_g, we then reconstruct the part of the chain that spans g and its successors by at most $|D_g|(k_{\bar{W}} - 1)$ comparisons, as in Proposition 4.

Complexity: The set of comparisons can be divided into two parts: those that are required to identify the sets D_g and those required to reconstruct the chain parts induced by these sets. For the latter, at most $k_W k_{\bar{W}}$ comparisons are needed in total, since every pair consisting of a W-regulator and a $\bar{W}$-regulator is compared at most once. Thus, it suffices to compute the expectation of the first part. Let $T(x)$ be this expectation, given that the current $\bar{W}$ set (i.e., the set of $\bar{W}$-genes that were not removed in previous iterations) contains x elements, where $T(0) = 0$. For $x \geq 1$, with probability $\frac{1}{x}$ the qth rightmost element of $\bar{W}$ is chosen in the current iteration. Hence, $T(x) \leq \frac{1}{x} \sum_{q=1}^{x} (d(N - k_{\bar{W}}) + T(x - q))$. By induction, $T(x) \leq d(N - k_{\bar{W}})(\log x + 1)$. Substituting $x = k_{\bar{W}}$, we obtain the required bound. □

The expected number of experiments improves over the upper bound of Proposition 5 for $d < k_{\bar{W}}$, which is the case in many real biological regulations, e.g., the filamentous-invasion pathway ($n = 9$, $k_{\bar{W}} = 2$, and $d = 1$, illustrated in [11, Figure 3]), and the HOG signaling pathway ($n = 6$, $k_{\bar{W}} = 3$, and $d = 2$ [13]) in yeast.

3.3. Using high-order experiments. In this section we show how to improve the above results when using experiments of order $q > 2$. The results in this section

are mainly of theoretical interest, since high-order experiments may not be practical.

PROPOSITION 8. *Given the set of n regulators of a chain function, the function can be reconstructed using $O(\frac{n^2}{q}\log q)$ experiments of order at most $q \le n$. This is optimal up to constant factors for $q = \Theta(n)$.*

Proof. The number of possible chain functions with n regulators is $\Theta((\log_2 e)^{n+1}n!)$ [9]. Since each experiment provides one bit of information, the information lower bound is $\Omega(n \log n)$ experiments.

Suppose at first that $q = n$. Let n_i be the number of regulators in block i, where blocks are indexed in right-to-left order. Our reconstruction algorithm is as follows: First, we perform n typing experiments. Next, we identify the type of the first block using one experiment of order n, in which all regulators are perturbed (this way we perturb also the genes in the first block, and thus its type is identical to the output). We proceed to reconstruct the blocks one by one, according to their order along the chain. Note that the type of each block is now known, since the two types alternate. Suppose we have already reconstructed blocks $1, \ldots, i-1$. For reconstructing the ith block we only consider the set of regulators that do not belong to the first $i-1$ blocks. Out of this set, let A be the subset of regulators that have the same type as block i, and let B be the subset of regulators of the opposite type. In order to identify the members of the ith block we use a binary-search-like procedure: We divide A into two halves. For each half we perform a perturbation that includes that half and all regulators in B. If the result is the type of block i, we continue recursively with that half. Otherwise, we discard it. The search requires $O(n_i \log n)$ experiments. Thus, altogether we perform $O(n \log n)$ experiments.

When $q < n$, we use the above algorithm as a component in our reconstruction scheme, allowing us to reconstruct a subchain of size q within a chain of size n using $O(q \log q)$ experiments of order at most q. Our reconstruction scheme is based on Proposition 4, which shows that for reconstruction it suffices to compare every W-regulator with every $\bar{W}$-regulator. To this end we form $O(\frac{n^2}{q^2})$ regulator subsets, each of size at most q, such that every pair consisting of a W-regulator and a $\bar{W}$-regulator appears in one of the subsets. To compute these subsets we form a $k_W \times k_{\bar{W}}$ matrix, whose entries are in 1-1 correspondence with $(W, \bar{W})$-regulator pairs. We then cover this matrix using $O(\frac{k_W k_{\bar{W}}}{q^2})$ disjoint submatrices of dimension at most $\lfloor q/2 \rfloor \times \lceil q/2 \rceil$, each identifying a regulator subset of the required size.

Next, we reconstruct the subchain of size q associated with each subset using $O(q \log q)$ experiments of order at most q. After this process, each $(W, \bar{W})$-regulator pair appears in one of the subchains, and thus its relative order has been determined. This is sufficient in order to computationally reconstruct the chain (as in Proposition 4). Altogether we use $O(\frac{k_W k_{\bar{W}}}{q} \log q) = O(\frac{n^2}{q} \log q)$ experiments for reconstructing the chain from its regulators. □

We now provide a reconstruction scheme for the case that the set of regulators is not known. Let f be a chain function. For a gene $g \in \bar{W}$, denote as before by D_g its set of W-successors in f. A building block in our reconstruction scheme is a method to efficiently identify the members of D_g using $O(|D_g| \log q + N/q)$ experiments of order at most q. The process is as follows: We partition the W-genes into $\lceil \frac{N}{q-1} \rceil$ subsets of size at most $q-1$. For each subset R we test whether it contains some successor of g using an $(R \cup \{g\})$-perturbation, in which g and the subset members are perturbed. If as a result of the perturbation the output changes to w, then at least one of the members in R succeeds g. In this case we use standard binary search to identify all the m successors in R by performing additional $O(m \log q)$ experiments of order at

most $(\lfloor q/2 \rfloor + 1)$. Otherwise, all the subset members precede g and we discard R. Each of the successors of g is discovered exactly once, which gives the required bound.

PROPOSITION 9. *For $q \leq n$, a chain function can be reconstructed using $O(nN/q + n^2 \log q/q)$ experiments of order at most q. For $q > n$, $O(N + n \log q)$ experiments of order at most q are sufficient.*

Proof. The reconstruction is done in three stages. First, we perform N typing experiments. Second, we discover all W-regulators as follows: For each regulator $b \in \bar{W}$ we use the scheme described above to identify its successors in W, and remove them from further consideration. Each W-regulator is discovered exactly once and, thus, we need $O(k_{\bar{W}} N/q + k_W \log q)$ experiments of order at most q altogether. Last, we reconstruct the chain, given the regulators and their types, in $O(n^2 \log t/t)$ experiments, using the method given in Proposition 8, where $t = \min\{q, n\}$. In total $O(N + k_{\bar{W}} N/q + k_W \log q + n^2 \log t/t)$ experiments are used. □

A lower bound on the number of experiments that are required is given in the following proposition.

PROPOSITION 10. $\Omega(\max\{N/q, nN/q^2, n \log N\})$ *experiments of order at most q are necessary to reconstruct a chain function.*

Proof. We give three different lower bounds, whose union yields the required result. First, $\Omega(N/q)$ experiments are required to identify at least one $\bar{W}$-regulator. Second, $\Omega(nN/q^2)$ experiments are required to cover every pair of a W- and a $\bar{W}$-gene. Third, the number of possible chain functions is $\Theta(\binom{N}{n}(\log_2 e)^{n+1} n!)$ [9]. Hence, the information theoretic lower bound on the reconstruction is $\Omega(n \log N)$. □

Finally, we give an optimal reconstruction scheme when g_n is known and $q = \lceil N/2 \rceil + 1$.

PROPOSITION 11. *In case g_n is known, there is an optimal reconstruction scheme that uses $\Theta(n \log N)$ experiments of order at most $\lceil N/2 \rceil + 1$.*

Proof. We perform the reconstruction in two stages. In the first stage we discover the set of regulators and their types. In the second stage we apply Proposition 8 to reconstruct the chain function. To discover the set of regulators we perform a binary-search-like process as follows: We partition all variables excluding g_n and g_0 into two halves, H_1 and H_2. For $i = 1, 2$ we apply an $H_i \cup \{g_n\}$-perturbation. Since g_n is perturbed, all nonregulator effects are masked, and we get the result w iff H_i contains some W-regulators. Therefore, for each set that gives the results w, we continue recursively until we reach single genes. In this way we have identified a subset T of the W-regulators, including all those in the first (rightmost) block. We now repeat the recursive process on $U \setminus (T \cup g_n \cup g_0)$, but this time do not include g_n in the perturbations. This process identifies a subset T' of the $\bar{W}$ regulators, including the first $\bar{W}$-block. By repeating these two recursive processes (with and without including g_n in the perturbations) we eventually identify all regulators. The total effort is $O(n \log N)$ since each path that identifies one of the n regulators is a binary search in N variables and thus takes $O(\log N)$ experiments. □

4. Combining several chains. In this section we extend the notion of a chain function to cover common biological examples in which the regulatee state is a boolean function of several chains. Frequently, a combination of several signals influences the transcription of a single regulatee via several pathways that carry these signals to the nucleus, and a regulation function that combines them together. Here, we formalize this situation by modeling each signal transduction pathway by a chain function, and letting the outputs of these paths enter a boolean gate.

Define a *k-chain function* f as a boolean function which is composed of k chain

functions over disjoint sets of regulators that enter a boolean gate $G(f)$. Let f^i be the ith chain function and let g^i_j denote the jth regulator in f^i. The output of the function is $G(infl(g^1_1), \ldots, infl(g^k_1))$.

In the following we present several biological examples for k-chain functions that arise in transcriptional regulation in different organisms: The lac operon [21] codes for lactose utilization enzymes in *E. coli*. It is under both negative and positive transcriptional control. In the absence of lactose, lac-repressor protein binds to the promoter of the lac operon and inhibits transcription. In the absence of glucose, the level of cAMP in the cell rises, which leads to the activation of CAP, which in turn promotes transcription of the lac operon. In our formalism, the lac operon is controlled by a 2-chain function with an AND gate. The chains are $f^1(g^1_2, g^1_1) = f^1$(lactose, lac-repressor), with control pattern 11, and $f^2(g^2_3, g^2_2, g^2_1) = f^2$(glucose, cAMP, CAP), with control pattern 100. Other examples of 2-chains with AND gates are the regulation of arginine metabolism and galactose utilization in yeast [18]. A 2-chain with an OR gate regulates lysine biosynthesis pathway enzymes in yeast [18].

These examples motivate us to restrict attention to gates that are either OR or AND. We first show that we can distinguish between OR and AND gates. We then show how to reconstruct k-chain functions in the case of OR and later extend our method to handle AND gates.

Denote the output of f^i by O_i. If $O_i = 1$ in wild type, we call f^i a 1-*chain* and, otherwise, a 0-*chain*. A regulator g^i_j is called a 0-*regulator* (1-*regulator*) if its perturbation produces $O_i = 0$ ($O_i = 1$). Let k_0 (k_1) be the number of 0-regulators (1-regulators) in f. A block is called a 0-*block* (1-*block*), if it consists of 0-regulators (1-regulators).

LEMMA 12. *Given a k-chain function f with gate $G(f)$ which is either AND or OR, $k \geq 2$, we can determine, using $O(N^2)$ experiments of order at most 2, whether $G(f)$ is an AND gate or an OR gate.*

Proof. We perform N typing experiments. If $w = 0$ and $\bar{W} = \emptyset$, then $G(f)$ is an AND gate. If $w = 1$ and $\bar{W} = \emptyset$, then $G(f)$ is an OR gate. Otherwise, $\bar{W} \neq \emptyset$. In this situation the cases of $w = 0$ and $w = 1$ are similarly analyzed. We describe only the former.

If $w = 0$, we have to differentiate between the case of an OR gate, whose inputs are all 0-chains, and the case of an AND gate, whose inputs are one 0-chain and $(k-1)$ 1-chains. To this end we perform all comparisons of a W-gene and a $\bar{W}$-gene. Let T be the set of genes g such that the result of a $\{g, g'\}$-perturbation is w for every $g' \in \bar{W}$. Then $T \neq \emptyset$ iff $G(f)$ is an AND gate. □

We now study the reconstruction of an OR gate. Let S be the (possibly empty) set of regulators that reside in one of the first blocks (i.e., the blocks containing g^i_1), that are also 1-blocks. We observe that a perturbation of any regulator in S results in $state(g_0) = 1$ regardless of any other simultaneous perturbations we may perform. Hence, determining the specific chain to which an element from S belongs is not possible. Therefore, our reconstruction will be unique up to the ordering within blocks and up to the assignment of the regulators in S to their chains. The next lemma handles the case $w = 0$. The subsequent lemma treats the case $w = 1$.

LEMMA 13. *Given a k-chain function f with an OR gate and assuming that $w = 0$, we can reconstruct f using N typing experiments and $(N - k_1)k_1$ comparisons.*

Proof. We perform N typing experiments. Then, for each 1-regulator b, we perform all possible comparisons, thereby identifying all 0-regulators that succeed b in its chain. This completes the reconstruction. □

LEMMA 14. *Let f be a k-chain function with an OR gate. Assume that $w = 1$, and let r be the number of 1-chains entering the OR gate. Then f can be reconstructed using $O(N^r + Nn)$ experiments of order at most $r + 2$.*

Proof. First, we determine r, the minimum order of an experiment that will produce output 0 for f. For $i = 1, 2, \ldots$ we perform all possible i-order experiments; r is determined as the smallest i for which we obtain output 0. In total we perform $O(N^r)$ experiments. We call the set of perturbed genes in an r-order experiment which results in output 0, a *reset combination*.

Next, we reconstruct the 1-chains. Fix an arbitrary reset combination R. For every $a \in R$ we perform a set of experiments of order $r + 1$ as follows: For every reset combination $R' \supset R \setminus \{a\}$ with $a \notin R'$, we perturb R' and in addition each other gene, one at a time, recording those that produce output 1 as 1-regulators. For every a, the sets of 1-regulators discovered in these experiments form a linear order under set inclusion. The 1-regulators that are *not* common to all these sets are exactly the 1-regulators (that are not in S) of the chain that includes a. For each 0-regulator in $R' \setminus R$ our experiments determine the 1-regulators that succeed it in this chain. Thus, we can infer all the 1-chains. The total number of experiments performed is $O(Nk_0)$.

Finally, we reconstruct the 0-chains. To this end we perturb the 1-regulators in R, thereby deactivating the 1-chains and reducing the problem of reconstructing the 0-chains to that of reconstructing a $(k - r)$-chain function with an OR gate and $w = 0$ (removing the already discovered regulators of the 1-chains from consideration). This is done by applying the reconstruction method of Lemma 13 using $O(Nk_1)$ experiments of order at most $r + 2$. The assignment of 1-regulators in S will remain uncertain. ∎

Note that for $k = 1$ the above algorithms will reconstruct a single chain. Indeed, for $w = 0$ the algorithm of Lemma 13 coincides with that of section 3, and for $w = 1$, applying the algorithm of Lemma 14 we shall discover that $r = k = 1$. Further note that for every reconstructed chain we can identify whether its first block is a 1-block (i.e., contains genes in S). This is simply done by computing for that chain the value of $state(g_n) \oplus (\oplus_i y_i)$ on its known members and comparing it to the chain's output. Last, note that if k is known and $r = k$, then the order of the experiments that are required to reconstruct the k-chain is at most $r + 1$, since f contains no 0-chains.

The reconstruction method for the case of an OR gate can be used for the reconstruction of an AND gate as well, by exchanging the roles of 0 and 1 in the above description. This gives rise to the following result:

THEOREM 15. *A k-chain function with an OR or an AND gate can be reconstructed using $O(N^k)$ experiments of order at most $k+1$. The reconstruction requires $\Omega(\binom{N}{k}/k)$ experiments of this order.*

Proof. The upper bound follows from Lemmas 12, 13, and 14 and the duality of AND and OR gates. For the lower bound consider a k-chain function with an OR gate consisting of k 1-chains, each of which contains a single 0-regulator. Such a function has a single reset combination, which must be identified in the process of reconstructing the chain. Since each experiment of order $k + 1$ can test at most k combinations, $\Omega(\binom{N}{k}/k)$ experiments are required for the reconstruction. ∎

5. A biological application. The methods we presented above can be applied to reconstruct chain functions from biological data. We describe one such application to the reconstruction of the yeast galactose regulation function, for which some of the required perturbations have been performed. We show that one additional experiment suffices to fully reconstruct the regulation function.

The galactose utilization in the yeast *Saccharomyces cerevisiae* [18] occurs in a biochemical pathway that converts galactose into glucose-6-phosphate. The transporter gene gal2 encodes a protein that transports galactose into the cell. A group of enzymatic genes, gal1, gal7, gal10, gal5, and gal6, encode the proteins responsible for galactose conversion. The regulators gal4p, gal3p, and gal80p control the transporter, the enzymes, and to some extent each other (Xp denotes the protein product of gene X). In the following, we describe the regulatory mechanism. gal4p is a DNA binding factor that activates transcription. In the absence of galactose, gal80p binds gal4p and inhibits its activity. In the presence of galactose in the cell, gal80p binds gal3p. This association releases gal4p, promoting transcription. This mechanism can be viewed as a chain function, where $f^1(g_4^1, g_3^1, g_2^1, g_1^1) = f^1(galactose, gal3, gal80, gal4)$, and the corresponding control pattern is 0110 (see also [9]). The gal7, gal10, and gal1 regulatees are also negatively controlled by another chain $f^2(g_2^2, g_1^2) = f^2(glucose, mig1)$ with control pattern 01. The two chains are combined by an AND gate (see Figure 2(A)).

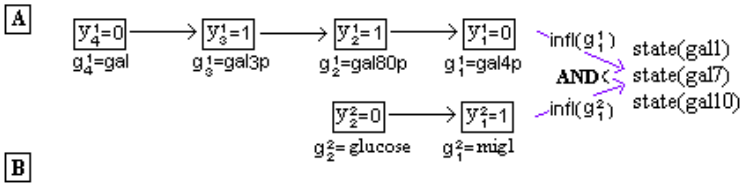
Ideker et al. [14] performed several experiments to interrogate the galactose utilization mechanism. In these experiments glucose was absent from the media. Consequently, the output of f^2 was always 1, and hence we shall focus on the reconstruction of f^1 using the experimental data of [14]. Using the discretization procedure employed by Ideker et al. [14], the measured wild-type levels of gal3, gal80, and gal4 were 1, in accordance with our model assumption. The wild-type level of galactose was also 1.

Assuming we know the group of four regulators, we need, according to Proposition 4, a total of 4 typing experiments and 3 comparisons (since only gal80 is of type W) to reconstruct the chain. Notably, all 4 typings and 2 of the 3 comparisons[2] were performed by Ideker et al. [14] (see Figure 2(B)). Using the same discretization procedure, the experiments yielded the correct results for all three regulatees. The results suggest two possible chain functions: $f^1(g_4^1, g_3^1, g_2^1, g_1^1) = f^1(galactose, gal3, gal80, gal4)$ or $f^1(g_4^1, g_3^1, g_2^1, g_1^1) = f^1(galactose, gal80, gal3, gal4)$, both with control pattern 0110. The missing experiment is a comparison of gal80 and gal3. A correct result of this experiment will lead to full and unique reconstruction of the chain function.

6. Concluding remarks. In this paper we studied the computational problems arising when wishing to reconstruct regulation relations using a minimum number of experiments, assuming that the experiments' results are noiseless. We restricted attention to common biological relations, called chain functions, and exploited their special structure in the reconstruction. We also suggested an extension of that model, which combines several chain functions, and studied some of the same reconstruction questions for the extended model. On the practical side, we have shown an application of our reconstruction scheme for inferring the regulation of galactose utilization in yeast.

The task of designing optimal experimental settings is fundamental in meeting the great challenge of regulatory network reconstruction. While this task entails coping with complex interacting regulation functions and noisy biological data, we chose here to focus on the reconstruction of a single regulation relation of a single regulatee and assume that the function can be studied in isolation. Hence, upon any perturbation, none of the other regulators change their states. Another major

[2]In fact, the gal80Δgal4Δ-gal experiment was of order 3 but allowed the comparison of gal80 and gal4.



B

Ideker et al. Experiment name	galactose	gal3	gal80	gal4	regulatee	
wt-gal	0	1	1	1	$\bar{w}$	Typing experiments
gal3Δ+gal	1	0	1	1	$\bar{w}$	
gal80Δ+gal	1	1	0	1	w	
gal4Δ+gal	1	1	1	0	$\bar{w}$	
gal80Δ-gal	0	1	0	1	w	Comparison experiments
gal80Δgal4Δ-gal	0	1	0	0	$\bar{w}$	
not perfomed	1	0	0	1	?	

FIG. 2. *Galactose pathway regulation.* (A) *The* 2*-chain function regulating gal*1*, gal*7*, and gal*10 *transcription.* (B) *Typing and comparison experiments performed by Ideker et al.* [14].

assumption is that the wild-type state of all regulators (except possibly g_n) is 1. This assumption, which is necessary for the analysis (e.g., Lemma 3) is commonly held in undelayed biological systems, where all the regulators exist in a certain basal level and the signal can propagate fast (e.g., MAPK systems in unicellular organisms such as yeast and multicellular organisms including humans, reviewed in [17]). Regulations that involve production of absent regulators are typically (slow) temporal processes. Our analysis should be extended in order to deal with such complex regulations and temporal processes.

This analysis focuses on theoretical complexity of regulation reconstruction, assuming perturbation experiments that measure (accurately) only gene states. It is clear, however, that other experimental techniques (e.g., interaction measurements [7, 20]) might help to constrain the reconstruction and reduce the solution space. In a practical approach, diverse data sources should be incorporated, and the experiments should be designed dynamically and take into consideration the experimental noise. The theoretical analysis here could hopefully serve as a component in such a practical experimental design.

REFERENCES

[1] T. AKUTSU, S. KUHARA, O. MARUYAMA, AND S. MIYANO, *Identification of genetic networks by strategic gene disruptions and gene overexpressions under a boolean model*, Theoret. Comput. Sci., 298 (2003), pp. 235–251.

[2] N. ALON, M. BLUM, A. FIAT, S. KANNAN, M. NAOR, AND R. OSTROVSKY, *Matching nuts and bolts*, in Proceedings of the 5th Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, Philadelphia, 1994, pp. 690–696.

[3] M. ANTHONY, *The Sample Complexity and Computational Complexity of Boolean Function Learning*, Tech. Report LSE-CDAM-2002-13, London School of Economics and Political Science, London, UK, 2002.

[4] N. H. BSHOUTY, *Exact learning Boolean function via the monotone theory*, Inform. and Comput., 123 (1995), pp. 146–153.

[5] T. H. CORMEN, C. E. LEISERSON, AND R. L. RIVEST, *Introduction to Algorithms*, MIT Press, Cambridge, MA, 1990.

[6] J. Derisi, V. Iyer, and P. Brown, *Exploring the metabolic and genetic control of gene expression on a genomic scale.*, Science, 282 (1997), pp. 699–705.
[7] A. H. Tong et al., *Global mapping of the yeast genetic interaction network*, Science, 303 (2004), pp. 808–13.
[8] N. Friedman, M. Linial, I. Nachman, and D. Pe'er, *Using Bayesian networks to analyze expression data*, J. Comp. Biol., 7 (2000), pp. 601–620.
[9] I. Gat-Viks and R. Shamir, *Chain functions and scoring functions in genetic networks*, Bioinformatics, 19, Supplement 1 (2003), pp. 108–117.
[10] I. Gat-Viks, R. Shamir, R. M. Karp, and R. Sharan, *Reconstructing chain functions in genetic networks*, in Proceedings of the Ninth Pacific Symposium on Biocomputing (PSB'04), 2004.
[11] M. C. Gustin, J. Albertyn, M. Alexander, and K. Davenport, *Map kinase pathways in the yeast Saccharomyces cerevisiae*, Microbiol. Mol. Biol. Rev., 62 (1998), pp. 1264–1300.
[12] D. Hanisch, A. Zien, R. Zimmer, and T. Lengauer, *Co-clustering of biological networks and gene expression data*, Bioinformatics, 18, Supplement 1 (2002), pp. 145–154.
[13] S. Hohmann, *Osmotic stress signaling and osmoadaptation in yeasts.*, Microbiol. Mol. Biol. Rev., 66 (2002), pp. 300–372.
[14] T. Ideker et al., *Integrated genomic and proteomic analyses of systematically perturbed metabolic network*, Science, 292 (2001), pp. 929–933.
[15] T. Ideker, O. Ozier, B. Schwikowski, and A. F. Siegel, *Discovering regulatory and signaling circuits in molecular interaction networks.*, Bioinformatics, 18, Supplement 1 (2002), pp. 233–240.
[16] T. Ideker, V. Thorsson, and R. M. Karp, *Discovery of regulatory interaction through perturbation: Inference and experimental design*, in Proceedings of Pacific Symposioum in Biocomputing, 2000, pp. 305–316.
[17] G. L. Johnson and R. Lapadat, *Motigen-activated protein kinase pathways mediated by ERK, JNK, and p*38 *protein kinases*, Science, 298 (2002), pp. 1911–12.
[18] E. W. Jones, J. R. Pringle, and J. R. Broach, eds., *The Molecular and Cellular Biology of the Yeast Saccharomyces: Gene Expression*, Cold Spring Harbor Laboratory Press, Cold Spring Harbor, NY, 1992.
[19] J. Komlós, Y. Ma, and E. Szemerédi, *Matching nuts and bolts in* $o(n \log n)$ *time*, SIAM J. Discrete Math., 11 (1998), pp. 347–372.
[20] T. I. Lee et al., *Transcriptional regulatory networks in Saccharomyces Cerevisiae*, Science, 298 (2002), pp. 799–804.
[21] F. C. Neidhardt, ed., *Escherichia coli and Salmonella: Cellular and Molecular Biology*, ASM Press, 1996.
[22] D. Pe'er, A. Regev, and A. Tanay, *Minreg: Inferring an active regulator set*, Bioinformatics, 18, Supplement 1 (2002), pp. 258–267.
[23] E. Segal, B. Taskar, A. Gasch, N. Friedman, and D. Koller, *Rich probabilistic models for gene expression*, Bioinformatics, 17, Supplement 1 (2001), pp. 243–252.
[24] A. Tanay and R. Shamir, *Computational expansion of genetic networks*, Bioinformatics, 17, Supplement 1 (2001), pp. 270–278.