Reverse engineering discrete dynamical systems from data sets with random input vectors

Winfried Just*

April 25, 2006

Abstract

Recently a new algorithm for reverse engineering of biochemical networks was developed by Laubenbacher and Stigler. It is based on methods from computational algebra and finds most parsimonious models for a given data set. We derive mathematically rigorous estimates for the expected amount of data needed by this algorithm to find the correct model. In particular, we demonstrate that for one type of input parameter (graded term orders), the expected data requirements scale polynomially with the number n of chemicals in the network, while for another type of input parameters (randomly chosen lex orders) this number scales exponentially in n. We also show that for a modification of the algorithm, the expected data requirements scale as the logarithm of n.

1 Introduction

Reverse engineering of biochemical networks is a central problem of systems biology. This process can utilize data from a variety of sources [18]. *Top-down* approaches are based on the observed network response to different inputs. For example, microarray data collected on time series for selected knockout/overexpression experiments or environmental perturbations of the wild-type network can be used for top-down reverse engineering of gene regulatory networks.

A number of algorithms for top-down reverse engineering under a variety of modeling paradigms have been proposed (see [7], [8], [9] for recent surveys). It would be very useful to know which of these algorithms (or even which modeling paradigms) are most suitable for the analysis of which kind of data sets. The greatest handicap in reverse engineering of biochemical networks is that the problem tends to be heavily underdetermined, so that a large number of network models are consistent with the data (see Section 4 for details). Any reverse engineering algorithm must select one or a few of these feasible models according to some criteria. Therefore, the perhaps most important criterion for usability of an algorithm

*Mathematical Biosciences Institute, 250 Mathematics Building, 231 W 18th Ave, Columbus, OH 43210 and Department of Mathematics, Ohio University, Athens, OH 45701

is whether for a data set of given size it can be expected to select the correct model with reasonably high probability.

In [17] and [21], Laubenbacher and Stigler developed a top-down reverse engineering algorithm that is based on treating the network as a discrete dynamical system, obtained by discretizing concentration levels of the participating chemicals to elements of a finite field. This generalizes the familiar notion of Boolean dynamical systems [14] (where the field is $\mathbb{F}_2$) and allows the use of advanced tools from computational algebra. Moreover, this approach permits a strict mathematical definition of the *most parsimonious* model consistent with the data, according to some chosen term order $\preceq$, which is an input parameter. We will refer to this algorithm as the *LS-algorithm.* Recently, a modification of the LS-algorithm has been developed, implemented and tested [11]. We will refer to the latter as the *LS-algorithm with preprocessing.*

The purpose of this note is to investigate the expected performance of these algorithms. More specifically, we estimate the expected amount of data needed by these algorithms for finding the correct model. In order to be able to derive mathematically rigorous estimates of these data requirements, we assume that data have been collected on the system response to randomly chosen concentration vectors. A detailed discussion of the appropriateness of this assumption is included in Section 4. Under this assumption, we are able to give upper and lower bounds for the expected data requirements of the LS-algorithm for the most common types of input parameters, graded and lex term orders. We show that for a randomly chosen graded term order, the expected data requirements scale as a polynomial in the number n of chemicals in the network. For randomly chosen lex orders, the expected data requirements of the LS-algorithm scale exponentially in n. In contrast, for the LS-algorithm with preprocessing the expected data requirements scale logarithmically in terms of n and are within the best known bounds for reverse engineering algorithms of discrete dynamical systems.

Since our work is aimed primarily at finding ball-park estimates that could give useful guidance for use of the algorithm and development of future refinements, we will state our major technical results twice, both as a precise formula and as a more user-friendly estimate in terms of the big-Oh notation.

The paper is organized as follows. In Section 2 we give a description of the LS-algorithm, a mathematically rigorous formulation of the questions studied in this paper, as well as the necessary background from abstract algebra. Section 3 contains our technical results. Section 4 contains a summary and discussion of the main results proved in Section 3. The reader who wishes to skip proofs during the first reading of this paper may proceed to Section 4 right after Subsection 2.3.

2 Mathematical background

We will use the notation $[n]$ for the set $\{1, \ldots, n\}$ of the first n positive integers, and $\mathbb{Z}^+$ for the set of all positive integers. The cardinality of a set X is denoted by $|X|$.

2.1 The LS-algorithm in a nutshell

The LS-algorithm attempts to reconstruct models for regulation of biochemical networks (such as gene networks) from data such as time series of discretized concentration level vectors. It works as follows:

- There are n chemical species (mRNA, proteins, or metabolites) in the network.
- The concentration level measurements for each species have been discretized to elements of a finite field F (typically $\mathbf{F}_p$ for some prime p).
- There are n input variables $x_1, \ldots, x_n$ that measure the concentration levels of the n chemical species. They take values in F.
- There is one output variable y that also takes values in F. In [17] and [21], based on one or several time series, this output variable is also one of the input variables for the next time point. However, this assumption is not needed in general.
- Experimental data take the form of a set $D = \{< \bar{x}(t), y(t) >: t \in [m]\}$, where each $\bar{x}(t) = [x_1(t), \ldots, x_n(t)]$ is the input vector of concentrations, and $y(t)$ is the measurement obtained in response to input $\bar{x}(t)$. The set of inputs will be denoted by $C = \{\bar{x}(t) : t \in [m]\}$.
- The standing assumption is that all experimental results will be consistent, that is, the responses to the same input vector will be the same if we repeat an experiment.
- If the set C of data inputs is the set of all possible input vectors F^n, then there exists exactly one polynomial $h_{true} \in F[x_1, \ldots, x_n]$ such that $h_{true}(\bar{x}(t)) = y(t)$ for all $t \in [m]$. Thus if we could perform each of the $|F|^n$ possible experiments, then we could completely characterize the system (network).
- The set $I_C = \{h \in F[x_1, \ldots, x_n] : \forall t \in [m]\ h(\bar{x}(t)) = 0\}$ of all polynomials that vanish on all data inputs forms an ideal in $F[x_1, \ldots, x_n]$.
- We call a polynomial $h \in F[x_1, \ldots, x_n]$ a *model for* D if $h(\bar{x}(t)) = y(t)$ for all $t \in [m]$. The set of all models for D is the set $h_{true} + I_C$.
- The LS-algorithm takes as input a data set D and a term order $\preceq$, and outputs a model of D.
- The LS-algorithm finds a "most parsimonious guess" of h_{true} by first constructing one model h for D and then computing and returning the remainder $h\%G_{\preceq}$ under division of h by a Gröbner basis $G_{\preceq}$ of I_C. If the data set is complete (*i.e.*, if $C = F^n$), then the algorithm is guaranteed to return h_{true}.

2.2 Some elementary facts about Gröbner bases

A *monomial* in the polynomial ring $F[x_1, \ldots, x_n]$ is an expression of the kind x^α, where α is a function from $[n]$ into the set of nonnegative integers, called a *multiexponent.* This notation should be interpreted as $x^\alpha = \prod_{i \in supp(\alpha)} x_i^{\alpha(i)}$. A *term* in a polynomial is an expression of the kind ax^α, where x^α is a monomial and $a \in F \backslash \{0\}$. The *support of* α is the set $supp(\alpha) = \{i \in [n] : \alpha(i) > 0\}$. The *support of a polynomial* h*,* denoted by $supp(h)$, is the union of the sets $supp(\alpha)$ for all terms ax^α of h.

In a finite field F we always have $x^{|F|} = x$ and hence we may ignore multiexponents α with $max\, \alpha \geq |F|$; from now on we will always assume that $max\, \alpha < |F|$. In particular, if we work in $\mathbf{F}_2$, then we can identify multiexponents with their support sets.

Every ideal I in $F[x_1, \ldots, x_n]$ has sets of generators called *Gröbner bases,* which will be defined below. Division of a polynomial h by all polynomials in a Gröbner basis G yields a unique remainder $h\%G$. In the literature, the remainder $h\%G$ is often called the *normal form (with respect to G) of* h*.* This remainder has the property that $h - h\%G \in I$. Moreover, if $h - h' \in I$, then $h\%G = h'\%G$. The LS-algorithm computes a certain Gröbner basis G and returns h_{true} iff $h_{true}\%G = h_{true}$.

A *term order* is any well-order $\preceq$ on the set of monomials such that $x^\alpha \preceq x^\beta$ implies $x^\alpha x^\gamma \preceq x^\beta x^\gamma$. We will slightly abuse notation and use the symbol $\preceq$ also on the set of multiexponents; *i.e.*, we will write $\alpha \preceq \beta$ interchangably with $x^\alpha \preceq x^\beta$ and call both usages "term order."

An example of term orders are *lex orders.* Each lex(icographical) term order is given by a *variable order* $x_{\pi(1)} \succeq \ldots \succeq x_{\pi(n)}$, where $\pi : [n] \to [n]$ is a permutation. The lex order $\preceq_\pi$ is then defined as follows: If $\alpha \neq \beta$, let j_d be the smallest $j \in [n]$ such that $\alpha(\pi(j)) \neq \beta(\pi(j))$. Then

$$\alpha \preceq_\pi \beta \leftrightarrow (\alpha = \beta \vee \alpha(\pi(j_d)) < \beta(\pi(j_d))).$$

Another important class of term orders are the *graded term orders.* For a multiexponent α, let $\sum \alpha = \sum_{i \in [n]} \alpha(i)$. A term order $\preceq$ is called *graded* if $\sum \alpha < \sum \beta$ implies $\alpha \prec \beta$.

With each term order $\preceq$ and ideal I one can associate a canonical Gröbner basis (a so-called *reduced Gröbner basis*), which will be denoted here by $G_\preceq(I)$.

2.3 The questions studied in this paper

The general question we are investigating is:

Question 1 *If h_{true} is a given polynomial in $F[x_1, \ldots, x_n]$ and $\preceq$ is a term order taken by the LS-algorithm as input, how much data do we need on average so that we can expect the LS-algorithm to return h_{true}? Equivalently, how large does the set C of data inputs on average need to be so that we can expect $h_{true}\%G_\preceq(I_C) = h_{true}$?*

Since most regulatory functions in biochemical networks have relatively small support relative to the total number of chemicals in the network [4], we will be especially interested in Question 1 for h_{true} whose support has cardinality bounded by some constant.

In order to give precise meaning to the above question we need to define suitable random variables. Our general framework in this note will be the following: The letter h will always denote a polynomial in $F[x_1, \ldots, x_n]$. Then there exists exactly one data set of maximal size such that $h = h_{true}$ for this data set; it will be denoted by D_{max}. Now we imagine an experimenter who randomly samples data inputs $\bar{x}(t)$ from F^n and takes measurements $y(t)$. We will assume that the underlying distribution of data inputs is the uniform distribution on F^n and the sampling allows replacement. Thus our (extremely well-funded) experimenter will produce an infinite sequence of data points $\bar{D} =<< \bar{x}(t), y(t) >: t \in \mathbb{Z}^+ >$ with $y(t) = h(\bar{x}(t))$ for all t. Let $\bar{\mathcal{D}}_h$ denote the probability space of all possible such sequences. If we do not wish to specify h, we will work with the probability space $\bar{\mathcal{D}} = \bigcup_{h \in F[x_1,\ldots,x_n]} \bar{\mathcal{D}}_h$ with the uniform distribution (more precisely, the product measure of of the uniform distribution on single-point data sets). For each positive integer m we let $D_m = \{< \bar{x}(t), y(t) >: t \in [m]\}$ and $C_m = \{\bar{x}(t) : t \in [m]\}$. Note that with probability one there will be an m such that $D_m = D_{max} = D_{m'}$ and $C_m = C_{m'} = F^n$ for all $m' > m$. Thus for sufficiently large m, the LS-algorithm will return h_{true}.

Let $\mathcal{S}$ be a finite nonempty set of term orders together with a probability distribution. We will be interested in the cases where $\mathcal{S}$ is the set $\mathcal{L}$ of all lex orders with the uniform distribution, the set $\mathcal{G}$ of all graded term orders with the uniform distribution[1], or $\mathcal{S} = \{\preceq\}$ for a fixed term order $\preceq$.

Our experimenter now has two principally different ways of analyzing the data: In a Type 1 Analysis, the experimenter randomly picks a term order $\preceq$ from $\mathcal{S}$ and analyzes all data sets D_m by running the LS-algorithm with input $\preceq$. In a Type 2 Analysis, the experimenter randomly and independently picks a term order $\preceq_m$ for each m and analyzes data set D_m by running the LS-algorithm with input $\preceq_m$. We say that a Type 1 or Type 2 analysis *returns a polynomial h at step m* if the LS-algorithm returns h when run on D_m with the corresponding term order.

Definition 2 *Let $\mathcal{S}$ be a set of term orders and let $h \in F[x_1, \ldots, x_n]$. We define a random variable $\lambda_{h,\mathcal{S}}$ on $\bar{\mathcal{D}}_h$ as the smallest number m such that a Type 1 Analysis returns h at step m, a random variable $\kappa_{h,\mathcal{S}}$ on $\bar{\mathcal{D}}_h$ as the smallest number m such that a Type 2 Analysis returns h at step m, and a random variable $\nu_{h,\mathcal{S}}$ on $\bar{\mathcal{D}}_h$ as the smallest number m such that there exists $\preceq \in \mathcal{S}$ so that a Type 1 Analysis that uses $\preceq$ returns h at step m.*

Strictly speaking, the above random variables are only defined on a subset of $\bar{\mathcal{D}}_h$ of measure one, but this does not impact our results in any way, so we will ignore this technicality in the remainder of this paper. Note that in the definition of $\kappa_{h,\mathcal{S}}$ and $\lambda_{h,\mathcal{S}}$ we assume random choices of term orders; whereas in the definition of $\nu_{h,\mathcal{S}}$ we assume that the optimal $\preceq \in \mathcal{S}$ is used for the analysis.

[1]Formally, $\mathcal{G}$ is an infinite set, but since we are restricting our attention to multiexponents α with $\max \alpha < |F|$, we can treat this set as finite.

Question 1 translates into our new terminology as follows:

Question 3 *Given a finite set $\mathcal{S}$ of term orders and a polynomial $h \in F[x_1, \ldots, x_n]$, find estimates of* $\min \lambda_{\mathcal{S}}$, $\min \kappa_{h,\mathcal{S}}$, $\min \nu_{h,\mathcal{S}}$, $E(\lambda_{h,\mathcal{S}})$, $E(\kappa_{h,\mathcal{S}})$, $E(\nu_{h,\mathcal{S}})$.

Proposition 4 *Let h be any polynomial in $F[x_1, \ldots, x_n]$ and let $\mathcal{S}$ be a set of term orders. Then* $\min \lambda_{h,\mathcal{S}} = \min \kappa_{h,\mathcal{S}} = \min \nu_{h,\mathcal{S}}$.

Proof: Immediate from the definition. □

It also follows immediately from Definition 2 that $\nu_{h,\mathcal{S}}(\bar{D}) \leq \kappa_{h,\mathcal{S}}(\bar{D}), \lambda_{h,\mathcal{S}}(\bar{D})$ for all $\bar{D} \in \bar{\mathcal{D}}_h$, and hence $E(\nu_{h,\mathcal{S}}) \leq E(\kappa_{h,\mathcal{S}}), E(\lambda_{h,\mathcal{S}})$. It is perhaps also intuitively clear that $E(\kappa_{h,\mathcal{S}}) \leq E(\lambda_{h,\mathcal{S}})$, but the formal proof is not entirely straightforward, so we include it for completeness.

Proposition 5 *Let h be any polynomial in $F[x_1, \ldots, x_n]$ and let $\mathcal{S}$ be a set of term orders. Then $E(\kappa_{h,\mathcal{S}}) \leq E(\lambda_{h,\mathcal{S}})$.*

Proof: Let $h, \mathcal{S}$ be as above. For a given term order $\preceq$, let $G^m_{\preceq}$ denote the reduced Gröbner basis for I_{C_m} constructed from $\preceq$. Fix $\bar{D} \in \bar{\mathcal{D}}_h$. Define a random variable $\xi_{\bar{D}}$ on $\mathcal{S}$ as follows:

$$\xi_{\bar{D}}(\preceq) = \min \{m : h\%G^m_{\preceq} = h\}.$$

Let $\bar{\mathcal{S}} = \{\bar{S} =<\preceq_m\colon\ m \in \mathbb{Z}^+ >\colon\ \forall m \quad \preceq_m \in \mathcal{S}\}$. For $\bar{D} \in \bar{\mathcal{D}}_h$ and $\bar{S} \in \bar{\mathcal{S}}$ we let $m(\bar{S}, \bar{D}) = \min \{m : \xi_{\bar{D}}(\preceq_m) \leq m\}$. Note that both $E(\lambda_{h,\mathcal{S}})$ and $E(\kappa_{h,\mathcal{S}})$ can be computed as:

$$E = \sum_{\bar{D} \in \bar{\mathcal{D}}_h} Pr(\bar{D}) \sum_{\bar{S} \in \bar{\mathcal{S}}} Pr(\bar{S}) m(\bar{S}, \bar{D}). \tag{1}$$

The difference is that in the definition of $E(\kappa_{h,\mathcal{S}})$ we assume the product probability measure on $\bar{\mathcal{S}}$, whereas in the definition of $E(\lambda_{h,\mathcal{S}})$ all nonconstant sequences have probability zero. However, note that for any given $\preceq \in \mathcal{S}$ and any m, the probability $Pr(\preceq_m = \preceq) = Pr(\preceq)$ does not depend on which of these two measures we consider.

We can rewrite sum (1) as follows:

$$E = 1 + \sum_{\bar{D} \in \bar{\mathcal{D}}_h} Pr(\bar{D}) \sum_{\preceq \in \mathcal{S}} \sum_{m \in \mathbb{Z}^+} \sum_{\{\bar{S} \in \bar{\mathcal{S}}\colon \preceq_m = \preceq\}} Pr(\preceq_m = \preceq \ \& \, m(\bar{S}, \bar{D}) > m), \tag{2}$$

which can be written as:

$$E = 1 + \sum_{\bar{D} \in \bar{\mathcal{D}}_h} Pr(\bar{D}) \sum_{\preceq \in \mathcal{S}} Pr(\preceq) \sum_{m \in \mathbb{Z}^+} \sum_{\{\bar{S} \in \bar{\mathcal{S}}\colon \preceq_m = \preceq\}} Pr(m(\bar{S}, \bar{D}) > m | \preceq_m = \preceq). \tag{3}$$

Now note that

$$\begin{aligned} &Pr(m(\bar{S},\bar{D}) > m \mid \preceq_m = \preceq) = \\ &= Pr(\xi_{\bar{D}}(\preceq_1) > 1 \,\&\, \ldots \,\&\, \xi_{\bar{D}}(\preceq_{m-1}) > m-1 \,\&\, \xi_{\bar{D}}(\preceq) > m) \leq Pr(\xi_{\bar{D}}(\preceq) > m). \end{aligned} \tag{4}$$

In the calculation of $E(\lambda_{h,\mathcal{S}})$ only constant sequences have positive probability, and thus the inequality in equation (4) turns into an equality; whereas in the calculation of $E(\kappa_{h,\mathcal{S}})$ the inequality may sometimes be strict. Since $Pr(\xi_{\bar{D}}(\preceq) > m)$ is always either 0 or 1 and does not depend on the rest of the sequence, the result follows. □

2.4 More facts about Gröbner bases

Let $h = a_1 x^{\alpha_1} + \cdots + a_\ell x^{\alpha_\ell} \in F[x_1, \ldots, x_n]$ be a polynomial with all coefficients $a_j \neq 0$, and let $\preceq$ be a fixed term order. The *leading term* of h is the term $a_j x^{\alpha_j}$ such that $x^{\alpha_s} \prec x^{\alpha_j}$ for all $s \in [\ell] \backslash \{j\}$; the corresponding monomial x^{α_j} is called the *leading monomial.* A basis (set of generators) G for an ideal I is a *Gröbner basis for I with respect to $\preceq$* iff for every $f \in I$ there exists $g \in G$ such that the leading term of f is divisible by the leading term of g. It can be shown that for every term order $\preceq$ and every ideal I there exists a unique *reduced* Gröbner basis $G_{\preceq}(I)$ for I with respect to $\preceq$. We will not need the formal definition of when a Gröbner basis is reduced; it suffices to know that it is uniquely determined by I and $\preceq$. We will often write $G_{\preceq}$ instead of $G_{\preceq}(I)$ when I is implied by the context. A monomial x^α is a *standard monomial* for a Gröbner basis G for I with respect to a term order $\preceq$ iff x^α is *not* the leading monomial of any $f \in I$. From this definition we can easily observe the following.

Proposition 6 *Let G be any Gröbner basis, let x^α be a standard monomial for G, and assume that β is such that $\beta(i) \leq \alpha(i)$ for all $i \in [n]$. Then x^β is also a standard monomial for G.*

Given a set $C = \{\bar{x}(t) : t \in [m]\}$ of data inputs and polynomials $h, h_1, \ldots, h_\ell \in F[x_1, \ldots, x_n]$, we say that *$h$ is a linear combination over C of $h_1, \ldots, h_\ell$* if there exist constants $a_1, \ldots, a_\ell \in F$ such that $h(\bar{x}) = a_1 h_1(\bar{x}) + \cdots + h_\ell(\bar{x})$ for all $\bar{x} \in C$. The notions of linear dependence and linear independence over C are defined accordingly. The phrase "over C" will be omitted if C is specified by the context. The following facts can be found in any standard text on Gröbner bases, such as [5].

Lemma 7 *Let C be a set of data inputs and let G be any Gröbner basis for I_C. Then the set of standard monomials for G has cardinality $|C|$.*

Proposition 8 *Let $h \in F[x_1, \ldots, x_n]$ and let G be a Gröbner basis for an ideal I_C with respect to a given term order $\preceq$.*
(i) The set of standard monomials is linearly independent, and the remainder $h\%G$ is a linear combination of standard monomials for G.
(ii) In particular, $h\%G = h$ iff h is a linear combination of standard monomials for G.
(iii) If x^α is the leading monomial of h and x^β is the leading monomial of $h\%G$, then $x^\beta \preceq x^\alpha$, and equality occurs iff x^α is a standard monomial.

Definition 9 *A linear combination of monomials* $dep = a_1x^{\alpha_1} + \cdots + a_\ell x^{\alpha_\ell}$ *will be called a* dependency *if all* $a_w \neq 0$. *We say that a set of data inputs* C removes dependency *dep if there exists* $\bar{x} \in C$ *such that* $dep(\bar{x}) \neq 0$.

Of course, a dependency is the same thing as a nonzero polynomial in $F[x_1, \ldots, x_n]$. However, we will use this word in order to imply that all coefficients are presumed to be nonzero or to draw attention to its removal/nonremoval by a certain data set.

Corollary 10 *Let* C *be a set of data inputs and let* G *be any Gröbner basis for* I_C. *Then*
(i) Any monomial x^α *is a standard monomial for* G *iff* $x^\alpha \% G = x^\alpha$.
(ii) If $h = a_1x^{\alpha_1} + \cdots + a_\ell x^{\alpha_\ell}$ *is a dependency, then* $h\%G = h$ *iff* $x^{\alpha_w}\%G = x^{\alpha_w}$ *for all* $w \in [\ell]$.

Proof: Parts (i) and (ii) are a consequence of Proposition 8(i),(ii) since linear combinations of linearly independent monomials are unique. □

Lemma 11 *Let* x^α *be a monomial in* $F[x_1, \ldots, x_n]$, *let* C *be set of data inputs, let* $\preceq$ *be a term order, and and let* $G = G_{\preceq}(I_C)$.
(i) If x^α *is a standard monomial for* G, *then* C *removes all dependencies in which* x^α *is the leading term.*
(ii) If x^α *is not a standard monomial, then* $x^\alpha - x^\alpha\%G$ *is a dependency that is not removed by* C.

Proof: Point (i) is immediate from the definition of a standard monomial. Point (ii) follows from the fact that $x^\alpha - x^\alpha\%G$ is an element of I. □

Lemma 12 *Let* $C_m = \{\bar{x}(t) : t \in [m]\}$ *be a set of data inputs, let* G *be a Gröbner basis for* $I_{C_{m-1}}$, *let* $dep = a_1x^{\alpha_1} + \cdots + a_\ell x^{\alpha_\ell}$ *be a dependency with leading term* $a_1x^{\alpha_1}$ *that is not removed by* C_{m-1} *but is removed by* C_m. *Then* $\{\bar{x}(m)\}$ *removes the dependency* $x^{\alpha_1} - x^{\alpha_1}\%G$.

Proof: Wlog we may assume that $a_1 = -1$ (in the sense of F, *i.e.,* $a_1 + 1 = 0$). Consider $dep^* = -x^{\alpha_1} + a_2(x^{\alpha_2}\%G) + \cdots + a_\ell(x^{\alpha_\ell}\%G)$. Then $dep - dep^*$ is a sum of elements of $I_{C_{m-1}}$, and since dep is an element of $I_{C_{m-1}}$, so is dep^*. Thus $\{\bar{x}(m)\}$ removes dep^*. Moreover, all monomials of dep^* other than x^{α_1} are standard monomials for G. Thus $dep^* + x^{\alpha_1}$ and $x^{\alpha_1}\%G$ are two linear combinations of standard monomials that take the same values on C_{m-1}. Since standard monomials for G are linearly independent over C_{m-1}, it follows that $dep^* = -(x^{\alpha_1} - x^{\alpha_1}\%G)$, and we conclude that $\{\bar{x}(m)\}$ removes the dependency $x^{\alpha_1} - x^{\alpha_1}\%G$. □

Let $\bar{D} =<< \bar{x}(t), y(t) >: t \in \mathbb{Z}^+ >\in \bar{\mathcal{D}}$ be a sequence of data points, and let $\preceq$ be a fixed term order. The Gröbner basis for $G_{\preceq}(I_{C_m})$ will be denoted by $G^m_{\preceq}$. We say that x^α *becomes a standard monomial at step* m if x^α is a standard monomial for $G^m_{\preceq}$, but x^α

is not a standard monomial for $G_{\preceq}^{m-1}$. The number m such that x^α becomes a standard monomial at step m will be denoted by $m_{\preceq}(\alpha)(\bar{D})$ or simply $m(\alpha)$ if $\bar{D}$, $\preceq$ are implied by the context. By Lemma 7, at each step m at most one x^α becomes a standard monomial at step m.

If we pick $\preceq$ randomly from some set $\mathcal{S}$ of term orders, then $m(\alpha)$ becomes a random variable that also depends on the particular choice of $\preceq$. In order to specify the probability space $\mathcal{S}$ from which $\preceq$ is drawn, we will use the notation $m_{\mathcal{S}}(\alpha)$ in this case.

Lemma 13 *Let $h = a_1 x^{\alpha_1} + \cdots + a_\ell x^{\alpha_\ell}$ be a dependency.*
(i) If $\preceq$ is a term order and $\bar{D} \in \bar{\mathcal{D}}$, then the LS-algorithm with inputs D_m and $\preceq$ returns h iff h is a model of D_m and $m_{\preceq}(\alpha_w)(\bar{D}) \leq m$ for all $w \in [\ell]$.
(ii) Let $\mathcal{S}$ be a set of term orders. Then

$$\min_{\bar{\mathcal{D}}_h} \lambda_{h,\mathcal{S}} = \min_{\bar{\mathcal{D}}} \max\{m_{\mathcal{S}}(\alpha_w) : w \in [\ell]\}$$

and

$$E(\lambda_{h,\mathcal{S}}) = E(\max\{m_{\mathcal{S}}(\alpha_w) : w \in [\ell]\}).$$

Proof: Part (i) follows immediately from Corollary 10(ii). By definition, $m_{\preceq}(\alpha)(\bar{D})$ depends only on the sequence of data inputs for $\bar{D}$ and not on which particular $\bar{\mathcal{D}}_h$ the data sequence belongs to. Thus part (ii) is a consequence of part (i). $\square$

Let us investigate under which conditions $m_{\preceq}(\alpha) = m$ holds. By Lemma 11(ii), for every x^γ that is not a standard monomial for $G_{\preceq}^{m-1}$, the polynomial $x^\gamma - (x^\gamma \% G_{\preceq}^{m-1})$ is a dependency that is not removed by C_{m-1}. We get the following characterization.

Lemma 14 *Let α be a multiexponent, let $\preceq$ be a term order, and let $\bar{D} \in \bar{\mathcal{D}}$. Assume x^α is not a standard monomial for $G_{\preceq}^{m-1}$. Then $m_{\preceq}(\alpha)(\bar{D}) = m$ iff both of the following hold:*
(i) $\bar{x}(m)$ removes the dependency $x^\alpha - (x^\alpha \% G_{\preceq}^{m-1})$.
(ii) $\bar{x}(m)$ does not remove any of the dependencies $x^\gamma - (x^\gamma \% G_{\preceq}^{m-1})$ for $\gamma \prec \alpha$.

Proof: First note that in point (ii) we may restrict our attention to $\gamma \prec \alpha$ such that x^γ is not a standard monomial, because if x^γ is a standard monomial, then $x^\gamma - (x^\gamma \% G_{\preceq}^{m-1})$ is simply the zero polynomial.

By Lemma 11, point (i) is a necessary condition for x^α to become a standard monomial at step m.

Now let us show that points (i) and (ii) together are sufficient conditions for x^α to become a standard monomial at step m. Suppose not. Then $x^\alpha - (x^\alpha \% G_{\preceq}^{m})$ is a dependency that is not removed by $\bar{x}(m)$, and hence it must be different from $x^\alpha - (x^\alpha \% G_{\preceq}^{m-1})$ by point (i). No two different linear combinations of standard monomials for $G_{\preceq}^{m-1}$ can be identical on C_{m-1}, hence $x^\alpha - (x^\alpha \% G_{\preceq}^{m})$ contains a standard monomial x^γ for $G_{\preceq}^{m}$ that is not a standard monomial for $G_{\preceq}^{m-1}$. But since x^α is the leading monomial of $x^\alpha - (x^\alpha \% G_{\preceq}^{m})$, we must have

$\gamma \prec \alpha$. Since we have already seen that point (i) is necessary, $\bar{x}(m)$ removes the dependency $x^\gamma - (x^\gamma \% G_{\preceq}^{m-1})$, which contradicts point (ii).

It remains to show that if x^α becomes a standard monomial at step m, then point (ii) holds. Suppose not, and let $\beta \prec \alpha$ be the $\preceq$-smallest counterexample. Then both point (i) and point (ii) hold for β in the role of α, and thus x^β becomes a standard monomial at step m. Since we already know (from Lemma 7) that at most one monomial becomes a standard monomial at step m, it follows that x^α does not become at standard monomial at this step. $\square$

3 Results

3.1 Bounds for $\min \lambda_{h,\mathcal{S}}$

Definition 15 *Let $K \subseteq [n]$ and let $C = \{\bar{x}(t) : t \in [m]\}$ be a set of data inputs. We say that C* fully resolves *K if for every $f : K \to F$ there exists $t \in m$ such that $x_i(t) = f(i)$ for all $i \in K$. We say that C* weakly resolves *K if for every $f : K \to \{0, 1\}$ there exists $t \in m$ such that $x_i^{|F|-1}(t) = f(i)$ for all $i \in K$ (i.e., $x_i(t) \neq 0$ iff $f(i) = 1$).*

Note that if $F = \mathbb{F}_2$, then C fully resolves K iff C weakly resolves K. If a set of data inputs C fully resolves a set of variables K, then $|C| \geq |F|^{|K|}$; if C weakly resolves a set of variables K, then $|C| \geq 2^{|K|}$.

Lemma 16 *Let $h = a_1 x^{\alpha_1} + \cdots + a_\ell x^{\alpha_\ell} \in F[x_1, \ldots, x_n]$, and let K be such that $supp(h) \subseteq K \subseteq [n]$.*
(i) Suppose that C is a set of data inputs that fully resolves K. If $\preceq$ is a lex order with $x_i \prec x_j$ whenever $i \in K$ and $j \notin K$, then $h \% G_{\preceq}(I_C) = h$.
(ii) Let $C = \{\bar{x}(t) : t \in [m]\}$ be a set of data inputs that fully resolves K and such that $x_i(t) = 0$ for all $i \in [n] \backslash K, t \in [m]$. If G is any Gröbner basis for I_C, then $h \% G = h$.
(iii) Let $C = \{\bar{x}(t) : t \in [m]\}$ be a set of data inputs that weakly resolves K and such that $x_i(t) = 0$ for all $i \in [n] \backslash K, t \in [m]$ and $x_i(t) \in \{0, 1\}$ for all $i \in K, t \in [m]$. If $\max \alpha_w \leq 1$ for all $w \in [\ell]$ and if G is any Gröbner basis for I_C, then $h \% G = h$.

Proof: We will write $F[K]$ as shorthand for $F[\{x_i : i \in K\}]$. For the proof of part (i), note that for every lex order $\preceq$ as in the assumption, every $w \in [\ell]$ and every $x^\beta \prec x^{\alpha_w}$ we have $supp(\beta) \subseteq K$. By Proposition 8(iii), each of the dependencies $x^{\alpha_w} - x^{\alpha_w} \% G_{\preceq}$ is a polynomial in $F[K]$. If C fully resolves K, then no nonzero polynomial in $F[K]$ with $\max \beta < |F|$ for every of its terms x^β can be zero on all points in C. In particular, C removes all dependencies in $F[K]$. It follows that each monomial of h is a standard monomial, and thus $h \% G_{\preceq}(I_C) = h$.

For the proof of part (ii), note that if C is as in the assumptions, then $supp(\beta) \subseteq K$ for any standard monomial x^β for G, and hence $x^{\alpha_w} - x^{\alpha_w} \% G$ is again a polynomial in $F[K]$ for every monomial x^{α_w} of h. Now the same argument as in the proof of part (i) works.

For the proof of part (iii), let C be as in the assumptions, and let $\preceq$ be any term order. First note that since the data inputs take only values from the set $\{0,1\}$, we have $x_i^r = x_i$ for all $r \geq 1$, and it follows that if α is a multiexponent with $\max \alpha > 1$ and β takes the value 1 on all $i \in supp(\alpha)$ while it takes the value 0 outside of $supp(\alpha)$, then $x^\alpha - x^\beta \in I_C$. Similarly, if $supp(\alpha)$ is not a subset of K, then $x^\alpha \in I_C$. Thus such x^α are the leading terms of elements of I_C and cannot be standard monomials. Since $|C| = 2^{|K|}$, it follows from Lemma 7 that the set of standard monomials for any Gröbner basis for I_C is the set $SM = \{x^\alpha : supp(\alpha) \subseteq K \ \& \ \max \alpha \leq 1\}$. Since the assumption of part (iii) tells us that h is a linear combination of terms in SM, we conclude that $h\%G = h$. $\square$

Corollary 17 *Let $h \in F[x_1, \ldots, x_n]$ and let $\mathcal{S}$ be any set of term orders. Then* $\min \lambda_{h,\mathcal{S}} \leq |F|^{|supp(h)|}$.

Proof: If C is as in the assumption of part (ii) of Lemma 16, then $|C| = |F|^{|K|}$, for $K = supp(h)$ and we can find $\bar{D} \in \bar{\mathcal{D}}$ with $C_m = C$ for $m = |C| = |F|^{|supp(h)|}$. $\square$

Lemma 18 *Let $h = x^\alpha$ be a monomial, and let $K = supp(\alpha)$. Assume moreover that $\alpha(i) = |F| - 1$ for all $i \in supp(\alpha)$. If C is any set of data inputs and G is any Gröbner basis for I_C with $h\%G = h$, then C fully resolves K.*

Proof: Let x^α be as in the assumption. For each $f \in F^K$ let $dep_f = \prod_{i \in K} \prod_{a \in F \setminus \{f(i)\}} (x_i - a)$. Then dep_f is a dependency with leading term x^α. This dependency is removed by C iff there is a point $\bar{x}(t) \in C$ with $x_i(t) = f(i)$ for all $i \in K$. Thus if $x^\alpha\%G = x^\alpha$, that is, if x^α is a standard monomial of G, then C must contain, for every $f \in F^K$, a point that agrees with f on K. This shows that C fully resolves K. $\square$

Note that if $F = \mathbb{F}_2$, then the second assumption of Lemma 18 is always satisfied. If $F \neq \mathbb{F}_2$, then the set C as in the assumption of Lemma 16(iii) does not fully resolve K, and it follows that this second assumption cannot be dropped.

For h as in Lemma 18, the lower bound of Corollary 17 is sharp.

Corollary 19 *Let $h \in F[x_1, \ldots, x_n]$, and let $\mathcal{S}$ be a set of term orders. If h contains a monomial x^α such that $\alpha(i) = |F| - 1$ for all $i \in supp(\alpha)$, then* $\min \lambda_{h,\mathcal{S}} = |F|^{|supp(\alpha)|}$.

Proof: This follows from Lemma 18 and Lemma 13(ii). $\square$

3.2 Data sets with high resolution

The previous section indicates how crucial it is for the workings of the LS-algorithm that the set of data inputs resolve the support set of the function h_{true}. This observation motivates the following definition.

Definition 20 *(i) Let $K \subseteq [n]$. We define a new random variable ξ on $\bar{\mathcal{D}}$ as follows:*

$$\xi_K(\bar{D}) = \min\{m \in \mathbb{Z}^+ : C_m \textit{ fully resolves } K\}.$$

(ii) Let k be a positive integer. We say that a set C of data inputs has resolution k *if C fully resolves every subset of $[n]$ of size k. We define random variables ρ_k on $\bar{\mathcal{D}}$ as*

$$\rho_k(\bar{D}) = \min\{m \in \mathbb{Z}^+ : C_m \textit{ has resolution } k\}.$$

Note that if a data set has resolution k, then it also has resolution j for all $1 \leq j < k$.

Lemma 21 *Let $K \subseteq [n]$, and let $|K| = k$. Then*

$$|F|^k\, k \ln|F| < E(\xi_K) < |F|^k\,(1 + k \ln|F|).$$

Proof: Let $\bar{D} \in \bar{\mathcal{D}}$. We define a random variable ξ_j as follows:
$\xi_0 = 1$.
For $|F|^k > j > 0$ we let $t_j = \xi_0 + \cdots + \xi_{j-1}$ and define
$\xi_j = \min\{s : \forall t \leq t_j\ \bar{x}(t_j + s) \restriction K \neq \bar{x}(t) \restriction K\}$.
Note that ξ_j is well defined except on a set of measure zero. One can think of ξ_j as measuring the time it takes to sample a new possible behavior of $\bar{x}(t) \restriction K$, and of ξ as the total time it takes until all possible behaviors of $\bar{x}(t) \restriction K$ have been sampled. Thus

$$\xi = \sum_{j=0}^{|F|^k-1} \xi_j, \qquad \text{and it follows that} \qquad E(\xi) = \sum_{j=0}^{|F|^k-1} E(\xi_j).$$

Note that ξ_j has a geometric distribution with success probability $\frac{|F|^k-j}{|F|^k}$. Thus $E(\xi_j) = \frac{|F|^k}{|F|^k-j}$, and we get

$$E(\xi) = \sum_{j=0}^{|F|^k-1} E(\xi_j) = \sum_{j=0}^{|F|^k-1} \frac{|F|^k}{|F|^k - j} = |F|^k \sum_{j=0}^{|F|^k-1} \frac{1}{|F|^k - j} = |F|^k \sum_{\ell=1}^{|F|^k} \frac{1}{\ell}.$$

Since

$$k \ln|F| = \ln|F|^k < \sum_{\ell=1}^{|F|^k} \frac{1}{\ell} < 1 + \ln|F|^k = 1 + k \ln|F|,$$

the lemma follows. □

The following result can be found as Theorem 4.9 in [15] or in [16].

Lemma 22 *Let k be a positive integer, let $\bar{D} \in \bar{\mathcal{D}}$ be randomly chosen, and let $c \geq 1$ be a constant. If $m \geq |F|^k(k(\ln n + \ln|F|) + \ln c)$, then the probability that C_m does* not *have resolution k is less than $\frac{1}{c}$.*

Corollary 23 $\min \rho_k \leq |F|^k k(\ln n + \ln |F|)$.

Proof: This follows from Lemma 22 by letting $c = 1$. $\square$

It turns out that $E(\rho_k)$ is very close to $\min \rho_k$.

Lemma 24

$$E(\rho_k) < |F|^k(k(\ln n + \ln |F|) + \frac{1}{1 - e^{-1}}).$$

Proof: For a nonnegative integer j, let $\ell_j = |F|^k(k(\ln n + \ln |F|) + j)$. We get the following estimate for $E(\rho_k)$:

$$E(\rho_k) \leq |F|^k(k(\ln n + \ln |F|) + \sum_{j=0}^{\infty}(j+1)Pr(\ell_j \leq \rho_k < \ell_{j+1})). \tag{5}$$

Note that by changing its right-hand side to an equivalent form, equation (5) can be written as follows:

$$E(\rho_k) \leq |F|^k(k(\ln n + \ln |F|) + \sum_{j=0}^{\infty} Pr(\ell_j \leq \rho_k)). \tag{6}$$

Applying Lemma 22 to the right-hand side of equation (6) we obtain:

$$E(\rho_k) \leq |F|^k(k(\ln n + \ln |F|) + \sum_{j=0}^{\infty} Pr(\ell_j \leq \rho_k)) < |F|^k(k(\ln n + \ln |F|) + \sum_{j=0}^{\infty} \frac{1}{e^j}), \tag{7}$$

and Lemma 24 follows. $\square$

Recall that $\mathcal{L}$ denotes the set of all lex orders.

Theorem 25 *Let $h \in F[x_1, \ldots, x_n]$ be such that $|supp(h)| \leq k$. Then*
(i)

$$E(\nu_{h,\mathcal{L}}) < |F|^k\,(1 + k \ln |F|).$$

(ii) If $h = x^\alpha$ is a monomial such that $\alpha(i) = |F| - 1$ for all $i \in supp(\alpha)$ and if $|supp(\alpha)| = j$, then

$$|F|^j\,(j \ln |F|) < E(\nu_{h,\mathcal{L}}).$$

(iii)

$$E(\kappa_{h,\mathcal{L}}) < |F|^k\,(1 + k \ln |F|) + \binom{n}{k}.$$

(iv) Suppose $\bar{D} \in \bar{\mathcal{D}}_h$ and m is on the order of $\omega(k|F|^k)$. If the data set D_m is analyzed by running the LS-algorithm on the order of $\omega(n^k)$ times with randomly and independently chosen lex orders, then the probability that the algorithm returns h at least once will approach one.

Proof: Part (i) follows immediately from Lemmas 21 and 16(i). Part (ii) follows from Lemmas 21 and 18.

For the proof of point (iii), let $K = supp(h)$. Recall that in a Type 2 Analysis, the LS-algorithm is run on D_m with input parameter $\preceq_m$, and $\kappa_{h,\mathcal{L}}$ is the first m for which this procedure returns h when $\preceq_m$ is randomly drawn from $\mathcal{L}$. By Lemma 16(i), $\kappa_{h,\mathcal{L}}$ is less than or equal to the waiting time for the first success, where a "success" occurs at step m if C_m fully resolves K and $\preceq_m$ is such that $x_i \prec_m x_j$ whenever $i \in K$ and $j \notin K$. For $m > \xi_K$, the conditional probability of success is equal to $\frac{1}{\binom{n}{k}}$. Thus the time we need to wait for the first success from the moment that C_m fully resolves K (that is, from $\xi_K(\bar{D})$) is bounded by a random variable ζ with geometric distribution and success probability in a single trial of $\frac{1}{\binom{n}{k}}$. Thus $\kappa_{h,\mathcal{L}} \leq E(\xi_K) + E(\zeta)$. Since $E(\zeta) = \binom{n}{k}$, point (iii) follows from Lemma 21.

The essentially same argument can be used to derive part (iv). □

3.3 Random graded orders

Recall that a term order $\preceq$ is called *graded* if $\sum \alpha < \sum \beta$ implies $\alpha \prec \beta$, and that $\mathcal{G}$ denotes the set of all graded term orders with uniform distribution. In this section we derive bounds for $E(\lambda_{h,\mathcal{G}})$ and $E(\nu_{h,\mathcal{G}})$.

Theorem 26 *Let $\preceq$ be a graded term order, let $h \in F[x_1, \ldots, x_n]$ be a polynomial, let x^α be the leading monomial of h, and let $k = \sum \alpha$. Then*

$$E(\lambda_{h,\{\preceq\}}) \leq |F|^k \sum_{\ell=0}^{k} \binom{n}{\ell} (\min\{|F-1|, k+1-\ell\})^\ell.$$

In particular,

$$E(\lambda_{h,\mathcal{G}}) \leq |F|^k \sum_{\ell=0}^{k} \binom{n}{\ell} (\min\{|F-1|, k+1-\ell\})^\ell.$$

Corollary 27 *Let h be a polynomial with leading term x^α and let $k = \sum \alpha$. If $n >> |F|$, then $E(\lambda_{h,\mathcal{G}})$ is $O(|F|^k n^k)$.*

Proof: If n is larger relative to $|F|$, then the sum in Theorem 26 is dominated by its last term $|F|^k \binom{n}{k}$, which is $O(|F|^k n^k)$. □

In order to prove Theorem 26 we need some preliminaries.

Lemma 28 *Let $\preceq$ be any term order. Suppose $\bar{D}$ is a randomly chosen data sequence from $\bar{\mathcal{D}}$ such that x^α is not a standard monomial for $G_{\preceq}^{m-1} = G_{\preceq}(I_{C_{m-1}})$, and let $\bar{v} \in F^n$. Then*

$$Pr(\{\bar{x}(m)\} \text{ removes dependency } x^\alpha - (x^\alpha \% G_{\preceq}^{m-1}) | \forall i \in [n] \backslash supp(\alpha)\ x_i = v_i) \geq |F|^{-|supp(\alpha)|}.$$

Proof: Let $dep_{\bar{v}}$ be the polynomial obtained by replacing any occurrence of x_i for $i \notin supp(\alpha)$ in the polynomial $x^\alpha - (x^\alpha \% G_{\preceq}^{m-1})$ by the corresponding value v_i. Then $dep_{\bar{v}}$ becomes a nonzero polynomial in $F[\{x_i : i \in supp(\alpha)\}]$, and hence there exists at least one vector $\bar{z} \in F^{supp(\alpha)}$ such that $dep_v(\bar{z}) \neq 0$. Since our probability distribution on the input vectors was assumed uniform, Lemma 28 follows. $\square$

Recall that $m_{\preceq}(\alpha)$ denotes the minimum m such that x^α becomes a standard monomial with respect to the Gröbner basis $G_{\preceq}(I_{C_m})$.

Let $\preceq$ be any term order, and let A be an initial segment of the set of multiexponents ordered by $\preceq$; *i.e.,* such that if $\alpha \in A$ and $\beta \preceq \alpha$, then also $\beta \in A$. We define a random variable $\eta_{A,\preceq}$ on $\bar{\mathcal{D}}$ as follows:

$$\eta_{A,\preceq} = \max\{m_{\preceq}(\beta) : \beta \in A\}.$$

Lemma 29 *Let $\preceq$ be any term order, and let A be an initial segment of set of multiexponents ordered by $\preceq$ such that $|supp(\alpha)| \leq k$ for all $\alpha \in A$. Then*

$$E(\eta_{A,\preceq}) \leq |A| \cdot |F|^k.$$

Proof: Let us look at a sequence $\bar{D}$ of experiments in the following way: Consider the m-th experiment a "success" if $\alpha(m)$ is the $\preceq$-smallest α such that x^α was not a standard monomial for $G_{\preceq}^{m-1}$. For any positive integer M, let σ_M be the waiting time for the M-th success. Since A is an initial segment we have $E(\eta_{A,\preceq}) \leq E(\sigma_{|A|})$.

Moreover, since the support of any multiexponent in A has at most k elements, it follows from Lemma 28 and Lemma 14 that the success probability in each experiment is at least $|F|^{-k}$. By the well-known formula for the expected waiting time for the M-th success we get $E(\sigma_{|A|}) \leq |A| \cdot |F|^k$, and Lemma 29 follows. $\square$

Proof of Theorem 26: Let $\preceq$ be a graded term order, let h, α, k be as in the assumption, and let $A = \{\beta : \beta \preceq \alpha\}$. Then A is an initial segment of $\preceq$ that contains all monomials of h. If $\beta \preceq \alpha$ and $\preceq$ is graded, then $|supp(\beta)| \leq \sum \beta \leq \sum \alpha$. Moreover, if $|supp(\beta)| = \ell$, then for every $i \in supp(\beta)$ we have $1 \leq \beta(i) \leq |F-1|$ by the definition of $supp(\beta)$, and we also must have $\beta(i) \leq 1 - |supp(\beta)| + \sum \alpha$ because $\sum \beta \leq \sum \alpha$. Thus

$$|A| \leq \sum_{\ell=0}^{\sum \alpha} \binom{n}{\ell} (\min\{|F-1|, 1 - \ell + \sum \alpha\})^\ell,$$

and it follows from Lemma 29 that the right-hand side of the equations in Theorem 26 is an upper bound for the expected number of data points needed for all monomials of h to become standard monomials. Now the result follows from Lemma 13(ii). $\square$

Let $\pi : [n] \to [n]$ be a permutation. We can naturally extend π to a permutation of the set of all terms orders defined by:

$$\alpha \ \pi(\preceq) \ \beta \ \text{ iff } \ \alpha \circ \pi \preceq \beta \circ \pi.$$

Similarly, we can extend π to a permutation of all polynomials in $F[x_1, \ldots, x_n]$ defined by:

$$\pi(a_1 x^{\alpha_1} + \cdots + a_\ell x^{\alpha_\ell}) = a_1 x^{\alpha_1 \circ \pi} + \cdots + a_\ell x^{\alpha_\ell \circ \pi},$$

the set of all input vectors:

$$\pi(\bar{x})_i = x_{\pi(i)},$$

and also to a permutation of data sequences:

$$\pi(\{< \bar{x}(t), y(t) >: \ t \in \mathbb{Z}^+\}) = \{< \pi(\bar{x}(t)), y(t) >: \ t \in \mathbb{Z}^+\}.$$

Lemma 30 *Let $\preceq$ be a term order, let π be a permutation of $[n]$, let $\bar{D}$ a data sequence, and let α be a multiexponent. Then*

$$m_{\{\preceq\}}(\alpha)(\bar{D}) = m_{\{\pi(\preceq)\}}(\alpha \circ \pi)(\pi(\bar{D})).$$

Proof: This result should be intuitively clear, because the simultaneous application of π to everything in sight amounts just to a consistent relabeling of the variables in all relevant objects for the calculation of $m(\alpha)$. For a formal proof, one can use Lemma 14 to show that x^α is a standard monomial for $G_{\preceq}(I_{C_m})$ iff $x^{\alpha \circ \pi}$ is a standard monomial for $G_{\pi(\preceq)}(I_{\pi(C_m)})$. This is rather tedious but straightforward, and we will omit details. $\square$

We say that a set $\mathcal{S}$ of term orders is *invariant under permutations of the variables* if $\pi(\preceq) \in \mathcal{S}$ whenever $\preceq \in \mathcal{S}$ and π is a permutation of $[n]$. Note that both the set $\mathcal{G}$ of all graded term orders and the set $\mathcal{L}$ of all lex orders are invariant under permutations of the variables.

Let us introduce a partial order relation on the set of all multiexponents as follows: We write $\beta \trianglelefteq \alpha$ iff there exists a permutation $\pi : [n] \to [n]$ such that $\pi(\beta)(i) \leq \pi(\alpha)((i))$ (equivalently, $\beta(i) \leq \alpha(\pi(i))$) for all $i \in [n]$. For example, if $\max \alpha = \max \beta = 1$, then $\beta \trianglelefteq \alpha$ iff $|supp(\beta)| \leq |supp(\alpha)|$. In general, $\beta \trianglelefteq \alpha$ iff for $|\{i : \ \beta(i) > j\}| \leq |\{i : \ \beta(i) > j\}|$ for all nonnegative integers j. We will write $\alpha \sim \beta$ and say that α *and* β *are permutation-equivalent* if $\beta \trianglelefteq \alpha$ and $\alpha \trianglelefteq \beta$. Note that $\alpha \sim \beta$ iff $\beta = \pi(\alpha)$ for some permutation π.

Lemma 31 *Let α be a multiexponent and let $\mathcal{S}$ be a set of term orders with the uniform probability distribution that is closed under permutations of variables and endowed with the uniform probability distribution. Then $E(m_{\mathcal{S}}(\alpha)) \geq \frac{1}{2}|\{\beta : \ \beta \trianglelefteq \alpha\}|$.*

Proof: Note that

$$E(m_{\mathcal{S}}(\alpha)) = \sum_{m>0} m \, Pr(|\{\beta : \ m_{\mathcal{S}}(\beta) \leq m_{\mathcal{S}}(\alpha)\}| = m) = \sum_{\beta} Pr(m_{\mathcal{S}}(\beta) \leq m_{\mathcal{S}}(\alpha)). \quad (8)$$

The second sum in equation (8) is taken over the set of all multiexponents, and the probability is calculated in $\mathcal{S}$. Since $\mathcal{S}$ is permutation-invariant, we must have $Pr(m_{\mathcal{S}}(\beta) \leq$

$m_{\mathcal{S}}(\alpha)) = \frac{1}{2}$ whenever $\alpha \sim \beta$ and $\alpha \neq \beta$. To see this, let π be a permutation of $[n]$ such that $\pi(\alpha) = \beta$. By Lemma 30, $m_{\{\preceq\}}(\alpha)(\bar{D}) > m_{\{\preceq\}}(\beta)(\bar{D})$ iff $m_{\{\pi(\preceq)\}}(\pi(\alpha))(\pi(\bar{D})) < m_{\{\pi(\preceq)\}}(\pi(\beta))(\pi(\bar{D}))$. Since the probability distribution on $\bar{\mathcal{D}}$ was assumed uniform, the assumptions on $\mathcal{S}$ imply that $Pr(m_{\mathcal{S}}(\alpha) < m_{\mathcal{S}}(\beta)) = Pr(m_{\mathcal{S}}(\alpha) > m_{\mathcal{S}}(\beta)) = \frac{1}{2}$.

Similarly, when $\beta \triangleleft \alpha$, then by Proposition 6 we must have $Pr(m_{\mathcal{S}}(\beta) \leq m_{\mathcal{S}}(\alpha)) > \frac{1}{2}$. The lemma follows by ignoring the contributions of all other β to the second sum. $\square$

Theorem 32 *Let $\mathcal{S}$ be a set of term orders that is closed under permutations, with uniform probability distribution. Let $h \in F[x_1, \ldots, x_n]$ and let x^α be the leading monomial of h. Then*

$$E(\lambda_{h,\mathcal{S}}) \geq \frac{1}{2}|\{\beta : \beta \trianglelefteq \alpha\}|.$$

In particular,

$$E(\lambda_{h,\mathcal{S}}) \geq \frac{1}{2} \sum_{k=0}^{|supp(\alpha)|} \binom{n}{k}.$$

Proof: The first inequality follows from Corollary 31 since $\mathcal{S}$ is closed under permutations, and $\lambda_{h,\mathcal{S}} \geq m_{\mathcal{S}}(\alpha)$ by Lemma 13(i).

The second inequality follows from the first one and the fact that if $|supp(\beta)| \leq |supp(\alpha)|$ and $\max \beta \leq 1$, then $\beta \trianglelefteq \alpha$. $\square$

Since the set $\mathcal{G}$ of graded term orders is closed under permutations, we get the following:

Corollary 33 *If $h \in F[x_1, \ldots, x_n]$ has leading monomial x^α with $k = |supp(\alpha)|$, then $E(\lambda_{h,\mathcal{G}}) = \Omega(n^k)$. If $F = \mathbb{F}_2$ and k is fixed, then $E(\lambda_{h,\mathcal{G}}) = \Theta(n^k)$.*

Proof: The first part follows from the fact that for small k the sum in Theorem 32 is of order $\Omega(n^k)$. The second part follows from the first part and Corollary 27. $\square$

Note that if $F = \mathbb{F}_2$, then the second part of Theorem 32 is the best possible estimate that one can derive from the first part. For other fields and multiexponents that take larger values than one, the estimate can be slightly improved. However, as Corollary 27 shows, even for larger F the growth of $E(\lambda_{h,\mathcal{G}})$ will still be bounded by a polynomial in n, and we will not attempt to derive sharper bounds for the general case here.

The proof of Theorem 32 heavily relies on the fact that we pick $\preceq$ independently of $\bar{D}$. What if we can optimize $\preceq$ for a given data sequence $\bar{D}$? We will show that if F and α are fixed and $n \to \infty$, then we still get a polynomial lower bound.

Theorem 34 *Let $h \in F[x_1, \ldots, x_n]$ and let x^α be the leading monomial of h. Then for sufficiently large n we have*

$$E(\nu_{h,\mathcal{G}}) \geq \frac{1}{4}\binom{n}{(\sum \alpha) - 1}.$$

Proof: Let $k = (\sum \alpha) - 1$. We will be interested only in the case where $k > 0$. For the purpose of this proof, an *antichain* will be a collection of pairwise disjoint sets of cardinality k each. We need a lemma.

Lemma 35 *Let Q be the set of all subsets of $[n]$ of size k and suppose $J \subset Q$. Let $z = \frac{|J|}{|Q|}$. Then there exists an antichain A such that $|A \cap J| \geq z \cdot \lfloor \frac{n}{k} \rfloor$.*

Proof: Let P be the set of all pairs $< A, q >$, where A is an antichain of size $\lfloor \frac{n}{k} \rfloor$ and $q \in Q$. Let $\mathcal{A}$ be the collection of all antichains of this maximal size, and let N be the number of such antichains that contain a fixed $q \in Q$ (this number is independent of q). Then

$$|P| = |\mathcal{A}| \cdot \lfloor \frac{n}{k} \rfloor = |Q| \cdot N.$$

Let $P_J = \{< A, q > \in P : q \in J\}$. Then, on the one hand,

$$|P_J| = |J| \cdot N = z \cdot |Q| \cdot N = z \cdot |P|.$$

On the other hand,

$$|P| = \sum_{A \in \mathcal{A}} |A \cap J|,$$

and it follows that for some $A \in \mathcal{A}$ we must have $|A \cap J| \geq z \cdot |A| \geq z \cdot \lfloor \frac{n}{k} \rfloor$. $\square$

Now let us derive the lower bound. Fix a graded term order $\preceq^*$, and let $B = \{\gamma : |supp(\gamma)| = k \ \& \ \max \gamma = 1\}$. Then $|B| = \binom{n}{k}$. Let $m \leq \frac{1}{2}\binom{n}{k}$, and let C_{m-1} be a set of data inputs of size at most m. Let $G = G_{\preceq^*}(I_{C_{m-1}})$, and let $J \subseteq B$ be the set of all $\gamma \in B$ such that x^γ is not a standard monomial for G. Then $|J| \geq \frac{|B|}{2}$, and by Lemma 35 we find a set $A \subset J$ of cardinality $\geq 0.5 \cdot \lfloor \frac{n}{k} \rfloor$ such that the supports of the multiexponents in A are pairwise disjoint. Fix such A. By Lemma 28, for each $\gamma \in A$, the conditional probability that dependency $x^\gamma - (x^\gamma \% G)$ is removed by a randomly chosen new data point $\bar{x}(m)$ *given* any arbitrary values for $\bar{x}(m)$ on the set $[n] \backslash supp(\gamma)$ is at least $|F|^{-k}$. Since these are conditional probabilities, we are allowed to multiply, and it follows that the probability Pnr that *none* of the dependencies $x^\gamma - (x^\gamma \% G)$ for $\gamma \in A$ is removed by $\{\bar{x}(m)\}$ can be estimated as:

$$Pnr \leq (1 - |F|^{-k})^{0.5 \cdot \lfloor \frac{n}{k} \rfloor} < e^{-zn}, \tag{9}$$

where z can be chosen arbitrarily close to $\frac{1}{2k|F|^k}$ as long as n is sufficiently large.

Note that since $\preceq^*$ was assumed to be graded, for all γ in B and for all monomials x^β that occur in $x^\gamma - (x^\gamma \% G)$ we have $\sum \beta \leq k$, and the choice of k implies that $x^\beta \prec x^\alpha$ for all such β and *every* graded term order $\preceq$. Of course, if $\preceq$ is different from $\preceq^*$, then x^γ may no longer be the leading monomial of $x^\gamma - (x^\gamma \% G)$, but we have just shown that the leading monomial x^β (with respect to $\preceq$) of $x^\gamma - (x^\gamma \% G)$ will satisfy $x^\beta \prec x^\alpha$. Thus by Lemma 12, if $\{\bar{x}(m)\}$ removes dependency $x^\gamma - (x^\gamma \% G)$, then $\{\bar{x}(m)\}$ will also remove the dependency

$x^\beta - (x^\beta \% G_{\preceq}(I_{C_{m-1}}))$ for some $\beta \prec \alpha$. By Lemma 11 and inequality (9) it follows that given an arbitrary set of data inputs C_{m-1} of size at most $m-1$ and for randomly chosen $\bar{x}(m)$, the probability that there exists *any* graded order $\preceq$ such that $m_{\{\preceq\}}(\alpha) = m$ is less than e^{-zn} for z as above.

By the definition of ν and Lemma 13(i), the preceding paragraph shows that if $m = \frac{1}{2}\binom{n}{k}$, then

$$Pr(\nu_{h,\mathcal{G}} \leq m) < me^{-zn} < n^k e^{-zn}, \tag{10}$$

where z is a positive constant. Clearly, the right-hand side of inequality (10) approaches 0 as $n \to \infty$; in particular, for sufficiently large n we will have

$$Pr(\nu_{h,\mathcal{G}} \leq \frac{1}{2}\binom{n}{k}) < \frac{1}{2}.$$

Now Theorem 34 follows from the choice of k. $\square$

Corollary 36 *Let $h \in F[x_1, \ldots, x_n]$, let x^α be the leading monomial of h, and let $k = \sum \alpha$. Then $E(\nu_{h,\mathcal{G}})$ is $\Omega(n^{k-1})$.*

3.4 $E(\lambda)$ for random lex orders

Recall that each lex order is given by a *variable order* $x_{\pi(1)} \succeq \ldots \succeq x_{\pi(n)}$, where $\pi : [n] \to [n]$ is a permutation. The lex order $\preceq_\pi$ is then defined as follows: If $\alpha \neq \beta$, let j_d be the smallest $j \in [n]$ such that $\alpha(\pi(j)) \neq \beta(\pi(j))$. Then

$$\alpha \preceq_\pi \beta \leftrightarrow (\alpha = \beta \vee \alpha(\pi(j_d)) < \beta(\pi(j_d))).$$

Thus randomly picking an element $\preceq$ from $\mathcal{L}$ amounts to randomly picking the permutation π of $[n]$ for which $\preceq = \preceq_\pi$.

Throughout this section, let r be a constant such that $0 < r < 1$. The next proposition follows immediately from the definition of a lex order:

Proposition 37 *If $\min\{\pi(i) : i \in supp(\alpha)\} < (1-r)n$ and β is such that $\pi(i) \geq (1-r)n$ for all $i \in supp(\beta)$, then $\beta \prec_\pi \alpha$.*

Now let π be a fixed permutation of $[n]$. For any positive integer t, let $V^+(t) = \{i \in [n] : \pi(i) \geq (1-r)n \ \& \ x_i(t) \neq 0\}$. For every positive integer m, let OLD_m be the event that $V^+(m)$ is contained in $V^+(t)$ for some $t < m$, and let NEW_m be the complement of OLD_m.

Lemma 38 *If $\min\{\pi(i) : i \in supp(\alpha)\} < (1-r)n$ and NEW_m occurs, then $m_{\preceq_\pi}(\alpha) \neq m$.*

Proof: Let β be any multiexponent such that $supp(\beta) = V^+(m)$, and let $G^\ell = G_{\preceq_\pi}(I_{C_\ell})$ for $\ell \in \{m-1, m\}$. Then $\bar{x}(m)^\beta \neq 0$. Thus $x^\beta \notin I_{C_m}$, and hence $x^\beta \% G^m$ is a nonzero linear combination of standard monomials x^γ for G^m such that $\gamma \preceq_\pi \beta$. On the other hand, if NEW_m occurs, then for all $t < m$ there is at least one $i \in supp(\beta)$ with $x_i(t) = 0$, and hence $\bar{x}(t)^\beta = 0$. The latter implies that $x^\beta \in I_{C_{m-1}}$, hence $x^\beta \% G^{m-1} = 0$ and x^β cannot be written as a nonzero linear combination of standard monomials x^γ for G^{m-1}. It follows that $\alpha(m) = \gamma$ for some γ with $\gamma \preceq_\pi \beta$. By Proposition 37, we have $\gamma \preceq_\pi \beta \prec_\pi \alpha$, and thus $m(\alpha) \neq m$ by Lemma 14. □

Lemma 39 *Let m be a positive integer. Then*

$$Pr(OLD_m) \leq (m-1)e^{-\frac{\lfloor rn \rfloor (|F|-1)}{|F|^2}}.$$

Proof: We have:

$$\begin{aligned} Pr(OLD_m) &\leq \sum_{t=1}^{m-1} Pr(V^+(m) \subseteq V^+(t)) \\ &\leq (m-1)(1 - \frac{1}{|F|}\frac{|F|-1}{|F|})^{\lfloor rn \rfloor} \\ &\leq (m-1)e^{-\frac{\lfloor rn \rfloor (|F|-1)}{|F|^2}}. \end{aligned} \tag{11}$$

□

Theorem 40 *Let h be any nonconstant polynomial, let $1 > q > 0$, and let $r = \frac{q}{2}$. Then for sufficiently large n:*

$$Pr(\lambda_{h,\mathcal{L}} \leq \sqrt{q} e^{\frac{\lfloor rn \rfloor |F|-1}{2|F|^2}}) \leq q.$$

Proof: Let x^α be a nonconstant monomial in h and let $m = \sqrt{q} e^{\frac{\lfloor rn \rfloor |F|-1}{2|F|^2}}$. It follows from inequality (11) that

$$Pr(\exists \ell \leq m \; OLD_\ell) \leq \sum_{\ell=1}^{m} (\ell-1)e^{-\frac{\lfloor rn \rfloor (|F|-1)}{|F|^2}} < \frac{m^2}{2} e^{-\frac{\lfloor rn \rfloor (|F|-1)}{|F|^2}} = \frac{q}{2}. \tag{12}$$

Let A denote the event that our randomly chosen lex order $\preceq_\pi$ is such that $\min\{\pi(i) : i \in supp(\alpha)\} \geq (1 - \frac{q}{2})n$, and let B be the complement of A. Note that $Pr(A) \leq \frac{q}{2} + o(1)$. Moreover, Lemma 38 implies that $Pr(m_{\preceq_\pi}(\alpha) > m|B) < \frac{q}{2}$. Thus $Pr(m_{\preceq_\pi}(\alpha) > m) \leq Pr(m_{\preceq_\pi}(\alpha) > m|B)(1 - Pr(A)) + Pr(A) \leq \frac{q}{2}(1 - Pr(A)) + \frac{q}{2} + o(1)$. Now the theorem follows from Lemma 13(ii). □

Corollary 41 *There exists a constant $c > 1$ such that $E(\lambda_{h,\mathcal{L}})$ is $\Omega(c^n)$ for every nonconstant $h \in F[x_1, \ldots, x_n]$. In particular, this is true for $c = e^{\frac{|F|-1}{4|F|^2}}$.*

Proof: Let $q = \frac{1}{2}$ in Theorem 40. □

3.5 A modification of the LS-algorithm

In the previous section we found that when run with a randomly chosen lex order, the LS-algorithm is expected to need exponentially many data points before it converges to the correct solution. In contrast, by Theorem 25(i), very few data points suffice if the lex order is optimally chosen. Here we will explore a modification of the LS-algorithm that tries to first find a near optimal lex order for running the algorithm.

Any lex order is uniquely determined by its variable order $x_{\pi(1)} \succ x_{\pi(2)} \succ \cdots \succ x_{\pi(n)}$. The idea for choosing a near optimal lex order is to choose a variable order in such a way that the variables that h_{true} is likely to depend on come last. If the data are concentration levels of chemical species in a biochemical network, one can try to use prior biological knowledge to identify those among the species that are likely candidates for regulating the concentration level x_i, and rank them last when running the LS-algorithm for finding the regulatory function h_i of x_i. This approach has been tried in [17] and [21]. Alternatively, one could base the choice of variable order directly on the given data set. Here we will investigate one algorithm for doing the latter. A version of this algorithm has been implemented and successfully tested on some data sets [11].

Definition 42 *Let $D = \{< \bar{x}(t), y(t) >: t \in m\}$ be a data set. A subset $L \subseteq [n]$ is called a* dependency set for D *if*

$$\forall t_1, t_2 \in m \ (\bar{x}(t_1) \restriction L = \bar{x}(t_2) \restriction L \rightarrow y(t_1) = y(t_2)).$$

In other words, $L \subseteq [n]$ is a dependency set for D iff there exists a model h for D such that $supp(h) \subseteq L$.

The LS-algorithm with preprocessing

1. Find a dependency set L for D of minimum size.
2. Choose a variable order that puts the elements of L last.
3. Run the LS-algorithm on the lex order associated with the variable order found in Step 2.

The above description does not entirely specify the algorithm and leaves room for further improvement. For example, prior biological knowledge can be incorporated in step 2 to choose the most promising among all lex orders permitted by it. But the description given here will allow us to prove an estimate of the algorithm's performance on random data sets. This estimate will be valid for any specific implementation of the algorithm.

Suppose $h \in F[x_1, \ldots, x_n]$ and let $K = supp(h)$. If D is a data set for which h is a model, then K is a dependency set for h, but it does not need to be the case that K is a dependency set of minimum cardinality. Even if K is of the smallest possible size, there may be another dependency set L of the same size. This cannot happen though if D has resolution $2|K|$.

Lemma 43 *Let $D = \{< \bar{x}(t), y(t) >: t \in [m]\}$ be a data set, let K be a dependency set for D of size k, and assume that K is minimal in the sense that no proper subset of K is a dependency set for D. Assume D has resolution $2k - \ell$. Then $K = L$ for every dependency set L for D with $|L| \leq k$ and $|L \cap K| \geq \ell$.*

Proof: Let D, K be as in the assumptions. Assume towards a contradiction that there exists a dependency set L for D such that $|L| \leq k$, $|K \cap L| \geq \ell$, and $K \backslash L \neq \emptyset$. Choose $x_i \in K \backslash L$. By minimality of K, there are $\bar{x}(t_1), \bar{x}(t_2) \in K$ such that $x_j(t_1) = x_j(t_2)$ for all $j \in K \backslash \{i\}$ and $y(t_1) \neq y(t_2)$. Note that this implies $x_i(t_1) \neq x_i(t_2)$.

Since D has resolution $2k - \ell$ and $|K \cup L| \leq 2k - \ell$, there exist $t_3, t_4 \in [m]$ such that both $\bar{x}(t_3) \restriction K = \bar{x}(t_1) \restriction K$ and $\bar{x}(t_4) \restriction K = \bar{x}(t_2) \restriction K$, and also $\bar{x}(t_3) \restriction L = \bar{x}(t_4) \restriction L$. Then $y(t_3) = y(t_1) \neq y(t_2) = y(t_4)$, which contradicts the assumption that L is a dependency set for D. $\square$

By setting $\ell = 0$ in the above lemma we get:

Corollary 44 *Let D be a data set with resolution $2k$, and let K be a minimal dependency set for D of size k. Then K is the unique dependency set for D of size $\leq k$.*

Example 45 *Let $k > 1$ be an integer, and let $n \geq 2k$. Then there exist a data set D and subsets $K, L \subseteq [n]$ such that:*
(i) $K \cap L = \emptyset$,
(ii) $|K| = |L| = k$,
(iii) Both K and L are minimal dependency sets for D,
(iv) D has resolution $2k - 1$.

Proof: Fix k, n as in the assumptions, and let K, L be disjoint subsets of $[n]$ of size k each. Let

$$D = \{< \bar{x}, y >: y = \sum_{i \in K} x_i = \sum_{j \in L} x_j\}, \tag{13}$$

where the sums are taken with respect of the addition operation in F. It is immediate from the definition of D that both K and L are dependency sets for D. Moreover, these dependency sets are minimal, because any proper subset of K or L leaves out at least one variable which can be used to make y any desired value. Similarly, suppose J is a subset of $[n]$ of size $2k - 1$ and $f : J \to F$ is any given function. We can construct $< \bar{x}, y > \in D$ such that $\bar{x}$ agrees with f on J because at least one variable index i in either K or L is not in J, and this index allows us to make the two sums in (13) equal. This shows that D has resolution $2k - 1$. $\square$

The above example shows that, in general, the assumption of Corollary 44 that D have resolution $2k$ cannot be weakened. However, it is possible to relax this condition if the data set D admits models of a certain kind.

Definition 46 *A function $f : F^n \to F$ is called* canalizing *if there exists a variable x_i called a* canalizing variable, *a value $u \in F$ called a* canalizing value, *and a value $v \in F$ called the* canalized value *such that $f(\bar{x}) = v$ whenever $x_i = u$.*

Definition 46 generalizes the well-known definition of canalizing, or forcing, Boolean functions [20], [13], [10], [19], [12]. Canalizing variables may not be unique; for example, all monomials x^α are canalizing (with canalizing and canalized value 0) in every variable x_i such that $\alpha(i) > 0$. The canalizing value is not in general unique; however, if $F = \mathbb{F}_2$, then the canalizing value is unique unless the function depends on at most one variable. It was shown in [10] that a great majority of Boolean gene regulatory functions that have been experimentally characterized are canalizing functions with several canalizing variables. We will show that for data sets D that are generated by such regulatory functions h the conclusion of Corollary 44 remains valid even if D has resolution that is somewhat smaller than $2k$. Actually, we will prove this for a wider class of functions than the ones that are canalizing in several variables.

Definition 47 *Let $h : F^n \to F$. We say that h* is iteratively canalizing with canalizing variable sequence $\bar{i} =< i_1, \ldots i_\ell >$, canalizing value sequence $\bar{u} =< u_1, \ldots, u_\ell >$, and canalized value sequence $\bar{v} =< v_1, \ldots, v_\ell >$ *if for all $r \in [\ell]$ and all $\bar{x} \in F^n$:*

$$(x_{i_r} = u_r \ \& \ \forall j < r \ x_{i_j} \neq u_j) \to f(\bar{x}) = v_r.$$

We say that h is iteratively canalizing for ℓ variables *if there exist sequences $\bar{i}$, $\bar{u}$ and $\bar{v}$ of length ℓ such that h is iteratively canalizing with canalizing variable sequence $\bar{i}$, canalizing value sequence $\bar{u}$, and canalized value sequence $\bar{v}$.*

Clearly, every function is iteratively canalizing in $\ell = 0$ variables, and any function that is canalizing in ℓ variables is also iteratively canalizing in ℓ variables. But the notion of being iteratively canalizing is much broader. For example, consider the Boolean function $f(x_1, x_2) = x_1x_2 + x_1$. For this function, x_2 is not a canalizing variable, therefore f is canalizing only in one variable. However, f is iteratively canalizing with canalizing variable sequence $< 1, 2 >$. Note that $< 2, 1 >$ is *not* a canalizing variable sequence for f.

Theorem 48 *Let $D = \{< \bar{x}(t), y(t) >: t \in [m]\}$ be a data set and let K be a minimal dependency set for D of size k. Assume moreover that D has a model h that is iteratively canalizing with canalizing variable sequence $\bar{i} =< i_1, \ldots i_\ell >$ such that $\{i_1, \ldots, i_\ell\} \subseteq K$. If D has resolution $\max\{2k - \ell, k + 1\}$, then $K \subseteq L$ for every dependency set L for D with $|L| \leq k$. In particular, K is the unique dependency set for D of size $\leq k$.*

Proof: Suppose $D, K, h, \bar{i}$ are as in the assumptions of Theorem 48, let $\bar{u} =< u_1, \ldots, u_\ell >$ be a canalizing value sequence for h and $\bar{i}$, and let $\bar{v} =< v_1, \ldots, v_\ell >$ be the corresponding canalized value sequence. Let L be a dependency set for D of size at most k. We show that $L = K$. If $|K \cap L| \geq \ell$, then the result follows from Lemma 43. So assume towards

a contradiction that $|K \cap L| < \ell$. Let $r \leq \ell$ be the smallest positive integer j such that $i_j \notin L$. We distinguish two cases.

Case 1: There exist t_1, t_2 with $y(t_1) \neq y(t_2)$ and $x_{i_j}(t_s) \neq u_j$ for all $j < r$ and $s \in \{1, 2\}$.

Then at most one of the values $y(t_1), y(t_2)$ can be v_r; assume wlog that $y(t_1) \neq v_r$. Since D has resolution $k+1$, we find t_3 such that $\bar{x}(t_3) \restriction L = \bar{x}(t_1) \restriction L$ and $x_{i_r}(t_3) = u_r$. Since L is a dependency set, the former implies that $y(t_3) = y(t_1) \neq v_r$. On the other hand, the latter implies $y(t_3) = v_r$ by the choice of v_r and the definition of being iteratively canalizing. We have reached a contradiction.

Case 2: For all t_1, t_2 such that $x_{i_j}(t_s) \neq u_j$ for all $j < r$ and $s \in \{1, 2\}$ we have $y(t_1) = y(t_2)$.

In this case the set $K_r = \{i_j : j < r\}$ will be a dependency set for D, because for all $\bar{x}(t)$ with $x_{i_j}(t) = u_j$ for some $j < r$ the value $y(t)$ will be determined by the iteratively canalizing property, and for all other $\bar{x}(t)$ the value $y(t)$ will be fixed by the assumptions of Case 2. But K_r is a proper subset of K, which contradicts our assumption that K was minimal. $\square$

For $h \in F[x_1, \ldots, x_n]$ let us define a new random variable λ_h^+ on all $\bar{D} \in \bar{\mathcal{D}}_h$ as the smallest m such that (an implementation of) the LS-algorithm with preprocessing returns h when run on I_{C_m}.

Theorem 49 *Let h be such that $|supp(h)| \leq k$. Let $0 \leq \ell \leq k$ be such that h is iteratively canalizing in ℓ variables from $supp(h)$, and let $j = \max\{2k - \ell, k+1\}$. Then*

$$E(\lambda_h^+) \leq |F|^j \, (j \, (ln\, n + \ln |F|) + \frac{1}{1 - e^{-1}}).$$

In particular, for any h with $|supp(h)| \leq k$ we have

$$E(\lambda_h^+) \leq |F|^{2k} \, (2k \, (ln\, n + \ln |F|) + \frac{1}{1 - e^{-1}}).$$

Proof: By Lemma 24 it suffices to show that $E(\lambda_h^+) \leq E(\rho_j)$. So let $h, k, \ell, j, \bar{D}$ be as in the assumption, and assume that $m \geq \rho_j(\bar{D})$, *i.e.,* assume that D_m has resolution j. Then $K = supp(h)$ is a dependency set for D_m that is iteratively canalizing in ℓ variables. Thus the assumptions of Theorem 48 are satisfied. It follows from this theorem that the first step of the LS-algorithm with preprocessing returns $L = supp(h)$. Thus in Step 2 of the algorithm we will pick a variable order that puts the variables in $supp(h)$ last, and in the third step we will work with the associated lex order $\preceq$. Moreover, our assumption on m implies that D_m fully resolves $supp(h)$. Now the theorem follows from Lemma 16(i). $\square$

Corollary 50 *Let h be such that $|supp(h)| \leq k$. Let $0 \leq \ell \leq k$ be such that h is iteratively canalizing in ℓ variables from $supp(h)$, and let $j = \max\{2k - \ell, k+1\}$. Then $E(\lambda_h^+)$ is $O(|F|^j \, j \, ln\, n)$. In particular, for any h with $|supp(h)| \leq k$ the expected value $E(\lambda_h^+)$ is of order $O(|F|^{2k} \, 2k \, ln\, n)$.*

4 Summary of results and discussion

It is by now commonplace that reverse engineering problems of biochemical networks tend to be vastly underdetermined. Within the framework of modeling a regulatory function by a function $h : F^n \to F$, the precise meaning of this phrase is the following: If D is a data set of size m, then there exist $|F|^{|F|^n - m}$ distinct models consistent with h. Reverse engineering algorithms will typically report just one of these models, and the perhaps most important quality measure for comparing such algorithms is how quickly they converge to the correct model of D, that is, how much data are needed on average before the algorithm finds the correct solution. Note that if the algorithm were just to pick a solution randomly from the set of all possible solutions, the expected amount of data needed for its convergence would grow superexponentially in the number of variables n. This is clearly unacceptable in practice, and any usable algorithm will need to perform much better for h that are likely to be the true models of our data sets.

Since the true regulatory functions in biochemical networks tend to have relatively small support [4], we are especially interested in the convergence rate for h with small support. Moreover, one would like to know which kind of term order one should use in the LS-algorithm to maximize the probability of convergence to the true model with this property. The results we have obtained in this paper give some guidelines, at least under the assumption that the set of data inputs is sufficiently random.

1. If h is any polynomial that depends on at most k variables, then there exists a data set D with $|D| \leq |F|^k$ such that for *every* term order $\preceq$ the LS-algorithm will return h (Corollary 17).

2. If h contains a monomial of maximum multidegree that depends on at most k variables and D is any data set such that for *some* term order $\preceq$ the LS-algorithm returns h, then $|D| \geq |F|^k$ (Corollary 19).

3. Let $h = a_1 x^{\alpha_1} + \cdots + a_\ell x^{\alpha_\ell}$ be a polynomial with $\max\{\sum \alpha_w : w \in [\ell]\} = k$ and $\max\{supp(\alpha_w) : w \in [\ell]\} = j$, and let $\bar{D} \in \bar{\mathcal{D}}_h$ be a random sequence of data for which h is a model. Then the expected number of data points needed before the LS-algorithm that is run with a randomly chosen graded term order returns h is on the order of at least $\Omega(n^j)$ (Corollary 33) and at most $O(|F|^k n^k)$ (Corollary 27). The expected number of data points needed before the LS-algorithm that is run with an optimally chosen graded term order returns h is on the order of at least $\Omega(n^{k-1})$ (Corollary 36).

4. Let h be a nonconstant polynomial that depends on at most k variables and let $\bar{D} \in \bar{\mathcal{D}}_h$ be a random sequence of data for which h is a model. Then:

   (a) The expected number of data points needed before the LS-algorithm with an *optimally chosen* lex order $\preceq$ returns h is on the order of $O(|F|^k k \ln |F|)$. (Theorem 25(i)).

(b) The expected number of data points needed before the LS-algorithm with an *randomly chosen* lex order $\preceq$ returns h is on the order of $\Omega(c^n)$ for some constant $c > 1$. (Corollary 41). Moreover, the number of data points needed for the LS-algorithm to return h with probability $> q$ for any fixed positive q grows exponentially in n (Theorem 40).

5. Let $h \in F[x_1, \ldots, x_n]$ be a polynomial that depends on at most k variables. Then the expected number of data points needed before the LS-algorithm with preprocessing returns h is on the order of $O(|F|^{2k}\, 2k\, ln\, n)$. Moreover, if h is iteratively canalizing in ℓ of its variables, then this expected number of data points needed for convergence is on the order of $O(|F|^j\, j\, ln\, n)$, where $j = \max\{2k - \ell, k + 1\}$ (Corollary 50).

Recall that the LS-algorithm was designed to find most parsimonious models for the data. One can interpret items 1 and 2 above as bounds on the number of variables and the multidegree of monomials that such most parsimonious models may contain. Items 3 and 4 have practical significance for the choice of input parameter $\preceq$. Item 3 implies that if we expect the true model to depend on at most k variables and to contain only monomials x^α with $\sum \alpha \leq k$, and if we have on the order of n^k data points, then running the LS-algorithm with a graded term order may be a safe bet, and it really does not matter all that much which particular graded term order we use.

Unfortunately, in applications to molecular biology, the number n of variables will typically be larger than the number m of data points, possibly by one or more orders of magnitude. In this case, item 3 implies that we should not expect the algorithm to return any nonlinear models when a graded term order is used; regardless of which particular graded term order is chosen as the input. Thus if we are looking for nonlinear models for such data sets, using lex orders might be a better strategy, and item 4(a) shows that this strategy can work even with very few data points, as long as we choose a near optimal lex order. However, item 4(b) shows that a random choice of lex order will make it very unlikely that the algorithm returns the true regulatory function, unless the number of data points were exponentially large.

These pitfalls can be avoided by judiciously choosing a lex order $\preceq$ based on preprocessing. Item 5 shows that versions of the LS-algorithm with preprocessing, such as in [11], are expected to need only $O(|F|^{2|supp(h)|}\, |supp(h)|\, ln\, n)$ data points for convergence to the correct model h. This is the same order of magnitude as the best known upper bound derived in [2] and [15] for related algorithms. For h that are iteratively canalizing in at least some variables (a property that should be expected at least of gene regulatory functions by the results of [10]), we found an even better convergence rate.

All our results about expected values are based on the assumption of random data inputs. Clearly, the data inputs of real experiments on *in vivo* response of a biochemical network to certain conditions will not be "random," for at least two reasons: First of all, most of the $|F|^{|F|^n}$ possible concentration vectors are likely to be lethal and would not elicit an *in vivo* response. Second, a real experimenter will typically have limited control over the

choice of the input vectors. She may be able to collect time series data (in which case the input vector for the next measurement is dictated by the network itself), or to knock out or overexpress a few genes in the network, but not a substantial proportion of them. In view of this, we must carefully consider the question to what extent our results are of relevance to the analysis of real biomolecular data.

First of all, except for quantum effects, nothing in nature is truly "random." A random coin flip becomes a deterministic event if we can measure the initial position and momentum of the coin with sufficient precision. Assumptions of randomness are usually just a way of formalizing our ignorance about the conditions that influence an outcome. While it seems clear that most of the theoretically possible data sets could not be produced in an actual wetlab, it is not presently known what the true distribution of feasible sets of data inputs is, and even if it were known, this distribution would likely depend on the particular network that is being studied. Our assumption of a uniform distribution of sets of data inputs is just a way of formalizing this ignorance. Since no other distribution is supported by the current state of our knowledge, if we want to study expected performance of any algorithm at all, the assumption of a uniform distribution of data inputs is practically forced upon us.

It does follow from the above though that one should exercise great care when interpreting our results. It would be inappropriate to conclude that all our estimates of expected values remain strictly valid for the unknown distribution of data sets that biologists will want to analyze in the near future. However, we believe that our results are of practical relevance if one is willing to treat them as ball-park figures. For example, we have shown that running the LS-algorithm with graded or randomly chosen lex orders is expected to be inadequate for most data sets of realistic size. It would be very surprising indeed if the actual distribution of real data sets would improve the expected performance of the algorithm in these cases by orders of magnitude, and we believe that our results make a solid case for the need of preprocessing.

Our most optimistic result, Theorem 49, does not require that the set of data inputs is random, only that it has sufficient resolution. This suggests a clear recommendation: An experimenter who has already obtained a partial data set should plan subsequent experiments in such a way as to maximize the expected resolution of the final data set. The question of how to translate this general recommendation into specific guidelines as to which of the possible knockout/overexpression experiments to perform suggests itself as a question of future research (see also [1] for related work in a different framework). Another important question for future research suggested by this result is whether and to what extent currently feasible experimental procedures and properties of discrete dynamical systems place limitations on the resolution of data sets. If not, then Theorem 49 should be directly applicable to real data sets. If on the other hand there are limitations, then it is worth investigating whether and how any observed tendencies towards limited resolution may themselves be used to make inferences about the network.

Theorem 48 leads to a strict performance guarantee of the LS-algorithm with preprocessing when the data set does have the required resolution. It is likely though that data sets encountered in practical applications may "almost" have a reasonably high resolution, but

a few small subsets of the variable set will remain unresolved. Thus studying the expected performance of the algorithm on such data sets is an important question for future research. It should be noted that the strategy outlined in our version of the "LS-algorithm with preprocessing" is not the only feasible way of choosing a promising term order. Suppose the true model h has support of size k. By Theorem 25(iv), if we run the LS-algorithm on a data set with random inputs of size $\omega(k|F|^k)$ on the order of $\omega(n^k)$ times with randomly and independently chosen lex orders, then the probability that the algorithm returns h at least once will approach one. This already reduces the number of candidate models from superexponential to polynomial, and various strategies can be tried to extract a most likely model or a most promising term lex order for the final run of the algorithm from these models. The former has been attempted in [17] and [21], where a majority or consensus criterion was used and in [3], where the analysis of candidate models was based on the Deegan-Packel index of power [6]. It should be of interest to compare the expected performance of such alternative strategies of pre- or postprocessing with the one described in this this paper and [11] on data sets that have "almost" high resolution.

In summary, the author believes that the results presented here give some guidance for the use of the LS-algorithm and its refinements for the analysis of biochemical data. In particular, they clearly demonstrate the value of preprocessing. The working biologist is faced with a bewildering variety of modeling paradigms and algorithmic tools for network data analysis. In order to choose the tool most appropriate for a given data set, one needs to know how well a given tool is expected to perform. Clearly, there is a need for comparative analysis of different algorithms and modeling paradigms. This paper contains such an analysis for one of the available tools, the LS-algorithm. It is our hope that similar analyzes for alternative algorithms will eventually equip biologists with a set of useful criteria for choosing the tool that is most appropriate for analyzing a given data set.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Agreement No. 0112050 while the author was a visitor at the Mathematical Biosciences Institute. The author wishes to thank Brandy Stigler and Jennifer Galovich for helpful comments and the MBI for providing an excellent research environment.

References

[1] Akutsu, T., Kuhara, S., Maruyama, O., and Miyano, S. (1998). Identification of Gene Regulatory Networks by Strategic Gene Disruptions and Gene Overexpressions. *Proc. 9th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '98),* 695–702.

[2] Akutsu, T., Miyano, S., and Kuhara, S. (1999). Identification of genetic networks from a small number of gene expression patterns under the Boolean network model. *Pacific Symp. Biocomput.* **4**, 17–28.

[3] Allen, E. E., Fetrow, J. S., Daniel, L. W., Thomas, S. J, and David, J. J. (2006). Algebraic dependency models of protein signal transduction networks from time-series data. *J. theor. Biol.* **238**(2), 317–330.

[4] Arnone, M. I. and Davidson, E. H. (1997). The hardwiring of development: organization and function of genomic regulatory systems. *Develoment* **124**, 1851–1864.

[5] Cox, D., Little, J., and O'Shea, D. (1992). *Ideals, varieties, and algorithms : an introduction to computational algebraic geometry and commutative algebra.* Springer-Verlag.

[6] Deegan J. and Packel E. (1978). A new index for simple n-person games. *Int. J. Game Theory* **7**, 113–123.

[7] De Jong, H. (2002). Modeling and Simulation of Genetic Regulatory Systems: A Literature Review. *J. Comput. Biol.* **9**(1), 67–103.

[8] D'haseleer, P., Liang, S., and Somogyi, R. (2000). Genetic network inference: from co-expression clustering to reverse engineering. *Bioinformatics* **16**(8), 707–726.

[9] Gardner, T. S. and Faith, J. (2005). Reverse-engineering transcription control networks. *Physics of Life Reviews* **2**, 65–88.

[10] Harris, S. E., Sawhill, B. K., Wuensche, A., Kauffman, S., A model of transcriptional regulatory networks based on biases in the observed regulation rules. *Complexity* **7**(4), (2002) 23–40.

[11] Jarrah, A., Laubenbacher, R., Stigler, B. and Stillman, M. (200?). Reverse-engineering of polynomial dynamical systems. *In preparation.*

[12] Just, W., Shmulevich, I., and Konvalina, J. (2004). The number and probability of canalizing functions. *Physica D* **197**(3-4), 211–221.

[13] Kauffman, S. A. (1990). Requirements for Evolvability in Complex Systems: Orderly Components and Frozen Dynamics. *Physica D,* **42**, 135–152.

[14] Kauffman, S. A. (1993). *The origins of order: Self-organization and selection in evolution.* Oxford U Press.

[15] Krupa, B. (2002). On the Number of Experiments required to Find the Causal Structure of Complex Systems. *J. theor. Biol.* **219**, 257–267.

[16] Kleitman, D. J. and Spencer, J. (1973). Families of k-independent sets. *Discrete Math.* **6**, 255–262.

[17] Laubenbacher, R. and Stigler, B. (2004). A computational algebra approach to reverse engineering of gene regulatory networks. *J. theor. Biol.* **229**, 523–537.

[18] Palsson, B. (2006). *Systems Biology: Properties of Reconstructed Networks.* Cambridge U Press.

[19] Shmulevich, I., Lähdesmäki, H., Dougherty, E. R., Astola, J., and Zhang, W. (2003). The role of certain Post classes in Boolean network models of genetic networks. *Proc. Natl. Acad. Sci. USA* **100**(19), 10734–10739.

[20] Stauffer, D. (1987). On Forcing Functions in Kauffman's Random Boolean Networks. *J. Stat. Phys.* **46**(3-4), 789–794.

[21] Stigler, B. (2005). An Algebraic Approach to Reverse Engineering with an Application to Biochemical Networks. *Ph.D. Thesis,* Virginia Tech. http://scholar.lib.vt.edu/theses/available/etd-08252005-075644/