UNIVERSIDAD DE CONCEPCIÓN

Centro de Investigación en Ingeniería Matemática (CI^{2}MA)

An exploratory approach to the compatibility and inference of unate functions

Julio Aracena, Katerin De La Hoz,
Alexis Poindron, Lilian Salinas

PREPRINT 2025-27

SERIE DE PRE-PUBLICACIONES

An exploratory approach to the compatibility and inference of unate functions

Katerin de la Hoz[*], Alexis Poindron[†], Lilian Salinas[‡], and Julio Aracena[§]

November 24, 2025

Abstract

Unate functions arise naturally in diverse areas of mathematics and computer science and play a central role in the modeling of gene regulatory systems with Boolean networks, where each regulator acts as an inducer or a repressor. Inferring such functions from observational data is a fundamental task that has been primarily approached from a biological perspective. We address the problem of determining whether there exists a unate function compatible with a set of observations by introducing two central concepts: the multiset of discrepancies and the coverage vector. We demonstrate that the existence of a covering vector is a necessary and sufficient condition for solving the unate compatibility problem, and we characterize the entire space of functions associated with a coverage vector, thereby formalizing the unate inference process. We propose two complementary algorithms: a coverage algorithm that efficiently constructs sparse coverage vectors, and an exploratory algorithm that analyzes collections of coverage solutions to identify patterns of influence and derive representative quasi-solutions. Experiments on synthetic datasets demonstrate scalability and feasibility, while the *Drosophila* ventral furrow case study indicates that, together, the algorithms constitute an exploratory approach for Boolean network inference when information is limited to observations and that they have the potential to systematically explore plausible regulatory architectures.
Keywords: Unate Boolean functions, Coverage vector, Gene regulatory systems, Boolean networks, Inference.

1 Introduction

A *Boolean function* f of n Boolean variables is a map $f : \mathbb{B}^n \to \mathbb{B}$, where $\mathbb{B} = \{0, 1\}$. The Boolean function f is said to be *monotone increasing* in the i-th variable if, for all $x \in \mathbb{B}^n$ with $x_i = 0$, we have $f(x) \leq f(x \oplus e_i)$; and it is *monotone decreasing* if $f(x) \geq f(x \oplus e_i)$, where $e_i \in \mathbb{B}^n$ denotes the vector with all coordinates equal to 0 except for the coordinate i, which is equal to 1. Additionally, $\oplus$ is the component-wise addition modulo 2. When the inequality is strict for at least one x, we say that i is an *influencer*; otherwise, f is constant in the i-th variable. These notions lead to the definition of unate functions, which are the focus of this study. A Boolean function f is called a *unate function* if, for each $i \in [n]$ (where $[n] = \{1, \ldots, n\}$), it is monotone increasing or monotone decreasing in the i-th variable. Unate functions gain particular importance in the study of Boolean

[*]Departamento de Ingeniería Informática y Ciencias de la Computación, Universidad de Concepción, kdelahoz@udec.cl

[†]Unité d'Economie Appliquée, ENSTA, Institut Polytechnique de Paris, alexis.poindron@ensta-paris.fr

[‡]CI^{2}MA and Departamento de Ingeniería Informática y Ciencias de la Computación, Universidad de Concepción, lilisalinas@udec.cl

[§]CI^{2}MA and Departamento de Ingeniería Matemática, Universidad de Concepción, jaracena@udec.cl

networks (BNs), especially in the modeling of gene regulatory systems ([11]), where they naturally capture inducer and repressor interactions ([8]).

A central question is the induction of Boolean functions from observations: given binary data partitioned into 'positive examples' and 'negative examples', the objective is to decide whether there exists a function belonging to a specific class that accepts all the positives and rejects all the negatives and, if so, to identify it. This problem is framed within the theory of partially defined Boolean functions ([6, 13]), and it is shown that, for several classes, including unate functions, deciding existence is NP-complete ([6]). In this work, we refer to this as the *compatible unate function problem*. The inference problem of BNs from observational data, in its general form, has been widely studied ([1, 9, 14]), and numerous algorithms have been proposed and compared ([16, 17]). Most of these approaches aim to approximate the entire biological system, aligning the inferred BN with the underlying molecular mechanisms ([15]). By contrast, our approach is exploratory: it operates at the node level, inferring compatible unate functions independently, which could be combined to reveal the structural properties of the network as a whole.

The compatible unate function problem is addressed through two key concepts: the *multiset of discrepancies*, which summarizes how positive and negative observations differ; and the *coverage vector*, which represents a possible way to solve these differences. Once the theoretical results have been established, we propose two complementary algorithms: a *coverage algorithm* designed to produce sparse coverage vectors and to converge quickly, subject to a stop condition that limits computational effort; and an *exploratory algorithm* to analyze collections of coverage vectors, identify recurrent patterns of influence, and derive representative quasi-solutions. Together, they define an *exploratory approach* that is flexible and adaptable to diverse experimental scenarios. In experiments with synthetic datasets, the coverage algorithm exhibits scalability, maintaining practical execution times even for functions with several hundred variables. Under constraints on the weight of the coverage solutions, execution times remain practical, and the observed success rates indicate feasibility, even though the constraint makes it difficult to find solutions. In the *Drosophila* ventral furrow regulatory BN case study presented in this work, the inferred functions preserve the regulatory structure in terms of the number of influencers, which indicates that our approach is suitable for inferring BNs when the available information is limited to observations.

In Corollary 4, we establish that the existence of a coverage vector is a necessary and sufficient condition for the existence of a compatible unate function. Meanwhile, in Theorem 6, we characterize all compatible unate functions associated with a given coverage vector, bounding them between lower and upper functions, thereby grounding the inference process. We also analyze optimization aspects of coverage vectors: *minimal subcoverage* captures the local effort to simplify coverage solutions and admits a linear time procedure, while *minimum subcoverage* reflects the global effort and defines an NP-complete problem.

2 Theoretical Results

2.1 The main problem

An *observation* is a pair (x, t), where $x \in \mathbb{B}^n$ and $t \in \mathbb{B}$. Given a set F of observations, we distinguish $F_0 = \{x : (x, 0) \in F\}$ and $F_1 = \{x : (x, 1) \in F\}$, and assume that $F_0 \cap F_1 = \emptyset$. We say that $f : \mathbb{B}^n \to \mathbb{B}$ is *compatible* with F_0 and F_1 if, for all $y \in F_0$, $f(y) = 0$, and for all $z \in F_1$, $f(z) = 1$. We refer to the process of identifying one or more functions that satisfy this condition as the *inference* of compatible functions.

For the general case, it is always possible to infer a function f that is compatible with a set of observations simply by defining $f(x) = 1 \Leftrightarrow x \in F_1$. However, such a function does not need

to be unate, and even if it is, it is not necessarily unique. For example, consider $F_0 = \{000, 111\}$ and $F_1 = \{100, 011\}$, and let $f : \mathbb{B}^3 \to \mathbb{B}$ be compatible with F_0 and F_1. From $f(000) = 0$ and $f(100) = 1$, we obtain that f is monotone increasing in x_1, while $f(111) = 0$ and $f(011) = 1$ force it to be monotone decreasing in x_1; hence, there is no compatible unate function. Conversely, for $F_0 = \{011\}$ and $F_1 = \{101, 110\}$, there are at least two compatible unate functions: $f(x) = x_1$ and $g(x) = \overline{x}_2 \wedge \overline{x}_3$.

The first purpose of the paper is to determine whether there exists a unate function f that is compatible with a set of observations. If so, the second purpose is to infer it.

Problem 1 (**Compatible Unate Function Problem (CUF)**)**.** Given F_0, $F_1 \subseteq \mathbb{B}^n$. Does there exist a unate function $f : \mathbb{B}^n \to \mathbb{B}$ compatible with F_0 and F_1?

Theorem 1. ([6]) CUF is NP-Complete.

2.2 The notion of discrepancy

In the following, subscripts indicate coordinates and superscripts label vectors. We say that a vector x is *signed* if $x \in \{0, -, +\}^n$. The *support* of a signed vector x is $\operatorname{supp}(x) = \{i : x_i \neq 0\}$, and its *weight* is $w(x) = |\operatorname{supp}(x)|$.

For $y \in F_0$ and $z \in F_1$, the *discrepancy* between y and z is defined as the signed vector $\delta(y, z)$, such that for each $i \in [n]$,

$$\delta(y,z)_i = \begin{cases} + & \text{if } y_i < z_i, \\ - & \text{if } y_i > z_i, \\ 0 & \text{otherwise} \end{cases}$$

We write simply δ when no confusion can arise.

Since $F_0 \cap F_1 = \emptyset$, no discrepancy is equal to the vector with all its coordinates equal to 0, denoted by $\mathbf{0}$. Furthermore, as different pairs of observations can share the same discrepancy, we maintain multiplicities and obtain what we call the *multiset of discrepancies*, denoted by Δ_F.

Example 1. Given $F_0 = \{1000, 0010\}$ and $F_1 = \{1111, 0100\}$, we get the following discrepancies: $\delta(1000, 1111) = [\ 0\ +\ +\ +\ ]$, $\delta(1000, 0100) = [\ -\ +\ 0\ \ 0\ ]$, $\delta(0010, 1111) = [\ +\ +\ 0\ +\ ]$, and $\delta(0010, 0100) = [\ 0\ +\ -\ \ 0\ ]$.

Every set of observations induces a multiset of discrepancies, but the converse does not necessarily hold. This is not due to the function being unate or not; rather, it is a dimensionality issue.

Example 2. Let $S = \{[\ +\ +\ -\ ], [\ 0\ -\ -\ ]\}$. For this to be a multiset of discrepancies, necessarily $[\ +\ +\ -\ ] = \delta(001, 110)$ and $[\ 0\ -\ -\ ] = \delta(\alpha 11, \alpha 00)$, where $\alpha \in \mathbb{B}$. So, $F_0 = \{001, 011\}$ and $F_1 = \{110, 000\}$, or $F_0 = \{001, 111\}$ and $F_1 = \{110, 100\}$. In both cases, $[\ 0\ \ 0\ \ -\ ] = \delta(001, 000) = \delta(111, 110) \notin S$, hence S is not a multiset of discrepancies.

Given a signed vector Σ, we say that a discrepancy δ is *covered by* Σ if there exists $k \in [n]$ such that $\Sigma_k = \delta_k = +$ or $\Sigma_k = \delta_k = -$. Moreover, Σ is a *coverage vector* for a multiset of discrepancies Δ if every $\delta \in \Delta$ is covered by some coordinate of Σ.

Example 3. Continuing with Example 1, $[\ -\ \ 0\ \ -\ \ +\ ]$ is a coverage vector for Δ_F. Note that the second coordinate is 0, since all discrepancies are covered by the other coordinates. Besides, $[\ 0\ \ +\ \ 0\ \ \ 0\ ]$ is also a coverage vector, proving that coverage vectors are not necessarily unique.

The dependence of each variable in a unate function f is summarized by a signed vector, denoted by $\Sigma(f)$, and referred to as the *influence vector* of f. Its i-th entry is $+$ (for monotone increasing), $-$ (for monotone decreasing), or 0 (for constant); based on the behavior of f in the i-th variable. For $\Sigma(f)$ to be uniquely defined, we assume minimality, i.e., $\Sigma(f)_i \neq 0$ if and only if there exists $x \in \mathbb{B}^n$ such that $f(x) \neq f(x \oplus e_i)$. A fundamental observation is:

Proposition 2. Let $F_0, F_1 \subseteq \mathbb{B}^n$. If f is a unate function compatible with F_0 and F_1, then $\Sigma(f)$ is a coverage vector for Δ_F.

Proof. Let $y \in F_0$ and $z \in F_1$, with $\text{supp}(\delta(y,z)) = \{s_1, s_2, \ldots, s_l\}$. We define a path $P : x^0 = y, x^1, x^2, \ldots, x^l = z$ in $\mathbb{B}^n$, where $\forall i \in [l], x^i := x^{i-1} \oplus e_{s_i}$. Since $f(y) < f(z)$, $\exists j \in [l]$ such that $f(x^{j-1}) < f(x^j)$, which is equivalent to $f(x^{j-1}) < f(x^{j-1} \oplus e_{s_j})$. Note that $y_{s_j} = x^{j-1}_{s_j}$ and $z_{s_j} = x^j_{s_j}$, which implies that $\delta(x^{j-1}, x^j)_{s_j} = \delta(y,z)_{s_j}$. We now show that $\delta(y,z)$ is covered by $\Sigma(f)$. If $\delta(x^{j-1}, x^j)_{s_j} = +$, then by the definition of monotone increasing, we have $\Sigma(f)_{s_j} = +$; hence $\Sigma(f)_{s_j} = \delta(y,z)_{s_j}$. Analogously, if $\delta(x^{j-1}, x^j)_{s_j} = -$, then by the definition of monotone decreasing, we have $\Sigma(f)_{s_j} = -$, and again $\Sigma(f)_{s_j} = \delta(y,z)_{s_j}$. As $y \in F_0$ and $z \in F_1$ were arbitrary, every discrepancy is covered; therefore, $\Sigma(f)$ is a coverage vector for Δ_F. □

Proposition 2 suggests that the coverage vector is a good candidate for solving Problem 1. We now show that at least two compatible unate functions can be inferred from a coverage vector, which may be identical. For this purpose, we use a well-known result: a Boolean function is unate if it can be written in DNF or CNF with each variable appearing only in its negated or non-negated form ([2]).

Proposition 3. Let $F_0, F_1 \subseteq \mathbb{B}^n$, and let Σ be a coverage vector for Δ_F. The functions:

$$f^{\downarrow_\Sigma}(x) := \bigvee_{z \in F_1} C_z \quad \text{and} \quad f^{\uparrow_\Sigma}(x) := \bigwedge_{y \in F_0} C_y,$$

where,

$$C_z := \bigwedge_{\substack{z_i=1 \\ \Sigma_i=+}} x_i \wedge \bigwedge_{\substack{z_i=0 \\ \Sigma_i=-}} \overline{x_i} \quad \text{and} \quad C_y := \bigvee_{\substack{y_i=1 \\ \Sigma_i=-}} \overline{x_i} \vee \bigvee_{\substack{y_i=0 \\ \Sigma_i=+}} x_i,$$

are unate functions and compatible with F_0 and F_1.

Proof. We begin by introducing the notation used in the proof. Define $\text{sgn} : \mathbb{B}^n \to \{-,+\}^n$ such that $\forall x \in \mathbb{B}^n$, $\forall i \in [n]$:

$$\text{sgn}(x)_i = \begin{cases} - & \text{if } x_i = 0, \\ + & \text{if } x_i = 1. \end{cases}$$

We also rely on the following two observations, which follow directly from the definitions of discrepancies and coverage:

(a) there does not exist $y \in F_0$ such that $\forall i \in [n]$, $\Sigma_i \neq 0 \implies \text{sgn}(y)_i = \Sigma_i$;

(b) there does not exist $z \in F_1$ such that $\forall i \in [n]$, $\Sigma_i \neq 0 \implies \text{sgn}(y)_i \neq \Sigma_i$.

Indeed, if either of the two conditions fails, we could find a pair $\hat{y} \in F_0$ and $\hat{z} \in F_1$ such that:

$$\forall i \in [n], (\Sigma_i = + \implies \hat{y}_i = 1 \wedge \hat{z}_i = 0) \wedge (\Sigma_i = - \implies \hat{y}_i = 0 \wedge \hat{z}_i = 1),$$

which would imply,

$$\forall i \in [n], (\Sigma_i = + \implies \delta(\hat{y}, \hat{z})_i = -) \wedge (\Sigma_i = - \implies \delta(\hat{y}, \hat{z})_i = +),$$

so $\delta(\hat{y}, \hat{z})$ is not covered by Σ, which is a contradiction.

With these preliminaries established, we now prove the proposition. By (a) and (b), all clauses C_z and C_y are non-empty, and each variable appears only in its negated or non-negated form; hence $f^{\downarrow\Sigma}$ and $f^{\uparrow\Sigma}$ are unate. Now we prove that $f^{\downarrow\Sigma}$ is compatible with F_0 and F_1.

Let $\hat{y} \in F_0$. Since Σ is a coverage vector, for every $z \in F_1$ there exists $j \in [n]$ such that $\Sigma_j = \delta(\hat{y}, z)_j = +$ or $\Sigma_j = \delta(\hat{y}, z)_j = -$. If $\Sigma_j = +$, then $\hat{y}_j = 0$ and $z_j = 1$, which implies that C_z evaluated on $\hat{y}$ is 0. Similarly, if $\Sigma_j = -$, then $\hat{y}_j = 1$ and $z_j = 0$, which also implies that C_z evaluated on $\hat{y}$ is 0. Therefore, $f^{\downarrow\Sigma}(\hat{y}) = 0$.

Let $\hat{z} \in F_1$ and evaluate C_z at $\hat{z}$:

$$\bigwedge_{\substack{\hat{z}_i=1 \\ \Sigma_i=+}} \hat{z}_i \wedge \bigwedge_{\substack{\hat{z}_i=0 \\ \Sigma_i=-}} \overline{\hat{z}_i} = 1\,.$$

Hence $f^{\downarrow\Sigma}(\hat{z}) = 1$. As $y \in F_0$ and $z \in F_1$ were arbitrary, $f^{\downarrow\Sigma}$ is compatible with F_0 and F_1.

The proof that $f^{\uparrow\Sigma}$ is compatible is analogous. □

Therefore, once we have a coverage vector, a compatible unate function can be inferred in polynomial time, as is also shown in [18].

Remark 1. The functions $f^{\downarrow\Sigma}$ and $f^{\uparrow\Sigma}$ are referred to as the lower and upper bounds, respectively, as shown in Theorem 6. Due to compatibility, it is evident that $F_1 \subseteq \{x \in \mathbb{B}^n : f^{\downarrow\Sigma}(x) = 1\}$ and $F_0 \subseteq \{x \in \mathbb{B}^n : f^{\uparrow\Sigma}(x) = 0\}$; however, the inclusions may be strict. For example, consider $F_0 = \{01, 11\}$ and $F_1 = \{10\}$, where $\Sigma = [\ 0\ -\ ]$ is a coverage vector for Δ_F. It is easy to verify that $f^{\downarrow\Sigma} = f^{\uparrow\Sigma} = \overline{x_2}$. Thus $F_1 \subsetneq \{x \in \mathbb{B}^2 : f^{\downarrow\Sigma}(x) = 1\} = \{00, 10\}$.

The following corollary captures the key theoretical idea behind our approach; and it is a direct consequence of Proposition 2 (necessity) and Proposition 3 (sufficiency).

Corollary 4. The existence of a coverage vector is a necessary and sufficient condition for the existence of a compatible unate function.

Corollary 4 solves Problem 1: finding a coverage vector is equivalent to finding a compatible unate function, and once it exists and is found, we can infer at least two functions in polynomial time, which may be identical. This result yields the following immediate consequences:

(i) If δ^1 and δ^2 are discrepancies with the same support $\{j\}$, then there exists a compatible unate function if and only if $\delta^1_j = \delta^2_j$.

(ii) A unate function f that is monotone increasing in every variable is compatible with F_0 and F_1 if and only if, for every $\delta \in \Delta_F$, there exists $j \in [n]$ such that $\delta_j = +$.

(iii) A unate function f is not compatible with F_0 and F_1 if there exists $\delta \in \Delta_F$ such that, for every $j \in \mathrm{supp}(\Sigma(f)) \cap \mathrm{supp}(\delta)$, $\Sigma(f)_j \neq \delta_j$.

2.3 Characterization of compatible unate functions

Our next goal is to characterize the set of compatible unate functions beyond $f^{\downarrow\Sigma}$ and $f^{\uparrow\Sigma}$, providing a complete description of their structure and addressing the inference process. This description relies on the role of antichains in Boolean functions; it is well known that the sets formed by the variables of the clauses of an irreducible CNF form an antichain in the set of literals under the inclusion relation ([5]).

We define the partial order $\leq_\Sigma$ on $\mathbb{B}^n$ induced by a signed vector Σ. For $x, y \in \mathbb{B}^n$, $x \leq_\Sigma y$ if and only if for each $i \in [n]$, $x_i \leq y_i$ when $\Sigma_i = +$, and $x_i \geq y_i$ when $\Sigma_i = -$. We write $x <_\Sigma y$ when at least one inequality is strict. A subset $\mathcal{A} \subseteq \mathbb{B}^n$ is a Σ-antichain if, for all $x, y \in \mathcal{A}$, they are incomparable with respect to $\leq_\Sigma$, i.e., neither $x \leq_\Sigma y$ nor $y \leq_\Sigma x$.

For a unate function f with influence vector $\Sigma = \Sigma(f)$, we say that $\mathcal{A}_f^1 \subseteq \mathbb{B}^n$ *describes the ones of* f when $f(x) = 1$ if and only if there exists $z \in \mathcal{A}_f^1$ such that $z \leq_\Sigma x$; and that $\mathcal{A}_f^0 \subseteq \mathbb{B}^n$ *describes the zeros of* f when $f(x) = 0$ if and only if there exists $y \in \mathcal{A}_f^0$ such that $x \leq_\Sigma y$.

With these definitions, F_1 and F_0 naturally describe the behavior of $f^{\downarrow\Sigma}$ and $f^{\uparrow\Sigma}$, respectively.

Proposition 5. Let $F_0, F_1 \subseteq \mathbb{B}^n$, and let Σ be a coverage vector for Δ_F. Then:

(1) F_1 describes the ones of $f^{\downarrow\Sigma}$;

(2) F_0 describes the zeros of $f^{\uparrow\Sigma}$.

Proof. Let us first prove (1). We want to show that for each $x \in \mathbb{B}^n$, $f^{\downarrow\Sigma}(x) = 1$, if and only if there exists $z \in F_1$ such that $z \leq_\Sigma x$.

By definition of $f^{\downarrow\Sigma}$, we have that:

$$\forall x \in \mathbb{B}^n,\ f^{\downarrow\Sigma}(x) = 1 \Leftrightarrow \exists z \in F_1,\ \bigwedge_{\substack{z_i=1 \\ \Sigma_i=+}} x_i \wedge \bigwedge_{\substack{z_i=0 \\ \Sigma_i=-}} \overline{x}_i = 1\,.$$

The last equality is satisfied if:

$$\forall i \in [n],\ (\Sigma_i = + \wedge z_i = 1 \implies x_i = 1) \ \wedge\ (\Sigma_i = - \wedge z_i = 0 \implies x_i = 0),$$

which implies that $x_i = z_i$.

On the other hand,

$$\forall i \in [n],\ (\Sigma_i = + \wedge z_i = 0 \implies x_i \in \{0,1\}) \ \wedge\ (\Sigma_i = - \wedge z_i = 1 \implies x_i \in \{0,1\}),$$

indicating that $z_i \leq x_i$ and $z_i \geq x_i$, respectively. Hence, for each $i \in [n]$, $z_i \leq x_i$ when $\Sigma_i = +$, and $z_i \geq x_i$ when $\Sigma_i = -$. Therefore, $z \leq_\Sigma x$, which proves the claim.

The proof of (2) is similar: using the definition of $f^{\uparrow\Sigma}$ and analogous reasoning, we prove that for each $x \in \mathbb{B}^n$, $f^{\uparrow\Sigma}(x) = 0$, if and only if there exists $y \in F_0$ such that $x \leq_\Sigma y$. □

Remark 2. The sets describing the ones and zeros of f can be turned into antichains through a process of elimination: if there is $z^1, z^2 \in \mathcal{A}_f^1$ with $z^1 <_\Sigma z^2$, then by the transitivity of $<_\Sigma$, $\mathcal{A}_f^1 - \{z^2\}$ still describes the ones. Similarly, if there is $y^1, y^2 \in \mathcal{A}_f^0$ with $y^1 <_\Sigma y^2$, then $\mathcal{A}_f^0 - \{y^1\}$ still describes the zeros. Removing all redundancies yields minimal sets with the same property, called the 1-antichain and 0-antichain of f, respectively.

The above descriptive overview provides the basis for a complete characterization of the space of compatible unate functions, thus addressing the inference problem:

Theorem 6. Let $F_0, F_1 \subseteq \mathbb{B}^n$, and let Σ be a coverage vector for Δ_F. A unate function f such that $\Sigma(f) = \Sigma$ is compatible with F_0 and F_1 if and only if $f^{\downarrow\Sigma} \leq f \leq f^{\uparrow\Sigma}$.

Proof. We prove the double implication.

$\boxed{\Longleftarrow}$ If $f^{\downarrow\Sigma} \leq f \leq f^{\uparrow\Sigma}$, it is immediate that f is compatible with F_0 and F_1, since both $f^{\downarrow\Sigma}$ and $f^{\uparrow\Sigma}$ are.

$\boxed{\Longrightarrow}$ We first show that $f^{\downarrow\Sigma} \leq f$. On the contrary, suppose that there exists $x \in \mathbb{B}^n$ such that $f^{\downarrow\Sigma}(x) = 1$ and $f(x) = 0$. From Proposition 5 (1), F_1 describes the ones of $f^{\downarrow\Sigma}$. Let $\mathcal{A}_f^0$ denote the set that describes the zeros of f. Then, there exist $\hat{z} \in F_1$ and $\hat{y} \in \mathcal{A}_f^0$ such that $\hat{z} \leq x$ and $x \leq_\Sigma \hat{y}$. By the transitivity property of $\leq_\Sigma$, we have $\hat{z} \leq_\Sigma \hat{y}$. By the definition of $\mathcal{A}_1^0$, we obtain that $f(\hat{z}) = 0$, which contradicts the compatibility of f with F_1.

The inequality $f \leq f^{\uparrow\Sigma}$ follows by an analogous argument, denoting $\mathcal{A}_f^1$ as the set that describes the ones of f and applying Proposition 5 (2). □

Remark 3. If $\bigcup_{y\in F_0}\{x : x \leq_\Sigma y\} \cup \bigcup_{z\in F_1}\{x : z \leq_\Sigma x\} \neq \mathbb{B}^n$, then $f^{\downarrow\Sigma} < f^{\uparrow\Sigma}$. Hence, compatible unate functions are in one-to-one correspondence with the antichains obtained by adding to F_0 or F_1 those vectors that are incomparable with any element of the respective set.

2.4 About minimal and minimum subcovering

Questions about the minimal and minimum weight of a coverage vector arise naturally, as these notions are relevant to analyzing the effort required to simplify solutions. Therefore, we establish basic terminology and briefly discuss the complexity of related problems.

Given two signed vectors x and y, we say that x is a *subvector* of y, denoted $x \subseteq y$, if for each $i \in [n], x_i \neq 0 \Rightarrow y_i = x_i$.

A coverage vector is *minimal* if no proper subvector of it is also a coverage vector. This raises the following question: given Δ a multiset of discrepancies and Σ a coverage vector for Δ, find a minimal subvector of Σ that is also a solution. This can be done in linear time in the number of coordinates: it suffices to choose any order in $\text{supp}(\Sigma)$ and check, one by one, whether each entry can be replaced by zero while maintaining Σ as a coverage vector.

A coverage vector is *minimum* if its weight is the smallest among all solutions. Analogous to the minimal case, we consider a subvector problem associated with the minimum:

Problem 2 (**Minimum Subcoverage Problem (MinSUB)**)**.** Given Δ a multiset of discrepancies, Σ a coverage vector for Δ and k an integer, decide if there exists a subvector of Σ that is a coverage vector for Δ and whose weight is at most k.

It is clear that the minimum coverage vector problem is NP-complete, and it turns out that the restricted subcoverage version is also NP-complete.

Theorem 7. MinSUB is NP-complete.

Proof. It is clear that MinSUB is NP. We prove that MinSUB is NP-hard by providing a polynomial reduction from the Set Cover problem, which is known to be NP-complete. In Set Cover, given a universal set $U = [m]$, a family of subsets of U denoted by $S = \{S_1, S_2, \ldots, S_n\}$ and an integer k; the goal is to determine whether there exists a subfamily of S of size at most k whose union is equal to U. We now transform an instance of Set Cover into an instance of our problem.

We define $F_0, F_1 \in \mathbb{B}^n$, where $F_0 = \{\mathbf{0}\}$ and $F_1 = \{z^1, z^2, \ldots, z^m\}$, such that $\forall i \in [m]$:

$$\forall j \in [n],\ z_j^i = \begin{cases} 1 & \text{if } i \in S_j, \\ 0 & \text{otherwise.} \end{cases}$$

Then, the set of discrepancies is given by $\Delta_F = \{\delta^1, \delta^2, \ldots, \delta^m\}$ such that $\forall i \in [m]$:

$$\forall j \in [n],\ \delta_j^i = \begin{cases} + & \text{if } z_j^i = 1, \\ 0 & \text{otherwise,} \end{cases}$$

so that $\Sigma = [\,+\ +\ \ldots\ +\,] \in \{0,+\}^n$ is a coverage vector for Δ_F.

Note that S_i is an element of a solution to the Set Cover problem if and only if the i-th coordinate is nonzero in a MinSUB solution. Therefore, $\hat{S} = \{S_{i_1}, S_{i_2}, \ldots, S_{i_k}\}$ is a subfamily of S that covers U if and only if the subvector $\hat{\Sigma}$ of Σ, defined as follows: $\hat{\Sigma}_i = +$ for all $i \in \{i_1, i_2, \ldots, i_k\}$ and 0 otherwise, is a coverage vector for Δ_F. □

3 Materials and methods

In the following, we introduce the practical components of our exploratory approach: a coverage algorithm for constructing a solution and an exploratory algorithm to analyze the structure of a collection of solutions in terms of patterns of influence.

3.1 The coverage algorithm

We present the Algorithm 1, which focuses on finding a covering vector for the multiset of discrepancies induced by F_0 and F_1.

The process begins by computing $\Delta = \Delta_F$ and initializing $\Sigma = \mathbf{0}$. In cases where prior knowledge is available, such as known signs of specific variables, this information can be preassigned in Σ, with subsequent steps adapted accordingly. Each discrepancy $\delta \in \Delta$ is characterized by its support and weight and marked as unselected. At each iteration, an unselected discrepancy δ^* under Rule A is chosen, a coordinate j from its support under Rule B is selected, and Σ_j is set to δ^*_j. This assignment ensures that every discrepancy consistent with the sign at coordinate j is covered by Σ and subsequently marked as selected. For the remaining unselected discrepancies containing j in their supports, j is removed and weights are updated to prevent j from being selected again in subsequent steps. When a discrepancy reaches weight equal to 0 before being covered by Σ, the attempt is aborted and restarted. The process is repeated until all discrepancies are covered or a stop condition holds, for example, a limit on the execution time or on the number of operations.

Although this algorithm does not include the explicit construction of a compatible unate function, once a coverage vector is obtained, it can be inferred by the procedure presented in Proposition 3, constructing $f^{\downarrow\Sigma}$ or $f^{\uparrow\Sigma}$; moreover, an intermediate compatible function can be derived as discussed in Remark 3.

3.1.1 Practical rules

The rules used in Algorithm 1 are motivated by fast convergence, either producing a coverage vector or concluding that no compatible unate function can be found. Also, there is a preference for simple solutions, where simplicity is interpreted as low weight.

Given $\Delta = \{\delta^1, \delta^2, \ldots, \delta^d\}$ and $j \in [n]$, write $[\Delta]_j = \left[\begin{array}{cccc} \delta^1_j & \delta^2_j & \ldots & \delta^d_j \end{array}\right]$. The *coverage potential* of δ^i_j is defined as the number of discrepancies that would be covered if Σ_j is set to δ_j.

Rule A. Choose the unselected discrepancy δ^* with the smallest positive weight, as it imposes the strictest restrictions; in the extreme case, a discrepancy with support $\{j\}$ sets the value of Σ_j. If there is a tie, select the discrepancy whose support has the largest sum of coverage potentials across its coordinates. This criterion prioritizes a sign assignment under which many other discrepancies are covered by Σ.

Rule B. Choose $j \in \mathrm{supp}(\delta^*)$ for which $[\Delta]_j$ exhibits the greatest deviation from a balanced sign distribution. Because we maintain multiplicities in the multiset of discrepancies, sign frequencies in $[\Delta]_j$ are accurately represented. This selection criterion offers practical advantages over simply choosing the coordinate with the highest coverage potential. An unbalanced pattern may indicate

Algorithm 1 Coverage algorithm

Require: Number of variables n, sets F_0 and F_1
Ensure: A coverage vector $\Sigma \in \{0, -, +\}^n$ or report failure

$\Delta \leftarrow \{\delta(x, y) : x \in F_0, y \in F_1\}$
for all $\delta \in \Delta$ **do**
    $\texttt{Supp}[\delta] \leftarrow \text{supp}(\delta)$, $\texttt{Wt}[\delta] \leftarrow w(\delta)$, $\texttt{Selected}[\delta] \leftarrow$ **False**
repeat
    $S \leftarrow \texttt{Supp}$, $W \leftarrow \texttt{Wt}$, $C \leftarrow \texttt{Selected}$
    $\Sigma \leftarrow \mathbf{0}$
    while $(\exists \delta \in \Delta, C[\delta] =$ **False**$)$ **do**
        if $\exists \delta \in \Delta, C[\delta] =$ **False** $\wedge\, W[\delta] = 0$ **then break**
        Choose $\delta^* \in \Delta$ with $C[\delta^*] =$ **False** under Rule A
        Choose $j \in S[\delta^*]$ under Rule B
        $\Sigma_j \leftarrow \delta^*_j$, $C[\delta^*] \leftarrow$ **True**
        for all $\delta \in \Delta, C[\delta] =$ **False** $\wedge\, j \in S[\delta]$ **do**
            if $\delta_j = \Sigma_j$ **then**
                $C[\delta] \leftarrow$ **True**
            else
                $S[\delta] \leftarrow S[\delta] \setminus \{j\}$, $W[\delta] \leftarrow W[\delta] - 1$
    if $(\forall \delta \in \Delta, C[\delta] =$ **True**$)$ **then return** Σ
until a specified stop condition is met
return FAILURE: no compatible unate function can be found

a more decisive role of a coordinate as an influencer; thus, favoring that bias tends to produce more robust solutions, i.e., more stable and consistent patterns of influence.

For $i \in \text{supp}(\delta^*)$, we write $\nu^i = w([\Delta]_i)$. To quantify the deviation exhibited by i, we define the proportion of signs in $[\Delta]_i$ that are equal to δ^*_i as $p^*_i = \frac{1}{\nu^i} \sum_{\delta \in \Delta} \mathbb{I}(\delta_i = \delta^*_i)$, where $\mathbb{I}(\alpha = \beta)$ is the indicator function, which is equal to 1 if $\alpha = \beta$ and 0 otherwise. Assuming that the signs in $[\Delta]_i$ are drawn independently from $\text{Bin}(\nu^i, 1/2)$ and approximating this by a normal distribution, we define $\alpha^*_i := \Phi(|p^*_i - 1/2|/\sqrt{1/(4\nu^i)})$, where Φ is the cumulative distribution function of the standard normal $\mathcal{N}(0, 1)$. The quantity α^*_i measures the deviation of p^k_i from randomness: values near 0.5 suggest balance, whereas values near 1 indicate pronounced bias. For $i \notin \text{supp}(\delta^*)$, we set $\alpha^*_i = 0$. The resulting vector is then normalized to form a probability distribution, and a coordinate j is randomly selected according to this distribution. We present an illustrative example:

Example 4. Consider $n = 6$ and the set of observations given by:

$$F_0 = \{011110, 100101, 011001\},$$
$$F_1 = \{101100, 000010, 000011, 110000, 100110\}.$$

The multiset of discrepancies $\Delta = \{\delta^1, \ldots, \delta^{15}\}$, together with their supports and weights, is shown in Figure 1a. In what follows, we display only the discrepancies marked as unselected.

We initialize the signed vector $\Sigma = \mathbf{0}$. Next, we show the iterations necessary to determine a coverage vector.

(i) Iteration 1: δ^6 and δ^{10} have the smallest positive W, with sums of coverage potentials over the coordinates in S equal to 9 and 14, respectively. Under Rule A, $\delta^* = \delta^{10}$. We compute $p^*_5 = \frac{6}{8}$

and $p_6^* = \frac{8}{9}$, yielding the probability distribution $[0, 0, 0, 0, 0.482, 0.518]$. Suppose that, under Rule B, we randomly choose $j = 5$. We set $\Sigma_5 = \delta_5^* = +$, mark the discrepancies covered by Σ as selected, and update S and W for the discrepancies that have a sign $-$ in the 5th coordinate. The corresponding updates are shown in Figure 1b.

(ii) Iteration 2: δ^1 and δ^6 have the smallest positive W, with sums of coverage potentials over the coordinates in S equal to 10 and 5, respectively. Under Rule A, $\delta^* = \delta^1$. We compute $p_1^* = \frac{5}{5}$ and $p_2^* = \frac{5}{6}$, yielding the probability distribution $[0.51, 0.49, 0, 0, 0, 0]$. Suppose that, under Rule B, we randomly choose $j = 1$. We set $\Sigma_1 = \delta_1^* = +$, mark the discrepancies covered by Σ as selected, and update S and W for the discrepancies that have a sign $-$ in the 1st coordinate. The corresponding updates are shown in Figure 1c.

(iii) Iteration 3: δ^6 has the smallest positive W. Under Rule A, $\delta^* = \delta^6$. We compute $p_3^* = \frac{1}{3}$ and $p_6^* = \frac{2}{3}$, yielding the probability distribution $[0, 0, 0.5, 0, 0, 0.5]$. Suppose that, under Rule B, we randomly choose $j = 6$. We set $\Sigma_6 = \delta_6^* = -$, mark the discrepancies covered by Σ as selected, and update S and W for the discrepancies that have a sign $+$ in the 6th coordinate. The corresponding updates are shown in Figure 1d.

(iv) Iteration 4: δ^2 and δ^3 have the smallest positive W, both with same sum of coverage potentials equal to 6. We randomly choose $\delta^* = 2$. We compute $p_2^* = p_3^* = p_4^* = 1$ and thus all coordinates have the same probability $\frac{1}{3}$. We randomly choose $j = 2$. We set $\Sigma_2 = \delta_2^* = -$, so that there are no discrepancies to be covered, and the algorithm returns the coverage vector $\Sigma = [\,+\;-\;0\;\;0\;+\;-\,]$.

	1	2	3	4	5	6	S	W
δ^1	+	−	0	0	−	0	$\{1,2,5\}$	3
δ^2	0	−	−	−	0	0	$\{2,3,4\}$	3
δ^3	0	−	−	−	0	+	$\{2,3,4,6\}$	4
δ^4	+	0	−	−	−	0	$\{1,3,4,5\}$	4
δ^5	+	−	−	0	0	0	$\{1,2,3\}$	3
δ^6	0	0	+	0	0	−	$\{3,6\}$	2
δ^7	−	0	0	−	+	−	$\{1,4,5,6\}$	4
δ^8	−	0	0	−	+	0	$\{1,4,5\}$	3
δ^9	0	+	0	−	0	−	$\{2,4,6\}$	3
δ^{10}	0	0	0	0	+	−	$\{5,6\}$	2
δ^{11}	+	−	0	+	0	−	$\{1,2,4,6\}$	4
δ^{12}	0	−	−	0	+	−	$\{2,3,5,6\}$	4
δ^{13}	0	−	−	0	+	0	$\{2,3,5\}$	3
δ^{14}	+	0	−	0	0	−	$\{1,3,6\}$	3
δ^{15}	+	−	−	+	+	−	$\{1,2,3,4,5,6\}$	6

(a)

	1	2	3	4	5	6	S	w
δ^1	+	−	0	0	−	0	$\{1,2\}$	2
δ^2	0	−	−	−	0	0	$\{2,3,4\}$	3
δ^3	0	−	−	−	0	+	$\{2,3,4,6\}$	4
δ^4	+	0	−	−	−	0	$\{1,3,4\}$	3
δ^5	+	−	−	0	0	0	$\{1,2,3\}$	3
δ^6	0	0	+	0	0	−	$\{3,6\}$	2
δ^9	0	+	0	−	0	−	$\{2,4,6\}$	3
δ^{11}	+	−	0	+	0	−	$\{1,2,4,6\}$	4
δ^{14}	+	0	−	0	0	−	$\{1,3,6\}$	3

(b)

	1	2	3	4	5	6	S	w
δ^2	0	−	−	−	0	0	$\{2,3,4\}$	3
δ^3	0	−	−	−	0	+	$\{2,3,4,6\}$	4
δ^6	0	0	+	0	0	−	$\{3,6\}$	2
δ^9	0	+	0	−	0	−	$\{2,4,6\}$	3

(c)

	1	2	3	4	5	6	S	w
δ^2	0	−	−	−	0	0	$\{2,3,4\}$	3
δ^3	0	−	−	−	0	+	$\{2,3,4\}$	3

(d)

Figure 1: Step-by-step evolution of the discrepancy during the execution of Algorithm 1.

3.2 The exploratory algorithm

Because multiple compatible unate functions may exist, the coverage algorithm may return different solutions across runs. This variability provides an opportunity to explore the solution space and identify recurrent patterns of influence. We present Algorithm 2, whose purpose is to determine a representative vector for a collection of minimal solutions constructed for the covering algorithm.

Let $\mathcal{M} = \{\Sigma^1, \ldots, \Sigma^r\}$ denote the collection of minimal covering vectors for Δ obtained by applying Algorithm 1 independently r times. For each coordinate i, we compute the normalized frequencies of the signs $+$ or $-$ assigned to i across the elements of $\mathcal{M}$. This yields two frequency vectors, $F^+(\mathcal{M})$ and $F^-(\mathcal{M})$, defined for each i as follows:

$$F^+(\mathcal{M})_i = \frac{1}{r}\sum_{\ell=1}^{r} \mathbb{I}(\Sigma_i^\ell = +) \quad \text{and} \quad F^-(\mathcal{M})_i = \frac{1}{r}\sum_{\ell=1}^{r} \mathbb{I}(\Sigma_i^\ell = -)\,.$$

To extract a meaningful representative vector from these frequency profiles, we apply a filtering step with a user-defined threshold $p \in [0, 1]$. We define the vector $R(\mathcal{M}) \in \{0, -, +, \pm\}^n$ where $R(\mathcal{M})_i$ is $+$, $-$, or $\pm$ whenever $F^+(\mathcal{M})_i$, $F^-(\mathcal{M})_i$, or both exceed p, respectively; otherwise 0.

The choice of p is central to the outcome of the exploratory algorithm: lower values of p favor the inclusion of weaker but recurrent signals that may reflect underlying causal relationships; conversely, under limited sample size or high noise, higher values of p improve robustness by filtering less consistent attributions. Thus, the algorithm can be adjusted according to the characteristics of the dataset and the goals of the analysis, with p balancing sensitivity and robustness in the detection of patterns of influence. However, the optimal choice of p is beyond the scope of this work.

It is important to note that the representative vector $R(\mathcal{M})$ obtained by the algorithm is not necessarily a coverage vector: a unate function can be derived from it by fixing $+$ or $-$ in coordinates where $\pm$ appears, but coverage is not guaranteed. Nevertheless, $R(\mathcal{M})$ can be viewed as a quasi-solution that preserves partial coverage. This reflects the purpose of the exploratory approach: the trade-off between exact solutions and informative patterns when only observational data are available.

Algorithm 2 The exploratory algorithm

Require: Number of experiments r, filtering parameter p and the collection $\mathcal{M} = \{\Sigma^1, \ldots, \Sigma^r\}$
Ensure: A vector $R \in \{0, -, +, \pm\}^n$

1: **for** $i = 1$ **to** n **do**
2: **if** $F^+(\mathcal{M})_i \geq p \wedge F^-(\mathcal{M})_i \geq p$ **then** $R[i] \leftarrow \pm$
3: **if** $F^+(\mathcal{M})_i \geq p \wedge F^-(\mathcal{M})_i < p$ **then** $R[i] \leftarrow +$
4: **if** $F^+(\mathcal{M})_i < p \wedge F^-(\mathcal{M})_i \geq p$ **then** $R[i] \leftarrow -$
5: **if** $F^+(\mathcal{M})_i < p \wedge F^-(\mathcal{M})_i < p$ **then** $R[i] \leftarrow 0$
6: **return** R

4 Results

4.1 Stop condition

We need to stablish a practical stop condition for the coverage algorithm. To do so, we analyze how the search effort, denoted by ω, scales with the number of discrepancies d and the size of the network n using synthetic data. Here, ω is defined as the number of failed attempts before the first

valid coverage vector is found. For brevity, we will refer to this measure as *complexity* in figures and tables. We model ω as a function of problem size as:

$$\boxed{\omega = \alpha_d \cdot d + \alpha_n \cdot n + \alpha_{dn} \cdot d \cdot n,} \tag{1}$$

where the term $d \cdot n$ reflects the $O(d \cdot n)$ cost of testing whether a candidate vector covers all discrepancies.

We evaluate this model empirically for six families of unate functions: majority functions, and bounded in-degree functions with degrees from 1 and 5. Together, these families represent extreme cases in terms of in-degree: in majority functions all variables can influence the output, while in bounded in-degree functions only a few variables are active. This contrast allows us to measure ω across the spectrum of relevant structures.

To generate data under these settings, each run begins by fixing the network size n, drawn uniformly between 5 and 200. Given n, we then sample one function instance from the chosen family. For majority functions, we draw random signed vectors uniformly from $\{-,+\}^n$, with a tie-breaking rule for the discontinuity in 0. For bounded in-degree functions, we generate random unate functions within the specified in-degree class, sampled uniformly. Once a function instance is fixed, we draw an observation set of size m independently and uniformly from $\mathbb{B}^n$. From these sets, we build the multiset of discrepancies with cardinality d. For each family, we perform 3000 independent runs, discarding those with $d = 0$; and for each retained run, we record ω. We then study the scaling across function classes using linear regressions of ω on d, n, and $d{\cdot}n$. The regression coefficients for the families are summarized in Table 1.

The following are the most relevant conclusions:

(i) The coefficient of discrepancies d is consistently positive and highly significant, so the search effort grows monotonically with the number of discrepancies.

(ii) The coefficient of network size n is negative, evidencing the fact that larger networks provide more opportunities to cover discrepancies, which reduces the average complexity.

(iii) The estimated coefficients show consistent signs across the families, and their associated p values are uniformly small, indicating statistical significance. This stability suggests a reasonable degree of robustness in the empirical results.

(iv) Adjusted R^2 values are high across all families and tend to increase with in-degree. This indicates that the proposed linear specification captures most of the variance in search effort, providing a reliable empirical basis for the stop condition.

(v) The effect of d strengthens with increasing in-degree, reflecting that more complex functions require more effort to cover discrepancies. By contrast, the effect of n shows no clear trend for bounded in-degree functions. Comparing across families, the scaling observed for in-degree 5 is already close to that of majority functions, indicating that the additional density of influencers does not substantially increase complexity.

We also provide visual representations: Figure 2 shows simple regressions of ω on d for bounded in-degree 5 and for majority functions, because they represent the sparse and dense extremes of in-degree, respectively. The figures illustrate the stability of the linear trend and that the upper quantiles determine the maximum complexity observed. We also include log–log regressions of ω on d alone to test wether the scaling can be captured by a near–power law, thereby providing a more robust basis for the stop condition.

Family	Coefficient	Estimate	Std. Error	t value	Pr($>\|t\|$)
In-degree 1	α_d	9.206	0.463	19.863	$< 2e{-}16$
	α_n	-49.52	18.061	-2.742	0.0062
	α_{dn}	0.120	0.004	29.907	$< 2e{-}16$
	Adjusted $R^2 = 0.782$				
In-degree 2	α_d	13.56	0.322	42.081	$< 2e{-}16$
	α_n	-34.00	9.558	-3.557	0.00038
	α_{dn}	0.097	0.003	34.868	$< 2e{-}16$
	Adjusted $R^2 = 0.889$				
In-degree 3	α_d	14.67	0.336	43.702	$< 2e{-}16$
	α_n	-35.86	10.89	-3.292	0.0010
	α_{dn}	0.094	0.003	32.534	$< 2e{-}16$
	Adjusted. $R^2 = 0.894$				
In-degree 4	α_d	16.39	0.287	57.195	$< 2e{-}16$
	α_n	-42.32	9.876	-4.285	$1.9e{-}05$
	α_{dn}	0.085	0.002	34.422	$< 2e{-}16$
	Adjusted $R^2 = 0.917$				
In-degree 5	α_d	18.39	0.270	68.054	$< 2e{-}16$
	α_n	-19.11	9.368	-2.040	0.0415
	α_{dn}	0.072	0.002	31.312	$< 2e{-}16$
	Adjusted $R^2 = 0.934$				
Majority	α_d	19.88	0.231	86.071	$< 2e{-}16$
	α_n	-23.11	6.626	-3.489	0.00049
	α_{dn}	0.062	0.002	31.244	$< 2e{-}16$
	Adjusted $R^2 = 0.951$				

Table 1: OLS regression coefficients for bounded in-degree 1 to 5, and majority functions. Each block shows coefficient estimates with standard errors, t values and p-values; adjusted R^2 is reported in the last row of each block.

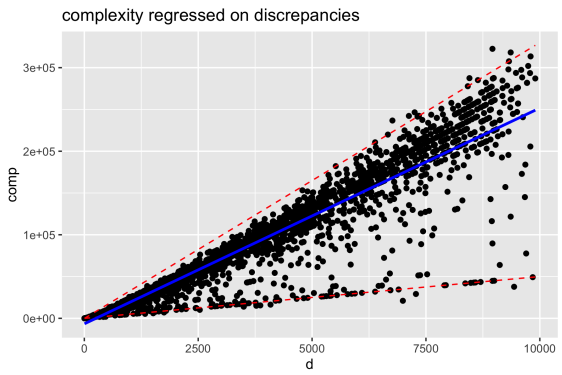
The above results indicate that a logarithmic analysis is more appropriate for advancing toward a practical stop condition. The dependence of ω on discrepancies is nearly linear in the upper tail, with slope $\gamma \approx 1.16$ on majority functions. This motivates a power-law on the number of discrepancies, which we extend to a multiplicative form in both d and n: $\omega \sim n^{\beta_1} d^{\beta_2}$. Concretely, we fit a $\tau = 0.999$ quantile regression on the majority family with predictors $\log d$ and $\log n$ (the interaction $d \cdot n$ is omitted in log scale due to collinearity). This yields:

$$\boxed{\log(\omega) \;=\; 1.78277 \;+\; 1.15688\,\log(d) \;+\; 0.07946\,\log(n)}\,.$$

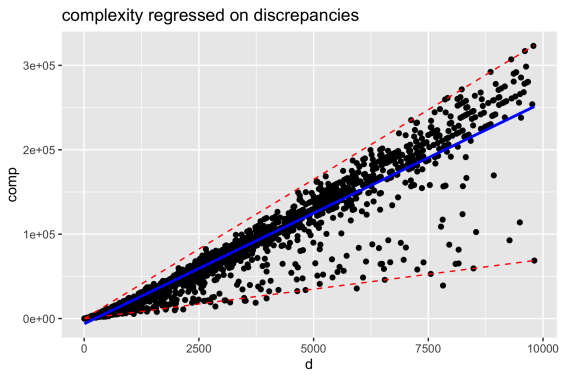
For the range of our experiments ($5 \le n, m \le 200$), less than 0.1% of cases exceed this bound, so we adopt it as a conservative stop condition. In the main text, we extrapolate this bound to larger networks as a pragmatic approximation, supported by the approximately power-law relation between discrepancies and complexity.

4.2 Synthetic datasets

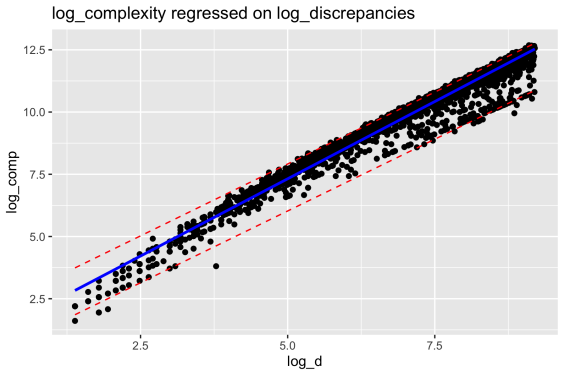
To evaluate the computational behavior of the coverage algorithm, we generated synthetic datasets based on random Boolean networks (BNs) constructed using the R-package *BoolNet*. The procedure



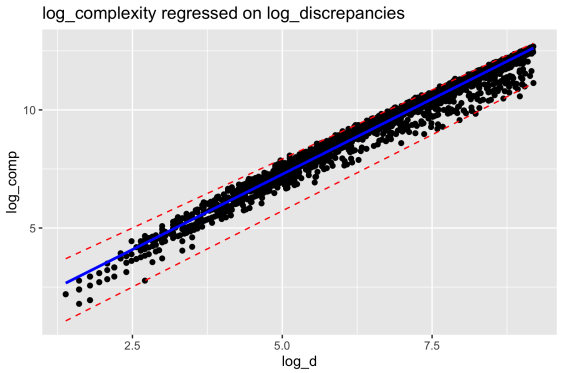
(a) OLS regression (blue) of complexity on discrepancies for in-degree 5, with 0.99 and 0.01 quantile regressions (red).



(b) OLS regression (blue) of complexity on discrepancies for majority functions, with 0.99 and 0.01 quantile regressions (red).



(c) Log–log regression of complexity on discrepancies for in-degree 5. OLS in blue; 0.99 and 0.01 quantile regressions in red.



(d) Log–log regression of complexity on discrepancies for majority functions. OLS in blue; 0.99 and 0.01 quantile regressions in red.

Figure 2: Complexity regressed on discrepancies for representative families. Bounded in-degree 5 (sparse) and majority functions (dense) are displayed. Blue lines correspond to OLS regressions; red lines to upper (0.99) and lower (0.01) quantile regressions.

is designed to produce sets of observations under the following main parameters: the number of variables (n), the number of observations (m), and the number of discrepancies (d).

For each parameter configuration, we generate a single random BN:

$$\mathcal{N} : \mathbb{B}^n \to \mathbb{B}^n, \qquad x = (x_1, \ldots, x_n) \in \mathbb{B}^n \;\mapsto\; \mathcal{N}(x) = (f_1(x), \ldots, f_n(x)),$$

where each coordinate function $f_i : \mathbb{B}^n \to \mathbb{B}$ is the *local activation function* of node i. For every $i \in [n]$, we define the *dependency set* $\mathcal{D}_i \subseteq [n]$ such that f_i depends only on the variables in $\mathcal{D}_i$; the *in-degree* of node i is then given by $|\mathcal{D}_i|$. In this work, we consider random BNs with bounded in-degree, meaning that all local activation functions have a maximum in-degree k.

Each BN $\mathcal{N}$ is generated in R using the following command:

```
generateRandomNKNetwork(n, k, simplify = TRUE, readableFunctions = TRUE),
```

where the option `simplify = TRUE` ensures that redundant dependency variables are removed for each f_i, and `readableFunctions = TRUE` displays the DNF representation for the local functions. Other parameters are set to their default values:

`topology = "fixed"`, `linkage = "uniform"`, `functionGeneration = "uniform"`

For further details, see the documentation `https://cran.r-project.org/web/packages/BoolNet/index.html`.

For a BN $\mathcal{N} = (f_1, \ldots, f_n)$, a *trajectory* is a sequence of $m+1$ states in $\mathbb{B}^n$, denoted by $\mathcal{T} = (x(0), x(1), \ldots, x(m))$, which evolves according to the rule:

$$\forall t \in \{0, 1, \ldots, m-1\},\ x(t+1) = \mathcal{N}(x(t)) = \big(f_1(x(t)), \ldots, f_n(x(t))\big)\,,$$

known as the *synchronous update scheme*.

For each BN generated, we simulated 100 trajectories of length $m+1$, each initialized from a distinct randomly drawn state using the following command:

```
generateTimeSeries(net, 100, m, type = "synchronous")
```

where `net` represents the random BN, and the option `type = "synchronous"` specifies the update scheme.

For each trajectory $\mathcal{T}^i$ of length $m+1$, starting from the initial state $x^i(0)$, we construct the corresponding set of observations F^i as:

$$F^i = \{\big(x^i(t), f_i(x^i(t))\big)\}_{t=0}^{m-1}\,.$$

where $x^i(t) \in \mathbb{B}^n$ is the state of the entire BN at time t, and $f_i(x^i(t))$ is the value of node i at the next time step. Then, F^i is partitioned into F_0^i and F_1^i according to the value of $f_i(x^i(t))$.

This construction, which uses only the evolution of node i along trajectory $\mathcal{T}^i$, ensures that each F^i originates from a distinct trajectory starting from a different initial state, thereby reducing overlap among the sets of observations and providing an approximate form of independence, even though all trajectories are derived from the same BN.

4.2.1 Scalability with respect to n and m

The practical applicability of the coverage algorithm depends on the behavior of the execution time as the input size increases. Thus, we evaluate the execution time for different values of n and m. The parameter settings are given by all pairs (n, m) with $n \in \{100, 200, 300, 400, 500, 600\}$ and $m \in \{50, 100, 150, 200, 250, 300\}$, where the sets of observations F_0 and F_1 are balanced, i.e., $|F_0| \approx |F_1| \approx m/2$. Because our algorithm is more sensitive to d than to m, choosing m provides a consistent basis for comparison.

Figure 3 summarizes the results, showing the average execution time as a function of m for different values of n. The algorithm maintains practical execution times even for $n = 600$, demonstrating scalability across the tested range. As expected, the execution time grows with both n and m, but the increase is more pronounced with respect to m. This asymmetry arises because d, which grows quadratically with m, dominates the total computational cost, whereas the contribution of n is only linear. For small m ($m \leq 100$), the execution times are similar across different n; for larger m ($m > 100$), the curves exhibit comparable trends, but the differences in the execution times among n become more significant. Finally, at higher values of n, the observed growth becomes more pronounced, indicating that additional factors, possibly related to the structure of the discrepancies, contribute to the performance of the execution times.

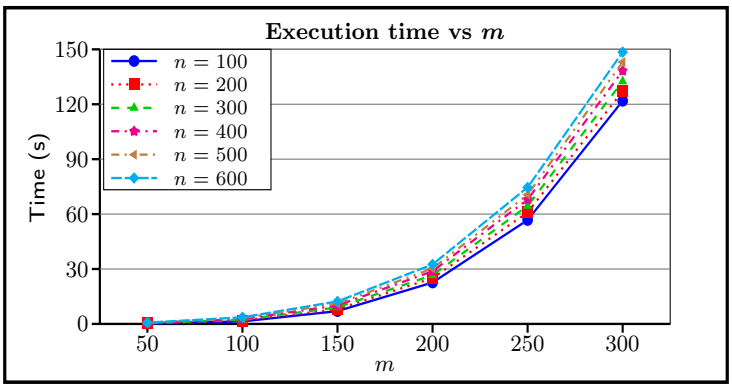


Figure 3: Average execution time of Algorithm 1 as a function of the number of observations m, for different values of the number of variables n. For each pair (n, m), the reported execution time corresponds to the average across the dataset of 100 instances generated. Every run successfully returned a coverage vector, and no failures were reported.

4.2.2 Feasibility under bounded coverage weight

In many applications, particularly in biological network modeling, relevant solutions involve only a few influencers. To explore the behavior of the coverage algorithm in this scenario, we evaluate the execution time and success rate under the constraint that solutions have a weight bounded by $k \in \{3, 5, 10\}$. The parameter settings are defined by all pairs (n, d) with $n = 50$ fixed and $d \in \{d_1, \ldots, d_{18}\}$, where each d_i denotes an interval $d_i = \big(100(i-1), 100i\big]$ for $i \in \{1, \ldots, 18\}$. Because d varies with m when the number of influencers k is small, grouping by intervals provides a consistent basis for comparison.

Since the bounding restriction significantly increases the difficulty of finding a solution, with frequent restarts when the weight exceeds k, especially for small values of k, we employed a relaxed version of the stopping condition, replacing $\log(\omega)$ with $\log(\omega) \cdot \log(5)$ for $k = 3$ and $k = 5$, a factor chosen empirically to expand the search. In contrast, for $k = 10$, the constraint is less strict, so the original $\log(\omega)$ rule is sufficient.

Figure 4 reports the average execution times as a function of the discrepancy intervals d_i, together with the corresponding success rates. Across all values of k, execution time increases almost steadily with d, remaining within a moderate range. For $k = 5$, the times are slightly smaller than for $k = 3$, whereas for $k = 10$ the reduction is considerably more pronounced. A different behavior is observed for the success rates across the values of k. For $k = 3$, the success rate declines as the multiset of discrepancies grows, but the decrease is not strictly monotonic: certain intervals with higher values of d still exhibit substantial success rates, suggesting that feasibility may depend not only on d but also on its internal structure, such as redundancies among discrepancies. For $k = 5$, the success rate also decreases with increasing d but tends to stabilize toward the higher intervals. For $k = 10$, it remains nearly constant and significantly higher than in the other cases. These differences reflect the fact that the bounds $k = 3$ and $k = 5$ impose stricter constraints on feasible solutions, whereas $k = 10$ allows substantially greater flexibility.

4.3 Real datasets: Drosophila ventral furrow

We consider the *Drosophila* ventral furrow regulatory BN published by Aracena et al. [3], for which both the interaction graph and the Boolean update rules are available. The BN topology is shown in Figure 5 and the corresponding Boolean functions are listed in Table 2. This model provides a

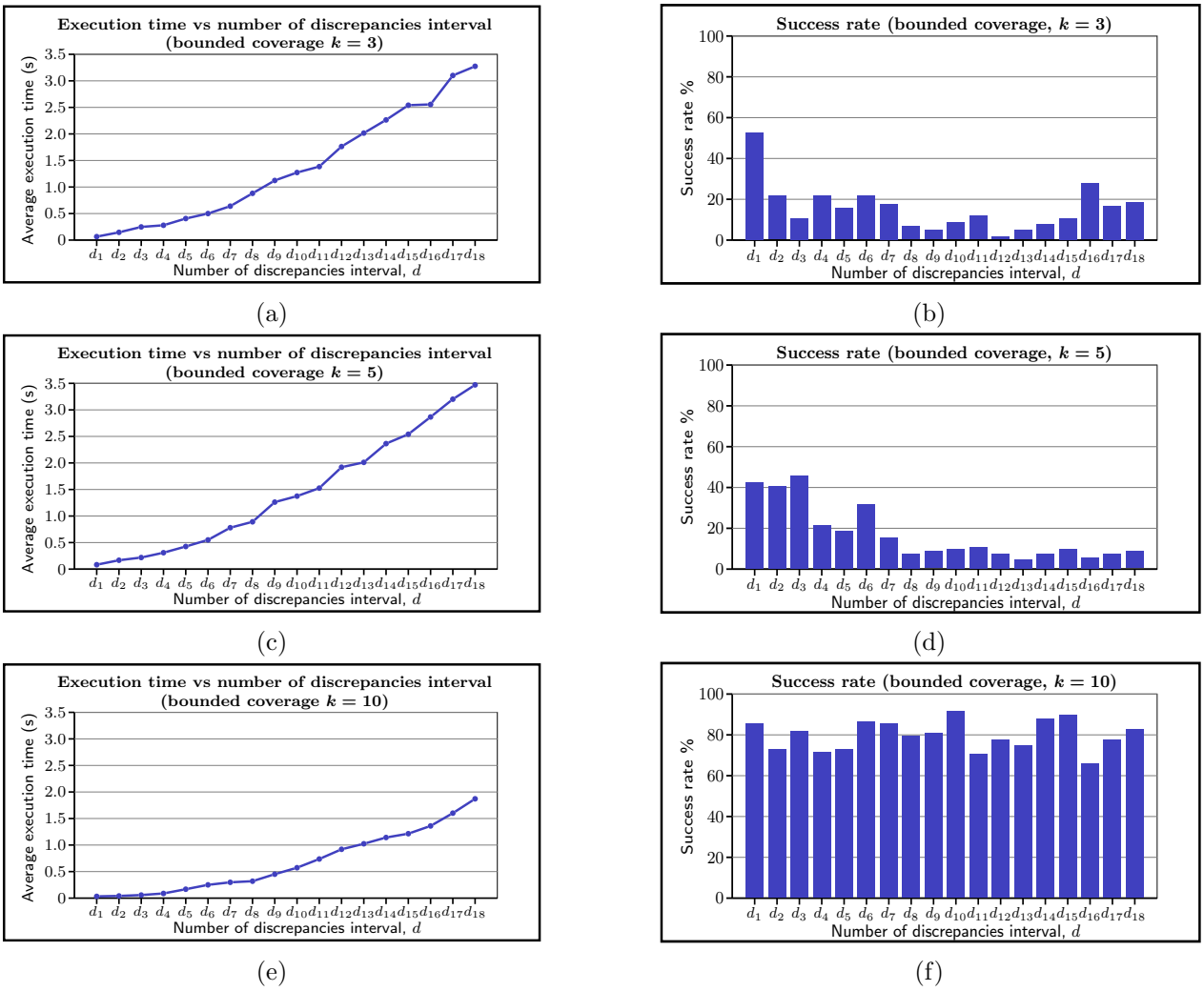


Figure 4: Average execution time and success rate of Algorithm 1 under bounded coverage with $k = 3$ ((a)-(b)), $k = 5$ ((c)-(d)) and $k = 10$ ((e)-(f)), all measures as a function of the discrepancy intervals d_i. For each interval, the reported execution time corresponds to the average across the dataset of 100 instances generated; and the percentage corresponds to the proportion of bounded coverage vectors that were found.

concrete ground truth for evaluating our algorithms on a biologically meaningful problem.

We constructed two random BN trajectories of lengths 21 and 51. From each trajectory, we extracted the observations for the update functions of each node, obtaining sets of observations of sizes 20 and 50, respectively. For each node, we applied the coverage algorithm 1000 times and generated a collection of minimal solutions with weights bounded by $k = 3$. Finally, we applied the exploratory algorithm to construct a representative vector that satisfies the coverage property and inferred the lower bound function described in Proposition 3.

The results show that the inferred functions preserve the structure of the BN in terms of the number of influencers. Nodes with an in-degree equal to one or two are consistently identified, confirming that even with a limited number of observations, our procedure finds simple dependencies. On the other hand, the inferred functions for the node with an in-degree equal to three (Rho) are also consistently identified, and they differ because the number of coverage vectors tends to be

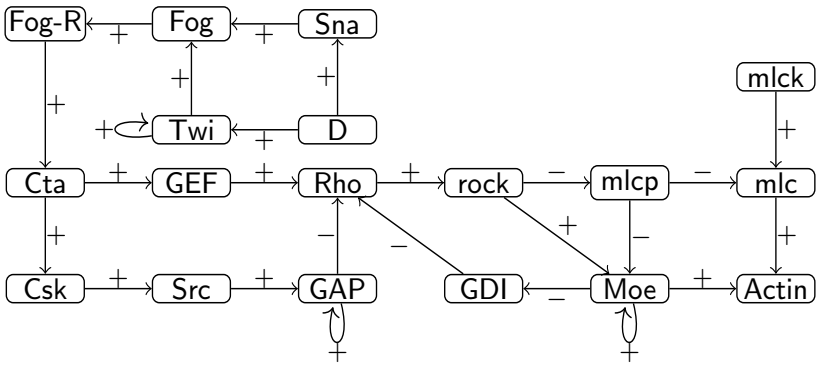


Figure 5: *Drosophila* ventral furrow regulatory BN topology.

Gene	True f	Inf. f (20 obs)	Inf. f (50 obs)
D	1	1	1
Twi	Twi∨D	Twi∨D	Twi∨D
Sna	D	D	D
Fog	Twi∨Sna	Twi∨Sna	Twi∨Sna
FogR	Fog	Fog	Fog
Cta	FogR	FogR	FogR
GEF	Cta	Cta	Cta
Csk	$\overline{\text{Cta}}$	$\overline{\text{Cta}}$	$\overline{\text{Cta}}$
Src	$\overline{\text{Csk}}$	$\overline{\text{Csk}}$	$\overline{\text{Csk}}$
GAP	Src∨GAP	$\overline{\text{Src}}$	Src
Rho	$\overline{\text{GDI}}\wedge\text{GEF}\wedge\overline{\text{GAP}}$	Moe∧$\overline{\text{FogR}}$∧Csk	$\overline{\text{GDI}}\wedge\text{GEF}\wedge\overline{\text{GAP}}$
rock	$\overline{\text{Rho}}$	$\overline{\text{Rho}}$	$\overline{\text{Rho}}$
GDI	$\overline{\text{Moe}}$	$\overline{\text{Moe}}$	$\overline{\text{Moe}}$
Moe	rock∧$\overline{\text{mlcp}}$∧Moe	rock∧$\overline{\text{mlcp}}$∧GAP	rock∧$\overline{\text{mlcp}}$∧Moe
mlcp	$\overline{\text{rock}}$	$\overline{\text{rock}}$	$\overline{\text{rock}}$
mlck	1	1	1
mlc	$\overline{\text{mlcp}}$∧rock∧mlck	$\overline{\text{mlcp}}$∧rock∧mlck	$\overline{\text{mlcp}}$∧rock∧mlck
Actin	Moe∧mlc	Moe∧mlc	Moe∧mlc

Table 2: Inference of the Drosophila regulatory BN with 20 and 50 observations and bounded weight restriction $k = 3$.

higher when there are few observations from functions with larger in-degrees. In general, this case study indicates that our exploratory approach is well suited for inferring a compatible regulatory BN when available information is limited to observations. This is mainly because the algorithms presented are based on techniques that favor sparse solutions, i.e., involving few influencers, and their rapid convergence allows the construction of large collections of solutions in a reasonable time.

5 Discussion

From a theoretical perspective, we present a framework that connects observational data with compatible unate functions through the notions of multisets of discrepancies and coverage vectors. In this way, the CUF problem is reformulated as the search for a coverage vector, and we provide the basis for the inference problem: once such a vector is found, the corresponding set of compatible unate functions can be characterized. Furthermore, the notions of minimal and minimum subcoverage highlight the computational limits involved in simplifying coverage vectors.

On a practical level, we establish an exploratory approach that integrates a coverage algorithm and an exploratory algorithm. The coverage algorithm exhibits scalability with execution times remaining convenient even for functions with several hundred variables and growth driven primarily by the number of discrepancies. In addition, it is flexible enough to incorporate prior knowledge about specific influencers or constraints on the weight of the solutions. These features enable the generation of a large collection of solutions within a reasonable time frame, which is analyzed by the exploratory algorithm to identify recurring patterns of influence and derive a representative quasi-solution. In the context of BN inference, this approach is effective when the information is limited to a small number of observations; although its node by node design does not directly exploit global dependencies, the results indicate its potential to systematically explore plausible regulatory architectures and guide the future refinement of the model.

There are several ways to expand this work. Beyond unate functions, it is natural to ask whether this approach can be extended to canalizing functions ([12]), whose stability makes them relevant as models for key regulatory genes. At the level of observations, one direction is to explore how the framework could be adapted to observations with missing bits ([4]) or to non-Boolean observations with continuous expression values ([10]). Finally, the problem of efficiently inferring all compatible unate functions remains largely unexplored ([7]).

6 Acknowledgments

A.P. acknowledges valuable discussions with Laurent Tournier, Loïc Paulevé, Claudine Chaouiya, Pedro Monteiro and Michel Grabisch.

7 Funding

This work was supported by the *Agencia Nacional de Investigación y Desarrollo* (ANID-Chile)[FB210005 to J.A and A.P] through the Basal Project for the *Centro de Modelamiento Matemático* (CMM); *Agencia Nacional de Investigación y Desarrollo* (ANID-Chile)[21212217 to K.H] through the Beca Doctorado Nacional; and *Agence de l'Innovation de Défense*[to A.P] through the Polaris Project.

References

[1] T. Akutsu, S. Miyano, and S. Kuhara. Algorithms for identifying boolean networks and related biological networks based on matrix multiplication and fingerprint function. *Journal of Computational Biology*, 7:331–344, 2000.

[2] M. Anthony. *Discrete Mathematics of Neural Networks. Selected Topics.* SIAM Monographs on Discrete Mathematics and Applications, 2001.

[3] J. Aracena, M. González, A. Zuniga, M. Mendez, and V. Cambiazo. Regulatory network for cell shape changes during the drosophila ventral furrow formation. *Journal of Theoretical Biology*, 239:49–62, 2006.

[4] E. Boros, T. Ibaraki, and M. Kazuhisa. Fully consistent extensions of partially defined boolean functions with missing bitsv. In *Theoretical Computer Science: Exploring New Frontiers of Theoretical Informatics*, pages 257–272, 2000.

[5] Y. Crama and P. L. Hammer. *Boolean functions: Theory, algorithms, and applications*. Cambridge University Press, 2011.

[6] Y. Crama, P. L. Hammer, and T. Ibaraki. Cause-effect relationships and partially defined boolean functions. *Annals of Operations Research*, 28(16):299–325, 1988.

[7] J. E. R. Cury, P. Tenera Roxo, V. Manquinho, C. Chaouiya, and P. T. Monteiro. Computation of immediate neighbours of monotone boolean functions. In *Computational Methods in Systems Biology*, pages 3–22. Springer Nature Switzerland, 2025.

[8] J. Grefenstette, S. Kim, and S. Kauffman. An analysis of the class of gene regulatory functions implied by a biochemical model. *Biosystems*, 84(2):81–90, 2006.

[9] G. J. Hickman and T. C. Hodgman. Inference of gene regulatory networks using boolean-network inference methods. *Journal of Bioinformatics and Computational Biology*, 7:1013–1029, 2009.

[10] M. Hopfensitz, C. Mussel, C. Wawra, M. Maucher, M. Kuhl, H. Neumann, and H. A. Kestler. Multiscale binarization of gene expression data for reconstructing boolean networks. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 9(2):487–498, 2012.

[11] S. Kauffman. Homeostasis and differentiation in random genetic control networks. *Nature*, 224: 177–178, 1969.

[12] S. Kauffman, C. Peterson, B. Samuelsson, and C. Troein. Genetic networks with canalyzing boolean rules are always stable. *PNAS*, 101(49), 2004.

[13] M. Lejeune, V. Lozin, I. Lozina, A. Ragab, and S. Yacout. Recent advances in the theory and practice of logical analysis of data. *European Journal of Operational Research*, 275(1):1–15, 2019.

[14] A. Poindron. The maximal coordination principle in regulatory boolean networks. *Journal of Computer and System Sciences*, 142:103518, 2024.

[15] R. Porreca, E. Cinquemani, J. Lygeros, and G. Ferrari-Trecate. Identification of genetic network dynamics with unate structure. *Bioinformatics*, 26(9):1239–1245, 2010.

[16] Ž. Pušnik, M. Mraz, N. Zimic, and M. Moškon. Review and assessment of boolean approaches for inference of gene regulatory networks. *Heliyon*, 2022.

[17] M. Saint-Antoine and A. Singh. Network inference in systems biology: recent developments, challenges, and applications. *Current Opinion in Biotechnology*, 2020.

[18] L. Tournier, A. Goelzer, and V. Fromion. Optimal resource allocation enables mathematical exploration of microbial metabolic configurations. *Journal of Mathematical Biology*, 75(6): 1349–1380, 2017.

Centro de Investigación en Ingeniería Matemática (CI^{2}MA)

PRE-PUBLICACIONES 2025

2025-16 Juan Barajas-Calonge, Raimund Bürger, Pep Mulet, Luis M. Villada: *A second-order invariant-region-preserving scheme for a transport-flow model of polydisperse sedimentation*

2025-17 Raimund Bürger, Stefan Diehl, María Carmen Martí, Yolanda Vásquez: *A numerical scheme for a model of a flotation column including the transport of liquid components*

2025-18 Raimund Bürger, Enrique D. Fernández Nieto, José Garres-Díaz, Jorge Moya: *Well-balanced physics-based finite volume schemes for Saint-Venant-Exner-type models of sediment transport*

2025-19 Harold D. Contreras, Paola Goatin, Luis M. Villada: *Well-posedness of a nonlocal upstream-downstream traffic model*

2025-20 Thierry Coulbois, Anahi Gajardo, Pierre Guillon, Victor H. Lutfalla: *Aperiodic monotiles: from geometry to groups*

2025-21 Esteban Henriquez, Tonatiuh Sanchez-Vizuet, Manuel Solano: *An unfitted HDG discretization for a model problem in shape optimization*

2025-22 Fernando Artaza-Covarrubias, Tonatiuh Sanchez-Vizuet, Manuel Solano: *A coupled HDG discretization for the interaction between acoustic and elastic waves*

2025-23 Eider Aldana, Ricardo Oyarzúa, Jesús Vellojín: *Numerical analysis of a three-field formulation for a reverse osmosis model*

2025-24 Julio Aracena, Florian Bridoux, Maximilien Gadouleau, Pierre Guillon, Kevin Perrot, Adrien Richard, Guillaume Theyssier: *On the Dynamics of Bounded-Degree Automata Networks*

2025-25 Daniel Cajas Guijarro, John Cajas Guijarro: *Resonant and non-resonant double Hopf bifurcation in a 4D Goodwin model with wage inequality*

2025-26 Alonso J. Bustos, Sergio Caucao: *A Banach space mixed formulation for the unsteady Brinkman problem with spatially varying porosity*

2025-27 Julio Aracena, Katerin De La Hoz, Alexis Poindron, Lilian Salinas: *An exploratory approach to the compatibility and inference of unate functions*

Para obtener copias de las Pre-Publicaciones, escribir o llamar a: Director, Centro de Investigación en Ingeniería Matemática, Universidad de Concepción, Casilla 160-C, Concepción, Chile, Tel.: 41-2661324, o bien, visitar la página web del centro: `http://www.ci2ma.udec.cl`



Centro de Investigación en
Ingeniería Matemática (CI²MA)
Universidad de Concepción

Casilla 160-C, Concepción, Chile
Tel.: 56-41-2661324/2661554/2661316
http://www.ci2ma.udec.cl