



HAL
open science

A Linear Algorithm For Computing Polynomial Dynamical Systems

Ines Abdeljaoued-Tej, Alia Benkahla, Ghassen Haddad, Annick Valibouze

► **To cite this version:**

Ines Abdeljaoued-Tej, Alia Benkahla, Ghassen Haddad, Annick Valibouze. A Linear Algorithm For Computing Polynomial Dynamical Systems. 2018. hal-01890698

HAL Id: hal-01890698

<https://hal.science/hal-01890698v1>

Preprint submitted on 8 Oct 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Linear Algorithm For Computing Polynomial Dynamical Systems

October 7, 2018

Ines Abdeljaoued-Tej^{1,*}, Alia Benkahla², Ghassen Haddad³, Annick Valibouze⁴

1 University of Tunis El Manar, Laboratory of BioInformatics, bioMathematics and bioStatistics (BIMS) in Institute Pasteur of Tunis, 1002 Tunis and University of Carthage, ESSAI; Tunisia

2 Laboratory of BioInformatics, bioMathematics and bioStatistics (BIMS), Institute Pasteur of Tunis, University of Tunis El Manar; Tunisia

3 University of Tunis El Manar, Laboratory of BioInformatics, bioMathematics and bioStatistics (BIMS), Institute Pasteur of Tunis and ENIT; Tunisia

Sorbonne University, Laboratory Jacques Louis Lions (LJLL); France

4 Sorbonne University, Laboratoire d'Informatique de Paris 6 (LIP6) and Laboratoire de Statistique Théorique et Appliquée (LSTA); France

* inestej@gmail.com

Abstract

Computation biology helps to understand all processes in organisms from interaction of molecules to complex functions of whole organs. Therefore, there is a need for mathematical methods and models that deliver logical explanations in a reasonable time. For the last few years there has been a growing interest in biological theory connected to finite fields: the algebraic modeling tools used up to now are based on Gröbner bases or Boolean group. Let n variables representing gene products, changing over the time on p values. A Polynomial dynamical system (PDS) is a function which has several components; each one is a polynom with n variables and coefficient in the finite field $\mathbb{Z}/p\mathbb{Z}$ that model the evolution of gene products. We propose herein a method using algebraic separators, which are special polynomials abundantly studied in effective Galois theory. This approach avoids heavy calculations and provides a first Polynomial model in linear time.

Keywords: Polynomial dynamical system; Finite field; Mathematical model; Gene regulatory network.

Code: 37-XX, 37-E-15, 12E20, 12E05, 11Txx, 92C42, 37N25, 12F10, 12Y05, 11Y40

1 Introduction

A living cell can be compared to a complex factory animated by DNA and large amounts of biological data are now available. Therefore, the challenge is to extract biological meaning: it consists of developing methodologies for using these data to address biological questions [15]. System biology is a study of several networks related to biology data. We choose gene expression data to design regulatory gene network and to build a mathematical model from observations of the system response to well-constructed perturbations. The data are measurements of concentrations of biochemicals, recorded at time intervals: we propose an adjusted model, defined in a finite field, that fits these data.

Several methods have been proposed to infer gene interrelations from expression data. To solve this issue, many researchers have proposed various approaches [21]. In fact, mathematical network modelling is an important step toward covering the dynamic behaviour of biological networks. New areas of mathematics, not traditionally considered applicable, are now contributing with powerful tools. Some applied research were focused on investigating dynamical properties. They propose original analysis for regulatory interactions where Boolean models were generated [20, 11]. A linear system of differential equations, obtained from measured gene expression, can be used to infer gene regulatory network [6].

Bayesian networks are used to measure gene expression data [12], as well as modeling gene expression data [9].

Polynomial dynamical system is an approach used for understanding the behavior of complex systems over time [26]. All these approaches deal with biological systems which studies and describes the interactions between micro-biological outputs. It finds its roots in symbolic computation and mathematical modelling. One of the precursors of Polynomial dynamical system is R. Thomas with his Boolean dynamical system [25]. In the last decade, several studies have been made, including contributions in Hybrid systems biology [4]. The main objective of the paper is to define an algorithm which computes biological systems, over finite fields, in linear time. Our approach is part of a broader framework of a multidisciplinary team working on an algebraic modelling as well as based on EDO or Bayesian networks. With this in mind, we started in [3] by presenting a summary of the main methods adapted to this framework. The algebraic method presented in this paper adapts the techniques of Galois theory to issues of bio-systems. This approach can effectively model the important size of the biological data, with relatively simple tools: the calculation of elementary symmetric polynomials in the case of boolean or the fundamental modules in the polynomial case.

The first use of Polynomial dynamical system (PDS) on System biology was published by R. Laubenbacher and B. Stigler [17, 23]. Their model is a deterministic graphical model which depends on the degree of data discretization p ($p = 2$ in boolean modelling). For example, their model describes the relationships between diseases and symptoms by producing dependency graphs (also known as the wiring diagram [19]) and space state graphs. Given symptoms, the space-state graph can be used to compute the presence of various diseases. Thus, when n entries evolve in time, the dynamical function can be represented by n polynomials which describe a table of p^n possible state. Our researches were inspired by a classical method based on Lagrange interpolation [16]. We propose herein two methods using algebraic separators, both are special polynomials abundantly studied in effective Galois theory. Algebraic separators are directly determined by using symmetric functions, or by linear combinations of fundamental modules. These approaches avoid heavy calculations of Gröbner bases and provide a first Polynomial model in linear time, similar to the one produced by the team of R. Laubenbacher. The computation of a first Polynomial dynamical system, as detailed in this article, can perfectly be computed in parallel. Tools based on differential equations, Bayesian networks or Boolean networks are commonly used to model biological networks [5]. Unlike these techniques and despite the important work done within R. Laubenbacher's team [14], the Polynomial modelling remains underexploited. Very few publications are available in the literature that addresses the issue of algebraic modelling in biology. Because of the large amount of information present in a PDS, other network forms can easily be derived from it. Further work is needed to discern models that balance these needs optimally. Issues regarding model personalization and boundary condition tuning are particularly important. Instead, we use computational algebra to discuss the biochemical networks using the example of gene regulatory networks.

We propose a linear algorithm that performs learning in Polynomial dynamical systems. The paper offers an effective alternative for Gröbner basis when modeling biological systems. We introduce a novel method of modelling based on polynomial over finite fields. The remainder of the paper is organized as follows: Section 2 details an approach allowing separators computation: we start by introducing useful definitions and notations. Section 2.1 presents a method based on Galois theory tools as the *fundamental modules* or *elementary symmetric functions*. Section 2.2 is devoted to introduce a main theorem for computing algebraic separators with optimal degree; it allows us to compute a finite number of other Polynomial dynamical systems. Section 2.3 summarizes the particular case of Boolean dynamical Systems. Experimental results are given in Section 2.4: some basic techniques that enable us to compute affine separators. Section 3 gives an algebraic technique for defining different types of rules between genes. We illustrate our algorithm with a network where states are in a finite field. The main feature of the data generated by DNA microarrays, called *gene expression data*, is that they have few experiments with regard to the number of genes tested simultaneously. Unfortunately, it is difficult to get all the data's necessary information to verify if the model is valid. In order to propose a mathematical model based on the polynomial dynamical systems, we discretize the data by using a double filter for changing genes expressions. The data we have chosen to study, are divided in two lines for two different media. Taking into account the expression data of genes over time, we discusses an example of computing a Polynomial dynamical system for each cell line computed in linear time. Finally, Section 4 gives the conclusion with the perspective.

2 Polynomial dynamical system

In the following, let p be a prime number and k be a finite field with p elements: k is also noted as $\mathbb{Z}/p\mathbb{Z}$. Let $\mathcal{E} = k^n$ be the set of p^n states. A *Polynomial dynamical system of dimension n* , denoted by PDS, is a polynomial function $f = (f_1, \dots, f_n)$ whose components f_i are polynomials of the quotient ring $k[x_1, \dots, x_n]/\langle x_1^p - x_1, \dots, x_n^p - x_n \rangle$, i.e., polynomials on n variables with coefficients in k and degree smaller than $p - 1$ in each variable.

For example, the data commonly used in the modelling of biological systems are concentrations of n proteins and the change of these concentrations over m time points. We usually need normalized data but even if normalization is sufficient for continuous models, it is inappropriate for discrete models, as it is for Boolean or Polynomial Networks. Indeed, in the latter cases, a discretization is also necessary. It can be done using E. Dimitrova's method [8]. The latter gives an integer p , called degree of discretization, and replaces each value by x_j with $j \in [1, n]$, a corresponding value in $\mathbb{Z}/p\mathbb{Z}$. At each time $t \in [1, m]$, the vector $\mathbf{s}_t = (x_1, x_2, \dots, x_n)$ is called *the state of the system at time t* . The finite trajectory $\mathbf{s}_1 \mapsto \mathbf{s}_2 \mapsto \dots \mapsto \mathbf{s}_m$ is called *a discrete trajectory of length m* . We consider a PDS, which is a polynomial function $f = (f_1, \dots, f_n)$ satisfying, for time $i \in [1, m - 1]$:

$$f(\mathbf{s}_i) = \mathbf{s}_{i+1}. \quad (1)$$

There is an infinite number of such PDS because it is a solution of a multivariable interpolation problem. In fact, let $I(V) = \{h \in k[x_1, \dots, x_n] \mid h(\mathbf{s}) = 0, \forall \mathbf{s} \in V\}$ be the ideal of affine variety $V \subset \mathcal{E}$, $j \in [1, n]$. Let $j \in [1, n]$. For each $i \in [1, m - 1]$, $f_j(\mathbf{s}_i) = \mathbf{s}_{i+1, j}$. For every $g \in k[x_1, \dots, x_n]$ there exists $h \in I(V)$ such as $g = f_j + h$. The polynomial $f_j - g \in I(V)$ verifies also $g(\mathbf{s}_i) = \mathbf{s}_{i+1, j}$ for $i \in [1, m - 1]$.

It is possible to separate an element of V with respect to others thanks to a polynomial of $k[x_1, \dots, x_n]$:

Definition 1. Let $V \subset \mathcal{E}$ and \mathbf{s} a state in V . A polynomial $r(\mathbf{x}) \in k[x_1, \dots, x_n]$ which equals 1 at $\mathbf{x} = \mathbf{s}$ and 0 in the other elements of V is called a (polynomial) *separator of \mathbf{s} in V* .

Separators have the same purpose as Lagrange's polynomials in a univariate interpolation problem. For example, following the results of [18] and taking into account a Gröbner basis's computation of the ideal $I(V)$, we obtain separators r_j of \mathbf{s}_j in V for $j \in [1, m - 1]$:

$$r_1(x_1, x_2, x_3) = -2x_1x_3 - 2x_2x_3 + 2x_3^2 \quad (2)$$

$$r_2(x_1, x_2, x_3) = x_2^2 - 2x_2x_3 + 2x_3^2 \quad (3)$$

$$r_3(x_1, x_2, x_3) = -2x_1 + 2x_2 + x_3 \quad (4)$$

$$r_4(x_1, x_2, x_3) = -x_2x_3 - 2x_3^2. \quad (5)$$

We compute a PDS $f = (f_1, \dots, f_n)$ by interpolation resolution based on separators: starting from the respective concentrations $(4, 3, 1, 0)$, $(4, 3, 2, 0)$ and $(0, 1, 3, 4)$ of proteins 1, 2 and 3, we compute a PDS verifying $f_j(\mathbf{s}_i) = \mathbf{s}_{i+1, j}$ for $j = 1, 2, 3$ and $i = 1, 2, 3, 4$:

$$f_1(x_1, x_2, x_3) = 4r_1 + 3r_2 + 1r_3 + 0r_4 = 3x_2^2 + 2x_1x_3 + x_2x_3 - x_3^2 + 3x_1 + 2x_2 + x_3 \quad (6)$$

$$f_2(x_1, x_2, x_3) = 4r_1 + 3r_2 + 2r_3 + 0r_4 = 3x_2^2 + 2x_1x_3 + x_2x_3 - x_3^2 + x_1 - x_2 + 2x_3 \quad (7)$$

$$f_3(x_1, x_2, x_3) = 0r_1 + 1r_2 + 3r_3 + 4r_4 = x_2^2 - x_2x_3 - x_3^2 - x_1 + x_2 + 3x_3 \quad (8)$$

From $f = (f_1, f_2, f_3)$, we can deduce all the trajectories using DVD developed in [32]. This example illustrates a Polynomial Networks reproduced by the Buchberger-Möller algorithm which is implemented in Computer Algebra Systems like Macaulay2 [10] or Cocoa [1]. R. Laubenbacher and B. Stigler had discussed the Lagrange interpolation without pushing the analysis to develop a specific method for computing algebraic separators. However, they adapted Gröbner's basis computation of $I(V)$ to all PDS, and in particular, to obtain algebraic separators in $o(n^2m^3 + nm^2)$.

2.1 Algebraic Separators

Let $\mathbf{s} = (s_1, \dots, s_n) \in \mathcal{E}$ where $\mathcal{E} = k^n$ is the set of p^n states. In $k[x_1, \dots, x_n]$, the maximal ideal $\mathfrak{M}_{(\mathbf{s})}$ of \mathbf{s} -relations is generated by

$$p_1 = x_1 - s_1, \quad p_2 = x_2 - s_2, \quad \dots \quad p_n = x_n - s_n. \quad (9)$$

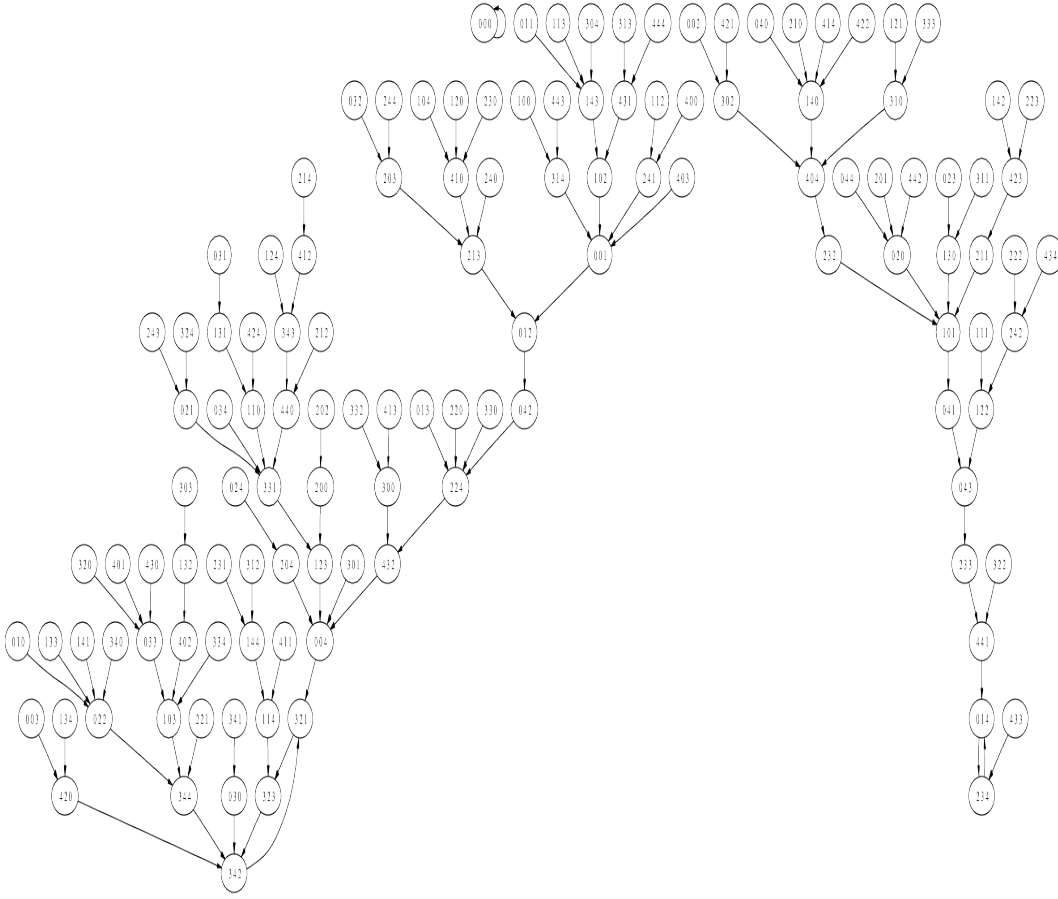


Figure 1: Space state Graph ($n = 3$ and $p = 5$) with respect to (6), (7) and (8)

The set $\mathbf{p} = \{p_1, \dots, p_n\}$ is called by N. Tchebotarev the set of *fundamental modules of s* [24]. The fundamental module of \mathbf{s} satisfies for all $\mathbf{t} \in \mathcal{E}$:

$$\forall j \in [1, n] \quad p_j(\mathbf{t}) = 0 \quad \Leftrightarrow \quad \mathbf{t} = \mathbf{s} \quad . \quad (10)$$

Which means that the variety of $\mathfrak{M}_{(\mathbf{s})}$ is reduced to the single element \mathbf{s} . It is a special case of Galois theory in which the Galois group over the field k of the polynomial $(x - s_1) \cdots (x - s_n)$ is the identity group [29]. Let J be the set of indices j for which all elements of $V = \{\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_m\}$ have the same j -th coordinate: $J = \{j \in [1, n] \mid \forall l \in [1, m] \ s_{l,j} = s_{1,j}\}$; On the coordinates indexed by j no separation is possible. Fix $S = \{1, \dots, n\} \setminus J$, the minimum subset of $\{1, \dots, n\}$ which separates V 's elements, S is called *separator set of V*. Note that the set J excludes the gene products (like proteins) whose concentration is constant in time. We keep our calculations for genes products that vary over time: these are S 's elements. In particular, for $V = \mathcal{E}$, the separator set is $S = \{1, \dots, n\}$. Let us consider this first theorem that lights us on the algebraic computation of separators:

Theorem 1. *Let be the univariate polynomial*

$$g(\mathbf{x}) = \prod_{j \in S} \prod_{l \in E} (p_j(\mathbf{x}) - l) \quad (11)$$

where S is the separator set of V and E all non-zero values taken by the points of V on the ideal's generators of \mathbf{s} -relations (except for those unnecessary to separation):

$$E = \{p_j(\mathbf{t}) \mid j \in S; p_j(\mathbf{t}) \neq 0; \mathbf{t} \in V\} \subset \{1, \dots, p-1\}. \quad (12)$$

Then the polynomial

$$r(\mathbf{x}) = \frac{g(\mathbf{x})}{g(\mathbf{s})} \quad (13)$$

is a separator of \mathbf{s} in V .

Proof. Let $g(\mathbf{s}) \neq 0$ and so $r(\mathbf{s}) = 1$. In fact, for all $j \in S$ and $l \in E$, $p_j(\mathbf{s}) - l = -l \neq 0$. The product $\prod_{l \in E} -l \neq 0$ in k because $E \subset \{1, \dots, p-1\}$ and p is a prime number. Let $\mathbf{t} \in V \setminus \{\mathbf{s}\}$, then it exists $j \in [1, n]$ where $p_j(\mathbf{t}) \neq 0$; by definition of E , $l = p_j(\mathbf{t}) \in E$; as the g 's factor $p_j(\mathbf{x}) - l \neq 0$ for $\mathbf{x} = \mathbf{t}$, we obtain the identity $r(\mathbf{t}) = g(\mathbf{t}) = 0$; \square

Proposition 1. *The complexity of calculation proposed in Theorem 1 equals $o(n(p-1))$.*

Proof. Let $\text{card}(E)$ be the cardinal of the set E and $\text{card}(S)$ the cardinal of the set S . In the worst case, computing the univariate polynomial $g(\mathbf{x}) = \prod_{j \in S} \prod_{l \in E} (p_j(\mathbf{x}) - l)$ uses $\text{card}(S) = n$ and $\text{card}(E) = p-1$. On the other hand, $p_j(x) = x_j - s_j$ and computing a separator $r(x)$ needs $2n(p-1)$ additions, $(n-1)(p-1)$ multiplications and 1 division in $\mathbb{Z}/p\mathbb{Z}$. \square

To obtain a PDS associated with a discrete path of length m , it suffices to compute $m-1$ separators. The result is not optimal (see Corollary 1) but the computational complexity remains linear in n , which is very interesting to find a first study model (remember that biological data are such that p and m are very small compared to n).

2.2 Optimal Algebraic Separators

In the previous section, and specially in Theorem 1, we separate repeatedly the same state \mathbf{s} from a state \mathbf{t} in V . To avoid this redundant operation, let us reduce the degree of g . For $\mathbf{t} \in V$ and $j \in [1, n]$, $p_{t,j}(\mathbf{x}) = x_j - t_j$.

Definition 2. *A separate set of two distinct points \mathbf{s} and \mathbf{t} of V is given by:*

$$S(\mathbf{s}, \mathbf{t}) = \{j \in S \mid s_j \neq t_j\} = \{j \in S \mid p_{s,j}(\mathbf{t}) \neq 0\}$$

where S is the separator set of V . The *initial set of separators of \mathbf{s} and \mathbf{t}* is:

$$PS(\mathbf{s}, \mathbf{t}) = \{p_{t,j}(\mathbf{x}) \mid j \in S(\mathbf{s}, \mathbf{t})\}.$$

A *minimal initial separator's set of \mathbf{s} in V* , denoted by $\text{Min}(\mathbf{s}, V)$ is composed of less than $m-1$ distinct polynomials g the same applies to each $\mathbf{t} \in V$ distinct from \mathbf{s} , the intersection of $\text{Min}(\mathbf{s}, V)$ with $PS(\mathbf{s}, \mathbf{t})$ is reduced to a single element.

For every distinct point \mathbf{s} and \mathbf{t} in V and for each $g \in PS(\mathbf{s}, \mathbf{t})$, we have $g(\mathbf{t}) = p_{t,j}(\mathbf{t}) = 0$ and $g(\mathbf{s}) = p_{t,j}(\mathbf{s}) = -p_{s,j}(\mathbf{t}) \neq 0$ because $j \in S(\mathbf{s}, \mathbf{t})$. Any product of initial separators of \mathbf{s} and \mathbf{t} is a separator between these two points.

Corollary 1. *Let $\text{Min}(\mathbf{s}, V)$ a minimal initial separator's set of \mathbf{s} in V and let*

$$\mathcal{G} = \prod_{g \in \text{Min}(\mathbf{s}, V)} g \quad (14)$$

then

$$r(\mathbf{x}) = \frac{\mathcal{G}(\mathbf{x})}{\mathcal{G}(\mathbf{s})} \quad (15)$$

is a separator of \mathbf{s} in V .

We put $r_s = r$ or $r_i = r$ when \mathbf{s} is indexed by i . To illustrate the above definitions, consider the statements $\mathbf{s}_1 = (2, 1, 2)$, $\mathbf{s}_2 = (1, 1, 0)$, $\mathbf{s}_3 = (0, 0, 1)$ and $\mathbf{s}_4 = (1, 2, 0)$. The separator set is $S = \{1, 2, 3\}$ and the respective separator sets of pair of elements of V are: $S(\mathbf{s}_1, \mathbf{s}_2) = \{1, 3\}$, $S(\mathbf{s}_2, \mathbf{s}_4) = \{2\}$, $S(\mathbf{s}_1, \mathbf{s}_3) = S(\mathbf{s}_1, \mathbf{s}_4) = S(\mathbf{s}_2, \mathbf{s}_3) = S(\mathbf{s}_3, \mathbf{s}_4) = \{1, 2, 3\}$.

We get \mathbf{s}_1 's initial sets of separators: $PS(\mathbf{s}_1, \mathbf{s}_2) = \{x_1 - 1, x_3\}$, $PS(\mathbf{s}_1, \mathbf{s}_3) = \{x_1, x_2, x_3 - 1\}$, $PS(\mathbf{s}_1, \mathbf{s}_4) = \{x_1 - 1, x_2 - 2, x_3\}$. Here are some minimal initial separator's sets $\text{Min}(\mathbf{s}_1, V)$ of \mathbf{s}_1 in V (there is a finite number): $\{x_1 - 1, x_1\}$, $\{x_1 - 1, x_2\}$, $\{x_3, x_2\}$. We compute a separator r_1 of $\mathbf{s}_1 = (2, 1, 2)$. From the minimum set $\{x_3, x_2\}$ of \mathbf{s}_1 in V , we obtain the polynomial $r_1 = x_2 x_3 / 2 = -x_2 x_3$ which equals to 1 in \mathbf{s}_1 and 0 everywhere in V . This is the same as the separator in [19].

A separator of s_4 obtained by Corollary 1 is $r_4 = (x_1 - 2)(x_2 - 1)(x_3 - 1) = x_1x_2x_3 - x_1x_2 - x_1x_3 + x_2x_3 + x_1 - x_2 - x_3 + 1$.

The computation of each separator can be done simultaneously [30]: in fact, the separators of V 's elements are independent from each other. So, the computation of a PDS can be done in parallel by applying for each separator the Theorem 1. Note that the separators obtained by Corollary 1 have degree $\leq n$, while those obtained by Theorem 1 have degree $\leq n \cdot \text{card}(E)$. For example, for $n = 3$ and $E = \{1, 2\}$, $n \cdot \text{card}(E) = 6$ and the \mathbf{s}_1 separator obtained by Theorem 1 (with linear complexity) is $r_1 = -x_1(x_1 - 1)(x_2 - 2)x_2x_3(x_3 - 1)$.

2.3 Boolean Dynamical Systems

We suppose, for simplicity, that the separator's set S of V is $\{1, \dots, n\}$. In the case $p = 2$, E is reduced to $\{1\}$ and separators are in a more compact form. In fact, let $k = \mathbb{Z}/p\mathbb{Z}$ and let two polynomials in $k[x_1, \dots, x_n]$ given by:

$$\begin{aligned} q &= \sum_{r=1}^n e_r(\mathbf{p}) \quad \text{and} \\ r &= q + 1 \end{aligned}$$

where $e_1(\mathbf{p}), \dots, e_n(\mathbf{p})$ are elementary symmetric functions in the elements of $\mathbf{p} = \{p_1, \dots, p_n\}$. To compute, $e_i(\mathbf{p})$, we use tools developed in [28]. For all $\mathbf{t} \in \mathcal{E}$

$$\begin{aligned} q(\mathbf{t}) &= 0 \quad \Leftrightarrow \quad \mathbf{t} = \mathbf{s} \\ r(\mathbf{t}) &= 0 \quad \Leftrightarrow \quad \mathbf{t} \neq \mathbf{s}. \end{aligned}$$

In particular, $q(\mathbf{t}) = 1 \quad \forall \mathbf{t} \neq \mathbf{s}$ and $r(\mathbf{s}) = 0$. To see these properties on q and r , we consider the polynomial g in univariate y with coefficients in $k[x_1, \dots, x_n]$ whose roots are the fundamental modules, being written as follows:

$$g(y)(\mathbf{x}) = \prod_{j=1}^n (y - p_j(\mathbf{x})).$$

The polynomial $g(1)(x)$ is the same as that of Theorem 1. For $t \in \mathcal{E}$, we have the following equivalences:

$$g(1)(\mathbf{t}) = 0 \Leftrightarrow \exists j \in [1, n] : p_j(\mathbf{t}) = 1 \Leftrightarrow \mathbf{t} \neq \mathbf{s}.$$

So $g(1)$ is a polynomial separating for \mathbf{s} in \mathcal{E} . Moreover, according to the following identity on the coefficients of univariate polynomials:

$$g(y) = y^n - e_1(\mathbf{p}) \cdot y^{n-1} + e_2(\mathbf{p}) \cdot y^{n-2} \dots + (-1)^n \cdot e_n(\mathbf{p}),$$

we have $r = g(1) = q + 1$ in $\mathbb{Z}/2\mathbb{Z}$.

To illustrate this result, let $m = 3$, $p = 2$, $\mathbf{s}_1 = (0, 1, 1)$, $\mathbf{s}_2 = (1, 1, 0)$ and $\mathbf{s}_3 = (0, 1, 1)$. The fundamental module sets associated to each state \mathbf{s}_i is denoted by \mathbf{p}_i ; we obtain:

$$\mathbf{p}_1 = \{x_1, x_2 + 1, x_3 + 1\} \quad \text{and} \quad \mathbf{p}_2 = \{x_1 + 1, x_2 + 1, x_3\}.$$

The elementary symmetric functions of \mathbf{p}_1 are:

$e_1(\mathbf{p}_1) = x_1 + x_2 + x_3$, $e_2(\mathbf{p}_1) = x_1x_2 + x_1x_3 + x_2x_3 + x_2 + x_3 + 1$ and $e_3(\mathbf{p}_1) = x_1x_2x_3 + x_1x_2 + x_1x_3 + x_1$; so a separator of \mathbf{s}_1 is

$$r_1(x_1, x_2, x_3) = x_1x_2x_3 + x_2x_3.$$

In parallel, we find the separator r_2 of \mathbf{s}_2 :

$$r_2(x_1, x_2, x_3) = x_1x_2x_3 + x_1x_2.$$

For $t_1 = (1, 0)$, $t_2 = (1, 1)$ and $t_3 = (0, 1)$, we obtain $f_1 = r_1$, $f_2 = r_1 + r_2$ and $f_3 = r_2$. So, $f = (f_1, f_2, f_3)$ is defined by:

$$f_1 = x_1x_2x_3 + x_2x_3 \quad , \quad f_2 = x_1x_2 + x_2x_3 \quad \text{and} \quad f_3 = x_1x_2x_3 + x_1x_2. \quad (16)$$

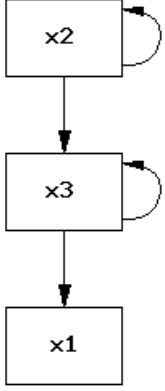


Figure 2: Dependency graph of $f = (x_3, x_2, x_2 + x_3)$ where $n = 3$, $m = 3$ and $p = 2$

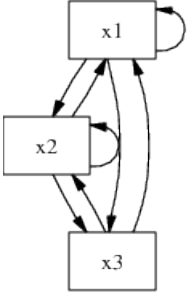


Figure 3: Dependency graph of $f = (x_1x_2x_3 + x_2x_3, x_1x_2 + x_2x_3, x_1x_2x_3 + x_1x_2)$ where $n = 3$, $m = 3$ and $p = 2$

The base of each component of the PDS f is used to draw the dependency graph (shown in Figure 2). Note that using Cocoa, we must consider the states \mathbf{s}_1 and \mathbf{s}_2 that give $r_1 = x_3$ and $r_2 = x_2 + x_3$. Hence, taking into account \mathbf{s}_3 : $f = (x_3, x_2, x_2 + x_3)$. This Boolean dynamical system allows us to obtain the dependency graph in Figure 3 (graph which includes a smaller number of arcs than the PDS (16)). The dependency graph is a directed graph with vertex set $\{x_1, \dots, x_n\}$ and edge set $\{(t, x_i) \mid t \in \text{supp}(f_i), i = 1, \dots, n\}$ where the support of f_i , denoted $\text{supp}(f_i)$, is the set of variables that appear in f_i .

2.4 Affine Separators

We detail in this section a method for determining, in some cases, affine separators. Note $CL(\mathbf{t})$ the set of all linear combinations of generators and $\mathfrak{M}_{(\mathbf{t})}$ the ideal of (\mathbf{t}) -relations. To find affine separators of \mathbf{s} in V , we must determine

$$(\cap_{\mathbf{t} \neq \mathbf{s}} CL(\mathbf{t})) \setminus CL(\mathbf{s}).$$

This set is a set of affine separators of \mathbf{s} in V . To determine if a relationship does not belong to $\mathfrak{M}_{(\mathbf{s})}$ (i.e., to $CL(\mathbf{s})$), simply divide in the fundamental modules of $\mathfrak{M}_{(\mathbf{s})}$ (or what amounts to even evaluate \mathbf{s}). Generally

$$r_j \in \cap_{i \neq j} \mathfrak{M}_{(\mathbf{s}_i)} \quad \text{and} \quad r_j \notin \mathfrak{M}_{(\mathbf{s}_j)},$$

for any separator r_j of \mathbf{s}_j in V .

Let $\mathbf{s}_1 = (2, 1, 2)$, $\mathbf{s}_2 = (4, 4, 0)$, $\mathbf{s}_3 = (3, 3, 1)$, $\mathbf{s}_4 = (1, 2, 3)$; $S = \{1, 2, 3, 4\}$. First, we add the generators 2-by-2 ($p_{i,j} + p_{i,k}$), second we subtract ($p_{i,j} - p_{i,k}$) and $2p_{i,j} + p_{i,k}$, etc. Then we pass $p_{i,1} + p_{i,2} + p_{i,3}$; for example

$$\begin{aligned}\mathfrak{M}_{(\mathbf{s}_1)} &= \langle x_1 + 3, x_2 + 4, x_3 + 3 \rangle \\ CL(\mathbf{s}_1) &= \{0, 2x_1 + 3x_2 + x_3 + 1, 3x_1 + 2x_2 + 3x_3 + 1, 3x_1 - x_2 + x_3 + 3, \dots, \\ &\quad x_2 - x_3 + 1, x_1 + x_2 + 2x_3 + 3, 3x_1 + 2x_2 + 2, \dots, \mathbf{2x_1 + 2x_2 - 1}, \dots\} \\ \mathfrak{M}_{(\mathbf{s}_2)} &= \langle x_1 + 1, x_2 + 1, x_3 \rangle \\ CL(\mathbf{s}_2) &= \{-x_2 + 4, -x_1 - x_2 - x_3 + 3, 3x_1 + 2x_2 + 3x_3, 3x_1 - x_2 + x_3 + 2, \\ &\quad -x_1 - x_3 + 4, 3x_1 + x_3 + 3, 2x_1 + x_2 + x_3 + 3, x_1 + x_2 + 3x_3 + 2, \\ &\quad 3x_1 + x_2 + 2x_3 + 4, x_1 + 3x_2 + x_3 + 4 \quad \mathbf{2x_1 + 2x_2 - 1}, 2x_1 + 2x_2 + 3x_3 + 4, \dots\}\end{aligned}$$

$$\begin{aligned}\mathfrak{M}_{(\mathbf{s}_3)} &= \langle x_1 + 2, x_2 + 2, x_3 + 4 \rangle \\ CL(\mathbf{s}_3) &= \{-x_2 + 3, 3x_1 + 3x_2 + 2, 3x_1 + 2x_2 + 3x_3 + 2, 3x_1 - x_2 + x_3 + 3, \\ &\quad \dots, x_1 - x_2 + 3x_3 + 2, x_1 + 3x_3 + 4\} \\ \mathfrak{M}_{(\mathbf{s}_4)} &= \langle x_1 + 4, x_2 + 3, x_3 + 2 \rangle \\ CL(\mathbf{s}_4) &= \{-x_1 - x_2 - x_3 + 1, 3x_1 - x_2 + x_3 + 1, -x_1 - x_3 + 4, 3x_1 + 2x_2 + 3x_3 + 4, \\ &\quad \mathbf{2x_1 + 2x_2 - 1}, x_1 + 3x_3, 3x_1 + 3x_2 + x_3 + 3, -x_2 + 2 \dots\}\end{aligned}$$

We quickly find the separator $2x_1 + 2x_2 - 1$ of \mathbf{s}_3 and we can also find a linear identification. $r_3 = 2x_1 + 2x_2 - 1$ is an element of $(\cap_{i \neq 3} CL(\mathbf{s}_i))$ that does not belong to $CL(\mathbf{s}_3)$. In this example, it is not possible to find affine separators of \mathbf{s}_1 , \mathbf{s}_2 and \mathbf{s}_4 .

This method can fully be treated dynamically and there is no question to compute full $CL(\mathbf{t})$. In fact, there is an interesting algorithm for implementation. Several strategies are possible including parallel computing. On the other hand, to avoid repeating the same relationship in the ideal of (\mathbf{t}) -relations $\mathfrak{M}_{(\mathbf{t})}$, we choose a standard format for each polynomial. For example, the polynomial is the same denominator (which we withdrew), it is divided by its content (the gcd of its coefficients) and multiplying the coefficient of the smallest variable by its sign to make it positive. The authors of [18] explain that the order of x_i can be predetermined by information concerning the influence of certain genes over others: e.g. $x_1 < x_2 < \dots < x_n$.

Thus, we obtain a first model in the form of a PDS with linear complexity in the order of $o(n \cdot (p-1) \cdot (m-1))$. Many other PDS can be determined with our approach optimally. In the other hand, getting all Polynomial dynamical systems from one PDS computed algebraically is also possible: a calculation of the generators of the ideal $I(V) = \{h \in k[\mathbf{x}] \mid h(\mathbf{s}) = 0, \forall \mathbf{s} \in V\}$. The computation of all the PDS through the determination of all separators: a separator $r(\mathbf{x})$ of \mathbf{s} in V follows from the choice of an element of all

$$\{r \notin \mathfrak{M}_{(\mathbf{s})} \text{ and } r \in \mathfrak{M}_{(\mathbf{t})} \quad \forall \mathbf{t} \in V, \mathbf{t} \neq \mathbf{s}\} \quad .$$

3 Application to System biology

Let us begin with a brief descriptions and reminders of some of the important concepts, as well as of our example data. Once that has been done; we show how this information can be carried out using PDS. DNA repair pathways maintain the integrity of the genome. There are at least 150 different proteins that catalyze DNA repair [2]. Conversely, the requirement for DNA repair and genome maintenance in response to therapy implicates DNA repair proteins. This offers novel approaches for tumour selective treatment intended to relieve or heal a disorder. To seed new therapies, researches are needed to explore the detailed consequences of an alteration in each of these repair pathways in general, and in deficient cell lines in particular. Different cell lines provide essential tools to address this need. At first, we selected genes regarding the homologous recombinations pathway [27].

The used information is compared with respect to the evolution of genes in different cell lines, according to the data at different time [31]. Furthermore, we consider the evolution of each gene independently considering the discretization of their expression [13]. We assume that given the small changes in gene expressions, Boolean discretization is sufficient for

our model. Indeed, we calculate the average expression of each gene over time and we discretize in two Boolean states with respect to this average. Then, we selected genes that have consistent behaviour compared to the first filter. After this filtering, we have kept the genes common to two cell lines, noted C_1 and C_2 . Among these genes, we chose to keep those who behave differently on each cell lines: we consider for simplicity 5 genes in Table 1, for which we compute a model for each cell line, which allows us to compare their evolution, over time.

Gene cell lines	C_1 at 0h	C_1 at 24h	C_1 at 72h	C_2 at 0h	C_2 at 24h	C_2 at 72h
g_1	1	1	0	1	0	0
g_2	1	1	0	1	0	0
g_3	0	1	1	1	1	0
g_4	1	0	0	1	0	1
g_5	1	1	0	0	1	0

Table 1: Example of 5 genes, common to 2 cell lines C_1 and C_2 in 3 steps of time.

We can now assume that the number of variables appearing in each component of the PDS will make the difference. What counts is to have a model that sticks most to the realities of data. We obtain two models: f_{C_1} describes C_1 cell lines and f_{C_2} describes C_2 cell lines. Simulation and computation were obtained by using SageMath [22, 7].

$$\begin{aligned}
f_{C_1} &= (0, 0, -x_1x_2x_3x_4x_5 + x_1x_2x_3x_4, \\
&\quad -x_1x_2x_3x_4x_5 + x_1x_2x_3x_5 + x_1x_3x_4x_5 + x_2x_3x_4x_5 - x_1x_3x_5 - x_2x_3x_5 - x_3x_4x_5 + x_3x_5, \\
&\quad -x_1x_2x_3x_4x_5 + x_1x_2x_3x_4) \\
f_{C_2} &= (-x_1x_2x_3x_4x_5 + x_1x_2x_4x_5, -x_1x_2x_3x_4x_5 + x_1x_2x_4x_5, x_1x_2x_3x_5 + x_1x_2x_4x_5, \\
&\quad 0, -x_1x_2x_3x_4x_5 + x_1x_2x_4x_5)
\end{aligned}$$

Here, x_1 is the concentration of g_1 , x_2 the concentration of g_2 , x_3 the concentration of g_3 , x_4 the concentration of g_4 and x_5 the concentration of g_5 . Then we deduce a wiring diagram and a state space graph of the given input data. Describing a gene network in terms of Polynomial dynamical system has advantages. First, it describes gene interactions in an explicitly numerical form. Second, these are casual relations between genes: a coefficient x_i in a function f_j determines the effect of gene i on gene j .

4 Perspective and Conclusion

Many biological systems are modeled with discrete models. From the research that has been carried out, it is possible to conclude that effective alternative for Gröbner basis when modeling biological systems is realizable. We use classical method based on Lagrange's interpolation. This paper details an approach allowing separators's computation: we present a method based on Galois theory's tools as the fundamental modules or elementary symmetric functions.

The findings have directly practical relevance. We propose an algorithm that performs learning in Polynomial dynamical systems. For this purpose, we introduce a main theorem for computing algebraic separators with optimal degree, it allows us to compute a finite number of other Polynomial dynamical systems. We also introduce some basic techniques that enable us to compute affine separators.

In this context, we presented an analytical method of easily readable expression and easily interpretable specific data. Thus, we obtain a first model in the form of a Polynomial dynamical systems with linear complexity. Many other Polynomial dynamical systems can be determined with our approach optimally. Besides, getting all polynomial dynamical systems from one computed algebraically is also possible: a calculation of the generators of the ideal $I(V)$. The calculation of all the PDS through the determination of all separators. The gain is that very quickly we propose models to bio-informatics and molecular biologists in which they can advance and refine their queries.

Clearly, further research will be required on experimental data. Continuing research on this field appears fully justified because of the simplicity of this approach. Finally, there is only one parameter p introduced into the model, unlike the

continuous model using differential equations which must be added a number of constraints and parameters for successful modelling.

References

- [1] Abbott, J., Bigatti, A.M., Cocalib: a C++ library for doing Computations in Commutative Algebra. Available at <http://cocoa.dima.unige.it/cocalib>.
- [2] Alberts, B., Redefining cancer research. *Science*, volume 325, pages 1319–1319, 2009.
- [3] Benkahla, A., Guizani-Tabbane, L., Abdeljaoued-Tej, I., BenMiled, S., Dellagi, K.. Systems biology and infectious diseases. *Handbook of Research on Systems Biology Applications in Medicine*, volume 1, pages 377–402, 2008.
- [4] Bortolussi, L., Policriti, A., *Hybrid Systems and Biology*. Springer Berlin Heidelberg, Berlin, Heidelberg, pages 424–448 2008..
- [5] Cinquemani, E., Donzé, A.. Hybrid systems biology, in: *Proceedings of International Workshop on Hybrid Systems Biology*, Springer, Grenoble, France, 2016.
- [6] De Hoon, M.J.L., Imoto, S., Kobayashi, K., Ogasawara, N., Miyano, S., Inferring gene regulatory networks from time-ordered gene expression data of bacillus subtilis using differential equations. *Pac Symp Biocomput*, pages 17–28, 2003.
- [7] Developers, T.S., SageMath, the Sage Mathematics Software System. <http://www.sagemath.org>, 2017.
- [8] Dimitrova, E.S., Vera-Licona, P., McGee, J., Laubenbacher, R.C., Discretization of time series data. *Journal of Computational Biology*, volume 17, pages 853–868, 2010.
- [9] Friedman, N., Linial, M., Nachman, I., Using bayesian networks to analyze expression data, *Journal of Computational Biology*, volume 7, pages 601–620, 2000.
- [10] Grayson, D.R., Stillman, M.E., Macaulay2, a software system for research in algebraic geometry. Available at <http://www.math.uiuc.edu/Macaulay2/>, 2000.
- [11] Hinkelmann, F., Brandon, M., Guang, B., McNeill, R., Blekherman, G., Veliz-Cuba, A., Laubenbacher, R., Adam: Analysis of discrete models of biological systems using computer algebra, *BMC Bioinformatics*, volume 12, pages 295, 2011.
- [12] Imoto, S., Goto, T., Miyano, S., Estimation of genetic networks and functional structures between genes by using bayesian networks and nonparametric regression. *Pac Symp Biocomput*, pages 175–186, 2002.
- [13] Irizarry, R.A., Hobbs, B., Collin, F., BeazerBarclay, Y.D., Antonellis, K.J., Scherf, U., Speed, T.P., Exploration, normalization, and summaries of high density oligonucleotide array probe level data. *Biostatistics*, volume 4, pages 249–264, 10.1093/biostatistics/4.2.249, 2003.
- [14] Jarrah, A.S., Laubenbacher, R., Stigler, B., Stillman, M., Reverse-engineering of polynomial dynamical systems. *Advances in Applied Mathematics*, volume 39, Issue 4, pages 477–489, 2007.
- [15] Khatri, P., Dràghici, S., Ontological analysis of gene expression data: current tools, limitations, and open problems. *Bioinformatics* volume 21, pages 3587–3595, 2005.
- [16] Lagrange, J., *Réflexions sur la résolution algébrique des équations*, 1770.
- [17] Laubenbacher, R., A computer algebra approach to biological systems, in: *booktitleProceedings of the 2003 International Symposium on Symbolic and Algebraic Computation*, ACM, New York, NY, USA, 2003.
- [18] Laubenbacher, R., *Modeling and Simulation in Applied Mathematics*, volume 64 of *seriesModeling and Simulation of Biological Networks: American Mathematical Society, Short Course*. AMS Bookstore, 2007.

- [19] Laubenbacher, R., Stigler, B., A computational algebra approach to the reverse engineering of gene regulatory networks. *Journal of Theoretical Biology*, volume 229, pages 523 – 537, 2004.
- [20] Paroni, A., Graudenzi, A., Caravagna, G., Damiani, C., Mauri, G., Antoniotti, M., Cabernet: a cytoscape app for augmented boolean models of gene regulatory networks. *BMC Bioinformatics*, volume 17, pages 64, 2016.
- [21] Schilstra, M., Bolouri, H., *Models of Genetic Regulatory Networks*. Springer Berlin Heidelberg, Berlin, Heidelberg, pages 149–159, 2004.
- [22] Stein, W., Joyner, D., SAGE: System for algebra and geometry experimentation. *ACM SIGSAM Bulletin*, volume 39, pages 61–64, 2005.
- [23] Stigler, B., *Polynomial Dynamical Systems in Systems Biology*, volume 64 of *series Modeling and Simulation of Biological Networks: American Mathematical Society, Short Course*, AMS Bookstore, 2006.
- [24] Tchebotarev, N., *Gründzüge des Galois'shen Theorie*. P. Noordhoff, 1950.
- [25] Thomas, R., *Kinetic logic : a boolean approach to the analysis of complex regulatory systems*, volume 29, *Lecture Notes in Biomathematics*, 1979.
- [26] Thomas, R., D'Ari, R., *Biological Feedback*, CRC Press, Inc., 1990.
- [27] Tomasz, S., Oncogenic tyrosine kinases and the dna-damage response, *Nature Reviews Cancer*, volume 2, pages 351– 360, 2002.
- [28] Valibouze, A., Symbolic computation with symmetric polynomials, an extension to Macsyma, in: *Computers and Mathematics*, Springer-Verlag, New York Berlin, pages 308–320, 1989.
- [29] Valibouze, A., Sur les relations entre les racines d'un polynôme. *Acta Arithmetica*, volume 131.1, pages 1–27, 2008.
- [30] Valibouze, A., Abdeljaoued, I., Kahla, A.B., Galoisian separators for biological systems, in: *Mathematics Algorithms Proofs - Formalization of Mathematics*, Monastir, Tunisia, 2009.
- [31] Valsecchi, M., Daz-Cantn, E., Vega, M., Littman, S., Recent treatment advances and novel therapies in pancreas cancer: A review, *Journal of Gastrointestinal Cancer*, volume 45, pages 190–201, 2014.
- [32] Vastani, H., Jarrah, A.S., Laubenbacher, R.. package, *Discrete Visualizer of Dynamics*, 2005.