Biological Regulatory Network reconstruction: a mathematical programming approach

Fabien Tarissan[1], Leo Liberti[1], Camilo La Rota[2]
[1]École Polytechnique, Palaiseau, France
[2]Complex Systems Institute, Lyon, France

May 30, 2008

Abstract

We propose a method employing mathematical programming and global optimization techniques for solving inverse problems arising in biological regulatory network (BRN) reconstruction. This problem consists in estimating unknown parameters of a model that describe the structure and dynamics of a biological system from a set of experimental observations and can be naturally cast as an optimization problem: choose the parameter values minimizing a given distance between the observed and estimated values of some observable variables. This minimization is subject to constraints derived from the models.

Two significant examples are presented on how to handle different kinds of dynamics: pattern formation occurring from diffusable and non-diffusable gene products in the *drosophila melanogaster* morphogenesis and reconstruction of the gene regulatory network of *arabidospsis thaliana* based on the identification of stable sub-networks during morphogenesis.

1 Introduction

A typical problem in biological systems consists in the simulation of a biological process, where from certain initial or boundary conditions one proceeds to solve a system of equations describing the biological process and obtains trajectories describing the system evolution in time and/or space. This is what is known as a forward problem. An example of a forward problem is the computation of the time evolution of a Gene Regulatory Network (GRN) configuration (values associated to the genes and representing gene products concentrations) given an initial condition and some local interaction rules. The GRN may be described in a detailed-quantitative manner by a system of Nonlinear Differential-Algebraic Equations (NDAE), where the values associated to the genes are real and directly represent the gene products concentrations, or in a coarse-qualitative manner by a system of nonlinear Binary or Multivalued Difference Equations (NBdE,NMdE) or by a system of Boolean or Multivalued Logical Equations (LE,MLE), where the set of values represent discrete levels of gene products concentrations such as abscence (0) or saturation (1) in the case of boolean representations.

That, however, presupposes that all "known values" (coefficients of the DAE system, initial concentration of gene products) are known apriori, which is rarely the case. Thus, pre-simulation methods addressing the problem of estimating the unknown parameters from a set of experimental observations become of paramount importance. Such problems are often termed *inverse problems* as in some sense one needs to solve a problem having the same form, but where the roles of known and unknown values are interchanged and because only the model equations for the forward problem are known.

The usual approach developped in the context of molecular biology in order to adress the problem is often based on simulation techniques. It requires first to design specific heuristics that will infer values to the missing parameters and then, in a second step, to simulate the evolution of the system in order to compare the values generated by the simulation with the observed ones. The gap between generated and observed values then defines the quality of the estimation. As one might guess, the efficiency of this approach is directly related to the heuristic used which induces generally to spend time and effort on tuning the algorithms for each study. This remark has a great impact as soon as we intend to cover a wide range of organisms.

The approach we propose in this paper relies on the remark that the problem of estimating unknown parameters from a set of experimental observations can be naturally cast as an optimization problem: choose the parameter values minimizing the sum of a given norm of the differences between the observed and estimated values ofsome observables. Although designing specific algorithms for a given organism could provide faster results, the approach we propose has the advantage of generality: potentially any parameter estimation problem can be modeled by mathematical programming and solved with corresponding general-purpose algorithms. More precisely, we use the modeling language AMPL [2] in order to describe the biological system and then apply different reformulations (both exact and approximative) in order to ease the computation process carried out by dedicated Non Linear Programs (NLP) and Mixed Integer NLP (MINLP) numerical solvers [5]. The relevance of a general-purpose approach that sacrifices some computational efficiency is to provide general modeling environments and software packages with solution methods that can easily adapt to all (or at least most) models input by the user. This need is in our case explicitly established by the EU Morphex project and its associated modeling language.

The rest of the paper is organized as follow. In Section 2 we present the basic concepts underlying mathematical programming. In Section 3 we show how to use such a technique to reconstruct the GRN of *Drosophila meloganaster* in the early development stage relying on NDAE. In Section 4 we test the same approach but on the *Arabidopsis thaliana* modeled using BNdE. Finally, in Section 5 we discuss the main advantages and drawbacks of our approach.

2 Mathematical programming and optimization

A mathematical programming problem is formulated as follows:

$$\left.\begin{array}{rll} \min_x & f(x) & \\ \text{subject to} & g(x) & \leq \quad 0, \end{array}\right\} \tag{1}$$

where $x \in \mathbb{R}^n$ are the *decision variables* and $f : \mathbb{R}^n \to \mathbb{R}$ is the *objective function* to be minimized subject to a set of constraints $g : \mathbb{R}^n \to \mathbb{R}^m$ which may also include variable ranges or integrality constraints on the variables. Global Optimization is concerned with the solution of problems (1) where f, g are nonconvex nonlinear forms. A problem where f, g are nonlinear is known as a Nonlinear Programming problem (NLP); if some integrality constraints are present on the variable bounds the problem is known as a Mixed-Integer NLP (MINLP). A mathematical programming problem which has some integer variables but whose objective function and constraints are linear forms is called a Mixed-Integer Linear Programming (MILP) problem. There exists general-purpose solution algorithms, both exact and heuristic, for all problem forms in NLP, MINLP, MILP. Currently, MILP solution methods are the most advanced, and the *de facto* standard solver is the ILOG CPLEX [3] solver. MINLPs and nonconvex NLPs can be solved by many different global optimization methods such as BARON [8]. Most frequently, NLPs and MINLPs undergo a reformulation stage before being solved [5, 6].

3 Reproducing a continuous regulation of genes

The motivation of the present section is to test the relevancy of this framework on a first biological case study. We focused here on the Gap-Gene formation in the early development of *Drosophila melanogaster*. At this stage (between cycle 13 and 14A), the embryo contains static nuclei which can diffuse their gene products within their neighborhood. The model [4, 7] uses a discrete representation of the space (linear topology) but a continuous regulation of the genes in time. The differential equation that describe this continuous regulation is as follows:

$$\frac{dg_{ia}(t)}{dt} = R_a \Phi(u_{ia}(t)) - \lambda_a g_{ia}(t) + D_a(g_{i+1,a}(t) - 2g_{ia}(t) + g_{i-1,a}(t)), \quad (2)$$

where $g_{ia}(t)$ is the concentration of gene a (belonging to the set of genes N^γ) in nucleus i (belonging to the set of nuclei N^ι) at time t, R_a is the production rate for gene a, $\Phi(y)$ is the sigmoid regulation function

$$\Phi(y) = \frac{1}{2}\left(\frac{y}{\sqrt{y^2+1}} + 1\right) \quad (3)$$

defined for all y, $u_{ia}(t)$ is a function defined as follows:

$$u_{ia}(t) = \sum_{b \in N^\gamma} W_{ba} g_{ib}(t) + m_a g_i^{\mathsf{bcd}} + h_a \quad (4)$$

where W_{ba} is the weight on the arc (b, a) in the digraph representation $G = (N^\gamma, A)$ of the gene regulatory network, m_a is the regulatory influence of the maternal gene bcd, h_a is the activation threshold for Φ; λ_a is the decay rate and D_a is the diffusion coefficient for gene a.

The solution method proposed in [7] is a Global Optimization (GO) evolutionary algorithm applied to the following unconstrained optimization problem:

$$\min \sum_{i \in N^\iota} \sum_t (g_{ia}(t) - g_{ia}^{\mathsf{data}}(t))^2 + \Pi_R + \Pi_\lambda + \Pi_D + \Pi_u, \quad (5)$$

where $g_{ia}^{\mathsf{data}}(t)$ is a set of experimentally observed values for the concentration level of gene a in nucleus i at time t; Π_R is a penalty function that helps restricting R_a to lie in a pre-determined range $[R_a^L, R_a^U]$ (Π_λ and Π_D are similar to Π_R); and finally, Π_u is defined as

$$\Pi_u = e^{\Theta} - 1 \tag{6}$$

(where $\Theta = \Lambda(\sum_{(b,a)\in A}(W_{ba}v_b^{\mathsf{max}})^2 + (m_a v_{\mathsf{bcd}}^{\mathsf{max}})^2 + h_a^2))$ if $\Theta > 0$ and 0 otherwise, and v_b^{max} and $v_{\mathsf{bcd}}^{\mathsf{max}}$ are the maximum values for gene $b \in N^\gamma$ and bcd respectively.

We remark straight away that since Θ is a sum of squares, it is necessarily nonnegative, which implies that it suffices to describe Π_u simply as $e^{\Theta} - 1$ without any need for conditional definitions. Consider now the functions $\alpha_1(y) = y + 1$ and $\alpha_2(y) = \log(y)$ for all $y > 0$. Since both are monotonically increasing, the minimization of Π_u determines the same set of optimal solutions as the minimization of $\alpha_2(\alpha_1(\Pi_u))$. Hence (5) is equivalent to:

$$\min \sum_{i\in N^\iota} \sum_t (g_{ia}(t) - g_{ia}^{\mathsf{data}}(t))^2 + \Pi_R + \Pi_\lambda + \Pi_D + \Theta. \tag{7}$$

Thus, the model used in this example contains two different kinds of elements: the description of the equations (rules 2, 3, 4) that control the evolution of the parameters and the description of the criterion that allows to establish if the estimated values are close the real ones (rule 7). This distinction will be naturally reflected in the model described in AMPL. The rule 5 is for instance clearly the objective function we are looking for whereas all the equations that describes the dynamic of the model will be stated as constraints. In order to ease the reading, we don't present the complete model in AMPL in this section but in the appendix 6.1 . The only main difference between the two versions that worth being noticed is related to the way we model the terms Π_R, Π_λ and Π_D. Since those functions corresponds to specific boundaries for the parameters R, λ and D, it is natural to use constraints as follow:

$$\forall a \in N^\gamma \begin{cases} R^L \leq R_a \leq R^U \\ \lambda^L \leq \lambda_a \leq \lambda^U \\ D^L \leq D_a \leq D^U \end{cases}$$

It induces in particular that the objective function will have a form slightly different from the equation (7):

$$\min \sum_{\substack{a\in N^\gamma \\ i\in N^\iota \\ t\in T^{\mathsf{data}}}} (\mathsf{g}_i^a(t) - \mathsf{g}_{\mathsf{data}}{}_i^a(t))^2 + \sum_{\substack{a\in N^\gamma \\ b\in N^\gamma}} (\mathsf{W}_b^a v_{\mathsf{max}}^b)^2 + \sum_{a\in N^\gamma} ((m_a v_{\mathsf{max}}^{\mathsf{bcd}})^2 + h_a^2). \tag{8}$$

4 Using fixed points for GRN reconstruction

In the previous section, we showed an example where the objective function derived from the equation (7) used to establish the quality of the estimated values can easly be translated into AMPL (8). This is of course directly related to the way the model is presented. In order to validate our approach, it is necessary to experience the same process on another kind of criterion. This is what this section is devoted to. More precisely, we solve here an inverse

problem consisting in the determination of the parameters of a GRN in such a way that the reconstructed network show observed states at given development stages. Those transitory states of the GRN are expressed as stable states of sub-networks such that the activities of the nodes involved in a sub-network do not change anymore as soon as the stable state is reached and the sub-network is isolated from the rest of the network. The case of study chosen here is the *Arabidopsis thaliana* floral meristem GRN and the expression patterns observed at the very first development stages. This network is still poorly understood, even if data concerning genes and morphogenes involved, single regulatory interactions and small circuits exist in the literature [1].

Given a directed graph $G = (V, A)$, a discrete set of time instants T, a set of development stages S (both of which we suppose to be an initial contiguous proper subset of $\mathbb{N}$) and the following functions:

- a function $\alpha : A \to \{+1, -1\}$ called the *arc sign function*;
- a function $\omega : A \to \mathbb{R}_+$ called the *arc weight function*;
- a function $x : V \times T \to \{0, 1\}$ called the *gene activation function*;
- a function $\iota : V \to \{0, 1\}$ called the *initial configuration*;
- a function $\theta : V \to \mathbb{R}_+$ called the *threshold function*,

A *gene regulatory network* (GRN) is a 7-tuple $(G, T, \alpha, \omega, x, \iota, \theta)$ such that:

$$\forall v \in V \quad x(v, 1) = \iota(v) \tag{9}$$

$$\forall v \in V, t \in T \setminus \{1\} \quad x(v, t) = \begin{cases} 1 & \text{if } \sum_{u \in \delta^-(v)} \alpha(u, v)\omega(u, v)x(u, t-1) \geq \theta(v) \\ 0 & \text{otherwise,} \end{cases} \tag{10}$$

where $\delta^-(v) = \{u \in V \mid (u, v) \in A\}$ for all $v \in V$. Eqns. (9)-(10) together are called the *evolution rules* of the GRN. For any particular $t \in T$, $x(\cdot, t) : V \to \{0, 1\}$ is called a *configuration*. Since the evolution rules relate a configuration at time t with a configuration at time $t - 1$, if $x(\cdot, t) = x(\cdot, t - 1)$ then $x(\cdot, t') = x(\cdot, t)$ for all $t' > t$: such configurations are called *fixed points* of the GRN.

In this section we address the following problem:

> Stable Subnetworks Reconstruction of a GRNs (SSRGRN). Given a digraph $G = (V, A)$, a time instant set T, a set S of development stages, a set R_s of observed cellular types at each stage, an arc sign function α, a set of initial configurations for each stage and for each observed cellular type $I_{r_s} : V \to \{0, 1\}$, a set $U_s \subseteq V$ determining the nodes of the (induced) subnetwork G_s at each stage to be reconstructed ($U_s \subseteq U_{s+1} \subseteq V$) and the observed configurations $O_{r_s} : U_s \to \{0, 1\}$, find an arc weight function ω and a threshold function θ with the property that for all $s \in S$ and all $\iota \in I_{r_s}$ there exists a gene activation function x such that $(G_s, T, \alpha, \omega, x, \iota, \theta)$ is a GRN subnetwork whose fixed points are at a minimum distance to the observed data O_{r_s}.

In other words, we attempt to estimate the arc weights and threshold functions of a GRN from the knowledge of the digraph topology G, the induced subnetwork topology U and the arc sign function α in such a way that (a) the GRN evolution rules are consistent with respect to a certain set of initial configurations and (b) the fixed points in the subnetwork induced by the estimated values are as close as possible to the given ones.

Our primary concern in solving the SSRGRN is thus modellistic rather than algorithmic. One of the foremost difficulties is that of employing a static modelling paradigm — such as mathematical programming — in order to describe a problem whose very definition depends on time. Another important difficulty resides in expressing the necessary and sufficient conditions for a configuration to be a fixed point as constraints suitable for use in a formulation like (1). In appendix 6.2, we present a simplified version in which only one targeted configuration S is used.

5 Conclusion

As seen in this paper, the approach we propose present the advantage of presenting a unified framework able to handle two different kinds of organisms. The chosen organisms are such that both the dynamics induced by the networks (continuous in the case of drosophila and discrete in arabidopsis) and the criterion used to drive the way the values are infered tends to prove that this approach covers any parameter estimation problem as soon as the evolution rules can be described as constraints. This provides therefore an attractive environment for solving inverse problems in general.

However, the arabidopsis case showed that the definition of the objective function must be delt carefully in order to render properly the dynamics that one wants to capture. It is then compelling to try to automatize the way we define the objective function given a set of crieria (mimicking continuous regulation, identifying sub-networks which are stable, etc...). We are already engaged in this direction.

Besides, the description in AMPL of the model might also rise problems for the solvers (the definition of Φ in (3 for instance) and it is then tempting to automatize the reformulation (both exact and approximative) of the models in order to ease the solving part. This is also a promising extension we intend to perform in order to improve significantly the efficiency of this approach. In particular, we intend to rely on [6] to implement the reformulations.

Acknowledgments

We are deeply grateful to Françoise Monéger and Jan Traas from the RDP-ENS-Lyon institute for having provided the data on arabidopsis on which we tested our approach.

References

[1] V. Balanzá, M. Navarrete, M. Trigueros, and C. Ferrándiz. Patterning the female side of arabidopsis: the importance of hormones. *J Exp Bot*, 57:3457–3469, 2006.

[2] R. Fourer and D. Gay. *The AMPL Book*. Duxbury Press, Pacific Grove, 2002.

[3] ILOG. *ILOG CPLEX 8.0 User's Manual*. ILOG S.A., Gentilly, France, 2002.

[4] J. Jaeger, M. Blagov, D. Kosman, K. N. Kozlov, E. Myasnikova, S. Surkova, C. E. Vanario-Alonso, M. Samsonova, D. H. Sharp, and J. Reinitz. Dynamical analysis of regulatory interactions in the gap gene system of drosophila melanogaster. *Genetics*, 167(4):1721–1737, 2004.

[5] L. Liberti. Writing global optimization software. In L. Liberti and N. Maculan, editors, *Global Optimization: from Theory to Implementation*, pages 211–262. Springer, Berlin, 2006.

[6] L. Liberti. Reformulation techniques in mathematical programming, November 2007. Thèse d'Habilitation à Diriger des Recherches.

[7] Y.F. Nanfack, J.A. Kaandorp, and J. Blom. Efficient parameter estimation for models of pattern formation in early embryogenesis of *Drosophila* melanogaster. *Bioinformatics*, to appear.

[8] N.V. Sahinidis and M. Tawarmalani. *BARON 7.2.5: Global Optimization of Mixed-Integer Nonlinear Programs,* User's Manual, 2005.

6 Appendix

6.1 Drosophila model in AMPL

We present in this appendix how the model presented in Section 3 is written in AMPL. One may see in particular that the formulation is very close to the biological model as it is proposed in [4, 7].

- *Sets*:
  1. set $N^\iota \subset \mathbb{N}$ of nuclei;
  2. set $N^\gamma \subset \mathbb{N}$ of genes;
  3. set $T \subset \mathbb{N}$ of time instants.
  4. set $T^{\mathsf{data}} \subseteq T$ of time instants which we have observed data for.
- *Parameters*:
  1. $\Delta t \in \mathbb{R}$: interval length for time discretization;
  2. for $i \in N^\iota$, $a \in N^\gamma$ let $\mathsf{g}_{\mathsf{data}}{}_i^a(t) \in \mathbb{R}^{|T^{\mathsf{data}}|}$ be the observed data: for $t \in T^{\mathsf{data}}$, $\mathsf{g}_{\mathsf{data}}{}_i^a(t)$ is the observed concentration level of gene a in nucleus i at time t;

3. for $a \in N^\gamma$, v^a_{max} is the maximum value for gene a
4. $v^{\mathtt{bcd}}_{\mathsf{max}}$ is the maximum value for gene `bcd`;
5. for $i \in N^\iota$, $t \in T^{\mathsf{data}}$, $\mathsf{g}_{\mathsf{data}}{}^{\mathtt{bcd}}_i(t) \in \mathbb{R}$ is the observed data with respect to gene `bcd`;
6. λ^L, λ^U: lower and upper bounds for λ;
7. R^L, R^U: lower and upper bounds for R;
8. D^L, D^U: lower and upper bounds for D;
9. for all $a \in N^\gamma$, $i \in N^\iota$, $\mathsf{init}^a_i \in \mathbb{R}$ is the concentration level of gene a at the first time instant (boundary conditions for time=0).

- *Variables*: For all gene $a \in N^\gamma$, we define:

1. for all $b \in N^\gamma$ such that $(b, a) \in A$, W^a_b is the weight on (b, a);
2. m_a is the influence of gene `bcd` on gene a;
3. h_a is the threshold parameter for the sigmoid regulation function;
4. λ_a is the decay coefficient for gene a;
5. R_a is the production rate for gene a;
6. D_a is the diffusion coefficient for gene a;
7. for all $i \in N^\iota$, $t \in T$ $\mathsf{g}^i_a(t) \in \mathbb{R}$ represents the estimated data: $\mathsf{g}^a_i(t)$ is the concentration level of gene a in nucleus i at time t;
8. for all $i \in N^\iota$, $t \in T$, $\mathsf{u}^a_i(t) \in \mathbb{R}$ is an auxiliary function.

- *Objective function*:

$$\min \sum_{\substack{a \in N^\gamma \\ i \in N^\iota \\ t \in T^{\mathsf{data}}}} (\mathsf{g}^a_i(t) - \mathsf{g}_{\mathsf{data}}{}^a_i(t))^2 + \sum_{\substack{a \in N^\gamma \\ b \in N^\gamma}} (\mathsf{W}^a_b v^b_{\mathsf{max}})^2 + \sum_{a \in N^\gamma} ((m_a v^{\mathtt{bcd}}_{\mathsf{max}})^2 + h^2_a). \tag{11}$$

- *Constraints*:

1. the discretized differential equation (2): $\forall i \in N^\iota \setminus \{0, |N^\iota| - 1\}, t \in T \setminus \{0\}$

$$\mathsf{g}^a_i(t) - \mathsf{g}^a_i(t-1) = \Delta t \left(\frac{R_a}{2} \left(\frac{\mathsf{u}^a_i(t)}{\sqrt{\mathsf{u}^a_i(t)^2 + 1}} + 1 \right) - \lambda_a \mathsf{g}^a_i(t) + D_a(\mathsf{g}^a_{i+1}(t) - 2\mathsf{g}^a_i(t) + \mathsf{g}^a_{i-1}(t)) \right); \tag{12}$$

2. the definition of u_i:

$$\forall i \in N^\iota, t \in T \quad \mathsf{u}^a_i(t) = \sum_{b \in N^\gamma} \mathsf{W}^a_b \mathsf{g}^b_i(t) + m_a \mathsf{g}^{\mathtt{bcd}}_i(t) + h_a; \tag{13}$$

3. the initial conditions and ranges:

$$\forall a \in N^\gamma, i \in N^\iota \qquad \mathsf{g}^a_i(0) = \mathsf{init}^a_i;$$
$$\forall a \in N^\gamma \begin{cases} R^L \leq R_a \leq R^U \\ \lambda^L \leq \lambda_a \leq \lambda^U \\ D^L \leq D_a \leq D^U \end{cases}$$

6.2 Arabidopsis model in AMPL

- *Sets*:
  1. set V of genes in the network;
  2. set A of arcs in the network;
  3. set T of time instants.
  4. set $U \in V$ of genes involved in the stable sub-network.
- *Parameters*:
  1. $C_0 : V \to \{0, 1\}$ is the initial configuration of the network (vector of boolean values assigned to the genes).
  2. $\alpha : A \to \{+1, -1\}$ is the sign of the arc weights;
  3. θ^L, θ^U are the bounds on the threshold values;
  4. $S : U \to \{0, 1\}$ is the targeted configuration of the sub-network composed by U.
- *Variables*:
  1. for all $i \in V$, $t \in T$, $x_i^t \in \{0, 1\}$ is the activation status of gene i at time t;
  2. $s : T \to \{0, 1\}$ is a decision variable indicating that the sub-network is stable during at least two successive time steps.
  3. $y : T \to \{0, 1\}$ is a decision variable that indicates the first time the sub-network reaches a stable state.
  4. $\theta : V \to \mathbb{Z}$ is the threshold function;
  5. $w : A \to \mathbb{N}$ is the arc weight function.
- *Objective function*:

$$\min \sum_{t \in T \setminus \{1\}} \big((y^{t-1} - y^t) \sum_{u \in U} |x_u^t - S_u| \big).$$

- *Constraints*:
  1. evolution rule:

$$\forall t \in T \setminus \{1\}, v \in V$$
$$\theta_v x_v^t - |V|(1 - x_v^t) \leq \sum_{u \in \delta^-(v)} \alpha_{uv} w_{uv} x_u^{t-1} \leq (\theta_v - 1)(1 - x_v^t) + |V| x_v^t \quad (14)$$

  2. fixed point conditions:

$$\forall t \in T \setminus \{1\} \quad \sum_{u \in U} |x_u^t - x_u^{t-1}| \leq |U| s^t \quad (15)$$
$$\forall t \in T \setminus \{1\} \quad s^t \leq \sum_{u \in U} |x_u^t - x_u^{t-1}| \quad (16)$$
$$\forall t \in T \setminus \{1, |T|\} \quad y^t = s^t y^{t-1} \quad (17)$$
$$\forall t \in T \setminus \{1, |T|\} \quad \sum_{r > t} y^r \leq (|T| - t) y^t; \quad (18)$$

3. boundary conditions:

$$\forall v \in V \quad x_v^0 = C_0(v).$$