Automated Inference of Gene Regulatory Networks Using Explicit Regulatory Modules

Clémence Réda[a,*], Bartek Wilczyński[b]

[a] *École Normale Supérieure Paris-Saclay,* 61 *avenue du Président Wilson, Cachan, France*
[b] *Faculty of Mathematics, Informatics and Mechanics, University of Warsaw, ulica Stefana Banacha* 2, 02 − 097 *Warsaw, Poland*

Abstract

Gene regulatory networks are a popular tool for modelling important biological phenomena, such as cell differentiation or oncogenesis. Efficient identification of the causal connections between genes, their products and regulating transcription factors, is key to understanding how defects in their function may trigger diseases. Modelling approaches should keep up with the ever more detailed descriptions of the biological phenomena at play, as provided by new experimental findings and technical improvements. In recent years, we have seen great improvements in mapping of specific binding sites of many transcription factors to distinct regulatory regions. Recent gene regulatory network models use binding measurements; but usually only to define gene-to-gene interactions, ignoring regulatory module structure. Moreover, current huge amount of transcriptomic data, and exploration of all possible cis-regulatory arrangements which can lead to the same transcriptomic response, makes manual model building both tedious and time-consuming.

In our paper, we propose a method to specify possible regulatory connections in a given Boolean network, based on transcription factor binding evidence. This is implemented by an algorithm which expands a regular Boolean network model into a "Cis Regulatory" Boolean network model. This expanded model explicitly defines regulatory regions as additional nodes in the network, and

*Corresponding author
Email address: creda@ens-paris-saclay.fr (Bartek Wilczyński)

adds new, valuable biological insights to the system dynamics. The expanded model can automatically be compared with expression data. And, for each node, a regulatory function, consistent with the experimental data, can be found. The resulting models are usually more constrained (by biologically-motivated metadata), and can then be inspected in *in silico* simulations.

The fully automated method for model identification has been implemented in Python, and the expansion algorithm in R. The method resorts to the Z3 Satisfiability Modulo Theories (SMT) solver, and is similar to the RE:IN application (Yordanov et al., 2016).

It is available on `https://github.com/regulomics/expansion-network`.

Keywords: transcriptional regulation, transcription factor binding, Boolean network, model synthesis

1. Introduction and Problem Statement

Gene expression regulation is a key mechanism, used by multicellular eukaryotes, to differentiate cells into specific cell-types and react to changes in the environment: e.g. the myeloid differentiation in the mouse [23]. In eukaryotes, regulation occuring during the transcriptional phase of gene expression seemingly plays the most prominent part in this almost universal phenonemon. Typically, several transcriptional factors (TFs) regulate a gene, in order to ensure its activation or its repression at key times, by binding to specific DNA regions that are usually called *cis-regulatory modules* (CRMs). Usually, there are multiple such elements per target gene; however, due to incomplete information, they are usually not modelled explicitly in Boolean gene regulatory networks, as defined by Sugita, Kauffman, Thomas and others since the late 1960's [34, 20, 35]. The whole set of such interactions would constitute a Cis-Regulatory Network (CRN) – an expanded model that has been discussed in a number of papers [8, 13, 40]. TF bindings to CRMs can be identified by ChIP-seq (chromatin immunoprecipitation sequencing) experiments. In most cases, binding patterns are considered enough to determine the function of associated

TFs, as shown by Zinzen et al. [44]. There were efforts in the field to explicitly model cis-regulatory interactions as early as in 2002 [8]. However, then there was not enough data on TF-DNA binding events. In most of the typical Boolean models, the regulatory functions are identified at the gene level [25, 30]. Nevertheless, as binding data from ChipSeq and other assays are becoming available for increasing number of TFs and conditions, they are implicitly integrated into models, such as the Collombet lymphoid/myeloid differentiation model [10]. From these studies, it becomes apparent that the regulatory functions can actually be decomposed into "physical", almost self-contained, functional units, according to the TF bindings associated with their regulatory modules. It is not trivial to automate such decomposition; but having such a procedure would maybe allow to better grasp the modular [9], hierarchically organized [24] structure of CRNs. This would potentially allow us to address the issue of modelling seemingly redundant parts [27], and to understand the underlying dynamics of the regulatory interactions with more accuracy.

Gene regulatory networks (GRNs) comprise a collection of gene nodes and some signed, directed, pairwise regulatory interactions. Sometimes, a set of fixed possible "consistent" regulatory functions is attributed to each node [43]. Such a representation (called sometimes *interaction graphs*) captures the regulatory network topology; while *state graphs*, that can be computed from the topology and regulatory functions, describe the dynamics of the considered system and its stable states.

Generally speaking, GRNs rely on four main assumptions as described by Crombach [11]. Firstly, a fixed set of TFs can possibly have an influence on the expression level of a given gene. The expression levels of these TFs combine into some discrete or continuous value (input function) [4]. Secondly, this value is then thresholded by the response function (that is, a function which will output a decision: "expressed" or "non expressed" given some input) of the target gene (generally a sigmoidal or step function): if this value is over the given threshold, then the target gene expression level is considered high enough, and thus the gene is considered active (meaning that a functional amount of its products will

be present in the cell); otherwise, the gene is considered inactive, with a low expression level. Thirdly, such a network acknowledges the presence of feedback loops, that are characterized by cycles in the corresponding graph, where the expression levels of TFs can themselves be the result of gene expression regulation. Fourthly, the result of the input and the response functions should match the observed gene expression patterns at any time, under any condition. The composition of the input and the response functions is sometimes called gene regulatory function (GRF).

We adopt partially in our approach the formalism of the RE:IN method [43]. In this method, the notion of the network is extended by associating a collection of possibly acting regulatory interactions (called *optional interactions*) with the Boolean Network. Then, one single structure (called *abstract model*) represents the space of many possible network topologies for the model. Each node is associated with a set of possible regulatory function types, which are monotonous with respect to the regulators (both activators and repressors). Each regulatory function type is actually a Boolean function, associated with a unique truth table, with activator/repressor states as input. Once all potential (*optional*) or existing (*definite*) interactions between the nodes of interest are defined, comes the model synthesis, i.e. the selection of relevant optional interactions, that give a state graph which is consistent with constraints based on a set of wet lab experiments. Once the "abstract" model is built, one can convert the problem of finding such relevant optional interactions and regulatory functions (*network inference* problem) into a SMT (satisfiability modulo theories) problem instance. To achieve this, a list of logical constraints on the allowed GRFs are built, according to a set of time-series gene expression patterns obtained from wet lab experiments. Then, these logical constraints will define the requirements for the selection of a subset of optional interactions. When these constraints are fed to a SMT solver, the latter can deduce one or several candidate models that match the results of all the wet-lab experiments used to build the logical constraints.

Each solution model comprises a subset of regulatory functions/GRFs (one for each node/gene) and a subset of relevant optional interactions, that describes

completely a plausible regulatory scenario which explains the experimental results. However, in the RE:IN framework, hypothesis testing for instance can be directly performed on the abstract model, without resorting to selection of a solution model.

Another method is exemplified by the GINsim model [37, 1, 25], where the model is built manually, but allows the user to check if the model dynamics match the experimental results. Other logical models have been suggested [36, 7, 5]. However, many of these approaches require multiple kinetic/logical parameter values [22], that are frequently difficult to obtain experimentally (due to a lack of technicians to produce experiments, or technological limitations), or computationally (because of large number of degrees of freedom in the model fitting process).

Our goal is to show that, as a proof-of-concept, firstly: the addition of cis-regulatory modules to network models can be done automatically, provided that binding patterns are known; and secondly, that it can lead to models that are constrained to more realistic dynamics, biologically speaking.

2. Methods

Our stated goal is to turn a "classic" Boolean network into a model that takes into account regulatory modules. Thus the formal definition of the modelling framework needs to be modified. This section aims at describing our formalism and its most important features.

CRM annotations correspond to the number, the names, the positions, and the function of detected cis-regulatory regions, or CRMs (cis-regulatory modules), around genes of interest (that is, appearing in the original model), and to the TFs binding to these regions. CRM annotations related to genes, and TF bindings to CRM were obtained via CisView database: `https://lgsun.irp.nia.nih.gov/geneindex/cisview` (both of the tested models are based on mice), and were processed manually: for each gene, we have filtered for "high-quality" CRMs that span at less than 2kbp upstream and downstream TFBSs (transcription factor biding sites) that

are related to the considered gene, and we have kept TFs that already appeared in the original models. CRM identifiers are the same as the ones from CisView.

2.1. Expanded Boolean model.

The objective of integrating regulatory modules to a regular Boolean network is to add more biological relevance to the model, by distinguishing between physically-assessed (which are generally considered more relevant) and other interactions.

An "expanded" (or "augmented") Boolean network is a quintuplet (G, C, T, I, F) where G is the set of genes or biological entities of interest, $C = \{C_g | g \in G\}$ is a set of sets of CRMs for each gene, and $T = \{T_{c_g} \subseteq G | c_g \in C_g, g \in G\}$ is the set of TF bindings to each CRM. Let us denote $T_g = \cup_{c_g \in C_g} T_{c_g} \subseteq G, g \in G$. Then:

$$\cup \cup_{g \in G} \cup_{c_g \in C_g} (T_{c_g} \times \{c_g\} \times \{+, -\}) \text{ (TF bindings)}$$

$$\cup \cup_{g \in G} (C_g \times \{g\} \times \{+, -\}) \text{ (cis-regulatory interactions). (1)}$$

where I is the set of gene-pairwise, signed, directed regulatory interactions. I can be decomposed into two disjoint sets I_{def} (set of regulatory interactions which have been assessed physically) and I_{opt} (set of "optional" regulatory interactions, which represent current biological assumptions on the system). F is the set of regulatory functions for each gene. The set of nodes V of the corresponding interaction graph is $V = G \cup \cup_{g \in G} C_g$, and the set of edges is I. Each Boolean function $f_g, g \in G$ has the following signature: $f_g : B^{|\prod_{c_g \in C_g} \{c_g\} \times \prod_{g' \in G - T} \{g'\}|} \rightarrow B$, and each Boolean function $f_{c_g}, c_g \in C_g, g \in G$ has the following signature: $f_{c_g} : B^{|\prod_{t \in T_{c_g}} \{t\}|} \rightarrow B$.

2.2. Decomposition of a GRF Into Its Regulatory Modules.

Moreover, adding regulatory modules to the model allows to account for modularity [9] and redundancy [27] of TF bindings for a given gene; it can ease the interpretation of the synthetized model, and allow a better understanding of the actual interactions between genes. Indeed, as stated in [40]: "An important feature of CRMs is the modularity of their activity, allowing CRM function to be assessed independently of each other". The regulatory functions for each CRM are actually the input function part in the GRF associated with the regulated gene. That is why it can be advantageous to choose preferentially multiple input with low in-degree, physically "decomposable" GRFs (with respect to the CRMs of the considered gene) over more complex, input regulatory functions with higher in-degree.

Let us denote the restriction of a Boolean vector q to the coordinates associated with variable in set V $q_{|V}$, that is, if $V = \{a, b\}$, then $q_{|V} = q[a, b]$ (vector q restricted to coordinates associated with variables a and b). As stated before, for a given gene $g \in G$, the gene regulatory function f_g is the "composition" (in the regular function composition sense) of the input functions $f_{c_g}, c_g \in C_g$, and the response function r_g, that is:

$$\begin{gathered} \forall q \in B^{|G|}, \text{ if } C_g = \{c_1, c_2, ..., c_n\}, \\ f_g(q) = r_g(f_{c_1}(q_{|T_{c_1}}), ..., f_{c_n}(q_{|T_{c_n}}), q_{|G-T}). \end{gathered} \tag{2}$$

If f_g can be written this way, then it is said to be *decomposable with respect to its regulatory modules/TF bindings*, or physically decomposable. If $C_g = \emptyset$, then f_g is always decomposable.

This "physical" decomposition can be seen as the decomposition of an arbitrary Boolean function, with respect to a given cover of its variable set (that is, a set of subsets of the set of variables that appear in the input Boolean function, such as their union is equal to the whole variable set present in this function). This has already been widely studied, and building such a decomposition requires recursively testing cases on the input function, and the functions

present in its decomposition. The ten theorems associated with each case, and the theoretical algorithm to build such a decomposition are presented in Curtis's book [12], and have originally been applied to electronic circuits. They are not dealt with here in detail, because Ashenhurst and Curtis's theorems are enough to establish the theoretical constraint on functions, which is introduced by an explicit implementation of the regulatory modules.

Apart from trying to describe more properly the regulatory mechanisms, searching for physically decomposable functions also allows to lower the number of possible regulatory Boolean functions, adding to the fixed subset of acceptable types of functions used in [43]. This is especially important when the number of genes is increasing, for computational reasons; indeed, Shannon (quoted by Curtis in [12]) showed that the number of such decomposable functions[1] greatly decreases when the number of input variables increases. Then, decomposability actually adds an implicit constraint on the input of regulatory functions, which potentially decreases the number of possible function candidates when their in-degree is high, thus the number of model candidates.

2.3. Monotonicity of a GRF With Respect to the Regulators.

We also want the resulting regulatory functions to satisfy a fixed set of *relevance-related* criteria, in order to consider it consistent with the concept

[1]Functions (GRFs) are counted modulo logical equivalence (i.e. equality of associated truth table) and when the time delay for TFs to bind to CRMs is considered negligible compared to time delays associated with other reactions: degradation of proteins, mRNAs, CRMs regulating the target gene, ... The latter means that regulatory functions associated with CRMs can be replaced by their own definition inside a GRF: if a gene is only regulated by a single activating CRM M, as defined by its GRF $f_g : q \to f_M(q)$, whose regulatory function is $f_M : q \to q(TF1) \wedge q(TF2)$, then f_g can be written as $f_g : q \to q(TF1) \wedge q(TF2)$. The proposition above still holds, just note that, for instance, with initial GRF $f_g : q \to q(A) \wedge q(B) \wedge q(C)$ for a gene g regulated by two CRMs M_1 and M_2, such as TFs A, B, C bind to both modules, different solutions (which give the same gene regulatory function) can arise in the expanded model: e.g. $f_g : q \to f_{M_1} \wedge f_{M_2}$, where $f_{M_1} : q \to q(A) \wedge q(B) \wedge q(C)$ and $f_{M_2} : q \to q(A)$, or $f_{M_1} : q \to q(A) \wedge q(B)$ and $f_{M_2} : q \to q(A) \wedge q(C)$, etc.

of regulatory function. We would like it to ensure the consistency with the description of GRFs provided by [11] (composition of input subfunctions and response function). Furthermore, if we had a reference model for the considered biological phenomenon, then we could have compared the two models using analysis methods for their equivalent homogeneous Markov chains [42]. For example, steady states or a similarity coefficient, according to a certain measure, could be computed. Since no reference is available, we decided to focus on the preservation of the *monotonicity* property for each gene regulatory function, as done in [16] – since the "bounded" property reported in [19] is trivially ensured by the fact that all regulatory functions here are Boolean.

A gene regulatory function f_g is intuitively said to be **monotonic** with respect to a given regulator r of its associated gene g iff. r is an activator (resp. a repressor) of g, then if r is activated, at fixed other regulators's states, then the output of associated regulatory function f_g is nondecreasing (resp. nonincreasing) [43, 21].

For all $q \in Q$ (where $Q = B^{|G|}$ is the set of system states in the GRN), this condition can be formally written as follows:

- If r is an activator of gene g:

$$f_g(G - \{r\} = q_{|G-\{r\}}, r = 1) \geq f_g(G - \{r\} = q_{|G-\{r\}}, r = 0) \quad (3)$$

- If r is a repressor of gene g:

$$f_g(G - \{r\} = q_{|G-\{r\}}, r = 1) \leq f_g(G - \{r\} = q_{|G-\{r\}}, r = 0) \quad (4)$$

which can be rewritten equivalently as:

$$\epsilon_g(r) \times f_g(G - \{r\} = q_{|G-\{r\}}, r = 1) \leq \epsilon_g(r) \times f_g(G - \{r\} = q_{|G-\{r\}}, r = 0), \quad (5)$$

where:

$$\epsilon_g(r) = \begin{array}{l} 0 \text{ if } r \text{ is not a regulator of gene } g \\ 1 \text{ if } r \text{ is a repressor of gene } g \\ -1 \text{ if } r \text{ is an activator of gene } g \end{array} . \quad (6)$$

Monotonicity has been sometimes described by the study of signs in the associated Jacobian matrix, but this last definition may not apply to systems with positive self-regulatory interactions [19]. It should be noted, that our definition cannot be directly applied to regulatory networks where the same TF represses and activates regulatory elements of its target gene. In this case, monotonicity of the regulatory functions in the resulting models cannot be assessed; however, our method still works (i.e. finds solutions) for models involving such bifunctional genes.

2.4. Network Inference Given Experiments And Abstract Expanded Model.

Provided an abstract model, such as described in Section 1, to which the expansion procedure has been applied, and an experiment file (that is, a file that contains the time-series gene expression patterns obtained from a wet-lab experiment, for a given subset of the genes present in the model), we use a similar method to [43] to infer a plausible model candidate. That means that this network satisfies all experiments provided in the input file. This inference is performed by a tool called SMT (Satisfiability Modulo Theories) solver, which finds solutions to a problem described with Boolean variables, that is equivalent to the network "reverse-engineering" problem. This method can be used directly on the expanded abstract model. However, in order to use fully the knowledge about regulatory modules, we slightly modified the procedure by adding two other types of constraints, such as an interaction between a gene and one of its regulating module is selected iff. at least one interaction between a TF and this regulatory module is selected:

Let g be a gene, c_g one of its associated regulatory modules, t a TF binding to c_g, and $s \in \{+, -\}$:

- If an optional interaction (t, c_g, s) is selected, then the interaction $(c_g, g, +)$ is also selected.

- If none of the interactions (t, c_g, s) is selected, then $(c_g, g, +)$ is **not** selected.

Using a SMT solver allows to return one or several network solutions that are completely consistent with the experimental results (described as binary time-series gene expression patterns). Such a method can then automatically enumerate multiple solutions, i.e. different sets of model parameters, up to a user-selected number, which give a same transcriptomic response. This is consistent with the biological fact that different arrangements of binding sites can give a same expression pattern [38]. Eventually, using this method of network inference is a flexible method, as logical conditions on solutions and regulatory function types can easily be modified, according to the considered biological phenomenon.

3. Analysis

Modelling explicitly cis-regulatory modules besides the regular Boolean network actually rules out some types of regulatory functions, that is, some types of Boolean functions. We assume in the following that solution models are a set of selected node pairwise interactions (among interactions defined as optional in the abstract model), and a set of GRFs, one per node, which characterizes the way the associated biological entity is regulated. Variation in either the set of selected interactions, or the set of GRFs (modulo logical equivalence), results in a different solution. Since we require decomposability property (because it directly affects the input of the GRFs), all the solution models, using GRFs that do not satisfy Definition 2.2, will not be representable by our expanded model. It means that, for instance, in the left-hand example depicted in Figure 2, with a non decomposable function with respect to its two CRMs, there is no way to select three Boolean functions (in the whole space of Boolean functions,

not only limited to the GRF types in RE:IN) for the gene and CRM nodes, such as the resulting GRF of this gene in the expanded model (as defined in Definition 2.2) is logically equivalent to the initial non decomposable GRF in the regular model. Biologically speaking, this constraint implements the TF binding and transcriptional regulation by the CRMs. The TFs cannot act on the gene expression without binding to CRMs, which adds a biological meaningful uncompressible time step to the regulation of the considered gene.

3.1. Regular Solution Models to Expanded Solution Models.

Let us focus on the Krumsiek model [23], for instance, which describes myeloid differentiation of myeloid progenitors to megakaryocytes, erythrocytes, granulocytes and monocytes in the mouse (network shown on the left-hand side of Figure 1). In this model, gene $Gfi1$ is activated when TF $C/EBP\alpha$ is active, and TFs $Egr1$, $Egr2$, $Nab2$ (denoted $EgrNab$ in Krumsiek et al.'s work) give a global repressed response. According to the results from RE:IN, if Q is the set of global system states, the associated Boolean GRF is $\forall q \in Q, f_{Gfi1}(q) = q(C/EBP\alpha) \wedge \neg q(EgrNab)$, with the same notations provided in Definition 2.2. According to CisView database [29], $Egr1$ binds to one promoter of $Gfi1$ noted $CM05020166$, and $C/EBP\alpha$ binds to another promoter of $Gfi1$ noted $CM05020167$. Thus the corresponding Boolean function is decomposable with respect to the TF bindings found: $\forall q \in Q, f_{Gfi1}(q) = f_{CM05020166}(q) \wedge f_{CM05020167}(q)$, where $f_{CM05020166}(q) = q(C/EBP\alpha)$, and $f_{CM05020167}(q) = \neg q(EgrNab)$.

Yet, there might be GRFs that cannot be decomposed this way. For instance, let us consider a gene G, such as it requires either the activation of both transcription factors t_1 and t_2, or only the repression of transcription factor t_3 (rule #13 in RE:IN). Let us denote the two cis-regulatory modules that regulate G respectively C_1 and C_2, such as t_1 and t_3 bind to module C_1, and t_2 binds to module C_2 (see Figure 2). Then the regulatory function f_G associated with gene G is: $\forall q \in Q, f_G(q) = (q(t_1) \wedge q(t_2)) \vee \neg q(t_3)$. It is not decomposable with respect to C_1 and C_2. It can be proven using Ashenhurst's criterion [3]

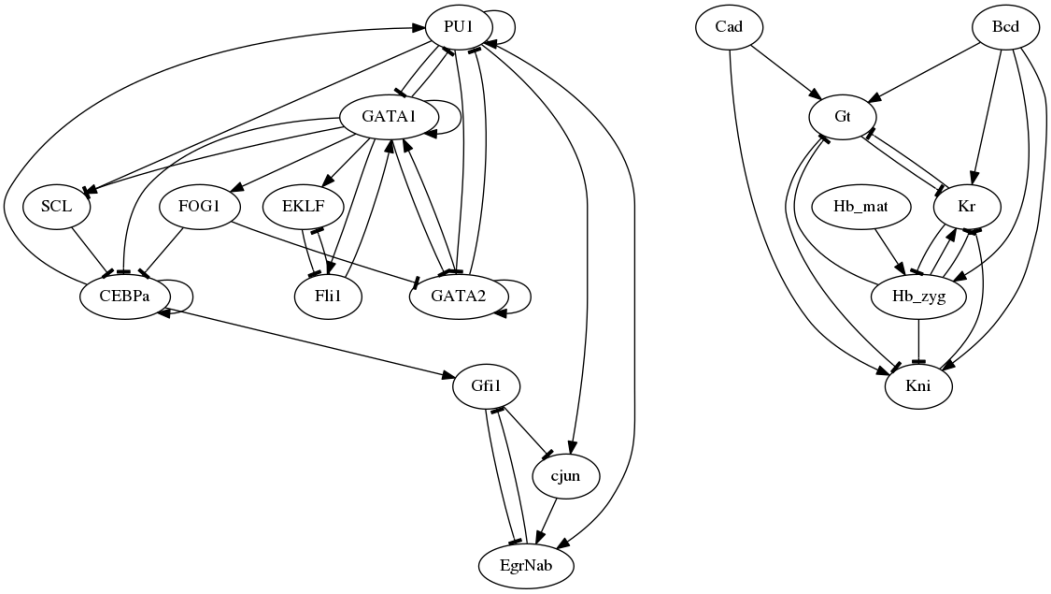


Figure 1: **Examples of Boolean Networks Quoted in the Paper**. Krumsiek model from [23] (*left*). Drosophila gap-gene model from [28]. Activating (resp. repressive) interactions have regular (resp. tee)-headed arrows.

for disjunctive decompositions of Boolean functions, for instance. Thus there are solution models found using the "regular" Boolean model that cannot be directly converted to solutions for the expanded model; i.e. solutions that involve non-decomposable functions.

Some types of GRFs may also not be monotonous (in the sense of Equation C.1). For instance, in the Drosophila gap-gene network, involved in early embryonic development (shown on the right-hand side of Figure 1), TF *Hunchback* can both repress and activate gap-gene *Krppel* [28], by repressing the distal and activating the proximal shadow enhancers of the latter [41]. But *Hunchback* does not seem to act directly on at least one of the shadow enhancers [41]. The same phenomenon occurs with the *eve 2+7* and *eve 3+7* enhancers of *stripe 7*: *Hunchback* directly represses *eve 3+7* [31], and indirectly activates *2+7* by counter-repression by *Caudal* binding [39]. Many TFs have a documented bifunctional behaviour [31], e.g. *Dorsal* [14]. The method we suggested here accepts models with bifunctional genes, and can generate solutions, if they exist. The main issue is to give a biological significance to this situation when dealing with CRMs. This issue can dealt with different strategies, depending on the molecular mechanism that enables bifunctionality. If bifunctionality is assumed to be concentration-dependent [28, 33], then it might be best modelled by a multi-level formalism [25], which can be converted into a Boolean model [32]. If it is caused by activator synergy, then a better definition of monotonicity which is not at single-gene level, but at (gene+enhancers) level, might be more useful – this is close to the representation of gene nodes in the cis-regulatory logics defined by [13]. It should be noticed that difficulties in representing non monotonous GRFs are not limited to expanded models, and present the same challenges in more classical GRN models. Because defining monotonicity only at single node-level might not be enough to describe a higher-order cis-regulatory property [4], that has an impact on several nodes at a time.

3.2. Expanded Solution Models to Regular Solution Models.

Let us then prove that solutions for expanded models satisfy some useful properties (see Appendix for proofs).

Properties of the Solutions For Expanded Models. A solution model, when it exists, returned by the SMT solver we defined on an expanded model, satisfies the three following conditions:

1. **Decomposability**: All GRFs are physically decomposable with respect to their regulatory modules (Equation 2).
2. **Consistency**: The model satisfies all gene expression patterns at each step in every experiment provided.
3. **Monotonicity**: All gene regulatory functions found are monotonic (Equation C.1).

3.3. Cases When No Solution Is Found.

There might be several reasons why no solution is found by the solver when using an expanded model: firstly, there might be no solution found by the solver when using the associated regular model, which means that the constraints provided (either on the gene nodes, or on the experiments/system dynamics) cannot be satisfied. It can be useful to check the experiment file in order to make it consistent, or to relax the constraints on the nodes (for instance, increase the number of possible types of regulatory functions that can be used for a given node). Secondly, all solutions found with the original model might involve non-decomposable functions for some nodes. Since we assume that a decomposable function, with respect to the actual binding sites, is biologically more relevant that an arbitrary Boolean function, it may mean that the TF bindings found are incorrect, or incomplete, for this node. Eventually, if one regulatory function in the model candidates cannot be described as one of the regulatory functions implemented in RE:IN, or as a Boolean composition of such subfunctions, then this GRF is considered invalid. Either the assumptions on the GRFs –that is, the subset of acceptable regulatory functions– should be modified, or either the

TF bindings might again be faulty. The monotonicity property, as defined in Equation C.1, does not provide a constraint on the resulting models, contrary to the decomposability property. Our lemma only ensures that, provided the regulatory function types are monotonic (in the sense of our definition), and that there is no bifunctional genes, the resulting models only involve monotonic regulatory functions. But, since the monotonicity is not actually used in the constraint building (unlike the decomposability property which acts on the inputs of GRFs), our method finds solutions (or not), regardless of the monotonicity of the regulatory function types and the presence of bifunctional TFs.

4. Results

We have tested our method on two published Boolean models about different differentiation pathways, and compared our results with the associated published data.

4.1. Dunn Mouse Pluripotency Model

This model aims at describing the phenomenon of cell differentiation in the mouse [15]. To decrease the computational cost and the running time, it has only been partially expanded, as described in Appendix. The original (non expanded) model and the experiments files needed were available on the web interface of RE:IN [2].

4.1.1. Analysis of the Resulting Model Candidates.

When provided the partially expanded model alone, the solver does not find any solution, because some data about the regulatory modules was incomplete. In order to overcome this issue, we applied the TF-inference procedure on only a subset of the edges of type "TF to gene" so that a viable solution could be found. For the corresponding solution displayed in the Appendix, TF-inference procedure has been applied to the following edges: "Sall4 to Sox2" and "Tcf3 to Esrrb". Adding only one of the preceding edges was not leading to a solution.

The decomposition of the gene regulatory functions with respect to their modules is clearly shown in the model. It helps interpreting the regulatory function of gene *Esrrb* for instance, by grouping regulatory interactions into modules. The importance of redundancy in the TF bindings can be shown when removing module $CM03005877$ for gene *Sox*2; for instance, TF *Nanog* binds to both of the modules $CM03005877$ and $CM03005876$. In absence of binding site on $CM03005877$, *Nanog* may bind to $CM03005876$. Same situation occurs when removing module $CM12018795$ of gene *Esrrb*: TF *Nanog* binds to both modules $CM12018772$ and $CM12018795$, allowing *Nanog* to act on the module $CM12018772$, when the binding site for $CM12018795$ is not available. All this redundancy might be helping the pluripotency system to maintain its behavior even in case of some potential mutations. Redundancy of TF bindings between regulatory elements then matches redundancy of gene expression patterns [4]. The expansion thus gives us a possible interpretation of different solutions found by the solver, in terms of binding site availability. In turn, this can lead to speculation on different possible molecular scenarii behind the identified regulatory functions.

We also noticed that some of the optional interactions that are defined as "required" by the authors of the original study (meaning that they should be selected in every model considered by RE:IN) are not selected in the expanded model. This includes many interactions including: Klf4 to Klf2, Klf4 to Tfcp2l1, Oct4 to Nanog, Sall4 to Klf2 and Oct4 to Tfcp2l1. While we cannot completely determine the necessity of these interactions purely on the grounds of a computational model, one can speculate that some of these interactions might be unnecessary, given the overall redundancy in a solution model. On the other hand, adding more constraints, based on perturbation experiments to the system, might prove these interactions to be indeed necessary. Nonetheless, in our opinion, these examples are indeed showing that our analysis can identify some potential points of improvement of the original model, by considering the actual connectivity and redundancy between the regulatory modules.

4.2. Collombet Lymphoid/Myeloid Differentiation Model

This model aims at describing the phenomena of lymphoid and myeloid differentiation into macrophages and B cells, and transdifferentiation from cells of B cell lineage into cells of Macrophage lineage [10]. To decrease the computational cost and the running time, it has only been partially expanded, as described in Appendix. The XML file from BioModels has been converted to a RE:IN-formatted file using our custom scripts (function *sbml2rein* in the GitHub). For the experiments file, we used the experiments describing stimulation with CSF1 and Il7 of lymphoid/myeloid progenitor cells. The expanded model generated from the RE:IN-formatted version of the original model was not modified, as the solver could find solutions. It could be noticed that the splitting of node associated with gene *Spi1* (or *PU.1*) into two nodes according to the concentration-level of its expression was indeed necessary for the expected behavior: considering the gene *Spi1* as a single Boolean variable lead to oscillations instead of a stable state.

4.2.1. Analysis of the Resulting Model Candidates.

The redundancy of TF bindings among regulatory modules connected to the same gene could be once again noticed (for instance, for *Runx1*, *Mef2c* and *Ebf1*). In each of the found model solutions, functions associated with regulatory modules can influence the regulation of the target gene in several different ways. For example, in one of the solutions, functions associated with modules that target gene *Egr2* appear in a disjunction (that is, the modules act complementarily); whereas for gene *Runx1*, the functions associated with its regulatory modules appear in a conjunction (that is, the modules operate cooperatively). It can also be noticed that, in this very solution, modules associated with gene *Ebf1* are not relevant for its regulatory function. Again, such situations may indicate that some of the interactions identified based on the binding ChipSeq data may indeed not be functional to a detectable degree, but more likely, this may indicate that more perturbation experiments are needed to uncover their relevance.

4.2.2. Study of The Stable States

In order to investigate more deeply into the preservation of stable states by the expansion operation, we have applied the following procedure to the Collombet model [10]:

1. We have downloaded the last stable release of GINsim and the associated GINsim model.
2. We have searched for all stable states using the static analysis procedure developped in GINsim (with no perturbation). GINsim finds 8 stable states, and among them, transcriptomic signatures related to the four cellular states reported in the original paper, namely, Macrophage cells, B-cells, granulocyte-monocyte progenitor (GMP) cells, and common myeloid progenitor (CMP) cells (according to the annotation provided in the GINsim model).
3. We then have checked whether these stable states found by GINsim are actually preserved by the (partial) expansion procedure described in our paper, by simulation.

For the partially expanded Collombet model (where CRM-gene regulatory interactions are all activatory and where CRM with less than one TF binding have been removed, with the splitting of Spi1 into two nodes), with the asynchronous model update, all 8 stable states from GINsim are indeed preserved, meaning that the corresponding phenotypes are stable states in the expanded model.

5. Conclusions

In this paper, we show an expansion procedure that works reliably for GRNs with Boolean regulatory functions. We show that it can be applied in realistic situations using two different biological models as examples; one originally developed in RE:IN formalism, and the other in GINsim. However, it is not restricted to either of the formalisms and can be quite easily applied to any

other Boolean GRN, provided a good coverage of the regulatory modules and their binding patterns. It is also possible to apply the expansion only to parts of the model if the ChipSeq data is not available for some TFs or some target genes.

Our method is based on the presence of discrete cis-regulatory modules. Hence, it requires accurate definition of such regions to yield a modular description of the gene regulatory network, associated with a higher degree of biological relevance. This leaves the user responsible for finding a method for delineating module boundaries, and their binding profiles. Since, for most model organisms, we have large scale databases of regulatory modules already available (e.g. CisView for the mouse and human [29], RedFly [17] for the Drosophila), we think that the identification of CRMs will be less problematic in the future.

Nonetheless, the user decision on the exact boundaries of CRMs may have profound influence on the resulting model, as we make extensive use of the assumption of functional independency of CRMs that regulate the same gene. This assumption justifies the separated computation of each CRM regulatory function, and leads to lowering the number of functions that need to be considered [40]. These functions are only allowed to depend on the TFs binding to their associated CRM. While in most studied systems, the experimental evidence is consistent with the independent action of enhancers, there might be contexts where this assumption can be violated. For example, it might fail in cases where *cooperative action* of some enhancers [26] was shown, i.e. when the regulatory response of a given enhancer depends on the actions of other enhancers. This might be alleviated by allowing cis-regulatory modules to have cooperative interaction terms in the associated response function.

Regarding the limitations of any such approach, it should be stressed, that providing diverse (in terms of types of perturbations) experiments to the solver, and very clear annotations of the perturbations, is the most important factor for the biological relevance of the results. And, in this area, there is much more work to be done (compared to identification of CRMs) in terms of experimental methods to measure gene expression patterns in diverse cell types, under diverse

conditions, and in multiple perturbation (gene knock-out or over expression) situations. Recent advances in single-cell sequencing are giving us hope that there will be a significant progress in this area in the coming years.

Our method can be compared to the "multiplex" model (introduced in [6]). This modelling framework is based on a regulatory automata formalism [36], but the number of parameter values is decreased using biological information about TF cooperativity or concurrency on a common target gene. This is implemented in practice by adding intermediate nodes between TFs and genes. Contrary to our method, which is limited to a single additional level of cis-regulation (via enhancers and promoters), the number of intermediate nodes can represent any type of complex needed to ensure the transcription of the target gene, thus can be arbitrary. Hence, rather than elements of regulation, these nodes represent interaction labels between TFs. Using the argument of decomposability we have exhibited before, having more than one intermediate "level" of nodes adds a stronger constraint on the system dynamics than in our framework. This reduces in practice the number of parameters needed, as noticed in [6], and in our work. Indeed, a combination of parameter values in their framework corresponds to a logical function; the decrease of the number of parameters is performed by "merging" the regulatory functions at each additional level of nodes. We believe our work formalizes, and explains in mathematical terms where this decrease in the number of parameters originates from. Moreover, we think that our work is a reasonable tradeoff between a biological description at a fixed level of accuracy (which takes into account TFs binding to cis-regulatory modules), and a tractable model synthesis (the more nodes and interactions are added, especially when they are labelled as optional, the longer it takes for the solver to generate the first solution).

In summary, we think that given the fast progress both in identification of CRMs, and quicker sampling of expression, we should have an increasing need for extension of regulatory network models to include actual connectivity based on molecular measurements. Our work gives a proof-of-concept implementation of one such approach, which has clear limitations, but may be modified and ex-

tended. We provide an open-source implementation of our method, in the hope that the community of researchers interested in gene regulation modelling will not only be able to apply our method to other models, but also create improved solutions to this problem.

Declarations of interest: none.

Roles of the authors: C. R. and B. W. have designed the modelling framework. C. R. has written the associated code, and run the tests. C. R. and B. W. have written the article. All authors have approved the final article.

Funding: This work was partially (B. W.) supported by the Polish National Science Centre grant decision No. [DEC 2015/16/W/NZ2/00314].

Supplementary material: Files related to the tests can be found on the project GitHub: `https://github.com/regulomics/expansion-network`.

References

[1] Abou-Jaoudé, W., Monteiro, P. T., Naldi, A., Grandclaudon, M., Soumelis, V., Chaouiya, C., & Thieffry, D. (2014). Model checking to assess t-helper cell plasticity. *Frontiers in bioengineering and biotechnology*, *2*.

[2] et al., Y. (). Re:in web interface. `https://rein.cloudapp.net/`.

[3] Ashenhurst, R. L. (1957). The decomposition of switching functions. In *Proceedings of an international symposium on the theory of switching, April 1957*.

[4] Barolo, S. (2012). Shadow enhancers: Frequently asked questions about distributed cis-regulatory information and enhancer redundancy. *Bioessays*, *34*, 135–141.

[5] Behaegel, J., Comet, J.-P., & Folschette, M. (2016). A hybrid hoare logic for gene network models. *arXiv preprint arXiv:1610.06715*, .

[6] Bernot, G., Comet, J.-P., & Khalis, Z. (2008). Gene regulatory networks with multiplexes. In *European simulation and modelling conference proceedings* (pp. 423–432).

[7] Bernot, G., Comet, J.-P., Richard, A., & Guespin, J. (2004). Application of formal methods to biological regulatory networks: extending thomas asynchronous logical approach with temporal logic. *Journal of theoretical biology*, *229*, 339–347.

[8] Bolouri, H., & Davidson, E. H. (2002). Modeling dna sequence-based cis-regulatory gene networks. *Developmental biology*, *246*, 2–13.

[9] Clune, J., Mouret, J.-B., & Lipson, H. (2013). The evolutionary origins of modularity. In *Proc. R. Soc. B* (p. 20122863). The Royal Society volume 280(1755).

[10] Collombet, S., van Oevelen, C., Ortega, J. L. S., Abou-Jaoudé, W., Di Stefano, B., Thomas-Chollier, M., Graf, T., & Thieffry, D. (2017). Logical

modeling of lymphoid and myeloid cell specification and transdifferentiation. *Proceedings of the National Academy of Sciences*, (p. 201610622).

[11] Crombach, A. (2017). Modelling the evolution of dynamic regulatory networks: Some critical insights. In *Evolutionary Biology: Self/Nonself Evolution, Species and Complex Traits Evolution, Methods and Concepts* (pp. 257–273). Springer.

[12] Curtis, H. (1962). *A new approach to the design of switching circuits*. Van Nostrand. URL: `https://books.google.pl/books?id=hbM8AAAAIAAJ`.

[13] Davidson, E. H. (2010). *The regulatory genome: gene regulatory networks in development and evolution*. Academic press.

[14] Dubnicoff, T., Valentine, S. A., Chen, G., Shi, T., Lengyel, J. A., Paroush, Z., & Courey, A. J. (1997). Conversion of dorsal from an activator to a repressor by the global corepressor groucho. *Genes & development*, *11*, 2952–2957.

[15] Dunn, S.-J., Martello, G., Yordanov, B., Emmott, S., & Smith, A. (2014). Defining an essential transcription factor program for naive pluripotency. *Science*, *344*, 1156–1160.

[16] Fages, F., Martinez, T., Rosenblueth, D. A., & Soliman, S. (2016). Influence systems vs reaction systems. In *International Conference on Computational Methods in Systems Biology* (pp. 98–115). Springer.

[17] Gallo, S. M., Gerrard, D. T., Miner, D., Simich, M., Des Soye, B., Bergman, C. M., & Halfon, M. S. (2010). Redfly v3. 0: toward a comprehensive database of transcriptional regulatory elements in drosophila. *Nucleic acids research*, *39*, D118–D123.

[18] Gunawardena, J. (2014). Models in biology:accurate descriptions of our pathetic thinking. *BMC biology*, *12*, 29.

[19] Kaderali, L., & Radde, N. (2008). Inferring gene regulatory networks from expression data. In *Computational Intelligence in Bioinformatics* (pp. 33–74). Springer.

[20] Kauffman, S. A. (1969). Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of theoretical biology*, *22*, 437–467.

[21] Klamt, S., Saez-Rodriguez, J., Lindquist, J. A., Simeoni, L., & Gilles, E. D. (2006). A methodology for the structural and functional analysis of signaling and regulatory networks. *BMC bioinformatics*, *7*, 56.

[22] Klarner, H., Streck, A., Šafránek, D., Kolčák, J., & Siebert, H. (2012). Parameter identification and model ranking of thomas networks. In *Computational Methods in Systems Biology* (pp. 207–226). Springer.

[23] Krumsiek, J., Marr, C., Schroeder, T., & Theis, F. J. (2011). Hierarchical differentiation of myeloid progenitors is encoded in the transcription factor network. *PloS one*, *6*, e22649.

[24] Mengistu, H., Huizinga, J., Mouret, J.-B., & Clune, J. (2016). The evolutionary origins of hierarchy. *PLoS computational biology*, *12*, e1004829.

[25] Naldi, A., Berenguier, D., Fauré, A., Lopez, F., Thieffry, D., & Chaouiya, C. (2009). Logical modelling of regulatory networks with ginsim 2.3. *Biosystems*, *97*, 134–139.

[26] Potvin, É., Beuret, L., Cadrin-Girard, J.-F., Carter, M., Roy, S., Tremblay, M., & Charron, J. (2010). Cooperative action of multiple cis-acting elements is required for n-myc expression in branchial arches: specific contribution of gata3. *Molecular and cellular biology*, *30*, 5348–5363.

[27] Ravasz, E., Somera, A. L., Mongru, D. A., Oltvai, Z. N., & Barabási, A.-L. (2002). Hierarchical organization of modularity in metabolic networks. *science*, *297*, 1551–1555.

[28] Sanchez, L., & Thieffry, D. (2001). A logical analysis of the drosophila gap-gene system. *Journal of theoretical Biology*, *211*, 115–141.

[29] Sharov, A. A., Dudekula, D. B., & Ko, M. S. (2006). Cisview: a browser and database of cis-regulatory modules predicted in the mouse genome. *DNA research*, *13*, 123–134.

[30] Shavit, Y., Yordanov, B., Dunn, S.-J., Wintersteiger, C. M., Hamadi, Y., & Kugler, H. (2015). Switching gene regulatory networks. In *International Conference on Information Processing in Cells and Tissues* (pp. 131–144). Springer.

[31] Staller, M. V., Vincent, B. J., Bragdon, M. D., Lydiard-Martin, T., Wunderlich, Z., Estrada, J., & DePace, A. H. (2015). Shadow enhancers enable hunchback bifunctionality in the drosophila embryo. *Proceedings of the National Academy of Sciences*, *112*, 785–790.

[32] Stoll, G., Caron, B., Viara, E., Dugourd, A., Zinovyev, A., Naldi, A., Kroemer, G., Barillot, E., & Calzone, L. (2017). Maboss 2.0: an environment for stochastic boolean modeling. *Bioinformatics*, *33*, 2226–2228.

[33] Struffi, P., Corado, M., Kaplan, L., Yu, D., Rushlow, C., & Small, S. (2011). Combinatorial activation and concentration-dependent repression of the drosophila even skipped stripe 3+ 7 enhancer. *Development*, *138*, 4291–4299.

[34] Sugita, M. (1963). Functional analysis of chemical systems in vivo using a logical circuit equivalent. ii. the idea of a molecular automaton. *Journal of Theoretical Biology*, *4*, 179–192.

[35] Thomas, R. (1973). Boolean formalization of genetic control circuits. *Journal of theoretical biology*, *42*, 563–585.

[36] Thomas, R. (1991). Regulatory networks seen as asynchronous automata: a logical description. *Journal of theoretical biology*, *153*, 1–23.

[37] Traynard, P., Fauré, A., Fages, F., & Thieffry, D. (2016). Logical model specification aided by model-checking techniques: application to the mammalian cell cycle regulation. *Bioinformatics*, *32*, i772–i780.

[38] Veitia, R. A. (2008). One thousand and one ways of making functionally similar transcriptional enhancers. *Bioessays*, *30*, 1052–1057.

[39] Vincent, B. J., Staller, M. V., Lopez-Rivera, F., Bragdon, M. D., Wunderlich, Z., Estrada, J., & DePace, A. H. (2017). Caudal counter-represses hunchback to regulate even-skipped stripe 2 expression in drosophila embryos. *bioRxiv*, (p. 226373).

[40] Wilczynski, B., & Furlong, E. E. (2010). Challenges for modeling global gene regulatory networks during development: insights from drosophila. *Developmental biology*, *340*, 161–169.

[41] Wunderlich, Z., Bragdon, M. D., Vincent, B. J., White, J. A., Estrada, J., & DePace, A. H. (2015). Krüppel expression levels are maintained through compensatory evolution of shadow enhancers. *Cell reports*, *12*, 1740–1747.

[42] Xiao, Y. (2009). A tutorial on analysis and simulation of boolean gene regulatory network models. *Current genomics*, *10*, 511–525.

[43] Yordanov, B., Dunn, S.-J., Kugler, H., Smith, A., Martello, G., & Emmott, S. (2016). A method to identify and analyze biological programs through automated reasoning. *NPJ systems biology and applications*, *2*, 16010.

[44] Zinzen, R. P., Girardot, C., Gagneur, J., Braun, M., & Furlong, E. E. (2009). Combinatorial binding predicts spatio-temporal cis-regulatory activity. *Nature*, *462*, 65.

Appendix A. Methods

Appendix A.1. Boolean Network Expansion Procedure

A "fully expanded" model comprises all the CRMs which regulate one (or several) genes of the original model. A "partially expanded" model is only componed of "relevant" CRMs among the latter, according to a given criterion. We also explain how TF bindings to known regulatory modules can be inferred with such a type of model.

Appendix A.1.1. Full Expansion.

Let us denote G the set of genes which appear in the original model M, $I = I_{def} \uplus I_{opt}$ the set of interactions (which is a disjoint union of definite and optional interactions) in this model. Let C be the set of cis-regulatory modules: C_g, $g \in G$ is the set of CRMs that regulate gene g, and let T_{c_g} be the set of TF bindings to $c_g \in C_g, g \in G$, and $T_g = \cup_{c_g \in C_g} T_{c_g}$. The notation $(a, b, .)$ means either an activating interaction $(a, b, +)$ of gene a on gene b, or a repressive interaction $(a, b, -)$ of gene a on gene b, according to the context.

From a model $M(G, I)$ and $(C = \cup_{g \in G} C_g, T = \cup_{g \in G} T_g)$, we build a "fully expanded" model $M'(G', I' = I'_{def} \uplus I'_{opt})$ such as:

$$G' = G \cup \cup_{g \in G} C_g \ . \tag{A.1}$$

$$\cup \{(g_1, g_2, .) | (g_1, g_2, .) \in I_{def}, g_1 \in G - T, g_2 \in G\}$$

$$\cup \{(g_1, g_2, .) | (g_1, g_2, .) \in I_{def}, g_1 \in T, g_2 \in G, C_{g_2} = \emptyset\} \ . \tag{A.2}$$

$$\cup\{(t, c_g, .)|(t, g, .) \in I_{opt}, t \in T_{c_g}\}$$

$$\cup\{(g_1, g_2, .)|g_1 \in G - T, g_2 \in G, (g_1, g_2, .) \in I_{opt}\}$$

$$\cup\{(g_1, g_2, .)|g_1 \in T, g_2 \in G, C_{g_2} = \emptyset, (g_1, g_2, .) \in I_{opt}\}. \quad \text{(A.3)}$$

One can notice that only activating interactions are set between a CRM and the gene(s) it regulates. This is because a CRM that has a repressive impact on the expression level (cis-regulatory information as black arrows in first case in Figure A.1) can equivalently be modelled: either by a CRM linked to the target gene with a repressive interaction (middle case in Figure A.1); either by a CRM with an activating interaction, which associated input function uses negated values of the TF expression levels (third case in Figure A.1, with TF input negation in parentheses). This case corresponds to an input function giving a value above the threshold of the response function, as such described in the "problem statement" section of the core paper. This negation will be sound because, in the regulatory function types provided in RE:IN [43], activators and repressors have a symmetric role in the regulation.

If no regulatory module is known/deemed useful (see next section) for a given gene g (i.e. $C_g = \emptyset$), then all regulatory interactions present in the initial model are added to the expanded model (even those having as input a TF), in order to comply with Definition 2.2. Thus, according to this definition, the regulatory function associated with gene g will automatically be decomposable. The main reason why we decided to use this strategy instead of, for instance, creating one "shadow regulatory" node to which would be bound all interactions of type "TF to g" (that would be **not** necessarily match a known shadow enhancer as defined in [4]), is because the absence of regulatory elements is probably due to incomplete data; thus, adding one node provides a constraint on the

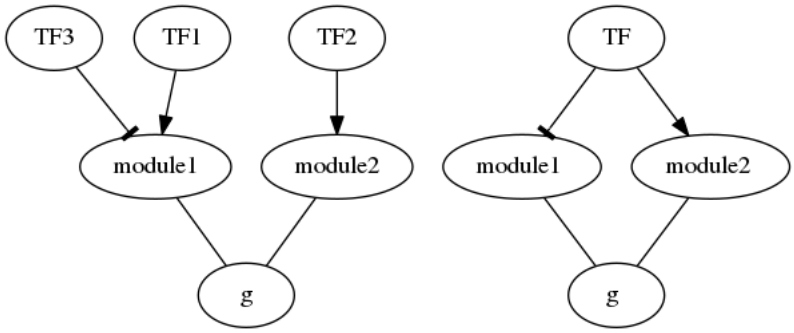


Figure 2: **Counter-Examples to Decomposability and Monotonicity**. Example of a non-decomposable regulatory function (where three TFs, $t1$, $t2$, and $t3$, bind to CRMs *module1*, *module2*, that regulate gene g, which needs either both $t1$ and $t2$ active, either $t3$ inactive). Arrow heads represent the interaction effects on each CRM: regular (activating), tee-headed inhibiting. Undirected edges are cis-regulatory connections (*left*). Example of a gene g regulated by bifunctional TF t, which binds to both of the regulatory modules of g.

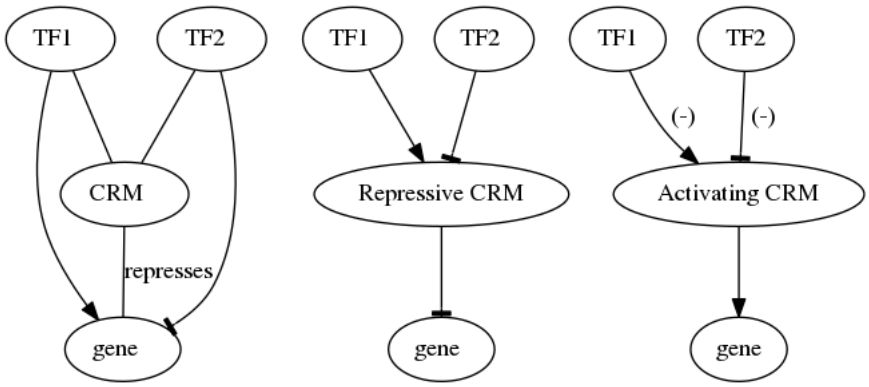


Figure A.1: **Converting a Regular Model to an Expanded Model**. The initial model (first figure from the left) can be modelled equivalently in two ways (provided a given CRM which regulates the gene and to which TFs 1 and 2 bind): either according to the middle figure, either according to the right-hand figure. Undirected edges are TF bindings and cis-regulatory interactions. Tee-headed (resp. regular) arrows are inhibitory (resp. activating) interactions. The "(-)" sign means that the TF input is negated in the function associated with the CRM it binds to.

decomposability of the regulatory function that might be incorrect (e.g. there might be more than one regulatory element in reality, and any function matching the experimental results is deemed non-decomposable whereas it can actually be fit). Another strategy would have been removing all interactions of type "TF to gene", but then again, incompleteness of data could have caused the lack of regulatory elements, and thus the model itself would be incomplete, and unable to find a solution that explains correctly the experiments. The main goal of a solution model found by the solver is to generate falsifiable hypotheses from current, possibly incorrect or incomplete, assumptions; not necessarily to give the ground truth biological mechanism [18]. If the solution returned is found to be wrong, then the abstract model automatically expanded might be modified, and tested again.

If one is confident enough in the TF binding data they have, TF bindings non documented in the model can also be added. Non-documented TF bindings are interactions such as $(t, g, .)$ does not belong to the initial interaction set of the model, where g is a gene, and t binds to a given CRM of g. The expansion procedure is then modified as follows:

$$\cup\{(t, c_g, .)|(t, g, .) \in I_{opt}, t \in T_{c_g}\}$$

$$\cup\{(t', c_g, +), (t', c_g, -)|(t', g, .) \notin I, t' \in T_{c_g}\}$$

$$\cup\{(g_1, g_2, .)|g_1 \in G - T, g_2 \in G, (g_1, g_2, .) \in I_{opt}\}. \quad \text{(A.4)}$$

Appendix A.1.2. Partial Expansion.

It can be seen that not all CRMs will actually be useful; for instance, if no known TF binding has been detected in a given CRM, or if there is only one CRM associated with a given gene (see preceding section). Partial expansion

allows to get a smaller model, by avoiding to build useless CRM nodes, thus reducing the computing time without altering the system dynamics. Let us denote C^v as the set of "valid" cis-regulatory modules, and the associated set of genes $G^v = \{g \in G, |C_g^v| \geq 1\}$, and:

$$\forall g \in G^v, C_g^v = \cup_{c_g \in C_g, c_g \neq \emptyset} \{c_g\} \text{ and } C^v = \cup_{g \in G^v} C_g^v \tag{A.5}$$

Then full expansion is applied to the model $M(G, I)$ with data $C^v, T^v = \{T_g^v | g \in G^v\}$.

Appendix A.1.3. TF Binding Inference.

This subsection explains how to transform an expanded model to identify further on unknown possible TF bindings to a CRM. Data from ChIP-seq experiments is often incomplete, thus we might miss some existing TF bindings that are not detected. The goal is then, for an initial interaction from TF g_1 on gene g_2, to infer which CRM regulating g_2 the TF might bind to. This is done by creating one optional interaction between the considered TF g_1 and every CRM of gene g_2. Note that, after applying this operation, interactions between TFs and CRMs are not necessarily direct anymore. Let us use the previous notations, and let us denote, for any subset $I_s \subseteq I$:

$$PDI_{I_s} = \{(g_1, g_2, .) | g_1 \in T - T_{g_2}, g_2 \in G \text{ such as } C_{g_2} \neq \emptyset, (g_1, g_2, .) \in I_s\}. \tag{A.6}$$

PDI_{I_s} is the set of possible direct interactions, i.e. TF binding interactions, in the interaction subset I_s. "$g_1 \in T - T_{g_2}$" means that g_1 is a known TF, but there is no known binding to any of the CRMs of gene g_2.

In the model, we then replace interaction subset PDI_{I_s}, $I_s \subseteq I$, by the following set:

$$PDI_{I_s}^{TF} = \cup_{(g_1, g_2, .) \in PDI_{I_s}} \{(g_1, c_{g_2}, .) | c_{g_2} \in C_{g_2}\} \tag{A.7}$$

Note that for all three types of expansion procedures, the algorithm preserves the GRF type conditions on all pre-existing nodes (that is, the possible regulatory roles played by activators and repressors of the considered node on its expression), and allows as a default value all possible regulatory functions for CRM nodes, so not to implement a too restrictive constraint on the model. This can be modified manually by manipulating the text file associated with the model, if needed.

Appendix B. About the Expansion Procedures

Appendix B.1. Tables for Some Expanded Models

Appendix B.2. Runtime for Dunn and Collombet Partially-Expanded Models

The following table show the runtimes to find the first model solution for the regular and partially-expanded versions of the Dunn [15] and Collombet [10] models. Tests were run on a single thread (non-parallel run) Intel Xeon 2.20GHz, all using less 6GB in RAM, using Python 2.7.6, Z3 4.5.1, and R 3.4.4 on Ubuntu 14.04.5 (64 bit).

Runtime-fold change is approximately 3 between regular and expanded versions of the Collombet model, while it goes up to 14 for the Dunn model. This can be explained by the difference in the number of optional-labelled interactions (see the tables in Subsection Appendix B.1). When more information is added to the abstract model, solutions are more quickly found, which seems consistent.

Appendix C. Proofs for the Solutions of Expanded Models

Appendix C.1. Proof of the Monotonicity of Template Functions In RE:IN

Monotonicity is defined such as, f_g is monotonic iff. for all nodes $g, r \in G$ and system state $q \in Q$:

$$\epsilon(r) \times f_g(G - \{r\} = q_{|G-\{r\}}, r = 1) \leq \epsilon(r) \times f_g(G - \{r\} = q_{|G-\{r\}}, r = 0), \quad \text{(C.1)}$$

Properties	**Original Model**	**Full Expansion Model**	**FE + TF-inference Model**
# nodes	16	40	40
- CRMs	- 0	- 24	- 24
# edges	83	54	155
- definite	- 13	- 9	- 7
- optional	- 70	- 44	- 148
# Edges TF to CRM	0	21	125
- definite	- 0	- 1	- 1
- optional	- 0	- 20	- 124

Table B.1: Statistics (1) For Fully Expanded Dunn Model Structure.

Properties	**Original Model**	**Partial Expansion Model**	**PE + TF-inference Model**
# nodes	16	30	30
- CRMs	- 0	- 14	- 14
# edges	83	85	135
- definite	- 13	- 12	- 12
- optional	- 70	- 73	- 123
# Edges TF to CRM	0	16	66
- definite	- 0	- 1	- 1
- optional	- 0	- 15	- 65

Table B.2: Statistics (2) For Partially Expanded Dunn Model Structure.

Properties	**Original Model**	**Full Expansion Model**	**FE + TF-inference Model**
# nodes	33	86	86
- CRMs	- 0	- 53	- 53
# edges	89	148	232
- definite	- 89	- 95	- 151
- optional	- 0	- 53	- 81
# Edges TF to CRM	0	28	126
- definite	- 0	- 28	- 28
- optional	- 0	- 0	- 98
# TF-gene edges to infer	0	0	98

Table B.3: Statistics (1) For Fully Expanded Collombet Model Structure.

Properties	**Original Model**	**Partial Expansion Model**	**PE + TF-inference Model**
# nodes **- CRMs**	33 - 0	59 - 26	59 - 26
# edges **- definite** **- optional**	89 - 89 - 0	127 - 101 - 26	187 - 101 - 86
# Edges TF to CRM **- definite** **- optional**	0 - 0 - 0	22 - 22 - 0	82 - 22 - 60
# TF-gene edges to infer	0	0	60

Table B.4: Statistics (2) For Partially Expanded Collombet Model Structure.

where:

$$\epsilon(r) = \begin{array}{l} 0 \text{ if } r \text{ is not a regulator of gene } g \\ 1 \text{ if } r \text{ is a repressor of gene } g \\ -1 \text{ if } r \text{ is an activator of gene } g \end{array} . \tag{C.2}$$

Let us denote in the remaining part of this section $f_{r=i} = f(V - \{r\} = q_{|V-\{r\}}, r = i)$, and q^i such as $q^i_{|V-\{r\}} = q_{|V-\{r\}}$ and $q^i(r) = i$, where $i \in \{0, 1\}$.

Appendix C.1.1. Closure by Regularization Function in RE:IN.

Let us assume that f_g is a monotonic GRF on a given GRN $G(V, I)$, where V is the set of nodes, and I the network topology (i.e. the positive and repressive interactions between nodes), associated with a gene $g \in V$. Then $reg(f) = q \rightarrow (f_g(q) \wedge \text{InducibleRegulation}(q)) \vee \text{RepressibleRegulation}(q)$ is monotonic.

Let $r \in V, q \in Q$. If r is not a regulator of g, EQ. C.2 is satisfied. If r is a regulator of g:

Using the monotonicity of function f, $reg(f)$ is thus monotonic.

Appendix C.1.2. Closure by Boolean Composition.

Let us assume that f, f' are two monotonic GRFs on a same given GRN $G(V, I)$, where V is the set of nodes, and I the network topology (i.e. the positive and repressive interactions between nodes), respectively associated with genes $g, g' \in V$. Then, if every activator (resp. repressor) of g (resp. g') that is a present regulator of g' (resp. g) is an activator (resp. repressor) of g' (resp. g), $f \wedge f' = q \rightarrow f(q) \wedge f'(q)$, $f \vee f' = q \rightarrow f(q) \vee f'(q)$ and $\neg f = q \rightarrow \neg f(q)$ are monotonic.

Function $f \wedge f'$ is the GRF of a (possibly intermediate) node g" of G that is active if and only if both nodes f and f' are active. This means that actually every regulator of g or of g', which is a causal variable in the logical formula associated with $f \wedge f'$, is a regulator of this node g". Let us consider r a regulator of g (symmetrically for g') that is a causal variable in the logical formula associated with $f \wedge f'$, and $q \in Q$ a system state ("x" means that the result is the same whatever the value of x is on the whole line).

Models	**Time for Original Model (sec.)**	**Time for Expanded Model (sec.)**
Collombet	28.67 (constraint building) + 25.20 (checking)	83.26 (constraint building) + 116.91 (checking)
Dunn	123.69 (constraint building) + 745.57 (checking)	260.29 (constraint building) + 12,689.97 (checking)

Table B.5: Time (in seconds) for the First Model Solution.

InducibleRegulation(q)	RepressibleRegulation(q)	$reg(f)_{r=1}$	$reg(f)_{r=0}$
1	0	$(f_{r=1} \wedge 1) \vee 0 = f_{r=1}$	$(f_{r=0} \wedge 1) \vee 0 = f_{r=0}$
0	0	$(f_{r=1} \wedge 0) \vee 0 = 0$	$(f_{r=0} \wedge 0) \vee 0 = 0$
0	1	$(f_{r=1} \wedge 0) \vee 1 = 1$	$(f_{r=0} \wedge 0) \vee 1 = 1$
1	1	$(f_{r=1} \wedge 1) \vee 1 = 1$	$(f_{r=0} \wedge 1) \vee 1 = 1$

Table C.1: Checking for Each Case of Regularization Function if Equation C.1Stands (provided f is monotonic).

r **activates** g	r **regulates** g'	$f_{r=1}$	$f_{r=0}$	$f'_{r=1}$	$f'_{r=0}$	**Is EQ. C.2 satisfied?**
0	0	0	0	x	x	$f_{r=1} \wedge f'_{r=1} = 0 \wedge x = f_{r=0} \wedge f'_{r=0}$
0	0	1	1	x	x	$f_{r=1} \wedge f'_{r=1} = 1 \wedge x = f_{r=0} \wedge f'_{r=0}$
0	0	0	1	x	x	$f_{r=1} \wedge f'_{r=1} = 0 \wedge x \leq 1 \wedge x = f_{r=0} \wedge f'_{r=0}$
0	1	0	0	0	0	$f_{r=1} \wedge f'_{r=1} = 0 = f_{r=0} \wedge f'_{r=0}$
0	1	1	1	0	0	$f_{r=1} \wedge f'_{r=1} = 0 = f_{r=0} \wedge f'_{r=0}$
0	1	0	0	1	1	$f_{r=1} \wedge f'_{r=1} = 0 = f_{r=0} \wedge f'_{r=0}$
0	1	1	1	1	1	$f_{r=1} \wedge f'_{r=1} = 1 = f_{r=0} \wedge f'_{r=0}$
0	1	0	1	0	0	$f_{r=1} \wedge f'_{r=1} \leq 1 \wedge 0 = f_{r=0} \wedge f'_{r=0}$
0	1	0	1	0	1	$f_{r=1} \wedge f'_{r=1} < 1 \wedge 1 = f_{r=0} \wedge f'_{r=0}$
0	1	0	0	0	1	$f_{r=1} \wedge f'_{r=1} \leq 0 \wedge 1 = f_{r=0} \wedge f'_{r=0}$
1	0	0	0	x	x	$f_{r=1} \wedge f'_{r=1} = 0 \wedge x = f_{r=0} \wedge f'_{r=0}$
1	0	1	1	x	x	$f_{r=1} \wedge f'_{r=1} = 1 \wedge x = f_{r=0} \wedge f'_{r=0}$
1	0	1	0	x	x	$f_{r=1} \wedge f'_{r=1} = 1 \wedge x \geq 0 \wedge x = f_{r=0} \wedge f'_{r=0}$
1	1	0	0	0	0	$f_{r=1} \wedge f'_{r=1} = 0 = f_{r=0} \wedge f'_{r=0}$
1	1	1	1	0	0	$f_{r=1} \wedge f'_{r=1} = 0 = f_{r=0} \wedge f'_{r=0}$
1	1	0	0	1	1	$f_{r=1} \wedge f'_{r=1} = 0 = f_{r=0} \wedge f'_{r=0}$
1	1	1	1	1	1	$f_{r=1} \wedge f'_{r=1} = 1 = f_{r=0} \wedge f'_{r=0}$
1	1	1	0	0	0	$f_{r=1} \wedge f'_{r=1} = 1 \wedge 0 \geq 0 \wedge 0 = f_{r=0} \wedge f'_{r=0}$
1	1	1	0	1	0	$f_{r=1} \wedge f'_{r=1} = 1 > 0 \wedge 0 = f_{r=0} \wedge f'_{r=0}$
1	1	0	0	1	0	$f_{r=1} \wedge f'_{r=1} = 0 \wedge 1 \geq 0 \wedge 0 = f_{r=0} \wedge f'_{r=0}$

Table C.2: Checking for Each Case of $\wedge$ Composition if Equation C.1Stands (provided f and f' are monotonic).

r **activates** g	r **regulates** g'	$f_{r=1}$	$f_{r=0}$	$f'_{r=1}$	$f'_{r=0}$	**Is EQ. C.2 satisfied?**
0	0	0	0	x	x	$f_{r=1} \vee f'_{r=1} = 0 \vee x = f_{r=0} \vee f'_{r=0}$
0	0	1	1	x	x	$f_{r=1} \vee f'_{r=1} = 1 \vee x = f_{r=0} \vee f'_{r=0}$
0	0	0	1	x	x	$f_{r=1} \vee f'_{r=1} = 0 \vee x = x \leq 1 \vee x = 1 = f_{r=0} \vee f'_{r=0}$
0	1	0	0	0	0	$f_{r=1} \vee f'_{r=1} = 0 = f_{r=0} \vee f'_{r=0}$
0	1	1	1	0	0	$f_{r=1} \vee f'_{r=1} = 1 = f_{r=0} \vee f'_{r=0}$
0	1	0	0	1	1	$f_{r=1} \vee f'_{r=1} = 1 = f_{r=0} \vee f'_{r=0}$
0	1	1	1	1	1	$f_{r=1} \vee f'_{r=1} = 1 = f_{r=0} \vee f'_{r=0}$
0	1	0	1	0	0	$f_{r=1} \vee f'_{r=1} = 0 < 1 \vee 0 = 1 = f_{r=0} \vee f'_{r=0}$
0	1	0	1	0	1	$f_{r=1} \vee f'_{r=1} = 0 < 1 \vee 1 = f_{r=0} \vee f'_{r=0}$
0	1	0	0	0	1	$f_{r=1} \vee f'_{r=1} = 0 < 0 \vee 1 = 1 = f_{r=0} \vee f'_{r=0}$
1	0	0	0	x	x	$f_{r=1} \vee f'_{r=1} = 0 \vee x = x = f_{r=0} \vee f'_{r=0}$
1	0	1	1	x	x	$f_{r=1} \vee f'_{r=1} = 1 \vee x = 1 = f_{r=0} \vee f'_{r=0}$
1	0	1	0	x	x	$f_{r=1} \vee f'_{r=1} = 1 \vee x = 1 \geq x = 0 \vee x = f_{r=0} \vee f'_{r=0}$
1	1	0	0	0	0	$f_{r=1} \vee f'_{r=1} = 0 = f_{r=0} \vee f'_{r=0}$
1	1	1	1	0	0	$f_{r=1} \vee f'_{r=1} = 1 = f_{r=0} \vee f'_{r=0}$
1	1	0	0	1	1	$f_{r=1} \vee f'_{r=1} = 1 = f_{r=0} \vee f'_{r=0}$
1	1	1	1	1	1	$f_{r=1} \vee f'_{r=1} = 1 = f_{r=0} \vee f'_{r=0}$
1	1	1	0	0	0	$f_{r=1} \vee f'_{r=1} = 1 \vee 0 = 1 > 0 \vee 0 = f_{r=0} \vee f'_{r=0}$
1	1	1	0	1	0	$f_{r=1} \vee f'_{r=1} = 1 > 0 \vee 0 = f_{r=0} \vee f'_{r=0}$
1	1	0	0	1	0	$f_{r=1} \vee f'_{r=1} = 0 \vee 1 = 1 > 0 \vee 0 = f_{r=0} \vee f'_{r=0}$

Table C.3: Checking for Each Case of $\vee$ Composition if Equation C.1Stands (provided f and f' are monotonic).

When the condition "every regulator of both genes g and g' has the same (positive or repressive) effect on g and g'" is not fulfilled, there is harder to define the sign of a given regulator: for instance, if $r, a \in V$, $f : q \to q[r] \wedge \neg q[a]$ and $f' : q \to \neg q[r] \wedge q[a]$, r has a repressive effect whenever a is present, and vice-versa ($f \vee f' = a \oplus r$). Our definition of monotonicity does not account for this sort of regulators. Since in the expansion procedure we have chosen, regulators of a given gene have the same effect on its CRMs than on the very gene in the regular model, the condition required above is satisfied. However, monotonicity is ill-defined for models having bifunctional genes (see paper).

$\neg f$ is the GRF for a node that is active if and only if g is inactive. All activators (resp. repressors) for g are repressors (resp. activators) for "$\neg g$".

Thus $\neg f$ is monotonic, for any monotonic function f.

Appendix C.1.3. Monotonicity of RE:IN Functions

Let us show that all 20 template functions introduced in RE:IN [43] are monotonic as defined by EQ. C.2 (without composing with the regularization function, since we have proved above that monotonicity was stable by this operation). For the remaining part of this section, let us denote $g \in G$ an arbitrary gene and r one of its regulators (otherwise, the condition in EQ. C.2 is always satisfied). Since we also proved that monotonicity was stable by boolean composition, let us show only that all "atom" terms of the functions are monotonic:

AllActivators(g, q), NoActivators(g, q), AllRepressors(g, q), NoRepressors(g, q), and $\#A(g,q) > \#R(g,q)$ (there are strictly more activators than repressors), and $\#A(g,q) > \#R(g,q) \vee (\#A(g,q) = \#R(g,q) \wedge q(g))$ (there are as many activators as repressors, and gene g is active in the current state) are monotonic. See [43] for the exact definition of those terms.

Note that the term "$\#A(g,q) = \#R(g,q) \wedge q(g)$" alone is **not** monotonic.

Appendix C.2. Proof of the Lemma

The lemma about the properties of solutions found by the RE:IN method on expanded models is as follows:

r **activates** g	$f_{r=1}$	$f_{r=0}$	Is EQ. C.2 satisfied?
0	0	0	$\neg f_{r=1} = 1 = \neg f_{r=0}$
0	0	1	$\neg f_{r=1} = 1 \geq 0 = \neg f_{r=0}$
0	1	1	$\neg f_{r=1} = 0 = \neg f_{r=0}$
1	0	0	$\neg f_{r=1} = 1 = \neg f_{r=0}$
1	1	0	$\neg f_{r=1} = 0 \leq 1 = \neg f_{r=0}$
1	1	1	$\neg f_{r=1} = 0 = \neg f_{r=0}$

Table C.4: Checking for Each Case of $\neg$ if Equation C.1Stands (provided f is monotonic).

r **activates** g	$f_{r=0}$	$f_{r=1}$
0	0	0
0	1	1
1	0	if $A(g, q^1) = \{r\}$, then 1, otherwise 0
1	1	impossible, because activator r is not active in q^0

Table C.5: Checking for Each Case of $f : q \rightarrow$ AllActivators(g, q) if Equation C.1Stands (provided f is monotonic).

r **activates** g	$f_{r=0}$	$f_{r=1}$
0	0	0
0	1	1
1	0	0
1	1	0

Table C.6: Checking for Each Case of $f : q \rightarrow$ NoActivators(g, q) if Equation C.1Stands (provided f is monotonic). When r activates g, then it represses node whose regulatory function is f.

r **activates** g	$f_{r=0}$	$f_{r=1}$
0	0	if $R(g, q^1) = \{r\}$, then 1, otherwise 0
0	1	impossible, because repressor r is not active in q^0
1	0	0
1	1	1

Table C.7: Checking for Each Case of $f : q \rightarrow$ AllRepressors(g, q) if Equation C.1Stands (provided f is monotonic). When r represses g, then it activates node whose regulatory function is f.

r **activates** g	$f_{r=0}$	$f_{r=1}$
0	0	0
0	1	0
1	0	0
1	1	1

Table C.8: Checking for Each Case of $f : q \rightarrow$ NoRepressors(g, q) if Equation C.1Stands (provided f is monotonic).

r **activates** g	$f_{r=0}$	$f_{r=1}$
0	0	0
0	1	if $\#A(g, q^0) = \#R(g, q^0) + 1$, then 0, otherwise 1
1	0	if $\#A(g, q^0) = \#R(g, q^0)$, then 1, otherwise 0
1	1	1

Table C.9: Checking for Each Case of $f : q \rightarrow \#A(g, q) > \#R(g, q)$ if Equation C.1Stands (provided f is monotonic).

r **activates** g	$q(g)$	$f_{r=0}$	$f_{r=1}$
0	0	0	0
0	1	0	0
0	0	1	0
0	1	1	if $\#A(g,q^0) = \#R(g,q^0)$, then 0, otherwise 1
1	0	0	0
1	1	0	if $\#A(g,q^0) = \#R(g,q^0) - 1$, then 1, otherwise 0
1	0	1	1
1	1	1	1

Table C.10: Checking for Each Case of $f : q \to \#A(g,q) > \#R(g,q) \vee (\#A(g,q) = \#R(g,q) \wedge q(g))$ if Equation C.1Stands (provided f is monotonic).

r **activates** g	$q(g)$	$f_{r=0}$	$f_{r=1}$
0	0	0	0
0	1	0	if $\#A(g,q^0) - \#R(g,q^0) = 1$, then 1, else 0
0	0	1	0
0	1	1	0
1	0	0	0
1	1	0	if $\#R(g,q^0) - \#A(g,q^0) = 1$, then 1, else 0
1	0	1	0
1	1	1	0

Table C.11: Checking for Each Case of $f : q \to \#A(g,q) = \#R(g,q) \wedge q(g)$ if Equation C.1Stands (provided f is monotonic).

A solution model, when it exists, returned by the SMT solver we defined on an expanded model, satisfies the three following conditions:

1. **Decomposability**: All GRFs are physically decomposable with respect to their regulatory modules (Equation 2).
2. **Consistency**: The model satisfies all gene expression patterns at each step in every experiment provided.
3. **Monotonicity**: All gene regulatory functions found are monotonic (Equation C.1).

1. The only nodes that are linked to a given gene are its regulatory modules, and genes that are not TFs.
2. The model is a solution of a SMT problem where experiment-related constraints have been implemented, so by construction it satisfies all the gene expression patterns at each given step present in every experiment.
3. Finally, it can be proven that every regulatory function template in RE:IN is monotonic, as defined in Equation C.1, and that monotonicity is preserved by Boolean composition by connectors $\wedge$, $\vee$, $\neg$ (see Subsection Appendix C.1). Then, let $g \in G$ be a gene, and f_g having the same form as described in Definition 2.2:

$$\begin{aligned} \forall q \in B^{|G|}, \text{ if } C_g = \{c_1, c_2, ..., c_n\}, \\ f_g(q) = r_g(f_{c_1}(q_{|T_{c_1}}), ..., f_{c_n}(q_{|T_{c_n}}), q_{|G-T}), \end{aligned} \tag{C.3}$$

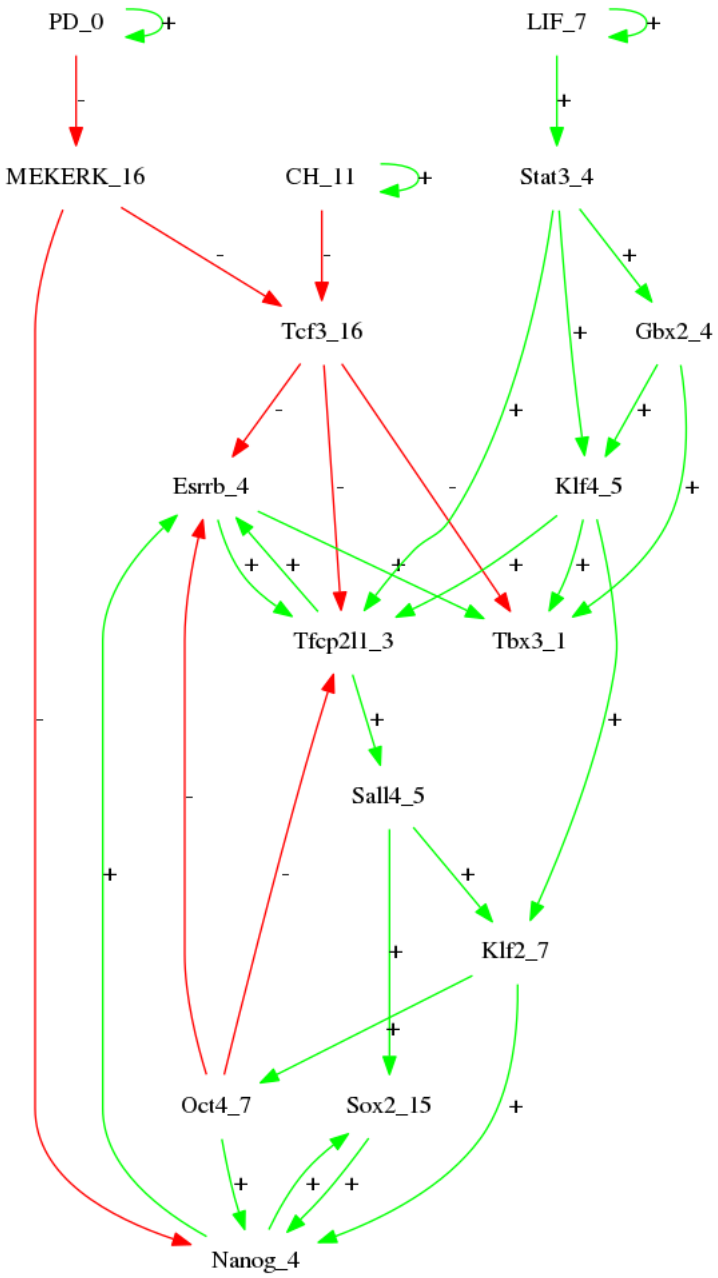
where r_g is the response function for gene g, and f_{c_i} is the input function associated with the i^{th} regulatory module c_i of g.

Each of the input functions is a regular regulatory function (meaning that it is one of the 20 available templates, or a function with only one variable), thus is monotonic. r_g only comprises connectors $\wedge$, $\vee$ and $\neg$. Thus f_g is also monotonic.

Appendix D. Figures for the Tested Models

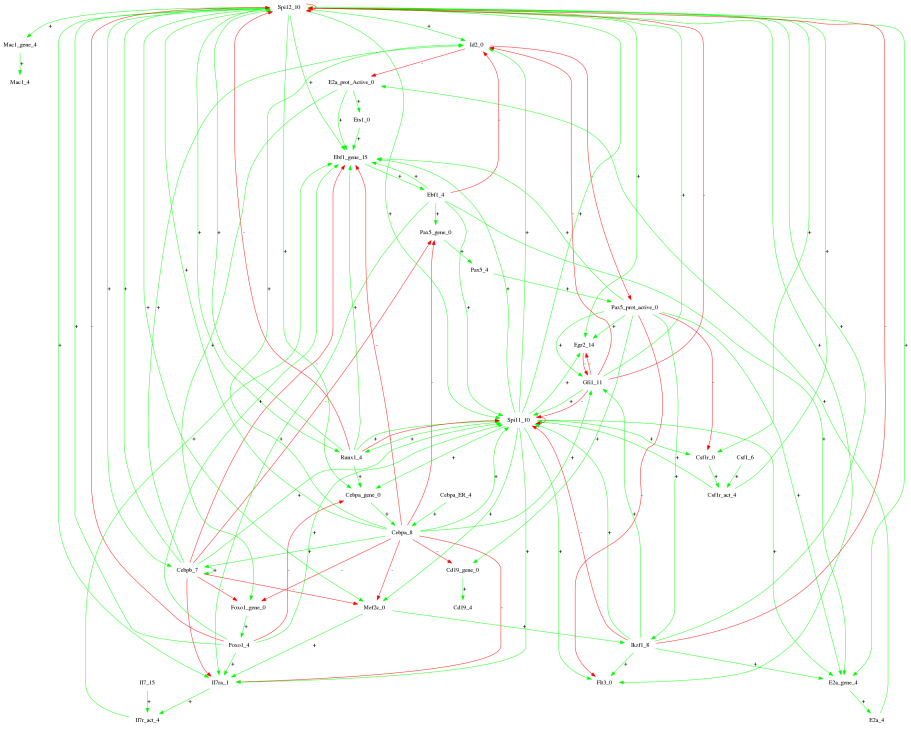
Since all solutions returned by the solver are equally valid with respect to the expansion procedure, we have decided to only display the first returned solution, in order to clearly illustrate our method.

[Colors should be used for these figures when printing].

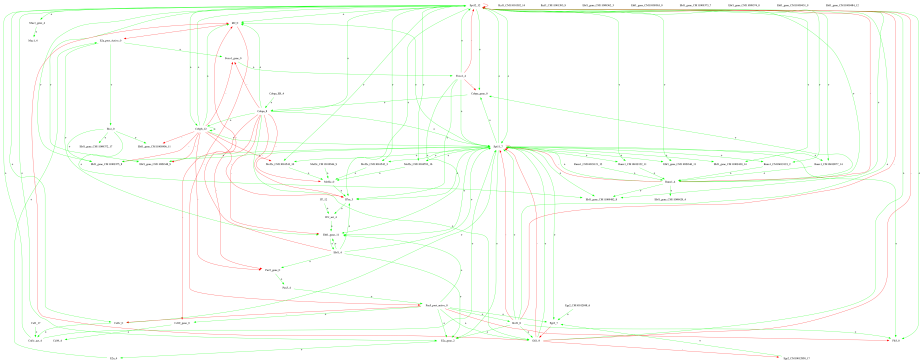


Result for the Regular Dunn Model. GRN Associated With the First Returned Solution For the Regular (Non Expanded) Dunn Model.

Result for the Expanded Dunn Model. GRN Associated With the First Returned Solution For the Expanded Dunn Model.



Result for the Regular Collombet Model. GRN Associated With the First Returned Solution For the Regular (Non Expanded) Collombet Model.



Result for the Expanded Collombet Model. GRN Associated With the First Returned Solution For the Expanded Collombet Model.

List of Figures

List of Tables