Constraint Programming in Community-based Gene Regulatory Network Inference

F. Fioretto and E. Pontelli

Dept. Computer Science
New Mexico State University
ffiorett,epontell@cs.nmsu.edu

Abstract. Gene Regulatory Network (GRN) inference is a major objective of Systems Biology. The complexity of biological systems and the lack of adequate data have posed many challenges to the inference problem. *Community networks* integrate predictions from individual methods in a "meta predictor", in order to compose the advantages of different methods and soften individual limitations. This paper proposes a novel methodology to integrate prediction ensembles using Constraint Programming, a declarative modeling paradigm, which allows the formulation of dependencies among components of the problem, enabling the integration of diverse forms of knowledge. The paper experimentally shows the potential of this method: the addition of biological constraints can offer improvements in the prediction accuracy, and the method shows promising results in assessing biological hypothesis using constraints.

1 Introduction

Within a cellular context, genes interact to orchestrate a multitude of important tasks. These interactions are regulated by different gene products, as proteins called *Transcription Factors (TFs)* and RNA, and they constitute an intricate machinery of regulation referred to as *Gene Regulatory Networks (GRNs).* In turn *GRN inference* describes the process of inferring the topology of a particular GRN. GRN inference from high-throughput data is of central importance in computational system biology. Its use is crucial in understanding important genetic diseases, such as cancer, and to devise effective medical interventions.

The availability of a wealth of genomic data has encouraged the development of diverse methods for GRN inference. However, data sets are quite heterogeneous in nature, containing information which is limited and difficult to analyze [20]. This reverberates on performance of GRN inference methods, which tend to be biased toward the type of data and experiments. For instance, methods based on linear models perform poorly on highly non-linear data, such as the one produced in presence of severe perturbations like gene knock-outs [11]. To alleviate these difficulties several alternatives have been proposed, such as integrating heterogeneous data into the inference model [17], or integrating a collection of predictions across different inference methods in *Community Networks (CNs)* [13, 14]. The former is a promising research direction but it has to face several challenges which span from how to relate different types of data to data sets normalization processes. The latter has the advantage of promoting the benefits of individual methods while smoothing out their drawbacks. Moreover it does not exclude the use of the former solution within the initial prediction set. The CN integration process poses many challenges, raising questions like: **(i)** how to take into account strengths and weaknesses of individual inference methods—e.g., the difficulty for Mutual Information (MI) or correlation based methods to discriminate TFs; and **(ii)** how to leverage additional information which cannot be taken into account by the individual methods.

In this paper, we propose a novel methodology based on *Constraint Programming (CP)* to integrate community predictions. CP is a declarative problem solving paradigm, where logical rules are used to model problem properties and to guide the construction of solutions. CP offers a natural environment where heterogeneous information can be actively handled. The use of constraint expressions allows the incremental refinements of a model. This is particularly suitable to take care of biological knowledge integration, when such knowledge cannot be directly handled by individual prediction methods.

We test our method on a set of 110 benchmarks proposed by the DREAM3 [14] and DREAM4 [16] challenges. We show increases in prediction accuracy with respect to a CN prediction based on the Borda count election method [13]. In addition, we show promising results in assessing biological hypotheses that could be used to guide the biological experimental design process.

2 Background

2.1 Basic Definitions

Gene Regulatory Networks. A GRN can be described by a weighted directed graph $G = (V, E)$, where V is the set of regulatory elements of the network and $E \subseteq V \times V \times [0, 1]$ is the set of regulatory interactions. The presence of an edge $\langle s, t, w\rangle \in E$ indicates that an interaction between the regulatory elements s and t is present with *confidence* value w. The number $|V|$ of regulatory elements of the GRN is referred to as its *size*. If the GRN has no uncertainty, then each edge in E has weight 1. In the problem of *GRN inference*, we are given the set of vertices V and a set of experiments describing the behavior of the regulatory elements. The goal is to accurately detect the set of regulatory interactions E.

Constraint Programming. CP is a declarative programming methodology commonly used to address combinatorial search problems. It focuses on capturing properties of the problem in the form of *constraints*, which are satisfied exclusively by solutions of the problem. CP models are fully declarative and elaboration tolerant, enabling the incremental integration of new knowledge.

A *Constraint Satisfaction Problem (CSP)* is formalized as a triple $\langle \mathcal{X}, \mathcal{D}, \mathcal{C}\rangle$, where $\mathcal{X} = \langle x_1, \ldots, x_n\rangle$ is an n-tuple of variables, $\mathcal{D} = \langle D_1, \ldots, D_n\rangle$ is a corresponding n-tuple of domains (and each D_i is a set of possible values for the variable x_i), and $\mathcal{C} = \langle C_1, \ldots, C_k\rangle$ is a k-tuple of constraints. A constraint C_j

over a set of variables $S_j \subseteq \mathcal{X}$ is a subset of the Cartesian product of the domains of the variables in S_j. A constraint represents the set of joint assignments that can be given to the tuple of variables in S_j. Given an n-tuple $A = \langle a_1, \dots, a_n \rangle$, we denote with $A|_{S_j}$ the restriction of the tuple to the variables in S_j.

A *solution* of a CSP $\langle \mathcal{X}, \mathcal{D}, \mathcal{C} \rangle$ is an n-tuple $A = \langle a_1, \dots, a_n \rangle$ where $a_i \in D_i$ (for $1 \leq i \leq n$) and $A|_{S_j} \in C_j$ (for $1 \leq j \leq k$)—i.e., the projection of A onto the set of variables involved in C_j satisfies the relation C_j. Typical resolution algorithms for CSP rely on efficient procedures to explore the search space of possible solutions and on *consistency methods*, where constraints are used to remove infeasible elements from the domains of the variables.

Related Work. A wide variety of GRN inference methods from expression data have been proposed [17]. These include: **(1)** Discrete models based on Boolean networks and Bayesian networks [11]; **(2)** Regression methods like *TIGRESS*—which imposes a regression problem to each gene; **(3)** Methods based on mutual information (MI) theory, such as *ARACNE* [15] and CLR [5], based on statistical likelihood of MI values. *Ensemble learning* has been explored for example by *GENIE3*, which uses a Random Forest approach [10]. Meta approaches have also been explored, such as *INFLEATOR*, based on re-sampling combining *median-corrected z-scores(MCZ)*, *time-lagged CLR (tlCLR)*, and linear ODE models [8]. *Community Networks (CNs)* integrate multiple inference methods to obtain a common consensus prediction. They have been shown to achieve better average confidence across different datasets and produce more robust results with respect to the individual methods being composed [13]. A simple scheme for combining predictions in a community network has been proposed in [13], where each interaction is re-scored by averaging the ranks it obtained within each of all the employed predictions. In the rest of the paper we will refer to it with CN_{rank}.
Constraint Technologies have been recently successfully applied in the field of System Biology [19]. For example, Answer Set Programming has been adopted to address problems in network inconsistencies detection [7] and in metabolic network analysis [18]. CP has been investigated to reason over discrete network models, where GRNs are modeled using multi-valued variables and transition rules [4]. In particular, CP is exploited to represent GRNs' possible dynamics [6].

3 Methods

The CN approach adopted in this work is built by combining four GRN inference procedures and creating an *inference ensemble*. Three of them are top-ranking methods that have been presented in the past DREAM competitions [13]: *(i) TIGRESS* [9], *(ii) INFLEATOR* [8], and *(iii) GENIE3* [10]. The fourth is an "off-the-shelf" widely adopted MI-based method *(CLR)* [5]. We use the *GP-DREAM* web platform (`http://dream.broadinstitute.org`) to develop the predictions from each of these methods. These methods have been selected to provide robustness and diversity, avoiding method redundancies that could potentially bias the inference ensemble.

3.1 Problem Formalization

Given a set of n genes, a GRN inference problem is formalized as a CSP $\langle \mathcal{X}, \mathcal{D}, \mathcal{C} \rangle$, with $\mathcal{X} = \langle x_1, \ldots, x_{n^2-n} \rangle$; each x_k describes a regulatory relation (without self regulations), and each $D_k = \{0, \ldots, 100\}$ is the set of possible confidence values associated with such relation. A variable x_i is said to be *assigned* if its associated domain D_i has been reduced to a singleton set. We adopt the notation $d(x_i)$ to indicate the value of an assigned variable x_i. For the sake of presentation, we denote with $x_{\langle s,t \rangle}$ the variable associated with the regulatory relation *"s regulates t"* and $D_{\langle s,t \rangle}$ its domain. A solution to the above CSP defines a GRN prediction $G = (V, E)$, with $V = \{1, \ldots, n\}$ and $E = \{\langle s, t, w \rangle \mid d(x_{\langle s,t \rangle}) > 0\}$, where $w = d(x_{\langle s,t \rangle})/100$.

Variables and Domains. The proposed CSP solution leverages the collection of GRN predictions obtained employing all the methods described in Sec. 3 by: **(1)** considerably reducing the size of the solution search space[1] and **(2)** taking into account the discrepancies among the community predictions. These objectives are achieved by mapping the edge confidence levels of each prediction to the corresponding CSP variable domain. The greater the agreement in the inference ensemble, the smaller is the set of values in the domain of the variable representing the relation being considered. The size of each domain captures the degree of uncertainty expressed by an edge prediction within the inference ensemble.

Let us consider a set of predictions $\mathcal{G}$ of a GRN $G = (V, E)$. We denote with G_j each prediction in the inference ensemble, and we denote with E_j the edges of E that have been identified by G_j. We also assume that each prediction has been normalized with respect to the ensemble itself. Furthermore, let θ_d $(0 \leq \theta_d \leq 1)$ be a given disagreement threshold. The procedure described in Alg. 1 reduces the content of the domains in $\mathcal{D}$ to at most three values. For each edge (s, t) we calculate the average confidence value (`w_rank`)—according to the *Borda* count election method, as presented in [13], which averages the ranked edge confidence values assigned by each prediction—and the discrepancy value (`w_d`) within $\mathcal{G}$ (line 4). The latter captures the ensemble prediction disagreement for a given edge, averaging the pairwise differences of the edge ranks associated to each prediction of the ensemble. If the discrepancy value exceeds the discrepancy threshold θ_d and the average confidence value is not strongly informative (line 6), we force the domain $D_{\langle s,t \rangle}$ to take account of the prediction disagreement by adding a variation of `w_d`/2 to the average confidence value. `fd` is the nearest integer function which converts a prediction confidence value into an integer domain encoding, and it is defined as: $\texttt{fd}(x) = \lfloor 100\,x + 0.5 \rfloor$. Line 5 ensures the presence of the value `w_rank` in $D_{\langle s,t \rangle}$. For a given prediction G_j, $\omega_j^{\#}(s, t)$ is the function ranking the prediction confidence for the edge (s, t) within the confidence values in E_j.

[1] An upper bound for the search space of a GRN inference problem of size n is 101^{n^2}.

Algorithm 1 Domain Variable Population

```
Require: normalized $G_j \in \mathcal{G}, \theta_d, G = (V, E)$
$J \leftarrow |\mathcal{G}|$
for all $(s,t) \in E$ do
    $B \leftarrow \emptyset$
    $(\texttt{w\_rank}, \texttt{w\_d}) \leftarrow \left( \frac{1}{J} \sum_{j=1}^{J} \omega_j^{\#}(s,t), \quad \frac{1}{\binom{J}{2}} \sum_{j=1}^{J} \sum_{i=j+1}^{J} \left| \omega_j^{\#}(s,t) - \omega_i^{\#}(s,t) \right| \right)$
    $B \leftarrow B \cup \{\texttt{fd}(\texttt{w\_rank})\}$
    if $\texttt{w\_d} \geq \theta_d \ \wedge \ 0.1 < \texttt{w\_rank} < 0.9$ then
        $B \leftarrow B \cup \left\{ \max\left(0, \texttt{fd}\big(\texttt{w\_rank} - \frac{\texttt{w\_d}}{2}\big)\right), \min\left(100, \texttt{fd}\big(\texttt{w\_rank} + \frac{\texttt{w\_d}}{2}\big)\right) \right\}$
    end if
    $D_{\langle s,t \rangle} \leftarrow D_{\langle s,t \rangle} \cap B$
end for
```

Constraint Modeling. Let us analyze the constraints that can be exploited to enforce the satisfaction of GRNs' specific properties and to take into account collective strengths and individual weaknesses of the CN predictions. Furthermore, we will discuss the propagation rules associated with the various constraints used to reduce the domain size of the variables ensuring constraint consistency.

Sparsity Constraints. It is widely accepted that the GRN machinery is controlled by a relatively small number of genes. Several state-of-the-art methods for reverse engineering GRN encourage sparsity in the inferred networks [13]. Nevertheless, when combining predictions in a community based approach, no guarantees on the sparsity of the resulting prediction can be provided. To address this issue we introduce a sparsity constraint, which is built from two more general constraints: atleast_k_ge and atmost_k_ge. They both enforce a relation among a set of variables and ensure that among the variables involved at least (resp. at most) k of them have values greater or equal than a threshold. Formally, the constraint:

$$\texttt{atleast\_k\_ge}(k, X, \theta): \quad \big|\{x_i \in X \mid d(x_i) > \theta\}\big| \geq k \tag{1}$$

enforces a lower bound (k) on the number of variables in X whose confidence value is greater than θ; the constraint:

$$\texttt{atmost\_k\_ge}(k, X, \theta): \quad \big|\{x_i \in X \mid d(x_i) > \theta\}\big| \leq k \tag{2}$$

limits to at most k the variables in X with confidence value greater than θ.

The propagation of the atmost_k_ge constraint is exploited during the solution search to enforce the property (2) by the following:

$$\texttt{atmost\_k\_ge}(k, X, \theta): \frac{S = \{x_i \in X \mid d(x_i) > \theta\},\ |S| = k}{\bigwedge\limits_{x_j \in X \setminus S} D_{\langle x_j \rangle} = D_{\langle x_j \rangle} \cap \{0, \ldots, \theta\}} \tag{3}$$

The atleast_k_ge cannot benefit from a powerful propagation rule, but early failures can be detected during the solution search by checking the upper bound on the number of variables not yet instantiated which satisfy property (1).

The sparsity constraint `g-sparsity` is a global constraint over the variables in X. It enforces lower and upper bounds on the number of edges whose confidence value is outside a given threshold. Formally, given $k_l, k_m, \theta_l, \theta_m$:

$$\texttt{atleast\_k\_ge}(k_l, X, \theta_l) \ \cap \ \texttt{atmost\_k\_ge}(k_m, X, \theta_m) \tag{4}$$

Redundant Edge Constraints. Several state-of-the-art inference methods rely on MI or correlation techniques; the community approach adopted for this work employs *CLR* and *INFLEATOR*, which are both MI-based methods (see Sec. 3). One of the disadvantages of such methods is the difficulty in speculating on the directionality of a given prediction. We define a constraint that has been effective in our experiments in detecting the edge directionality based on the collective decision of the CN predictions, among the non MI- or correlation-based methods.

Let us consider a collection of predictions $\mathcal{G} = \{G_1, \ldots, G_n\}$ for a GRN $G = (V, E)$, and a non-empty set of MI- or correlation-based methods $\mathcal{H} \subseteq \mathcal{G}$. An edge (t, s) is said to be *redundant* if:

$$\forall\, G_i \in \mathcal{G} \setminus \mathcal{H}\,. \quad \omega_i(t, s) < \omega_i(s, t) \wedge (\omega_i(s, t) - \omega_i(t, s)) > \beta \tag{5}$$

where $\omega_i(s, t) : V \times V \to [0, 1] \subseteq \mathbb{R}$ expresses the confidence value of the edge (s, t) in the prediction G_i. Given a redundant edge (t, s) we call the edge (s, t) the *required* edge. The `redundant_edge` constraint enforces a relation between two variables $x_{\langle s,t\rangle}$ and $x_{\langle t,s\rangle}$. Let X_R be the set of all the redundant and required variables.[2] For a pair of variables $x_{\langle s,t\rangle}, x_{\langle t,s\rangle} \in X_R$ the constraint:

$$\texttt{redundant\_edge}(x_{\langle s,t\rangle}, x_{\langle t,s\rangle}, \theta_e, L) : \quad x_{\langle s,t\rangle} > \theta_e \wedge \max(D_{\langle t,s\rangle}) < L \tag{6}$$

ensures that the confidence value assigned to the required variable $x_{\langle s,t\rangle}$ is greater than a given threshold value $\theta_e \in \mathbb{N}$, with $0 \leq \theta_e \leq 100$, and that the domain of the redundant edge variable $x_{\langle t,s\rangle}$ contains no values greater than L. The propagation of the `redundant_edge` constraint is exploited during the solution search to enforce property (6):

$$(x_{\langle s,t\rangle}, x_{\langle t,s\rangle}, \theta_e, L) : \frac{\min(D_{\langle s,t\rangle}) > \theta_e,\ \max(D_{\langle t,s\rangle}) \geq L}{D_{\langle t,s\rangle} =\ D_{\langle t,s\rangle} \cap \{0, \ldots, L-1\}} \tag{7}$$

Transcriptor Factor Constraints. Often, GRN specific information, such as sequence DNA-binding TFs or functional activity of a set of genes, is available from public sources (e.g., DBD [12]). Moreover, several studies show that similar mRNA expression profiles are likely to be regulated via the same mechanisms [1]. Not every method may be designed to handle such information, or this information can become available in an incremental fashion, and hence not suitably usable by prediction methods. We propose constraints that can directly incorporate such information in the CN model.

A regulatory element is a *Transcription-factor (TF)* if it regulates the production of other genes. This property is described through a relation on the out-degree

[2] $x_{\langle s,t\rangle}$ is redundant/required if the corresponding edge (s, t) is redundant/required.

of the involved gene for those edges with an adequate confidence value. The transc-factor constraint over a gene s is enforced by an $\texttt{atleast\_k\_ge}(k, X_s, \theta)$ constraint with $X_s = \{x_{\langle s,u \rangle} \in \mathcal{X} \mid u \in V\}$, and k representing the co-expressing degree, i.e., the number of genes targeted by the TF.

Multiple TFs can cooperate to regulate the transcription of specific genes; these are referred to as *Co-regulators*. When this information is available it can be expressed by a coregulator constraint. The latter involves two TFs, s' and s''; it enforces a relation over a set of variables X, to guarantee the existence of at least k elements that are co-regulated by both s' and s'' for which an interaction is predicted with confidence value greater than θ $(0 < \theta \leq 1)$. Formally:

$$\begin{aligned} \texttt{coregulator}(k, X, \theta) : \quad & \forall x_{\langle s',t' \rangle}, x_{\langle s'',t'' \rangle} \in X \\ \big| \{(s', s'', t') \mid s' \neq s'' \wedge t' = t'' \wedge d(x_{\langle s',t' \rangle}) > \theta \wedge d(x_{\langle s'',t'' \rangle}) > \theta\} \big| & \geq k \end{aligned} \tag{8}$$

Search Strategy. The proposed modeling of GRN prediction allows a great degree of flexibility in exploring the solution space. We implement two search strategies: **(1)** a classical prop-labeling tree exploration (DFS), where constraint propagation phases are interleaved with non-deterministic branching phases used to explore different value assignments to variables [2], and **(2)** a Monte Carlo (MC)-based prop-labeling tree exploration, which performs a random value assignment to each variable. We set a trial limit for the MC-based solution and a solution number limit for both strategies.

GRN Consensus. A challenge in GRN inference is the absence of a widely accepted objective function to drive the solution search. We decided to generate an ensemble of m solutions and propose three criteria to compute the final GRN prediction. Given a set of m solutions $S = \{S_1, \ldots, S_m\}$, where each $S_i = \langle a^i_1, \ldots, a^i_{n^2-n} \rangle$, let $S|_{x_k} = \bigcup_{i=1}^{m} \{a^i_k\}$ be the set of values assigned to the variable x_k in the different solutions, and $\texttt{freq}(a, k)$ be the function counting the occurrences of the value a among the assignments to x_k in the solution set. The consensus value a^*_k associated with the variable x_k is computed by:

- *Max Frequency:* $a^*_k = \arg\max_{a \in S|_{x_k}} (\texttt{freq}(a, k))$. This estimator rewards the edge confidence value appearing with the highest frequency in the solution set. The intuition is that edge-specific confidence values appearing in many solutions may be important for the satisfaction of the constraints.
- *Average:* $a^*_k = \frac{1}{m} \sum_{i=1}^{m} a^i_k$. It computes the average edge consensus among all solution in order to capture recurring predictive trends.
- *Weighted average:* $a^*_k = \frac{1}{\sum_{a \in S|_{x_k}} \texttt{freq}(a,k)^2} \sum_{a \in S|_{x_k}} \texttt{freq}(a, k)^2 a$. This estimator combines the intuitions of the two above by weighting the average edge confidence by the individual quadratic value frequencies.

We also investigated some potential *global* measures—i.e., acting collectively on the prediction values of all edges—in terms of the solution which minimizes the Hamming distance among all edge prediction values. These global measures were always outperformed by the three estimators discussed above.

3.2 A Case Study

We provide an example to illustrate our approach. We adopt the "E.coli2" network from the 10-node DREAM3 subchallenge [14] (Fig. 1). The target network has two co-regulators (G_1 and G_5) which are in turn regulated by gene G_9. The network has 15 interactions.

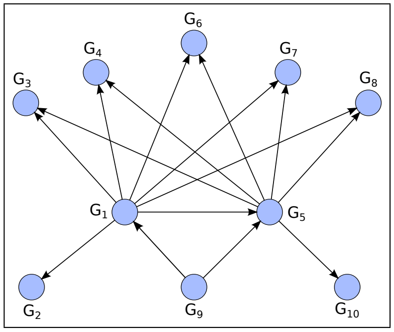

Fig. 1: An extract of E.coli GRN

Phase 1: CN Predictions. The inference ensemble was generated by feeding the datasets provided within the DREAM3 challenge to each of the four methods adopted in the community network schema (see Sec. 3). In addition, we generate a CN_{rank} as done in [13], and use it as baseline to build the domain variables (see Alg. 1) and for evaluation.

Phase 2: Modeling the CSP. The execution of Alg. 1 for the prediction disagreements analysis reduced the initial domain sizes to 1 for 64 cases, and to 3 for the others. The disagreement threshold was set to $\theta_d = 0.20$. As the inference ensemble adopted employs methods that may suffer from the *edge redundancy* problem, we impose a `redundant_edge` constraint for all the edge pairs $(s,t),(t,s)$ that satisfy the definition with $\beta = 0.15$ as:

$$\texttt{redundant\_edge}(x_{\langle s,t\rangle}, x_{\langle t,s\rangle}, 75, 50). \tag{r}$$

This constraint was able to reduce the value uncertainty for two additional variables—only one element in their domains can possibly satisfy the conditions above for any value choice of the required edge variable.

A sparsity constraint was imposed at a global level as:

$$\texttt{g-sparsity}: \quad \texttt{atleast\_k\_ge}(10, \mathcal{X}, 65) \cap \texttt{atmost\_k\_ge}(25, \mathcal{X}, 65). \tag{s}$$

Phase 3: Generating the Consensus. We performed 1,000 Monte Carlo samplings and return the first 100 solutions found, which we refer to as *Constrained Community Networks (CCNs).* To illustrate the effect of constraints integration on the CCNs we consider the best prediction returned by each CSP exhibiting a different combinations of the imposed constraints. We plot it as a graph containing all and only the edges of highest confidence necessary to make such graph weakly connected. These resulting predictions are illustrated in Fig. 2, together with the CN_{rank} (top-right). In each network the green edges (thick with filled arrows) denote the true positive predictions, the red edges (with empty arrows) denote the false positive predictions, and the gray (dotted) edges denote the false negatives. The results are also summarized in Table 1, where we report the AUC scores [3] for the best prediction (CCN_{best}) generated and for each CCN generated by the evaluation criteria presented in in Sec. 3.1.

Phase 4: Employing network specific information. Let us now model some specific information about the target network. The target network includes three TFs: G_1, G_5, G_9, which can be modeled via three `transc_factor` constraints as:

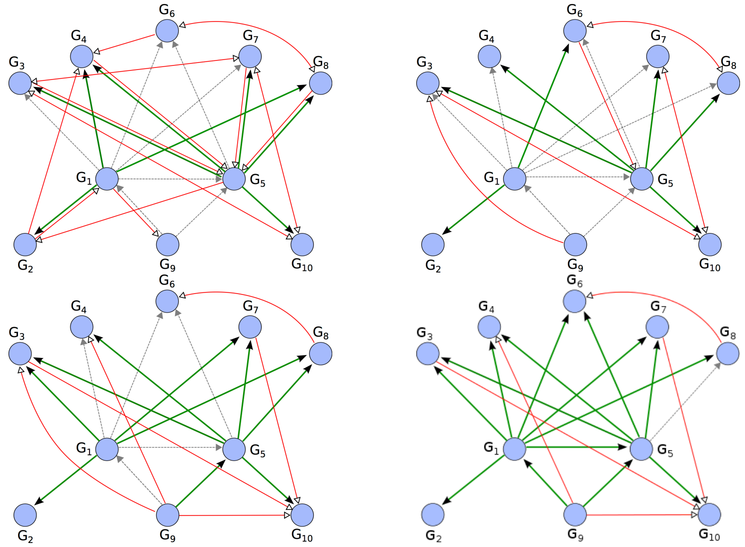

Fig. 2: The CN_{rank} consensus (top-right) and the CCN prediction after the integration of the redundant edge and sparsity constraints (top-left), the TF constraints (bottom-left) and Co-factor constraint (bottom-right).

$$\texttt{atleast\_k\_ge}(2, N_1, 85), \texttt{atleast\_k\_ge}(2, N_5, 85), \texttt{atleast\_k\_ge}(2, N_9, 85) \qquad (t)$$

with $N_i = \{x_{\langle i,s\rangle} \,|\, (\forall G_j \in \mathcal{G})\ \omega_j(i, s) > 0.10\}$. Note that $\omega_j(i, s)$ is the prediction confidence assigned to edge (i, s) by the inference method J in the prediction G_j. Fig. 2 and Table 1 show the improvements using the latter formalization.

Finally, speculation about the activity of genes G_1 and G_2 as co-regulators can be captured via a `coregulator` constraint expressed by:

$$\texttt{coregulator}(1, V, 75) \qquad (c)$$

with V defined as in (8) with $s' = 1, s'' = 5$. As shown in Fig. 2 and in Table 1, the application of this additional constraint produces further improvements.

Constr.	CN_{rank}	CCN_{best}	$\text{CCN}_{\text{max-f}}$	CCN_{avg}	$\text{CCN}_{\text{w-avg}}$
r	0.7271	0.8036	0.7556	0.7644	0.7751
s	0.7271	0.8044	0.7529	0.7164	0.7591
r, s	0.7271	0.8453	0.7778	0.7609	0.7760
r, s, t	0.7271	0.9209	0.7458	0.8489	0.8587
r, s, t, c	0.7271	0.9378	0.7929	0.8622	0.8729

Table 1: The effects of constraint integration on the AUC scores for the “Ecoli2” CCNs.

4 Results and Discussions

Benchmark Networks & Datasets. The proposed approach has been tested using benchmarks from the DREAM3 and DREAM4 competitions [14, 16]. The datasets adopted include the steady state expression levels for wild type and for knock-outs of every gene and the time-series data (a variable number of trajectories, depending on the size of the network). We generate 110 predictions: 50 of size 10, 25 of size 50, and 50 of size 100. For each problem we generate four consensus from each of the community methods described in Sec. 3 together with a consensus network constructed by averaging individual edges ranks (CN_{rank}).

Validation. To measure prediction accuracy against the corresponding reference network we adopted the AUC score [3], which relates the ratio between the *true positive* rate and the *false positive* rate. An AUC value of 0.5 corresponds to a random prediction, whereas a value of 1.0 indicates perfect prediction.

Settings. For each experiment we perform a $1,000$ Monte Carlo samplings and return the first 100 solutions found. We observed that the DFS was always outperformed by the MC search and therefore not reported. To guide the parameter selection for the sparsity constraint, given the thresholds θ_l, θ_m (see Eq. (4)), we identity the bounds k_l and k_m which would make the constraint unsatisfiable and use them to set the sparsity parameters. In this way, k_l and k_m are set so that they are bounded, respectively, above by $|\{x_i | x_i \in \mathcal{X} \wedge \max(D_{x_i}) > \theta_l\}|$, and below by $|\{x_i | x_i \in \mathcal{X} \wedge \min(D_{x_i}) > \theta_m\}|$, provided that $k_l < k_m$. The closer are their values to the respective bounds, the more restrictive is the constraint.

The `g_sparsity` (s) and `redundant_edge` (r) constraints have been enabled for all the experiments, with parameters $k_l = \frac{n^2}{10}$, $k_m = \frac{n^2}{4}$, $\theta_l, \theta_m, \theta_e$ in $\{65, 75\}$, and $L = 50$ (from ref. (4) and (6)). The latter was applied to all the pairs of edges satisfying (6) with $\beta = 0.15$. The disagreement threshold θ_d was set to 0.2 (see Alg. 1). We observed that such settings, for both search and constraints parameters, produced stable results across the whole benchmark set, which in turn was designed to capture a variety of network topologies to asseses GRN inference algorithms. We generate four CN consensus (CCNs), one for each estimator described in Sec. 3.1 ($\text{CCN}_{\text{max-f}}$, CCN_{avg}, $\text{CCN}_{\text{w-avg}}$) and CCN_{best}, as best prediction with respect to the AUC score, and compare them against CN_{rank}. The estimators-based CCNs may outperform the CCN_{best} as they are not elements of the set of solutions returned. We experimentally verified their constraints consistency, which was always satisfied.

Experiments. We first focused on examining the predicted CCNs using the sparsity and redundant edge constraints to leverage community-method features and networks properties. We categorize the benchmarks by DREAM edition and size, and average their respective AUC scores. Table 2 reports the percentage of the average AUC improvements for the best CCN_{best} and best $\text{CCN}_{\text{w-avg}}$ with respect to CN_{rank} across all the benchmarks (first two columns). Our choice of reporting only the weighted average estimator, among all those defined in Sect. 3.1, is driven by the observation that the former offers higher stability to parameter tuning and in general outperforms the other two. The CCNs achieved higher average prediction accuracy with respect to CN_{rank} for small and medium size networks, while performance improvements decreased for bigger networks.

	Dream3 10	**Dream4 10**	**Dream3 50**	**Dream3 100**	**Dream4 100**
$\mathrm{CCN}^{s\,r}_{\text{best}}$	+10.52	+7.01	+3.63	+1.75	−0.17
$\mathrm{CCN}^{s\,r}_{\text{w-avg}}$	+3.01	+1.96	+1.49	+0.43	+0.05
$\mathrm{CCN}^{s\,r\,t}_{\text{best}}$	+15.02	+8.43	+8.49	+4.13	+2.29
$\mathrm{CCN}^{s\,r\,t}_{\text{w-avg}}$	+5.42	+2.48	+6.32	+3.21	+4.21

Table 2: Average AUC score improvements (in percentage) with respect to $\mathrm{CN}_{\text{rank}}$

This is probably due to the high permissiveness of the CSP model for bigger networks. We show next that the application of additional constraints overcomes such effect.

We extended the set of constraints to include specific knowledge about individual networks. We enabled the transcriptor-factor constraint over a set of randomly selected genes which were verified TFs in the target networks. The TFs set sizes were chosen to be at most 30%, 15% and 10%, respectively, for the networks of size 10, 50 and 100; the co-expressing degree was set as $k=2$ and $\theta=85$. We performed 5 repetitions and for each TF t the set of possible regulators X has been chosen among the variables $x_{\langle t,s\rangle}$ such that $\omega^{\#}(t,s) > 0.25$. Moreover to promote such constraint we increased the uncertainty for the regulation $x_{\langle t,s\rangle}$ such that $\max D_{\langle t,s\rangle} \leq 50$. These parameters were chosen in accordance to the study presented in [1]. The integration of additional knowledge produced improvements of the AUC scores for both the *best* and the *weighted average* measures—see the last two columns of Table 2. A complete summary of the results is reported in Table 3.

The CCNs outperformed in general $\mathrm{CN}_{\text{rank}}$, and $\mathrm{CCN}_{\text{w-avg}}$ offers larger improvement for the bigger networks with respect to the version without the TF constraint. This supports our hypothesis that the addition of biological knowledge can better guide the predictions even if re-adopting the same inference ensemble. From a preliminary analysis of the incorrect predicted regulations supported by the TF constraint we observed that many of the erroneous inferences relate genes located in different regions of the graph. This effect could be attenuated by clustering the consensus graph for different connectivities, and targeting the TF constraint on the same cluster (if no prior knowledge on the specific TF is given). We plan to investigate this direction as future work.

4.1 Other uses: validating biological hypothesis.

The underlying technology adopted in this work allows us to test biological hypotheses, expressed in form of constraints, that may assist the phase of experimental design. The solver verifies the existence of a set of solutions consistent with the given hypotheses and its size can be related to confidence strength of the answer.

Consider a case study based on the "E.coli2" network presented in Sec. 3.2 (Fig. 1) to verify the hypothesis on the presence of a co-regulatory interaction. We perform 90 experiments, one for each pair of vertexes s', s'' of the network, with $s' \neq s''$, involved in a constraint of the type $\texttt{coregulator}(2, V, 75)$ with V defined as in (8) and employ the same settings as the one adopted in Sec. 3.2.

Among the entire set of problems only four satisfied the imposed hypothesis returning a non-empty set of solutions. These were the ones associated with the putative co-regulators $(G_1, G_5), (G_5, G_3), (G_5, G_6)$ and (G_1, G_8), generating respectively $115, 151, 32$ and 48 solutions. This notably restricts the number of possible biological tests to be performed, and also assigns higher probability to the first two co-regulators as they were able to generate more consistent solutions. We tried to shrink the set of putative co-regulators even more by imposing a stronger constraint: `coregulator`$(3, V, 70)$ with the same settings used above. This produced only one consistent set of solutions, associated with the pair (G_1, G_5), containing 151 elements. The result confirms the biological value of the experiment (see Figure 1). We stress that the hypothesis tested leverage the collective knowledge as well as additional network specific constraints (e.g., `sparsity`, `redundant_edge`) which are collectively handled in the CP model.

5 Conclusions

In this paper we introduced a novel approach based on CP to infer GRNs by integrating a collection of predictions in a CN. Our approach does not impose any hypothesis on the datasets adopted nor on the type of inference methods. We introduced a class of constraints able to (1) enforce the satisfaction of GRNs' specific properties and (2) take account of the community prediction collective agreements on each edge, and of method-specific limitations. Experiments over a set of 110 benchmarks proposed in past editions of the DREAM challenges show that our approach can consistently outperform the consensus networks constructed by averaging individual edges ranks, as proposed in [13] (up to 15.02% for small networks and 4.13% for big networks). We have shown how knowledge specific about target networks could provide further improvements in the AUC measure. This was possible as our model encourages the modular integration of biological knowledge, in form of logical rules, and proposes a set of candidate solutions satisfying the imposed constraints rather than an arbitrary one chosen among many. We introduced three estimators to compute a consensus from the set of consistent candidates and verified their consistency among the imposed constraints. We also show the potential of the proposed solution to assess biological hypotheses by verifying the consistency of the constrained model. This can be helpful in assisting the biological experimental design.

We plan to investigate new optimization measures by taking into account local and global network properties, e.g., the number of specific network motifs in a target GRN region, or the scale free degree in a given a portion of the graph. This can be achieved by including soft constraints in our model. We also plan to use this information to address method-specific biases towards different connectivity patterns. On the CP side, we will extend existing constraints, for instance by studying the most likely set where a TF constraint could be targeted, and model new constrains and propagators to capture different type of biological knowledge, such us information about cell conditions at the time of the experiments.

Network	CN_{rank}	$CCN^{s\,r}_{\text{best}}$	$CCN^{s\,r}_{\text{max-f}}$	$CCN^{s\,r}_{\text{avg}}$	$CCN^{s\,r}_{\text{w-avg}}$	$CCN^{s\,r\,t}_{\text{best}}$	$CCN^{s\,r\,t}_{\text{max-f}}$	$CCN^{s\,r\,t}_{\text{avg}}$	$CCN^{s\,r\,t}_{\text{w-avg}}$
		DREAM3 Size 10							
Ecoli1	0.7192	0.8101	0.7443	0.7422	0.7319	0.8642	0.7756	0.7664	0.7664
Ecoli2	0.7271	0.8453	0.7778	0.7778	0.7778	0.9209	0.8444	0.8676	0.8456
Yeast1	0.7413	0.8550	0.7625	0.7350	0.7637	0.8613	0.7325	0.7688	0.7685
Yeast2	0.6191	0.7145	0.6560	0.6123	0.6640	0.7606	0.6794	0.6646	0.6557
Yeast3	0.5428	0.6507	0.5742	0.5588	0.5622	0.6935	0.6457	0.5822	0.5842
Avg	0.6699	**0.7700**	**0.7023**	**0.6852**	**0.7000**	**0.8201**	**0.7355**	**0.7299**	**0.7241**
		DREAM4 Size 10							
Net1	0.7493	0.7858	0.7244	0.7324	0.7324	0.8471	0.7680	0.7600	0.7644
Net2	0.6943	0.7981	0.7618	0.7188	0.7188	0.8218	0.7331	0.7365	0.7348
Net3	0.8018	0.8622	0.8396	0.8329	0.8364	0.8649	0.8062	0.8329	0.8338
Net4	0.8501	0.9171	0.8771	0.8601	0.8601	0.9201	0.8511	0.8701	0.8721
Net5	0.8718	0.9541	0.9006	0.9006	0.9038	0.9348	0.8974	0.8921	0.8857
Avg	0.7934	**0.8635**	**0.8207**	**0.8090**	**0.8103**	**0.8777**	**0.8112**	**0.8183**	**0.8182**
		DREAM3 Size 50							
Ecoli1	0.6678	0.7317	0.6801	0.6871	0.6991	0.7919	0.7195	0.7724	0.7740
Ecoli2	0.7010	0.7214	0.7083	0.7075	0.7064	0.8205	0.7481	0.7954	0.7954
Yeast1	0.6539	0.6817	0.6545	0.6496	0.6586	0.7205	0.6520	0.6782	0.6842
Yeast2	0.6273	0.6609	0.6466	0.6429	0.6442	0.6866	0.6780	0.6712	0.6725
Yeast3	0.6181	0.6536	0.6236	0.6236	0.6340	0.6731	0.6360	0.6580	0.6579
Avg	0.6536	**0.6899**	**0.6626**	**0.6621**	**0.6685**	**0.7385**	**0.6867**	**0.7150**	**0.7168**
		DREAM3 Size 100							
Ecoli1	0.7704	0.7831	0.7647	0.7716	0.7746	0.8131	0.7921	0.8128	0.8128
Ecoli2	0.7152	0.7353	0.7179	0.7186	0.7226	0.7826	0.7479	0.7693	0.7692
Yeast1	0.6975	0.7141	0.7031	0.6984	0.7014	0.7337	0.7086	0.7181	0.7187
Yeast2	0.6116	0.6371	0.6116	0.6164	0.6134	0.6524	0.6201	0.6394	0.6438
Yeast3	0.5596	0.5723	0.5605	0.5629	0.5642	0.5794	0.5684	0.5633	0.5704
Avg	0.6709	**0.6884**	**0.6716**	**0.6736**	**0.6752**	**0.7122**	**0.6874**	**0.7006**	**0.7030**
		DREAM4 Size 100							
Net1	0.7829	0.7797	0.7523	0.7788	0.7785	0.7975	0.7606	0.8273	0.8273
Net2	0.7511	0.7396	0.7228	0.7549	0.7535	0.7773	0.7637	0.8085	0.8085
Net3	0.7158	0.7254	0.6956	0.7191	0.7200	0.7455	0.7288	0.7454	0.7492
Net4	0.7408	0.7380	0.7200	0.7429	0.7429	0.7604	0.7230	0.7652	0.7742
Net5	0.7222	0.7220	0.7054	0.7198	0.7205	0.7466	0.7241	0.7626	0.7643
Avg	0.7426	0.7409	0.7192	**0.7431**	**0.7431**	**0.7655**	0.7400	**0.7818**	**0.7847**

Table 3: AUC scores for the benchmark test set categorized by DREAM challenge and network size. The left part of the Table summarizes the CCNs results computed by enabling the redundant edge and the sparsity constraint. The right part of the Table summarizes the CCNs returned by extending the model to include the transcriptor factor constraints. The table reports the best predictions found by each estimator. The average AUC scores outperforming CN_{rank} are highlighted in bold.

References

1. D. Allocco et al. Quantifying the relationship between co-expression, co-regulation and gene function. *BMC Bioinformatics*, 5(1):18+, Feb. 2004.
2. K. Apt. *Principles of Constraint Programming.* Cambridge University Press, 2009.
3. P. Baldi et al. Assessing the accuracy of prediction algorithms for classification: an overview. *Bioinformatics*, 16(5):412–424, 2000.
4. F. Corblin, E. Fanchon, and L. Trilling. Applications of a formal approach to decipher discrete genetic networks. *BMC Bioinformatics*, 11:385, 2010.
5. J. J. Faith et al. Large-scale mapping and validation of escherichia coli transcriptional regulation from a compendium of expression profiles. *PLoS Biol*, 5(1), 2007.
6. J. Fromentin et al. Analysing gene regulatory networks by both constraint programming and model-checking. *Conf Proc IEEE Eng Med Biol Soc.*, 4595–8, 2007.
7. M. Gebser et al. Detecting inconsistencies in large biological networks with answer set programming. *CoRR*, abs/1007.0134, 2010.
8. A. Greenfield et al. Dream4: Combining genetic and dynamic information to identify biological networks and dynamical models. *PLoS ONE*, 5(10):e13397, 10 2010.
9. A. Haury et al. Tigress: Trustful inference of gene regulation using stability selection. *BMC Syst Biol*, 6(1):145, 2012.
10. V. A. Huynh-Thu et al. Inferring regulatory networks from expression data using tree-based methods. *PLoS ONE*, 5(9):e12776, 09 2010.
11. S. Kim et al. Dynamic Bayesian network and nonparametric regression for modeling of GRNs from time series gene expression data. *Biosystems*, 104–113, 2003.
12. S. K. Kummerfeld and S. A. Teichmann. DBD: a transcription factor prediction database. *Nucl. Acids Res.*, 34(suppl_1):D74–81, 2006.
13. D. Marbach et al. Wisdom of crowds for robust gene network inference. *Nat Meth*, 9(8):796–804, Aug. 2012.
14. D. Marbach et al. Revealing strengths and weaknesses of methods for gene network inference. *Proc Natl Acad Sci U S A*, pages 6286–6291, 2010.
15. A. A. Margolin et al. Aracne: An algorithm for the reconstruction of gene regulatory networks in mammalian cellular context. *BMC Bioinformatics*, 7(S1), 2006.
16. R. J. Prill et al. Towards a rigorous assessment of systems biology models: The dream3 challenges. *PLoS ONE*, 5(2):e9202, 02 2010.
17. A. Sîrbu et al. Integrating heterogeneous gene expression data for gene regulatory network modelling. *Theory in Biosciences*, 131(2):95–102, 2012.
18. T. Soh and K. Inoue. Identifying necessary reactions in metabolic pathways by minimal model generation. In *ECAI*, pages 277–282, IOS Press, 2010.
19. S. Videla et al. Revisiting the training of logic models of protein signaling networks with asp. In *CMSB*, pages 342–361, 2012.
20. X. Zhou et al. *Genomic Networks: Statistical Inference from Microarray Data.* Wiley, 2006.