Toward Automated Selection of Parts for Genetic Regulatory Networks

Fusun Yaman[1], Swapnil Bhatia[2], Aaron Adler[1],
Douglas Densmore[2], Jacob Beal[1], Ron Weiss[3], and Noah Davidsohn[3]
[1]BBN Technologies 10 Moulton Street Cambridge, MA, USA 02138
[2]Boston University 8 Saint Mary's St. Boston, MA, USA 02215
[3]MIT 77 Massachusetts Ave Cambridge, MA, USA 02139
{fusun, aadler, jakebeal}@bbn.com, {swapnilb, dougd}@bu.edu,
{rweiss,ndavidso}@mit.edu

1. INTRODUCTION

The state-of-the-art techniques in synthetic biology require practitioners to design organisms at the DNA level. This low-level manual process becomes unmanageable as the size of a design grows. In electronic computing, high-level languages and compilers have enabled computer scientists to produce more sophisticated programs more quickly and with less effort. The same principles can be applied to synthetic biology, making the design of large and complex systems tractable.

In this paper, we define the problem of going from high-level descriptions of behavior to DNA sequences, and develop an automated solution using constraint satisfaction and optimization algorithms. Our research builds on the BioCompiler [1], which compiles an organism-level behavioral description into a network of abstract biological parts. This paper focuses on transforming such an abstract network into a concrete network realized with specific DNA sequences.

2. BACKGROUND

There are several natural mechanisms that can be manipulated in a cell to achieve a desired behavior. Our work focuses on transcriptional logic systems, where the computation is through the execution of a transcriptional network. The steps of this execution are: 1) *transcription*—the copying of a region of DNA to RNA—a process that can be regulated by protein-promoter interaction; 2) *translation*—the linking together of amino acids in the order specified in the RNA sequence into a protein; and 3) *regulation*—the suppression or activation of DNA regions by the protein produced.

In the Clotho [2] ontology, a *feature* is a DNA sequence responsible for a specific biochemical behavior. We consider transcriptional networks comprising two types of features: promoters and sequences coding for regulating proteins. The relationship between a regulating protein and the promoter preceding the DNA region containing a gene determines if and when that gene can be transcribed and then translated. A regulating protein can repress or activate a promoter. Repressors disable the ability of a promoter to initiate transcription; activators enhance its ability to initiate transcription. Consequently, transcription of a gene downstream of a promoter is controlled by activators and repressors of the promoter. Moreover, a promoter may be regulated by multiple proteins, thus implementing relationships analogous to boolean logic operations like NOT, AND, and IMPLIES.

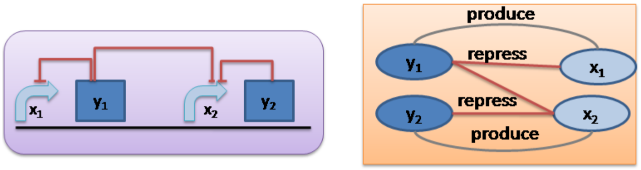

Figure 1: GRN visualizations: (Left) DNA sequence representation with rectangles as protein coding sequences (PCS) and block arrows as promoters. Red lines from PCS to promoters indicate repression. (Right) The same GRN in a graph representation with labeled edges.

A **genetic regulatory network** (GRN) is a bipartite graph with labeled edges and each vertex associated with a promoter (*i.e.*, promoter vertex) or a protein coding sequence (*i.e.*, protein vertex). In a GRN, the edges are always between a promoter vertex and a protein vertex. The edges have one of the following labels *produce, repress,* or *activate.* GRN visualizations are shown in Figure 1.

An **abstract genetic regulatory network** (AGRN) is similar to a GRN, with the difference that nodes are associated with a set of promoters or a set of protein coding sequences. An AGRN thus corresponds to a collection of GRNs. Our goal is to pick a near-optimal of these GRNs and choose a minimal set of available DNA parts covering the GRN so that it can be implemented in a cell. This translation of an AGRN to a GRN requires two kinds of solutions: a *topological solution*, choosing a single feature from the set associated with the node in the AGRN, such that all repression and activation relationships are satisfied, and a *quantitative solution*, which ensures that the choices also satisfy chemical signal compatibility constraints.

3. TOPOLOGICAL SOLUTION

The qualitative relationships between biological features are discovered by biologists experimentally. We define a *feature database* as a bipartite graph with each node associated with a single feature and the edges labeled from $\{repress, activate\}$. This is very similar to the GRN definition except the feature database does not have edges labeled

Work sponsored by DARPA I2O under contract HR0011-10-C-0168; the views and conclusions contained in this document are those of the authors and not DARPA or the U.S. Government.
IWBDA '11 San Diego, California, USA

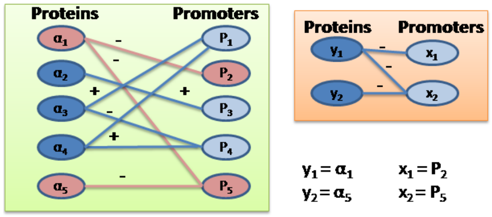

Figure 2: The constraint graph (right) is isomorphic to the strict subgraph induced by the vertices $\alpha_1, \alpha_5, P_2, P_5$ from the feature database (left).

produce. The left side of Figure 2 is a feature database (edge labels -/+ are short hand notations for *repress* and *activate* respectively), and the vertices are associated with proteins $\alpha_1, \ldots, \alpha_5$ and promoters $P_1, \ldots, P_5$.

We assume existence of a feature database containing such relationships. In translating an AGRN to a GRN by mapping each node to a feature, we need ensure that:

- The edges in the GRN are supported by the feature database. If the features we selected are not biologically capable of interacting in the desired manner, the GRN is not executable.
- The feature database does not imply additional relationships between the features of the GRN nodes. If two features are known to interact with each other, then whether or not we intended them to, they will interact when implemented in the cell, possibly disrupting the designed behavior.

For mapping the nodes of an AGRN to features we will ignore the production edges (edges labeled as *produce*) in the AGRN because they are unrelated to the two constraints above since any promoter can produce any protein. The **constraint graph**, induced by an AGRN is the same graph as the AGRN except the production edges are dropped. The right graph in Figure 2 is the constraint graph induced by the AGRN in Figure 1 where each x_i is associated with all promoters and y_i with all proteins.

In the *topological solution* of an AGRN w.r.t. a feature database, we look for a subset of vertices S in the feature database such that a *strict subgraph* of the feature database that contains only those vertices in S and all edges between them is isomorphic to the AGRN. More formally, a *topological solution* to an AGRN w.r.t. a feature database is a mapping between the nodes of the AGRN and a strict subgraph S of the feature database such that: 1) The mapping is an isomorphism between the induced constraint graph of the AGRN and the strict subgraph induced by S; 2)The labels of the edges in the constraint graph matches the edge labels in the strict subgraph induced by S; 3) For every node n in the AGRN, the set of features associated with n contains the feature associated with the node that n is mapped to. A topological solution to the AGRN in Figure 1 w.r.t. to the feature database in Figure 2 maps node y_1 to α_1, y_2 to α_5, x_1 to P_2 and x_2 to P_5.

4. QUANTITATIVE SOLUTION

Just like a digital circuit is composed of several devices, a GRN is a composition of **biological devices**. In a GRN, there is a device per promoter node. The inputs of the device are the protein nodes that are linked to the promoter with repression and activation edges. The outputs of the device are the protein nodes that are linked to the promoter with production edges. Without loss of generality, we will assume that each device has only one output. We will denote a device as $d = \langle \mathcal{I}, p, o \rangle$ where $\mathcal{I}$ is set of proteins which are inputs, p is a promoter and o is the output protein. In Figure 3, the GRN on the left has two devices: $d_1 = \langle \{W\}, P_1, Y \rangle$ and $d_2 = \langle \{Y\}, P_2, X \rangle$.

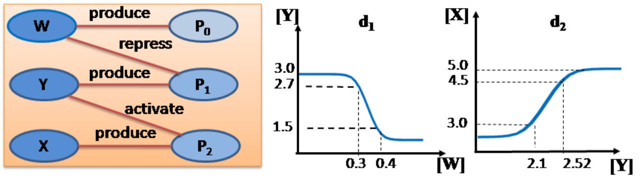

Figure 3: The concentration of Y is a function of W. The concentration of X is a function of Y. d_1 is signal compatible with d_2 because $2.7 > 2.52$ and $1.5 < 2.1$.

A device defines a function from the concentration of input proteins to concentration of output protein. The sigmoidal curves on the right side of Figure 3 are examples of such functions for devices d_1 and d_2 (single-input devices). The characteristics of the curve (slope, height, etc.) come from the biochemical properties of the features that make up the device. The slope (increasing vs. decreasing) is a function of repression or activation relationships.

Adapting from digital logic, any output o of the device higher (lower) than $high_o$ (low_o) will be considered as boolean true (false). Any output value between low_o and $high_o$ has an ill-defined truth value. Similar assumptions hold for the device inputs. In Figure 3, looking at the curve for d_1, the low value for the output Y is 1.5 and the high value is 2.7. The high and low values per input and output are the **specifications** of a device. We denote the specifications of a device $d = \langle \mathcal{I}, p, o \rangle$ as $S_d = \langle h, l \rangle$ where h (similarly l) is a function from $\mathcal{I} \cup \{o\}$ to reals for the high (similarly low) signal threshold.

Consider two devices, d with the specifications $\langle h, l \rangle$ and d' with specifications $\langle h', l' \rangle$. If the output o of d is an input of d' then d is **signal compatible** with d' iff $h(o) > h'(o)$ and $l(o) < l'(o)$. Note that if the output of first device is not an input to the second, by definition the devices are compatible. Finally, a GRN G is a **quantitative solution** to a AGRN A iff G corresponds to a topological solution of A w.r.t. a feature database and every device pair in G is signal compatible.

5. PROGRESS & RESULTS

By a reduction from subgraph isomorphism, we have shown that finding a topological solution is NP-Complete. To address this intractability, we have developed heuristic-based algorithms for topological and quantitative solutions and implemented them in a Clotho [2] app called MatchMaker.

6. REFERENCES

[1] J. Beal, T. Lu, and R. Weiss. Automatic compilation from high-level languages to genetic regulatory networks. In *Proceedings of IWBDA*, June 2010.

[2] D. Densmore, A. V. Devender, M. Johnson, and N. Sritanyaratana. A platform-based design environment for synthetic biological systems. In *TAPIA '09*, pages 24–29. ACM, 2009. (www.clothocad.org).