Scaling responsively: towards a reusable, modular, automatic gene circuit design

Linh Huynh
Department of Computer Science
& UC Davis Genome Center
University of California, Davis
huynh@ucdavis.edu

Ilias Tagkopoulos
Department of Computer Science
& UC Davis Genome Center
University of California, Davis
itagkopoulos@ucdavis.edu

1. INTRODUCTION

Scalability in computer-aided gene design is a formidable challenge given the expected increase in part availability and the ever-growing complexity of synthetic circuits. This is especially true in analog synthetic circuit design, where intermediate and final protein concentrations may not be constrained to binary values ("high"/"low"). In this abstract, we present the first steps towards a hybrid framework for optimal part selection that is able to cope with these challenges. First, we use a modular approach, where the initial circuit is divided in a set of modules, sub-circuits that are already present in the database or can be solved efficiently with exact optimization methods. Then the initial circuit is transformed to an equivalent topology that allows us to employ graph-theoretical methods to approximate the objective function. Complexity analysis shows the promise of this method to push forward the boundaries of biosystems design automation.

2. METHODS

Problem formulation: Given a circuit topology, a mutant promoter library, a set of user-defined constraints and objective function, find the optimal set of promoters so that the circuit behavior best approximates the user-defined dynamics (i.e. the objective function is minimized, subject to the constraints). In our previous work [1][2][3] we have solved this problem by using heuristics and piecewise linear optimization methods, here we provide a general framework that allows higher scalability and faster circuit construction, at the expense of lower accuracy to the intermediate protein concentrations (Figure 1).

Gene circuit representation: The nodes of a synthetic circuit, represented as a directed graph $G = (V, E)$, can be categorized into four mutually exclusive subsets: the ligand set V_L, the gene set V_G, the protein set V_P and the ligand-protein set V_B. The ligand set V_L contains inducers and other small molecules that are used as chemical exogenous circuit control. The gene set V_G contains all genes in the circuit, with each gene g in V_G consisting of it's promoter p_g and it's coding region c_g. The protein set V_P contains all proteins produced by the expression of the genes in V_G. Note that by using this formulation we need $m+n$ edges, instead of mn edges, to represent the interaction between m genes that encode for the same protein and n targets. Finally, nodes in the ligand-protein set V_B represent ligand-bound protein products. Edges e in E may represent activation or inhibition, labeled as *activatory* or *inhibitory* respectively. In addition, each edge captures a biological function, such as protein production, ligand binding, or gene regulation.

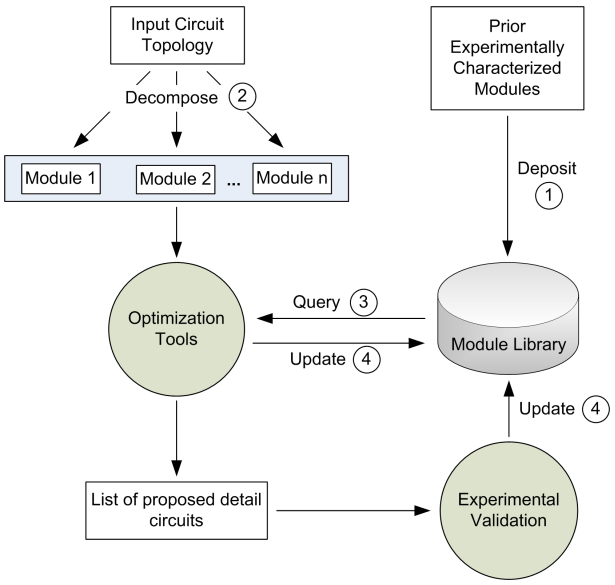

Figure 1: Overview of the proposed design automation framework

Computational framework: Fig. 1 illustrates our divide-and-conquer approach. First, we build a library that contains already constructed modules that have been experimentally characterized. We then decompose the circuit into small modules by partitioning the corresponding graph so that the number of links linking the modules is minimized. Subsequently, we quantize to discrete levels the concentration of proteins that "link" one module to another. This further reduces the dimensionality of our problem, while allowing the user to select the desired resolution for the representation of the "linkage" protein levels. The result-

ing modules are independently constructed and deposited in the database. The following paragraphs summarize the workflow of the proposed method.

1. Circuit transformation: The initial circuit is transformed to one of equivalent topology, by introducing intermediate product nodes and superimposing the effects of nodes that have the same end-product (Fig. 2). This transformation allows us to efficiently partition and perform further analysis on the graph.

2. Circuit decomposition: First, we use graph matching algorithms [4] to query the circuit for modules that currently exist in the database. All the nodes of sub-graphs that match to an existing module, will be concatenated to a single node, as the corresponding module will be used for that circuitry part. This will continue until all modules/subgraphs have been considered. Multi-level graph partitioning is then applied to the resulting graph [5] to partition this graph into equal size modules that minimize the total weight of cut-edges. If module size is constraint but can vary, then fast minimum cut (MINCUT) algorithms can be used recursively for partitioning the graph [6].

3. Library organization and query: The library/database will consist of circuit modules that have been experimentally constructed and/or computationally optimized. For experimentally constructed modules, the characterization data (steady state output protein concentrations, given the inputs) will be used. For computationally optimized modules, the information on the set of parts that best approximate the desired steady-state behavior will be returned.

4. Circuit optimization: After graph partitioning and library-based module matching, mixed-integer non-linear programming (MINLP) can be used to optimize the individual sub-graphs that do not have a library match. If f_i denotes the expression level of protein i, n is the total number of proteins in the module, and *Conditions* is the set of user-defined conditions, then the problem of finding the optimal set of parts that minimizes the difference between the desired and actual output concentration [1] is as follows:
Minimize

$$error \ = \sum_{C \in Conditions} (f_p(C) - f_p^*(C))^2 \qquad (1)$$

Subject to

$$\frac{df_i}{dt} = 0 \ \ \forall i = 1..n \qquad (2)$$

where $f_p(C)$ and $f_p^*(C)$ are the estimated and the desired concentration of protein p at condition C respectively, given a specific set of parts. The total error (i.e. the difference between the actual and the desired circuit output) will be the sum of individual module approximation errors, for all modules. The top ranked candidate circuit can be deposited in the library to be used for future designs.

3. DISCUSSION

We present a conceptual framework that uses a partitioning and optimization scheme to achieve design automation for high number of components. To compare the complexity of the proposed framework to exhaustive search, suppose that we have n genes and k promoter mutants to select from, for every gene. With exhaustive search, we need to search all k^n possible combinations. In our approach, if we partition the circuit into d modules and each module has 2θ "linkage" edges on average, each represented by l expression levels, we need at most $O(n^4 logn)$ to partition the circuit graph. In addition, searching for all possible combinations of linkage protein concentrations yields a $O(l^{\theta d} dk^{n/d})$ complexity. Therefore, the totally computational complexity in the absence of any reusable module in the library is $O(n^4 \log n) + O(l^{\theta d} dk^{n/d})$, which is less than the one of the exhaustive search approach when $n \log k > d(\theta d \log l + \log d)/(d-1)$. The speed up will greatly increase with library expansion (i.e. higher k) or circuit complexity (i.e. higher n). The downside of the proposed method is that this is achieved at the expense of global optimality guarantee, since we have to impose discrete concentration levels for the linkage edges. Still, since we perform global optimization at the module level and propose a scheme to reuse past modules for future designs, this approach has the potential to be used through automatic circuit design of very large number of components. The framework that is presented here can be integrated with multi-scale modeling and simulation efforts [7][8][9] to guide design of biological constructs for different biotechnological applications [10][11].

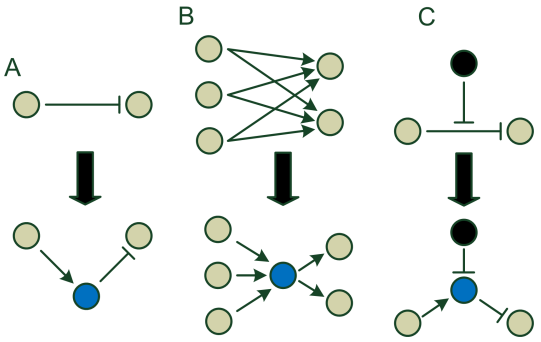

Figure 2: Graph transformation. Grey nodes represent genes (part of the gene set V_G), blue nodes represent proteins (part of the protein set V_P), and black nodes represent ligands (part of the ligand set V_L). (A) Protein-DNA interaction, (B) Protein-DNA interaction in a multiple gene copy, multiple target scenario, where the more than one copy of a specific gene exists, all contributing to the same protein product. C) Inducer-Protein interaction, where only the active form of the protein is shown.

4. REFERENCES

[1] L. Huynh, J. Kececioglu, M. Köppe, and I. Tagkopoulos, "Automatic design of synthetic gene circuits through mixed integer non-linear programming," *PLoS ONE, in press, doi:10.1371/journal.pone.0035529*, 2012.

[2] L. Huynh, J. Kececioglu, and I. Tagkopoulos, "Automated design of synthetic gene circuits through linear approximation and mixed integer optimization,"

[3] L. Hunyh and I. Tagkopoulos, "A robust, library-based, optimization-driven method for

automatic gene circuit design," in *Computational Advances in Bio and Medical Sciences (ICCABS), 2012 IEEE 2nd International Conference on*, pp. 1–6, IEEE, 2012.

[4] F. V and P. L, "Biological network querying techniques: analysis and comparison," *J. Comput. Biol.*, vol. 18, pp. 595–625, 2011.

[5] G. Karypis and V. Kumar, "A fast and high quality multilevel scheme for partitioning irregular graphs," *SIAM J. Sci. Comput.*, vol. 20, no. 1, pp. 359–392, 1998.

[6] J. Hao and J. B. Orlin, "A faster algorithm for finding the minimum cut in a directed graph," *Journal of Algorithms*, vol. 17, pp. 424–446, 1994.

[7] V. Mozhayskiy and I. Tagkopoulos, "Guided evolution of in silico microbial populations in complex environments accelerates evolutionary rates through a step-wise adaptation," *BMC bioinformatics*, vol. 13, no. Suppl 10, p. S10, 2012.

[8] V. Mozhayskiy and I. Tagkopoulos, "Horizontal gene transfer dynamics and distribution of fitness effects during microbial in silico evolution," *BMC bioinformatics*, vol. 13, no. Suppl 10, p. S13, 2012.

[9] I. Tagkopoulos, Y.-C. Liu, and S. Tavazoie, "Predictive behavior within microbial genetic networks," *science*, vol. 320, no. 5881, pp. 1313–1317, 2008.

[10] M. Dragosits, D. Nicklas, and I. Tagkopoulos, "A synthetic biology approach to self-regulatory recombinant protein production in escherichia coli," *J Biol Eng*, vol. 6, no. 2, 2012.

[11] A. S. Khalil and J. J. Collins, "Synthetic biology: applications come of age," *Nature Reviews Genetics*, vol. 11, no. 5, pp. 367–379, 2010.