IWBDA 2012

June 3-4th, Moscone Center, San Francisco, CA

49 DESIGN AUTOMATION CONFERENCE

Platinum Sponsor

SynBERC
Synthetic Biology Engineering Research Center

Gold Sponsor

ACS

Agilent Technologies

Autodesk

DNA 2.0

Hudson
ROBOTICS, INC.

life
technologies™

TECAN.

Raytheon
BBN Technologies

The following students were provided financial support by our sponsors to attend the workshop

Evan Appleton	Boston University
Swapnil Bhatia	Boston University
Federico Brunello	UC San Diego / University of Bologna, Italy
Kai-Yuan Chen	Cornell
Wilbert Copeland	University of Washington
Rishi Ganguly	Boston University
Traci Haddock	Boston University
Haiyao (Cassie) Huang	Boston University
Linh Huynh	University of California, Davis
Taek Kang	University of Texas at Dallas
Natasa Miskov-Zivanov	University of Pittsburgh
Ernst Oberortner	Boston University
Balaji Raman	Verimag Laboratory
Nicholas Roehner	University of Utah
Sean Sleight	University of Washington
Jason Stevens	University of Utah
Renato Umeton	Sapienza University of Rome

Foreword

Welcome to the Fourth International Workshop on Bio-Design Automation (IWBDA) at DAC.

IWBDA 2012 brings together researchers from the synthetic biology, systems biology, and design automation communities. The focus is on concepts, methodologies and software tools for the computational analysis and experimental development of biological systems and the synthesis of biological systems.

Still in its early stages, the field of synthetic biology has been driven by experimental expertise; much of its success can be attributed to the skill of the researchers in specific domains of biology. There has been a concerted effort to assemble repositories of standardized components. However, creating and integrating synthetic components remains an ad hoc process. The field has now reached a stage where it calls for computer-aided design tools. The electronic design automation (EDA) community has unique expertise to contribute to this endeavor. This workshop offers a forum for cross-disciplinary discussion, with the aim of seeding collaboration between the research communities.

This year, the program consists of 20 contributed talks and 10 poster presentations. Talks are organized into six sessions: CAD Tools for Synthetic Biology, Engineering, Parts, and Standardization, Characterization and System Identification, BioSimulators, Biological Circuit Design and Assembly I, and Biological Circuit Design and Assembly II. In addition, we are very pleased to have three distinguished invited speakers: William Shih, Milan Stojanovic, and Jasmin Fisher. Finally, we have an industrial panel session, entitled “What are the Missing Pieces in BioDesign Automation?”

We thank all the participants for contributing to IWBDA; we thank the Program Committee for reviewing abstracts; and we thank everyone on the Executive Committee for their time and dedication. Finally, we thank National Science Foundation, Synthetic Biology Engineering Research Center, American Chemical Society, Agilent Technologies, Autodesk, Raytheon BBN Technologies, DNA 2.0, Hudson Robotics, Life Technologies, and Tecan for their support.

Organizing Committee

Executive Committee

General Chair	Natasa Miskov-Zivanov (University of Pittsburgh)
General Secretary	Laura Adam (Virginia Tech)
Program Committee Chairs	Leonidas Bleris (UTDallas), Deepak Chandran (University of Washington) and Xiling Shen (Cornell)
Publication Chair	Chris Myers (University of Utah)
Industry Liaison Chair	Jonathan Babb (MIT)
Finance Chairs	Aaron Adler and Fusun Yaman (BBN Technologies)
DAC Liaison	Smita Krishnaswamy (Columbia)

Steering Committee

Douglas Densmore (Boston University)
Soha Hassoun (Tufts University)
Marc Riedel (University of Minnesota)
Ron Weiss (MIT)

Program Committee

J. Christopher Anderson (UC Berkeley)
Adam Arkin (UC Berkeley)
Jonathan Babb (MIT)
Jacob Beal (BBN Technologies)
Leonidas Bleris (UT Dallas)
Kevin Clancy (Life Technologies)
Douglas Densmore (Boston University)
Drew Endy (Stanford University)
Abishek Garg (Harvard University)
Soha Hassoun (Tufts University)
Mark Horowitz (Stanford University)
Alfonso Jaramillo (École Polytechnique)
Yannis Kaznessis (University of Minnesota)
Eric Klavins (University of Washington)
Tanja Kortemme (UCSF)
Smita Krishnaswamy (Columbia)
Natasa Miskov-Zivanov (University of Pittsburgh)
Kartik Mohanram (Rice)
Chris Myers (University of Utah)
Andrew Phillips (Microsoft Research)
Marc Riedel (University of Minnesota)
Herbert Sauro (University of Washington)
Xiling Shen (Cornell)
Ilias Tagkopoulos (UC Davis)
David Thorsley (University of Washington)
Christopher Voigt (UCSF)
Ron Weiss (MIT)
Erik Winfree (Caltech)
Chris Winstead (Utah State University)

IWBDA 2012 Program

Sunday – June 3rd

9:00am – 9:15am: Opening Remarks: Natasa Miskov-Zivanov (General Chair)

9:15am – 10:15am: Invited Talk: **William Shih, Harvard**

Title: “Self-Assembly of DNA into Nanoscale Three-Dimensional Shapes”

10:15am – 10:30am: Coffee Break

10:30am – 12:00pm: Tech. Talks Session 1 - *Topic: CAD Tools for Synthetic Biology*

1BDA.1 **Eugene's Enriched Set of Features to Design Synthetic Biological Devices**
Haiyao Huang, Ernst Oberortner, Douglas Densmore and Allan Kuchinsky.

1BDA.2 **Results from TASBE**
Jacob Beal, Ron Weiss, Douglas Densmore, Aaron Adler, Evan Appleton, Jonathan Babb, Swapnil Bhatia, Noah Davidsohn, Traci Haddock, Joseph Loyall, Richard Schantz, Viktor Vasilev and Fusun Yaman.

1BDA.3 **Pathway Synthesis Using the Act Ontology**
Saurabh Srivastava, Jonathan Kotker, Stephi Hamilton, Paul Ruan, Jeff Tsui, J. Christopher Anderson, Rastislav Bodik, and Sanjit A. Seshia.

1BDA.4 **metaDesign: Bacterial Strain Design Automation Software**
Jole Costanza, Giovanni Carapezza, Claudio Angione, Pietro Liò and Giuseppe Nicosia.

12:00pm – 1:45pm: Lunch and Poster Session

1:45pm – 2:45pm: Tech. Talks Session 2 - *Topic: Engineering, Parts, and Standardization*

2BDA.1 **Gene Variant Library Design for High Throughput Experimentation**
Daniel Ryan and Dimitris Papamichail.

2BDA.2 **Design, Implementation and Practice of JBEI-ICE: An Open Source Biological Part Registry Platform**
Timothy Ham, Zinovii Dmytriv, Hector Plaha, Joanna Chen, Nathan Hillson and Jay Keasling.

2BDA.3 **Standardizing Promoter Activity Through Quantitative Measurement of Transcriptional Dynamics**
Wilbert Copeland and Herbert Sauro.

2:45pm – 3:00pm: Coffee Break

3:00pm – 4:00pm: Invited Talk**: Milan Stojanovic, Columbia**

Title: “Molecular Computing: From Games to Practical Applications”

4:00pm – 5:00pm: Tech. Talks Session 3 - *Topic: Characterization and System Identification*
3BDA.1 **Validation of Network Reverse Engineering Using a Benchmark Synthetic Gene Circuit** Taek Kang, Jacob White, Eduardo Sontag and Leonidas Bleris.
3BDA.2 **Model Checking for Studying Timing of Events in T cell Differentiation** Paolo Zuliani, Natasa Miskov-Zivanov, Penelope Morel, James R. Faeder, and Edmund M. Clarke.
3BDA.3 **Network-Based Genome Design and Engineering with Direct Logical-to-Physical Compilation** Chih-Hsien Yang, Jesse Wu, Chi Yang, Tao-Hsuan Chang and Chuan-Hsiung Chang.

6:30pm - 9:30pm: IWDBA Dinner (RSVP required)

Kuleto's Italian Restaurant
221 Powell Street, San Francisco, CA 94102
Phone: 415-397-7720

Monday – June 4th

10:30am – 12:00pm: Tech. Talks Session 4 - *Topic: BioSimulators*
4BDA.1 **Dynamic Modeling of Cellular Populations within iBioSim** Jason Stevens and Chris Myers.
4BDA.2 **A Multi-Scale Model of Stem Cell Niche Formation Inside Intestine Crypts** Kai-Yuan Chen, Amit Lakhanpal, Pengcheng Bu, Steven Lipkin, Michael Elowitz and Xiling Shen.
4BDA.3 **Can Probabilistic Model Checking Explore Ribo-Nucleic Acid Folding Space?** Stefan Janssen, Loic Pauleve, Yann Ponty, Balaji Raman and Matthias Zytnicki.
4BDA.4 **A Biomolecular Implementation of Systems Described by Linear and Nonliner ODE's** Vishwesh Kulkarni, Hua Jiang, Theerachai Chanyaswad and Marc Riedel.

12:00pm – 1:45pm: Lunch and Poster Session

1:45pm – 2:45pm: Invited Talk: **Jasmin Fisher, Microsoft UK**
Title: "From Coding the Genome to Algorithms Decoding Life"

2:45pm – 3:45pm: Tech. Talks Session 5 - *Topic: Biological Circuit Design and Assembly I*
5BDA.1 ***In Silico* Design of Functional DNA Constructs Based on Heuristic Data** Claes Gustafsson, Alan Villalobos, Mark Welch and Jeremy Minshull.

5BDA.2 **j5 and DeviceEditor: DNA Assembly Design Automation**
Joanna Chen, Rafael Rosengarten, Douglas Densmore, Timothy Ham, Jay Keasling and Nathan Hillson.

5BDA.3 **Automatic Design of RNA and Transcriptional Circuits in *E. coli***
Guillermo Rodrigo, Thomas Landrain, Boris Kirov, Raissa Estrela, Javier Carrera and Alfonso Jaramillo.

3:45pm – 4:00pm: Coffee Break

4:15pm – 5:15pm: Tech. Talks Session 6 - *Topic: Biological Circuit Design and Assembly II*

5BDA.4 **Integrating Synthetic Gene Assembly and Site-Specific Recombination Cloning**
Bianca J Lam, Federico Katzen, Kevin Clancy, Xiangdong Liu, Nian Liu, Gengxin Chen, Kimberly Wong, Todd Peterson, Antje Pörtner-Taliana.

5BDA.5 **Scaling Responsively: Towards a Reusable, Modular, Automatic Gene Circuit Design**
Linh Huynh and Ilias Tagkopoulos.

5BDA.6 **Chance-Constraint Optimization for Gene Modifications**
Mona Yousofshahi, Michael Orshansky, Kyongbum Lee and Soha Hassoun.

5:15pm - 6:15pm: Industrial Panel Session

6:15pm - 6:30pm: Closing Remarks and Post-Workshop Future Planning

Abstracts - Table of Contents

Sunday – June 3rd

Invited Talk: **William Shih, Harvard**

"Self-Assembly of DNA into Nanoscale Three-Dimensional Shapes"

I will present a general method for solving a key challenge for nanotechnology: programmable self-assembly of complex, three-dimensional nanostructures. Previously, scaffolded DNA origami had been used to build arbitrary flat shapes 100 nm in diameter and almost twice the mass of a ribosome. We have succeeded in building custom three-dimensional structures that can be conceived as stacks of nearly flat layers of DNA. Successful extension from two-dimensions to three-dimensions in this way depended critically on calibration of folding conditions. We also have explored how targeted insertions and deletions of base pairs can cause our DNA bundles to develop twist of either handedness or to curve. The degree of curvature could be quantitatively controlled, and a radius of curvature as tight as 6 nanometers was achieved. This general capability for building complex, three-dimensional nanostructures will pave the way for the manufacture of sophisticated devices bearing features on the nanometer scale.

William Shih is an Associate Professor in the Department of Biological Chemistry and Molecular Pharmacology at Harvard Medical School and the Department of Cancer Biology at the Dana-Farber Cancer Institute and a Core Faculty member at the Wyss Institute for Biologically Inspired Engineering at Harvard. William studied Biochemical Sciences at Harvard for his A.B. (1990–1994) and Biochemistry at Stanford for his Ph.D. (1994–2000) He did a postdoctoral fellowship at The Scripps Research Institute (2001–2004) and has since been back at Harvard as a faculty member.

Sunday – June 3rd
Tech. Talks Session 1 - *Topic: CAD Tools for Synthetic Biology*
1BDA.1 **Eugene's Enriched Set of Features to Design Synthetic Biological Devices** Haiyao Huang, Ernst Oberortner, Douglas Densmore and Allan Kuchinsky.
1BDA.2 **Results from TASBE** Jacob Beal, Ron Weiss, Douglas Densmore, Aaron Adler, Evan Appleton, Jonathan Babb, Swapnil Bhatia, Noah Davidsohn, Traci Haddock, Joseph Loyall, Richard Schantz, Viktor Vasilev and Fusun Yaman.
1BDA.3 **Pathway Synthesis Using the Act Ontology** Saurabh Srivastava, Jonathan Kotker, Stephi Hamilton, Paul Ruan, Jeff Tsui, J. Christopher Anderson, Rastislav Bodik, and Sanjit A. Seshia.
1BDA.4 **metaDesign: Bacterial Strain Design Automation Software** Jole Costanza, Giovanni Carapezza, Claudio Angione, Pietro Liò and Giuseppe Nicosia.

Eugene's Enriched Set of Features to Design Synthetic Biological Devices

Haiyao Huang, Ernst Oberortner,
Douglas Densmore
Department of Electrical and Computer Engineering
Boston University
{huangh,ernstl,dougd}@bu.edu

Allan Kuchinsky
Agilent Technologies
allan_kuchinsky@agilent.com

ABSTRACT

Eugene is a design language to support synthetic biologist in order to construct large and complex biological devices more accurately. Compared to its original version, Eugene provides now an enriched set of functionalities to specify and constrain synthetic biological devices and their design synthesis. This work highlights (1) the declaration of devices at various abstraction levels, (2) the control-flow management of design synthesis, (3) a design space exploration to generate devices, and (4) the prototyping of functions. Eugene allows synthetic biologists to specify, design, and constrain a large number of biological devices in a few lines of code, without having to specify every single device manually.

1. INTRODUCTION

The most common view in synthetic biology is to view DNA sequences as parts with certain properties to form composite parts, devices, or systems. Design languages that support various level of abstractions can make a synthetic biologist's life easier. Whereas Eugene's initial version [2] has focused on structure and functionality we present in this paper Eugene's new set of facilities, which includes the declaration of devices at various abstraction levels, the management of the control-flow of design synthesis, the automatic generation of devices, and user-defined reusable functions. Compared to other languages in the synthetic biology domain, Eugene offers certain advantages in the areas of flexibility, simple syntax, compatibility with other design tools, and extensibility.

2. EUGENE'S NEW FEATURES

Specifying Devices at Various Levels of Abstraction

Eugene allows synthetic biologist to design abstract, instantiated, and hybrid synthetic devices.

Abstract devices are assembled of part types, such as promoters, ribosome binding sites, or terminators. Instantiated devices are either instances of abstract devices or assembled of various parts, such as *pLac* or *lacI*. If an instantiated device is an instantiates an abstract device, the device's parts are ordered as specified in the abstract device. Hybrid devices are assembled of devices, part types, and parts.

We illustrate in Listing 1 examples of defining an abstract, instantiated, and hybrid inverter. To the best of our knowledge, there exists no language to design such types of synthetic devices.

```
/* Define an abstract Inverter */
Device Abstract_Inverter(
    Promoter, RBS, Repressor, Terminator,
    Promoter, RBS, Reporter, Terminator);

/* Instantiate the abstract Inverter */
Abstract_Inverter Instantiated_Inverter(
    pBad, BBa_J61100, cI, BBa_B0015,
    pCI, RFPc, BBa_B0015);

/* Declare a hybrid Inverter */
Device Hybrid_Inverter(
    Promoter, RBS, cI, BBa_B0015,
    Promoter, RBS, RFPc, BBa_B0015);
```

Listing 1: Declaration of Synthetic Devices

Control-Flow Facilities

Similar to computer programming languages, Eugene offers to its users conditional branches — `if-else` — and loop statements — `for`, `while`, and `do-while`. Conditional branches and loops make it possible to manage the control-flow of design synthesis (see Listing 4). Also, control-flow facilities reduce the lines of redundant code and allow to apply specific constraints various times in case of certain conditions.

Automatic Design Space Exploration

Eugene offers two built-in functions to automatically generate synthesized devices — `permute` and `product`. Though the syntax of both statements is equivalent, both functions generate devices differently. The `product` function changes the assembling parts while maintaining the order of the device's components. The `product` function takes a device and all available parts in the design space, and creates all possible variations of the given device while maintaining the order of the device's components. Given a device composed of n components, and m parts in the design space, the `product` function will generate m^n variations of that device. The `product` function allows, for example, to rapidly generate all instances of an abstract device that adhere to a given set of rules. The `permute` function permutes the order of a device's components. The `permute` function collects all defined parts and creates all possible permutations of them that comply to a given device's structure. For example, the `permute` function permutes the components of an abstract device. Hence, given a device composed of n parts, the permute function will generate $n!$ permutations of the parts.

```
/* Define two Rules */
Rule r01(STARSWITH Promoter);
Rule r02(ENDSWITH Terminator);

/* PRODUCT */
product(Abstract_Inverter, strict, 100);

/* PERMUTE */
permute(Abstract_Inverter, strict);
```

Listing 2: `product` and `permute`

In Listing 2 we define two rules and exemplify the functions' utilization. Both functions can take up to three arguments and return a list of the generated devices. Only the first argument — the input device — is required whereas the second and third arguments are optional. For the second parameter, which specifies the level of rule enforcement, two options can be specified: `strict` and `flexible`. The `strict` option only generates devices that obey the specified rules, while the `flexible` option, which is default, generates every possible device and labels them if they violate a rule. The third parameter is an integer number which limits the number of the generated devices. If the Eugene user calls the `product` or `permute` with a capacity smaller than the total number of possible variations, it will generate a random subset of the that size.

Function Prototyping

Eugene offers a rich set of built-in functions that are not described in this paper due to space restrictions. However, for synthetic biologists it is important to defining their own functions and parameters. Hence, Eugene offers facilities to extend the repertoire of functions. In Listing 3 we present an example of creating a function that returns the number of promoters in a given device.

```
// function defintion
function num countPromoters(Device d) {
    num nrOfPromoters = 0;
    for(num i=0; i<d.size(); i++) {
        if(d[i] INSTANCEOF Promoter) {
           nrOfPromoters++;
        }
    }
    return nrOfPromoters;
}

// call the function
num nr = countPromoters(Abstract_Inverter);
```

Listing 3: Function Prototyping

3. AN EXAMPLE OF USING EUGENE'S NEW FEATURES

The example in Listing 4 focuses on the replacement of an inverter's promoters whose strength is lower than a given threshold. First, we use the `product` function in order to generate all instantiated inverters from the design space whose structure equals to given `Abstract_Inverter` device. Next, we iterate over all generated devices and each device's components, to check if the current component is a `Promoter`, and if its strength is lower then the threshold `T`. If so, we replace the current promoter with a new promoter from the design space by calling the defined `getPromoter()` function. In the `getPromoter` function, we iterate over all promoters returned by Eugene's `getAllPromoters()` function, and return the first promoter with a strength higher then the given threshold `T`.

```
// generate all instances of an Inverter
Device[] arrDevices = product(
    Abstract_Inverter);

// evaluate all generated Inverters
num T = 6.2;
for(num i=0; i<arrDevices.size(); i++)  {
  inverter = arrDevices[i].
  for(num k=0; k<inverter.size(); k++) {
    if(inverter[k] INSTANCEOF Promoter AND
       inverter[k].strength < T) {
        // replace the current promoter
        // with the new promoter returned
        // by the getPromoter function
        inverter[k] = getPromoter(T);
      }
    }
  }
}

// define a function
function Promoter getPromoter(num T) {
  // find a promoter in the design space
  // whose strength is higher then
  // the given threshold T
  for(Promoter prom : getAllPromoters()) {
    if(prom.strength > T) {
      return prom;
    }
  }
}
```

Listing 4: Using Eugene's New Features to Replace a Device's Promoters

4. CONCLUSION AND FUTURE WORK

In this paper, we exemplified the new features of Eugene, namely to (1) design of devices at various levels of abstraction, (2) specify the control-flow of design synthesis, and (3) to generate devices automatically, and (4) to specify user-defined functions. We are currently working on the integration with Synthetic Biology Open Language (SBOL), making it easier to exchange synthetic biological between tools. Furthermore, we are planing to release Eugene with a user-friendly IDE for the International Genetically Engineered Machine (iGEM) 2012 competition [1], allowing the iGEM teams to evaluate Eugene's features and usability. In the future, we want to provide facilities to specify families of parts and devices, enhance the specification of rules, as well as to query characterization data of parts and devices. We believe that the Eugene language is a great step towards a full support of synthetic biologists in order to design and build complex and efficient synthetic biological systems automatically.

5. REFERENCES

[1] International Genetically Engineered Machine (iGEM) Foundation. `http://igem.org`.

[2] Bilitchenko, L. *et al.* Eugene: A domain specific language for specifying and constraining synthetic biological parts, devices, and systems. *PLoS ONE 6*, 4 (2011).

Results from TASBE

Jacob Beal
BBN Technologies
10 Moulton Street
Cambridge, MA, USA 02138
jakebeal@bbn.com

Ron Weiss
Massachusetts Institute of Technology
77 Massachusetts Ave
Cambridge, MA, USA 02139
rweiss@mit.edu

Douglas Densmore
Boston University
8 Saint Mary's St.
Boston, MA, USA 02215
dougd@bu.edu

1. INTRODUCTION

The TASBE (A Tool-Chain to Accelerate Synthetic Biological Engineering) project [2] developed a tool-chain (Figure 1) to design and build synthetic biology systems. These tools convert a circuit description written in a high-level language to an implementation in cells, assembled with laboratory robots. Each tool addresses a different sub-problem. This paper describes each tool and its key results.

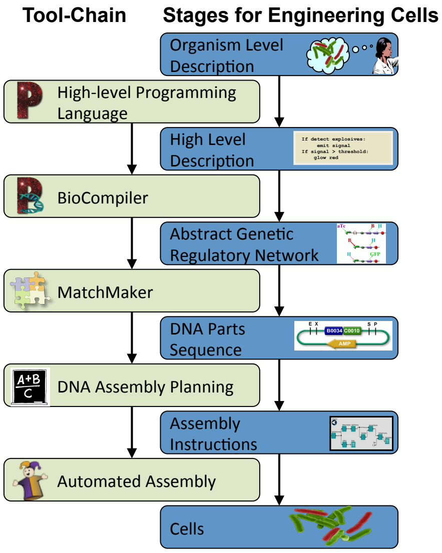


Figure 1: Engineering process and corresponding TASBE tools.

TASBE Characterization is a detailed methodology that gathers highly accurate data for synthetic biology parts. This data enable the transformations done by the tools in the tool-chain. The TASBE project gathered data for many biological parts. The **BioCompiler** begins with a design written in a high-level language. The design is compiled and optimized, producing an abstract genetic regulator network (AGRN). The resulting optimized designs are equivalent to those produced by human experts. The AGRN can be simulated to verify that the circuit produces the desired effect. **MatchMaker** converts this AGRN into an instantiated genetic regulatory network (GRN) by selecting parts from a database of parts that meet the TASBE Characterization standards. MatchMaker ensures that the parts used in the GRN are signal compatible, thus enabling composite design. Finally, **DNA Assembly Planning** and **Automated Assembly** (including robotic assembly) converts the GRN into a part sequence and assembly instructions for a robot or human. The DNA sequence can then be assembled and inserted into cells for execution.

2. TASBE CHARACTERIZATION

Our work in TASBE has shown us that, with regards to DNA part characterization, any type of compositional design will need at least: 1) Large numbers of single-cell measurements (as opposed to population average values), 2) Measurements of the level of part output signal(s) across the full dynamic range of levels of part input signal(s), 3) Data to determine the per-copy effect of the construct, and 4) The statistical distribution of single-cell output levels for each input level, in order to estimate the variability of behavior.

Prior characterization efforts, however, have generally not yielded enough high-quality information to enable compositional design. In the TASBE project we have developed a new characterization technique (both analytics and wet-lab) capable of producing such data (Figure 2). We have published a technical report [3] that describes the techniques we have developed, along with examples of their application, so that the techniques can be accurately used by others.

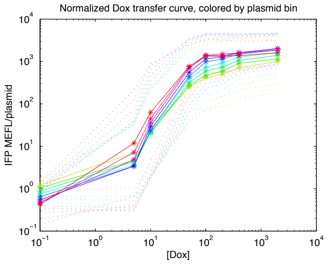


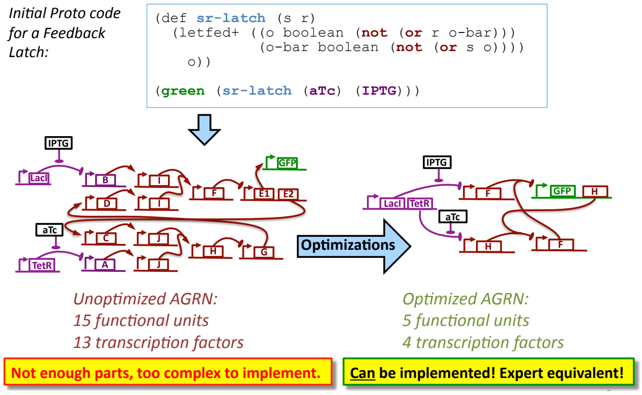
Figure 2: Transfer curve obtained for Dox induction using TASBE characterization techniques.

3. BIOCOMPILER

We defined a high-level programming language [1] for biological designs. This language is based on a spatial computing language to support modeling the multi-cellular interactions that will be necessary for synthetic biology applications. The designs specified in the high-level language are compiled to AGRNs by composing motifs and optimizations (Figure 3). BioCompiler is the first tool that allows arbitrary boolean logic and feedback systems to be specified and then designs an appropriate genetic regulatory network

Work partially 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 '12 San Fransisco, California, USA

automatically. The optimization is competitive with human experts and homologous with hand designed circuits. Additionally, the function of the circuit can be verified using an ODE simulation. Team biologists now routinely use the BioCompiler to design AGRNs because the output is less error prone and faster than hand designs.


Initial Proto code for a Feedback Latch:

```
(def sr-latch (s r)
  (letfed+ ((o boolean (not (or r o-bar)))
            (o-bar boolean (not (or s o))))
    o))

(green (sr-latch (aTc) (IPTG)))
```

Figure 3: A high-level program is compiled to an AGRN and then optimized.

4. MATCHMAKER

In order to realize an AGRN, we need to instantiate the abstract components of the network with actual biological parts. We formally defined the problem of transforming the abstract network produced by the BioCompiler into a sequence of DNA parts given the availability of the parts and the biological constraints on them. We identified three steps in this transformation (Figure 4): 1) *Feature Matching* is the problem of assigning a single feature to each node in the AGRN such that the repression/activation relationships are satisfied. Basically this converts an AGRN into a GRN. 2) *Signal Matching* is the problem of finding the best GRN with respect to the chemical signal compatibility. 3) *Part Matching* is the problem of finding the shortest part sequence that implements the GRN. We studied the theoretical complexity of these subproblems. We have implemented our algorithms in the software tool MatchMaker, which is also integrated with Clotho [4] for accessing databases.

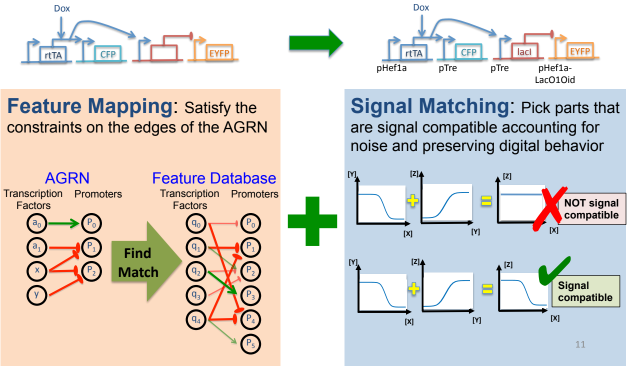


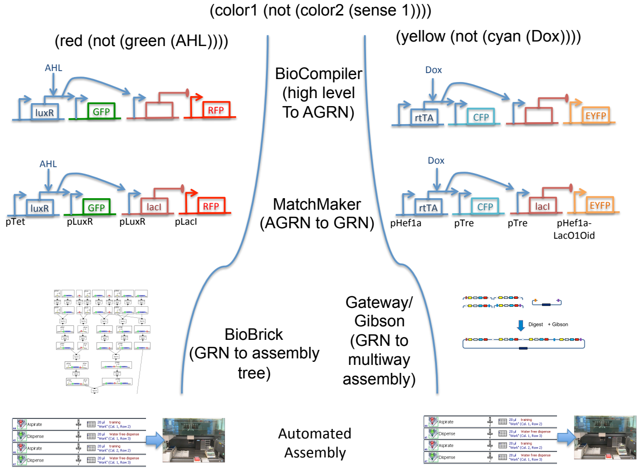
Figure 4: A visualization of the first two MatchMaker steps: feature matching and signal matching. The AGRN is transformed into a GRN.

5. DNA ASSEMBLY PLANNING AND AUTOMATED ASSEMBLY

The last stages of the tool-chain[5] plan how to assemble the part sequence and then convert the sequence into assembly instructions that can be executed on a laboratory robot. Versions of these tools, customized for specific laboratory hardware and cellular platforms, are running at the MIT and BU labs. These tools take into account resource allocation and integrate design and data management tools with a language for protocol specification and robotic execution.

6. RESULTS

A synthetic biology tool-chain can bring the ideas of programmability, abstraction, and languages to synthetic biology. The goal of this project was to validate the viability of the tool-chain approach. We have implemented a working proof-of-concept implementation of the TASBE infrastructure: decomposing the problem has made the development process more tractable, results are rapidly usable by other components (progress on characterization can be exploited by MatchMaker), and we have been able to bring programming language, artificial intelligence, CAD, and biology expertise to bear on the problem despite no individual member of the team being an expert in all fields. Three key results provide evidence that TASBE is a unique, novel and viable architecture: 1) High-level programs have compiled to designs equivalent to hand-designed systems of DNA parts that are operating correctly *in vivo*, 2) Characterization of transcriptional logic parts has shown acceptable amplification to support digital abstractions and tractable part matching, and 3) The upper portions of TASBE are completely modular with respect to the choice of assembly target between BioBrick-protocol parts for *E. coli* and new-protocol parts for mammalian cells (Figure 5).



(color1 (not (color2 (sense 1))))
(red (not (green (AHL))))
(yellow (not (cyan (Dox))))
BioCompiler (high level To AGRN)
AHL
luxR
GFP
RFP
Dox
rtTA
CFP
EYFP
MatchMaker (AGRN to GRN)
AHL
luxR
GFP
lacI
RFP
pTet
pLuxR
pLuxR
pLacI
Dox
rtTA
CFP
lacI
EYFP
pHef1a
pTre
pTre
pHef1a-LacO1Oid
BioBrick (GRN to assembly tree)
Gateway/ Gibson (GRN to multiway assembly)
Digest + Gibson
Automated Assembly

Figure 5: The same high-level program can be compiled to a platform specific (left: *E. coli*; right: mammalian) program using TASBE.

We plan to make these tools available either as Clotho Apps or in the case of BioCompiler and TASBE Characterization through a web service interface. Finally the TASBE project provided a foundation for DARPA efforts such as the Living Foundries program.

7. ADDITIONAL AUTHORS

Aaron Adler (BBN Technologies, email: aadler@bbn.com), Evan Appleton (Boston University, email: eapple@bu.edu), Jonathan Babb (MIT, email: jbabb@mit.edu), Swapnil Bhatia (Boston University, email: swapnilb@bu.edu), Noah Davidsohn (MIT, email: ndavidso@mit.edu), Traci Haddock (Boston University, email: thaddock@bu.edu), Joseph Loyall (BBN Technologies, email: jloyall@bbn.com), Richard Schantz (BBN Technologies, email: rschantz@bbn.com) Viktor Vasilev (Boston University, email: vvasilev@bu.edu), and Fusun Yaman (BBN Technologies, email: fusun@bbn.com).

8. REFERENCES

[1] J. Beal, T. Lu, and R. Weiss. Automatic compilation from high-level biologically-oriented programming language to genetic regulatory networks. *PLoS ONE*, 6(8):e22490, August 2011.

[2] J. Beal, R. Weiss, D. Densmore, A. Adler, J. Babb, S. Bhatia, N. Davidsohn, T. Haddock, F. Yaman, R. Schantz, and J. Loyall. TASBE: A tool-chain to accelerate synthetic biological engineering. In *Proceedings of the 3rd International Workshop on Bio-Design Automation*, pages 19–21, June 2011.

[3] J. Beal, R. Weiss, F. Yaman, N. Davidsohn, and A. Adler. A method for fast, high-precision characterization of synthetic biology devices. Technical Report MIT-CSAIL-TR-2012-008, MIT, April 2012. http://hdl.handle.net/1721.1/69973.

[4] 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, April 2009.

[5] V. Vasilev, C. Liu, T. Haddock, S. Bhatia, A. Adler, F. Yaman, J. Beal, J. Babb, R. Weiss, and D. Densmore. A software stack for specification and robotic execution of protocols for synthetic biological engineering. In *Proceedings of the 3rd International Workshop on Bio-Design Automation*, pages 24–25, June 2011.

Pathway Synthesis using the Act Ontology

Saurabh Srivastava[*]
CS, UC Berkeley
saurabhs@cs.berkeley.edu

Jonathan Kotker
EECS, UC Berkeley
jkotker@eecs.berkeley.edu

Stephi Hamilton
Bioengineering, UC Berkeley
stephi@berkeley.edu

Paul Ruan
CS, UC Berkeley
paul.ruan@gmail.com

Jeff Tsui
CS, UC Berkeley
tsui.jeff@gmail.com

J. Christopher Anderson
Bioengineering, UC Berkeley
jcanderson@berkeley.edu

Rastislav Bodik
CS, UC Berkeley
bodik@cs.berkeley.edu

Sanjit A. Seshia
EECS, UC Berkeley
sseshia@eecs.berkeley.edu

ABSTRACT

We describe here the Act Ontology, a formalism for uniformly describing biochemical function, and its use in building an enzymatic pathway synthesizer. A formal description of biochemical function allows us to reason about it, and for the particular case of enzymes, this function allows us to build a synthesizer tool that given a target chemical can automatically infer the most likely pathway that leads to it. The pathway can include known as well as hypothetical enzymes with predicted function.

Categories and Subject Descriptors

J.3 [**Computer Applications**]: Life and Medical Sciences—*Biology and genetics*

General Terms

Algorithms, Design, Standardization

Keywords

Biochemical formalism, enzymatic metabolic paths

1. INTRODUCTION

Synthetic biology is at the verge of a virtual overflow in sequence characterization and their availability for use in rationally designed genetic function. These designed genetic constructs, when inserted in a chassis such as *E. coli* (a bacteria) or *S. cerevisiae* (a yeast), impart desired function to the organism. While there will be a definite overabundance in the characterization of genetic material, there are as yet no formalisms in place to uniformly capture all that functional information to be used by computational tools.

Without a formal way of encapsulating that information the characterization data will remain computer-inaccessible and only available in human-readable tables and data sheets. Thus, design methodology will remain outside the purview of computational tools.

Towards remedying this situation, we propose the Act Ontology, which is a formal, uniform, and expressive mechanism for encoding biochemical function. We are developing the theoretical framework for specifying function, as well as a repository based on that formalism that will store biomolecular function.

Our current focus is on encoding enzyme function so that we can build a tool for automatically suggesting novel metabolic pathways to unnatural chemicals. The pathway synthesizer tool constructs pathways not only based on naturally-known reactions (for which there are many pre-existing tools), but also reactions that are inferred as plausible based on reaction operators. These operators are derived from abstracting from natural reactions and form an abstraction hierarchy that we intelligently traverse to derive pathways that have a high likelihood of success.

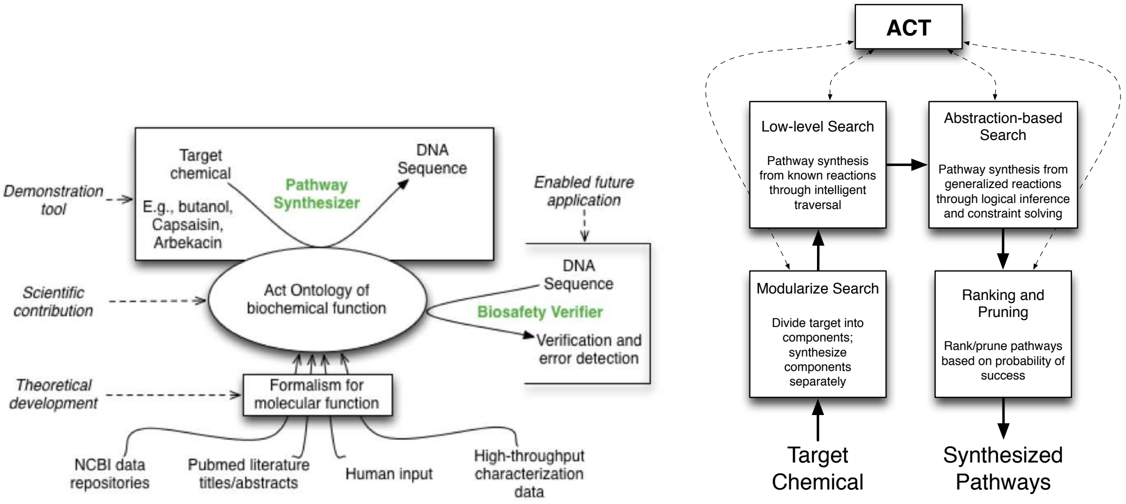
2. ACT ONTOLOGY

Act is a formalism for describing the molecular function of species. A specie is any entity that participates in a biochemical reaction. A genetic feature is a specific specie that encodes for and functions either a protein, a RNA, or in its DNA form itself. The central concept of Act is that of a family, akin to the Gene Ontology (GO [1]) concept of a family. In contrast to GO, however, the defining feature of Act families is not their location in the hierarchy of families, but the functional traits corresponding to a specie. In fact, in Act we do not even pre-specify the hierarchy, which can be inferred from containment of functional traits. As such, Act does not simply label families, but rather it provides a formal description of the species' chemical behavior according to a controlled vocabulary to support querying, synthesis, and verification.

Features are some of the most important species Act ascribes function to. Features are the DNA elements that

[*]Corresponding author.

This work was supported by the the DARPA Living Foundries initiative. SS is supported by NSF grant 1019343 to the Computing Research Association for the CIFellows Project. Additional support has been provided by the NSF Synthetic Biology Engineering Research Center (SynBERC).


Target chemical
DNA Sequence
Pathway Synthesizer
Demonstration tool
E.g., butanol, Capsaisin, Arbekacin
Enabled future application
Act Ontology of biochemical function
DNA Sequence
Scientific contribution
Biosafety Verifier
Verification and error detection
Theoretical development
Formalism for molecular function
NCBI data repositories
Pubmed literature titles/abstracts
Human input
High-throughput characterization data
ACT
Low-level Search
Pathway synthesis from known reactions through intelligent traversal
Abstraction-based Search
Pathway synthesis from generalized reactions through logical inference and constraint solving
Modularize Search
Divide target into components; synthesize components separately
Ranking and Pruning
Rank/prune pathways based on probability of success
Target Chemical
Synthesized Pathways

(a) (b)

Figure 1: (a) Populating and using Act. (b) The architecture of the synthesizer.

directly encode a particular molecule, including functional DNAs, RNAs, and proteins. While a protein, RNA, DNA molecules may all come from the same sequence through translation, and transcription respectively, we distinguish them as different features because they have different functional characteristics in their various forms.

Every Act species is assigned a family, possibly more than one. An Act family encapsulates a unit of functional characteristic, e.g., the enzyme activity of a protein. The formal representation of a family is in terms of a finite state transducer, specifically a Mealy Machine [3]. The states correspond to states of existence of the species, e.g., native form, or bound to a small molecule etc. The input and output alphabet is the same and consist of the universe of species.

Formally, the Mealy machine for an Act family is the 6-tuple, $(\mathrm{S}, \mathrm{S0}, \Sigma, \wedge, \mathrm{T}, \mathrm{G})$, defined as: - A finite set of states S: the various states the molecule exists in.
- A start state S0 $(\in S)$: The native state of the molecule.
- A finite input alphabet Σ, and output alphabet $\wedge$: The input and output are species and the empty symbol ϵ.
- A transition function $(\mathrm{T} : \mathrm{S} \times \Sigma \to \mathrm{S} \times \wedge)$ mapping pairs of a state and an input symbol to the corresponding next state and output symbol.

Currently, we populate Act by data repository and literature mining, but in the future Act will also get families from high-throughput characterization data, as shown in Figure 1.

3. ACT ENABLED PATHWAY SYNTHESIS

The architecture of the synthesizer and the role of the Act biochemical database is shown in Figure 1.

The pathway synthesizer has a very strict encoding of biosafety. We prohibit the synthesizer from exploring known harmful patterns in chemicals and known harmful families.

4. RELATED WORK

The GO and SO [2] ontologies are well known previous attempts at categorizing biochemical features into a functional hierarchy. The hierarchy of organization is the main contribution of these ontologies and provides the correlation between function. On the other hand, in Act, families are not defined by any hierarchy, but instead through their internal traits. A hierarchy can be inferred if so desired by checking a family pair whether one contains all the traits of the other.

The reaction operators that form the basis of our synthesizer are related to KEGG RCLASSes and BNICE operators. While those are manually authored and curated, in Act they are inferred based on chemical, biochemical, and chemoinformatics theory, and therefore scalable even with large amounts of fine-grained high-throughput data.

5. CONCLUSIONS

We have briefly described the Act Ontology, whose aim is to formally describe and encapsulate biochemical function of biomolecules. We also presented the encoding of enzyme function within Act, and its use in designing a synthesizer tool that automatically infers plausible metabolic pathways for unnatural target chemicals. These rationally designed new pathways consisting of natural and speculated enzymes, if inserted into *E. coli* or another chassis will allow the organism to produce the target chemical starting from its primary metabolites.

6. REFERENCES

[1] M. Ashburner, C. A. Ball, J. A. Blake, and et al. Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nat. Genet.*, 25(1), May 2000.

[2] K. Eilbeck, S. E. Lewis, C. J. Mungall, M. Yandell, L. Stein, R. Durbin, and M. Ashburner. The Sequence Ontology: a tool for the unification of genome annotations. *Genome Biol.*, 6(5):R44, 2005.

[3] G. H. Mealy. A method for synthesizing sequential circuits. *Bell System Tech. Jour.*, 34(5):1045–1079, 1955.

metaDesign: Bacterial Strain Design Automation Software

Jole Costanza
Dept. of Maths & CS
University of Catania - Italy
costanza@dmi.unict.it

Giovanni Carapezza
Dept. of Maths &CS
University of Catania - Italy
carapezza@dmi.unict.it

Claudio Angione
Computer Laboratory
University of Cambridge - UK
claudio.angione@cl.cam.ac.uk

Renato Umeton
Sapienza University of Rome - Italy
renato.umeton@uniroma1.it

Pietro Lió
Computer Laboratory
University of Cambridge - UK
pietro.lio@cl.cam.ac.uk

Giuseppe Nicosia
Dept. of Maths & CS
University of Catania - Italy
nicosia@dmi.unict.it

ABSTRACT

We propose a framework for the investigation and the automated design of bacteria to allow the overproduction of metabolites of industrial interest. Our framework, which consists of three main steps, includes the implementation of a sensitivity analysis method, a multi-objective optimization algorithm, and a robustness analysis algorithm. They exploit the Pareto optimality principle to explore species, reactions, pathways, and knockout parameter space. Furthermore, they provide theoretical and practical guidelines for design automation: applications on *Escherichia coli, Geobacter sulfurreducens, Yersinia pestis, Methanosarcina barkery* reveal a good performance on a variety of biotechnological products. The toolbox performs the following three interconnected tasks: 1) the *Pathway-oriented Sensitivity Analysis*, PoSA, evaluates the *sensitive and insensitive parameters* of the models of the four studied bacteria; 2) the *Genetic Design Multi-Objective*, GDMO, determines the *Pareto fronts* (PF) for specific biological functions (e.g., *acetate, succinate, biomass*) where each non-dominated point in the PF is a *strain*, which has been obtained turning on and turning off collections of *gene-sets/reactions*; 3) finally, *global robustness*, GR, assesses the *expected yield of the strains.*

Keywords

Metabolic CAD, Metabolic engineering, Biological Computer Aided Design, Multi-objective optimization, Sensitivity Analysis, Robustness Analysis, ϵ-dominance Analysis.

1. METHODS AND DATA

In bacteria, as well as in other organisms, our framework is able to design in silico genetic strategies, each of which consists of a genetic manipulation. A manipulation is the switching off of a gene-set with the aim of optimizing a desired biological function (i.e., *acetate* (Ac) or *succinate* (Succ) production for industrial purposes). Remarkably, when a gene is switched off, both the *biomass* (Bm) of the organism and its reproduction ability are altered. Therefore, the search for the best knockout strategies must ensure the survival of the organism. For this reason, we propose a multi-objective approach that, by means of a genetic algorithm, maximizes two objective functions: the Bm and the desired metabolic product. We test our method on four Flux Balance Analysis models with the same initial conditions: *Escherichia coli* [3], *Geobacter sulfurreducens* [5], *Yersinia pestis* [1], *Methanosarcina barkery* [2]. Multi-objective optimization provides a set of Pareto-optimal points, each of which represents a genetic strategy (e.g., an *E. coli* strain) and a phenotype. We also performed an ϵ-dominance analysis [4], in order to improve the diversity and capability of the solutions (the strains). After the optimization routine is performed, all the sampled points are revisited. Then, a new set of solutions is built by applying a relaxed condition of dominance. Remarkably, this set contains both the new "ϵ-non-dominated" solutions and the previous non-dominated ones. In Figure 2 we show the results according to several ϵ values. As ϵ increases, the number of ϵ-non-dominated points increases. Additionally, we propose a novel sensitivity method, called Pathway-oriented Sensitivity Analysis (PoSA). We tested PoSA in the metabolic network of *E. coli* (2382 reactions, 913 feasible genetic manipulations (i.e., gene-sets), 36 metabolic pathways). Thanks to PoSA, we rank pathways according to their influence in the whole metabolic network, turning off in a random way genes that are involved in the metabolism of each pathway. As a post-processing step, we implement a robustness analysis method inspired by [6], to find the most robust genetic strategy.

2. RESULTS AND DISCUSSION

Figure 1 shows the Pareto fronts obtained for 4 metabolic networks, to maximize Ac and Succ productions. For *Y. pestis* we consider two Bm compositions. The significance of these two temperatures stems from the two types of hosts that *Y. pestis* infects: insect vectors at ambient temperature and mammalian hosts with regulated body temperatures of about 37°C. Pareto fronts provide significant information in metabolic design automation. The size of non-dominated solutions, the first derivative and the area under the curve are important gauges for the best design within the same organism or between different organisms. Exploratory analyses suggest that the area underlying the Pareto front provides an estimate of the number of intermediates, which may be exploited for biotechnological purposes or to build synthetic pathways. The slope of the Pareto front reflects the progressive lack of pathways able to sustain the production of one component when we are optimizing the metabolism to maximize the other. Jumps mark the sudden loss of pathways; in other words, a jump occurs when a crucial hub is eliminated, such as the Krebs cycle. As an example, we report the re-

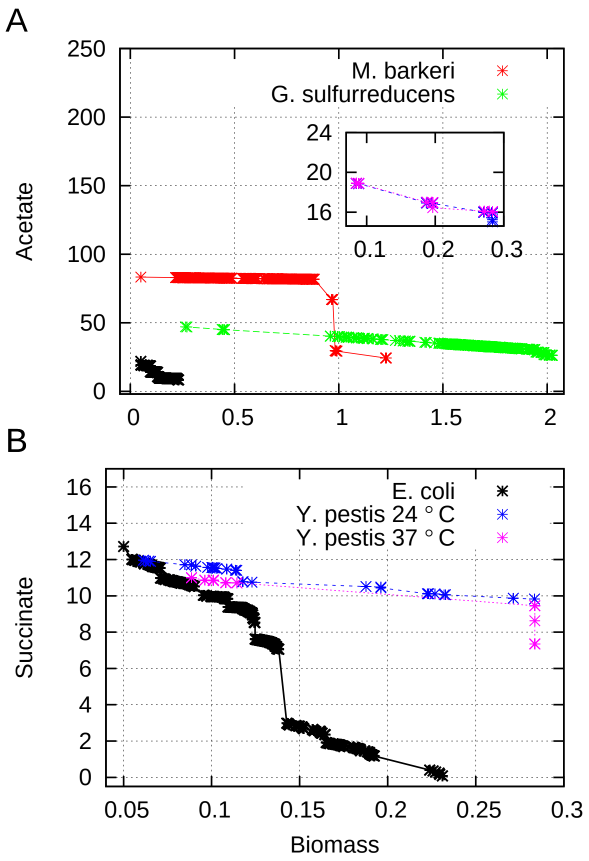


Figure 1: Pareto fronts obtained optimizing acetate (A)/succinate (B) production [mmolh^{-1} gDW^{-1}] and biomass formation [h^{-1}] in four metabolic networks. Succinate production is not computed for *M. barkeri* and *G. sulfurreducens* because these organisms do not provide succinate.

sults for Succ and Ac optimization in the *E. coli* network (Table 1). From the Pareto fronts of Figure 1, we selected six strains, each of which has a different genetic strategy. For each strain, we computed the *knockout cost* (k cost), i.e., the number of genes turned off. We are able to obtain mutants which produce +130% (A_1–A_2) of acetate in comparison with the wild type configuration, and +15000% (B_1) of succinate. Table 1 reports also the values obtained by the *Global Robustness* (GR) analysis. GR values represent the ability of the strain to ensure the desired production when small perturbations occur during the biotechnology design process. Moreover, ϵ-dominance analysis (Figure 2) reveals other suitable solutions. For example, with a minor k cost (11) we obtained 11.65 mmolh^{-1} gDW^{-1} in succinate (B_ϵ). Indeed, the (μ*,σ*) space of Figure 3 reveals pathways more sensitive in the model (at the upper-right corner). Thus, when we obtain solutions with the same production and different genetic strategies, we could choose the strategy that knocks out genes belonging to pathways located at the bottom left corner of Figure 3.

Table 1: Genetic strategies and Global Robustness analysis.

Strain	Acetate	Biomass	k cost	GR
A$_1$	19.198	0.052	12	0.43%
A$_2$	19.150	0.053	10	1.75%
A$_3$	18.532	0.096	9	13.55%
A$_4$	14.046	0.104	5	43.88%
Strain	**Succinate**	**Biomass**	**k cost**	**GR**
B$_1$	12.011	0.055	15	16.55%
B$_2$	10.610	0.087	8	19.58%
B$_\epsilon$	11.650	0.064	11	18.47%

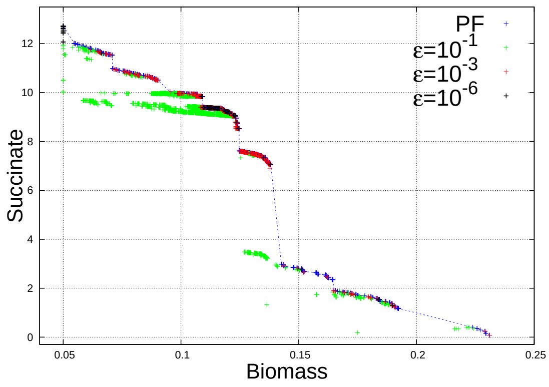


Figure 2: ϵ-dominance analysis results in *E. coli* network for succinate optimization.

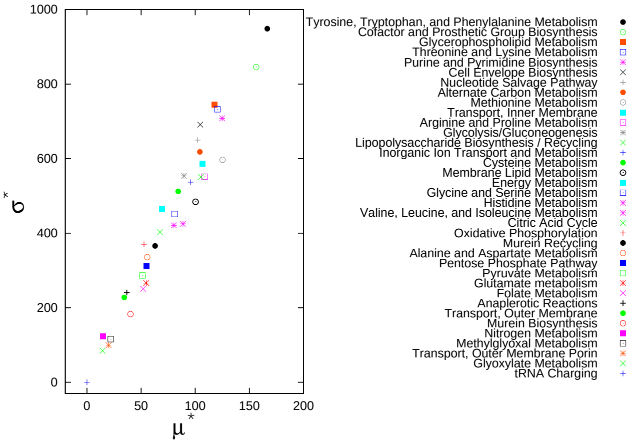


Figure 3: Pathway-oriented Sensitivity Analysis (PoSA) for *E. coli* network. A high mean μ* indicates an input with an important "overall" influence on the output. A large measure of variance σ* indicates an input whose influence is highly dependent on the values of the inputs.

3. REFERENCES

[1] P. Charusanti et al. *BMC Systems Biol.*, 5(1):163, 2011.

[2] A. M. Feist et al. *Mol Syst Biol*, 2, 2006.

[3] A. M. Feist et al. *Mol Syst Biol*, 3(121):291–301, 2007.

[4] M. Laumanns et al. *Evol. Comput.*, 10(3):263–282, 2002.

[5] J. Sun et al. *BMC Systems Biol.*, 3(1):15+, 2009.

[6] R. Umeton et al. In *Design Automation Conference, 2011 48th ACM DAC*, pages 747–752, 2011.

Sunday – June 3rd
Tech. Talks Session 2 - *Topic: Engineering, Parts, and Standardization*
2BDA.1 **Gene Variant Library Design for High Throughput Experimentation** Daniel Ryan and Dimitris Papamichail.
2BDA.3 **Standardizing Promoter Activity Through Quantitative Measurement of Transcriptional Dynamics** Wilbert Copeland and Herbert Sauro.

Gene Variant Library Design for High Throughput Experimentation

Daniel Ryan
National Institute for Mathematical
and Biological Synthesis
University of Tennessee
Knoxville, TN 37996, USA
+1 895 974 4962
ryan@nimbios.org

Dimitris Papamichail
Computer Science Department
University of Miami
Coral Gables, FL 33146, USA
+1 305 284 4189
dimitris@cs.miami.edu

ABSTRACT

Array-based oligonucleotide synthesis technologies provide access to thousands of custom-designed sequence variants at low cost. Large-scale synthesis and high-throughput assays have become valuable experimental tools to study in detail the interplay between sequence and function. We have developed algorithms for the design of diverse coding sequence libraries, to exploit the potential of multiplex synthesis and help elucidate the effects of codon utilization in gene expression.

Categories and Subject Descriptors

D.2.2 [**Algorithms**]: Nonnumerical Algorithms and Problems – *computations on discrete structures*

General Terms

Algorithms, Experimentation.

Keywords

Gene design, genomic libraries, synthetic biology.

1. INTRODUCTION

Gene synthesis is a process during which oligonucleotides are combined into larger DNA fragments, several hundred or thousand bases in length. In 2009, Gibson et al. introduced the *in-vitro isothermal assembly* technique [1], which was used to assemble a 16.3 kilo-base mouse mitochondrial genome from 600 overlapping 60-mers [2], and an entire 1.08 mega-base Mycoplasma genitalium genome from approximately 1000 cassettes of 1-kilo-base each [3]. In [4] this technique was used to create a combinatorial library of biochemical pathways, containing 144 combinations of 3 promoters and 4 gene variants of the acetate utilization pathway in E.coli. This feat demonstrates the effective use of assembly methods to accurately and efficiently construct combinatorial libraries and, more importantly, rationally designed sets.

Traditionally, due to the complexity of designing genomic sequences with well controlled attributes and the large gap of knowledge on the effect of these attributes, large scale gene design experiments have relied on random synonymous mutations to generate the gene libraries which are then studied in well characterized organisms and regulatory contexts. Welsh et al. [5] have performed experiments with genes encoding commercially valuable proteins, by synthesizing 72 variants and chimeric combinations. They showed that variation in expression is highly correlated to codon usage, although preferred codons were not those used most frequently by E.coli, the organism where the genes where expressed. In particular, they pinpointed 5-6 codons as most critical for expression. In constrast, a paper from the Plotkin lab [6] claimed that most of the variance in expression results from the amount of secondary structure in the 5’ end of the gene, after testing 154 variants of the GFP protein, carrying random synonymous mutations, also in E.coli. Further analysis of Plotkin’s dataset by Supec and Mac [7], using support vector machines and a M5’ regression tree model, identified 5 specific codons from 4 amino acids to contribute almost all of the variation in expression levels attributable to codon usage. These codons were different than the ones identified by Welsh et al. Additional findings from the Plotkin lab [8] indicate complex relationships between codon selection, translation initiation and elongation, misfolding of proteins and autocorrelation.

These results beg the question: which rules should one follow to design genes for optimized gene expression? Currently it is hard to tell, even in a well studied model organism such as E.coli. A common belief, emphasized by Plotkin’s group in [6] and by Super and Mac in [7], is that the mechanisms which determine gene expression can be established only by further large-scale experimentation. In this paper we present algorithms that allow the design of synthetic genes on the scale necessary to address the needs for large scale in-vitro experimentation.

2. METHODS AND RESULTS

We have investigated the problem of minimizing the number of genomic fragments needed to synthesize a library of gene variants, all coding for the same amino acid sequence, but each having a unique codon distribution. For example, assuming the aim is to test the contributions of a specific codon to the expression of the gene, one can alter the frequency of the codon in the mRNA encoding, For testing 4 different frequency levels (such as low, medium low, medium high and high) of the usage of a codon, 4 designs would be needed, each utilizing the codon at one of the 4 levels. For examining the effects of 5 different codons at 4 levels each, one would need to synthesize 1024 (4^5) different genes, to account for all possible combinations. The number of individual gene variants increases exponentially with the number of codons that we wish to investigate, as does the cost of synthesizing all the different genes.

IWBDA’12, June 3-4, 2012, San Francisco, CA, USA.

We developed an algorithm to minimize the number of sequence fragments required to construct gene variants, by sharing common fragments among variants. Let us examine a gene which can be assembled by 10 overlapping segments. If we define as 1x coverage the total worth of sequence required to synthesize one gene variant, we can observe that 2x coverage is sufficient to synthesize 2^{10} gene variants, as depicted in Fig. 1a, where all neighboring fragments share overlapping regions. In such a design, we have 2 variations for each of the 10 fragments, with one maximizing the target codon's occurrences, and the other minimizing it. Such a methodology though will not produce all unique designs from a codon frequency perspective, since the codon frequency values follow a binomial distribution. Out of 1024 gene variants, one contains the maximum number of occurrences of the target codon and another one the minimum, with the vast majority of constructs narrowly clustered midway. This effect is undesirable, independently of the high-throughput methodology used to assay the properties of the gene variants, since there always exist limits in the number of sequences that can be realistically sampled and tested, thus restricting the scope of the experiment with respect to codon frequency modulation.

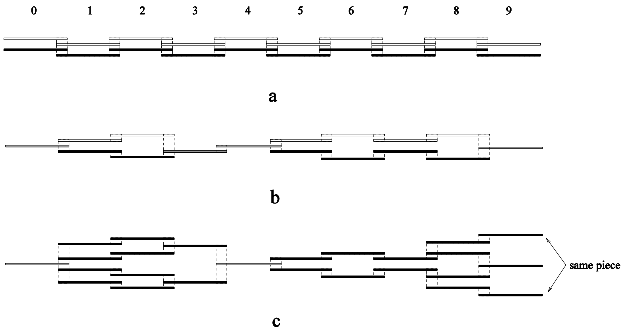


Figure 1: Combinatorial designs for gene variant synthesis. (a) single codon usage variation, two variant segments per position, binomial distribution of codon frequency, 1024 designs. (b) single codon usage variation, 4 frequency values, 4 designs. (c) 2 codon usage variations, 4 frequency values per codon, 16 designs

Our algorithm, for each codon whose frequency we intend to alter, examines consecutive DNA fragments to identify groups (intervals) containing enough corresponding amino acids to allow a 'step' between the frequency levels we wish to achieve. As an example, assuming that we would like to create constructs that vary the usage of a particular codon, call it 'C', according to the frequencies (.05, .30, .55, .80), we would identify consecutive fragments containing at least 25% (the step) of the corresponding amino acid's (call it 'A') occurrences in the gene. Then the algorithm identifies disjoint groups of consecutive fragments which, when combined, can produce the desired constructs, each with a unique frequency of the codon under consideration. An example of a design that can be used to construct gene variants utilizing that a codon at 4 frequencies, differing by 25%, is shown in Fig. 1b, where interval (1,2) contains at least 25% and interval (5,8) at least 50% of amino acid A. The problem as described reduces to prime factorization of the number of target codon frequencies and construction of intervals with minimized sum of lengths, which is similar to the *Frobenius coin problem* [9], but with bounded coefficients. An example varying the occurrences of two codons, with a second codon whose amino acid 'B' appears at least 50% in interval (1,3) and at least 25% in interval (8,9) is shown in Fig. 1c. In this example we can see that, instead of ordering 160 fragments to assemble the 16 desired gene variants, 24 segments suffice, with 24 being also the minimum number to realize this library design.

To demonstrate the optimization potential of our algorithm, we considered more elaborate and realistic test cases, involving the 4 codons and their corresponding amino acids (S, T, V, A) which Supek and Muc [7] identified as contributing most of the variation in expression in Plotkin's [6] experiments. Varying the occurrences of each codon at 4 frequency levels (.05, .30, .55, .80) would require synthesizing a library of 256 gene variants, in order to quantify the effect of these codons in the expression of a gene. In our example we use the GFP protein, with a length of 238 amino acids, setting the fragment size to 90bp and the overlap length to 18bp (with 10 overlapping fragments covering the whole coding sequence), values compatible with current synthesis and assembling technologies. Our algorithm produces a multiplex design with 7.2x coverage, for which we only need to order 2.8% as much sequence as we would for the 256 separate genes, reducing the synthesis cost of such an experiment from $70,000 to about $2,000, including modest cost of labor and material necessary to assemble the fragments. Similarly, varying single codon frequencies from 8 amino acids of the same protein at (0.1, 0.3, 0.5, 0.7) frequencies, and with all other parameters remaining the same, would require 65,536 gene variants to be ordered when no optimization is applied, where our algorithm can allow the exploration of the same design space with only 12.8x coverage, or in other words achieves in excess of 5000-fold savings.

3. REFERENCES

[1] Gibson, D.G., Young, L., Chuang, R.Y., Venter, J.C., Hutchison, C.A., 3rd, Smith, H.O.: Enzymatic assembly of DNA molecules up to several hundred kilobases. Nat Methods **6** (2009) 343-345

[2] Gibson, D.G., Smith, H.O., Hutchison, C.A., 3rd, Venter, J.C., Merryman, C.: Chemical synthesis of the mouse mitochondrial genome. Nat Methods **7** 901-903

[3] Gibson, D.G., Benders, G.A., Andrews-Pfannkoch, C., Denisova, E.A., Baden-Tillson, H., Zaveri, J., Stockwell, T.B., Brownley, A., Thomas, D.W., Algire, M.A., Merryman, C., Young, L., Noskov, V.N., Glass, J.I., Venter, J.C., Hutchison, C.A., 3rd, Smith, H.O.: Complete chemical synthesis, assembly, and cloning of a Mycoplasma genitalium genome. Science **319** (2008) 1215-1220

[4] Ramon, A., Smith, H.O.: Single-step linker-based combinatorial assembly of promoter and gene cassettes for pathway engineering. Biotechnol Lett **33** 549-555

[5] Welch, M., Govindarajan, S., Ness, J.E., Villalobos, A., Gurney, A., Minshull, J., Gustafsson, C.: Design parameters to control synthetic gene expression in Escherichia coli. PLoS One **4** (2009) e7002

[6] Kudla, G., Murray, A.W., Tollervey, D., Plotkin, J.B.: Coding-sequence determinants of gene expression in Escherichia coli. Science **324** (2009) 255-258

[7] Supek, F., Smuc, T.: On relevance of codon usage to expression of synthetic and natural genes in Escherichia coli. Genetics **185** 1129-1134

[8] Plotkin, J.B., Kudla, G.: Synonymous but not the same: the causes and consequences of codon bias. Nat Rev Genet **12** 32-42

[9] Beck, M., Robins, S.: The Coin-Exchange Problem of Frobenius. Computing the Continuous Discretely **C1** (2006) 3-23

Standardizing promoter activity through quantitative measurement of transcriptional dynamics

Wilbert B. Copeland
wcopelan@uw.edu

Herbert M. Sauro
hsauro@uw.edu

Department of Bioengineering, University of Washington
3720 15th Ave NE, Seattle, WA 98195

ABSTRACT

Current design strategies for synthetic gene networks often involve multiple rounds of experimental refinement, and this, in part, contributes to the inability to automate the design process. Functional characterization and standardization of synthetic gene network components can greatly aid the rational design of synthetic gene networks by increasing the predictive power of modeling methods employed in network design. Here we describe a fluorescence-based system to accurately and precisely measure promoter activity using RNA transcripts as reporter molecules. Measurements obtained describe promoter activity using standard units, and can be incorporated into computational models to better predict the function and contribution of promoter sequences to intracellular RNA and protein expression.

1. INTRODUCTION

Recent applications in synthetic biology have demonstrated the potential of rational biological design towards addressing a range of medical, environmental, and industrial problems [1-3]. In light of success in this area, it is still recognized that fundamental issues prevent microorganisms from being regularly engineered for meaningful, large-scale applications. Among these issues is the inability of scientists to accurately and precisely describe the functional characteristics of certain biomolecular components, such as promoter sequences. This limitation impedes the development of standardized measurements and reduces the predictive power of model techniques used for network design.

Work to characterize promoters for the purpose of standardization has been discussed in recent years [4-5], and these efforts have resulted in the large-scale, professional production of characterization data for promoters [6]. Previous works have used fluorescent proteins to gauge promoter activity; however, the signal reported from proteins is a conglomeration of many cellular processes, including transcription, translation, and protein maturation. The ability to measure promoter activity via RNA transcripts may increase characterization accuracy and aid standardization efforts; however, a canonical, high-throughput strategy for observing RNA dynamics has not been established.

Recent studies have demonstrated that RNA aptamers can stably bind particular non-fluorescent dyes *in vitro* to confer measureable levels of fluorescence [7]. The dyes used in these studies strongly absorb light at a specific wavelength and dissipate the stored energy as heat through molecular motion. When the molecular motion of the dye is restricted following aptamer binding, the dye releases the energy at a longer wavelength, yielding fluorescence. Even more recently, it was demonstrated that fluorescence upon intracellular binding of aptamer and dye could be measured [8].

Here we present an overview of our work towards characterizing promoter sequences using fluorescence-activating aptamers. The goal of this project is, in part, to improve upon the standardization efforts for promoters by providing accurate, quantitative measurements of transcriptional activity. These precise measurements can be incorporated into computational models as researchers attempt to predictably design gene networks with specific behaviors.

2. METHODS

2.1 System Model

The system model for our proposed fluorescence-based promoter activity reporter is based on cellular expression of an aptamer that binds with high affinity and specificity to malachite green, a typically non-fluorescent dye. The molecular events of our system can be described according to Figure 1 and Equations 1-2.

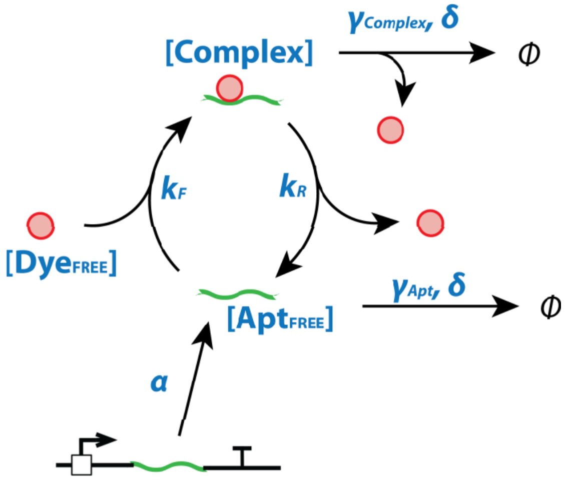


Figure 1. Schema of the molecular processes involved in fluorescence signal production via aptamer-dye binding.

$$\frac{d[APT_{FREE}]}{dt} = \alpha - (\gamma_{APT} + \delta)[APT_{FREE}] - k_F[DYE_{FREE}][APT_{FREE}] + k_R[COMPLEX] \quad \textbf{(1)}$$

$$\frac{d[COMPLEX]}{dt} = k_F - [DYE_{FREE}][APT_{FREE}] - (k_R + \gamma_{COMPLEX} + \delta)[COMPLEX] \quad \textbf{(2)}$$

2.2 Gene Network Assembly

An RNA aptamer was created that yields significant levels of fluorescence when bound to malachite green [4]. Flanking the aptamer with an artificial RNA scaffold increased the transcript's intracellular stability and allowed for observation of intracellular fluorescence. Expression in *E. coli* MG1655 is achieved by placing the RNA sequence (MGA5S) on a medium-copy number plasmid (pSB3K3) under the control of a promoter and followed by the transcriptional terminator (BBa_B0015).

2.3 Dynamics Measurements

A fluorescence signal is produced exclusively upon aptamer-dye binding; therefore fluorescence is a direct indicator of the concentration of the bound aptamer-dye complex, **[Complex]**. Kinetic parameters in our model that need to be determined are: aptamer synthesis rate (**α)**, aptamer degradation rate ($\boldsymbol{\gamma}_{\mathbf{APT}}$), complex degradation rate ($\boldsymbol{\gamma}_{\mathbf{COMPLEX}}$), and the forward ($\mathbf{k_F}$) and reverse ($\mathbf{k_R}$) aptamer-dye binding rates.

E. coli MG1655 cells are grown in M9CA until they reach steady-state RNA production levels. The aptamer is expressed behind various promoters while preserving all other cellular conditions and gene network components. Fluorescence was measured for each construct to observe the effective strength of the specific promoter. Steady-state RNA production can be approximated by observing d(FL/OD)/dt. To tease out the kinetic parameters, the measured fluorescence values can be correlated to total intracellular aptamer concentrations determined using qRT-PCR.

Characterizing a novel promoter sequence simply requires researchers to insert the promoter onto a standard plasmid in *E. coli* MG1655. Since the kinetic parameters have been previous determined, and should remain unchanged, aside from the aptamer synthesis rate due to the new promoter, characterizing promoter activity should be possible exclusively from fluorescence data.

3. RESULTS AND DISCUSSION

3.1 Characterization

Our initial tests have been performed by expressing MGA5S behind 10 unique promoters from the Anderson library [9], and we have observed varying levels of fluorescence. For most situations the fluorescence of MGA5S has correlated well with their respective fluorescent protein counterpart; however in a few cases expression levels between RNA and protein are significantly different. This result may highlight the importance of using fluorescent RNA reporters as a proxy for transcriptional activity rather that fluorescent proteins, especially for applications based on RNA logic. Future steps include correlating fluorescence data with absolute intracellular RNA concentrations. Following correlation, we should be able to apply this method to quickly and conveniently characterize massive libraries of natural and artificial promoters.

3.2 Standardization

This fluorescence-based method should allow for high-throughput characterization of large sets of promoters. Additionally, promoter activity measurements will be reported in standard units. Since fluorescence values are fundamentally arbitrary, when observed with a reference promoter, measurements obtained can be normalized across different types of equipment and meaningfully shared between laboratories. Additionally, since transcription rates can be measured purely from observing intracellular fluorescence data, measurement discrepancies between researchers due to varying levels of expertise with technically challenging protocols such as qRT-PCR should be reduced.

4. ACKNOWLEDGMENTS

This work is supported, in part, by the NSF Graduate Research Fellowship Foundation and NSF BEACON Center. Our thanks to members of the Sauro Lab.

5. REFERENCES

[1] Ro, D., et al., Production of the antimalarial drug precursor artemisinic acid in engineered yeast. Nature, 2006. 440(7086): p. 940-943.

[2] Steen, E., et al., Microbial production of fatty-acid-derived fuels and chemicals from plant biomass. Nature, 2010. 463(7280): p. 559-562.

[3] Lu, T. and J. Collins, Engineered bacteriophage targeting gene networks as adjuvants for antibiotic therapy. Proceedings of the National Academy of Sciences, 2009. 106(12): p. 4629.

[4] Kelly, J., et al., Measuring the activity of BioBrick promoters using an in vivo reference standard. Journal of biological engineering, 2009. 3(1): p. 4.

[5] Canton, B., A. Labno, and D. Endy, Refinement and standardization of synthetic biological parts and devices. Nature biotechnology, 2008. 26(7): p. 787-793.

[6] BioFab.org: http://biofab.org/

[7] Babendure, J.R., S.R. Adams, and R.Y. Tsien, Aptamers switch on fluorescence of triphenylmethane dyes. J Am Chem Soc, 2003. 125(48): p. 14716-7.Forman, G. 2003. An extensive empirical study of feature selection metrics for text classification. *J. Mach. Learn. Res.* 3 (Mar. 2003), 1289-1305.

[8] Paige, JS., Wu, KY., Jaffrey, SR. RNA Mimics of Green Fluorescent Protein Science 29 July 2011: 333 (6042), 642-646.

[9] Registry of Standard Biological Parts. http://partsregistry.org/Part:BBa_J23100

Sunday – June 3rd

Invited Talk**: Milan Stojanovic, Columbia**

“Molecular Computing: From Games to Practical Applications”

This talk will focus on two molecular systems capable of information processing: (i) Deoxyribozyme-based logic gates and various deoxyribozyme-based molecular automata playing games against human opponents; and (ii) Strand-displacement cascades and their ability to assess presence and absence of surface markers on cells.

Milan Stojanovic got his Ph. D. in Organic Chemistry with Yoshito Kishi at Harvard University. After a brief stint in industry, he joined Donald Landry's group for a postdoctoral fellowship. He remained at Columbia University, where he is now an Associate Professor or Medical Sciences and Biomedical Engineering.

Sunday – June 3rd

Tech. Talks Session 3 - *Topic: Characterization and System Identification*

3BDA.1 **Validation of Network Reverse Engineering Using a Benchmark Synthetic Gene Circuit**
Taek Kang, Jacob White, Eduardo Sontag and Leonidas Bleris.

3BDA.2 **Model Checking for Studying Timing of Events in T cell Differentiation**
Paolo Zuliani, Natasa Miskov-Zivanov, Penelope Morel, James R. Faeder, and Edmund M. Clarke.

3BDA.3 **Network-Based Genome Design and Engineering with Direct Logical-to-Physical Compilation**
Chih-Hsien Yang, Jesse Wu, Chi Yang, Tao-Hsuan Chang and Chuan-Hsiung Chang.

Validation of Network Reverse Engineering Using a Benchmark Synthetic Gene Circuit

Taek Kang
University of Texas at Dallas
Richardson, TX 75080

Jacob T White
University of Texas at Dallas
Richardson, TX, USA

Eduardo Sontag
Rutgers University
New Brunswick, NJ, USA
sontag@math.rutgers.edu

Leonidas Bleris
University of Texas at Dallas
Richardson, TX, USA
bleris@utdallas.edu

1. MOTIVATION

Developing automated and rigorously validated methodologies for unraveling the complexity of biomolecular networks in human cells is one of the central challenges to life scientists and engineers. We use synthetic gene circuits integrated in kidney cells, as platforms for the development of new and the refinement of existing reverse engineering methodologies. In this paper, we use modular response analysis, a method that builds a fine-grained view of local component connections through semi-quantitative estimates of connection strength for near-linear perturbations of a network. We show using a benchmark circuit that combines transcriptional and post-transcriptional reconstruction that we can reliably reconstruct causal relationships by perturbing selected components of the network and comparing the steady-state response of each component to the unperturbed steady-state.

An intrinsic difficulty in capturing direct interactions between components, at least in intact cells, is that any perturbation to a particular component using tools such as RNAi, hormones, or chemical interventions may rapidly propagate throughout the network, thus causing global changes which cannot be easily distinguished from direct effects. An approach to solving this global-to-local problem is the "unraveling", or Modular Response Analysis (MRA) method [2]. The MRA experimental design compares the steady states which occur after performing independent perturbations to each "modular component" of a network. These perturbations might be genetic or biochemical. In MRA, a set of experiments are run where each module is perturbed individually, all outputs are measured at steady-state, and these are compared to the unperturbed steady-state case to form a matrix of "global response" values. From this matrix we can obtain the Jacobian matrix of the system, that contains the "local response coefficients". Each element in this matrix which is not a diagonal element corresponds to a directed network connection between modules. If each connection is monotone, then we expect the sign of each connection recovered in this linear approximation to match the sign of the underlying system. For smaller perturbations, the linear model is more quantitatively close to the underlying system, but at a cost of greater uncertainty from noise.

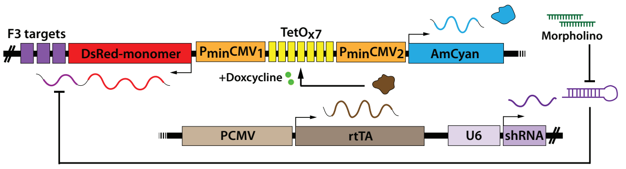


Figure 1: Synthetic circuit stably integrated in Flp-In cells. It has a U6 promoter constitutively producing shRNA, a CMV promoter constitutively producing rtTA, and a bidirectional minCMV 7xTetO promoter coding amCyan on one side and dsRed with three shRNA targets on the other. Doxycycline binds and activates rtTA, and Morpholino binds and inhibits shRNA.

Here we address cases where the calculation returns a value near zero, which may represent a weak connection, a saturated connection, or no connection. Synthetic gene circuits are perfectly suited for a validation process, since the network connections are known and may be checked against the reconstruction from data. In this study we measure the response coefficients of two outputs from two inputs using flow cytometry data. We identify non-connections through bootstrap resampling, where we calculate this value many times with random subsamples of the data, which generates a confidence interval for the measurement [1]. This interval determines whether the calculated response coefficient is statistically significant compared to the no-connection case.

2. EXPERIMENTS

The gene circuit studied has a bidirectional minCMV promoter under control of 7xTetO repeats, which produces amCyan and dsRed. The circuit constitutively expresses rtTA and shRNA via CMV and U6 promoters respectively. DsRed has three 3'UTR shRNA targets and is downregulated by the constitutive shRNA. Doxycycline (small chemical ligand which activates rtTA) controls the transcription rate of the two reporters, and a Morpholino oligomer (GeneTools) blocks the shRNA through complementary binding, thereby enhancing dsRed expression (Figure 1). This circuit was stably integrated in Flp-In 293 cells (Invitrogen). For the measurements the cells are plated, grown 24 hours, induced with doxycycline with/without morpholino, then grown an additional 72 hours before flow cytometry. We perform a titration of doxycycline from 0 to 10 μg/mL (Figure 2a,b) and observe that both fluorescent outputs respond. A titration of morpholino from 0 to 5 μmol/mL at full doxycycline induction is shown in Figure 2c,d; we observe that dsRed responds, while amCyan remains practically constant.

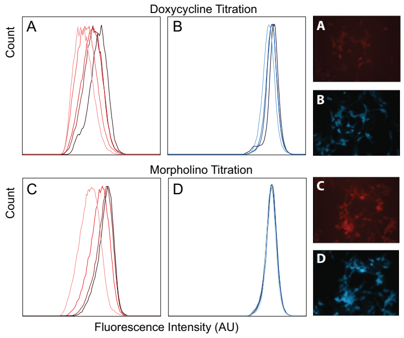


Figure 2: Cytometry data from input titrations. A) dsRed signal as Doxycycline varies (light red = no Dox, dark red = high Dox). B) amCyan signal as Doxycycline varies (light blue = no Dox, dark blue = high Dox). C) dsRed signal for Morpholino titration. D) amCyan signal for Morpholino titration, which should not be affected. The microscopy images correspond to the maximally induced well.

3. NETWORK RECONSTRUCTION

Small perturbations to either input, Doxycycline or Morpholino, yields an approximately linear response. We assume the following linear system, where R and C are the fluorescent outputs dsRed and amCyan, β and α are their unperturbed production and degradation rates, and D and M are the inputs, which are not observed:

$$\frac{d}{dt}\begin{bmatrix} C \\ R \end{bmatrix} = \begin{bmatrix} -\alpha_c & 0 \\ 0 & -\alpha_r \end{bmatrix} \times \begin{bmatrix} C \\ R \end{bmatrix} + \begin{bmatrix} a & c \\ b & d \end{bmatrix} \times \begin{bmatrix} D \\ M \end{bmatrix} + \begin{bmatrix} \beta_c \\ \beta_r \end{bmatrix}$$

The objective is to determine a, b, c, and d, which represent the partial derivatives of each rate equation with respect to each input near steady state. We know beforehand that c=0 because Morpholino should have no effect on amCyan expression but we must determine this from the analysis. In this experiment we cannot determine whether C and R have direct interconnections (we assume they don't) because varying our inputs do not perturb these nodes individually. Without interconnections between C and R, the coefficients a, b, c, and d are equal to the measured total derivatives, the difference in fluorescence divided by the difference in input. This input perturbation magnitude would not be included in the global response matrix because it cancels out when computing the local response matrix. As an analogy, here we know the ratio of a to b uniquely and without a quantitative perturbation magnitude (Figure 3d). We gate cytometry events which are positive for both fluorescent proteins; histograms of log-scale fluorescence are shown in Figure 2. We select two different wells from each titration to represent the perturbed and unperturbed cases. We calculate the average fractional change $\Delta ln(x_i)$ by log-transforming all data points, selecting 200 random events (with replacement) from a perturbed and unperturbed sample, averaging the log fluorescence of each set, and subtracting. This difference is calculated for 200 random sets of events, and box plots showing the 99.5% confidence interval for these repeated estimates are shown in Figure 3d. This figure shows that the response of amCyan signal for a Morpholino perturbation is statistically insignificant; similarly to a t-test, this says that we are not 99.5% confident that the global response $\Delta ln(x_i)$, which approximately equals the local response for the Morpholino-amCyan connection, is different from zero.

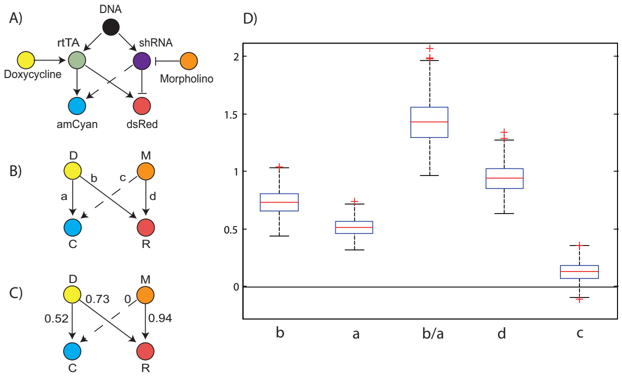


Figure 3: A) Gene circuit diagram with all major components. The dotted line indicates there is no connection from shRNA to amCyan, but we pretend not to know this before the analysis. B) Simplified circuit diagram showing the two inputs and their connections to the two outputs. C) Most likely interaction strength of each node calculated by the bootstrap method. D) Box plots showing the uncertainty in each measured local response. The ratio b/a is an example of a perturbation magnitude-independent estimate like might appear in the full calculation of a local response matrix. Connection c is statistically insignificant because its confidence interval intersects zero.

By applying bootstrap resampling we were able to identify local response components which have no statistical significance. For a more complex system, we would use random samples to calculate the global responses and use these single estimates to re-calculate the local responses many times, generating confidence intervals for each network connection. We used a synthetic gene circuit for validating a reverse engineering methodology. We believe that our results show promise towards automating the process of unraveling complexity in natural pathways.

4. ACKNOWLEDGMENTS

The authors would like to acknowledge support from TxACE SRC and NIH (GM098984).

5. REFERENCES

[1] B. Efron and R. Tibshirani. *An introduction to the bootstrap*, volume 57. Chapman & Hall/CRC, 1993.

[2] B.N. Kholodenko, A. Kiyatkin, F.J. Bruggeman, E. Sontag, H.V. Westerhoff, and J.B. Hoek. Untangling the wires: a strategy to trace functional interactions in signaling and gene networks. *Science's STKE*, 99(20):12841, 2002.

Model checking for studying timing of events in T cell differentiation

Paolo Zuliani[1], Natasa Miskov-Zivanov[2], Penelope Morel[3], James R. Faeder[2], and Edmund M. Clarke[1]

Short Abstract — **We use computational modeling and formal analysis techniques to study the temporal behavior of a logical model of the naïve T cell differentiation. The model is analyzed formally and automatically by performing temporal logic queries via statistical model checking.**

I. Introduction and Motivation

The goal of this study is to identify key factors and pathways that contribute to the discrimination of the T-cell receptor (TCR) signal strength (*i.e.*, antigen dose/duration/affinity presented to TCR) by the differentiating T cell (Figure 1(a)). Different T cell phenotype ratios play an important role in T-cell mediated immunity, in both autoimmune diseases and in cancer. The two primary phenotypes we consider are: 1) regulatory (Treg) cells that express the transcription factor Foxp3 but do not express the cytokine IL-2; 2) and helper (Th) cells that do not express Foxp3 but do express and secrete IL-2. Control of the Treg vs. Th cell phenotype induction is a promising approach to either eliminate antigen-specific Treg cells and decrease (or even reverse) immune suppression in cancer, or enhance Treg induction to prevent autoimmune diseases. Previous studies have indicated that the timing of T cell stimulation, both antigen dose and the duration of antigen stimulation, strongly influence the T cell phenotype choice [1].

To study this system, we apply computational modeling approaches and formal methods from electronic design automation (EDA). The model used in this work (described in [2]) couples exogenous signaling inputs to T cell phenotype decisions. This model was developed using a discrete, logical modeling approach, and simulated using random asynchronous approach and *BooleanNet* tool [3]. Model simulations described in [2] allow for recapitulating a number of experimental observations and provide new insights into the system. However, to test new properties of the model, it is usually necessary to write new parts of the simulator code, or manually analyze a significant amount of simulation data. This approach quickly becomes tedious and error-prone.

In this work, we apply temporal logic model checking to automatically analyze the behavior of the model. Since the underlying semantic model of *BooleanNet* is essentially a discrete-time Markov chain, we need to verify probabilistic (stochastic) models. The verification problem for stochastic systems amounts to compute the probability that a given temporal logic formula is satisfied by the system. One approach to the verification problem uses precise numerical methods to compute exactly the probability that the formula is true (*e.g.* [4]). However, these methods suffer from the state explosion problem, and do not scale well to large-scale systems. Statistical model checking can be effectively used for verifying temporal logic specifications for systems affected by the state explosion problem. The technique relies on system simulation, thereby avoiding a full state space search. This implies that the answer to the verification problem (*i.e.*, the probability that the property holds) is only approximate, but its accuracy can be arbitrarily bounded by the user. In return, statistical model checking is more scalable and hence more useful for large models.

II. Methodology

The steps of our methodology are presented in Figure 1(b) and described below. We encode relevant properties of the model as temporal logic formulae, which are then verified via statistical model checking. We use Bounded Linear Temporal Logic (BLTL) as our specification language. BLTL restricts the well-known Linear Temporal Logic (LTL) with time bounds on the temporal operators. For example, a BLTL formula expressing the specification "it is not the case that in the **F**uture 10 time steps CD25 is **G**lobally activated (*i.e.*, it equals 1) for 17 time steps" is written as

$$\neg \mathbf{F}^{10}\,\mathbf{G}^{17}(\mathrm{CD25} = 1)$$

where the $\mathbf{F}^{10}$ operator encodes "future 10 time steps", $\mathbf{G}^{17}$ expresses "globally for 17 time steps", and CD25 is a state variable of the model. The syntax of BLTL is given by:

$$\psi ::= y \sim v \mid \psi_1 \wedge \psi_2 \mid \psi_1 \vee \psi_2 \mid \neg \psi_1 \mid \psi_1\, \mathbf{U}^t \psi_2$$

where $\sim \in \{\leq, \geq, =\}$, $y \in$ SV (the finite set of state variables), $v \in R$, $t \in R_{>0}$, and $\neg$,$\vee$, $\wedge$ are the usual Boolean connectives. Formula of the type $y \sim v$ are also called atomic propositions. The formula $\psi_1\mathbf{U}^t\ \psi_2$ holds true if and only if, within time t, ψ_2 will be true and ψ_1 will hold until then. Note that the operators $\mathbf{F}^t$ and $\mathbf{G}^t$ referenced above are easily defined in terms of the until $\mathbf{U}^t$ operator: $\mathbf{F}^t\ \psi = \mathit{true}\ \mathbf{U}^t\ \psi$ requires ψ to hold true within time t (*true* is the atomic proposition identically true); $\mathbf{G}^t\ \psi\ = \neg\, \mathbf{F}^t \neg\, \psi$ requires ψ to hold true up to time t.

We have combined *BooleanNet* with a parallel statistical model checker, so that verification of BLTL properties can be performed efficiently and automatically on a multi-core system. Statistical model checking treats the verification problem for stochastic systems as a statistical inference problem, using randomized sampling to generate traces (or simulations) from the system model, then using model checking methods and statistical analysis on those traces. Efficient Bayesian techniques were introduced and successfully applied to the verification of rule-based models of signaling pathways and other stochastic systems [5][6]. In particular, the approach is based on sequential estimation, and given a coverage probability and an interval width, it returns a Bayesian confidence interval for the probability that the BLTL formula is true.

III. Results

Experimental observations from [1] that the induction/expansion of Foxp3+ Treg cells by low dose antigen is inversely correlated with the levels of signaling via the mTOR pathway suggest a complex interaction between cell surface receptors, signaling molecules and important transcription factors. The model in [2] captures critical

[1]Computer Science Department, Carnegie Mellon University, E-mail: {pzuliani, emc}@cs.cmu.edu.

[2]Department of Computational and Systems Biology, School of Medicine, University of Pittsburgh, E-mail: {nam66,faeder}@pitt.edu

[3]Department of Immunology, School of Medicine, University of Pittsburgh, E-mail: morel@pitt.edu

This work was supported in part by an NSF Expeditions in Computing grant (award ID 0926181).

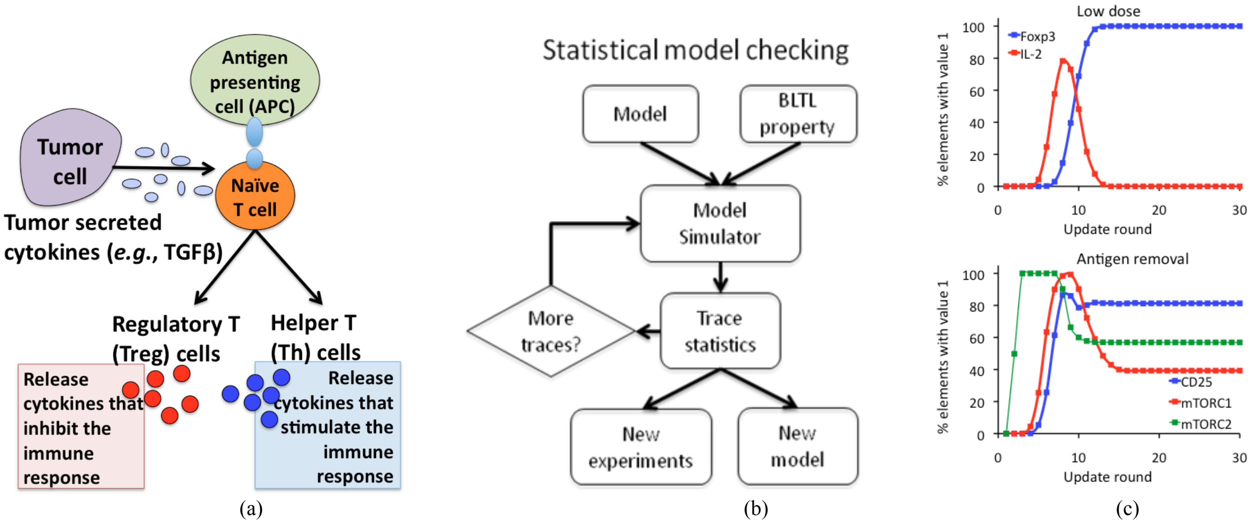


Figure 1. Modeling of immune systems cells: (a) Differentiation of naïve T cells into Teg or Th, induced antigen presented by APC, or cytokines secreted by tumor cells; (b) statistical model checking flow; (c) model simulation results for two scenarios.

signaling events, from stimulatory signals at receptors, through activation of transcription factors, to production of proteins representing different phenotypes.

Several model simulation results obtained using *BooleanNet* are shown in Figure 1(c). These results present the behavior of critical elements in the model averaged across 1000 simulation trajectories, for two different stimulation scenarios. When naïve T cells are stimulated with low antigen dose, they can differentiate into Treg cells expressing Foxp3. Similarly, model simulations that mimic the low antigen dose case result in steady state with Foxp3=1 (Figure 1(c) (top)). Model simulation results show that the behavior of IL-2 gene expression early after stimulation is similar for both low and high antigen dose. This is not so straightforward to measure in experiments as IL-2 is measured outside of cells, where it is consumed quickly after being expressed and secreted. What is not clear from averaged simulation trajectories (Figure 1(c) (top)) is whether IL-2 reaches value 1 on all trajectories, but at different update rounds, or whether it reaches value 1 on only 80% of trajectories. To test this, we consider the property F^{20} (IL2 = 1). Statistical model checking shows that the probability that this property holds is close to 1. We have also computed the probability that IL-2 remains at level 0 until its inhibitor, Foxp3, becomes 1. This property:

$$(\mathrm{IL2} = 0)\ \mathrm{U}^{15}\ (\mathrm{FOXP3} = 1)$$

is returned as a low-probability event. In other words, our model predicts initial increase in IL-2, irrespective of antigen dose scenario, and the criticality of variations in other element values for phenotype decision.

Another observation from experiments is that removal of antigen 18 hours after stimulation results in a mixed population of Treg and Th cells. Studies of the model have indicated that early events and relative timing of the Foxp3 activating and inhibiting pathways play crucial role in this differentiation. Figure 1(c)(bottom) shows transient behavior of CD25 (main element on Foxp3 activating pathway) and mTORC1/mTORC2 (inhibitors of Foxp3). With model checking, we were able to carry further and more efficient studies of early behavior of these elements. In Table I, we present a set of properties that we tested using statistical model checking and results obtained. We also include elapsed time that was necessary for checking those properties.

This scenario results in a mixed population of cells with different phenotype. Model checking results outline that early events in CD25, mTORC1 and mTORC2 are good predictors of the mixed population, as most of the results show close to 50% successes. In other words, although the tested properties would return more uniform behavior in other scenarios, in the case of antigen removal, we see more variability between possible trajectories. The next step now is to design further queries that could uncover exact relationship between early events and specific outcomes.

Table I.

Tested properties and model checker runtime on a 48-core system. Coverage probability=0.999; half-interval=0.01, except for Property 1 (=0.001)

	Property	Probability estimate and sample size	Elapsed time [s]

1	G^7 ~(MTORC1 = 1 & MTORC2 = 1)	estimate = 0.0188048
		samples = 200,160
			1,946

2	F^7 (MTORC1 = 1 & MTORC2 = 1)	estimate = 0.980884
		samples = 2,352
			23

3	F^{10} (MTORC1 = 1 & MTORC2 = 1 & CD25 = 0 & (F^{18} (CD25 = 1)))	estimate = 0.60104
		samples = 25,968
			253

4	F^{28} (MTORC1 == 1 & MTORC2 == 1 & CD25 == 0 & (F^1 (CD25 == 1)))	estimate = 0.592195
		samples = 26,160
			254

5	F^{10} (MTORC1 = 1 & MTORC2 = 1 & CD25 = 0 & (F^1 (G^{17} (CD25 = 1))))	estimate = 0.39669
		samples = 25,920
			254

IV. Conclusion

Model checking is an efficient approach for studying cell signaling network models, as it allows for answering a variety of questions about the system. Instead of manually analyzing simulation trajectories and large output files, one creates properties that can be automatically verified. We uncovered several relationships between early behavior of elements in our T cell model. With the framework that we created, we will continue to study this model, focusing on several other key relationships, such as the one between Foxp3 and PTEN.

References

[1] M. S. Turner et al., "Dominant role of antigen dose in CD4+Foxp3+ regulatory T cell induction and expansion," in J. of Immunology, 183, pp. 4895-903, 2009.

[2] N. Miskov-Zivanov et al., "Modeling and Analysis of Peripheral T Cell Differentiation," in preparation.

[3] I. Albert et al. "Boolean network simulations for life scientists." *Source Code for Biology and Medicine* 2008, **3**:16

[4] M. Kwiatkowska et al. "PRISM 4.0: Verification of Probabilistic Real-time Systems." In Proc. of CAV'11, LNCS 6806, pages 585-591, 2011.

[5] H. Gong et al. "Analysis and Verification of the HMGB1 Signaling Pathway." BMC Bioinformatics 2010, 11(Suppl 7):S10.

[6] P. Zuliani et al. "Bayesian Statistical Model Checking with Application to Stateflow/Simulink Verification." In HSCC 2010, pages 243-252, 2010.

Network-Based Genome Design and Engineering with Direct Logical-to-Physical Compilation

Chih-Hsien Yang
blent@gel.ym.edu.tw

Jesse Wu
jesse@gel.ym.edu.tw

Chi Yang
sheep@gel.ym.edu.tw

Tao-Hsuan Chang
doudi.tw@gel.ym.edu.tw

Chuan-Hsiung Chang
chc@gel.ym.edu.tw

ABSTRACT

We present a computer-aided platform for designing and engineering of mega base-pair (Mbp) genetic systems. To achieve a comprehensive design on a whole-genome scale, we have developed a new methodology to allow designers to make genetic manipulations directly on biological pathways and networks. These genetic manipulations trigger automatic adjustment on the underlying features and sequences, eliminating the need for the extensive manual modification of hundreds of genomic features or thousands of nucleotides at a time.

Categories and Subject Descriptors

J.3 [**Computer Applications**]: Life and Medical Science – *Biology and genetics*. J.6 [**Computer Applications**]: Computer-aided Engineering – *computer-aided design (CAD), computer-aided manufacturing (CAM).*

General Terms

Algorithms, Design, Standardization, Languages, Theory.

Keywords

Genome Design, Genome Engineering, Genome Compilation.

1. INTRODUCTION

DNA-based design and engineering technologies enable the constructions of artificial genetic materials for programing the behaviors of living organisms [1-3] or re-creating life forms [4-7] from scratch. To accelerate the development of design and engineering in synthetic biology, computer-aided methodologies play an essential and critical role in this process [8, 9].

Constructing a genetic system involves design and engineering at four different abstraction levels: sequence, part, device and system levels. Tools for designing single parts, such as DNA, RNA and protein individually are already available and relatively mature. The compilation between sequence and (genetic) part also has been implemented by MIT's registry of standard biological parts, and has been used as a form of design library in popular synthetic biology design tools, such as Gene Designer [10], GenoCAD [11] and TinkerCell [12]. The formalization of ordering design principles of synthetic genetic constructs (devices) from part libraries has been demonstrated by the use of attribute grammars [11] in GenoCAD system. The part-to-device compilation and optimization framework for genetic devices composed of tens of genetic parts have also been proposed by the uses of attributes or parameters of biological parts [13-15].

The works mentioned above are attempts to build a part-device-system design model in a bottom-up fashion for synthetic biology. However, many practical and industrial-scale biotechnological applications rely on the re-design and engineering of existing and commonly used genetic systems [1, 2, 16], such as baker's yeast (genome size: 12.1Mbp) or *Escherichia coli* (genome size: 4.6Mbp). Thus, we have established a biological interaction-oriented design and engineering approach for a whole-genome genetic system with a direct compilation between the logical connections within biological networks and the physical nucleotides of genomes. This method allows designers to efficiently generate initial design drafts after genetic manipulation on whole-genome networks.

2. MATERIALS AND METHODS

2.1 Data Model

Biological networks are represented with the SBNL (Synthetic Biology Network Language) protocol and visualized by Cytoscape Web. There are six primitive biological processes in our SBNL protocol: Transcription, Translation, Metabolic reaction, Transporting reaction, Signal Sensing, and Signal Transduction. There are six primitive biological components: Transcription Unit (TCU), Translation Unit (TLU), RNA, Protein, Ligand (small chemical compound) and Signal connected with these biological processes.

2.2 Data Source Collection

To implement and demonstrate logical-to-physical compilation, the supports of several data sets are required as follows:

We have collected: (1) TF-promoter pairs, (2) RBS (ribosome binding sites), (3) Transcriptional Terminators, and (4) Ribo-switches from RegulonDB (Release 7.4), RegTransBase, bioinformatics prediction and literature. These data sets support the first three actions (add/modify controls on transcription, add/modify controls on translation, and modify inputs/outputs on the configuration of TCU/TLU) shown in Table 1 for gene expression and control.

The last three actions in Table 1 are designs on metabolic and signaling pathways, and thus the primary data sources for metabolite-converting, metabolite-transporting, sensing and responding reactions are KEGG, BioCyc, TransportDB and TCDB.

2.3 Logical-to-Physical Compilation

To translate (genetic) manipulations on a genome-scale interactome network represented with SBNL into physical operations of physical nucleotide sequences, we have defined a set of "compilable" actions based on the types of biological components and interactions involved.

Table 1. Examples of compliable actions

Logical Action	Interpreted Physical Operations
Add a TF on selected TCU/TLU	Adds a TFBS of selected TF before the promoter of selected TCU/TLU (i.e. TF-dependent transcriptional regulation)
Add a Ligand on selected TLU	Replaces original RBS of selected TLU as a selected ligand-regulated RBS (i.e. Riboswitch)
Assign a new protein output of selected TLU	Replaces original CDS of selected TLU
Add a new reaction between two chemicals	Adds a new TLU encoding enzyme performing the selected chemical reaction
Add a new transporter for selected chemical	Adds a new TLU encoding transporter performing the selected transportation
Add a new sensory path on selected signal	Adds a new TLU encoding sensory protein performing the selected signal sensing

3. RESULTS

We have implemented a direct compilation method from logical design of biological networks to physical implementation of genomic nucleotides. A top-down genome re-design scenario has also been implemented in our Genome Design and Engineering Workbench (GDEW) as shown in Figure 1, including (1) a network editor for pathways and reactions, (2) a genome editor for direct manipulations of genomics features (i.e. genetic parts) on the chromosome, and (3) a sequence editor for detailed modifications.

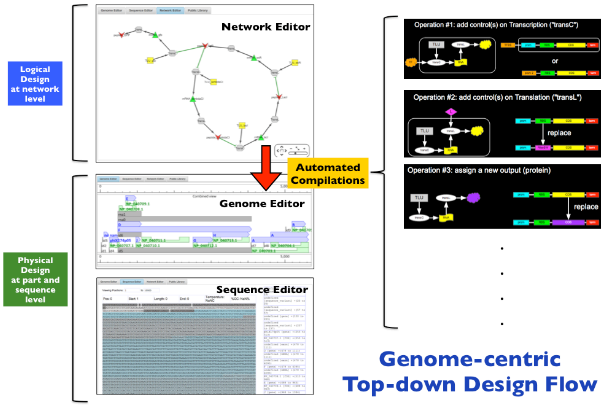

Logical Design at network level
Network Editor
Automated Compilations
Genome Editor
Sequence Editor
Physical Design at part and sequence level
Operation #1: add control(s) on Transcription ("transC")
or
Operation #2: add control(s) on Translation ("transL")
replace
Operation #3: assign a new output (protein)
replace
TLU
Genome-centric Top-down Design Flow

Figure 1. A top-down workflow for whole genome design

4. DISCUSSIONS

We have established a biological interaction-oriented design process for whole-genome design. Unlike the bottom-up design strategies and physical-to-logical compilation, our top-down design flow and logical-to-physical compilation allow the design of synthetic biology works starting from the pathway and reaction level automatically connecting to sequence details accordingly. It can greatly improve the productivity and efficiency of synthetic biology at the whole-genome scale.

5. ACKNOWLEDGMENTS

This work is supported by grants from the National Science Council (NSC) of Taiwan (NSC 100-2319-B-010-002 and NSC 101-3113-P-110-002) to Dr. Chuan-Hsiung Chang, as well as a funding of "Aim for the Top University Plan" from the Ministry of Education, Taiwan. Affiliation: Genome Engineering Laboratory, Center for Systems and Synthetic Biology, and Institute of Biomedical Informatics, National Yang Ming University, Taipei, Taiwan, R.O.C. (TEL: 886-2-28267316).

6. REFERENCES

[1] Anderson, J.C., et al., Environmentally controlled invasion of cancer cells by engineered bacteria. J. Mol. Biol., 2006. 355: p. 619-627.

[2] Ro, D.K., et al., Production of the antimalarial drug precursor artemisinic acid in engineered yeast. Nature, 2006. 440(7086): p. 940-3.

[3] Zhang, F., S. Rodriguez, and J.D. Keasling, Metabolic engineering of microbial pathways for advanced biofuels production. Curr Opin Biotechnol, 2011. 22(6): p. 775-83.

[4] Cello, J., A.V. Paul, and E. Wimmer, Chemical synthesis of poliovirus cDNA: generation of infectious virus in the absence of natural template. Science, 2002. 297(5583): p. 1016-8.

[5] Smith, H.O., et al., Generating a synthetic genome by whole genome assembly: phiX174 bacteriophage from synthetic oligonucleotides. Proc Natl Acad Sci U S A, 2003. 100(26): p. 15440-5.

[6] Donaldson, E.F., et al., Systematic assembly of a full-length infectious clone of human coronavirus NL63. J Virol, 2008. 82(23): p. 11948-57.

[7] Gibson, D.G., et al., Complete chemical synthesis, assembly, and cloning of a Mycoplasma genitalium genome. Science, 2008. 319(5867): p. 1215-20.

[8] MacDonald, J.T., et al., Computational design approaches and tools for synthetic biology. Integrative biology : quantitative biosciences from nano to macro, 2011. 3(2): p. 97-108.

[9] Medema, M.H., et al., Computational tools for the synthetic design of biochemical pathways. Nat Rev Microbiol, 2012.

[10] Villalobos, A., et al., Gene Designer: a synthetic biology tool for constructing artificial DNA segments. BMC Bioinformatics, 2006. 7: p. 285.

[11] Cai, Y., et al., A syntactic model to design and verify synthetic genetic constructs derived from standard biological parts. Bioinformatics, 2007. 23(20): p. 2760-7.

[12] Chandran, D., F.T. Bergmann, and H.M. Sauro, TinkerCell: modular CAD tool for synthetic biology. Journal of biological engineering, 2009. 3: p. 19.

[13] Rodrigo, G., J. Carrera, and A. Jaramillo, Asmparts: assembly of biological model parts. Systems and synthetic biology, 2007. 1(4): p. 167-70.

[14] Marchisio, M.A. and J. Stelling, Computational design of synthetic gene circuits with composable parts. Bioinformatics, 2008. 24(17): p. 1903-10.

[15] Cai, Y., et al., Modeling structure-function relationships in synthetic DNA sequences using attribute grammars. PLoS computational biology, 2009. 5(10): p. e1000529.

[16] Atsumi, S., W. Higashide, and J.C. Liao, Direct photosynthetic recycling of carbon dioxide to isobutyraldehyde. Nat Biotechnol, 2009. 27(12): p. 1177-80.

Monday – June 4th
Tech. Talks Session 4 - *Topic: BioSimulators*
4BDA.1 **Dynamic Modeling of Cellular Populations within iBioSim** Jason Stevens and Chris Myers.
4BDA.2 **A Multi-Scale Model of Stem Cell Niche Formation Inside Intestine Crypts** Kai-Yuan Chen, Amit Lakhanpal, Pengcheng Bu, Steven Lipkin, Michael Elowitz and Xiling Shen.
4BDA.3 **Can Probabilistic Model Checking Explore Ribo-Nucleic Acid Folding Space?** Stefan Janssen, Loic Pauleve, Yann Ponty, Balaji Raman and Matthias Zytnicki.
4BDA.4 **A Biomolecular Implementation of Systems Described by Linear and Nonliner ODE's** Vishwesh Kulkarni, Hua Jiang, Theerachai Chanyaswad and Marc Riedel.

Dynamic Modeling of Cellular Populations within iBioSim

Jason T. Stevens
Dept. of Elec. and Comp. Eng.
University of Utah
Salt Lake City, UT 84112
j.t.stevens@utah.edu

Chris J. Myers
Dept. of Elec. and Comp. Eng.
University of Utah
Salt Lake City, UT 84112
myers@ece.utah.edu

1. INTRODUCTION

As the complexity of synthetic genetic networks increases, modeling is becoming a necessary first step to inform subsequent experimental efforts [3, 5]. In recent years, the design automation community has developed a wealth of computational tools for assisting experimentalists in designing and analyzing new genetic networks at several scales. However, existing software [2, 4, 9] is primarily catered to either the DNA- or single-cell level, with little support for the multicellular level. While the initial focus of synthetic genetic networks has been on engineering single-cell behaviors, a number of publications [1, 6] have shown the promise of multi-cellular engineering and, therefore, a need for computational tools to make this work easier. To address this need, the `iBioSim` software package [7] has been enhanced to provide support for modeling, simulating, and visualizing coarse-grained, dynamic cellular populations in a two-dimensional space. This capacity is fully integrated into the software, capitalizing on `iBioSim`'s strengths in modeling, simulating, and analyzing single-celled systems.

2. SPATIAL MODELING

All of the population-based enhancements to `iBioSim` rely on a spatial modeling framework. This framework is grid-based, with a single compartment (e.g., a cell) allowed at each grid location. This creates spatial separation between grid locations and is thus a basis for modeling spatial diffusion. Species at every two adjacent grid locations can be connected via a diffusion reaction. To do this, users can mark species within compartments as diffusible then specify kinetic rate law parameters to apply to those species. The parameters are then used to automatically generate spatial diffusion reactions across the entire grid. As these diffusible species begin within compartments—and must move outside of the components in order to spatially diffuse—each grid location can contain an "intracellular" and "extracellular" space. This allows for membrane separation of chemical species and thus provides a basis for membrane diffusion. Both spaces (intra- and extracellular) are modeled as distinct, well-mixed containers, with connections possible via membrane diffusion reactions. As with the diffusion reactions between grid locations, the membrane diffusion reactions are created automatically across the entire grid using user-specified rate parameters. With both kinds of diffusion reactions, species can diffuse out of a cell, move across the extracellular space, and then diffuse into a different cell. This capability provides a user-friendly way of modeling cellular communication mechanisms.

When the counts of extracellular grid species grow large, the propensities of the corresponding diffusion reactions during simulation grow large as well, resulting in a simulation bottleneck. Indeed, up to ninety-nine percent of the reactions being fired can be grid diffusion reactions. To address this, our tool utilizes stoichiometry amplification, which allows users to group diffusion reactions. For instance, if a stoichiometry amplification value of five is chosen, extracellular grid reactions move five species per reaction, and this reaction's propensity is multiplied by one-fifth. So the reaction occurs one-fifth as frequently, but it moves five times the species. This speeds up simulation time significantly (roughly equivalent to the amplification value) for models with large quantities of diffusible species without appreciable macroscopic differences in the simulation outcome.

3. DYNAMIC MODELING

While static spatial modeling and diffusion can enable the creation of models for many interesting applications, the addition of dynamic processes, namely, cell duplication and death, enables modeling of important phenomena such as population control and artificial developmental programs (e.g., cells apoptosing to reveal a pattern). To provide this capacity, `iBioSim` supports new dynamic process events, which can be added to compartment models. When these events trigger during simulation, the corresponding dynamic process is also triggered.

A death event removes all traces of whichever compartment the event was triggered for, meaning that reactions or events relevant to this compartment can no longer fire. A duplication event creates a new copy of the compartment, with species apportioned to the parent and child compartments via assignments associated with the duplication event (the child gets whatever is left over after the assignments to the parent take place, in order to conserve species counts). For visualization purposes, the locations of compartments within the grid are tracked, and new child compartments are placed in a random neighboring location to the parent, with existing compartments shifted out of the way. If the new child compartment is placed in a location outside of the current grid bounds, or if a shifted cell is shifted outside of the current grid bounds, the grid automatically expands, creating new grid diffusion reactions and grid species for the new locations.

At the modeling level, we are in the process of introducing the concept of dynamic arrays into the *Systems Biology Markup Language* (SBML) so that dynamic models can be represented. Every SBML element on the grid is represented as an arrayed quantity in the model, and dynamic events can then use these arrays to adjust their number. Using arrays is also useful in a static modeling context, as the size of the files can be reduced dramatically. Currently, arrays are implemented using annotations, but we hope to integrate this into an SBML package in the future.

To enable dynamic events during simulation time, our new stochastic simulator uses dynamic data structures so that reactions and species can be added and removed easily. To improve performance, this simulator incorporates a faster Gillespie SSA algorithm for handling the large number of reactions inherent in multicellular models, namely, the composition and rejection method [8]. The composition and rejection method creates and maintains groups of reactions according to their propensities during runtime. To choose a reaction, the algorithm randomly chooses a group, then randomly chooses a reaction and a propensity. If the propensity is less than the chosen reaction's propensity, that reaction is chosen, otherwise a new reaction and propensity are chosen within the same group until the process finds a reaction to fire. Our experience indicates that this algorithm is faster and scales better than the Gillespie SSA Direct method augmented with a dependecy graph.

4. ANALYSIS

`iBioSim`'s visualization environment has been enhanced for both static and dynamic models. Users can see the model change over time by playing back simulation data and associating appearances with species counts. Appearances can be associated with species within compartments (which change the appearance of the compartment itself) and in the extracellular space to visualize diffusible species moving around the grid. For dynamic models, child compartments inherit the appearance of the parent, and grid appearances are extended as the grid expands, making it easy to visualize a population as it grows. Figure 1 shows this process occurring over the timecourse of a dynamic model. Furthermore, statistics for the time-series data files are generated for dynamic models, which can be used with the graphing functionality in `iBioSim` to further analyze the system, e.g., in aggregate over multiple simulation runs.

5. ACKNOWLEDGMENTS

Thanks to Tyler Patterson for developing the initial visualization engine, and to Curtis Madsen and Nicholas Roehner for helpful and constructive discussion during the project's development. This research is supported by the National Science Foundation under Grant CCF-0916042.

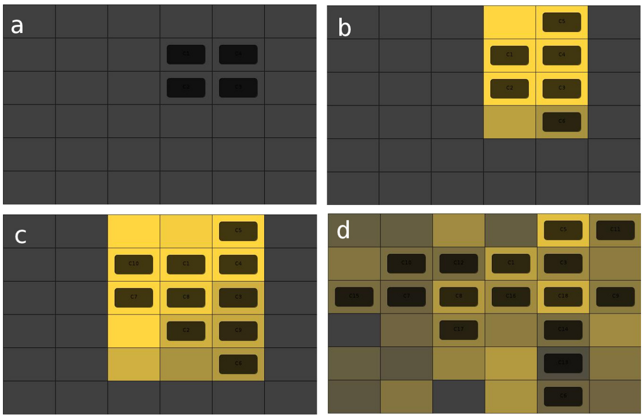


Figure 1: Shows four separate time points (a being first, d being last) in a dynamic simulation. The dark shapes on the grid represent cells, the population of which grows throughout the simulation. Additionally, the different colors of the grid squares themselves shows the diffusion of chemical species in a spatial manner.

6. REFERENCES

[1] S. Basu, Y. Gerchman, C. H. Collins, F. H. Arnold, and R. Weiss. A synthetic multicellular system for programmed pattern formation. *Nature*, 434(7037):1130–1134, Apr 2005.

[2] Y. Cai, B. Hartnett, C. Gustafsson, and J. Peccoud. A syntactic model to design and verify synthetic genetic constructs derived from standard biological parts. *Bioinformatics*, 23(20):2760–2767, Oct 2007.

[3] J. M. Carothers, J. A. Goler, D. Juminaga, and J. D. Keasling. Model-driven engineering of RNA devices to quantitatively program gene expression. *Science*, 334(6063):1716–1719, Dec 2011.

[4] D. Chandran, F. T. Bergmann, and H. M. Sauro. TinkerCell: modular CAD tool for synthetic biology. *J Biol Eng*, 3:19, 2009.

[5] T. Ellis, X. Wang, and J. J. Collins. Diversity-based, model-guided construction of synthetic gene networks with predicted functions. *Nat. Biotechnol.*, 27(5):465–471, May 2009.

[6] C. Liu, X. Fu, L. Liu, X. Ren, C. K. Chau, S. Li, L. Xiang, H. Zeng, G. Chen, L. H. Tang, P. Lenz, X. Cui, W. Huang, T. Hwa, and J. D. Huang. Sequential establishment of stripe patterns in an expanding cell population. *Science*, 334(6053):238–241, Oct 2011.

[7] C. J. Myers, N. Barker, K. Jones, H. Kuwahara, C. Madsen, and N.-P. D. Nguyen. iBioSim: a tool for the analysis and design of genetic circuits. 25(21):2848–2849, 2009.

[8] A. Slepoy, A. P. Thompson, and S. J. Plimpton. A constant-time kinetic Monte Carlo algorithm for simulation of large biochemical reaction networks. *J Chem Phys*, 128(20):205101, May 2008.

[9] E. Weeding, J. Houle, and Y. N. Kaznessis. SynBioSS designer: a web-based tool for the automated generation of kinetic models for synthetic biological constructs. *Brief. Bioinformatics*, 11(4):394–402, Jul 2010.

A multi-scale model of stem cell niche formation inside intestine crypts

Kai-Yuan Chen
School of Electrical and Computer Engineering, Cornell University
Room 302, Philips Hall, Cornell University, Ithaca, NY, 14850
Telephone: +1 607-793-0198
kc632@cornell.edu

Amit Lakhanpal
Howard Hughes Medical Institute
Department of Biology, Bioengineering, and Applied Physics, California Institute of Technology
amitl@caltech.edu

Pengcheng Bu
Department of Biomedical Engineering, Cornell University
pb345@cornell.edu

Steven Lipkin
Departments of Medicine and Genetic Medicine, Weill Cornell College of Medicine
stl2012@med.cornell.edu

Michael Elowitz
Howard Hughes Medical Institute
Department of Biology, Bioengineering, and Applied Physics, California Institute of Technology
melowitz@caltech.edu

Xiling Shen
School of Electrical and Computer Engineering, Department of Biomedical Engineering, Cornell University
Room 411, Philips Hall, Cornell University, Ithaca, NY, 14850
Telephone: +1 607-254-8550
xs66@cornell.edu

ABSTRACT

Intestinal crypts regulate homeostasis by creating a stem cell niche at the base, where stem cells and niche cells form a soccer-ball-like pattern. Divided cells migrate out of the niche and differentiate into a different pattern at the top. However, the mechanisms behind stem cell niche formation remain unclear. Here we built a multi-scale, physical based 3D model that includes intercellular interactions and intracellular signaling to investigate the niche formation. Our model shows that a Notch-dependent circuit forms a bistable latch to create the stem cell niche at the bottom of the crypt. The disruption of this circuit could prevent stem cell niche formation and cause dysplasia and tumor formation.

Keywords

Multi-scale modeling, intestinal crypt, stem cells, physically based modeling, autonomous cellular system modeling.

1. INTRODUCTION

The small intestine and colon are lined with a single layer of epithelium cells. The epithelium is full of crypts, which are invaginations into the underlying connective tissue. The intestinal epithelium is replaced every 3-5 days, making it the fastest regenerative tissue in the body. To maintain homeostasis, stem cells are tightly controlled by a niche at the bottom of the crypt. In side the niche, 12~14 Lgr5+ stem cells form soccer-ball-like pattern with CD24+ Paneth (niche) cells [1]. Divided cells leave the niche and migrate up while differentiating into absorptive (enterocyte) and secretory (Goblet) lineages, eventually forming more random cell fate patterns at the top (Figure 1). Disruption to various signaling mechanisms such as Wnt and Notch can perturb the homeostasis of intestine crypts and affect tumor progression [2]. However, it remains unclear how local cell interaction mechanisms give rise to the robust homeostasis of crypts patterns at intestine crypts.

2. MATERIALS AND METHOD

To understand how the regular stem cell niche pattern forms inside intestine crypts, we constructed a multi-scale 3D model with cell automata. This model incorporates a physically based multicellular model with subcellular signaling networks to study the regulation on niche formation with extrinsic environmental interaction and locally intrinsic signaling transduction. The model is developed in Java with OpenGL library for 3D computation. Each cell is treated as a soft body with perfect spherical structure. Compression energy, deformation energy, and adhesion energy are applied to simulate cell behaviors with regulations by cell-cell interaction and cell-matrix interaction under stochastic condition. Constrain energy was applied to direct the cell migration only along the membrane surface of intestine crypts from bottom of top. Intercellular Notch signaling circuit is simulated by Ordinary Differential Equations (ODEs). Each cell inherits same cellular properties. Cells can migrate, divide, and sense external microenvironment. The cell fates are programed by cell-cell interaction and cell-matrix interaction, thus the cells can grow, stop growing, and eventually go to apoptosis.

3. RESULT

Notch signaling depends on ligands on a cell activating receptors on a neighboring cell. Stem cells and enterocytes express high levels of Notch receptors while Paneth and Goblet cells express high levels of Notch ligands, suggesting that Notch plays a role in pattern formation [3-4].

3.1 3D Crypt Model

We first build a 3D multi-cellular model to simulate the cell organization of intestine crypt, which is shown in Figure 2. The structure of crypt is constructed with a 3D mesh to mimic the surface of physiological crypt membrane. Intestinal cells attach to the membrane surface, and moved along the vertical axis from bottom to top of the crypt. The Notch signaling level is represented in color saturation.

3.2 Pattern Formation of Intestine Crypts

The simulation of niche formation is shown in Figure 3. The center region in Figure 3 represents the bottom of the crypt, and the peripheral region represents the top of the crypt. Simulation shows that Notch high and Notch low cells form soccer-ball-like pattern in the center region. Through the regulation of microenvironment, Notch signaling shows regular pattern at the bottom of the crypt, and more random pattern at the top of the crypt in much lower level. This simulation result is consistent with published experimental evidence [1]. Our analysis further reveals that the stem cell niche pattern would be broken down and the cells will be switched to more proliferative cell fates when the Notch-dependent circuit is disrupted. This can ultimately lead to tumorgenesis.

3.3 Systems Dynamic Analysis

Here, we ask a question: Why can the circuit generate soccer-ball-like pattern robustly? DLL in one cell activates Notch signaling in the adjacent cell with direct ligand/receptor binding. The triggered Notch signal subsequently suppresses the DLL level in the same cell. From electrical engineering point of view, this Notch intercellular signaling circuit consists of a double negative feedback loop, which has similar circuit architecture to the design principle of a fundamental electrical circuit, latch, as shown in Figure 4a. To understand the system dynamic of this biological latch circuit, we analyzed its stability. In Figure 4b, the bifurcation diagram reveals that the intercellular circuit is a bistable switch with hysteresis. It has two stable levels and one unstable steady state between the two stable states. Thus, Notch levels in adjacent cells would be either elevated or repressed to two distinct steady levels robustly through the regulation of this intercellular latch circuit, which causes the soccer-ball pattern.

4. DISCUSSION

In this study, we constructed an autonomous multi-scale cell model to simulate the niche formation of intestine crypts. Our analysis reveals that Notch forms a bistable latch that can robustly generate stem cell niche pattern. The dysfunction of local Notch signaling could disrupt the homeostasis of crypts and trigger dysplasia and tumor formation.

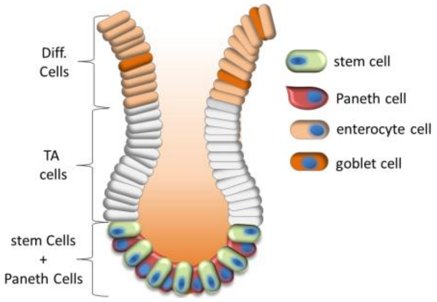


Figure 1: Cell organization of intestine crypt

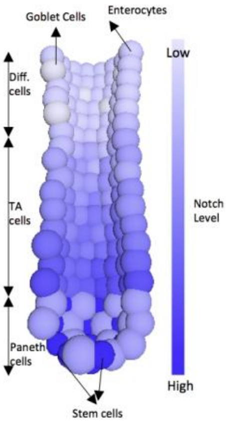


Figure 2: 3D crypt model

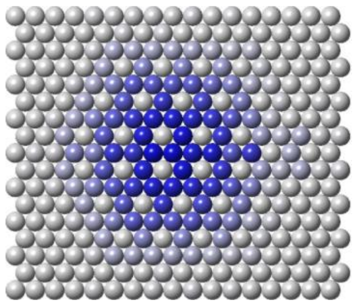


Figure 3: Niche formation through the Notch-dependent circuit.

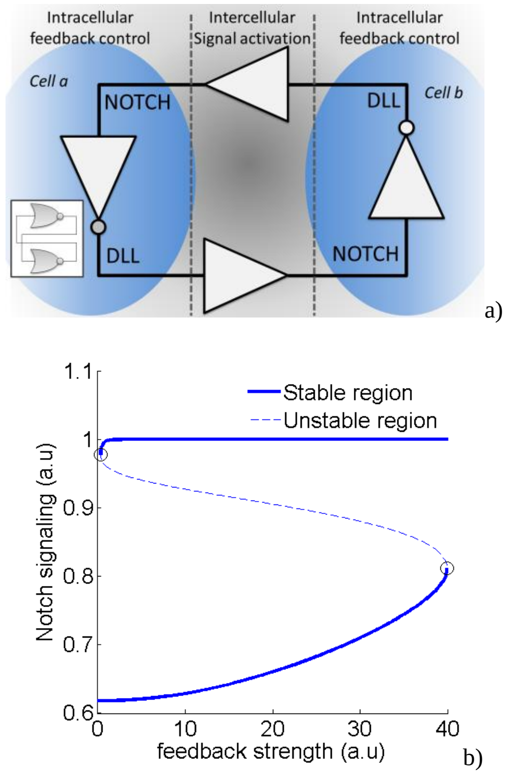


Figure 4: a) Notch intercellular feedback loop forms latch circuit. b) Notch signaling shows hysteresis.

5. REFERENCES

[1] Snippert HJ et al. (2010) Intestinal crypt homeostasis results from neutral competition between symmetrically dividing Lgr5 stem cells. *Cell* **143**, 134-44.

[2] Ranganathan P et al (2011). Notch signalling in solid tumours: a little bit of everything but not all the time. *Nature Reviews Cancer*.

[3] Crosnier C et al. (2006) Organizing cell renewal in the intestine: stem cells, signals and combinatorial control. *Nat Rev Genet* **7**, 349-59

[4] Sprinzak D, et al. (2011) Mutual inactivation of Notch receptors and ligands facilitates developmental patterning. *PLoS Comput Biol 7*

Can Probabilistic Model Checking Explore Ribo-Nucleic Acid Folding Space?

Stefan Janssen[1], Loic Pauleve[2], Yann Ponty[2], Balaji Raman[3], Matthias Zytnicki[4]
[1]Universitat Bielefed, Germany, [2] Ecole Polytechnique, Palaiseau, France,
[3]Verimag, Gieres, France, [4]Institut National De La Recherche Agrnomique, Versailles, France.
balaji.raman@imag.fr

ABSTRACT

The folding path of a ribo-nucleic acid from its free form to its complete structure is of significant interest to structural biologists. There are algorithms for simulating kinetics of the ribo-nucleic acid from its unfolded state to its stable structure (minimum free-energy structure). These computational techniques are expensive as the total number of structures in the folding space is exponentially large. We propose to use time-efficient algorithms from the field of probabilistic model checking for verifying certain hypothesis concerning ribo-nucleic acid folding landscape.

First, we explain how thermodynamic models of ribo-nucleic acid can be used to generate a Markov chain of structures, whose transitions within the chain are based on the energy difference w.r.t to their neighbors. Then we present a process algebra model to compactly represent the dynamics of ribo-nucleic acid folding. Both these approaches essentially generate input models for probabilistic model checking. Finally, we discuss if statistical model checking techniques can be applied for the problem of aligning ribo-nucleic acid sequences and structures.

1. MARKOV CHAIN: RNA STRUCTURES

In this work, we focus on secondary structure of the Ribo-Nucleic Acid(RNA). The RNA will reach a stable structure from the unfolded state after passing several intermediate structures. A question on this RNA folding process within the framework of model checking is: within how many steps a certain state is reached? (Similar approach had been applied for protein folding [3]). In what follows, we describe how we extract a Markov chain of secondary structures of RNA from the thermodynamic models. Essentially, we use energy value that the thermodynamic models estimate for each possible secondary structure in the folding space. The energy differences among the structures determine the neighboring states in the Markov chain and the transition probabilities between states.

Consider a Markov chain $\mathcal{M}$ with the set of states $\mathcal{S}$. Each state in the Markov chain corresponds to a secondary structure of the RNA. π is some probability distribution on the states. $P_{s \to s'}$ is the transition probability function of $\mathcal{M}$. Then the **detailed balance condition** requires that

$$\pi(s) \cdot P_{s \to s'} = \pi(s') \cdot P_{s' \to s}, \quad \forall s, s' \in \mathcal{S}.$$

The above condition alone is sufficient (although not necessary) to ensure convergence of the Markov chain towards a stationary distribution π. Now let us see how the Markov chain converges toward the Boltzmann distribution. In our case, the limit of the process of folding is the **Boltzmann equilibrium**, where one has

$$\pi(s) = \frac{e^{-E_s/RT}}{\mathcal{Z}} \quad \forall s \in \mathcal{S},$$

where E_s is the free-energy (e.g. obtained by running `RNAeval`), T is the temperature (Kelvins) and R is the perfect gas constant (kCal·K^{-1}·mol^{-1}, 0.0019858775).

It follows that

$$\begin{aligned} \pi(s) \cdot P_{s \to s'} &= \pi(s') \cdot P_{s' \to s} \\ \Rightarrow \frac{P_{s \to s'}}{P_{s' \to s}} &= \frac{\pi(s')}{\pi(s)} \\ &= e^{-(E_{s'} - E_s)/RT} \end{aligned}$$

The transition probabilities between two states are then computed as follows:

$$P_{s \to s'} := \begin{cases} 0 & \text{If } s \neq s' \text{ and } s, s' \text{ are } \textbf{not} \text{ neighbors} \\ \dfrac{e^{-\frac{E_{s'} - E_s}{2RT}}}{K} & \text{If } s \neq s' \text{ and } s, s' \text{ are neighbors} \\ 1 - \displaystyle\sum_{s'' \neq s \in \mathcal{S}} P_{s \to s''} & \text{If } s = s' \end{cases} \tag{1}$$

where

$$K := \max_{s \in \mathcal{S}} \left(\sum_{s' \neq s \in \mathcal{S}} e^{-\frac{E_{s'} - E_s}{2RT}} \right). \tag{2}$$

We developed a software for generating the Markov chain and used PRISM tool to verify some properties. We experimented with some examples of RNA sequences.

2. A STOCHASTIC PROCESS ALGEBRA TO MODEL RNA FOLDING

With the aim at providing a compact modelling of RNA folding dynamics, we propose a process algebra inspired by the stochastic π-calculus [5] and Bioambient [6]. Basically, a nucleotide is modelled as a process which can bind to another to form a pair. The formed pairs are differentiated by the channels on which the binding occur. A base pair is identified by a unique channel on which the unbinding can be triggered by the bound nucleotides. Processes enclosed by a base pair are isolated from the others with ambients, ensuring valid secondary structures.

Def. 1-3 give the syntax, congruence and reduction rules of the processes, which can then be used to derive a Continuous Time Markov Chain(CTMC) semantics. Def. 4 instantiates this calculus for the RNA folding, accounting for AU, GC, and GU base pairing. Because we do not allow process reordering, relative processes distance could be use to balance the action rates. Fig. 1 illustrates the obtained folding of the example sequence.

Future work will investigate the impact of various rate definitions, mixing the different strengths of RNA base pairs and the distance between nucleotides, w.r.t. obtained equilibrium. The stochastic simulation of our calculus could be instantiated from the generic abstract machine proposed in [2] to serve as input for statistical model checking techniques, which may be necessary to make tractable the analysis of large sequence folding. Finally, this process algebra approach opens the way to improve probabilistic model checking performance by exploiting the compositionality of the framework (identical ambients behave equivalently) and by using static analysis techniques to drive the analysis w.r.t. model structure.

3. RNA ALIGNMENT

In the past two sections, we presented the application of model checking for RNA folding kinetics problem. In this section, we briefly discuss possibility of using statistical model checking for RNA sequence and structural alignment. Here the question is: how well a given query sequence matches (in terms of sequence and secondary structure) with a model, which is derived from a set of homologous RNAs? With the focus on sequence content, Infernal [4] computes the most probable alignment in $O(n^4)$ time.

Shifting to structure, we ask for the most probable secondary structure of the query when compared to the model. This problem is considered to be similar to *path labeling problem*[1], which is shown to be NP hard. Thus there is an opportunity for model checking techniques to be used for RNA alignment problem too. Unlike the approach we discussed in the previous sub-sections, it seems a Markov chain cannot be generated for this particular problem. However, we are planning to use statistical model checking, which is not tied to any standard model, for determining the most probable structure common to the input sequence and the alignment model.

4. REFERENCES

[1] B. Brejová, D. G. Brown, and T. Vinař. The most probable annotation problem in hmms and its application to bioinformatics. *J. Comput. Syst. Sci.*, 73(7):1060–1077, Nov. 2007.

[2] M. R. Lakin, L. Paulevé, and A. Phillips. Stochastic simulation of multiple process calculi for biology. *Theoretical Computer Science*, in press, 2012.

[3] C. J. Langmead and S. K. Jha. Predicting Protein Folding Kinetics via Temporal Logic Model Checking. Technical report, Computer Science Department, Carniege Mellon, 2007.

[4] E. P. Nawrocki, D. L. Kolbe, and S. R. Eddy. Infernal 1.0: inference of rna alignments. *Bioinformatics*, 25(10):1335–1337, 2009.

[5] C. Priami. Stochastic π-Calculus. *The Computer Journal*, 38(7):578–589, 1995.

[6] A. Regev, E. M. Panina, W. Silverman, L. Cardelli, and E. Y. Shapiro. Bioambients: an abstraction for biological compartments. *Theor. Comput. Sci.*, 325(1):141–167, 2004.

$$\begin{array}{rcl}
P, Q ::= & C \mid X(n) \mid P|Q \mid \nu x\, P \mid \boxed{P} & \text{Process} \\
C ::= & \pi_1^{i_1}.P_1 + \ldots + \pi_N^{i_N}.P_N & \text{Choice} \\
E ::= & X_1(n_1) \mapsto P_1, \ldots, X_N(n_N) \mapsto P_N & \text{Environnment} \\
\pi ::= & \gamma!x(n) \mid \gamma?x(n) & \text{Action} \\
\gamma ::= & \mathtt{bind} \mid \mathtt{unbind} & \text{Action type}
\end{array}$$

Def 1. Syntax of the process algebra for RNA folding. A process is either a choice between actions, a definition instance, a parallel composition of processes, a channel x restriction, or an ambient. An action is either a sending (!) or receiving (?) of a channel n upon a channel x with a type `bind` or `unbind`.

$$\mathbf{0} \mid P \equiv P \mid \mathbf{0} \equiv P \qquad \nu x\, \mathbf{0} \equiv \mathbf{0} \qquad \nu x\, \nu y\, P \equiv \nu y\, \nu x\, P$$

$$\begin{array}{rcl}
P_1 \mid (P_2 \mid P_3) & \equiv & (P_1 \mid P_2) \mid P_3 \\
\nu x\, (P_1 \mid P_2) & \equiv & P_1 \mid \nu x\, P_2 \ \mathbf{if}\ x \notin \mathrm{fn}(P_1) \\
X(n) & \equiv & P_{\{n/m\}} \ \mathbf{if}\ E(X(m)) = P
\end{array}$$

Def 2. Structural congruence of processes. This equivalence relation assumes a global environment E and that processes are equal up to renaming of channels and reordering of actions in a choice. **No process reordering allowed.** fn(P) denotes the channels that are not restricted within P; $\mathbf{0}$ is the empty choice.

$$\mathtt{bind}!x(n)^i.P + C \mid Q \mid \mathtt{bind}?x(m)^{i'}.P' + C'$$
$$\xrightarrow{\mathrm{rate}(x,\mathtt{bind},i,i')} \boxed{P \mid Q \mid P'_{\{n/m\}}}$$

$$\mathtt{bind}?x(n)^i.P + C \mid Q \mid \mathtt{bind}!x(m)^{i'}.P' + C'$$
$$\xrightarrow{\mathrm{rate}(x,\mathtt{bind},i,i')} \boxed{P_{\{m/n\}} \mid Q \mid P'}$$

$$\boxed{\mathtt{unbind}!x(n)^i.P + C \mid Q \mid \mathtt{unbind}?x(m)^{i'}.P' + C'}$$
$$\xrightarrow{\mathrm{rate}(x,\mathtt{unbind},i,i')} P \mid Q \mid P'_{\{n/m\}}$$

$$\boxed{\mathtt{unbind}?x(n)^i.P + C \mid Q \mid \mathtt{unbind}!x(m)^{i'}.P' + C'}$$
$$\xrightarrow{\mathrm{rate}(x,\mathtt{unbind},i,i')} P_{\{m/n\}} \mid Q \mid P'$$

$$P \xrightarrow{r} P' \Rightarrow \begin{cases} \boxed{P} \xrightarrow{r} \boxed{P'} \\ \nu x\, P \xrightarrow{r} \nu x\, P' \\ Q_1 \mid P \mid Q_2 \xrightarrow{r} Q_1 \mid P' \mid Q_2 \end{cases}$$

$$Q \equiv P \xrightarrow{r} P' \equiv Q' \Rightarrow Q \xrightarrow{r} Q'$$

Def 3. Reduction rules of processes. $P_{\{n/m\}}$ renames channel m to n. rate(x, γ, i, i') associates the rate of channel x with type γ, which can be balanced by the distance between the processes, extracted from action indexes i, i'.

$$\begin{array}{rcl}
A & ::= & \nu x\, \mathtt{bind}!au(x).Ab(x) \\
G & ::= & \nu y\, \mathtt{bind}!gc(y).Gb(y) + \nu z\, \mathtt{bind}!gu(z).Gb(z) \\
C & ::= & \mathtt{bind}?gc(x).Cb(x) \\
U & ::= & \mathtt{bind}?au(x).Ub(x) + \mathtt{bind}?gu(x).Ub(x)
\end{array}$$

$$Ab(x) ::= \mathtt{unbind}!x.A \qquad Cb(x) ::= \mathtt{unbind}?x.C$$
$$Gb(x) ::= \mathtt{unbind}!x.G \qquad Ub(x) ::= \mathtt{unbind}?x.U$$

Def 4. Encoding of RNA base pairing. Rate of x, y, and z are rates of au, gc and gu channels, respectively.

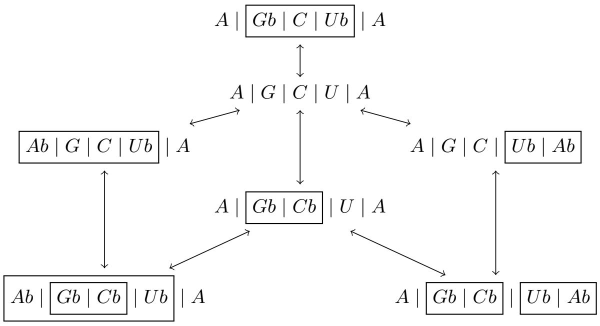


Figure 1: Underlying CTMC of the process $\nu au\, \nu gc\, \nu gu\, (A|G|C|U|A)$ (rates are not displayed).

A Biomolecular Implementation of Systems Described by Linear and Nonlinear ODE's

[Extended Abstract] *

Vishwesh V. Kulkarni†
University of Minnesota
Department of Electrical Engineering
Minnesota, MN 55455, USA
vkulkarn@umn.edu

Hua Jiang
University of Minnesota
Department of Electrical Engineering
Minnesota, MN 55455, USA
hua@umn.edu

Theerachai Chanyaswad
University of Minnesota
Department of Electrical Engineering
Minnesota, MN 55455, USA
chany001@umn.edu

Marc Riedel
University of Minnesota
Department of Electrical Engineering
Minnesota, MN 55455, USA
mriedel@umn.edu

ABSTRACT

Building on the recent work of Oishi and Klavins, we present new results on implementing linear time-invariant systems using biomolecular reactions. We then extend this framework to cover nonlinear dynamical systems and also present the DNA strand displacement implementations.

Categories and Subject Descriptors

J.3 [**Computer Applications**]: Life and Medical Sciences—*biology and genetics*

General Terms

Theory

Keywords

biomolecular reactions, DNA strand displacement, linear systems, nonlinear systems

1. INTRODUCTION

Systems synthesised *in vitro* from DNA are becoming substantially more complex and reliable due to well-understood models of hybridization and strand displacement that faciliate the means to design and predict molecular interactions (see [6, 4], and [5]). Any abstract chemical reaction that can be realized physically can now be well-approximated using a set of DNA strand displacement reactions [5]. Observing this, Oishi and Klavins have recently shown how a class of *linear time-invariant* (LTI) systems can be implemented using biochemical reactions and, in particular, using DNA strand displacement reactions (see [3]). In [3], after proving that an LTI system can be built using three types of reactions, viz., catalysis, degradation, and annihilation, a set of chemical reactions is proposed to implement a transfer function of the form $T(s) = \alpha/D(s)$ where α is a scalar and $D(s)$ is a polynomial in s. The construction can be attempted in a modular format since a method to implement the basic building blocks — such as a constant gain, an integrator, an adder, and an signal replicator — using biochemical reactions is also given in [3]. In this paper, we extend this framework to cover the entire class of LTI systems and, in addition, a class of nonlinear dynamical systems.

2. PROBLEM FORMULATION

Problem 1. *Obtain a set of biomolecular reactions that represents any given LTI system, and implement it using DNA strand displacement.* □

Note that an arbitrary *single-input single-output* (SISO) LTI system can be described by a transfer function $T(s)$ given by $T(s) = N(s)/D(s)$, where $N(s)$ and $D(s)$ are polynomials in s; for physically realizable systems, the degree of $N(s)$ is at most equal to the degree of $D(s)$. Such a system can be implemented by interconnecting building blocks such as constant gains, integrators, adders, and signal replicators (see [2, Ch. 2.1]). For example, suppose

$$N(s) = b_1 s^2 + b_2 s + b_3, \quad D(s) = s^3 + a_1 s^2 + a_2 s + a_3, \quad (1)$$

where b_i and a_i are constant gains. Then the LTI system with the transfer function $T(s) = N(s)/D(s)$ can be im-

*This abstract is submitted for an inclusion both poster and oral presentation categories. This research is supported, in part, by the NSF CAREER Award 0845650 and by NSF BIO Computing.

†Corresponding Author.

IWBDA2012 June 2012, San Francisco, CA, USA

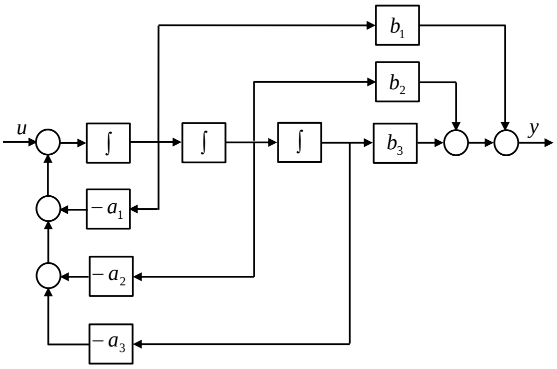


Figure 1: A physical realization of the LTI system $H : u \mapsto y$ defined by the transfer function given by (1). The circles denote the summation junctions.

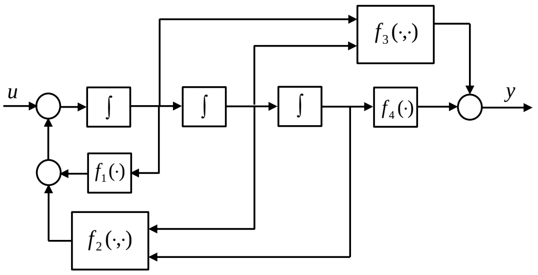


Figure 2: A sample block diagram illustrating a physical realization of the class of nonlinear systems of interest to us. The nonlinearities f_1 and f_4 can be Hill-type or power nonlinearities whereas $f_2(x,y) \doteq xy$ and $f_3(x,y) \doteq f_i(x)f_j(y)$ where $(i,j) \in \{(1,1),(1,4),(4,1),(4,4)\}$.

plemented using these basic building blocks as shown in Fig. 1 (see [2, Ch. 2.1]). The class of nonlinear dynamical systems of interest to us includes product nonlinearities ($f(x,y) = xy$), power nonlinearities ($f(x) = x^n$), and Hill-type nonlinearities. Such nonlinearities are commonly observed in wild type biological networks, and can be represented by a block diagram of the form shown in Fig. 2. Let us refer to this class of systems as $\mathcal{S}_N$.

Problem 2. *Obtain a set of biomolecular reactions that represents a given $\mathcal{S}_N$ system.* □

3. MAIN RESULT

Following [3], we represent a signal u using two components u^+ and u^-, where

$$u^+ = \begin{cases} u & \text{if } u \geq 0; \\ 0 & \text{else.} \end{cases} \quad \text{and} \quad u^- = \begin{cases} u & \text{if } u < 0; \\ 0 & \text{else.} \end{cases}$$

Then, the LTI system mapping u into y can be represented using the following three types of reactions [3]:

$$\textbf{catalysis:}\ u \to u + y, \tag{2}$$

$$\textbf{degradation:}\ u \to \phi, \tag{3}$$

$$\textbf{annihilation:}\ u^+ + u^- \to \phi. \tag{4}$$

Some of our results are as follows.

Lemma 1. *Let $w \doteq x^n$, where n is an integer. This nonlinearity can be realized using the following set of biomolecular reactions: $nx \xrightarrow{k} nz$, $nz \xrightarrow{k} w + (n-1)a$, $w \xrightarrow{k} b$.* □

Proof. The differential equations governing this set of biomolecular reactions are as follows: $\dot{w} = kz^n - kw$, $\dot{z} = nkx^n - nkz^n$. At the steady state, $\dot{w} = kz^n - kw = 0$ whence $w = z^n$. In addition,if x is a constant input then $nx \xrightarrow{k} nz$ implies that $z = x$ at the steady state as well. Hence the proof. QED. □

Likewise, the following results are proved.

Lemma 2. *Let $w \doteq x_1x_2$. This nonlinearity can be realized using the following set of biomolecular reactions: $x_1 \xrightarrow{k} z_1$, $x_2 \xrightarrow{k} z_2$, $z_1 + z_2 \xrightarrow{k} w + y$, $w \xrightarrow{k} y$.* □

Lemma 3. *Consider the system described by the following nonlinear ordinary differential equations: $\dot{x} = -k_7x + \frac{k_1k_3y}{k_1+k_2y}$, $\dot{y} = -k_3y + \frac{k_5k_7x}{1+x}$. This system can be realized using the following set of biomolecular reactions: $z \xrightarrow{k_1} x$, $z + y \xrightarrow{k_2} A + a$, $y \xrightarrow{k_3} z$, $A + y \xrightarrow{k_4} 2y$, $w \xrightarrow{k_5} y$, $w + x \xrightarrow{k_6} D + d$, $x \xrightarrow{k_7} w$, $D + x \xrightarrow{k_8} 2x$.* □

4. DISCUSSION

Our implementation of the pure integrator block differs from the one given in [3] and makes use of an impulse function approximated by a pulse function through a time-delay block. A discrete-time implementation of the time delay is already given in [1]. We implement these biomolecular reactions using the DNA strand displacement technique described in [5].

5. REFERENCES

[1] H. Jiang, M. Riedel, and K. Parhi. Digital signal processing with molecular reactions. *IEEE Design and Test of Computers: Special Section on Bio-Design Automation in Synthetic Biology*, May/June 2012.

[2] T. Kailath. *Linear Systems.* Prentice Hall, Englewood Cliffs, N.J., 1980.

[3] K. Oishi and E. Klavins. Biomolecular implementation of linear i/o systems. *IET Systems Biology*, 5(4):252–260, 2011.

[4] R. Schulman and E. Winfree. Programmable control of nucleation for algorithmic selfassembly. *DNA Computing*, 3384:319–328, 2005.

[5] D. Soloveichik, G. Seelig, and E. Winfree. DNA as a universal substrate for chemical kinetics. *Proc. Natl. Acad. Sci.*, 107(12):5393–5398, 2010.

[6] D. Zhang and E. Winfree. Control of DNA strand displacement kinetics using toehold exchange. *J. Am. Chem. Soc.*, 131(47):17303–17314, 2009.

Monday – June 4th

Invited Talk: **Jasmin Fisher, Microsoft UK**

"From Coding the Genome to Algorithms Decoding Life"

The decade of genomic revolution following the human genome's sequencing has produced significant medical advances, and yet again, revealed how complicated human biology is, and how much more remains to be understood. Biology is an extraordinary complicated puzzle; we may know some of its pieces but have no clue how they are assembled to orchestrate the symphony of life, which renders the comprehension and analysis of living systems a major challenge. Recent efforts to create executable models of complex biological phenomena - an approach we call Executable Biology - entail great promise for new scientific discoveries, shading new light on the puzzle of life. At the same time, this new wave of the future forces computer science to stretch far and beyond, and in ways never considered before, in order to deal with the enormous complexity observed in biology. This talk will focus on our recent success stories in using formal methods to model cell fate decisions during development and cancer, and on-going efforts to develop dedicated tools for biologists to model cellular processes in a visual-friendly way.

Jasmin Fisher received her PhD in Neuroimmunology from the Weizmann Institute of Science in Israel. She started her work on the application of formal methods to biology as a postdoctoral fellow in the department of Computer Science at the Weizmann Institute (2003-2004), where she worked with David Harel, and then continued to work on the development of novel formalisms and tools tailored for modelling biological processes as a postdoctoral researcher in the School of Computer Science at the EPFL in Switzerland (2004-2007), together with Tom Henzinger. In 2007, Jasmin moved to Cambridge to join Microsoft Research, and since 2009 she is also a Lecturer at Cambridge University. Jasmin is one of the founders of the field of Executable Biology and a leader in the area of formal methods in biology. Over the past decade, Jasmin has been pioneering the study on usage of program analysis techniques for the analysis of biological models. Her research focuses on the construction and analysis of executable models that mimic aspects of biological phenomena in order to better understand complex biological systems. She is mainly interested in processes of cell fate determination and signalling networks operating during normal development and cancer.

Monday – June 4th
Tech. Talks Session 5 - *Topic: Biological Circuit Design and Assembly I*
5BDA.1 ***In Silico* Design of Functional DNA Constructs Based on Heuristic Data** Claes Gustafsson, Alan Villalobos, Mark Welch and Jeremy Minshull.
5BDA.2 **j5 and DeviceEditor: DNA Assembly Design Automation** Joanna Chen, Rafael Rosengarten, Douglas Densmore, Timothy Ham, Jay Keasling and Nathan Hillson.
5BDA.3 **Automatic Design of RNA and Transcriptional Circuits in *E. coli*** Guillermo Rodrigo, Thomas Landrain, Boris Kirov, Raissa Estrela, Javier Carrera and Alfonso Jaramillo.

In Silico Design of Functional DNA Constructs Based on Heuristic Data

Alan Villalobos
DNA2.0 Inc.
1140 O'Brien Drive
Menlo Park, CA 94025
+1 650 853 8347
avillalobos@dna20.com

Mark Welch
DNA2.0 Inc.
1140 O'Brien Drive
Menlo Park, CA 94025
+1 650 853 8347
mwelch@dna20.com

Claes Gustafsson
DNA2.0 Inc.
1140 O'Brien Drive
Menlo Park, CA 94025
+1 650 853 8347
cgustafsson@dna20.com

Jeremy Minshull
DNA2.0 Inc.
1140 O'Brien Drive
Menlo Park, CA 94025
+1 650 853 8347
jminshull@dna20.com

ABSTRACT

The promise of synthetic biology lies in the creation of novel function from the proper combination of independent genetic elements. De novo gene synthesis has become a cost-effective method for building virtually any conceptualized genetic construct, removing the constraints of extant sequences, and greatly facilitating study of the relationships between gene sequence and function. However, much of the inherent biological function of genetic elements is derived from evolutionary pressure unknown to the designer of the element. Using systematically varied genetic elements and wet-lab testing them in the relevant context allows for construction predictive models of biological behavior that can be implemented in the required genetic design.

With the rapid increase in the number and variety of characterized and cataloged genetic elements, tools that facilitate assembly of such parts into functional constructs (genes, vectors, circuits, etc.) while simultaneously utilizing the heuristic predictive models are essential for future progress of the field. The Gene Designer software allows scientists and engineers to readily manage and recombine genetic elements into novel assemblies. It also provides tools for the simulation of molecular cloning schemes as well as the engineering and optimization of protein-coding sequences. Together, the functions in Gene Designer provide a complete capability to design functional genetic constructs.

Keywords

Synthetic biology, protein expression, gene optimization, graphic user interface, click-and-drag

1. SOFTWARE TOOLS FOR GENE DESIGN

Synthetic biology, with its focus on the design of new genetic function, will be enabled by computational tools facilitating the design of new DNA molecules that can encode these functions. Most of today's programs for handling DNA sequence information, however, are developed for 'top-down' applications, oriented toward analysis of existing sequences. Designing new genetic constructs instead requires 'bottom-up' focus where independent elements can be configured into a larger entity. This is presumably a legacy of exponentially increasing amounts of genomic and metagenomic sequence information and the tools developed to organize and systematize avalanches of genomic sequences coming online. In consequence, current software is poorly suited to de novo genetic design.

Gene Designer was developed was developed as a 'bottom-up' solution for design of genetic constructs where each element can be defined and incorporated using easy click-and-drag graphic user interface while retaining an advanced filter for management of design constraints [7].

The tool has import/export functions in genbank and fasta format and is built with the intent of facilitating information transfer between different emerging synthetic biology platforms [1].

2. GENE DESIGN VARIABLES

Designing genetic elements for novel properties in novel context is difficult as the majority of genetic constructs are highly context dependent, retain large amounts of functional biological information not necessarily related to the goal of the engineer, and the exact variables amenable for changing biological properties are often unknown and/or correlated to other non-relevant or non-preferred functional properties.

DNA2.0 has addressed the challenge of linking gene design variables with functional properties by designing systematically varied datasets where gene variables are explored orthogonal from each other. Previous work used gene synthesis in conjunction with machine learning algorithms to build and wet-lab validate predictive models for transcriptional promoter strength [5], and protein engineering [3, 6, 2].

3. APPLYING GENE DESIGNER FOR HETEROLOGOUS PROTEIN EXPRESSION

Natural genes are often difficult to express outside their original context. They might contain codons that are rarely used in the desired host, come from organisms that use non-canonical code or contain expression-limiting regulatory elements within their coding sequence [4]. Despite a plethora of side by side comparison protein expression studies between 'natural' and 'optimized' coding sequences, preciously little knowledge has emerged from these studies. This is primarily due to the highly correlated multidimensionality of variables affecting protein expression. Two data points is not sufficient to explore multidimensional space [12].

DNA2.0 has developed technologies to identify and quantify the variables affecting heterologous protein expression [9, 10, 11]. Systematic analysis of gene design parameters allowed us to identify codon usage within a gene as a critical determinant of achievable protein expression levels in E. coli. We proposed a biochemical basis for this, as well as design algorithms to ensure high protein production from synthetic genes [8]. Replication of this methodology has allowed similar design algorithms to be empirically derived for expression systems such as mammalian cell lines, plants and yeasts. Variables captured through this technology can directly be added as constraints for ORF design within Gene Designer.

4. ACKNOWLEDGMENTS

The codon optimization part of this work was supported by NSF grant no. 0750206 and the NIH Roadmap Center grant no. P50 GM073210. The software development part of this work was supported by DNA2.0 grant no. 1.

Gene Designer is available free of charge for download from https://www.dna20.com/genedesigner2.

5. REFERENCES

[1] Cai, Y., B. Hartnett, et al. 2007. A syntactic model to design and verify synthetic genetic constructs derived from standard biological parts. *Bioinformatics* 23: 2760-2767.

[2] Ehren, J., S. Govindarajan, et al. 2008. Protein engineering of improved prolyl endopeptidases for celiac sprue therapy. *Protein Eng Des Sel* 21: 699-707.

[3] Gustafsson, C., S. Govindarajan, et al. 2003. Putting engineering back into protein engineering: bioinformatic approaches to catalyst design. *Curr Opin Biotechnol* 14: 366-370.

[4] Gustafsson, C., S. Govindarajan, et al. 2004. Codon bias and heterologous protein expression. *Trends Biotechnol* 22: 346-353.

[5] Jonsson, J., T. Norberg, et al. 1993. Quantitative sequence-activity models (QSAM) - tools for sequence design. *Nucleic Acids Res.* 21: 733-739.

[6] Liao, J., M. K. Warmuth, et al. 2007. Engineering proteinase K using machine learning and synthetic genes. *BMC Biotechnol.* 7: 16.

[7] Villalobos, A., J. E. Ness, et al. 2006. Gene Designer: a synthetic biology tool for constructing artificial DNA segments. *BMC Bioinformatics* 7: 285.

[8] Welch, M., S. Govindarajan, et al. 2009. Design Parameters to Control Synthetic Gene Expression in Escherichia coli. *PLoS ONE* 4: e7002.

[9] Welch, M. and C. Gustafsson 2009. *Methods for determining properties that affect an expression property value of polynucleotides in an expression system.* US Patent Office 7,561,972, Assignee DNA2.0.

[10] Welch, M. and C. Gustafsson 2009. *Synthetic nucleic acids for expression of encoded proteins*. US Patent Office 7,561,973, Assignee DNA2.0.

[11] Gustafsson, C., S. Govindarajan, et al. 2010. *Systems and methods for designing and ordering polynucleotides.* US Patent Office 7,805,252, Assignee DNA2.0.

[12] Welch, M., A. Villalobos, et al. 2009. You're one in a googol: optimizing genes for protein expression. *J R Soc Interface* 6 : S467-476

j5 and DeviceEditor: DNA assembly design automation

Joanna Chen
Joint BioEnergy Institute
Emeryville, CA
joannachen@lbl.gov

Rafael D. Rosengarten
Joint BioEnergy Institute
Emeryville, CA
rdrosengarten@lbl.gov

Douglas Densmore
Boston University
Boston, MA
dougd@bu.edu

Timothy S. Ham
Joint BioEnergy Institute
Emeryville, CA
tsham@lbl.gov

Jay D. Keasling
Joint BioEnergy Institute
Emeryville, CA
JDKeasling@lbl.gov

Nathan J. Hillson
Joint BioEnergy Institute
Emeryville, CA
njhillson@lbl.gov

ABSTRACT

The production of advanced biofuels from cellulosic material requires concerted feedstock, cellulolytic enzyme, and microbial engineering efforts that share significant biological design and execution challenges, including the construction of combinatorial libraries of engineered protein-variants and metabolic pathways. With these challenges in mind, we have developed two on-line software tools, j5 and DeviceEditor, that automate and visualize the design of sequence agnostic, scar-less, multi-part assembly methodologies. Together, these tools offer a visual canvas for spatially arranging abstractions of genetic components, provide automated oligo, direct synthesis, and cost-optimal assembly process design, and integrate with liquid-handling robotic platforms to set up the PCR and multi-part assembly reactions. Our work aims to reduce the time and cost required to pursue large scale DNA construction tasks, and to enable research otherwise unfeasible without the assistance of biological design automation software tools.

Keywords

BioCAD, DNA assembly, design automation, visual design abstraction, combinatorial library

1. ACKNOWLEDGMENTS

This work conducted by the Joint BioEnergy Institute was supported by the Office of Science, Office of Biological and Environmental Research, of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231. This work was also supported by a Berkeley Lab LDRD award (N.J.H.).

2. REFERENCES

[1] Chen, J., Densmore, D., Ham, T.S., Keasling, J.D., and Hillson, N.J. 2012. DeviceEditor visual biological CAD canvas. *J. Biol. Eng.* 6, 1 (Feb. 2012).

[2] Hillson, N.J., Rosengarten, R.D., and Keasling, J.D. 2012. j5 DNA assembly design automation software. *ACS Synthetic Biology* 1, 1 (Jan. 2012), 14-21.

Automatic design of RNA and transcriptional circuits in *E. coli*

Guillermo Rodrigo, Thomas Landrain, Boris Kirov, Raissa Estrela, Javier Carrera, and Alfonso Jaramillo
Institute of Systems and Synthetic Biology.
Evry. France
+33-1-69474430

Alfonso.Jaramillo at issb.genopole.fr

ABSTRACT

We describe two automatic design methodologies allowing the engineering of functional RNA or transcriptional circuits in living cells. We validate it experimentally in *E. coli*.

Keywords

Synthetic Biology, RNA circuits, transcriptional networks, computational design, logic gates.

1. INTRODUCTION

The design and implementation of genetic circuits for cell reprogramming is propelling the emerging field of Synthetic Biology. We have developed novel *in silico* evolution methodologies to design circuits made of either RNA or transcription factors. We have applied such techniques to engineer novel logic and oscillatory devices that we characterize in *E. coli*. RNA is becoming a very designable macromolecule for synthetic biology [1].

In the first case, we will describe the first fully automated design methodology and experimental validation of synthetic RNA interaction circuits in living cells. We tested our methodology in *E. coli* by designing several positive riboregulators [2,3] with diverse structures and interaction models. The designed sequences exhibit very low similarity to any known non-coding RNA sequence. Our riboregulatory devices can work independently and in combination with transcription regulation to create complex logic circuits. RNA devices have been successfully engineered in eukaryotic systems [4,5] and we expect that our methodology could also be applicable there, although adapting it to the corresponding gene expression machinery.

In the second case, we have developed an automated design approach that combines models of regulatory elements to search the genotype space associated to a given phenotypic behavior. We apply it to the construction and characterization in *E. coli* of gene networks with logic or oscillatory behavior. We use microfluidic techniques to track the single-cell dynamics for several days. We have also engineered two coupled oscillators in a single cell. Coupling of two oscillators is known in physics to generate a number of interesting dynamics.

2. AUTOMATIC DESIGN OF RNA CIRCUITS

We have developed an inverse folding approach to design RNA interactions that we experimentally verify in *E. coli*. This is a *de novo* automated design of small RNA circuits, which allowed the engineering of fully synthetic positive riboregulators in *E. coli*. We introduce this model-driven approach as an automated methodology to design regulatory RNAs able to work in cellular circuits (see Fig. 1). Our software is called RNAdes. As evidence supporting our conclusion we designed several positive riboregulators, for which there is no known general rational design procedure, despite of their usefulness [1,2].

This complements previous methodologies reporting the design of RNA devices, where sequences of known natural RNA motifs with a given function (e.g., ribozymes) are used [7,8]. We developed a full sequence design methodology based on physics that explores all possible sequences compatible with the specifications. Contrary to the 10^{15} sequences that could be sampled in laboratory conditions, our approach allows us to explore spaces of 10^{40} sequences, which constitutes a clear step forward in our ability for the design of functional RNAs. Although we validated several designs in *E. coli*, our procedure could also be used with higher organisms and in several RNA frameworks. Our methodology also allows obtaining highly specific, RNA-controlled expression systems that will be useful in biotechnology.

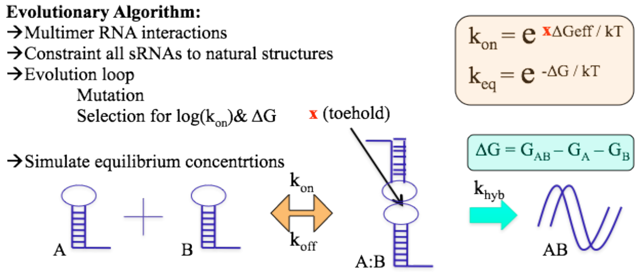

Evolutionary Algorithm:
→Multimer RNA interactions
→Constraint all sRNAs to natural structures
→Evolution loop
Mutation
Selection for $\log(k_{on})$& ΔG
x (toehold)
→Simulate equilibrium concentrtions
$k_{on} = e^{x\Delta Geff / kT}$
$k_{eq} = e^{-\Delta G / kT}$
$\Delta G = G_{AB} - G_A - G_B$
A
B
k_{on}
k_{off}
A:B
k_{hyb}
AB

Figure 1. The full sequence design methodology automatically finds novel sequences with predefined structures, toehold-mediated interactions (such as "kissing-loops") and stable intermolecular complexes. This is done by using a physical model of nucleotide interactions.

The methodology uses as input a target secondary structure for every RNA species and a target intermolecular interaction. The computational optimization optimizes the targeted interactions among all alternative ones. We have considered (see Fig 2.) several target interactions [6,7], giving rise to a diverse set of logic gate behaviors, although we have only tested experimentally the YES gates.

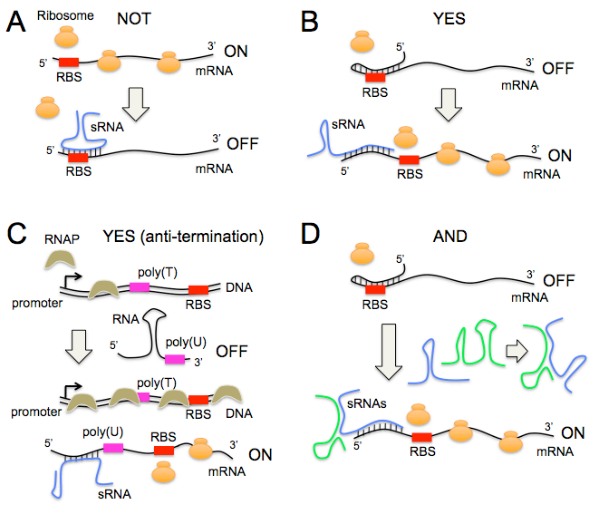


Figure 2. We can use our methodology to design sRNA implementing a variety of regulatory mechanisms.

An advantage of a full design methodology is that the sequences are completely different to any known natural or synthetic sequences. Fig. 3 shows the measured orthogonality between two designs.

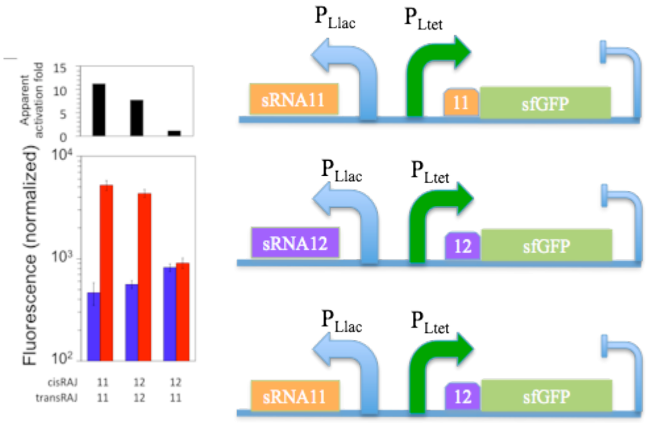


Figure 3. Testing of two of our YES riboregulators in MG1655Z1 (lacI+, tetR+), showing also their orthogonality.

3. AUTOMATIC DESIGN OF TRANSCRIPTIONAL CIRCUITS

Automatic design techniques could also be applied to the realm of transcriptional networks despite the lack of models able to quantitatively predict gene regulations from the sequence. This is possible thanks to the modularity of the components of a transcriptional circuit, where promoter and ORFs can be reassembled in novel combinations producing an often expected behavior [10] if the elements are chosen appropriately and fully characterized.

Our procedure, Genetdes++ [9], can be used to analyze the family of possible behaviors that could be engineered with a given parts library (Fig. 4). It can be used to engineer noise-tolerant circuits and obtain new design principles.

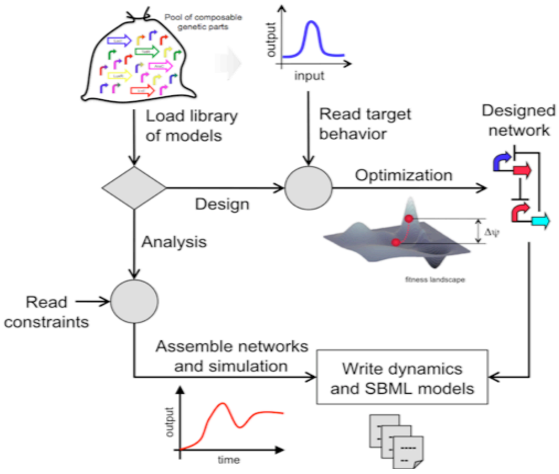


Figure 4. Genetdes++ algorithm to design circuits.

4. ACKNOWLEDGMENTS

Work supported by the FP7-ICT-043338 (BACTOCOM), the ATIGE-Genopole, and the Fondation pour la Recherche Medicale grants (to A.J.). G.R. is supported by an EMBO long-term fellowship co-funded by Marie Curie actions (ALTF-1177-2011), and T.E.L. by a PhD fellowship from the AXA Research Fund.

5. REFERENCES

[1] Isaacs, F.J., Dwyer, D.J., & Collins, J.J. (2006) RNA synthetic biology. *Nat. Biotechnol.* **24**, 545-554.

[2] Isaacs, F.J., *et al.* (2004) Engineered riboregulators enable post-transcriptional control of gene expression. *Nat. Biotechnol.* **22**, 841-847.

[3] Callura, J.M., *et al.* (2010) Tracking, tuning, and terminating microbial physiology using synthetic riboregulators. *Proc. Natl. Acad. Sci. USA* **107**, 15898-15903.

[4] Rinaudo, K., *et al.* (2007) A universal RNAi-based logic evaluator that operates in mammalian cells. *Nat. Biotechnol.* **25**, 795-801.

[5] Culler, S.J., Hoff, K.G., & Smolke, C.D. (2010) Reprogramming cellular behavior with RNA controllers responsive to endogenous proteins. *Science* **330**, 1251-1255.

[6] Dawid, A., Cayrol, B., & Isambert, H. (2009) RNA synthetic biology inspired from bacteria: construction of transcription attenuators under antisense regulation. *Phys. Biol.* **6**, 025007.

[7] Lucks, J.B., *et al.* (2011) Versatile RNA-sensing transcriptional regulators for engineering genetic networks. *Proc. Natl. Acad. Sci. USA* **108**, 8617-8622.

[8] Penchovsky, R., & Breaker, R.R. (2005) Computational design and experimental validation of oligonucleotide-sensing allosteric ribozymes. *Nat. Biotechnol.* **23**, 1424-1433

[9] Rodrigo, G., Carrera, J., & Jaramillo, A. (2011) Computational design of synthetic regulatory networks from a genetic library to characterize the designability of dynamical behaviors. *Nucl. Acids Res.* **39**, e138.

[10] Alper, H., *et al.* (2005) Tuning genetic control through promoter engineering. *Proc. Natl. Acad. Sci. USA* **102**, 12678-12683.

Monday – June 4th
Tech. Talks Session 6 - *Topic: Biological Circuit Design and Assembly II*
5BDA.4 **Integrating Synthetic Gene Assembly and Site-Specific Recombination Cloning** Bianca J Lam, Federico Katzen, Kevin Clancy, Xiangdong Liu, Nian Liu, Gengxin Chen, Kimberly Wong, Todd Peterson, Antje Pörtner-Taliana.
5BDA.5 **Scaling Responsively: Towards a Reusable, Modular, Automatic Gene Circuit Design** Linh Huynh and Ilias Tagkopoulos.
5BDA.6 **Chance-Constraint Optimization for Gene Modifications** Mona Yousofshahi, Michael Orshansky, Kyongbum Lee and Soha Hassoun.

Integrating Synthetic Gene Assembly and Site-Specific Recombination Cloning

Bianca J Lam
+ 1 7602688518
bianca.lam@lifetech.com

Federico Katzen
+ 1 7604767145
federico.katzen@lifetech.com

Kevin Clancy
+ 1 7602688356
kevin.clancy@lifetech.com

ABSTRACT

We developed an *in vitro* workflow streamlined to assemble full length genes from synthetic fragments directly into expression vectors for testing in a wide range of organisms. The genes of interest are first divided into small fragments for synthesis and are constructed with homology to other fragments or expression vector. The fragments are assembled into larger subfragments by high-fidelity PCR. For genes shorter than 12 kb, the gene subfragments and an expression vector are added to an enzymatic mix that assembles the subfragments and vector in the correct order and orientation. Finally, the enzymatic reaction is transformed into *E. coli* for plasmid propagation and screening resulting in seamless full length genes unmodified by extra or missing sequences caused by traditional cloning techniques. For genes larger than 12 kb, gene subfragments were first assembled into pUC19 and then through a second round of homologous recombination cloning, assembled into full length genes into the final expression vector. Assembled genes can also be site-specifically recombined to make multiple expression plasmids containing different elements and/or tags thereby circumventing the need to sequence re-verify gene. Thus, this technology allows for simultaneous testing of the genes of interest in bacteria, yeast, algae, plants, insect, and/or mammalian cells. The described workflow is currently being adapted for an automated high-throughput platform for the concurrent construction of multiple full-length genes. These technologies combined with computer-aided design of strategy, screening, automation, and LIMS will greatly advance gene editing, protein engineering, synthetic pathway engineering, and host engineering efforts.

Categories and Subject Descriptors

J.3. [**Life and Medical Sciences**]: *biology and genetics*

General Terms

Design, Experimentation

Keywords

Synthetic biology, synthetic genes, gene assembly, cloning, metabolic engineering

1. INTRODUCTION

Current methods to engineer biological systems and processes include gene and/or genome assembly, which requires advance cloning technologies. Homologous recombination cloning utilizes terminal end-homology between DNA fragments resulting in the directional and seamless insertion of multiple DNA fragments into a cloning vector. Using homologous recombination cloning, ~444 synthetic gene fragments were assembled into forty-four 7-27 kb full length genes with high cloning efficiency. Assembled genes were also exchanged into other vectors without the need for re-sequencing using site-specific recombination cloning. With computer-aided design of cloning strategies and screening of assembled genes, automation, and a LIMS for sample tracking, we aim to increase the efficiency and robustness of this synthetic gene assembly workflow.

2. RESULTS

2.1 Gene Assembly Strategy and Design

Full length genes ranging from 7-27 kb were divided into ~1 kb fragments. Fragments contained homology to vector or adjacent fragments. Gene fragments were received sequence confirmed in a cloning vector. Forward and reverse primers used for PCR amplification of the fragments hybridized to regions of overlap. The first and last primer contained homology to pcDNA-Dest40 (Invitrogen^{TM}) or pUC19. Additional primers used for sequencing were located in the middle of 1 kb fragments. GENEART® synthesized and assembled the gene fragments into a pMX vector.

2.2 First PCR

The 1 kb fragments were first PCR amplified and treated with Dpn I to digest supercoiled plasmid template.

2.3 Assembly PCR

PCR products from two or three consecutive 1 kb gene fragments were combined for assembly PCR reactions to create larger gene subfragments.

2.4 Seamless DNA Fragment Assembly

For genes ≤12 kb, the subfragments, linearized pcDNA-Dest40 (Invitrogen^{TM}), and GENEART® Seamless Cloning enzyme were incubated at room temperature for one hour. Reactions were transformed into Top 10 (Invitrogen^{TM}) for plasmid propagation and screening . For genes >12 kb, gene subfragments were first assembled into pUC19 as 4-6 kb fragments. The subfragments were then released from pUC19 by either PCR amplification or restriction digest for a second round of homologous recombination cloning into the desired plasmid.

2.5 Screening for Full Length Genes

Full length genes assembled into vector were screened by restriction digestion. Restriction endonucleases for screening were chosen based on the information content digested bands provided. Four clones that passed restriction digestion screening were selected for full length sequencing. Generally, ≥25% of sequenced clones with the correct banding pattern matched the predicted sequence 100%.

IWBDA'12, June 3–4, 2012, San Francisco, CA, USA.

For Research Use Only. Not for human or animal therapeutic or diagnostic use.

2.6 Integration into Site-Specific Recombination Cloning

Gene constructs with 100% sequence match to the predicted sequence underwent site-specific recombination cloning with pDONR™221 (Invitrogen) or equivalent to create Entry vectors. Entry vectors can be site-specifically recombined into any Destination vector to create different expression plasmids.

3. DISCUSSION/CONCLUSIONS

The cloning technologies and workflow discussed here consist of a robust method for assembling and cloning large genes and/or DNA fragments from smaller parts. For assembling PCR-amplified genes or large gene subfragments (up to 12 kb) into vector, we observed 10-80% of picked colonies contained all fragments in the vector. Typically, a minimum of 25% of plasmids containing all DNA fragments had no mismatches. We suspect that sequence content of the genes and the ad hoc method of fragment and primer design contributed to the wide range of variation in cloning efficiency. However, we found that primer quality was the primary factor on the success of this method. The use of HPLC or PAGE purified primers increased the success of fragment assembly and genes with 100% sequencing match. Assembly of pre-cloned gene subfragments into an expression vector was similarly efficient where 17-83% of picked colonies contained all fragments. We are in the process of implementing computer-aided design software, such as Vector NTI®, to choose optimal sites for gene fragmentation and PCR primers and to devise the screening strategy for assembled DNA fragments. Communication between the CAD software and LIMS will facilitate sample tracking and streamline each step within the cloning procedure.

4. ADDITIONAL AUTHORS

Xiangdong Liu (xiangdong.liu@lifetech.com +1 7606032894), Nian Liu (nian.liu@lifetech.com +1 7604764363), Gengxin Chen, Kimberly Wong (kimberly.wong@lifetech.com,+ 1 7602688405), Todd Peterson (todd.peterson@lifetech.com, + 1 7604767180), and Antje Pörtner-Taliana (antje.taliana@lifetech.com, + 1 7604766842).

5. REFERENCES

[1] GENEART® Seamless Cloning and Assembly Kit. Manual. http://tools.invitrogen.com/content/sfs/manuals/geneart_seamless_cloning_and_assembly_man.pdf**.**

[2] Gateway® Technology: A universal technology to clone DNA sequences for functional analysis and expression in multiple systems. Manual. https://tools.invitrogen.com/content/sfs/manuals/gatewayman.pdf.

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 [1] we have solved this problem by using exact 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.

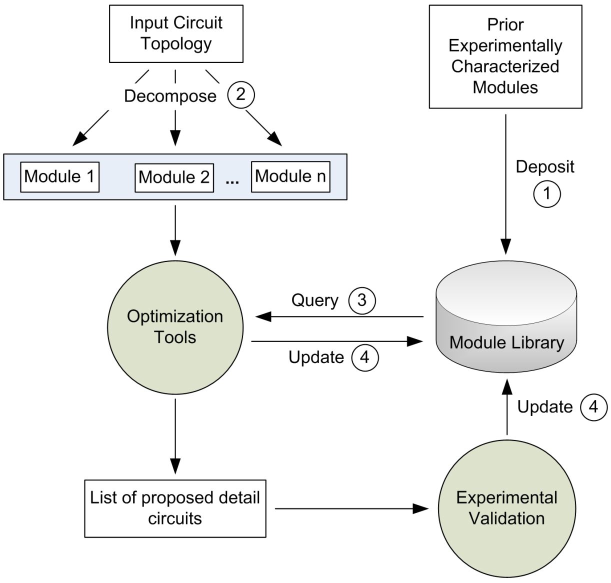


Figure 1: Overview of the proposed design automation framework

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 resulting 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 [2] 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 [3] 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 [4].

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 absense 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.

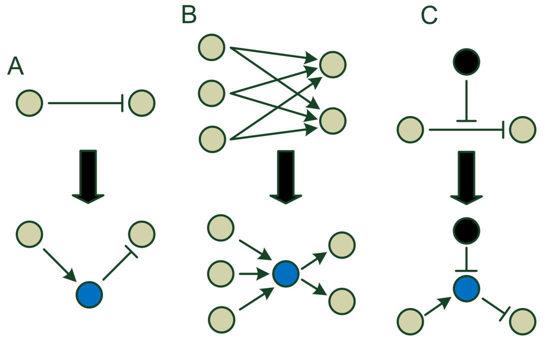


Figure 2: Graph transformation. Grey nodes represent genes (part of the gene set V_G), blue nodes represent proteins (part of the protein set V_P), and black nodes represent ligands (part of the ligand set V_L). (A) Protein-DNA interaction, (B) Protein-DNA interaction in a multiple gene copy, multiple 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] F. V and P. L, "Biological network querying techniques: analysis and comparison," *J. Comput. Biol.*, vol. 18, pp. 595–625, 2011.

[3] 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.

[4] 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.

Chance-constraint Optimization for Gene Modifications

Mona Yousofshahi[*], Michael Orshansky[♪], Kyongbum Lee[Δ], Soha Hassoun[*]

[*]Department of Computer Science. Tufts University
[♪]Department of Electrical and Computer Engineering, University of Texas at Austin
[Δ] Department of Chemical and Biological Engineering, Tufts University

{Mona.Yousofshahi, Kyongbum.Lee, Soha.Hassoun}@tufts.edu
orshansky @mail.utexas.edu

1. INTRODUCTION

Over the last decade, increasingly sophisticated molecular techniques have been used to engineer microbial cells capable of overproducing nonnative or "synthetic" biomolecules, including isoprenoid, polyketide, non-ribosomal peptide-based drugs and drug precursors, bioplastics, polymer building blocks, and biofuels. A central theme emerging from these efforts is that the optimality of these microbial production platforms critically depends on understanding the cell metabolism and the ability to engineer it to produce desired effects. Computational tools are thus needed to guide experimental efforts and automate the design process.

Current tools compute required genetic modifications, in the form of up-regulations (over-expression), down-regulations, and gene knockouts in metabolic networks, with the objective of maximizing the production (yield) of a desired product. Example approaches are OptReg [1], OptKnock [2], and GDLS [3]. Gene modification problems are formulated in terms of two kinds of variables: flux variables that represent the molecular turnover rate, and control (decision) variables that correspond to the presence or absence of regulation for each possible reaction and in each direction (up/down). The overall objective of the optimization procedure is to tune these variables optimally to maximize the production of a target metabolite. Mathematically, the solution must satisfy several linear constraints including: steady state constraints on the metabolic network, a minimum biomass production above a given threshold, and uptake values for some select fluxes. There are typically more reaction fluxes in the system than conservation equations that constrain the magnitude and direction of these fluxes. Consequently, the system of equations is typically underdetermined and this can be considered as "model uncertainty". Flux Balance Analysis can be used to characterize such a system, by maximizing and minimizing in turn each flux in the network. This model uncertainty can be further reduced by imposing experimental flux measurements. OptForce [4] addresses this uncertainty issue by identify minimal sets of engineering modifications that must be imposed to overproduce a target metabolite above a desired threshold.

Our work here addresses a source of uncertainty which emerges from the imprecision of engineering interventions. The uncertainty in achieving targeted enzyme values suggests that the enzyme levels, and hence the corresponding flux carrying capacities (bounds), could be considered statistical distributions rather than fixed value parameters. In this statistical interpretation, a flux constraint in a conventional deterministic optimization problem represents the most conservative point in the flux capacity distribution, since a deterministic problem enforces all constraints with zero uncertainty. We propose to use chance-constraint programming [5] to select gene modifications. Chance-constraint programming is a powerful paradigm for dealing with uncertainty in optimization and has been applied successfully in the optimization of integrated circuits [6]. In this work, we formulate the gene selection problem to optimize the yield of a target metabolite using chance-constrained programming, and compare its results against a deterministic method.

2. METHODS

We describe a *chance-constraint* formulation with the objective of maximizing a target metabolite with recourse to gene up/down regulation. The uncertain parameters in this formulation are the flux capacities, which are set by the expression levels of the corresponding genes. This is achieved by introducing a probabilistic constraint that the flux value resulting from up/down-regulation does not exceed the flux capacity with a given probability:

$$Prob\{v_j \le (1-y_j^u).(1-y_j^d).SSU_j + y_j^u(1-y_j^d).Cap_j^u + y_j^d(1-y_j^u).Cap_j^d\} \ge 1-\varepsilon$$

Here, v_j is the flux value for reaction j, y_j^u and y_j^d are the binary decision variables for the up and down regulation respectively, and SSU_j is the steady state upper bound. Cap_j^u and Cap_j^d are two random variables which denote flux capacities when reaction j is up/down regulated. $1-\varepsilon$ represents the confidence level of meeting the constraint. Figure 1 illustrates the uncertainty due to engineering interventions (upper flux capacity variations). To find the distribution of flux capacities, we multiply the maximum velocity, which is the maximum reaction rate for a given quantity of an enzyme, by the enzyme activity distributions, which are modeled with normal distributions.

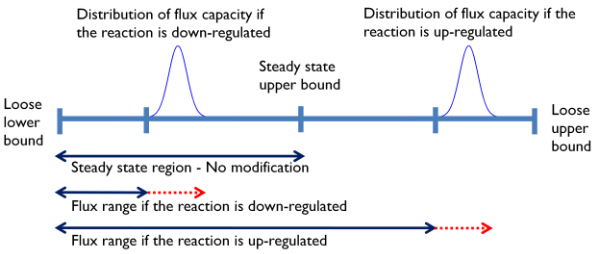

Distribution of flux capacity if the reaction is down-regulated
Distribution of flux capacity if the reaction is up-regulated
Steady state upper bound
Loose lower bound
Loose upper bound
Steady state region - No modification
Flux range if the reaction is down-regulated
Flux range if the reaction is up-regulated

Figure 1: Chance-constraint upper-bound constraint with or without regulations. The steady state region and the flux range during up/down regulation are shown. The normal distributions represent flux capacity distributions.

3. RESULTS

To evaluate our method, we compare results of the chance-constraint approach against a conservative *deterministic* approach, which maximizes the yield of the target metabolite while selecting gene modifications assuming *fixed* upper flux capacity bounds (lowest values of the flux capacity distributions). We chose a model of the Chinese hamster ovary (CHO) cell [7] with 46 irreversible reactions as a test case with the objective of maximizing the antibody production. The summary of results is shown in Fig. 2. The x-axis shows the upper bound on the number of gene modifications and the y-axis represents the maximum production rate. The set on each flux point lists the reaction numbers for the intervention set obtained by each method. As shown in the figure, the deterministic approach generates smaller rate values and more limited intervention sets than the chance-constraint method.

To evaluate these approaches, we performed Monte Carlo (MC) simulations on the CHO cell using interventions obtained using each approach, and then applied flux balance analysis to maximize the desired product. The MC method samples the parameter variation space from the flux capacity distributions. Each MC simulation generates a flux distribution of the target metabolite for chance-constraint and deterministic solutions. The 5th-95th percentile values of these distributions are calculated. The calculated rates using chance-constraint are always in 5th-95th percentile values of MC distributions while the deterministic rate values are close to the lower end of distributions.

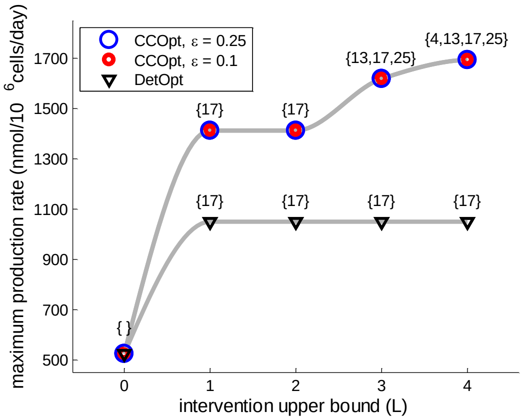


Figure 2: Maximum production rate and intervention sets obtained from chance-constraint and deterministic approaches for antibody production in the CHO cell. The x-axis represents the upper bound on the number of interventions and the y-axis shows the maximum production rate. The intervention sets are shown above each data point.

4. CONCLUSION

We proposed a chance-constraint method to identify gene modifications in a metabolic network leading to increased production of a target metabolite while considering the flux capacity uncertainty. Our results show that the chance-constraint method outperforms the deterministic approach in terms of predicted maximum rates and diverse set of interventions.

5. REFERENCES

[1] P. Pharkya and C. D. Maranas, "An optimization framework for identifying reaction activation/inhibition or elimination candidates for overproduction in microbial systems," *Metab. Eng.*, vol. 8, pp. 1-13, 1, 2006.

[2] A. P. Burgard, P. Pharkya and C. D. Maranas, "Optknock: A bilevel programming framework for identifying gene knockout strategies for microbial strain optimization," *Biotechnol. Bioeng.*, vol. 84, pp. 647-657, 2003.

[3] D. S. Lun, G. Rockwell, N. J. Guido, M. Baym, J. A. Kelner, B. Berger, J. E. Galagan and G. M. Church, "Large-scale identification of genetic design strategies using local search," *Mol Syst Biol*, vol. 5, 08/18/print, 2009.

[4] S. Ranganathan, P. F. Suthers and C. D. Maranas, "OptForce: An Optimization Procedure for Identifying All Genetic Manipulations Leading to Targeted Overproductions," *PLoS Comput Biol*, vol. 6, pp. e1000744, 04/15, 2010.

[5] A. Charnes and W. W. Cooper, "Chance-Constrained Programming," *Management Science*, vol. 6, pp. pp. 73-79, Oct., 1959.

[6] M. Mani, A. Devgan, M. Orshansky and Yaping Zhan, "A Statistical Algorithm for Power- and Timing-Limited Parametric Yield Optimization of Large Integrated Circuits," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 26, pp. 1790-1802, 2007.

[7] R. P. Nolan and K. Lee, "Dynamic model of CHO cell metabolism," *Metab. Eng.*, vol. 13, pp. 108-124, 1, 2011.