Estimation of gene network parameters from imaging cytometry data

Matthew William Lux

Dissertation submitted to the faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
In
Genetics, Bioinformatics, and Computational Biology

Jean Peccoud, Chair
William T. Baumann
Brett M. Tyler
John J. Tyson

April 30th, 2013
Blacksburg, Virginia

Keywords: synthetic biology, computational modeling, parameter estimation, systems biology

Estimation of gene network parameters from imaging cytometry data

Matthew William Lux

ABSTRACT

Synthetic biology endeavors to forward engineer genetic circuits with novel function. A major inspiration for the field has been the enormous success in the engineering of digital electronic circuits over the past half century. This dissertation approaches synthetic biology from the perspective of the engineering design cycle, a concept ubiquitous across many engineering disciplines. First, an analysis of the state of the engineering design cycle in synthetic biology is presented, pointing out the most limiting challenges currently facing the field. Second, a principle commonly used in electronics to weigh the tradeoffs between hardware and software implementations of a function, called co-design, is applied to synthetic biology. Designs to implement a specific logical function in three distinct domains are proposed and their pros and cons weighed. Third, automatic transitioning between an abstract design, its physical implementation, and accurate models of the corresponding system are critical for success in synthetic biology. We present a framework for accomplishing this task and demonstrate how it can be used to explore a design space. A major limitation of the aforementioned approach is that adequate parameter values for the performance of genetic components do not yet exist. Thus far, it has not been possible to uniquely attribute the function of a device to the function of the individual components in a way that enables accurate prediction of the function of new devices assembled from the same components. This lack presents a major challenge to rapid progression through the design cycle. We address this challenge by first collecting high time-resolution fluorescence trajectories of individual cells expressing a fluorescent protein, as well as snapshots of the number of corresponding mRNA molecules per cell. We then leverage the information embedded in the cell-cell variability of the population to extract parameter values for a stochastic model of gene expression more complex than typically used. Such analysis opens the door for models of genetic components that can more reliably predict the function of new combinations of these basic components.

Dedication

To my father, to whom I owe my interest in science. If only you could be here today.

To Mike, a great friend to me and so many others. You are missed.

Acknowledgements

I would first like to thank my family for their support over the years. My brother, because few things are as encouraging as a sybling's admiration. My father, for instilling in me a love of science and nature at a young age. My mother, for thinking everything I did was great; the self-confidence you instilled made it possible for me to get where I am today.

I am grateful to my adviser, Dr. Jean Peccoud, for 5+ years of mentorship. Throughout my PhD, Jean has allowed me to space to develop while providing guidance when needed. I have always felt that my thoughts have been valued, if not always agreed with, which has made my time here much more rewarding. Thanks to Jean, I have learned a great deal about what it is to be a scientist and feel well prepared for my career ahead.

For my initial nudge towards synthetic biology, I thank another of my committee members, Dr. William Baumann. As a senior undergraduate, his courses on control theory were two of my favorites in my entire academic career, though as a senior undergraduate, my attendance record perhaps did not reflect that sentiment. His mention of iGEM in class led directly to where I am today. Working through complex problems on the whiteboard in his office are some of my favorite memories of my graduate career.

I would also like to thank my other committee members, Drs. John Tyson and Brett Tyler, for fruitful conversations, solid advice, and inspiration. I would similarly like to thank the many other collaborators and colleagues that have contributed to my accomplishments over the past 5 years.

I am indebted to the GBCB program for providing such a flexible interdisciplinary environment in which to pursue my PhD. In particular, I thank Dennie Munson, without whom I don't know how the program would function. She has fixed or prevented a great many blunders as I've navigated through the various university bureaucracies.

I owe much to my synthetic biology co-workers. Dr. David Ball has provided invaluable scientific support, discussion, and friendship. Dr. Yizhi Cai always provided great support as the senior graduate student in the group, as well as being a close friend. As a fellow student in the group, Laura Adam has been a close friend and I've benefited much from our conversations about life as a graduate student. Jodi Lewis has helped me with many administrative tasks, and been a friend as well. I also thank all the other members of the group, past and present, for countless conversations and support of all kinds.

I thank all of my friends for motivation at times and distraction at others.

Finally, I thank my girlfriend, Devon, whose unconditional support has been invaluable. Through tragedy and adventure, through weeks of non-stop work and no motivation, through your own trials and successes, you have always been there for me. It is easy to take for granted what is always there, but looking back, it is clear how important your support has been. They say the only thing that doesn't change is change itself, but your unwavering support gives change a run for its money. I cannot thank you enough.

Attributions

Chapter 2: Genetic Design Automation: engineering fantasy or scientific renewal?

- Brian W. Bramlett (Lux Bio Group): a former engineer at Intel, Brian provided expertise in electronic design automation and perspectives on the parallels with genetic design automation.
- David A. Ball (Virginia Bioinformatics Institute, Virginia Tech): synthetic biologist and imaging expert, David contributed significantly to the section on future instruments in synthetic biology.
- Jean Peccoud (Virginia Bioinformatics Institute, Virginia Tech): Jean oversaw the work and wrote significant sections of the manuscript.

Chapter 3: Co-design in synthetic biology: a system-level analysis of the development of an environmental sensing device

- David A. Ball (Virginia Bioinformatics Institute, Virginia Tech): David contributed the design and preliminary data on the spectral unmixing section, as well as writing large sections of the text.
- Russell R. Graef (MITRE Corporation): Russel provided input on the target design and useful discussions on objectives and context
- Matthew W. Peterson (MITRE Corporation): Matthew provided input on the target design and useful discussions on objectives and context
- Jane D. Valenti (Virginia Bioinformatics Institute, Virginia Tech): a summer REU student, Jane contributed to the development of the hybrid promoter design
- John Dileo (MITRE Corporation): oversaw the work from the MITRE side and provided useful discussions on objectives and context
- Jean Peccoud (Virginia Bioinformatics Institute, Virginia Tech): Jean oversaw the work and wrote significant sections of the manuscript.

Chapter 4: Modeling structure-function relationships in synthetic DNA sequences using attribute grammars

- Yizhi Cai (Virginia Bioinformatics Institute, Virginia Tech): Yizhi is the primary author of the work and was responsible for the development of the framework to apply attribute grammars to DNA sequences. This framework enabled the exploration of the design space described in the manuscript, which is the key section as pertains to this dissertation.
- Laura Adam (Virginia Bioinformatics Institute, Virginia Tech): a summer student from ENSIMAG at the time, implemented the framework developed by Yizhi Cai in an alternative environment, ANTLR.
- Jean Peccoud (Virginia Bioinformatics Institute, Virginia Tech): Jean oversaw the work and wrote significant sections of the manuscript.

Chapter 5: Estimation of stochastic gene expression rate parameters from imaging cytometry data

- David A Ball (Virginia Bioinformatics Institute, Virginia Tech): David developed GenoSIGHT and performed the FISH experiments. He also provided feedback on the analysis and helped prepare the manuscript.
- William T. Baumann (Department of Electrical and Computer Engineering, Virginia Tech): William contributed many useful discussions and advice throughout the course of the project, as well as helped prepare the manuscript.
- Jean Peccoud (Virginia Bioinformatics Institute, Virginia Tech): Jean oversaw the work and wrote significant sections of the manuscript.

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction

1.1 Synthetic Biology

Synthetic biology is an emerging discipline that aims to forward engineer genetic circuits with novel function [1]. The ability to do so promises solutions to many to the world's most pressing problems [2], including applications in medicine, energy, and the environment. The field relies on the notion that we now know enough about how cells function that we can begin to assemble known basic components into new circuits with desired functions. In this sense, it is natural to draw parallels between synthetic biology and the engineering of electronics.

Indeed, many of the milestones in the field have involved replicating functions often seen in electronics, such as switches [3, 4], oscillators [5-8], and logic gates [9-12]. With a growing library of these proof-of-concept designs, synthetic biologists are beginning to assemble these simple networks into larger, more complex designs [13], with modest success [14]. The stagnation in the growth in the complexity of engineered genetic networks encourages a shift in focus towards where the limiting steps are in the process of engineering these circuits. This dissertation approaches the field from the perspective of the engineering design cycle and uses this point of view to identify and address some of the challenges limiting progress in the field.

1.2 Organization

Chapter 2 contains a manuscript that reviews the field of synthetic biology from the viewpoint of the engineering design cycle, and more specifically how it compares to the design cycle as it currently exists in the engineering of digital electronics. Briefly, the design cycle can be broken down into 3 distinct phases: design, build, and test. Throughout the review, comparisons are made between the tools and techniques that enable electronics engineers to rapidly progress through the design cycle and the successes and shortcomings of parallel tools in synthetic biology. The review points out the open challenges in each phase to rapid development of novel synthetic genetic circuits. The article also argues that formalizing the design principles of genetic devices enables us to test the depth of our understanding of these networks and identifies places where further investigation is required.

Chapter 3 introduces the concept of co-design to synthetic biology. Co-design is a common concept in the design of digital electronics where a tradeoff can often be made between implementing a specific function in hardware or software. Since cells have many layers of expression control, so too could synthetic biologists implement engineered logic at different levels. The paper presents three alternative designs for a 3-input logic gate with pairwise outputs implemented at the transcription, protein, and detection levels of gene expression.

Chapter 4 describes a framework to move between abstract descriptions of a genetic circuit design, the DNA sequence encoding it, and mathematical models describing its function. More specifically, the framework involves applying the concept of attribute grammars from formal language theory to synthetic DNA sequences. However, in the context of this dissertation, it is the application of the framework that is germane rather than its development, which is the work of Dr. Yizhi Cai. The application presented involves the exploration of a genetic design space consisting of variants of a simple genetic toggle switch. We demonstrate how automatic assembly of the set of toggle switch designs possible from a library of components allows the selection of specific designs with desired properties.

A major limitation to the framework presented in Chapter 4 is the lack of reliable parameter values for models of the individual components that make up a design. The same lack also severely limits the utility of the co-design approach presented in Chapter 3. Chapter 5 addresses this major gap between models and data in synthetic biology. One parallel between genetic circuits and electronic circuits is the notion that by composing models of basic components, one can predict the function of novel circuits. In electronics, this concept has been used prodigiously. In synthetic biology, the concept has been oft discussed but minimally demonstrated. The reason for this discrepancy stems from a general inability to collect data sufficient to uniquely attribute the contribution to the overall function of each component. To address this issue, we collect advanced datasets that increase the information available from a single dataset, while simultaneously reducing the labor cost of collecting the data. We then leverage the information embedded in the cell-cell variability to parameterize a stochastic model of gene expression. Importantly, the parameters are in absolute units rather than relative to some reference standard and they can be assigned uniquely to the genetic components. The ability to parameterize such models should facilitate the holy grail of synthetic biology: predictive models of novel genetic circuits from detailed models of the individual components.

Chapter 2: Genetic Design Automation: engineering fantasy or scientific renewal?

Authors

Matthew W Lux[1], Brian W Bramlett[2], David A. Ball[1], Jean Peccoud[1]

[1]Virginia Bioinformatics Institute, Virginia Tech, Blacksburg VA 24061
[2]Lux Bio Group SA

Abstract

Synthetic biology aims to make genetic systems more amenable to engineering, which has naturally led to the development of Computer-Aided Design (CAD) tools. Experimentalists still primarily rely on project-specific ad-hoc workflows instead of domain-specific tools, suggesting that CAD tools are lagging behind the front line of the field. Here, we discuss the scientific hurdles that have limited the productivity gains anticipated from existing tools. We argue that the real value of efforts to develop CAD tools is the formalization of genetic design rules, which sheds new light on the complex relationships between genotype and phenotype.

Publication status

This manuscript has been reproduced here, with permission, from its publication at:

Genetic design automation: engineering fantasy or scientific renewal?

Lux, Matthew W.; Bramlett, Brian W.; Ball, David A.; Peccoud, Jean

Trends in biotechnology doi:10.1016/j.tibtech.2011.09.001 (volume 30 issue 2 pp.120 - 126)

2.1 Computer-Aided Design tools for synthetic biology

Several groups have been developing Computer-Aided Design (CAD) solutions for synthetic biology [1-10] yet the transcriptional complexity of published artificial gene networks has been leveling off since 2005 [11]. After ten years of high-expectations and hype in synthetic biology, engineering biological systems has proved more challenging than anticipated [12]. The lack of sufficient tools in synthetic biology has spurred intense efforts to develop CAD software. Unfortunately, experimental synthetic biologists still rely largely on project-specific, ad-hoc development processes that combine construct assembly, data collection, data analysis, and mathematical modeling.

Five recent reviews have comprehensively covered the current state of computational tools for synthetic biology [13-17]. Thus, here we limit our description of these efforts to a brief, general overview. We also constrain the review to software specific to synthetic biology by excluding the many commercial software packages that are useful synthetic biologists, but intended for a broader user base. CAD tools for synthetic biology facilitate the design of larger systems from smaller genetic partsby providing users with visual, textual, or programming-language-like interfaces, or automatically generate designs from intended function. These tools assume that data such as sequence and description are attached to each part by the user or in some database. The aggregated parts sequences can then be leveraged to produce the corresponding physical DNA. Many tools include some level of functional modeling capabilities, which rely on the user providing the necessary equations and/or parameters (Figure 2.1).

Thus far in synthetic biology, relatively simple design goals such as “exhibits oscillations” [18] have advanced to increasingly sophisticated goals such as “fast, robust, tunable oscillators” [19] or “synchronized oscillators” [20]. As the field moves towards real-world applications [21], tools that can adequately predict functionality from design will be indispensible. Similarly, designers need to begin considering alternative design approaches and corresponding comparison metrics [22] to move beyond proof of concept designs. Recently announced design-to-spec competitions like CAGEN (Competitive Assessment of Genetically Engineered Networks) and GenoCon (International Rational Genome Design Contest) aim to help address this need (Table 2.1).

The goal of this paper is to first explore the scientific hurdles that have limited the productivity gains anticipated from existing CAD tools, and to second argue that the real value of these efforts is not in promised productivity gains, but in the formalization of genetic design rules. Formalizations inherently test commonly held conceptions of how genetic systems work and consequently drive investigation into one of the most fundamental questions of genetics: How does phenotype arise from complicated networks of elements coded in the genotype? [23]

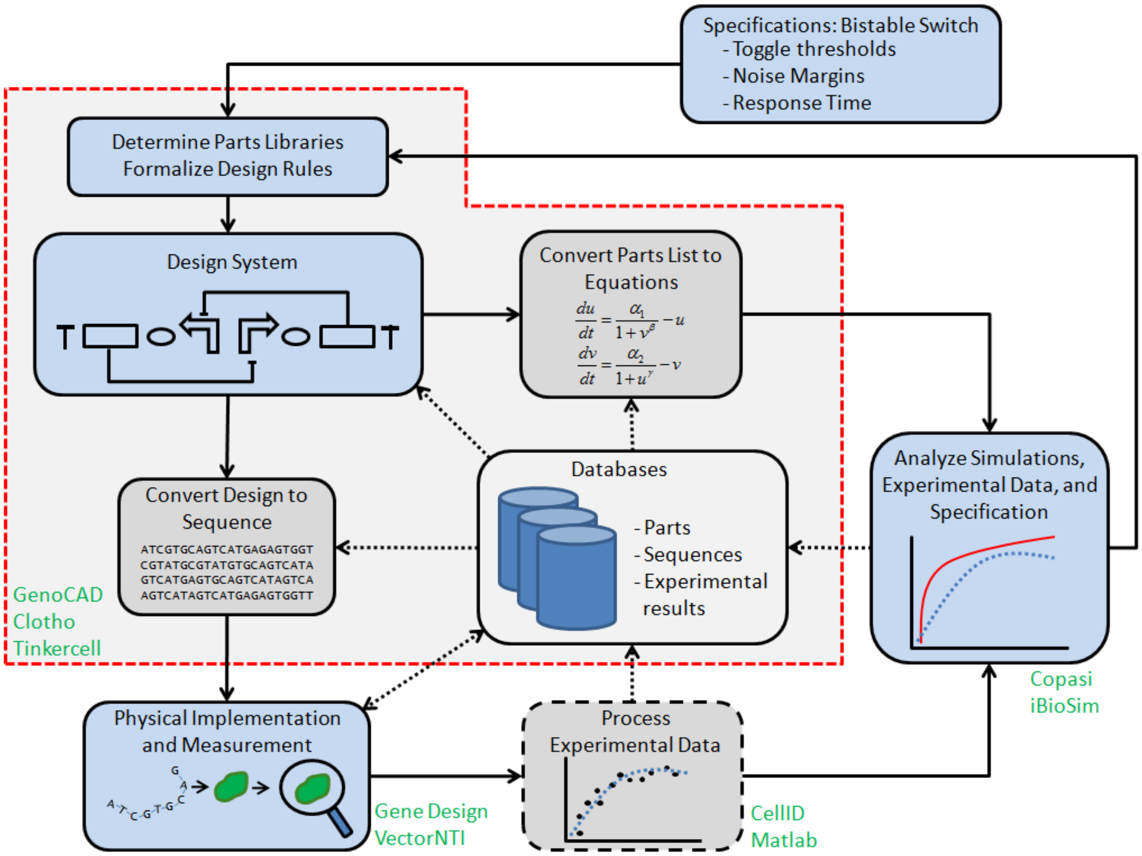
Specifications: Bistable Switch
- Toggle thresholds
- Noise Margins
- Response Time
Determine Parts Libraries
Formalize Design Rules
Design System
Convert Parts List to Equations
$\frac{du}{dt}=\frac{\alpha_1}{1+v^\beta}-u$
$\frac{dv}{dt}=\frac{\alpha_2}{1+u^\gamma}-v$
Convert Design to Sequence
ATCGTGCAGTCATGAGAGTGGT
CGTATGCGTATGTGCAGTCATA
GTCATGAGTGCAGTCATAGTCA
AGTCATAGTCATGAGAGTGGTT
Databases
- Parts
- Sequences
- Experimental results
Analyze Simulations, Experimental Data, and Specification
GenoCAD
Clotho
Tinkercell
Copasi
iBioSim
Physical Implementation and Measurement
A T C G T G C A G
Process Experimental Data
Gene Design
VectorNTI
CellID
Matlab

Figure 2.1 GDA design flow.

Synthetic biology projects typically rely on iterative workflows composed of different tasks. Emerging GDA tool chains rely on numerous software applications that support different phases of the project workflow. The development of a genetic switch [72] will start by expressing the design objective as a list of quantitative requirements: input toggle thresholds, noise margins, switching response time, etc. Once the objective is specified, it is possible to develop a list of genetic parts useable for the project. The choice of biological parts will involve factors such as use of the parts in prior projects, quality of the data characterizing the parts function, or intellectual property considerations. The formalization of design rules often takes place in parallel to the parts library development. Design rules may express rules such as whether it is acceptable to have polycistronic expression cassettes or if the design should be split between different plasmids. Only after parts have been selected and a strategy has been agreed upon is it possible to start designing constructs. In the fabrication phase, the construct is assembled usually by combining de novo gene synthesis and cloning of existing DNA sequences. Users use molecular biology software suites to facilitate assembly or order the sequence from a gene synthesis company. Experimentalists insert the synthetic DNA molecule into the host of choice and collect phenotypic data. Experimental data is then processed, for example by reducing microscopy images into time series of quantitative data. Performance is evaluated by considering simulations, experimental data, and the original specifications. At nearly every stage, software interacts with databases to reuse leverage past work or to store current work for future use. Red dashed line delineates stages facilitated by synthetic biology CAD software, while other stages are handled by more *general purpose software. Text in green indicates examples of software providing assisting at each stage.*

URL	Brief Description
www.biodesignautomation.org	Community and annual event focused on Bio-Design Automation
openwetware.org/wiki/CAGEN	Competition aimed at more predictable genetic circuits
genocon.org	Competition aimed at practical genetic circuit engineering
www.partsregistry.org	Library of genetic components
registry.jbei.org	Library of genetic components
www.biofab.org	Facility that provides standardized collections of genetic components
www.sbolstandard.org	Community effort to develop data exchange standards

Table 2.1: List of Relevant Websites

2.2 The slow maturation of EDA

CAD tools are ubiquitous in nearly all fields of engineering. They provide two primary functions: simplifying common tasks and making designs convenient for communication and evaluation. For example, blueprints are designed much faster on a computer than with traditional drafting. Moreover, CAD tools can generate layouts of each floor, 3D renderings of the building, or models of the building's thermal performance.

In electronics, the development of consistent suites of CAD tools is called Electronic Design Automation (EDA). EDA utilizes iterative "design flows," where key design processes are looped back on themselves until the design meets required specifications. Abstract representations of an electronic design in EDA can be organized into a hierarchy consisting of high-level description, logical description, and physical layout.

The spectacular success of EDA over the last 50 years [24] provides an inspiring model for synthetic biology. The synthetic biology counterpart of EDA is sometimes referred to as Bio-Design Automation (Table 2.1). However, the alternative name Genetic Design Automation (GDA) may better emphasize that synthetic biology focuses more on engineering DNA molecules than other biological objects [25]. Many have proposed that synthetic biologists leverage expertise in the design of electrical circuits, and in the same way GDA can draw from the development of EDA. In EDA, Hardware Description Languages (HDLs) are a special category of programming languages used to formally describe immensely complicated designs in a compact way. These languages rely extensively on the existing abstraction that digital circuits operate under the laws of Boolean algebra, which allows hugely complex circuits to be designed reliably.

This assumption does not exist for GDA. Exploratory work on an HDL for GDA has progressed under the assumption that similar enabling assumptions will emerge [10]. In parallel, efforts within the EDA community to describe analog and mixed analog-digital circuits with HDLs are ongoing. The challenges of extending HDLs to analog circuits are very similar to the challenges faced in GDA, and indeed some works have explored these similarities [26, 27].

The next sections describe three difficult problems that need to be solved before the level of automation in EDA can be achieved with GDA. There is mounting evidence that the first generation of GDA tools will not be able to ignore some of the most challenging problems currently faced by the EDA community.

2.3 Engineering fantasies: the scientific gaps facing GDA

Transitions between DNA sequence, model, and fabrication are currently hindered not so much by the implementation of CAD tools, but rather by three difficult scientific challenges: (i) predictability of components, (ii) decoupling of design and fabrication and (iii) experimental characterization methods.

2.3.1 Off-the-shelf Components

One of the popular visions for synthetic biology describes catalogues of clearly defined genetic parts that can be easily combined into larger genetic constructs with predictable biological function. This vision motivated the development of the BioBrick assembly standard and the Registry of Standardized Genetic Parts, a database of BioBrick compatible parts (Table 2.1). Tools that aggregate models of basic genetic components to form system-scale models are being developed [1, 2, 4, 28], but the lack of data sheets listing quantitative parameters characterizing the parts behavior has hampered the use of these tools for designing artificial gene networks [29]. Projects such as BioFAB (Table 2.1) are attempting to address this issue by characterizing large numbers of parts and standardizing data collection techniques [30].

Recent efforts to quantitatively characterize the effects of different parts on gene expression is revealing a complex landscape of context-dependencies that somewhat challenges the assumption that parts can be characterized in isolation. For instance, the RBS sequence was first assumed by many in the field to determine translation efficiency in prokaryotes independently from the downstream coding sequence. However, sequences around the translation start site can influence the secondary structure of the mRNA, which is long known to play a crucial role in the translation rate [31]. Tools utilizing thermodynamic models [32, 33] are now available to predict the translation initiation efficiency in prokaryotes using sequence both upstream and downstream of the translational start site. Coupling between translation and transcription elongation rates [34] also represents a challenge to the standardization of components, though the assumption that initiation, not elongation, is the rate limiting step in transcription may be a valid approximation. As a result, tools that can predict behavior based on sequence, thermodynamics, or other methods are emerging as increasingly attractive.

The above issues can be avoided by characterizing on a gene-by-gene or device-by-device level, a trend already apparent in the field [11]. Creation of device variants or automatically generated devices [35] should consider the many context dependencies that affect parts. Yet, even such low levels of granularity might prove to have unexpected context dependencies. Computational studies, inspired by impedance-matching in electronics, have demonstrated an effect termed "retroactivity" in which the performance of one genetic device is influenced by connecting a downstream device [36, 37]. Just as electronic circuit designers are currently running into major power limitations, synthetic biologists are almost certain to run into limits on the many ingredients necessary for gene expression. How the availability of resources within a cell impacts the performance of individual genetic components and devices will also become an important consideration.

2.3.2 Decoupling of design and fabrication

Historically, recombinant DNA technologies were so limited that fabrication constrained design to the point that software focused almost entirely on assisting cloning rather than design of function. The recent emergence of generic DNA fabrication methods, including standardized assembly of genetic parts or *de novo* gene synthesis [38] led to the emergence of DNA-sequence design as a new scientific problem [39, 40]. Because it is now possible to assume that generic DNA fabrication processes can assemble any sequence genetic engineers imagine, design and fabrication tend to be considered orthogonal engineering problems.

In EDA, the assumption of the Boolean abstraction has allowed fabrication to be considered mostly orthogonal to the rest of the design process. As circuit densities have rapidly increased, fabrication constraints have become more closely intertwined with other constraints such as timing delay and power consumption [41]. Consequently, increasingly integrated tools consider constraints from multiple design domains simultaneously. Genetic design is still in the process of moving away from fabrication technologies that constrain the design space to achieve complete independence between design and manufacture. For instance, BioBrick assembly standards have moved from the original standard precluding assembly of fusion proteins (BBF RFC 10) to proposed standards allowing fusions [42]or scarless assembly (BBR RFC 26, 39). Most recently, single-step assembly methods [43] have become popular.

Despite the simplification decoupling offers, there are distinct advantages to recoupling design and manufacturing during the design phase. Poor design strategies can create manufacturing problems. For example, repeated use of the same parts can cause sequence verification difficulty and structural instabilities [44]. Sequences with high GC content are notoriously difficult to amplify. Even though experienced gene synthesis companies will be able to synthesize most DNA sequences ordered by their customers, price and time to delivery vary greatly with sequence complexity. Ignoring such manufacturing constraints during the design phase will significantly increase the cost and duration of GDA cycles. Since many projects require the characterization of large numbers of design variants, the cost and time to fabricate these designs is still one of the bottlenecks of the GDA loop. Sophisticated design strategies are necessary to formalize manufacturing constraints [45] and optimize fabrication without serious detriment to function. For example, tools that can adjust codon bias or match non-unique sequences to function [32, 33, 35] could ameliorate manufacturing concerns. Sample tracking tools could suggest reuse of sequence segments resulting in shortened and cheaper assembly cycles. Algorithms to optimize fabrication processes [46] will have to be connected to design tools to give designers an immediate appreciation of the manufacturing cost of candidate designs.

2.3.3 Parts Characterization

The problem of part definitions extends to the ways experimental data are collected, used, and shared. Models vary from project to project and typically have only one or two fluorescent reporters as measured outputs. Since the models typically have many more parameters to estimate, finding parameter sets that are predictive from the many possible sets that match a given dataset is difficult. Combined with the range of measurement techniques used in the lab and the unknown impact of even small changes in experimental conditions, standardization and reuse of collected data are challenging.

Though not necessarily an explicit step in standard EDA design flows, Design for Test (DFT) is a background constraint throughout the process. As electronic circuits have become more complex, validation testing has become correspondingly more difficult. First, immense complexity makes exhaustive validation intractable, so intelligent minimization of test programs has evolved along with sophisticated testing equipment [47]. Second, the density of modern chips has made accessing nodes on internal layers without unintended performance effects a major problem. In living cells, the problem is much worse. Unknown cellular mechanisms, genetic instabilities, molecular noise, measurement accuracy, and inability to measure key components have prevented the verification of synthetic biology designs.

DFT in GDA starts by ensuring consistency between the design specification, experimental characterization methods, and mathematical paradigms used to model the behavior. For example, models based on deterministic equations can be supported by cell culture assays, whereas stochastic models call for single cell observations. Experimental design, common in some other engineering fields and recently emerging as a hot topic in systems biology [48], can guide experimentalists to optimal sets of measurements and avoid non-informative data collection. Depending on the system, this approach can also predict data sets that will determine all model parameters to within a given tolerance [49]. Similar techniques are frequently used in EDA to design intelligent testing programs, particularly for System-on-a-Chip designs where analog components commingle with digital circuits. Not considering the experimental characterization during the design phase may lead to designs that simply cannot be adequately tested.

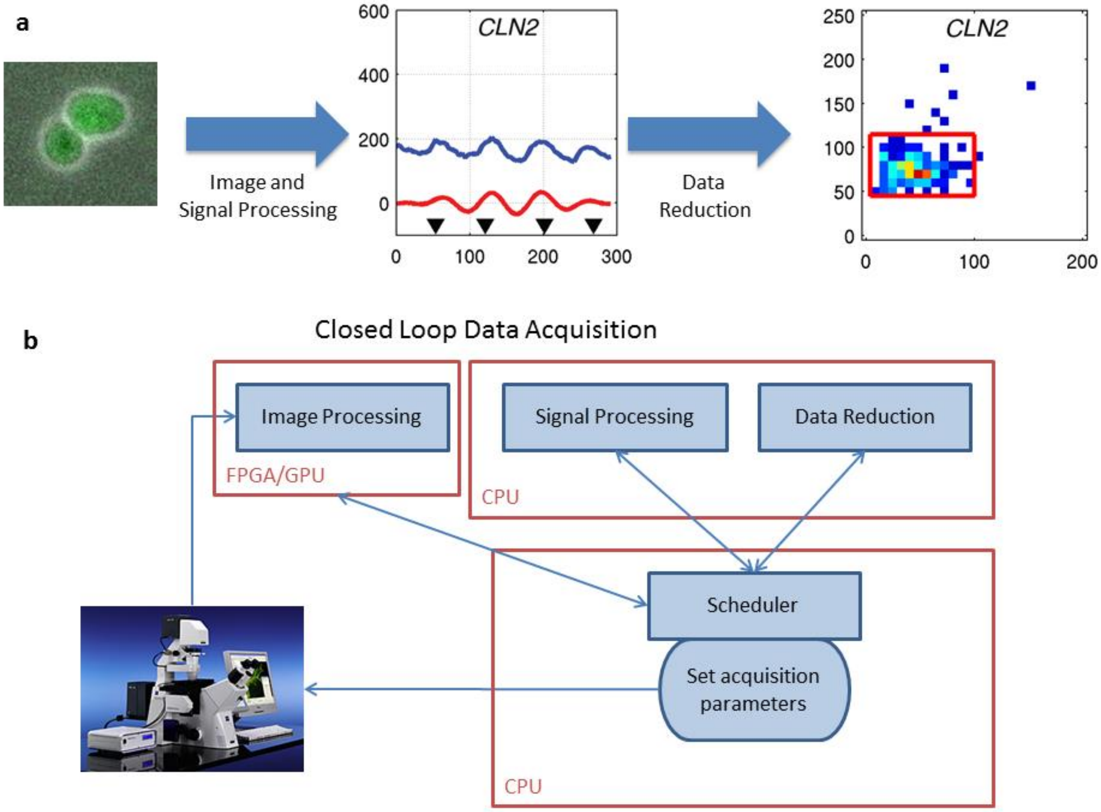

Figure 2.2: New Instruments Connect Design and Experiment.
(a) Using time-lapse microscopy for characterizing the dynamics of gene networks requires the development of custom suite of image and signal processing software along with data reduction algorithms. The mathematical models used to reduce movies into high-level statistics are necessarily related to the models used to design the gene network as ultimately experimental data need to be reconciled with model predictions. (b) Microscopy movies have traditionally been analyzed in a post-processing step. However, it is conceivable that in a near future the data analysis will be performed in real time by the computer controlling the microscope and the microfluidic system giving the user an experience similar to the use of a flow-cytometer. This information could also be used by the user to manually interact with the cells under observation. Alternatively, control algorithm could be developed to program the instrument to take specific actions such as changing the growth medium in response to specific behaviors of the cell populations.

The problem of collecting time-course data from inside living cells has been transformed by the emergence of fluorescent proteins. However, just like making an internal node in a chip accessible for test affects its performance, the maturation rate and protein half-life buffer fluorescence signals from the physiological events of interest [50]. Unfortunately, even parameters related to measurements such as the *in vivo* maturation rate of fluorescent proteins are difficult to determine accurately due to cell-to-cell variation and unknown dependencies on host strain, metabolic state, and environmental conditions. Also, the use of genes tagged with fluorescent domains is common, but the effects poorly understood. For example, how does a fluorescent tag affect the degradation rate of the fusion protein? Experimental work is needed to answer these questions.

New measurement devices will greatly facilitate the ability to determine *in vivo* parameter values and their dependencies on the continuously variable cellular environment. The emergence of time lapse fluorescent microscopy has allowed scientists to measure the dynamics of molecular mechanisms in individual cells [51]. This technique currently requires custom integration of optical equipment, image processing software, and microfluidic systems [52, 53]. Noise analysis connects the processed data to models of the underlying mechanisms [51]. Current image processing and analysis are typically post-processing steps, but these prototypes prefigure a new generation of instruments that will acquire raw images, process them in real time, and implement data reduction algorithms to extract high-level statistics for comparison with design-phase models (Figure 2.2). Microfluidic systems will allow for complicated input control of environmental parameters that should open avenues to applying some advanced testing techniques seen in EDA DFT, such as frequency response analysis.

2.3.4 Box 1. Software Integration

A crucial evolution in EDA has been the integration of previously independent design flow steps as the assumptions allowing for isolation have eroded. Some examples have been alluded to previously: device physics are infringing upon the Boolean assumption, layout and function are intermingling, and testing is becoming more interlinked with design. Beginning with the development of centralized CAD frameworks in the early 1980’s, this progression has led to integrated CAD tools in EDA [41, 66] that have evolved to be modular and able to communicate with one another. In GDA, design flow stages are intrinsically interlaced, and as such tools should be designed for integration rather than forcing them apart. In that perspective, the importance of ongoing efforts to develop open source software frameworks and data exchange standards like the Synthetic Biology Open Language (SBOL) (Table 2.1) cannot be underestimated [67, 68].
As the field matures, many of the GDA applications will need to be integrated into custom software stacks as is happening in mainstream bioinformatics [69]. Different integration models have different legal implications. One model consists of integrating data by accessing specialized web services. This model is illustrated by the rapid development of scientific workflow systems that facilitate this data integration [70, 71]. The model is attractive because the tools only need to share a common language like SBOL, saving the end-user the effort of installing, maintaining, and integrating different software components. Yet, there are many potential difficulties with this integration model: (i) dependance on computational services provided on a volunteer basis by a third party creates potential vulnerability (ii) moving large amounts of data to web-hosted services can be inefficient, and (iii) sending sensitive data to a third-party server may be undesirable. The other approach to software integration consists of integrating different applications that complement each other but execute on the end-user computational resources. This integration model raises some software licensing issues. In order to prevent corporations from segmenting GDA markets into proprietary silos like in EDA, it is prudent to foster the emergence of a vibrant GDA software development community.

In addition to avoiding market lockout resulting from proprietary software, it is crucial to ensure that the code base developed by the GDA community is free from hidden intellectual property claims and is licensed under permissive terms allowing academic and corporate stakeholders to reuse existing code bases.

2.3.5 Box 2. Outstanding questions

Standardized and designer parts. Standardization of genetic parts relies on the assumption that parts function is context-independent. This hypothesis greatly simplifies many aspects of the design process. Evidence of context-dependencies affecting parts function may limit the success of standardization efforts. Designing custom parts for use in a particular context is an alternative to standardization.

Designing for manufacturing. It is desirable to formalize manufacturing expertise as design rules that could be used to compare the cost of manufacturing functionally equivalent designs. Beyond the potential of substantial savings in manufacturing expenses, this effort is likely to uncover subtle relationships between the structure and functions of DNA sequences.

Designing for measurement. In order to ensure that a design can be validated it is important to integrate models of the measurement protocols in the design phase. Measurement strategies relying on fluorescent reporter genes introduce perturbations in the design that have been poorly characterized so far.

GDA as a collaboration hub. GDA can foster collaboration between specialists of different scientific and engineering domains. It is a social and intellectual challenge to develop languages and workflows that allow these specialists to communicate effectively.

2.4 A Scientific Renewal

Several authors recently argued that synthetic biology will lead to a better understanding of biology [54, 55]. In this spirit, we would like to question the assumption that the immediate value of GDA lies in its potential to accelerate the progress of experimental synthetic biologists. In the short term, efforts to develop GDA tools may be better justified as attempts to formalize genetic design principles. The assumption that models used in engineering can be extrapolated to biology can easily and rightfully be challenged by biologists. A more effective way of using GDA to engage the dialogue between engineers and life scientists might be to present GDA models as formal and compact representations of biological hypotheses. GDA then becomes a framework to express and test biological hypotheses, a form of scientific investigation common in the life sciences [56-58].

It is important to consider here two fundamental differences between EDA and GDA. The dynamics of genetic networks is largely determined by the interactions between large macromolecules confined to the small volume of a living cell. As a result, the dynamics of a genetic network are inherently stochastic in nature. There is even mounting evidence that many regulatory processes are based on molecular noise instead of merely attempting to mitigate its negative effects [59]. Electronic circuits use so many electrons that they behave deterministically. However, a consequence of the miniaturization and increasing power efficiency of silicon devices is a drastic reduction of the fluxes of electrons and a concomitant increase of the intrinsic electronic noise. It will be interesting to see if and how the EDA and GDA communities will work together to solve the problems associated with design automation of noisy systems.

Another important difference between GDA and EDA is that complexity in EDA was derived, while complexity in GDA was evolved. EDA has progressed by structured, rational improvements on the mathematical formalisms that express physical realities, incrementally allowing higher and higher complexity. As a result, the emergence of high-level function can always be traced to the lowest level components. On the other hand, since genetic systems evolved by random mutation, it is not clear that they follow rigorous design rules, and we cannot yet trace high-level function back to the low-level components in most cases. Synthetic biologists are walking in the footsteps of 50 years of effort by molecular geneticists to understand the design rules of genetic systems. Yet, the engineering mindset provides a new spark. The understanding of a genetic mechanism is truly put to the test when an engineer attempts to use the general principle to build something new. Formalization of these principles tests the theory and opens new areas of investigation when the theory is found lacking. A prime example of this is the aforementioned RBS calculator. Attempts to use “standard” ribosome binding sites failed and led to predictive thermodynamic models. The possibility to deoptimize the sequence of viral genes by taking advantage of codon pair bias [60, 61] is opening new research directions to better understand translation [62]. At a higher level of organization, the refactoring of the T7 genome resulted in reduced fitness that it is not completely understood [63].

In the current state, synthetic biology remains painfully slow, prohibitively expensive, and excessively labor-intensive. As an example, consider the progression from two 2002 theoretical papers on genetic oscillators [64, 65] to corresponding experimental publications in 2008 [19] and 2010 [20]. The unifying vision of a seamless GDA flow provides a collaborative framework for large interdisciplinary teams rather than relying on exceptional individual investigators familiar with all aspects of the design process. There is still a great deal of foundational work and biological discovery remaining before GDA materializes into suites of software tools facilitating design to specification entirely *in silico*. In the short term, closing some of the capability gaps will catalyze the emergence of more integrated teams that better handle the complex interdependencies between design, fabrication, and measurement. The tools these teams will generate may not have the elegance of an integrated solution, but they will provide new computational resources that should percolate beyond the confines of the synthetic biology community to benefit a larger population of life scientists.

2.5 Acknowledgements

This work is supported by NSF Awards 0850100 and 0963988 and by grant R01-GM078989 from the National Institutes of Health. MWL was supported by a fellowship from Science Application International Corporation and by a Department of Defense SMART Scholarship under OSD-T&E (Office of Secretary Defense-Test and Evaluation), Defense–Wide / PE0601120D8Z National Defense Education Program (NDEP) / BA-1, Basic Research. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript

2.6 References

1. Hill, A.D., *et al.* (2008) SynBioSS: the synthetic biology modeling suite. *Bioinformatics* 24, 2551-2553
2. Chandran, D., *et al.* (2009) TinkerCell: modular CAD tool for synthetic biology. *J Biol Eng* 3, 19
3. Czar, M.J., *et al.* (2009) Writing DNA with GenoCAD (TM). *Nucleic Acids Research* 37, W40-W47
4. Marchisio, M.A. and Stelling, J. (2008) Computational design of synthetic gene circuits with composable parts. *Bioinformatics* 24, 1903-1910
5. Rodrigo, G., *et al.* (2007) Asmparts: assembly of biological model parts. *Syst Synth Biol* 1, 167-170
6. Myers, C.J., *et al.* (2009) iBioSim: a tool for the analysis and design of genetic circuits. *Bioinformatics* 25, 2848-2849
7. Rialle, S., *et al.* (2010) BioNetCAD: design, simulation and experimental validation of synthetic biochemical networks. *Bioinformatics* 26, 2298-2304
8. Bilitchenko, L., *et al.* (2011) Eugene - a domain specific language for specifying and constraining synthetic biological parts, devices, and systems. *PLoS ONE* 6, e18882
9. Xia, B., *et al.* (2011) Developer's and User's Guide to Clotho v2.0 A Software Platform for the Creation of Synthetic Biological Systems. *Methods Enzymol.* 498, 97-135
10. Pedersen, M. and Phillips, A. (2009) Towards programming languages for genetic engineering of living cells. *J. R. Soc. Interface* 6 Suppl 4, S437-450
11. Purnick, P.E.M. and Weiss, R. (2009) The second wave of synthetic biology: from modules to systems. *Nat. Rev. Mol. Cell Biol.* 10, 410-422
12. Kwok, R. (2010) Five hard truths for synthetic biology. *Nature* 463, 288-290
13. Marchisio, M.A. and Stelling, J. (2009) Computational design tools for synthetic biology. *Curr. Opin. Biotechnol.* 20, 479-485
14. Clancy, K. and Voigt, C.A. (2010) Programming cells: towards an automated 'Genetic Compiler'. *Current Opinion in Biotechnology* 21, 572-581
15. MacDonald, J.T., *et al.* (2011) Computational design approaches and tools for synthetic biology. *Integr. Biol.* 3, 97-108
16. Alterovitz, G., *et al.* (2010) The challenges of informatics in synthetic biology: from biomolecular networks to artificial organisms. *Briefings in bioinformatics* 11, 80-95

17. Voigt, C.A., ed (2011) *Synthetic Biology Parts B: Computer Aided Design and DNA Assembly*. Academic Press
18. Elowitz, M.B. and Leibler, S. (2000) A synthetic oscillatory network of transcriptional regulators. *Nature* 403, 335-338
19. Stricker, J., *et al.* (2008) A fast, robust and tunable synthetic gene oscillator. *Nature* 456, 516-519
20. Danino, T., *et al.* (2010) A synchronized quorum of genetic clocks. *Nature* 463, 326-330
21. Khalil, A.S. and Collins, J.J. (2010) Synthetic biology: applications come of age. *Nat Rev Genet* 11, 367-379
22. Ball, D.A., *et al.* (2010) Co-design in synthetic biology: a system-level analysis of the development of an environmental sensing device. *Pac. Symp. Biocomput.*, 385-396
23. Benfey, P.N. and Mitchell-Olds, T. (2008) Perspective - From genotype to phenotype: Systems biology meets natural variation. *Science* 320, 495-497
24. Sangiovanni-Vincentelli, A. (2003) The tides of EDA. *Ieee Design & Test of Computers* 20, 59-75
25. Myers, C.J., *et al.* (2009) Genetic design automation. In *ICCAD '09 Proceedings of the 2009 International Conference on Computer-Aided Design* (Roychowdhury, J., ed), pp. 713-716, ACM
26. Gendrault, Y., *et al.* (2011) Synthetic biology methodology and model refinement based on microelectronic modeling tools and languages. *Biotechnol J* 6, 796-806
27. Pêcheux, F.M., M.; Lallement, C. (2010) Is SystemC-AMS an appropriate "promoter" for the modeling and simulation of bio-compatible systems? *Proceedings of 2010 IEEE International Symposium on Circuits and Systems (ISCAS)*, 1791-1794
28. Cai, Y., *et al.* (2009) Modeling structure-function relationships in synthetic DNA sequences using attribute grammars. *PLoS Comput Biol* 5, e1000529
29. Canton, B., *et al.* (2008) Refinement and standardization of synthetic biological parts and devices. *Nat. Biotechnol.* 26, 787-793
30. Kelly, J.R., *et al.* (2009) Measuring the activity of BioBrick promoters using an in vivo reference standard. *J Biol Eng* 3, 4
31. de Smit, M.H. and van Duin, J. (1990) Secondary structure of the ribosome binding site determines translational efficiency: a quantitative analysis. *Proc Natl Acad Sci U S A* 87, 7668-7672
32. Salis, H.M., *et al.* (2009) Automated design of synthetic ribosome binding sites to control protein expression. *Nat. Biotechnol.* 27, 946-950
33. Na, D., *et al.* (2010) Mathematical modeling of translation initiation for the estimation of its efficiency to computationally design mRNA sequences with desired expression levels in prokaryotes. *BMC Syst Biol* 4, 71
34. Proshkin, S., *et al.* (2010) Cooperation between translating ribosomes and RNA polymerase in transcription elongation. *Science* 328, 504-508
35. Kawamata, I., *et al.* (2009) Automatic design of DNA logic gates based on kinetic simulation. In *DNA Computing and Molecular Programming 15th International Conference, DNA 15* (Deaton, R. and Suyama, A., eds), pp. 88-96, Springer
36. Del Vecchio, D., *et al.* (2008) Modular cell biology: retroactivity and insulation. *Mol Syst Biol* 4, 161

37. Kim, K.H. and Sauro, H.M. (2011) Measuring retroactivity from noise in gene regulatory networks. *Biophys J* 100, 1167-1177
38. Czar, M.J., *et al.* (2009) Gene synthesis demystified. *Trends Biotechnol.* 27, 63-72
39. Goler, J.A., *et al.* (2008) Genetic design: rising above the sequence. *Trends Biotechnol.* 26, 538-544
40. Endy, D. (2005) Foundations for engineering biology. *Nature* 438, 449-453
41. Scheffer, L., *et al.* (2006) *EDA for IC implementation, circuit design, and process technology*. CRC Taylor & Francis
42. Anderson, J.C., *et al.* (2010) BglBricks: A flexible standard for biological part assembly. *J Biol Eng* 4, 1
43. Gibson, D.G. (2011) Enzymatic assembly of overlapping DNA fragments. *Methods Enzymol* 498, 349-361
44. Oliveira, P.H., *et al.* (2009) Structural instability of plasmid biopharmaceuticals: challenges and implications. *Trends Biotechnol.* 27, 503-511
45. Cai, Y., *et al.* (2010) GenoCAD for iGEM: a grammatical approach to the design of standard-compliant constructs. *Nucleic Acids Res* 38, 2637-2644
46. Densmore, D., *et al.* (2010) Algorithms for automated DNA assembly. *Nucleic Acids Res.* 38, 2607-2616
47. Scheffer, L.K., *et al.* (2006) *EDA for IC system design, verification, and testing*. CRC Taylor & Francis
48. Kreutz, C. and Timmer, J. (2009) Systems biology: experimental design. *FEBS J* 276, 923-942
49. Apgar, J.F., *et al.* (2010) Sloppy models, parameter uncertainty, and the role of experimental design. *Mol Biosyst* 6, 1890-1900
50. Wang, X., *et al.* (2008) Mathematical analysis and quantification of fluorescent proteins as transcriptional reporters. *Biophysical Journal* 94, 2017-2026
51. Locke, J.C.W. and Elowitz, M.B. (2009) Using movies to analyse gene circuit dynamics in single cells. *Nature Reviews Microbiology* 7, 383-392
52. Bennett, M.R. and Hasty, J. (2009) Microfluidic devices for measuring gene network dynamics in single cells. *Nat Rev Genet* 10, 628-638
53. Charvin, G., *et al.* (2010) Origin of irreversibility of cell cycle start in budding yeast. *PLoS Biol* 8, e1000284
54. Elowitz, M. and Lim, W.A. (2010) Build life to understand it. *Nature* 468, 889-890
55. Bashor, C.J., *et al.* (2010) Rewiring Cells: Synthetic Biology as a Tool to Interrogate the Organizational Principles of Living Systems. *Annual Review of Biophysics, Vol 39* 39, 515-537
56. Kell, D.B. and Oliver, S.G. (2004) Here is the evidence, now what is the hypothesis? The complementary roles of inductive and hypothesis-driven science in the post-genomic era. *Bioessays* 26, 99-105
57. King, R.D., *et al.* (2004) Functional genomic hypothesis generation and experimentation by a robot scientist. *Nature* 427, 247-252
58. Evans, J. and Rzhetsky, A. (2010) Machine Science. *Science* 329, 399-400
59. Balazsi, G., *et al.* (2011) Cellular decision making and biological noise: from microbes to mammals. *Cell* 144, 910-925
60. Coleman, J.R., *et al.* (2008) Virus attenuation by genome-scale changes in codon pair bias. *Science* 320, 1784-1787

61. Mueller, S., *et al.* (2010) Live attenuated influenza virus vaccines by computer-aided rational design. *Nat. Biotechnol.* 28, 723-U1729
62. Plotkin, J.B. and Kudla, G. (2011) Synonymous but not the same: the causes and consequences of codon bias. *Nat. Rev. Genet.* 12, 32-42
63. Chan, L.Y., *et al.* (2005) Refactoring bacteriophage T7. *Mol Syst Biol* 1, 2005.0018
64. Hasty, J., *et al.* (2002) Synthetic gene network for entraining and amplifying cellular oscillations. *Phys Rev Lett* 88, 148101
65. McMillen, D., *et al.* (2002) Synchronizing genetic relaxation oscillators by intercell signaling. *Proc Natl Acad Sci U S A* 99, 679-684
66. Barnes, T.J. (1992) *Electronic CAD frameworks*. Kluwer Academic Publishers
67. Galdzicki, M., *et al.* (2011) Standard biological parts knowledgebase. *PLoS ONE* 6, e17005
68. Peccoud, J., *et al.* (2011) Essential information for synthetic DNA sequences. *Nat. Biotechnol.* 29, 22; discussion 22-23
69. Stajich, J.E. and Lapp, H. (2006) Open source tools and toolkits for bioinformatics: significance, and where are we? *Briefings in bioinformatics* 7, 287-296
70. McPhillips, T., *et al.* (2009) Scientific workflow design for mere mortals. *Future Gener Comp Sy* 25, 541-551
71. Shon, J., *et al.* (2008) Scientific workflows as productivity tools for drug discovery. *Current opinion in drug discovery & development* 11, 381-388
72. Gardner, T.S., *et al.* (2000) Construction of a genetic toggle switch in Escherichia coli. *Nature* 403, 339-342

Chapter 3: Co-design in synthetic biology: a system-level analysis of the development of an environmental sensing device

Authors

David A. Ball[1*], Matthew W. Lux[1*], Russell R. Graef[2], Matthew W. Peterson[2], Jane D. Valenti[1], John Dileo[2], Jean Peccoud[1]

[1]Virginia Bioinformatics Institute, Virginia Polytechnic Institute and State University, Blacksburg, VA (USA)
[2]Emerging Technologies Office, The MITRE Corporation, McLean, VA (USA)
** These authors contributed equally to this work.*

Abstract

The concept of co-design is common in engineering, where it is necessary, for example, to determine the optimal partitioning between hardware and software of the implementation of a system features. Here we propose to adapt co-design methodologies for synthetic biology. As a test case, we have designed an environmental sensing device that detects the presence of three chemicals, and returns an output only if at least two of the three chemicals are present. We show that the logical operations can be implemented in three different design domains: (1) the transcriptional domain using synthetically designed hybrid promoters, (2) the protein domain using bi-molecular fluorescence complementation, and (3) the fluorescence domain using spectral unmixing and relying on electronic processing. We discuss how these heterogeneous design strategies could be formalized to develop co-design algorithms capable of identifying optimal designs meeting user specifications.

Publication status

This manuscript is reproduced here, with permission, from the original publication at:

Ball, D.A., et al. (2010) Co-design in synthetic biology: a system-level analysis of the development of an environmental sensing device. Pac. Symp. Biocomput., 385-396

3.1 Introduction

3.1.1 The need for co-design of synthetic biology systems

A major focus in the field of synthetic biology from the field's inception has been the design of biological "devices" [1]. To date, many biological analogs of common electronics parts have been developed. Some notable biological devices analogous to those commonly used for electronics system design include logic gates [2-4], asynchronous logic components [5], switches [6-8], oscillators [8-12], memory circuits [13] and most recently genetic counters [14]. While these devices have been excellent proofs-of-concept for the application of traditional engineering design to the design of biological systems, little progress has been made in the design of larger, more complex, biological constructs [15]. This was seen in 2004, when Blue Heron Biotechnology did not receive a single submission to their Big DNA Contest, which offered free synthesis of the most "interesting" DNA construct of over 40kb in length (http://tinyurl.com/bigdna).

The designs of early artificial gene networks were restricted to a single design domain: protein-DNA interactions regulating transcription [6,10,16]; intra-molecular interactions within RNA molecules [3,17-20], or even interaction between DNA molecules [21,22]. More recent publications however, report the combined use of multiple domains in a single design. The genetic counter [14] is a good example of a heterogeneous design combining circuitry operating in different domains. Two separate methodologies were used, resulting in counter devices that, while performing the same type of logic (being able to count to either two or three), were appropriate for different uses. The riboregulated transcriptional cascade (RTC) counter utilized a fast transcriptional cascade for counting. The DNA invertase cascade (DIC) counter used recombinases upstream of inverted promoters to count. Due to the dynamics of DNA recombination, these counters activate more slowly, and as such can be used only for the counting of low-frequency events. It is anticipated that future heterogeneous designs will also include non-biological elements, as well as biological circuits. One such example of this is an AND gate used to regulate protein folding, which utilized UV light and ATP as stimuli [23]. By including non-biological elements, the design space is increased, allowing for the design of more complex, and potentially more useful, constructs. In the AND gate example above, the wavelength and intensity of the UV light is an additional parameter that can be used to optimize the system. Similarly, one approach described here uses advanced detection to expand the list of viable reporters.

As synthetic biology matures, it becomes necessary to develop more sophisticated design strategies. The design of an industrial application calls for a systematic comparison of different possible designs meeting the user specification in order to identify an optimal design maximizing one or several figures of merit. Besides the design correctness (the design does what it is supposed to do), its performance, development cost, manufacturing cost, or reconfigurability are other criteria that an engineering team may need to optimize. When developing electronic systems, an important design decision is the partitioning of features implemented in hardware and those implemented in software, which is known as the hardware/software co-design problem [24,25]. Software ensures rapid time to market, design flexibility, and runs on inexpensive processors produced in large volumes. However, the development of application-specific circuits is needed when software running on generic processors cannot meet the required performance. The hardware/software co-design problem illustrates the issues arising when heterogeneous technologies are combined into a system. At a very high-level, the design of synthetic biology applications includes a wetware component (the living organisms), but also a hardware component represented by the instrument and software used to acquire and process signals generated by the biological component of the system. At a higher resolution, the wetware component is itself heterogeneous since the transcriptional, translational, and proteomic components of this machinery represent different design domains.. By leveraging methods acquired by electrical engineers to develop heterogeneous systems, the complexity, efficiency, and flexibility of synthetic biology applications will likely be dramatically increased, while reducing the production costs of these systems.

3.1.2 An environmental sensor as test case for co-design analysis

In the field of biosecurity there is an increasing recognition of the need for systems that can rapidly detect pollutants, contaminants, and biothreat agents (pathogenic bacteria, viruses and toxins) in food, agricultural products, pharmaceuticals, and environmental samples. For example, the safety of the US food supply is an ongoing concern because of potential impacts of contamination on both public health and the US economy. In addition to inadvertent contamination, concerns about a bioterrorism event such as the intentional introduction of pathogens and/or toxins, brings a new dimension to this problem.

Cells possess innate abilities that make them ideal for environmental sensing applications. Specifically, they are inherently able to detect small concentrations (parts per billion) of chemicals (or combinations of chemicals) in their environment and respond to it, usually with an amplified signal. Cells can be programmed by identifying three functional layers: an input layer, an information processing layer, and an output layer [26]. This abstract representation of the environmental sensing chain can help design environmental sensing devices relying on biological systems for transforming chemical information into electrical signals that can be recorded and processed by computer systems. For instance, a situational awareness monitoring system will rely on a network of geographically dispersed sensing units communicating chemical data to a central server or to personnel operating in their vicinity. In this scenario, the sensing units should be capable of some basic processing of chemical data in order to communicate informative data.

In many cases, the presence of individual molecules in the environment is not informative while the simultaneous presence of two molecules can provide valuable information worthy communicating. In the field of defense, many chemical agents and explosives are made from combinations of commonly available industrial chemicals. For example, mustard gas can be created with thiodiglycol, an industrial solvent used in dyes and other applications, and phosphorus trichloride, a common industrial chemical used to manufacture a wide range of organic phosphorous compounds. Rapid field detection of such combinations could be an important application of sensing devices [27]. In ecology, it is well established that levels of heavy metals can be below the safe threshold individually, but combine to be lethal in fish [28]. In human health, it was recently shown that the common herbicide Roundup is more toxic in the presence of its supposedly inert adjuvants, and has negative effects in pregnant women even at “safe” levels of the active ingredient [29]. These findings may lead to the elucidation of further chemicals that are safe alone, but unsafe together. Detection of such combinations could become important to assess environmental or security threats.

Input 1	Input 2	Input 3	Output 1	Output 2	Output 3
-	-	-	-	-	-
-	-	+	-	-	-
-	+	-	-	-	-
-	+	+	-	+	-
+	-	-	-	-	-
+	-	+	-	-	+
+	+	-	+	-	-
+	+	+	+	+	+

Table 3.1: Logic table of the environmental sensor

The following sections describe three possible methods for implementing a cell based system designed to be able to detect the presence of each pair of three different chemical inputs and produce three different electronic outputs in response. Formally, the system can be specified by a truth table (Table 3.1). These designs differ in where the logic is implemented in the system (Figure 3.1). In the first option, hybrid promoters that contain binding sites for transcription factors responsive to the inputs are used to control the expression of fluorescent proteins. Only when the proper inputs are present will reporter genes be expressed, thus the logic occurs at the transcriptional level. The second option is to implement the logic at the protein level. This is accomplished by coupling each input to the expression of a non-fluorescent fragment of a fluorescent protein. Only when the two proper inputs are present will the fragments associate to generate a fluorescent signal. A final option is to embed the logic in the electronic layer. In this case each input directly activates the expression of one of three different fluorescent proteins and the inputs present are determined by processing the pattern of fluorescence that is obtained. For all three scenarios, the fluorescent proteins used were cyan (CFP), yellow (YFP), and red (RFP), which are all easily separated from each other.

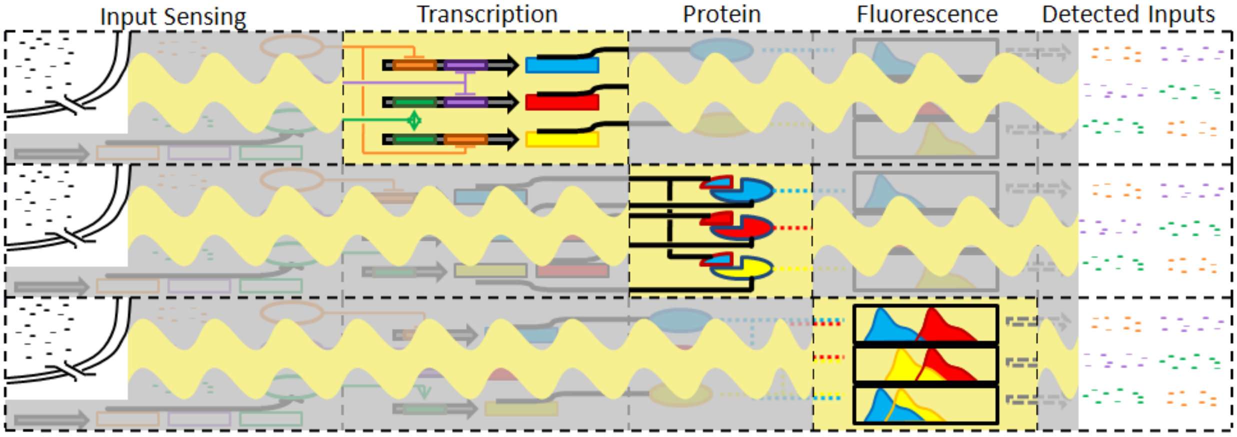

Figure 3.1: Implementation of logic in different design domains.
The figure gives an overview of how each approach processes the environmental inputs. Wavy yellow lines indicate signal transduction, and yellow boxes highlight where the logic occurs. The details of each design are presented in Figure 3.2, Figure 3.3, and Figure 3.4.

3.2 Results

3.2.1 Solution 1: Hybrid Promoters

3.2.1.1 Theoretical foundation

One approach is to embed the design logic into transcriptional control with hybrid promoters. Since the input to the system is a set of small molecules, the first step is to sense the presence of the small molecule in the environment. In the described situation, these small molecules are ligands capable of binding to specific transcription factors. The sensor function is therefore accomplished by constitutive expression of the corresponding transcription factors. This sensing mechanism is conserved through each approach.

Once bound by its ligand, the behavior of each transcription factor is altered. For example, some ligand-bound repressors can no longer bind to their corresponding promoters and exert their repressive properties. Thus, each transcription factor can be thought to switch on or off depending on the presence or absence of ligand.

In order to implement the design logic, the on/off state of these transcription factors must be processed. The hybrid promoter approach accomplishes the logic by controlling expression of a reporter gene through a promoter that responds to the state of pairs of transcription factors. That is, promoter A responds to the on/off state of transcription factors 1 and 2, promoter B to transcription factors 2 and 3, and promoter C to transcription factors 1 and 3. Each promoter responds only when both transcription factors have been toggled by the presence of the ligand molecule. The name “hybrid promoters” derives from the fact that they are engineered to respond to multiple transcription factors.

As a more detailed explanation, let us assume that inputs 1 and 2 are repressor proteins that repress only in the absence of their respective ligand. With no ligand present, both transcription factors effectively repress transcription of the reporter gene controlled by promoter A. In the presence of ligand 1 only, transcription factor 1 loses its ability to bind to promoter A, but transcription is still blocked by transcription factor 2. Likewise, if only ligand 2 is present, transcription factor 1 continues to block transcription of the reporter gene. However, in the presence of both ligand 1 and ligand 2, neither transcription factor can bind to promoter A, and the reporter gene is freely expressed.

The transcription factor does not have to be an inducible repressor to accomplish the appropriate logic. There are 4 possible transcription factor responses: (1) the transcription factor *represses* only in the *absence* of its ligand, (2) the transcription factor *represses* only in the *presence* of its ligand, (3) the transcription factor *activates* only in the *absence* of its ligand, and (4) the transcription factor *activates* only in the *presence* of its ligand. Only options (1) & (4) are viable for implementing the design logic because they do not invert the signal. For example, a positive signal from the presence of a ligand would become a negative signal if the induced transcription factor transitioned from an inactive repressor (therefore allowing transcription) to an active repressor (therefore blocking transcription). Put another way, the positive (+) input signal would be inverted by *activation of the repressor* (-) to block expression from the promoter (+/- = -). On the other hand, the positive (+) input signal would be conserved by *inactivation of the repressor* (+) to allow expression from the promoter (+/+ = +), which preserves the signal.

Figure 3.2 illustrates an implementation of the logic using specific transcription factors and ligands. The rationale behind the choice of specific parts is described below.

3.2.1.2 Proposed design

A number of features need to be considered in the selection of appropriate inducible transcription factors. These features include: high range of control, compatibility with other transcription factors, and prior use in other applications. High range of control is necessary to ensure that the final signal is detected over the cellular noise. Compatibility with other transcription factors refers to the design of the actual promoter. For example, activators frequently must bind to specific promoter regions and thus may prevent the use of a second transcription factor that must bind an overlapping site. Last, well characterized transcription factors that have been widely used in other designs are preferred.

In synthetic biology, the list of commonly used genes is small and thus selection of appropriate parts is restricted. The first two appropriate transcription factors that match the criteria are LacI, which is inducible by isopropyl β-D-1-thiogalactopyranoside (IPTG), and TetR which is inducible by anhydrotetracycline (aTc). Both have operator sites that can be effectively placed in multiple locations to prevent interference with other transcription factors and both have been shown to have a high range of control [30,31].

Furthermore, hybrid promoters under the simultaneous control of LacI and TetR were described previously [31] which matched our logic criteria.

The selection of the third transcription factor is not as straightforward. LuxR is an attractive candidate. While LacI and TetR induce transcription by derepression in the presence of their respective ligand, LuxR activates transcription in the presence of its ligand acyl-homoserine lactone (AHL). LuxR has a high range of control for the wild type promoter [32]; it is described extensively in the literature, and it is commonly used in synthetic biology applications [26,33-35]. Although previous attempts to design hybrid promoter responding to LuxR/LacI or LuxR/TetR proved unsuccessful [31]. Nevertheless, a careful investigation of these LuxR hybrid promoters shows that none of them had spacing between the -10 box and the LuxR binding site that was identical to the wild type promoter. Given that this spacing has been shown to be important to ensure proper regulation of gene expression [32], we predict LuxR hybrid promoters can be redesigned if the suitable spacing is used.

The first promoter for implementing the environmental sensing device uses the sequence of the promoter A90 responding to LacI/TetR [31]. For the promoters responding to LuxR, we modified the wild type promoter for *luxI* and added *lac* and *tet* operators, respectively, downstream of the -10 box. Specifically, we replaced the sequence downstream of the -10 sequence in the wild type promoter with sequence downstream of the -10 box taken from promoters successfully responding to LacI and TetR [31]. Since the designed promoters already contain sufficient spacing downstream of +1, the 3 full sequences were assembled by simply adding a ribosome binding site sequence, a coding sequence for three different fluorescent proteins (CFP, YFP, RFP), and a terminator. Figure 3.1 shows the specific combinations of hybrid promoter and fluorescent reporter gene.

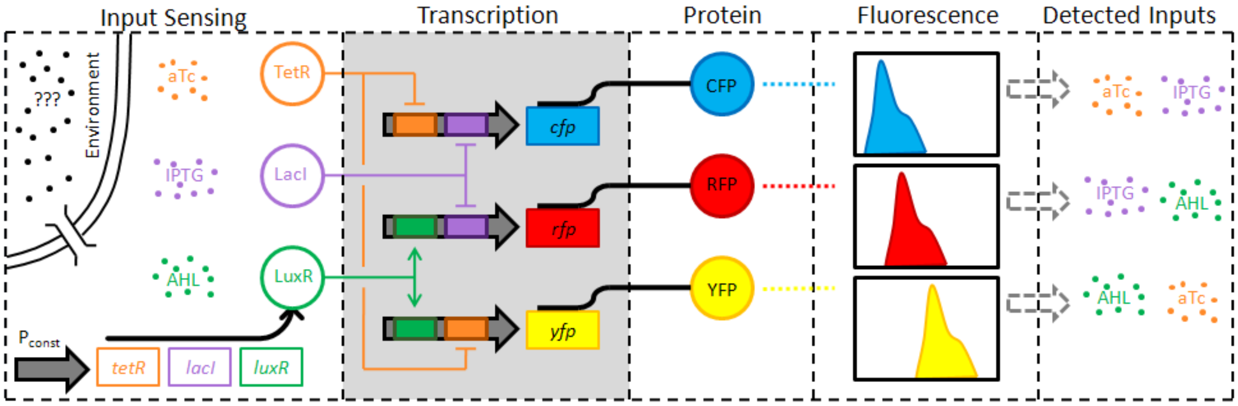

Figure 3.2: Hybrid Promoter Approach.
The system logic is implemented in the control of transcription of 3 reporter genes. See text for detailed explanation of how the logic is processed. Orange indicates the aTc signal, purple the IPTG signal, green the AHL signal, blue the CFP signal, red the RFP signal, and yellow the YFP signal. Dots indicate small molecule inducers, thick solid arrows indicate promoters with color-coded operator boxes inside and adjacent boxes indicating genes, thick black lines indicate production of protein from a gene, circles indicate proteins, dotted lines indicate fluorescence measurement, and the gray box indicates where the logic occurs.

3.2.2 Solution 2: Fluorescence complementation

3.2.2.1 Theoretical foundation

While transcriptional logic is prevalent in the majority of synthetic biology constructs, logic can also be performed in the protein domain. As previously reported, the anti-parallel Leucine zipper mediated direction of protein reassembly allows for the reconstitution of intact and functional GFP [36]. Further research has shown the ability to adapt protein reassembly of fluorescent proteins to visualize protein-protein interactions *in vivo* [37,38]. These methods describe “Bimolecular Fluorescence Complementation” (BiFC) which uses non-fluorescent fragments of fluorescent proteins bound to separate functional proteins; when interaction between these proteins takes place the non-fluorescent pairs combine to produce a fluorescent complex. In this second solution, inputs are sensed in the same manner as in Solution 1. However, here the promoters respond to single transcription factors and therefore a single environmental input. Hence, they simply transmit the signal to the protein domain. The information is processed by coupling non-fluorescent halves of fluorescent proteins. This makes an “AND” gate from the pairing of non-fluorescent halves to produce a final product of fluorescence dependent on the promoters engaged. The logic comes not only from which fragments are produced but from the fact that only certain combinations of fragments will produce a detectable fluorescence output. Fragment 1 (N-terminal) of GFP combines with fragment 2 (C-terminal) of GFP to form functional GFP, in contrast fragment 1 (N-terminal) of GFP cannot combine with fragment 1 (N-terminal) of CFP.

3.2.2.2 Design

In order to implement this method of logic processing, several factors must be considered. Dissection sites of CFP, YFP, and a monomeric form of DsRed, a Red Fluorescent Protein variant that yield two non-fluorescent halves capable of reassembly into fluorescent proteins have been previously reported [37]. CFP is split into two fragments at amino acid 155 yielding a CFP 1-155 (N-terminal) fragment and a CFP 155-239 (C-terminal) fragment, subsequently referred to as CFP-N and CFP-C respectively. Similarly DsRed and YFP are split into the following fragments: RFP-N (residues 1-168), RFP-C (residues 169-225), and YFP-N (residues 1-154). The C-terminal of YFP is not required, because YFP-N can combine with CFP-C to form a species that produces yellow fluorescence [37,38].

These fragments will be cloned downstream of three inducible promoters. Fragments CFP-N and RFP-C are placed under the control of LacI (inducible by IPTG), Fragments YFP-N and RFP-N are placed under the control of LuxR (inducible by AHL), and the final fragment CFP-C is placed under TetR (inducible by aTc) control. As shown in Figure 3.3, the expected outputs from this system are dependent on which fragments are expressed. Some factors which play a role in total fluorescence are: the concentration of the inducer, the relative strength of promoter, and the fragment complementation.

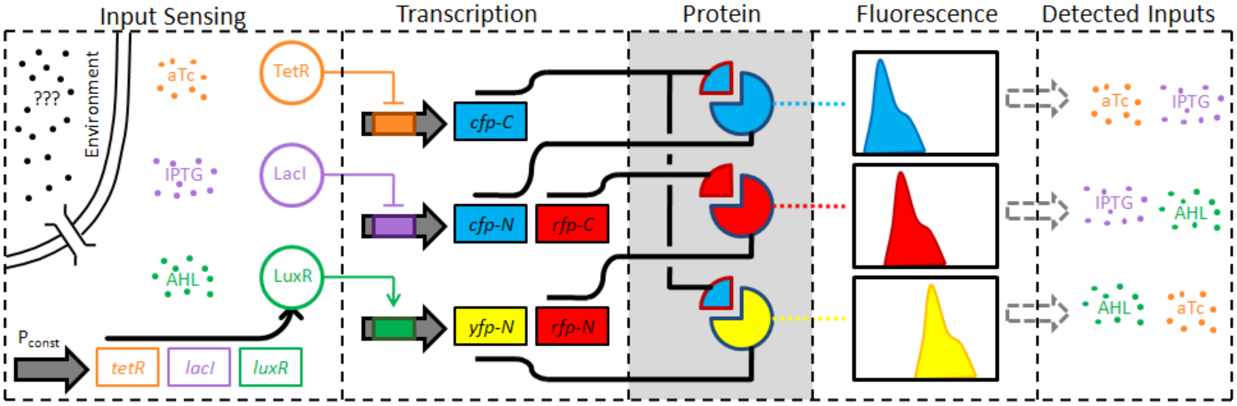

Figure 3.3: Fluorescence Complementation Approach.
The system logic is implemented in the complementation of 3 fluorescent proteins. See text for detailed explanation of how the logic is processed. Orange indicates the aTc signal, purple the IPTG signal, green the AHL signal, blue the CFP signal, red the RFP signal, and yellow the YFP signal. Dots indicate small molecule inducers, thick solid arrows indicate promoters with color-coded operator boxes inside and adjacent boxes indicating genes, thick black lines indicate production of protein from a gene, circles indicate proteins, dotted lines indicate fluorescence measurement, and the gray box indicates where the logic occurs.

Logic circuit processing may be carried out by protein reassembly mediated by inducible promoters. For future experimentation, synthetically designed proteins for binding novel small and macro molecules could be incorporated into the system outlined here for logical processing.

Fragment	CFP-N	CFP-C	RFP-N	RFP-C	YFP-N
CFP-N	N/C	BLUE	N/C	N/C	N/C
CFP-C	BLUE	N/C	N/C	N/C	YELLOW
RFP-N	N/C	N/C	N/C	RED	N/C
RFP-C	N/C	N/C	RED	N/C	N/C
YFP-N	N/C	YELLOW	N/C	N/C	N/C

Table 3.2: Fluorescent output for complementation fragments.
Previously reported fluorescent outputs generated by complementation of the different non-fluorescent protein fragments resulting in stable fluorescent protein complexes (given in color and name) or unstable/incompatible fragments (given by N/C "no color") [37].

3.2.3 Solution 3: Unmixing of fluorescence spectra

3.2.3.1 Theoretical foundation

For the simultaneous detection of 3 (or fewer) fluorescent proteins it is possible to find fluorescent proteins with suitably separated excitation and emission spectra, such that the 3 colors can be distinguished by the use of optical band-pass filters [39]. However, this becomes difficult, if not impossible, as the number of fluorescent proteins increases. Therefore, a spectral unmixing approach was developed for the detection of multiple fluorescent signals [40-42]. The use of spectral unmixing also allows the use of a wider range of fluorescent proteins, which is of great value when it is necessary to use fluorophores with similar properties such as maturation and degradation times.
Spectral unmixing relies on the *a priori* collected emission spectra of the individual fluorescent proteins in the system to determine which fluorophores have contributed to the observed signal. The experimental output spectrum, *F*, can be described by the system of linear equations:

$$F = \mathbf{X}A, \quad (1)$$

where the *m* data points in the output spectrum, *F*, and the weights of the *n* individual fluorophores, *A* are column vectors and X_{ij} is the ith point in the spectrum of fluorophore j,

$$F = \begin{pmatrix} F_1 \\ F_2 \\ \vdots \\ F_m \end{pmatrix}, \quad A = \begin{pmatrix} A_1 \\ A_2 \\ \vdots \\ A_n \end{pmatrix}, \quad \mathbf{X} = \begin{pmatrix} X_{11} & X_{12} & \cdots & X_{1n} \\ X_{21} & X_{22} & \cdots & X_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ X_{m1} & X_{m2} & \cdots & X_{mn} \end{pmatrix}. \quad (2)$$

If *m* > *n*, Eq. 1 describes an overdetermined system, which can be solved for *A* by a least squares fitting algorithm to minimize $F - \mathbf{X}A$.

3.2.3.2 Design

Biologically, spectral unmixing offers the simplest approach for implementing the logic required to detect combinations of environmental species, as it does not require any interactions between the various promoters or protein products. As illustrated in Figure 3.4, each input ligand triggers the production of a single fluorescent protein. The sensing mechanism is again the same as Solution 1. In this case, the input signal is transmitted all the way to the spectral detection via single-operator promoters controlling production of single fluorescent proteins. The various fluorescence components can be extracted by use of Eq, (1), and the concentrations of each chemical can be determined.

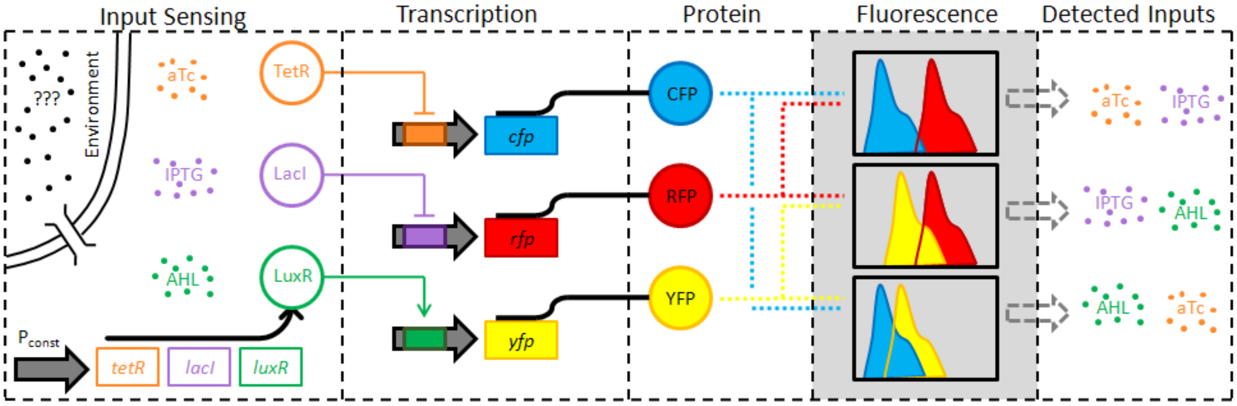

Figure 3.4: Spectral Detection Approach.
The system logic is implemented in the unmixing of the measured spectra. See text for detailed explanation of how the logic is processed. Orange indicates the aTc signal, purple the IPTG signal, green the AHL signal, blue the CFP signal, red the RFP signal, and yellow the YFP signal. Dots indicate small molecule inducers, thick solid arrows indicate promoters with color-coded operator boxes inside and adjacent boxes indicating genes, thick black lines indicate production of protein from a gene, circles indicate proteins, dotted lines indicate fluorescence measurement, and the gray box indicates where the logic occurs.

3.2.3.3 Preliminary data

A spectrofluorimeter is required to collect the needed data to unmix the contributions of multiple fluorescent proteins. For the experiments reported here, a NanoDrop 3300 (Thermo Scientific, Wilmington DE) was used. The NanoDrop System uses a small sample volume of 2 µL. It contains three light-emitting diodes (LEDs) for fluorescence excitation: UV (peak emission at 365 nm), blue (470 nm) and white (460-650 nm). The collected fluorescence is dispersed over a 1024-pixel linear CCD, and allows the collection of wavelengths from 400-750 nm with a 4 nm resolution. Because the system does not include any optical bandpass filters, it is necessary to obtain spectra for blank samples, cells similar to those being used in the experiment but lacking any fluorophores. Preliminary experiments were performed with four constitutively expressed fluorescent proteins: EGFP [43], acGFP [44], vYFP [45], and Citrine [46], each in a separate cell-line, to test the ability of the spectral unmixing algorithm to separate fluorescent proteins with similar emission spectra. Figure 3.5 presents the reference spectra for these four fluorophores. The spectra show that this combination of fluorescent proteins can be regarded as a worst-case scenario, as none of the four could be distinguished with the use of optical band-pass filters.

To account for differences in expression levels of the four different fluorescent proteins, the fluorescence spectra of the four cell-lines were normalized by their optical absorbance at 600 nm. The normalized spectra were then used as the inputs to the unmixing algorithm, so that the extracted coefficients, such as those recorded in Table 3.3, are also in units of absorbance.

The four cell cultures were then mixed at known concentrations, and the resulting spectra shown in Figure 3.5b-f were analyzed with the spectral unmixing algorithm in order to extract the components of the mixtures. The measured optical densities at 600 nm for the existing cells in each mixture along with the extracted values are given Table 3.3. In all cases, there were no false-negatives. There are some false positive results, however, but their coefficients remained small (value less than 0.006).

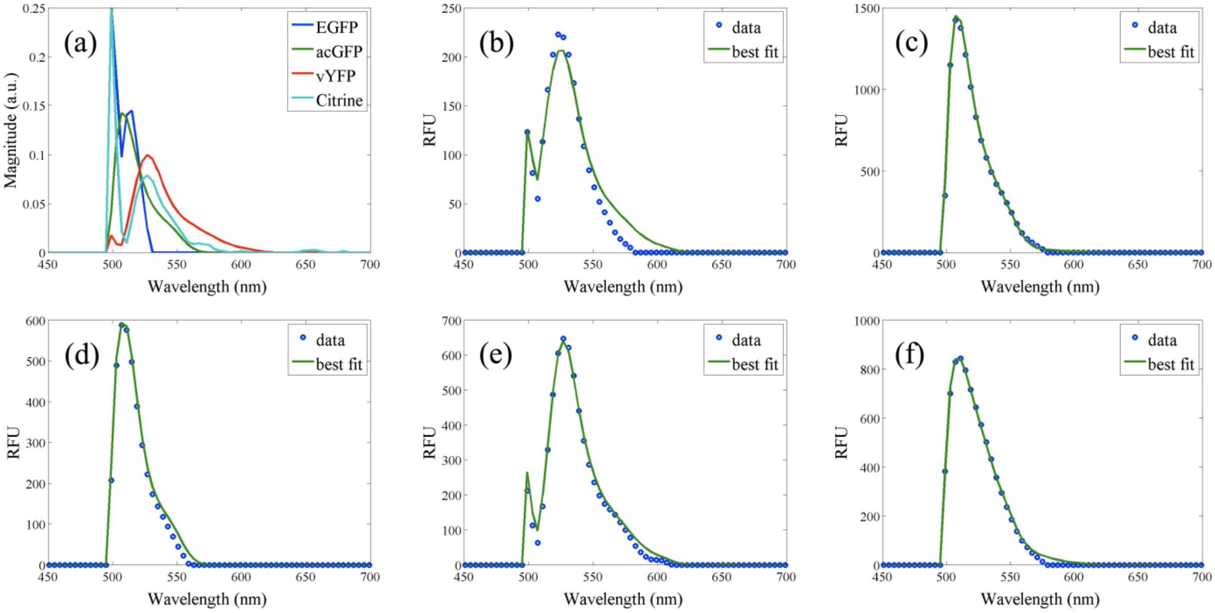

Figure 3.5: Experimentally measured emission spectra.
(a) Emission profiles of cell cultures expressing 1 of the 4 fluorescent proteins. (b-f) Collected fluorescence spectra for mixtures of cell cultures expressing (b) EGFP and vYFP, (c) acGFP and Citrine, (d) acGFP and EGFP, (e) Citrine and vYFP, and (f) acGFP, Citrine, EGFP, and vYFP.

There are several possible explanations for the observed discrepancies between the measured and extracted optical densities of the true positives listed in Table 3.3. First, the small sample size used (2 µL) could mean that the cell concentrations in the measured sample did not completely reflect the concentrations of the stock solution. Also, as can be seen in Figure 3.5a, there is a sharp peak visible in the fluorescence spectrum of all of the fluorophores except acGFP at ~500 nm. This peak appears to be part of the spectrum of the blue LED used for exciting all of the fluorescent proteins, and most likely contributes to some of the errors in the extracted contributions.

OD_{600}	Mix1		Mix 2		Mix 3		Mix 4		Mix 5	
	Meas.	Fit	Meas.	Fit	Meas.	Fit	Meas.	Fit	Meas.	Fit
acGFP	0	0.003	0.050	0.099	0.046	0.040	0	0.004	0.030	0.062
Citrine	0	0.006	0.070	0.039	0	0.002	0.081	0.014	0.027	0.017
EGFP	0.053	0.040	0	0	0.048	0.060	0	0	0.032	0.011
vYFP	0.048	0.029	0	0.006	0	0	0.082	0.100	0.027	0.037

Table 3.3: Actual and unmixed cell proportions.
Measured and extracted optical densities of mixtures of cells expressing a single fluorescent protein. Gray boxes indicate measured values of 0.

3.3 Discussion

3.3.1 Heterogeneous solutions to a synthetic biology design problem

We have proposed three distinct solutions to a specification, namely the detection of distinct combinations of chemical signals. The first proposed solution uses a set of three hybrid promoters, performing the logic at the transcriptional level. This transcriptional solution would quickly become intractable if the number of input chemical signals is increased. While we have shown a potential solution to the problem for three input signals, the number of parts needed to detect combinations from four possible inputs would require more parts than are currently available.

The second proposed solution still uses biology for the entirety of the logic, but does so in the protein domain. By moving the logic into the protein domain, the genetic circuitry is simplified. This allows the circuit to more easily be combined into a larger system, since the potential for cross-talk is decreased. Even though strategies to split fluorescent proteins still remain to be explored in a more systematic way, there are likely a limited number of fluorescent proteins that can be used in this context. So, the issue of the limited number of parts available faced in the transcriptional domain is also relevant in the protein domain. The assembly of the fluorescent proteins will increase the activation time of the system which can be an advantage or an inconvenience depending on the specific application of the environmental sensing device. In some cases, a fast response will be needed. In other cases, a longer maturation time can be used to time average the device response.

Finally, a third solution relies on the unmixing of fluorescence spectra for identification of molecules. In this case, biology is being used to generate the output signals, but the logic is being done outside of the biological domain. By implementing the logic outside of the biological system, the number of molecules possible to distinguish between is greatly increased, limited only by the number of transduction mechanisms and reporters.

Also, the use of simpler biological circuits in this implementation circumvents any possible difficulties that may arise from incompatibilities between biological parts in the other two design schemes.

As we proceed with this project by physically implementing and characterizing each of these approaches, we will meticulously observe the differences in the difficulty of implementation and the performance of each design. As a result of unbalanced component behavior, each approach is likely to be improved by tweaking such elements as promoter strength, translational efficiency, degradation, etc. Thus, the cost and benefit of iterative designs will be evaluated as well. By further considering the difficulty of the design process for each solution, we will be able to holistically compare the strengths and weaknesses of the different approaches. Some comparative measures are discussed below. At this stage, it is important for the design team to acknowledge that there are multiple solutions to design problems and that the solutions can be implemented in different design domains. Just like the design of electronic systems is often a heterogeneous combination of hardware and software solutions, the design of a synthetic biology device can include multiple domains for the wetware component of the design as well as the hardware and software used to integrate the information originating from the design wetware component into a larger system.

A possibility that has not been considered in this paper is the combination of solutions implemented in different design domains. Are there solutions that could combine hybrid promoters and fluorescence complementation? Or could spectral unmixing be combined with fluorescence complementation to achieve better performance? Even though this manuscript proposes three distinct solutions, the universe of possible solutions is large and difficult to explore manually.

3.3.2 Enabling co-design of synthetic biology application by design automation

In order to compare different solutions to a design problem, it is necessary to define various figures of merits that can be used to quantitatively compare different solutions. Sensitivity, dynamic range, response time, robustness, or noise can be used to characterize the design performance. The development cost could be estimated by a function of the number of previously characterized components that can be reused in a new design. For instance, a solution requiring the development of a new promoter is expected to be slower and more expensive to implement than a solution relying on well characterized genetic parts such as the fluorescence unmixing approach. The manufacturing or production cost may also be a factor. The development of Solution 3 is the simplest but it relies on a more refined optical components that would increase the size and manufacturing cost of the device. This option may not be practical if millions of sensing devices needed to be distributed over large geographic regions to detect facilities manufacturing chemical weapons. Each specific application will require optimizing these metrics using multi-objective optimization algorithms [47,48].

Formalizing the representation of the design space is necessary to automate its exploration while searching for optimal designs. Fortunately, the wetware component of the system can be represented by the sequence of the synthetic DNA molecule implementing the design. Our group recently proposed to use formal languages to represent the structure of synthetic DNA sequences [49]. More recently, this original syntactic model was augmented with a semantic model used to predict the behavior encoded in a DNA sequence (manuscript under revision). By implementing this formalism in a logic programming language like Prolog [50], we were able to systematically explore a design space by generating structurally correct DNA sequences, compiling them into SBML files describing their behavior, and simulating these files to identify solutions meeting a set of specification. Defining a distance in the design space, would make it possible to use optimization algorithms instead of a systematic exploration of all possible designs. In addition, it would be necessary to represent the non-DNA part of the designs by augmenting the language to represent detection systems and inputs.

The field of Synthetic Biology is growing by systematically adapting engineering practices to the design of biologically-inspired systems. The development of practical synthetic biology devices will require a system-level analysis and a co-design approach that have yet to be explored. In this paper, we have shown that the design space of a real-world device is large and may combine components developed in heterogeneous design domains. Finding optimal designs will require the use of design automation tools [51] like GenoCAD [52]. By adapting co-design methods used in more mature engineering fields [25,53,54], synthetic biology will fulfill its promise in the form of large, “interesting” circuits that were called for in the Big DNA contest .

3.4 Acknowledgments

This work was supported by the National Institutes of Health Grant 1 R01 GM078989, the MITRE Innovation Program, and a generous donation from Science Applications International Corporation (SAIC) to the Virginia Bioinformatics Institute. We are indebted to James J. Valdes for indicating the possible application of this environmental sensor to the enforcement of the Chemical Weapons Convention and other nonproliferation treaties.

3.5 References

1. Endy D (2005) Foundations for engineering biology. Nature 438: 449-453.
2. Seelig G, Soloveichik D, Zhang DY, Winfree E (2006) Enzyme-free nucleic acid logic circuits. Science 314: 1585-1588.
3. Win MN, Smolke CD (2008) Higher-order cellular information processing with synthetic RNA devices. Science 322: 456-460.
4. Buchler NE, Gerland U, Hwa T (2003) On schemes of combinatorial transcription logic. Proc Natl Acad Sci U S A 100: 5136-5141.

5. Nguyen NPD, Kuwahara H, Myers CJ, Keener JP (2007) The design of a genetic muller C-element. ASYNC 2007: 13th IEEE International Symposium on Asynchronous Circuits and Systems: 95-104.
6. Gardner TS, Cantor CR, Collins JJ (2000) Construction of a genetic toggle switch in Escherichia coli. Nature 403: 339-342.
7. Hasty J, Pradines J, Dolnik M, Collins JJ (2000) Noise-based switches and amplifiers for gene expression. Proceedings of the National Academy of Sciences of the United States of America 97: 2075-2080.
8. Atkinson MR, Savageau MA, Myers JT, Ninfa AJ (2003) Development of genetic circuitry exhibiting toggle switch or oscillatory behavior in Escherichia coli. Cell 113: 597-607.
9. Tigges M, Marquez-Lago TT, Stelling J, Fussenegger M (2009) A tunable synthetic mammalian oscillator. Nature 457: 309-312.
10. Elowitz MB, Leibler S (2000) A synthetic oscillatory network of transcriptional regulators. Nature 403: 335-338.
11. Fung E, Wong WW, Suen JK, Bulter T, Lee SG, et al. (2005) A synthetic gene-metabolic oscillator. Nature 435: 118-122.
12. Chilov D, Fussenegger M (2004) Toward construction of a self-sustained clock-like expression system based on the mammalian circadian clock. Biotechnology And Bioengineering 87: 234-242.
13. Ajo-Franklin CM, Drubin DA, Eskin JA, Gee EP, Landgraf D, et al. (2007) Rational design of memory in eukaryotic cells. Genes Dev 21: 2271-2276.
14. Friedland AE, Lu TK, Wang X, Shi D, Church G, et al. (2009) Synthetic Gene Networks That Count. Science 324: 1199-1202.
15. Goler JA, Bramlett BW, Peccoud J (2008) Genetic design: rising above the sequence. Trends Biotechnol 26: 538-544.
16. Guet CC, Elowitz MB, Hsing W, Leibler S (2002) Combinatorial synthesis of genetic networks. Science 296: 1466-1470.
17. Bayer TS, Smolke CD (2005) Programmable ligand-controlled riboregulators of eukaryotic gene expression. Nature Biotechnology 23: 337-343.
18. Win MN, Smolke CD (2007) A modular and extensible RNA-based gene-regulatory platform for engineering cellular function. Proc Natl Acad Sci U S A 104: 14283-14288.
19. Isaacs FJ, Dwyer DJ, Collins JJ (2006) RNA synthetic biology. NatBiotechnol 24: 545-554.
20. Isaacs FJ, Collins JJ (2005) Plug-and-play with RNA. Nature Biotechnology 23: 306-307.
21. Kim J, White KS, Winfree E (2006) Construction of an in vitro bistable circuit from synthetic transcriptional switches. Molecular Systems Biology: -.
22. Okamoto A, Tanaka K, Saito I (2004) DNA logic gates. Journal of the American Chemical Society 126: 9458-9463.
23. Muramatsu S, Kinbara K, Taguchi H, Ishii N, Aida T (2006) Semibiological molecular machine with an implemented "AND" logic gate for regulation of protein folding. J Am Chem Soc 128: 3764-3769.
24. DeMicheli G, Gupta RK (1997) Hardware/software co-design. Proceedings of the Ieee 85: 349-365.

25. Wolf WH (1994) Hardware-software co-design of embedded systems. Proceedings of the IEEE 82: 967-989.
26. Kobayashi H, Kaern M, Araki M, Chung K, Gardner TS, et al. (2004) Programmable cells: interfacing natural and engineered gene networks. Proceedings of the National Academy of Sciences of the United States of America 101: 8414-8419.
27. Institute of Medicine (U.S.). Committee to Survey the Health Effects of Mustard Gas and Lewisite., Pechura CM, Rall DP (1993) Veterans at Risk : the health effects of mustard gas and Lewisite. Washington, D.C.: National Academy Press. xviii, 427 p. p.
28. Witeska M, Jezierska B. The effects of environmental factors on metal toxicity to fish; 2002 Oct 14-16; Brno, Czech Republic. Parlar Scientific Publications (P S P). pp. 824-829.
29. Richard S, Moslemi S, Sipahutar H, Benachour N, Seralini GE (2005) Differential effects of glyphosate and roundup on human placental cells and aromatase. Environ Health Perspect 113: 716-720.
30. Lutz R, Bujard H (1997) Independent and tight regulation of transcriptional units in Escherichia coli via the LacR/O, the TetR/O and AraC/I1-I2 regulatory elements. Nucleic Acids Res 25: 1203-1210.
31. Cox RS, 3rd, Surette MG, Elowitz MB (2007) Programming gene expression with combinatorial promoters. Mol Syst Biol 3: 145.
32. Egland KA, Greenberg EP (1999) Quorum sensing in Vibrio fischeri: elements of the luxl promoter. Molecular Microbiology 31: 1197-1204.
33. Basu S, Gerchman Y, Collins CH, Arnold FH, Weiss R (2005) A synthetic multicellular system for programmed pattern formation. Nature 434: 1130-1134.
34. Karig D, Weiss R (2005) Signal-amplifying genetic enables in vivo observation circuit of weak promoter activation in the RhI quorum sensing system. Biotechnology And Bioengineering 89: 709-718.
35. Basu S, Mehreja R, Thiberge S, Chen MT, Weiss R (2004) Spatiotemporal control of gene expression with pulse-generating networks. Proceedings of the National Academy of Sciences of the United States of America 101: 6355-6360.
36. Ghosh I, Hamilton AD, Regan L (2000) Antiparallel leucine zipper-directed protein reassembly: Application to the green fluorescent protein. Journal of the American Chemical Society 122: 5658-5659.
37. Kodama Y, Wada M (2009) Simultaneous visualization of two protein complexes in a single plant cell using multicolor fluorescence complementation analysis. Plant Mol Biol 70: 211-217.
38. Hu CD, Kerppola TK (2003) Simultaneous visualization of multiple protein interactions in living cells using multicolor fluorescence complementation analysis. Nature Biotechnology 21: 539-545.
39. Shaner NC, Steinbach PA, Tsien RY (2005) A guide to choosing fluorescent proteins. Nature Methods 2: 905-909.
40. Zimmermann T (2005) Spectral imaging and linear unmixing in light microscopy. Microscopy Techniques. pp. 245-265.
41. Dickinson ME, Bearman G, Tille S, Lansford R, Fraser SE (2001) Multi-spectral imaging and linear unmixing add a whole new dimension to laser scanning fluorescence microscopy. Biotechniques 31: 1272-+.

42. Lansford R, Bearman G, Fraser SE (2001) Resolution of multiple green fluorescent protein color variants and dyes using two-photon microscopy and imaging spectroscopy. Journal of Biomedical Optics 6: 311-318.
43. Huh WK, Falvo JV, Gerke LC, Carroll AS, Howson RW, et al. (2003) Global analysis of protein localization in budding yeast. Nature 425: 686-691.
44. Gurskaya NG, Fradkov AF, Pounkova NI, Staroverov DB, Bulina ME, et al. (2003) Colourless green fluorescent protein homologue from the non-fluorescent hydromedusa Aequorea coerulescens and its fluorescent mutants. Biochemical Journal 373: 403-408.
45. Raser JM, O'Shea EK (2005) Noise in gene expression: origins, consequences, and control. Science 309: 2010-2013.
46. Heikal AA, Hess ST, Baird GS, Tsien RY, Webb WW (2000) Molecular spectroscopy and dynamics of intrinsically fluorescent proteins: Coral red (dsRed) and yellow (Citrine). Proceedings of the National Academy of Sciences of the United States of America 97: 11996-12001.
47. Goh C-K (2009) Evolutionary multi-objective optimization in uncertain environments : issues and algorithms. New York: Springer.
48. Deb K (2001) Multi-objective optimization using evolutionary algorithms. Chichester ; New York: John Wiley & Sons. xix, 497 p. p.
49. Cai Y, Hartnett B, Gustafsson C, Peccoud J (2007) A syntactic model to design and verify synthetic genetic constructs derived from standard biological parts. Bioinformatics 23: 2760-2767.
50. Colmerauer A (1990) An Introduction to Prolog-Iii. Communications of the Acm 33: 69-90.
51. Sangiovanni-Vincentelli A (2003) The tides of EDA. Ieee Design & Test of Computers 20: 59-75.
52. Czar MJ, Cai Y, Peccoud J (2009) Writing DNA with GenoCAD. Nucleic Acids Res (in press).
53. Benini L, De Micheli G (2000) System-level power optimization: Techniques and tools. Acm Transactions on Design Automation of Electronic Systems 5: 115-192.
54. Bolsens I, DeMan HJ, Lin B, VanRompaey K, Vercauteren S, et al. (1997) Hardware/software co-design of digital telecommunication systems. Proceedings of the Ieee 85: 391-418.

Chapter 4: Modeling structure-function relationships in synthetic DNA sequences using attribute grammars

Authors

Yizhi Cai[1], Matthew W. Lux[1], Laura Adam[1], Jean Peccoud[1]

[1]Virginia Bioinformatics Institute, Virginia Polytechnic Institute and State University, Washington St, MC 0477, Blacksburg VA 24061

Abstract

Background: Recognizing that certain biological functions can be associated with specific DNA sequences had led various fields of biology to adopt the notion of the genetic part. This concept provides a finer level of granularity than the traditional notion of the gene. However, a method of formally relating how a set of parts relates to a function has not yet emerged. Synthetic biology both demands such a formalism and provides an ideal setting for testing hypotheses about relationships between DNA sequences and phenotypes beyond the gene-centric methods used in genetics.

Method: Attribute grammars are used in computer science to translate the text of a program source code into the computational operations it represents. By associating attributes with parts, modifying the value of these attributes using rules that describe the structure of DNA sequences, and using a multi-pass compilation process, it is possible to translate DNA sequences into molecular interaction network models. These capabilities are illustrated by simple example grammars expressing how gene expression rates are dependent upon single or multiple parts. The translation process is validated by systematically generating, translating, and simulating the phenotype of all the sequences in the design space generated by a small library of genetic parts.

Conclusions: Attribute grammars represent a flexible framework connecting parts with models of biological function. They will be instrumental for building mathematical models of libraries of genetic constructs synthesized to characterize the function of genetic parts. This formalism is also expected to provide a solid foundation for the development of computer assisted design applications for synthetic biology.

Publication status

This article is reproduced here under an open access Creative Commons license, with permission of the primary author, Yizhi Cai, from the original publication at:

Cai Y, Lux MW, Adam L, Peccoud J (2009) Modeling Structure-Function Relationships in Synthetic DNA Sequences using Attribute Grammars. PLoS Comput Biol 5(10): e1000529. doi:10.1371/journal.pcbi.1000529

4.1 Author's summary

Deciphering the genetic code has been one of the major milestones in our understanding of how genetic information is stored in DNA sequences. However, only part of the genetic information is captured by the simple rules describing the correspondence between gene and proteins. The molecular mechanisms of gene expression are now understood well enough to recognize that DNA sequences are rich in functional blocks that do not code for proteins. It has proved difficult to express the function of these genetic parts in a computer readable format that could be used to predict the emerging behavior of DNA sequences combining multiple interacting parts. We are showing that methods used by computer scientists to develop programming languages can be applied to DNA sequences. They provide a framework to 1) express the biological functions of genetic parts, 2) how these functions depend on the context in which the parts are placed, and 3) translate DNA sequences composed of multiple parts into a model predicting how the DNA sequence will behave in vivo. Our approach provides a formal representation of how the biological function of genetic parts can be used to assist in the engineering of synthetic DNA sequences by automatically generating models of the design for analysis.

4.2 Introduction

“How much can a bear bear?” This riddle uses two homonyms of the word “bear”. The first instance of the word is a noun referring to an animal, and the second is a verb meaning “endure”. Although the word “bear” has over 50 different meanings in English, its meaning in any given sentence is rarely ambiguous. In a simple case like this riddle, the meaning of each word can be deciphered by looking at other words in the same sentence. In other cases, it is necessary to take into account a broader context to properly interpret the word. For instance, it may be necessary to read several sentences to decide if “bear claw” refers to a body part or a pastry. A reader will progressively derive the meaning of a text by recognizing structures consistent with the language grammar. It is often difficult to understand the meaning of a text by relying exclusively on a dictionary.

It is interesting to compare this bottom-up emergence of meaning with the top-down approach that made genetics so successful. The discipline was built upon a quest to define hereditary units that could be associated with observable traits well before the physical support of heredity was discovered (1,2). The one-to-one relationship between genes and traits was later refined by Beadle and Tatum’s hypothesis that the gene action was mediated by enzymes (3,4). Cracking the genetic code has been one of the major milestones in understanding the information content of nucleic acids sequences. By demonstrating the colinearity of DNA, RNA, and protein sequences, the genetic code was instrumental in the identification of specific DNA sequences as genes. The influence of this legacy on contemporary biology cannot be underestimated. Models used in quantitative genetics predict phenotypes from unstructured lists of alleles at different loci (5,6). Similarly, genome annotations remain very gene-centric. Most bioinformatic databases have been designed to collect information relative to coding regions or candidate genes. Few, if any, annotations of non-coding regions or higher order structures are being systematically recorded even for model organisms like yeast (7,8).

Yet, despite its success, the notion of the gene is being challenged (1,9). The elucidation of the molecular mechanisms controlling gene expression has revealed a web of molecular interactions that have been modeled mathematically to show that important phenotypic traits are the emerging properties of complex system (10-15). The development of this more integrated understanding of the cell physiology leads to a progressive adoption of the more neutral notion of genetic part as a replacement for the notion of genes associated with specific traits. Making sense of the list of parts generated in genomics, proteomics, and metabolomics has been a major challenge for the systems biology community (16-21).

It is becoming apparent that the genetic code captures only a small fraction of the information content of DNA molecules (22,23). Yet, if there is a general agreement that the cell dynamics is somehow coded in genetic sequences, no formal relationship between DNA sequences and dynamical models of gene expression has been proposed so far. In particular, the formalization of the biological functions of genetic parts has remained elusive. As a result, building models of gene networks encoded in DNA sequences remains a labor-intensive process. This limitation has hampered the development of large families of models needed to analyze phenotypic data generated by libraries of related genetic constructs (24-28).

Synthetic biology is likely to be instrumental in refining our understanding of the design of natural biological systems (29). Just like the genetic code was partly elucidated through the *de novo* chemical synthesis of DNA molecules (30,31), the redesign of genomic sequences will shed a new light on the relations between structure and function in genetic sequences (32-34). By considering biological parts as the building blocks of artificial DNA sequences (35), designing new parts that do not exist in nature (26-28), and making parts physically available to the community (36), synthetic biology calls for a systematic functional characterization of genetic parts (37). These efforts are still limited by the difficulty in expressing how the function of biological parts may be influenced by the structure of the DNA sequence in which they are used. It has been shown that a partial redesign of the genomic sequences of two viruses had a significant effect on the virus fitness even though the redesigns preserved the protein sequences (33,38). Just as the context the expression “bear claw” helps understand its meaning, it is necessary to consider the entire structure of the DNA molecule coding for particular genes to appreciate how those genes contribute to the phenotype.

One possible approach to this problem is to extend the linguistic metaphor used to formulate the central dogma. The notions of genetic code, transcription, and translation are derived from a linguistic representation of biological sequences. Several authors have modeled the structure of various types of biological sequences using syntactic models (39-46). However, these structural models have not yet been complemented by semantic models expressing the sequence function. An interesting attempt to use grammars to model the dynamics of gene expression did not rely on a description of the DNA sequence structure. Instead, this grammar described how various inducible or repressible promoters can transition between different states under the control of environmental parameters (47). The simple semantic model stored in a knowledge base established a correspondence between the strings generated by the syntax and the physiological state of the cell. The Sequence Ontology (48) and the Gene Regulation Ontology (49) represent other attempts to associate semantic values with biological sequences. Their controlled vocabularies can be used by software applications to manage knowledge. However, the semantic derived from these ontologies is a semantic of the sequence annotation, not of the sequences themselves.

We recently described a fairly simple syntactic model of synthetic DNA sequences (50) capable of generating a large number of previously published synthetic genetic constructs (24,25,51). We have now enhanced this initial syntactic model with a semantic model capable of expressing the dynamics of the molecular mechanisms coded by the DNA sequences. Specialized terms like syntax, semantics, and others are defined in Table 4.1. Our approach uses attribute grammars (52), a theoretical framework developed in the 60s to establish a formal correspondence between the text of a computer program and the series of microprocessor operations it codes for (53,54). Even though other types of semantic models have been developed since then (55,56), attribute grammars still represent a good compromise between simplicity and expressivity, an important

Attribute grammar	An attribute grammar is a context free grammar augmented with attributes, semantic rules, and conditions. Attribute grammars were developed as a means of formalizing the semantics of a context free grammar.
Context free grammar	A context free grammar is a quadruple (V, Σ, P, S) where V is a finite set of non-terminal symbols, Σ (the alphabet) is a finite set of terminal symbols, P is a finite set of rules, and S is a distinguished element of V called the start symbol. A rule P is of the following form $A \rightarrow \omega$ where A is a single non-terminal symbol and ω is a string of terminals and/or non-terminals (possibly empty). The term "context-free" expresses the fact that non-terminals are rewritten without regard to the context in which they occur.
CUSP Bifurcation	A codimension 2 bifurcation formed by the tangential meeting of two loci of saddle-node bifurcations. In other words, a cusp bifurcation traces the path of the points bounding a bistable region as they change with changes in two parameters. Bistability is implied within the cusp bounds.
Direct left recursion	A direct left recursion in context free grammar refers to rules of the form $A \rightarrow A\omega$. Parsing left recursion can possibly lead the parser down an infinite branch of the search tree in the corresponding logic program.
PoPS	The measurement of polymerase per second transcribing past a defined point of DNA.
SBML	The Systems Biology Markup Language (SBML) is a machine-readable language, based on XML, for representing models of biochemical reaction networks.
Semantics	Semantics reveals the meaning of syntactically valid strings in a language. For natural languages, this means correlating sentences and phrases with the objects, thoughts, and feelings of our experiences. For programming languages, semantics describes the behavior that a computer follows when executing a program in the language.
Syntax	Syntax refers to the ways symbols may be combined to create well-formed sentences (or programs) in a language. Syntax defines the formal relations between the constituents of a language, thereby providing a structural description of the various expressions that make up legal strings in the language. Syntax deals solely with the form and structure of symbols in a language without any consideration given to their meaning.

Table 4.1: Glossary of specialized terms used throughout this article.

characteristic if the framework is to be used by non-computer scientists. Attribute grammars make it possible to use well characterized compilation algorithms to translate a DNA sequence into a mathematical model of the molecular interactions it codes for. As the static source code of a program directs the dynamic series of operations carried out by the microprocessor based on user inputs, the compilation process translates the static information of cells coded by DNA sequences into a dynamical model of the development of a phenotype in response to environmental influences (57).

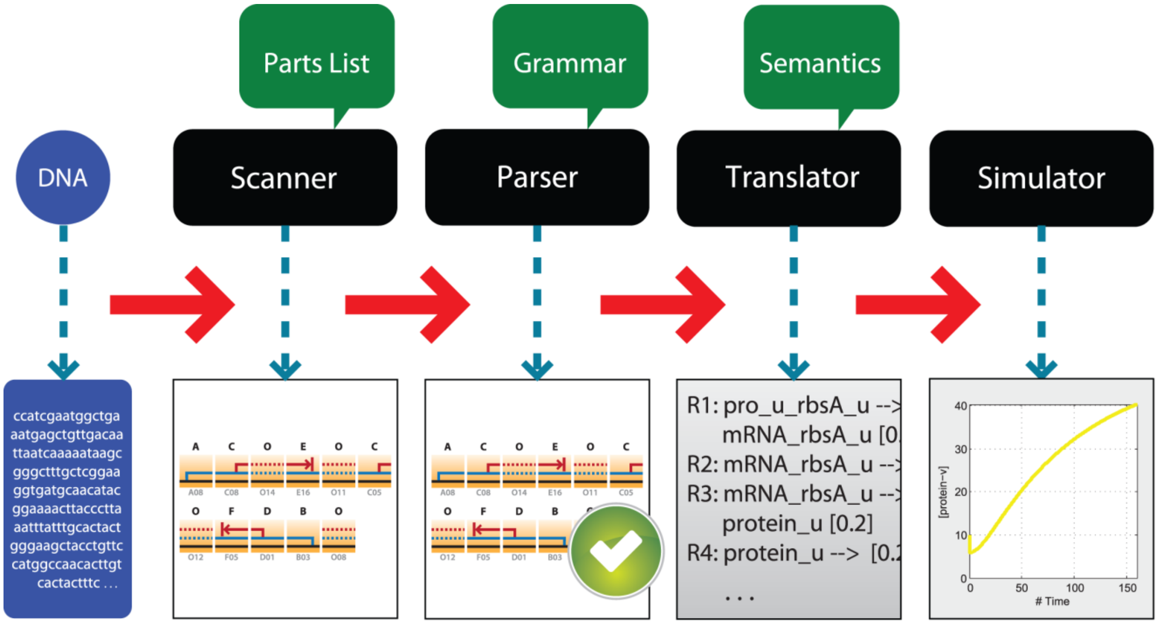

Figure 4.1: Workflow of generating the gene network model encoded in a DNA sequence. The input for this process is a DNA sequence that is first broken down into parts by the scanner. The combination of the parts is validated by the parser according to a syntactic model. After validation by the parser, the sequence is translated by applying semantic actions attached to the rules to transform the series of parts into a set of chemical equations. The resulting equations can then be solved using existing simulation engines. Each step takes the output of the previous step as input, so the workflow can start from any step if the appropriate input is provided.

4.3 Results

4.3.1 Compilation of a DNA sequence

The translation of a gene network model from a genetic sequence is very similar to the compilation of the source code of a computer program into an object code that can be executed by a microprocessor (Figure 4.1). The first step consists in breaking down the DNA sequence into a series of genetic parts by a program called lexer or scanner. Since the sequence of a part may be contained in the sequence of another part, the lexer is capable of backtracking to generate all the possible interpretations of the input DNA sequences as a series of parts. All possible combinations of parts generated by the lexer are sent to a second program called parser to analyze if they are structurally consistent with the language syntax. The structure of a valid series of parts is represented by a parse tree (50) (Figure 4.2). The semantic evaluation takes advantage of the parse tree to translate the DNA sequence into a different representation such as a chemical reaction network. The translation process requires attributes and semantic actions. Attributes are properties of individual genetic parts or combinations of parts. Semantic actions are associated with the grammar production rules. They specify how attributes are computed. Specifically, the translation process relies on the semantic actions associated with parse tree nodes to synthesize the attributes of the construct from the attributes of its parts. In our implementation, the product of the translation is a mass action model of the network of molecular interactions encoded in the DNA sequence. By using the standardized format of Systems Biology Markup Language (SBML), the model can be analyzed using existing simulation engines (58-60).

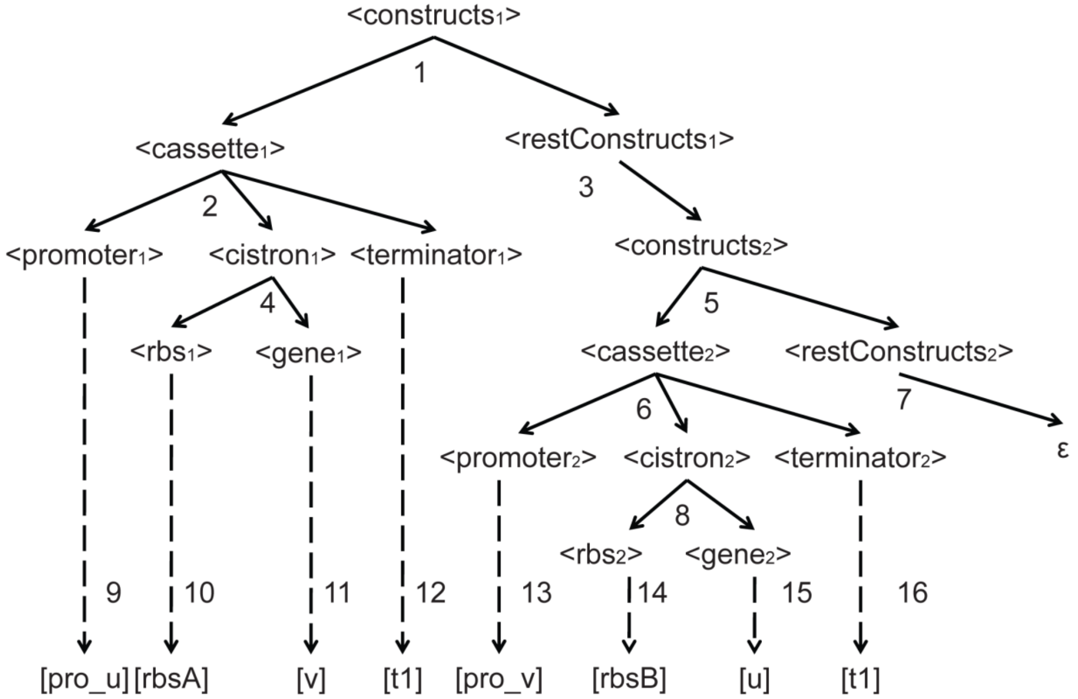

Figure 4.2: Parse tree showing the derivation process of a two-cassette genetic construct. In the derivation tree, terms in <> corresponds to the non-terminals in the grammar, while terms in [ ] are terminals, and the dashed lines indicate the transformation to terminals. The subscripts are used to distinguish different instances of the same category.

We have developed a simple grammar compact enough to be presented extensively, yet sufficiently complex to represent basic epistatic interactions. The grammar generates constructs composed of one or more gene expression cassettes. The gene expression cassettes are themselves composed of a promoter, cistron, and transcription terminator. Finally, a cistron is composed of a Ribosome Binding Site (RBS) and a coding sequence (gene). The syntax is composed of 12 production rules (P1 to P12) displayed in bold characters in Table 4.2. In this table, each entry is composed of a rewriting rule (bold),

P1. constructs → cassette, restConstructs {
constructs.promoter_list = cassette.promoter_list + restConstructs.promoter_list
constructs.equation_list = cassette.equation_list + restConstructs.equation_list)
cassette.protein_list = constructs.protein_list
restConstructs.protein_list = constructs.protein_list}
P2. restConstructs → constructs{
restConstructs.promoter_list = constructs.promoter_list
restConstructs.equation_list = constructs.equation_list
constructs.protein_list = restConstructs.protein_list}
P3. restConstruct → ε{
restConstructs.promoter_list = [ ]
restConstructs.equation_list = [ ]
restConstructs.protein_list = [ ]}
P4. cassette → promoter, cistron, terminator{
cassette.promoter_list= [promoter.name, cistron.transcript]
cassette.equation_list = cistron.equation_list + *promoter_protein_interaction*(cassette.promoter_list, cassette.protein_list) + *transcription*(promoter, cistron.transcript)}
P5. cistron → rbs, gene {
cistron.transcript = rbs.name + gene.name
cistron.equation_list= *translation*(rbs, gene)}
	P6. promoter → pro_u{
	promoter.name = [pro_u]
	promoter.transcription_rate = k_1
	promoter.leakiness_rate = k_{11}
	promoter.repressor_list = [(u,2, k_9, k_{9r})] }
	P7. promoter → pro_v{
	promoter.name = [pro_v]
	promoter.transcription_rate = [k_2]
	promoter.leakiness_rate = [k_{12}]
	promoter.repressor_list = [(v, 4, k_{10}, k_{10r})]}
	P8. rbs → rbsA{
	rbs.name = [rbsA]
	rbs.translation_rate = [k_3]}
	P9. rbs → rbsB{
	rbs.name = [rbsB]
	rbs.translation_rate = [k_4]}
	P10. gene → u {
	gene.name = [u]
	gene.mRNA_degradation_rate= [k_5]
	gene.protein_degradation_rate = [k_7]}
	P11. gene → v{
	gene.name = [v]
	gene.mRNA_degradation_rate = [k_6]
	gene.protein_degradation_rate = [k_8]}
	P12. terminator → t1{
	terminator.name=[t1]}

Table 4.2: An example of attribute grammar.

and semantic actions (curly brackets). The symbol ε refers to an empty string, [] means an empty list and the '+' sign indicates the concatenation operation on two lists. This syntax is comparable to the one described previously (50) except that we introduced the extra non-terminal `restConstructs` to allow the generation of constructs with multiple cassettes without introducing parsing problems due to direct left recursions (61).

The attributes of a part include the kinetic rates related to this part and the interaction information. For example, the attributes of a promoter include a transcription rate along with a list of proteins repressing it and the kinetic parameters of the protein-DNA interactions. For non-terminal variables corresponding to combinations of parts such as cistrons, the attributes include a list of proteins, a list of promoters, and a list of chemical equations. The equation list is used to store the model of the system behavior, while the lists of promoters and proteins are recorded for computing the molecular interactions resulting from the DNA sequence. The complete set of attributes used in this simple grammar is listed in Table 4.3.

Non-terminals	Inherited Attribute	Synthesized Attributes
constructs	protein_list	promoter_list, equation_list
cassette	protein_list	promoter_list, equation_list
restConstructs	protein_list	promoter_list, equation_list
cistron	protein_list	transcript, equation_list
promoter	-	name , transcription_rate, leakiness_rate, repressor_list
RBS	-	name, translation_rate
gene	-	name, mRNA_degradation_rate, protein_degradation_rate
terminator	-	name

Table 4.3: Attributes associated with non-terminals.

If many attributes can be computed locally by only considering a small fragment of the DNA sequence, other attributes are global properties of the system. For instance, the computation of protein-DNA interactions requires access to a global list of proteins expressed by the constructs. However, this list is not available until all of the different cassettes have been parsed. The problem is overcome by using a multiple-pass compilation method. In the first pass, the compiler does not do any structural validation but builds the list of proteins in the system and passes the list as an inherited attribute to the second pass. In the second pass, the promoter-protein interactions can be calculated locally at the level of each cassette. Rules P1 to P5 define the structure of a design, while rules P6 to P12 cover the selection of a specific part for each category. In the semantic action, the relation between an attribute and its variable is indicated by a dot and constants are enclosed by brackets. For instance, `gene.mRNA_degration_rate = [`k_6`]` indicates that the value of the attribute `mRNA_degration_rate` of a gene is a constant k_6. The attribute `repressor_list` used in P6 and P7 includes the name of the repressor, the stoichiometry, and the kinetic constants of the forward and reverse reactions of the protein-DNA interaction. Table A.1 details the parsing steps and computational dependence of each step. Finally, the equation writing operations are handled by functions typed in italics in Table 4.2 and defined in Table 4.4.

Generators	Parameters	Equations

promoter_protein_interaction	promoter_list,
	protein_list
		if promoter.repressor is in protein_list
		{
		promoter_transcript + protein ↔ promoter_transcript_x (binding_rate,release_rate)
		promoter_trancript_x → promoter_transcript_x + mRNA_transcript (promoter.leakiness_rate)
		}
		endif

transcription	promoter, transcript	promoter_transcript → promoter_transcript + mRNA_transcript (promoter.transcription_rate) mRNA_transcript → Ø (transcript.mRNA_degradation_rate)
translation	rbs, gene	mRNA_rbs_gene → mRNA_rbs_gene + protein_gene (rbs.translation_rate) protein_rbs_gene → Ø (gene.protein_degradation_rate)

Table 4.4: Equation generators

The translation of the DNA sequence into a mathematical model is available as the `equation_list` attribute of constructs. The model outputs are generated by equations generators, which are purposely decoupled from the semantic actions. The decoupling enables the flexibility of using different equation formats to describe a biological process. The translation of the construct composed of the parts `pro_u rbsA gene_v t1 pro_v rbsB gene_u t1` generates the equations displayed in the `[Reactions]` section of Table 4.5. Each line is composed of a reaction index (R1 to R12), the chemical equation itself, and one or two reaction parameters depending on the reaction reversibility. The initial values have been computed by assigning 1 to variables representing DNA sequences and prompting the user to set the initial condition of proteins.

[Reactions]

R1: pro_u_rbsA_v -->pro_u_rbsA_v + mRNA_rbsA_v [k_1]

R2: mRNA_rbsA_v --> [k_6]

R3: mRNA_rbsA_v -->mRNA_rbsA_v + protein_v [k_3]

R4: protein_v --> [k_8]

R5: pro_v_rbsB_u -->pro_v_rbsB_u + mRNA_rbsB_u [k_2]

R6: mRNA_rbsB_u --> [k_5]

R7: mRNA_rbsB_u -->mRNA_rbsB_u + protein_u [k_4]

R8: protein_u --> [k_7]

R9: pro_v_rbsB_u +4protein_v <-->pro_v_rbsB_u_x [k_{10}, k_{10r}]

R10: pro_v_rbsB_u_x -->pro_v_rbsB_u_x + mRNA_rbsB_u [k_{12}]

R11: pro_u_rbsA_v +2protein_u <-->pro_u_rbsA_v_x [k_9, k_{9r}]

R12: pro_u_rbsA_v_x -->pro_u_rbsA_v_x + mRNA_rbsA_v [k_{11}]

[InitialValues]

pro_u_rbsA_v= 1

pro_v_rbsB_u = 1

protein_v = user input

protein_u = user input

Table 4.5: Chemical equations translated from a DNA sequence.

4.3.2 Expressing context-dependencies of parts function

The semantic model presented in the previous section is completely modular since the parameters of the model describing the construct behavior are attributes of individual parts, not of higher order structures. For instance, in the previous model (Table 4.2 to Table 4.4), translational efficiency is primarily determined by the RBS sequence (62,63). This association between RBS and translation rate was successfully used to design one of the first artificial gene networks (24) and is still used by many synthetic biology software applications (64-67). Yet, it is also well known that translation initiation can be attenuated by stable mRNA secondary structures (68-70). This leads to a situation where a translational rate can no longer be considered the attribute of an individual part but needs to be considered as the attribute of a specific combination of parts. This type of context-dependency can naturally be expressed using attribute grammars since the translation reaction is computed at the cistron level, not at the level of individual parts. Rule P5 of Table 4.2 can be modified by introducing a new function to retrieve the translation rate for specific combination of gene and RBS.

```
P5. cistron --> rbs, gene
{
cistron.translation_rate = get_translation_rate(rbs, gene)
cistron.transcript = rbs.name + gene.name
cistron.equation_list = translation(rbs, gene,
cistron.translation_rate)
}
```

The get_translation_rate function checks for specific cases of interactions between an RBS and coding sequence first. If none is found, then the default RBS translation rate is used.

```
If exists translation_rate(rbs, gene)
      translation_rate = translation_rate(rbs, gene)
else
      translation_rate = translation_rate(rbs)
endif
```

This approach is illustrated in Table 4.6 using previously published data demonstrating the interference between the RBS and coding sequence (68). Specifically, this report provides the relation expression observed in 23 different constructs generated by combining different variants of the RBS and MS2 coat protein gene. This data set has been reorganized in Table 4.6 by sorting the constructs according to the RBS and gene variants they used. Three of the constructs using the WT RBS sequence resulted in a maximum level of expression while the expression of the gene variants ORF4, ORF5, and ORF6 were expressed at a much lower level due to the greater stability of the mRNA secondary structure. A similar pattern is observed for other RBS variants (RBS1, RBS2, RBS3, RBS7). For all of these RBS variants, it is possible to define the `translation_rate` function by associating the default translation rate with the maximum expression rate. Specific translation rates associated with particular pairs of RBS and gene variants are recorded separately.

4.3.3 Exploration of genetic design space

The semantic model in Table 4.2 to Table 4.4 is a compact proof of concept example but it does not capture a number of features commonly found in actual genetic constructs. In order to demonstrate that our approach is capable of modeling more realistic DNA sequences, we have extended this semantic model (Supplementary Materials) to translate the DNA sequences of previously published DNA plasmids that include polycistronic cassettes in different orientations (24). This plasmid library was generated by 29 different genetic parts (three promoters: $p_L$tetO-1, p_Ls1con, ptrc-2; eight RBS: rbsA to rbsH; and three genes: *tetR*, *cIts*, and *lacI*, and one terminator, all in both orientations). The syntax generates 72 different single gene expression constructs in each orientation. By combining two genes repressing each other in a construct, it is possible to make a bistable artificial gene network that can be used as a genetic switch.

Mutant	RBS	ORF	Expression	Translation rate function
1	RBS WT	ORF WT	100	translation_rate(RBS WT)
6	RBS WT	ORF2	100	translation_rate(RBS WT)
7	RBS WT	ORF3	100	translation_rate(RBS WT)
17	RBS WT	ORF4	3	translation_rate(RBS WT, ORF4)
20	RBS WT	ORF5	6	translation_rate(RBS WT, ORF5)
23	RBS WT	ORF6	0.3	translation_rate(RBS WT, ORF6)
4	RBS1	ORF WT	100	translation_rate(RBS1)
2	RBS1	ORF1	100	translation_rate(RBS1)
3	RBS1	ORF2	100	translation_rate(RBS1)
5	RBS1	ORF3	4	translation_rate(RBS1, ORF3)
14	RBS1	ORF4	<0.003	translation_rate(RBS1, ORF4)
9	RBS2	ORF WT	100	translation_rate(RBS2)
8	RBS2	ORF1	100	translation_rate(RBS2)
10	RBS2	ORF3	100	translation_rate(RBS2)
12	RBS3	ORF WT	100	translation_rate(RBS3)
11	RBS3	ORF1	20	translation_rate(RBS3, ORF1)
13	RBS3	ORF3	100	translation_rate(RBS3)
15	RBS4	ORF4	0.1	translation_rate(RBS4)
16	RBS5	ORF4	0.05	translation_rate(RBS5)
22	RBS6	ORF WT	0.2	translation_rate(RBS6, ORF WT)
18	RBS6	ORF4	80	translation_rate(RBS6)
21	RBS7	ORF WT	100	translation_rate(RBS7)
19	RBS7	ORF4	100	translation_rate(RBS7)

Table 4.6: Context-dependency of experimentally determined translation rates.

To demonstrate the potential use of a semantic model to search for a desirable behavior in a large genetic design space, we have generated the DNA sequences of all 41,472 possible sequences ($72^2 \times 8$ RBS for the reporter gene) having the same structure as previously described switches. All sequences were translated into separate model files and a script was developed to perform a bistability analysis of each model. Parameters of the semantic model were obtained by qualitatively matching the experimental results of the six previously published switches (24) and are summarized in Table A.2. Most of the automatically generated sequences led to inherently non-bistable networks because the necessary repressor/promoter pairs did not match. Since this specific example is particularly well understood, we could have generated a limited number of targeted constructs. Yet, we chose to generate all possible sequences to demonstrate the generality of our approach. In particular, it was important to evaluate the computational cost of generating and translating DNA sequences to ensure that it would not prevent a systematic exploration of more complex design spaces. It takes only minutes to generate 41,472 sequences and translate them into SBML files. Hence, the computational cost of this step is negligible compared to the time required by the simulation of the SBML files.

Bistability was tested numerically by integrating the differential equations until they converged to a steady state starting from two different initial conditions. The two initial conditions started with one protein level very high and the other very low and vice versa. We characterized the bistability by computing the ratio of reporter concentration for the two steady state values. In order to globally verify the behavior of this large population of models, we focused on the 3,072 constructs potentially capable of bistability, 1,408 of which were found to be bistable. We further reduced the number of constructs used to verify the translation process from 3,072 to 384 by assuming that two constructs differing only in the RBS in 5' of the reporter gene would produce the same ratio of steady state values. Figure 4.3 visualizes the behavior of these 384 constructs. Constructs that are not bistable have a ratio of 1. This ratio gives insight into how the construct is expected to be experimentally detectable. Since most experimental methods cannot give an exact value of protein concentration, a high ratio is desired to rise above experimental noise. Each of the 6 windows is analogous to the previously described two-parameter bifurcation diagram for that pair of repressors (24). This gives confidence that both the semantic model of DNA sequences and the compiler used to translate automatically generated DNA sequences give results consistent with manually developed models of this family of gene networks. In the long term, the advantage to our approach over a traditional two-parameter bifurcation is the association of discrete parameter values with specific parts. This will prove particularly valuable when the context-dependencies of parameter values are better documented experimentally.

This example demonstrates the benefit of building a semantic model of synthetic DNA sequences. Even a small library of genetic parts can generate large numbers of artificial gene networks having no more than a few interacting genes. A syntactic model describing how parts can be combined into constructs is a compact representation of the genetic design space generated from the parts library. While it is possible to manually build mathematical models capturing the dynamics of some of these artificial gene networks individually, it becomes desirable to automate the process to ensure the model consistency when building large families of related models derived from the same parts library. By considering genetic parts as the terminal symbols of an attribute grammar, it becomes possible to automatically generate models of numerous artificial gene networks derived from this parts library and quickly identify the optimal designs (71).

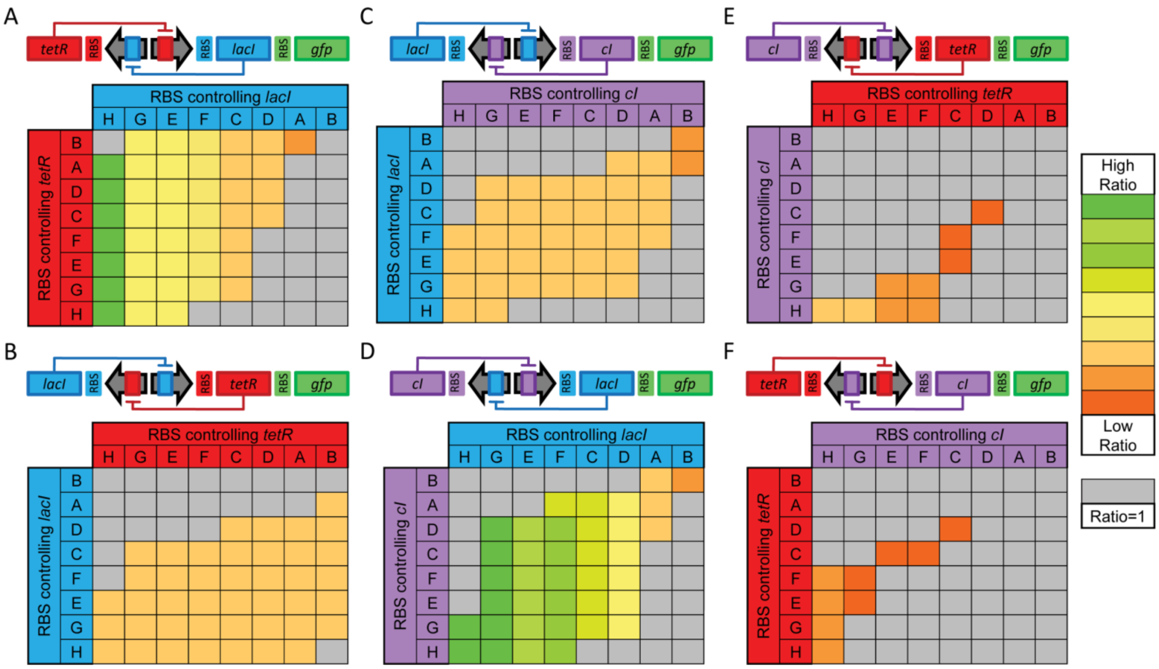

Figure 4.3: Mapping the behavior of 384 genetic constructs.
The map is organized by pairs of repressor genes and by RBSs connected to each repressor. RBSs are ordered by translational efficiency from low (H) to high (B), as determined by qualitatively fitting the results of Gardner et al. (8). Gene pairs that cannot lead to bistable behaviors are excluded from this map as described in the text. Numbers indicate the detectability ratio, defined as the steady state GFP concentration in the "on" state divided by the concentration in the "off" state. Monostable constructs have a ratio of 1. The ratio gives a measure of how easily the two steady states can be distinguished, which is important due to the large amount of experimental noise. Colors give a visualization of the relative detectability as shown in the legend to the right. The six panes show a portion of the two parameter cusp bifurcation for the overall system. For each repressor pair, the bifurcation diagram is shown with the window delineated by the RBS values displayed. The cusp borders reveal the transition between monostability and bistability. The behavior of constructs inside the cusp is robust. Constructs having the same pairs of repressors in reverse order are located in the same column. They show the structure of the cusp reflected over the x-y axis, but have different detectability values.

4.4 Discussion

4.4.1 Computer Assisted Design of synthetic genetic constructs

The parameter values used in the previous example were selected to match an extremely small set of six experimental data points. Although the under-determination of the model does not make it possible to precisely estimate the value of these parameters, the example illustrates how the framework could provide valuable guidance in selecting specific parts for a design. Considering that the exact value of parameters for parts is still a far off perspective, the automatic exploration of the design space presented here will provide useful guidance in construct design. For example, robust constructs from the cusp interior of the *tetR*/*cI* and *lacI*/*cI* pairings could be built and tested while less robust switches based on the *lacI*/*tetR* pairing would be avoided. As more is learned about these parts including the specific rates in different genetic contexts, the predictive ability of such maps will increase.

The approach presented in this report will be implemented into GenoCAD (72), the web-based tool we have developed to give biologists access to our syntactic design framework. Through GenoCAD, users will benefit from the syntactic and semantic models of various parts sources (GenoCAD provided library, MIT Registry of Standard Biological Parts, or user created parts library). Initially, users will be able to translate their designs into SBML files that could be imported in SBML-compliant simulation tools (www.sbml.org/SBML_Software_Guide) for further analysis. At a later stage, simulation results and more advanced numerical analyses will be seamlessly integrated in GenoCAD's workflow. One of the major obstacles toward the implementation of such semantic models in GenoCAD is the development of a data model allowing users to understand and possibly edit the functional model of the parts they use.

A function description language called Genetic Engineering of living Cells (GEC) was recently introduced to specify the properties of a design (67). GEC is capable of finding a DNA sequence that implements the desirable phenotypic functions. Several other software applications have been recently released to design biological systems from standardized genetic parts. ASMPART (65), SynBioSS (66), a specialized ProMot package (64) and TinkerCell (www.tinkercell.com) illustrate this trend. These tools are still exploratory. One of their limitations is the requirement to define parts in a specialized format, such as SBML or Modeling Description Language (MDL). Furthermore, instead of defining parts interactions in the underlying parts data models, these tools rely on the user to manually define them textually (66) or graphically (64). As a result of this specific limitation, several of these tools do not appear suitable for the automatic exploration of a design space. Moreover, they tend to rely on a loosely defined relationship between the structure of the genetic constructs and their behavior. They allow parts to be assembled in any order without regard for biological viability.

Still, the scripts developed to generate our results is of lesser importance than the application of the theory of semantics-based translation using attribute grammars upon which many computer languages and their compilers rely (56,73) to translate DNA sequences into dynamical models representing the molecular interactions they encode. As a result, a wealth of existing theoretical results and software tools can find new applications in the life sciences. For instance, we have implemented semantic models of DNA sequences into two widely used but very different programming environments, Prolog (74) and ANTLR (75). Future research efforts will need to investigate the pros and cons of different compiler generators and different parsing algorithms for analyzing even genome-scale DNA sequences and how they impact the ability of grammars to express various features of DNA sequences. Also, the type of attributes associated with parts is flexible. Here we primarily use mass action kinetic rates as attributes, but we could just as easily have used the emerging synthetic biology measurement units like polymerase per second (PoPS) (37,76).

Ultimately, tools capable of automatically generating models of the behavior of synthetic DNA sequences will be important for the advancement of synthetic biology (71). However, these tools will need to be able to express that the contribution of a genetic part to the phenotype of an organism depends largely on the local and global context in which it is placed. The interference between RBS and coding sequence is just one example of the biological complexity that computer assisted design applications will have to properly consider.

4.4.2 Functional characterization of genetic parts

Before it will be used to build synthetic genetic systems meeting user-defined specifications, the semantic model of DNA sequences presented in this report will be instrumental in the quantitative characterization of structure-function relationships in synthetic DNA sequences. The vision of applying quantitative engineering methods to biological problems has been recognized as promising avenue to biological discovery (29). The critical role of artificial gene networks in the characterization of molecular noise affecting the dynamics of gene networks (77) illustrates the potential of synthetic biology as a route to refine the understanding of basic biological processes.

Ongoing efforts aim to carefully define how parts should fit together syntactically and what attributes are needed to characterize their function. For example, the sequence between the RBS and the start codon has been shown to play an important role in translation rate (78). The question arises whether the RBS should be defined to include the spacing, or if there should be a separate parts category for the spacer. The rapid development of gene synthesis techniques (79) will make it possible to investigate these questions with a base-level resolution. Beyond libraries of parts for designing expression vectors, similar curation efforts could lead to the identification of parts in genomic sequences, whereby the hypothetical function of these parts as they are expressed in attribute grammars could be tested by genome refactoring (33).

4.5 Acknowledgements

The authors would like to thank Drs. Jacques Cohen and Mark Cooper for their critical reading of an early version of this manuscript, Stephan Hoops for helping us use COPASI in batch mode, and Emily Alberts for her editorial skills.

4.6 References

1. Keller, E.F. (2000) *The century of the gene*. Harvard University Press, Cambridge, Mass.
2. Sturtevant, A.H. (2001) *A history of genetics*. Cold Spring Harbor Laboratory Press, Cold Spring Harbor, N.Y.
3. Singer, M. and Berg, P. (2004) Timeline - George Beadle: from genes to proteins. *Nature Reviews Genetics*, **5**, 949-954.
4. Tatum, E.L. (1959) A case history in biological research. *Science*, **129**, 1711-1715.
5. Lynch, M. and Walsh, B. (1998) *Genetics and analysis of quantitative traits*. Sinauer, Sunderland.
6. Falconer, D.S. and MacKay, T.F.C. (1996) *Quantitative Genetics*. Longman Group Ltd., Harlow (U.K.).
7. Guldener, U., Munsterkotter, M., Kastenmuller, G., Strack, N., van Helden, J., Lemer, C., Richelles, J., Wodak, S.J., Garcia-Martinez, J., Perez-Ortin, J.E. *et al.* (2005) CYGD: the Comprehensive Yeast Genome Database. *Nucleic Acids Res*, **33**, D364-368.
8. Weng, S., Dong, Q., Balakrishnan, R., Christie, K., Costanzo, M., Dolinski, K., Dwight, S.S., Engel, S., Fisk, D.G., Hong, E. *et al.* (2003) Saccharomyces Genome Database (SGD) provides biochemical and structural information for budding yeast proteins. *Nucleic Acids Res*, **31**, 216-218.
9. Keller, E.F. and Harel, D. (2007) Beyond the gene. *PLoS ONE*, **2**, e1231.
10. Chen, K.C., Calzone, L., Csikasz-Nagy, A., Cross, F.R., Novak, B. and Tyson, J.J. (2004) Integrative analysis of cell cycle control in budding yeast. *Molecular Biology of the Cell*, **15**, 3841-3862.
11. Stricker, J., Cookson, S., Bennett, M.R., Mather, W.H., Tsimring, L.S. and Hasty, J. (2008) A fast, robust and tunable synthetic gene oscillator. *Nature*, **456**, 516-519.
12. Wang, S.Y., Zhang, Y.P. and Qi, O.Y. (2006) Stochastic model of coliphage lambda regulatory network. *Physical Review E*, **73**.
13. Tigges, M., Marquez-Lago, T.T., Stelling, J. and Fussenegger, M. (2009) A tunable synthetic mammalian oscillator. *Nature*, **457**, 309-312.
14. von Dassow, G., Meir, E., Munro, E.M. and Odell, G.M. (2000) The segment polarity network is a robust developmental module. *Nature*, **406**, 188-192.
15. Ramsey, S.A., Smith, J.J., Orrell, D., Marelli, M., Petersen, T.W., de Atauri, P., Bolouri, H. and Aitchison, J.D. (2006) Dual feedback loops in the GAL regulon suppress cellular heterogeneity in yeast. *Nat Genet*, **38**, 1082-1087.
16. Bains, W. (2001) The parts list of life. *Nat. Biotechnol.*, **19**, 401-402.
17. Brasch, M.A., Hartley, J.L. and Vidal, M. (2004) ORFeome cloning and systems biology: Standardized mass production of the parts from the parts-list. *Genome Res.*, **14**, 2001-2009.
18. Stewart, C.N. (2005) Plant functional genomics: beyond the parts list. *Trends in Plant Science*, **10**, 561-562.

19. Fitzkee, N.C., Fleming, P.J., Gong, H.P., Panasik, N., Street, T.O. and Rose, G.D. (2005) Are proteins made from a limited parts list? *Trends Biochem. Sci.*, **30**, 73-80.
20. Eggert, U.S., Mitchison, T.J. and Field, C.M. (2006) Animal cytokinesis: From parts list to mechanisms. *Annu. Rev. Biochem.*, **75**, 543-566.
21. Mueller, M., Martens, L. and Apweiler, R. (2007) Annotating the human proteome: Beyond establishing a parts list. *Biochimica Et Biophysica Acta-Proteins and Proteomics*, **1774**, 175-191.
22. Rabani, M., Kertesz, M. and Segal, E. (2008) Computational prediction of RNA structural motifs involved in posttranscriptional regulatory processes. *Proc. Natl. Acad. Sci. U. S. A.*, **105**, 14885-14890.
23. Segal, E., Fondufe-Mittendorf, Y., Chen, L., Thastrom, A., Field, Y., Moore, I.K., Wang, J.P. and Widom, J. (2006) A genomic code for nucleosome positioning. *Nature*, **442**, 772-778.
24. Gardner, T.S., Cantor, C.R. and Collins, J.J. (2000) Construction of a genetic toggle switch in Escherichia coli. *Nature*, **403**, 339-342.
25. Guet, C.C., Elowitz, M.B., Hsing, W. and Leibler, S. (2002) Combinatorial synthesis of genetic networks. *Science*, **296**, 1466-1470.
26. Cox, R.S., 3rd, Surette, M.G. and Elowitz, M.B. (2007) Programming gene expression with combinatorial promoters. *Mol Syst Biol*, **3**, 145.
27. Gertz, J., Siggia, E.D. and Cohen, B.A. (2009) Analysis of combinatorial cis-regulation in synthetic and genomic promoters. *Nature*, **457**, 215-218.
28. Murphy, K.F., Balazsi, G. and Collins, J.J. (2007) Combinatorial promoter design for engineering noisy gene expression. *Proc. Natl. Acad. Sci. U. S. A.*, **104**, 12726-12731.
29. Drubin, D.A., Way, J.C. and Silver, P.A. (2007) Designing biological systems. *Genes Dev.*, **21**, 242-254.
30. Agarwal, K.L., Buchi, H., Caruthers, M.H., Gupta, N., Khorana, H.G., Kleppe, K., Kumar, A., Ohtsuka, E., Rajbhandary, U.L., Van de Sande, J.H. *et al.* (1970) Total synthesis of the gene for an alanine transfer ribonucleic acid from yeast. *Nature*, **227**, 27-34.
31. Kay, L.E. (2000) *Who wrote the book of life? : a history of the genetic code*. Stanford University Press, Stanford, Calif.
32. Dymond, J.S., Scheifele, L.Z., Richardson, S., Lee, P., Chandrasegaran, S., Bader, J.S. and Boeke, J.D. (2009) Teaching Synthetic Biology, Bioinformatics and Engineering to Undergraduates: The Interdisciplinary Build-a-Genome Course. *Genetics*, **181**, 13-21.
33. Chan, L.Y., Kosuri, S. and Endy, D. (2005) Refactoring bacteriophage T7. *Mol Syst Biol*, **1**, 2005.0018.
34. 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. *et al.* (2008) Complete Chemical Synthesis, Assembly, and Cloning of a Mycoplasma genitalium Genome. *Science*, **319**, 1215-1220.
35. Endy, D. (2005) Foundations for engineering biology. *Nature*, **438**, 449-453.

36. Peccoud, J., Blauvelt, M.F., Cai, Y., Cooper, K.L., Crasta, O., DeLalla, E.C., Evans, C., Folkerts, O., Lyons, B.M., Mane, S.P. *et al.* (2008) Targeted Development of Registries of Biological Parts. *PLoS ONE*, **3**, e2671.
37. Canton, B., Labno, A. and Endy, D. (2008) Refinement and standardization of synthetic biological parts and devices. *Nat. Biotechnol.*, **26**, 787-793.
38. Coleman, J.R., Papamichail, D., Skiena, S., Futcher, B., Wimmer, E. and Mueller, S. (2008) Virus attenuation by genome-scale changes in codon pair bias. *Science*, **320**, 1784-1787.
39. Gimona, M. (2006) Protein linguistics - a grammar for modular protein assembly? *Nature Reviews Molecular Cell Biology*, **7**, 68-73.
40. Chiang, D., Joshi, A.K. and Searls, D.B. (2006) Grammatical representations of macromolecular structure. *J. Comput. Biol.*, **13**, 1077-1100.
41. Knudsen, B. and Hein, J. (2003) Pfold: RNA secondary structure prediction using stochastic context-free grammars. *Nucleic Acids Res*, **31**, 3423-3428.
42. Searls, D.B. (2002) The language of genes. *Nature.*, **420**, 211-217.
43. Rivas, E. and Eddy, S.R. (2000) The language of RNA: a formal grammar that includes pseudoknots. *Bioinformatics*, **16**, 334-340.
44. Searls, D.B. (1997) Linguistic approaches to biological sequences. *Comput. Appl. Biosci.*, **13**, 333-344.
45. Dong, S. and Searls, D.B. (1994) Gene Structure Prediction by Linguistic Methods. *Genomics*, **23**, 540-551.
46. Searls, D.B. (1992) The Linguistics of DNA. *American Scientist*, **80**, 579-591.
47. Bentolila, S. (1996) A grammar describing 'biological binding operators' to model gene regulation. *Biochimie*, **78**, 335-350.
48. Eilbeck, K., Lewis, S.E., Mungall, C.J., Yandell, M., Stein, L., Durbin, R. and Ashburner, M. (2005) The Sequence Ontology: a tool for the unification of genome annotations. *Genome Biology*, **6**, -.
49. Beisswanger, E., Lee, V., Kim, J.J., Rebholz-Schuhmann, D., Splendiani, A., Dameron, O., Schulz, S. and Hahn, U. (2008) Gene Regulation Ontology (GRO): design principles and use cases. *Stud. Health Technol. Inform.*, **136**, 9-14.
50. Cai, Y., Hartnett, B., Gustafsson, C. and Peccoud, J. (2007) A syntactic model to design and verify synthetic genetic constructs derived from standard biological parts. *Bioinformatics*, **23**, 2760-2767.
51. Elowitz, M.B. and Leibler, S. (2000) A synthetic oscillatory network of transcriptional regulators. *Nature*, **403**, 335-338.
52. Paakki, J. (1995) Attribute Grammar Paradigms - a High-Level Methodology in Language Implementation. *Acm Computing Surveys*, **27**, 196-255.
53. Knuth, D.E. (1968) Semantics of context-free languages. *Mathematical Systems Theory*, **2**, 127-145.
54. Knuth, D.E. (1990) The Genesis of Attribute Grammars. *Lecture Notes in Computer Science*, **461**, 1-12.
55. Stoy, J. (1977) *Denotational semantics : the Scott-Strachey approach to programming language theory*. MIT Press, Cambridge, Mass.
56. Slonneger, K. and Kurtz, B.L. (1995) *Formal syntax and semantics of programming languages : a laboratory based approach*. Addison-Wesley Pub. Co., Reading, Mass.

57. Lewontin, R.C. (2000) *The Triple Helix*. Harvard University Press, Cambridge, MA.

58. Hoops, S., Sahle, S., Gauges, R., Lee, C., Pahle, J., Simus, N., Singhal, M., Xu, L., Mendes, P. and Kummer, U. (2006) COPASI--a COmplex PAthway SImulator. *Bioinformatics*, **22**, 3067-3074.

59. Griffith, M., Courtney, T., Peccoud, J. and Sanders, W.H. (2006) Dynamic partitioning for hybrid simulation of the bistable HIV-1 transactivation network. *Bioinformatics*, **22**, 2782-2789

60. Adalsteinsson, D., McMillen, D. and Elston, T.C. (2004) Biochemical Network Stochastic Simulator (BioNetS): software for stochastic modeling of biochemical networks. *BMC Bioinformatics*, **5**, 24.

61. Moore, R.C. (2000), *6th Applied Natural Language Processing Conference/1st Meeting of the North American Chapter of the Association for Computational Linguistics, Proceedings of the Conference and Proceedings of the Anlp-Naacl 2000 Student Research Workshop*. Morgan Kaufmann, Vol. 6, pp. A249-A255.

62. Shultzaberger, R.K., Bucheimer, R.E., Rudd, K.E. and Schneider, T.D. (2001) Anatomy of Escherichia coli ribosome binding sites. *J. Mol. Biol.*, **313**, 215-228.

63. Vellanoweth, R.L. and Rabinowitz, J.C. (1992) The influence of ribosome-binding-site elements on translational efficiency in Bacillus subtilis and Escherichia coli in vivo. *Mol. Microbiol.*, **6**, 1105-1114.

64. Marchisio, M.A. and Stelling, J. (2008) Computational design of synthetic gene circuits with composable parts. *Bioinformatics*, **24**, 1903-1910.

65. Rodrigo, G., Carrera, J. and Jaramillo, A. (2007) Asmparts: assembly of biological model parts. *Syst Synth Biol*, **1**, 167-170.

66. Hill, A.D., Tomshine, J.R., Weeding, E.M., Sotiropoulos, V. and Kaznessis, Y.N. (2008) SynBioSS: the synthetic biology modeling suite. *Bioinformatics*, **24**, 2551-2553.

67. Pedersen, M. and Philipps, A. (2009) Toward programming languages for synthetic biology. *Journal of the Royal Society, Interface / the Royal Society*, **(in press)**.

68. Desmit, M.H. and Vanduin, J. (1990) Secondary Structure of the Ribosome Binding-Site Determines Translational Efficiency - a Quantitative-Analysis. *Proc. Natl. Acad. Sci. U. S. A.*, **87**, 7668-7672.

69. Desmit, M.H. and Vanduin, J. (1994) Control of translation of messenger-RNA secondary structure in Escherichia coli - A quantitative analysis of literature data. *J. Mol. Biol.*, **244**, 144-150.

70. Kudla, G., Murray, A.W., Tollervey, D. and Plotkin, J.B. (2009) Coding-Sequence Determinants of Gene Expression in Escherichia coli. *Science*, **324**, 255-258.

71. Goler, J.A., Bramlett, B.W. and Peccoud, J. (2008) Genetic design: rising above the sequence. *Trends Biotechnol*, **26**, 538-544.

72. Czar, M.J., Cai, Y. and Peccoud, J. (2009) Writing DNA with GenoCADTM. *Nucleic Acids Res*.

73. Appel, A.W. and Palsberg, J. (2002) *Modern compiler implementation in Java*. 2nd ed. Cambridge University Press, Cambridge, UK ; New York, NY, USA.

74. Bratko, I. (1986) *Prolog programming for artificial intelligence*. Addison-Wesley, Wokingham, England ; Reading, Mass.
75. Parr, T. (2007) *The complete ANTLR reference guide*. Pragmatic, Lewisville, TX.
76. Kelly, J.R., Rubin, A.J., Davis, J.H., Ajo-Franklin, C.M., Cumbers, J., Czar, M.J., de Mora, K., Glieberman, A.L., Monie, D.D. and Endy, D. (2009) Measuring the activity of BioBrick promoters using an in vivo reference standard. *J Biol Eng*, **3**, 4.
77. Raj, A. and van Oudenaarden, A. (2008) Nature, Nurture, or Chance: Stochastic Gene Expression and Its Consequences. *Cell*, **135**, 216-226.
78. Robert Luis Vellanoweth, J.C.R. (1992) The influence of ribosome-binding-site elements on translational efficiency in *Bacillus subtilis* and *Escherichia coli in vivo*. *Molecular Microbiology*, **6**, 1105-1114.
79. Czar, M.J., Anderson, J.C., Bader, J.S. and Peccoud, J. (2009) Gene synthesis demystified. *Trends Biotechnol.*, **27**, 63-72.

Chapter 5: Estimation of stochastic gene expression rate parameters from imaging cytometry data

Authors

Matthew W Lux[1], David A Ball[1], William T. Baumann[2], Jean Peccoud[1,3]

[1]Virginia Bioinformatics Institute, Virginia Tech, Blacksburg, VA 24061, USA
[2]Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg VA 24061, USA
[3]ICTAS Center for Systems Biology of Engineered Tissues, Virginia Tech, Blacksburg, VA 24061, USA

Abstract

Synthetic biology depends on the ability to predict the function of devices composed of well-characterized genetic components. Even though the community has long recognized the need to publish functional parameters in standardized parts data sheets, limitations of commonly available measurement techniques result in insufficient information to properly estimate parameter values. We used adaptive imaging cytometry to collect time-series data of gene expression in individual cells expressing a fluorescent protein under the control of an inducible promoter. We fit a stochastic model of gene expression to the statistical distributions of single-cell data over time. We demonstrate that we cannot estimate all gene expression rate parameters with protein data alone, but that we can constrain the estimates of all parameters by measuring mRNA with Fluorescence In Situ Hybridization. We report rate parameters in units of molecules/time rather than rates relative to some standard, which will facilitate the reliable characterization of parts.

Publication status

This article has been submitted for review and is awaiting decision.

5.1 Introduction

The ultimate goal of synthetic biology is the engineering of artificial genetic systems designed to propose alternative and sustainable solutions to humankind's most pressing needs (Khalil & Collins, 2010). It is envisioned that these systems will be composed of simple devices performing specific functions such as environmental sensing, switching, or logic analysis (Moon et al, 2012b; Purnick & Weiss, 2009; Regot et al, 2011; Tabor et al, 2009; Tamsir et al, 2011) The devices themselves are combinations of genes, promoters, and other genetic elements called parts (Galdzicki et al, 2011; Peccoud et al, 2008; Slusarczyk et al, 2012). This compelling vision of a predictable engineering of synthetic genetic systems relies on the concept of functional composition of DNA sequences. Being able to predict the behavior of combinations of parts or devices from the behavior of their components would greatly facilitate the development of complex systems composed of large numbers of parts.

This perspective calls for an unprecedented effort to quantitatively characterize the function of genetic parts through specific characterization experiments and reduction of the resulting datasets into mathematical models of parts' functions. One cannot underestimate the need for high-quality functional models. Since non-linear interactions between genetic parts can lead to qualitatively different behaviors depending on small quantitative differences in parts' functions, inaccuracies in the parts' functional models propagate to models of devices and systems jeopardizing the predictable functional composition of genetic elements (Ellis et al, 2009a). Synthetic biology projects are still plagued with years of tedious tuning by trial-and-error as a direct result of the lack of good estimates of the functional parameters of genetic parts (Kwok, 2010).

While the need to characterize parts has been recognized for some time (Canton et al, 2008; Kelly et al, 2009), estimates of parts' parameters are still remarkably scarce. For example, the maturation of fluorescent proteins *in vivo* remains poorly characterized despite their ubiquitous use as reporters of the dynamics of artificial and natural gene networks. This observation is puzzling since the molecular basis of the maturation of fluorescent proteins is simple and well understood. Several authors have described protocols to estimate this parameter. Yet, only a handful of estimates have been published (Ajo-Franklin et al, 2007; Gordon et al, 2007) leaving most of the fluorescent proteins largely undocumented.

This situation is a direct consequence of the inherent difficulty of estimating rate parameters. Current measurement techniques are only capable of partially observing the dynamics of genetic devices. Only a small fraction of the molecular species represented by variables in mathematical models of device behaviors is measurable, typically by the use of no more than a few fluorescent proteins. Hence, the experimental dataset does not allow the estimation of all the parameters in the underlying model (Jaqaman & Danuser, 2006; Raue et al, 2011) resulting in structurally non-identifiable model parameters (Raue et al, 2009). In addition, the limited amount and quality of the dataset can hamper the possibility of obtaining good estimates of some model parameters, a situation known as practical non-identifiability. Acknowledging the relationship between identifiability and measurability naturally leads to proposing model-driven experimental designs that minimize the risks of developing under-determined models with key parameter values that cannot be estimated from experimental datasets (Balsa-Canto et al, 2010; Raue et al, 2011).

Two routes are available to resolve model under-determination. Simplifying the model representing the function of genetic parts can reduce the number of parameters to be estimated. One approach consists of estimating relative behaviors among component types and variants (Anderson et al, 2007; Ellis et al, 2009b; Kelly et al, 2009; Salis et al, 2009; Wang et al, 2011) (Endy, personal communication). It is also possible to eliminate parameters by characterizing the model-steady state and ignoring the dynamics of the response to environmental perturbations (Regot et al, 2011). Similarly, models of systems constructed of distinct cell populations that communicate can ignore the effect of noise observed in individual cells because of population averaging (Danino et al, 2010; Prindle et al, 2012; Tamsir et al, 2011). These results have demonstrated that a model-driven design of complex genetic devices is possible. Selecting components that are “stronger” or “weaker” than others from a library of parts can go a long way towards tuning complex systems (Litcofsky et al, 2012; Moon et al, 2012a).

Extracting more information than is possible with traditional instruments is another approach to resolving model under-determination. New instruments combining fluorescent microscopy (Locke & Elowitz, 2009) and microfluidics (Ferry et al, 2011) are providing dense time series of gene expression data in individual cells. It has been demonstrated that the statistical distribution of single cell data can be related to the structure of the underlying gene expression mechanism (Munsky et al, 2012). In this work we describe our efforts to merge these advances to estimate physical parameters of gene expression for a single eukaryotic gene expression cassette. Specifically, we reused a previously described yeast strain that expresses the Venus fluorescent protein under the control of the endogenous GAL1 promoter (Raser & O'Shea, 2004). Our goal was to estimate the relevant parameters in absolute units and in the absence of comparisons to any reference standard, allowing the values to stand alone, consequently setting the stage for use in systematic characterization of libraries of components with standardized data sheets (Canton et al, 2008) and exploration of the dependencies of these rates on genetic and environmental contexts.

5.2 Results

5.2.1 Adaptive cytometry

We have developed an automated imaging cytometry platform, GenoSIGHT, to collect time course data on individual cells. The hardware includes an incubation chamber and computer-controllable microfluidics system to enable control of as many environmental variables as possible. The software collects and analyzes phase contrast and fluorescence images to automatically identify suitable fields-of-view, monitor cell count, and track changes in fluorescence. The images and resulting information are processed in real-time to allow for automatic control of an experiment via feedback through the microfluidics device (Figure 5.1a). This closed-loop control of the image acquisition process allows the collection of richer datasets than is possible using the more traditional open-loop control used to program imaging workstations. The system optimizes the use of a limited data acquisition bandwidth by focusing on the most informative fields of view based on user-defined balance between time resolution and total number of cells observed. It also uses a user-defined description of the experiment to coordinate image acquisition and environmental perturbations by changing media in the microfluidic system in response to observed changes in variables such as cell count and cell fluorescence. The idea of intelligent acquisition of microscopy images has been explored theoretically (Jackson et al, 2011; Jackson et al, 2009) but never implemented before.

Another benefit of this approach is a substantial reduction in experimental labor costs. Experiments that might have taken a week of off-and-on attention to prepare, run, and process can now be condensed to less than an hour of setup and a few hours of unmonitored execution. For instance, the system can detect problems in an experiment and alert the experimenter by email or text message, potentially averting wasted hours on a failed experiment. The operator is also provided with statistical plots generated on the fly as with flow cytometry. This high-level information can help detect problems arising during parts characterization experiments.

5.2.2 Model

We modeled our biological system using a variant of the two-state gene expression model (Peccoud & Ycart, 1995; Shahrezaei & Swain, 2008; So et al, 2011) in which the promoter switches stochastically between an inactive (OFF) and an active (ON) state. Specifically, we assumed that the transition between the OFF and ON state is irreversible since we performed only induction experiments. The model also includes a reaction corresponding to the transition of the fluorescent protein between an immature state and a fluorescent conformation. We use mass-action equations for each of the key steps in inducible gene expression (with corresponding rate parameters): promoter activation (*kACT*), transcription (*kTX*), mRNA degradation (*kMDEG*), translation (*kTL*), and protein maturation (*kMAT*) (Figure 5.1b, Table B.1, Table B.2). We also include a burst parameter (*B*) corresponding to the number of mRNA's produced at each transcription event. For the initial analysis of protein fluorescence data, we keep *B* fixed to 1.

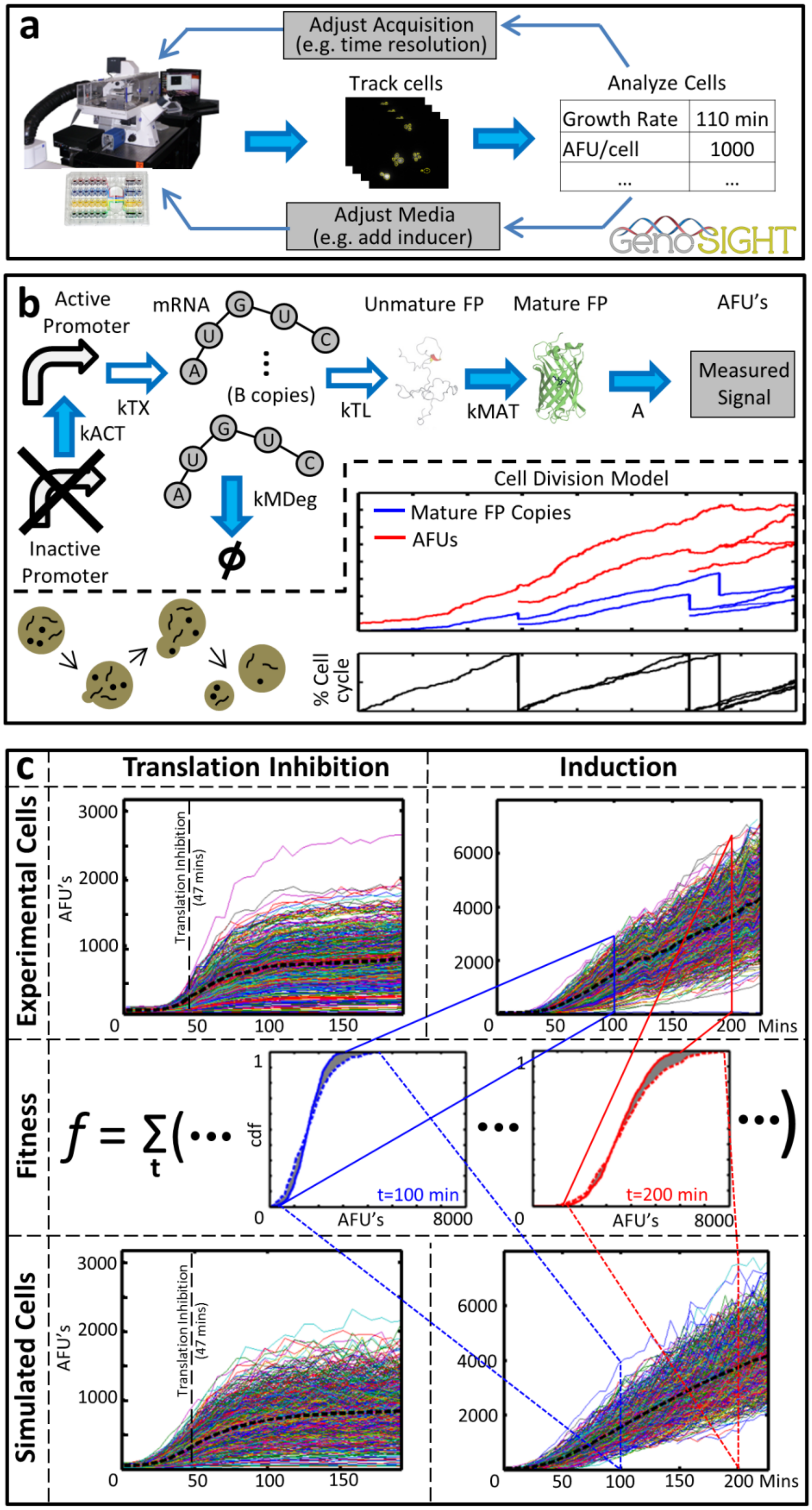
a
Adjust Acquisition
(e.g. time resolution)
Track cells
Analyze Cells
Growth Rate
110 min
AFU/cell
1000
...
...
Adjust Media
(e.g. add inducer)
GenoSIGHT
b
Active Promoter
mRNA
G
U
C
U
A
(B copies)
kTX
kACT
Unmature FP
Mature FP
AFU's
kTL
kMAT
A
Measured Signal
kMDeg
ϕ
Inactive Promoter
Cell Division Model
Mature FP Copies
AFUs
% Cell cycle
c
Translation Inhibition
Induction
Experimental Cells
Fitness
Simulated Cells
AFU's
3000
2000
1000
0
50
100
150
Translation Inhibition (47 mins)
6000
4000
2000
200
Mins
f = ∑t (··· ··· ···)
cdf
1
0
AFU's
8000
t=100 min
t=200 min

Figure 5.1: Experimental Data and Model.
Panel (a) provides an overview of GenoSIGHT. Images are automatically converted to fluorescence data and analyzed, then fed back to adjust acquisition parameters or media content as appropriate. Panel (b) describes the model. Blue arrows indicate mass-action equations. Hollow blue arrows indicate that the reactant is not consumed in the reaction. Transcription events produce B mRNA copies. Details below the dashed divider describe the model of cell division, where cells stochastically progress through the cell cycle and divide, randomly partitioning species between mother and daughter. AFU's in a cell (red curves) are scaled by cell-cycle time to more accurately imitate experimental measurements, which are scaled by cell size. Panel (c) depicts typical experimental datasets (top) and simulated datasets (bottom) for inhibition (left) and non-inhibition (right) experiments, as well as the objective function (center).

The model does not include a protein degradation rate because the analysis of an earlier version of this model showed that protein degradation was negligible compared to the dilution of fluorescent proteins associated with cell division. In order to account for random partitions of proteins between mother and daughter cells during cell division (Huh & Paulsson, 2011), we model individual cells as progressing stochastically between 0% and 100% of the cell cycle. When a cell reaches 100%, a new cell is added to the simulation queue, and the model species are divided approximately 60%-40% (mother-daughter) by random selection from an appropriate binomial distribution. This step proved important for capturing the experimentally observed cell-cell variability (Figure B.1).

Finally, since gene expression data are collected in units of arbitrary fluorescence units (AFUs), we include a scaling parameter (A) to convert the number of mature fluorescent protein copies to AFUs. AFU values are also scaled by cell cycle time to approximate the scaling by area that takes place in the experimental system. Measurement errors and cellular auto-fluorescence are modeled by adding a random error term from a constant normal distribution to each simulated data point. The parameters of the normal distributions were derived from data collected on uninduced cells. Since we typically collect data on roughly 700 cells, we typically run simulations in batches of 700 runs of the stochastic model to mimic the population of cells measured experimentally.

5.2.3 Parameter estimation from protein data

We first wanted to evaluate if it was possible to estimate the model parameters solely from the Venus fluorescence data. We performed translation inhibition experiments to measure fluorescent protein maturation rates as previously described (Ajo-Franklin et al, 2007; Gordon et al, 2007) (Online Methods). Briefly, cells are grown in non-inducing media and added to the microfluidics device. Upon detection of a normal growth rate, the inducer molecule is automatically added. Once the cells start fluorescing, a translation inhibitor is added to the media. At this point, any new increase in fluorescence is assumed to be due to protein copies that have been translated, but have not yet matured into observable fluorescent protein copies. A typical experiment is shown in Figure 4.1c, upper left. We also performed induction experiments similar to the translation inhibition experiment except that no translation inhibitor is added, and cells continue to grow (Figure 5.1c, upper right). We performed each experiment in triplicate (Figure B.2, Figure B.3, Figure B.4, and Table B.3).

To take advantage of the time series of single-cell fluorescence data captured in our dataset, we created an objective function to compare the time-evolution of statistical distributions of experimental and simulated fluorescence data (Figure 5.1c, center). For a given parameter set, the objective function computes the cumulative distribution function (cdf) of both the experimental and simulated data at each experimental time point. The function then computes the sum-squared difference between the simulated and experimental cdfs at each time point and sums these differences over all the time points. Using this metric to assess how simulated data match experimental data, it is possible to find parameter sets that simulate datasets that are visually very similar to experimental data (Figure 5.1c, bottom; Figure B.5, Figure B.6).

5.2.4 Uncertainty of parameter estimates

We estimate parameter values by interactively searching parameter space guided by the structure of this particular model. Since this heuristic approach to parameter estimation does not guarantee that the solution is optimal locally, much less globally, we explored the parameter space surrounding these solutions to assess the sensitivity of the objective function to small variations of parameter values. Since the model is stochastic, two sets of 700 simulated cells will have slightly different fitness values as a result of sampling error. As the perturbation size becomes small, a change in the objective function due to a parameter perturbation becomes smaller than the fluctuations associated with sampling errors. To decide if fluctuations of fitness values are significant, we generated a distribution of objective function values by producing *n* sets of 700 simulated cells for each set of parameters. After testing the normality of these distributions of fitness values, a Student's t-test is used to assess the statistical significance of differences between the distributions of fitness values obtained with two sets of parameter values (α=.05, Online Methods). Since the objective function values are on a different scale for the two experiments, we consider the fold change from the unperturbed system instead of absolute values of the objective function (Figure 5.2a). In order to consider both experiments simultaneously, we use the average of the two values as our primary sensitivity metric. We find that this combined objective function (*Favg*) is indeed sensitive to perturbations in all parameters (Figure 5.2b, Figure B.7).

By noting the minimum perturbation size at which the objective function is statistically different from the one obtained for a specific set of parameter values, we can evaluate the "simulation uncertainty." This simulation uncertainty could theoretically be made arbitrarily small given unlimited computational power to increase the number of simulations; however, there is also uncertainty resulting from differences in experimental replicates. By considering the standard error of each parameter value as estimated from different experimental replicates, we establish an approximate bound on this "experimental uncertainty." Though these uncertainties are not directly comparable, at the point where simulation uncertainty becomes small compared to experimental uncertainty, there is little value in reducing the simulation uncertainty further by increasing *n* and the number of cells per simulation (Figure B.8). We use this concept to guide the choice of *n* throughout.

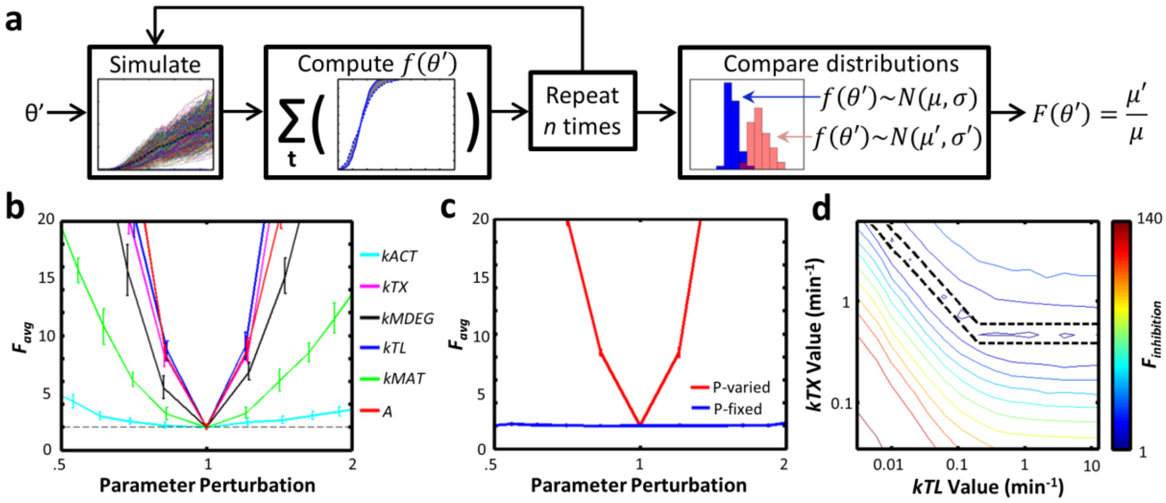

Figure 5.2: Sensitivity Analysis.
(a) We overcome the fact that stochastic simulation error is larger than the impact of small parameter perturbations by simulating a perturbed parameter set (θ') and computing the resulting objective function n times, and then determining if the perturbed parameter set results in a distribution of values that differs statistically from the same number of simulations of the unperturbed parameter set (θ). We use the fold change between mean objective function values, F (θ'), as the sensitivity metric. (b) The sensitivity of each parameter to multiplicative perturbations ranging from 0.5 to 2, with error bars representing SEM (all points different from θ by t-test, n=10). (c) The system is sensitive to perturbations in kTX, kTL, and A if they result in changes to the product P=kTX×kTL×A (P-varied, all points different from θ by t-test, n=10), but not if P remains constant (P-fixed, all points identical to θ by t-test, n=10)). (d) Focused analysis of P with increased n reveals a band of indistinguishable points (n=30).

Note that the shape of the sensitivity curves is highly similar for *kTX*, *kTL*, and *A* (Figure 5.2b, Figure B.7). This result indicates that the system may be sensitive to some combination of the parameters rather than each parameter individually. Indeed, we find that in the vicinity of our parameter estimates the system is sensitive to perturbations that change the product of the parameters, *P*=*kTX*×*kTL*×*A*, but seems to be insensitive to perturbations that maintain a constant value for *P* (Figure 5.2c, Figure B.9). However, intuition suggests that this insensitivity does not hold universally because at the limit where *kTX* and *kTL* are near zero, one would expect the vast majority of cells to have no FP copies, while exceedingly rare cells contain a single FP copy of near-infinite brightness. For large numbers of cells, this situation would maintain the mean fluorescence, but yield extremely different distribution shapes. For values closer to our solution, this concept manifests itself as a change in cell-cell variability; for *kTX* and *kTL* small and *A* large, stochastic noise increases cell-cell variability, while for *kTX* and *kTL* large and *A* small, cell-cell variability decreases (Figure B.10). The impact of this effect is subtle compared to changes in *P*. Yet by increasing *n* and focusing on perturbations of *kTX* and *kTL* while adjusting *A* to keep *P* fixed, a more detailed picture of the sensitivity to these parameters emerges. Figure 5.2d reveals a landscape of *kTX*/*kTL* pairs, including a region in which the solutions are indistinguishable (dashed lines) (Figure B.11). Simulating the induction experiment is roughly 10 times more computationally expensive than the inhibition experiment due to modeling of cell division, and initial probing showed that performing this analysis using simulations of the induction experiment did not provide any additional insights, so we only pursued inhibition simulations for these analyses. From these simulations it is possible to establish a minimum value for *kTX*, below which the cell-cell variability contributed by transcription is greater than that observed in the proteomic data experimentally. At the minimum value of *kTX*, *kTL* can become arbitrarily large because all of the cell-cell variability observed in the fluorescence data is generated by transcription (cell-partitioning error is negligible for the inhibition experiment). In this situation, for any translation rate fast enough to not contribute any significant amount of cell-cell variability, *kTL* effectively becomes another scaling parameter that cannot be disentangled from *A*. Thus, *kTL* and *A* remain unconstrained, at least for values that are conceivable biologically (translation half-times ranging roughly from 1sec-1hr). We hypothesized that this structural non-identifiability might be a result of the slow time-scale of the maturation step obscuring the underlying dynamics. However, repeating the analysis on a simulated dataset with very fast maturation resulted in the same picture (Figure B.12).

5.2.5 Refining parameter estimates with mRNA data

Since it was not possible to unambiguously estimate all the model parameters using time courses of proteomic data alone, we augmented this first data set with transcriptomic data collected by Fluorescence In Situ Hybridization (FISH) (Zenklusen et al, 2008) at 0, 20, 40, and 60 minutes after induction (Online Methods). Using the values for *kACT* and *kMDEG* as estimated from the inhibition data, we adjusted the transcription parameters to match the mRNA distributions collected by FISH. Changing *kTX* alone was not sufficient to match the experimental FISH distributions, so we also modified the burst parameter, *B*, and found a good match (Figure 5.3a). We verified the uniqueness of this solution by perturbing both *kTX* and *B* (Figure 5.3b, Figure B.13). With *kTX* and *B* known, we returned to the fluorescence data. We quickly found values for *kTL* and *A* that minimized the objective function, but these solutions are still non-unique for any reasonable values. As in the case described above, transcription is responsible for all of the experimentally observed cell-cell variability (cell-partitioning error is again negligible because this analysis focuses on the inhibition experiment). Thus, only for unreasonably slow translation rates can the objective function distinguish between *kTL*/*A* pairs, and even then the differences are small (Figure 5.3c, Figure B.14). We therefore report *kTL* and *A* as a product in Table 5.1.

The parameters agree well across 3 experimental replicates (Table 5.1). The model matches the data across most time points but the quality of the fit is marginal around the point when fluorescence is initially beginning to increase (Figure B.15). The same is true for the FISH data (Figure 5.3a). This discrepancy indicates that a single-step mass-action reaction is not adequate to capture the dynamics of the activation step. More complex models that incorporate the different stages of transcription initiation complex formation (Blake et al, 2003) or the detailed mechanism of *S. cerevisiae* response to the presence of galactose (Timson, 2007) might improve the fit.

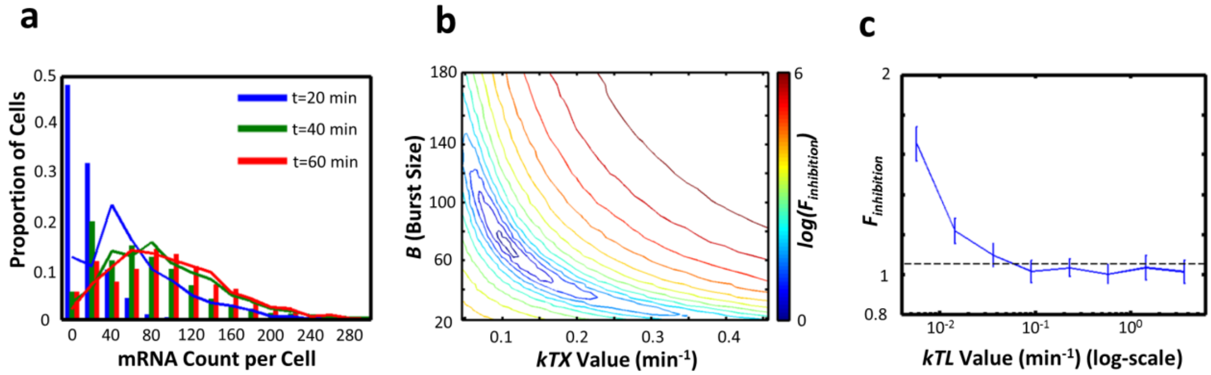

Figure 5.3: FISH Analysis.
(a) Distributions of mRNA counts in individual cells from FISH experiments (bars) and simulations (lines). (b) Sensitivity of the objective function for mRNA distributions to changes in kTX and B. (c) Sensitivity of the objective function to different values of kTL with A scaled to keep P constant (n=100).

The estimated rates are biologically reasonable. Our conclusion that the proteomic noise observed in the fluorescence experiments is predominantly due to (1) noise from stochastic activation of the promoter and bursty transcription (Bar-Even et al, 2006; Blake et al, 2003; Kar et al, 2009; Newman et al, 2006), and (2) cell partitioning errors (Huh & Paulsson, 2011) has been suggested elsewhere. The GAL promoter is reported to produce a >500-fold increase in transcription after induction (Bram et al, 1986); our results show a ~1200-fold induction after 60 mins (Online Methods). The rate of mRNA degradation falls within the range of values described in large-scale analyses of *S. cerevisiae* (Cacace et al, 2012). Since our translation rate is entangled with the instrument-dependent scaling parameter *A*, it is impossible to compare our results to others, though the only available comparisons are from experiments using population extracts that would need to be scaled by volume (Lee et al, 2011). Reported values available to compare to our estimated maturation half-time of Venus (26 min) range widely. Two *in vitro* studies citing identical protocols reported oxidation half times (the rate limiting step in fluorophore formation) as 68 min (Kremers et al, 2006) and 1.44 min (Nagai et al, 2002), respectively. Another study in *E. coli* gave 7.0 ± 2.5 min (Yu et al, 2006). Each of these half times was determined at 37^{o} C and none were in *S. cerevisiae*, which might explain the slower rate presented here. Indeed, Nagai et al. also report an oxidation half time of 4.6 min for EYFP *in vitro* at 37^{o} C (Nagai et al, 2002), while Gordon et al. report an overall maturation half time of 39 min (Gordon et al, 2007) in *S. cerevisiae* at 30^{o} C using the protocol upon which ours is based. These discrepancies underscore the need to better understand the dependencies of gene expression rate parameters.

Parameter	Unit	Function	Rep 1	Rep 2	Rep 3	Mean ± SE (3 replicates)	Mean half-time
kACT	[min^{-1}]	Activation	0.11	0.15	0.14	0.13 ± 0.01	5.2 min
kTX	[min^{-1}]	Transcription	0.110	0.114	0.109	0.111 ± 0.002	6.2 min
B	[]	Burst Size	68.9	72.7	73.7	71.8 ± 1.5	n/a
kMDeg	[min^{-1}]	mRNA Decay	0.087	0.099	0.092	0.093 ± 0.005	7.5 min
kTL·A	[AFU · molecule^{-1} · min^{-1}]	Translation	0.38	0.35	0.40	0.37 ± 0.01	1.8 min
kMAT	[min^{-1}]	Maturation	0.025	0.027	0.030	0.027 ± 0.002	26 min

Table 5.1: Parameter estimates for three replicates.

5.3 Discussion

Our results demonstrate the potential of merging the high information density of imaging cytometry methods with stochastic model analysis that leverages statistical distributions of single cells in order to generate reference-free estimates of gene network parameters in standard physical units. Imaging cytometry provides higher time resolution than flow cytometry and integration with microfluidics technologies allows control over environmental variables not available with other approaches. The ability to automate imaging cytometry experiments makes the characterization of large sets of constructs more achievable. Combining the advantages of imaging cytometry with sophisticated analysis based on statistical distributions and stochastic models enables the disentanglement of structural non-identifiabilies that limit methods that consider only the mean and deterministic models. Further advances that automate the parameter estimation or leverage the information contained in the trajectories of individual cells should further enable estimation of rate parameters on a larger scale.

Yet, despite all the information collected (~28,000 images for 7 parameters) and advanced analysis, it was still not possible to accurately estimate all model parameters from the protein data alone. We were able to almost fully constrain the model by performing FISH experiments, but the protocol requires expensive oligonucleotide probes and significant labor investment before imaging. It is important to keep in mind that the possibility to estimate parameter values depends a lot on the chosen model. A compromise needs to be found between the quality of the model's fit to available data and the level of detail in the model. For instance, it would be possible to come up with a more complicated model of the promoter activation step, but it is not clear that we could estimate the additional parameters it would include. Similarly, we model transcription bursting as a stochastic event with fixed burst size corresponding to 2 parameters. A more accurate model might describe a reversible promoter transition to "burst mode" where a fast transcription rate exists, but whether we could uniquely estimate all 3 parameters is an open question. The relative magnitude of the parameters themselves can have an impact on the ability to estimate them. For instance, the slow maturation time of the fluorescent reporters can obscure underlying dynamics (Wang et al, 2008) and may limit estimation power from datasets such as presented here.

The estimation of absolute parameter values in individual constructs is a stepping-stone toward the systematic exploration of the context dependencies of specific rate parameters. By characterizing the behavior of families of related constructs derived from the same set of parts, it will be possible to detect how the specific genetic or environmental context may affect the dynamics of the behavior encoded in specific parts, and to then formalize them into models to account for the behavior. Data that cannot be accounted for by the model, such as the activation behavior described above, can point to deficiencies in the specific model used. By probing the appropriateness of alternative models, the system can also serve as a hypothesis generator for the presence of unknown regulation or context dependencies. Such capabilities should be of interest well beyond the confines of synthetic biology: systems biologists can build better models for individual genes, pharmaceutical companies can perform more detailed drug screens, and traditional cell biologists can test hypotheses on novel pathway structures with minimized effort. In synthetic biology, the ability to accurately capture stochastic effects by modeling absolute gene expression rates and how context modifies them should dramatically improve our ability to predict the function of novel combinations of genetic components.

5.4 Materials and methods

5.4.1 Cell cultures

Cells used in this study are *S. cervisiae* containing a chromosomally integrated copy of the YFP variant venus, created by Raser et al. (Raser & O'Shea, 2004) Cultures were grown overnight at 30^{o}C from plate stocks (synthetic complete (SC) media +2% glucose) in pre-warmed liquid media under non-inducing conditions (SC +2% raffinose). Cultures were diluted 1:50 after ~16 hours and grown for another ~8 hours. At this point, cells were diluted to an OD of ~0.1 and loaded onto the microfluidics plate (CellASIC) along with appropriate pre-warmed media (Well 1: SC +2% raffinose, Well 2: SC +2% galactose, both experiment types; Well 3: SC +2% galactose +100ug/mL cycloheximide, inhibition experiments only). Changes of media were programmed in software.

5.4.2 Imaging

All images were collected on an Axio Observer Z1 (Carl Zeiss MicroImaging, Inc.) microscope equipped with a halogen lamp for bright field imaging, and a 120 W Metal Halide lamp for fluorescence excitation. The microscope is controlled by custom software developed in MATLAB that relies on the API of the open-source microscopy control software, μManager (Arthur Edelstein, 2010). The custom software performs image processing tasks during acquisition, and among other features, allows for the automatic identification of suitable fields of view (FOVs), and therefore maximizes the number of cells that can be sampled. For each data set 30 FOVs were automatically selected from a user-defined area of the trapping region. All images were collected with a 63x Ph3, phase contrast objective (Carl Zeiss MicroImaging, Inc., LCI Plan-Neofluar 63x/1.3 Imm Corr Ph3), and a GFP filterset (Chroma Technology Corp., set 49002). Exposure times were 10 ms for Phase contrast and 75 ms for fluorescence.

5.4.3 Image processing and cell identification

Phase-contrast images are segmented using custom software derived from Yeast Tree 1.6.3 (Bean et al, 2006). The application relies on the MATLAB Image Processing toolbox. First, the function 'imfill' is used to flood-fill local minimum not connected to the image border, which fills in the center of the groups of cells. As each group of cells will have slightly different levels to which the flood-fill will rise, we then search the image histogram for intensities greater than the calculated background, taken from the border pixels, and with a frequency greater than the minimum cell area, generally set to 200 pixels. To keep only large groups of connected pixels, an erosion (built-in function 'imerode') is performed, removing the outermost pixels of a region and eliminating small groups of pixels. The next step is to separate these groups into individual cells. This is done with another call to 'imerode' to cut the small necks that appear between touching cells. Once the cells are cut, the remaining connected regions are labeled with a call to the built-in function 'bwlabel', which identifies the individual cells and assigns each with a unique label. To finish, the cells are returned to their original sizes with a dilation (built-in function 'imdilate'), which adds pixels around the edges of each cell.

5.4.4 Simulations

Models were built in SimBiology (The Mathworks) and simulated using a combination of SimBiology and custom MATLAB (The Mathworks) code. Stochastic simulations were executed using the Gillespie Algorithm as implemented in SimBiology. The MATLAB Parallel Computing Toolbox (The Mathworks) was used to accelerate simulation times.

5.4.5 Statistical testing

We used statistical tests to differentiate distributions of objective function values generated from multiple simulations of two parameter sets. We first check if each distribution is normally distributed by fitting a normal distribution to each set, and seeing if each distribution is statistically different from the corresponding fit by the Kolmogororov-Smirnov test (α=.05). This test validates the assumption of normality required by the Student's t-test, which we use to determine if the means of the two distributions are significantly different (α=.05). Both tests were carried out by functions provided within MATLAB software.

5.4.6 FISH experiments

Five 50-mer oligonucleotides were designed to target Venus mRNA. The oligos were synthesized commercially (BioSearch Technologies), and each contains 4-5 modified T’s that contain an amine group. A DyeLight 550 NHS Ester labeling kit (Thermo Scientific) was used to conjugate the fluorescent dye DyeLight 550 to the modified T’s. Probe hybridization was carried out as described in Zenklusen et al (Zenklusen et al, 2008). Single mRNA spot detection was carried out on the same microscopy platform as above. For each data set 20 FOVs were automatically selected from a user-defined area of the slide. A z-stack was collected at each FOV containing 31 focal planes separated by 0.2 mm, and 2 color channels: DyeLight 550 (Chroma filter set SP102v1, exposure time: 2 s/plane), and Hoechst 33342 (Chroma filter set 49000, exposure time: 150 ms/plane). In addition, a single phase-contrast image was acquired at the central plane for automated cell identification (see above). The three-dimensional fluorescence z-stacks were reduced to 2D images by use of a maximum Z-projection. Single spots, corresponding to single mRNA molecules, were then identified using the algorithm described by Thompson, et al.12 First, a local background subtraction, in which the mean of a 19x19 pixel neighborhood is subtracted from the central pixel, is used to highlight the bright punctate spots from the autofluorescence background of the cells. Then, pixels five standard deviations above the mean pixel value of the entire image are chosen as initial seeds for the spot detection routine. Gaussian weighting is then used to move each seed to the center of the 2D Gaussian intensity distribution. Spots with an integrated intensity of <350 are likely the result of non-specifically bound individual probes, and are therefore removed. To avoid double counting a single mRNA, if the centers of two spots are within 2 pixels of each other, then the dimmer of the two spots is removed. Some cells in the 20 min time point and many in the 40 and 60 min time points had far too many spots for the algorithm to accurately detect. We therefore approximated the number of mRNAs in each cell by taking the total fluorescence of the cell summed over the z-stacks, subtracting a background fluorescence corresponding to a non-induced cell of comparable size, and dividing by the average intensity of a single spot as computed from cells that did not saturate the detection algorithm. We computed fold induction by dividing the mean number of mRNAs per cell approximated at the 60 min time point by the mean number of mRNAs detected by the algorithm in cells at the 0 min time point.

5.5 Acknowledgements

We are grateful to Erin K. O’Shea for providing us with the yeast strain used for this work, Federick R. Cross for giving us the YeastTree software, and Daniel R. Zenklusen for generously providing advice in setting up FISH experiments. John J. Tyson and Brett M. Tyler provided valuable comments on early versions of the manuscript. This project is supported by NSF Awards 0850100 and 0963988 and by grant R01-GM078989 from the National Institutes of Health. MWL was supported by a fellowship from Science Application International Corporation and by a Department of Defense SMART Scholarship under OSD-T&E (Office of Secretary Defense-Test and Evaluation), Defense–Wide / PE0601120D8Z National Defense Education Program (NDEP) / BA-1, Basic Research. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

5.6 References

Ajo-Franklin CM, Drubin DA, Eskin JA, Gee EP, Landgraf D, Phillips I, Silver PA (2007) Rational design of memory in eukaryotic cells. *Genes Dev* **21:** 2271-2276

Anderson JC, Voigt CA, Arkin AP (2007) Environmental signal integration by a modular AND gate. *Molecular systems biology* **3:** 133

Arthur Edelstein NA, Karl Hoover, Ron Vale, and Nico Stuurman (2010) Computer Control of Microscopes Using µManager. *Current Protocols in Molecular Biology* **14:** 1-17

Balsa-Canto E, Alonso AA, Banga JR (2010) An iterative identification procedure for dynamic modeling of biochemical networks. *Bmc Systems Biology* **4**

Bar-Even A, Paulsson J, Maheshri N, Carmi M, O'Shea E, Pilpel Y, Barkai N (2006) Noise in protein expression scales with natural protein abundance. *Nature genetics* **38:** 636-643

Bean JM, Siggia ED, Cross FR (2006) Coherence and timing of cell cycle start examined at single-cell resolution. *Molecular cell* **21:** 3-14

Blake WJ, M KA, Cantor CR, Collins JJ (2003) Noise in eukaryotic gene expression. *Nature* **422:** 633-637

Bram RJ, Lue NF, Kornberg RD (1986) A GAL family of upstream activating sequences in yeast: roles in both induction and repression of transcription. *The EMBO journal* **5:** 603-608

Cacace F, Paci P, Cusimano V, Germani A, Farina L (2012) Stochastic Modeling of Expression Kinetics Identifies Messenger Half-Lives and Reveals Sequential Waves of Co-ordinated Transcription and Decay. *PLoS Comput Biol* **8:** e1002772

Canton B, Labno A, Endy D (2008) Refinement and standardization of synthetic biological parts and devices. *Nat Biotechnol* **26:** 787-793

Danino T, Mondragon-Palomino O, Tsimring L, Hasty J (2010) A synchronized quorum of genetic clocks. *Nature* **463:** 326-330

Ellis T, Wang X, Collins JJ (2009a) Diversity-based, model-guided construction of synthetic gene networks with predicted functions. *Nat Biotechnol* **27:** 465-471

Ellis T, Wang X, Collins JJ (2009b) Diversity-based, model-guided construction of synthetic gene networks with predicted functions. *Nature biotechnology* **27:** 465-471

Ferry MS, Razinkov IA, Hasty J (2011) Microfluidics for synthetic biology: from design to execution. *Methods Enzymol* **497:** 295-372

Galdzicki M, Rodriguez C, Chandran D, Sauro HM, Gennari JH (2011) Standard biological parts knowledgebase. *PLoS ONE* **6:** e17005

Gordon A, Colman-Lerner A, Chin TE, Benjamin KR, Yu RC, Brent R (2007) Single-cell quantification of molecules and rates using open-source microscope-based cytometry. *Nature methods* **4:** 175-181

Huh D, Paulsson J (2011) Non-genetic heterogeneity from stochastic partitioning at cell division. *Nature genetics* **43:** 95-100

Jackson C, Glory-Afshar E, Murphy RF, Kovacevic J (2011) Model building and intelligent acquisition with application to protein subcellular location classification. *Bioinformatics* **27:** 1854-1859

Jackson C, Murphy RF, Kovacevic J (2009) Intelligent acquisition and learning of fluorescence microscope data models. *IEEE Transactions on Image Processing (***in press)**

Jaqaman K, Danuser G (2006) Linking data to models: data regression. *Nat Rev Mol Cell Biol* **7:** 813-819

Kar S, Baumann WT, Paul MR, Tyson JJ (2009) Exploring the roles of noise in the eukaryotic cell cycle. *Proceedings of the National Academy of Sciences of the United States of America* **106:** 6471-6476

Kelly JR, Rubin AJ, Davis JH, Ajo-Franklin CM, Cumbers J, Czar MJ, de Mora K, Glieberman AL, Monie DD, Endy D (2009) Measuring the activity of BioBrick promoters using an in vivo reference standard. *J Biol Eng* **3:** 4

Khalil AS, Collins JJ (2010) Synthetic biology: applications come of age. *Nat Rev Genet* **11:** 367-379

Kremers GJ, Goedhart J, van Munster EB, Gadella TW, Jr. (2006) Cyan and yellow super fluorescent proteins with improved brightness, protein folding, and FRET Forster radius. *Biochemistry* **45:** 6570-6580

Kwok R (2010) Five hard truths for synthetic biology. *Nature* **463:** 288-290

Lee MV, Topper SE, Hubler SL, Hose J, Wenger CD, Coon JJ, Gasch AP (2011) A dynamic model of proteome changes reveals new roles for transcript alteration in yeast. *Molecular systems biology* **7:** 514
Litcofsky KD, Afeyan RB, Krom RJ, Khalil AS, Collins JJ (2012) Iterative plug-and-play methodology for constructing and modifying synthetic gene networks. *Nature Methods* **9:** 1077-+

Locke JC, Elowitz MB (2009) Using movies to analyse gene circuit dynamics in single cells. *Nature reviews Microbiology* **7:** 383-392

Moon TS, Lou C, Tamsir A, Stanton BC, Voigt CA (2012a) Genetic programs constructed from layered logic gates in single cells. *Nature* **491:** 249-253

Munsky B, Neuert G, van Oudenaarden A (2012) Using gene expression noise to understand gene regulation. *Science* **336:** 183-187

Nagai T, Ibata K, Park ES, Kubota M, Mikoshiba K, Miyawaki A (2002) A variant of yellow fluorescent protein with fast and efficient maturation for cell-biological applications. *Nature biotechnology* **20:** 87-90

Newman JR, Ghaemmaghami S, Ihmels J, Breslow DK, Noble M, DeRisi JL, Weissman JS (2006) Single-cell proteomic analysis of S. cerevisiae reveals the architecture of biological noise. *Nature* **441:** 840-846

Peccoud J, Blauvelt MF, Cai Y, Cooper KL, Crasta O, DeLalla EC, Evans C, Folkerts O, Lyons BM, Mane SP, Shelton R, Sweede MA, Waldon SA (2008) Targeted Development of Registries of Biological Parts. *PLoS ONE* **3:** e2671

Peccoud J, Ycart B (1995) Markovian Modeling of Gene-Product Synthesis. *Theor Popul Biol* **48:** 222-234

Prindle A, Samayoa P, Razinkov I, Danino T, Tsimring LS, Hasty J (2012) A sensing array of radically coupled genetic 'biopixels'. *Nature* **481:** 39-44

Purnick PEM, Weiss R (2009) The second wave of synthetic biology: from modules to systems. *Nat Rev Mol Cell Biol* **10:** 410-422

Raser JM, O'Shea EK (2004) Control of stochasticity in eukaryotic gene expression. *Science* **304:** 1811-1814

Raue A, Kreutz C, Maiwald T, Bachmann J, Schilling M, Klingmueller U, Timmer J (2009) Structural and practical identifiability analysis of partially observed dynamical models by exploiting the profile likelihood. *Bioinformatics* **25:** 1923-1929

Raue A, Kreutz C, Maiwald T, Klingmueller U, Timmer J (2011) Addressing parameter identifiability by model-based experimentation. *IET systems biology* **5:** 120-U178

Regot S, Macia J, Conde N, Furukawa K, Kjellen J, Peeters T, Hohmann S, de Nadal E, Posas F, Sole R (2011) Distributed biological computation with multicellular engineered networks. *Nature* **469:** 207-211

Salis HM, Mirsky EA, Voigt CA (2009) Automated design of synthetic ribosome binding sites to control protein expression. *Nat Biotechnol* **27:** 946-950

Shahrezaei V, Swain PS (2008) Analytical distributions for stochastic gene expression. *Proc Natl Acad Sci U S A* **105:** 17256-17261

Slusarczyk AL, Lin A, Weiss R (2012) Foundations for the design and implementation of synthetic genetic circuits. *Nat Rev Genet* **13:** 406-420

So LH, Ghosh A, Zong CH, Sepulveda LA, Segev R, Golding I (2011) General properties of transcriptional time series in Escherichia coli. *Nat Genet* **43:** 554-U584

Tabor JJ, Salis HM, Simpson ZB, Chevalier AA, Levskaya A, Marcotte EM, Voigt CA, Ellington AD (2009) A synthetic genetic edge detection program. *Cell* **137:** 1272-1281

Tamsir A, Tabor JJ, Voigt CA (2011) Robust multicellular computing using genetically encoded NOR gates and chemical 'wires'. *Nature* **469:** 212-215

Timson DJ (2007) *Galactose metabolism in Saccharomyces cerevisiae*, Vol. 1: Global Science Books.

Wang B, Kitney RI, Joly N, Buck M (2011) Engineering modular and orthogonal genetic logic gates for robust digital-like synthetic biology. *Nature communications* **2:** 508

Wang X, Errede B, Elston TC (2008) Mathematical analysis and quantification of fluorescent proteins as transcriptional reporters. *Biophys J* **94:** 2017-2026

Yu J, Xiao J, Ren X, Lao K, Xie XS (2006) Probing gene expression in live cells, one protein molecule at a time. *Science* **311:** 1600-1603

Zenklusen D, Larson DR, Singer RH (2008) Single-RNA counting reveals alternative modes of gene expression in yeast. *Nature structural & molecular biology* **15:** 1263-1271

Chapter 6: Conclusions

Synthetic biology remains a fledgling field with exciting potential. Viruses that seek out and selectively kill cancer cells, microbes that regulate insulin levels in diabetic patients, and phages that attack antibiotic resistant bacteria all promise to revolutionize treatment of major diseases. Production of biofuels, pharmaceuticals, plastics, and industrial products from sunlight or waste products promise to reduce costs and alleviate dependencies on depleting oil supplies. Crops that warn farmers of disease at the earliest stages could prevent famine. Organisms that consume pollutants or sequester carbon could solve major environmental issues. Further down the road, one can imagine many things, such as plants that grow into predefined shapes, regulate building temperatures, or produce bioluminescence for zero-energy lighting. And these are only the tip of the iceberg; surely the most powerful applications have yet to be imagined.

All of the tremendous applications of synthetic biology have also lead to a significant overhyping of the field [14]. The way that relatively simple components are assembled into hugely complex networks in both modern electronics and cells makes comparison between electrical engineering and synthetic biology a natural exercise. Many examples exist, including software tools to facilitate design [15], abstraction levels that enable higher level design [16], and the co-design work presented here. Undoubtedly, this reuse of knowledge will continue.

The example of electrical engineering and the profound impact that field has had on everyday life affords synthetic biology a perspective on its own potentially enormous impact that may not have been available to early electrical engineers. As a result, much can learned from how the thorough, methodical way in which the design cycle in electronics has developed. Thus far, progress in synthetic biology has necessarily consisted mostly of proof-of-concept projects achieved primarily through trial and error. Moving forward, the field could gain a great deal by shifting focus away from what genetic circuits can be engineered to do and towards how genetic circuits can be engineered more effectively.

In this sense, perhaps the biggest gap in synthetic biology is the inability to adequately assign function to the basic biological components. With the relatively crude ability to measure molecular components in cells available today, attempts to create models of basic components that can be composed to predict the function of novel circuits are very limited. In the long-term, breakthroughs in measurement technologies will undoubtedly alleviate these problems to a great degree. In the short-term, leveraging all of the information in a dataset, such as the cell-cell variability, will help close the gap. Moreover, by carefully formalizing the ability or inability of a particular dataset to constrain a particular model, researchers should be able to get as much out of their data as possible.

More abstractly, this gap derives from perhaps the most fundamental difference between electronic and genetic circuits, namely that electronic circuits have been engineered, while genetic circuits have evolved. In electronics, engineers have always used well-defined characterizations of the most basic components and their interactions to build increasingly larger systems. In synthetic biology, we only understand the basic components and their interactions in broad strokes. As such, attempts by synthetic biologists to formalize the function of genetic components serves to enhance our understanding of natural genetic systems as well. A quote on the chalkboard of Nobel Prize-winning physicist Richard Feynman at the time of his death read, “What I cannot create, I do not understand.” Such a sentiment points to a powerful and less obvious impact of synthetic biology: that by building life, we will better understand it. It will be exciting to see how synthetic biology impacts the world, both by creating novel solutions to major problems and by shedding new light on the mysteries of how life functions.

References

1. Slusarczyk, A.L., A. Lin, and R. Weiss, *Foundations for the design and implementation of synthetic genetic circuits.* Nat Rev Genet, 2012. **13**(6): p. 406-20.
2. Khalil, A.S. and J.J. Collins, *Synthetic biology: applications come of age.* Nat Rev Genet, 2010. **11**(5): p. 367-79.
3. Ellis, T., X. Wang, and J.J. Collins, *Diversity-based, model-guided construction of synthetic gene networks with predicted functions.* Nat Biotechnol, 2009. **27**(5): p. 465-71.
4. Gardner, T.S., C.R. Cantor, and J.J. Collins, *Construction of a genetic toggle switch in Escherichia coli.* Nature, 2000. **403**(6767): p. 339-42.
5. Elowitz, M.B. and S. Leibler, *A synthetic oscillatory network of transcriptional regulators.* Nature, 2000. **403**(6767): p. 335-8.
6. Stricker, J., et al., *A fast, robust and tunable synthetic gene oscillator.* Nature, 2008. **456**(7221): p. 516-9.
7. Danino, T., et al., *A synchronized quorum of genetic clocks.* Nature, 2010. **463**(7279): p. 326-30.
8. Prindle, A., et al., *A sensing array of radically coupled genetic 'biopixels'.* Nature, 2012. **481**(7379): p. 39-44.
9. Regot, S., et al., *Distributed biological computation with multicellular engineered networks.* Nature, 2011. **469**(7329): p. 207-11.
10. Anderson, J.C., C.A. Voigt, and A.P. Arkin, *Environmental signal integration by a modular AND gate.* Mol Syst Biol, 2007. **3**: p. 133.
11. Tamsir, A., J.J. Tabor, and C.A. Voigt, *Robust multicellular computing using genetically encoded NOR gates and chemical 'wires'.* Nature, 2011. **469**(7329): p. 212-5.
12. Moon, T.S., et al., *Genetic programs constructed from layered logic gates in single cells.* Nature, 2012. **491**(7423): p. 249-53.
13. Purnick, P.E. and R. Weiss, *The second wave of synthetic biology: from modules to systems.* Nat Rev Mol Cell Biol, 2009. **10**(6): p. 410-22.
14. Kwok, R., *Five hard truths for synthetic biology.* Nature, 2010. **463**(7279): p. 288-90.
15. MacDonald, J.T., et al., *Computational design approaches and tools for synthetic biology.* Integr Biol (Camb), 2011. **3**(2): p. 97-108.
16. Clancy, K. and C.A. Voigt, *Programming cells: towards an automated 'Genetic Compiler'.* Curr Opin Biotechnol, 2010. **21**(4): p. 572-81.

Appendix A: Supplementary Information for Chapter 4

No.	Semantic Actions	Dependence
9	promoter1.name = [pro_u]	N/A
10	rbs1.name = [rbsA]	N/A
11	gene1.name = [v]	N/A
12	terminator1.name = [t1]	N/A
13	promoter2.name = [pro_v]	N/A
14	rbs2.name = [rbsB]	N/A
15	gene2.name = [u]	N/A
16	terminator2.name = [t1]	N/A

4	cistron1.transcript = rbs1.name + gene1.name = [rbsA_v]
	cistron1.equation_list = translation(rbsA, v)
		10, 11

8	cistron2.transcript = rbs2.name + gene2.name = [rbsB_u]
	cistron2.equation_list = translation(rbsB, u)
		14, 15

7	restConstruct2.equation_list = [ ]	N/A
2	cassette1.equation_list = cistron1.equation_list + transcription(pro_u, cistron1.transcript) = translation(rbsA, v) + transcription(pro_u, rbsA_v) + promoter_protein_interaction([pro_u, rbsA_v], cassette1.protein_list)	4, 9, 12
6	cassette2.equation_list = cistron2.equation_list + transcription(pro_u, cistron2.transcru) + transcription(pro_v,	8, 13, 16

	rbsB_u) + promoter_protein_interaction([pro_v, rbsB_u], cassette2.protein_list)	
5	construct2.equation_list = cassette2.equation_list + restConstructs2.equation_list = translation(rbsB, u) + transcription(pro_v, rbsB_u) + promoter_protein_interaction([pro_v, rbsB_u], cassette2.protein_list)	6, 7
3	restConstructs1.equations_list = construct2.equation_list = translation(rbsB, u) + transcription(pro_v, rbsB_u) + promoter_protein_interaction([pro_v, rbsB_u], cassette2.protein_list)	5

1	cassette1.protein_list = constructs1.protein_list = [protein_u, protein_v]
	cassette2.protein_list = constructs1.protein_list = [protein_u, protein_v]
	constructs1.equation_list = cassette1.equation_list + restConstructs.equation_list = transcription(pro_u, rbsA_v) + translation(rbsA, v) + transcription(pro_v, rbsB_u) + translation(rbsB, u)+ promoter_protein_interaction([pro_u, rbsA_v], [protein_u, protein_v]) + promoter_protein_interaction([pro_v, rbsB_u], [protein_u, protein_v])
		2, 3

Table A.1: Computation dependence corresponding to the derivation tree in Figure 4.2. The computation starts from the leaves of the tree, and the semantic values computed are transferred to upstream nodes. The computation of each node cannot proceed until all of its sub-trees are computed. For example, the computation of semantic values of <constructs1> (2) is pending until its subtrees<cassette1> (3) and <restConstructs1> (4) are computed.

Part Name	Part Type	Associated Parameter	Parameter Value
ptrc2	Promoter	promoter.name	ptrc2
		promoter.transcription_rate	25
		promoter.leakiness_rate	.25
		promoter.repressor_list	[[lacI, 4, 0.001, 1], [lacIrc, 4, 0.001, 1]]
pls1con	Promoter	promoter.name	pls1con
		promoter.transcription_rate	50
		promoter.leakiness_rate	0.00833333
		promoter.repressor_list	[[cIts, 2, 0. 1, 1], [cItsrc, 2, 0.1, 1]]
pltet01	Promoter	promoter.name	tetR
		promoter.transcription_rate	10
		promoter.leakiness_rate	0.1
		promoter.repressor_list	[[tetR, 2, 0.1, 1], [tetRrc, 2, 0. 1, 1]]
ptrc2rc	Reverse Promoter	promoter.name	ptrc2rc
		promoter.transcription_rate	25
		promoter.leakiness_rate	.25
		promoter.repressor_list	[[lacI, 4, 0.001, 1], [lacIrc, 4, 0.001, 1]]
pls1conrc	Reverse Promoter	promoter.name	pls1conrc
		promoter.transcription_rate	50
		promoter.leakiness_rate	0.00833333
		promoter.repressor_list	[[cIts, 2, 0. 1, 1], [cItsrc, 2, 0.1, 1]]
pltet01rc	Reverse Promoter	promoter.name	tetRrc
		promoter.transcription_rate	10
		promoter.leakiness_rate	0.1
		promoter.repressor_list	[[tetR, 2, 0.1, 1], [tetRrc, 2, 0. 1, 1]]
rbsA	RBS	rbs.name	rbsA
		rbs.translation_rate	25
rbsB	RBS	rbs.name	rbsB
		rbs.translation_rate	50
rbsC	RBS	rbs.name	rbsC
		rbs.translation_rate	10
rbsD	RBS	rbs.name	rbsD
		rbs.translation_rate	12.5
rbsE	RBS	rbs.name	rbsE
		rbs.translation_rate	6.25
rbsF	RBS	rbs.name	rbsF
		rbs.translation_rate	7
rbsG	RBS	rbs.name	rbsG
		rbs.translation_rate	5

rbsH	RBS	rbs.name	rbsH
		rbs.translation_rate	2
rbsArc	Reverse RBS	rbs.name	rbsA
		rbs.translation_rate	25
rbsBrc	Reverse RBS	rbs.name	rbsB
		rbs.translation_rate	50
rbsCrc	Reverse RBS	rbs.name	rbsC
		rbs.translation_rate	10
rbsDrc	Reverse RBS	rbs.name	rbsD
		rbs.translation_rate	12.5
rbsErc	Reverse RBS	rbs.name	rbsE
		rbs.translation_rate	6.25
rbsFrc	Reverse RBS	rbs.name	rbsF
		rbs.translation_rate	7
rbsGrc	Reverse RBS	rbs.name	rbsG
		rbs.translation_rate	5
rbsHrc	Reverse RBS	rbs.name	rbsH
		rbs.translation_rate	2
lacI	Gene	gene.name	lacI
		gene.mRNA_degradation_rate	1
		gene.protein_degradation_rate	0.1
gfpmut3	Gene	gene.name	gfpmut3
		gene.mRNA_degradation_rate	1
		gene.protein_degradation_rate	0.1
cIts	Gene	gene.name	cIts
		gene.mRNA_degradation_rate	1
		gene.protein_degradation_rate	0.1
tetR	Gene	gene.name	tetR
		gene.mRNA_degradation_rate	1
		gene.protein_degradation_rate	0.1
lacIrc	Reverse Gene	gene.name	lacIrc
		gene.mRNA_degradation_rate	1
		gene.protein_degradation_rate	0.1
gfpmut3rc	Reverse Gene	gene.name	gfpmut3rc
		gene.mRNA_degradation_rate	1
		gene.protein_degradation_rate	0.1
cItsrc	Reverse Gene	gene.name	cItsrc
		gene.mRNA_degradation_rate	1
		gene.protein_degradation_rate	0.1
tetRrc	Reverse Gene	gene.name	tetRrc
		gene.mRNA_degradation_rate	1
		gene.protein_degradation_rate	0.1
b0010	Terminator	none	
b0012	Terminator	none	

b0016	Terminator	none	
b0010rc	Reverse Terminator	none	
b0012rc	Reverse Terminator	none	
b0016rc	Reverse Terminator	none	

Table A.2: List of parts used in the "exploration of genetic space" section and values of associated attributes

Appendix B: Supplementary Information for Chapter 5

B.1 Model details

B.1.1 List of model quantities and equations

Table B.1 and Table B.2 list the model species and equations used in the model.

Species	Symbol
Inactive Promoter	Px
Active Promoter	P
mRNA	M
Unfolded FP Copy	FPu
Folded FP Copy	FPm

Table B.1: List of model species and symbols.

Function	Equation	Parameter(s)	Units
Activation	$Px \rightarrow P$	*kACT*	[min^{-1}]
Transcription	$P \rightarrow P + B * M$	*kTX, B*	[min^{-1}], []
mRNA Degradation	$M \rightarrow 0$	*kMDeg*	[min^{-1}]
Translation	$M \rightarrow M + FPu$	*kTL*	[min^{-1}]
Maturation	$FPu \rightarrow FPm$	*kMAT*	[min^{-1}]
AFU Scaling	$AFU = AFUperFP * FPm$	*A*	[AFU · molecule^{-1}]

Table B.2: List of equations and corresponding parameters.

B.1.2 Modeling cell division and cell-partitioning noise

We initially used the stochastic model without modeling cell division. This model allowed us to obtain good fits to the maturation data, but the cell-cell variability in the induction experiments was consistently too small (Figure B.1a-c). We were unable to match it by changing parameters, at least not without severely decreasing our fits to the maturation data. We hypothesized that the increased variability stemmed from cell-partitioning errors, which are known to be significant[1, 2]. To model cell division, we add a mass-action equation, *cell1 -> cell1 + cell* with rate *kCell*. *Cell1* always has a value of 1, while *cell2* progresses from 0 to 100, corresponding to percentage completion of a cell cycle. We choose *kCell* such that the mean division of simulated cells time matches the experimental growth rate (for a discussion of the estimation of experimental growth rates, see below). Whenever the value of *cell2* is 100, an event is triggered to emulate cell division. The model species of mRNA, FPu, and FPm are partitioned by random selection from a binomial distribution, where N_{mother} = B($N_{pre\text{-}division}$, p=.6) and $N_{daughter} = N_{pre\text{-}division} - N_{mother}$. The choice of p=.6 reflects that size at division is typically split 60-40% between mothers and daughters, respectively. The promoter state (P=0 for inactive, P=1 for active) is conserved between mother and daughter, i.e. once a promoter becomes active, it stays active and passes that state to any daughter cells. Finally, *cell2* is reset to 0 and the cell continues with a new cell cycle. Each simulated cell does not simultaneously simulate any daughters; rather, the times of division and inherited species quantities are stored for subsequent simulation. The simulation ends after the duration of the current experimental dataset of interest has been reached. After a cell finishes, the next unsimulated daughter is simulated between its time of birth and the end of the experiment, and any new daughters are added to the list. A while loop ensures that all cells are simulated. Simulating a population begins with a number of cells equal to the number of cells observed at the beginning of the current experimental dataset. Incorporation of cell division immediately rectified the issue of insufficient noise in the non-inhibition experiments, with no adjustment of the parameters used to model cell division (Figure B.1d-e). The cell division does have some impact on the mean apparent AFUs, not just the cell-cell variability, presumably due to apparent degradation of species due to dilution.

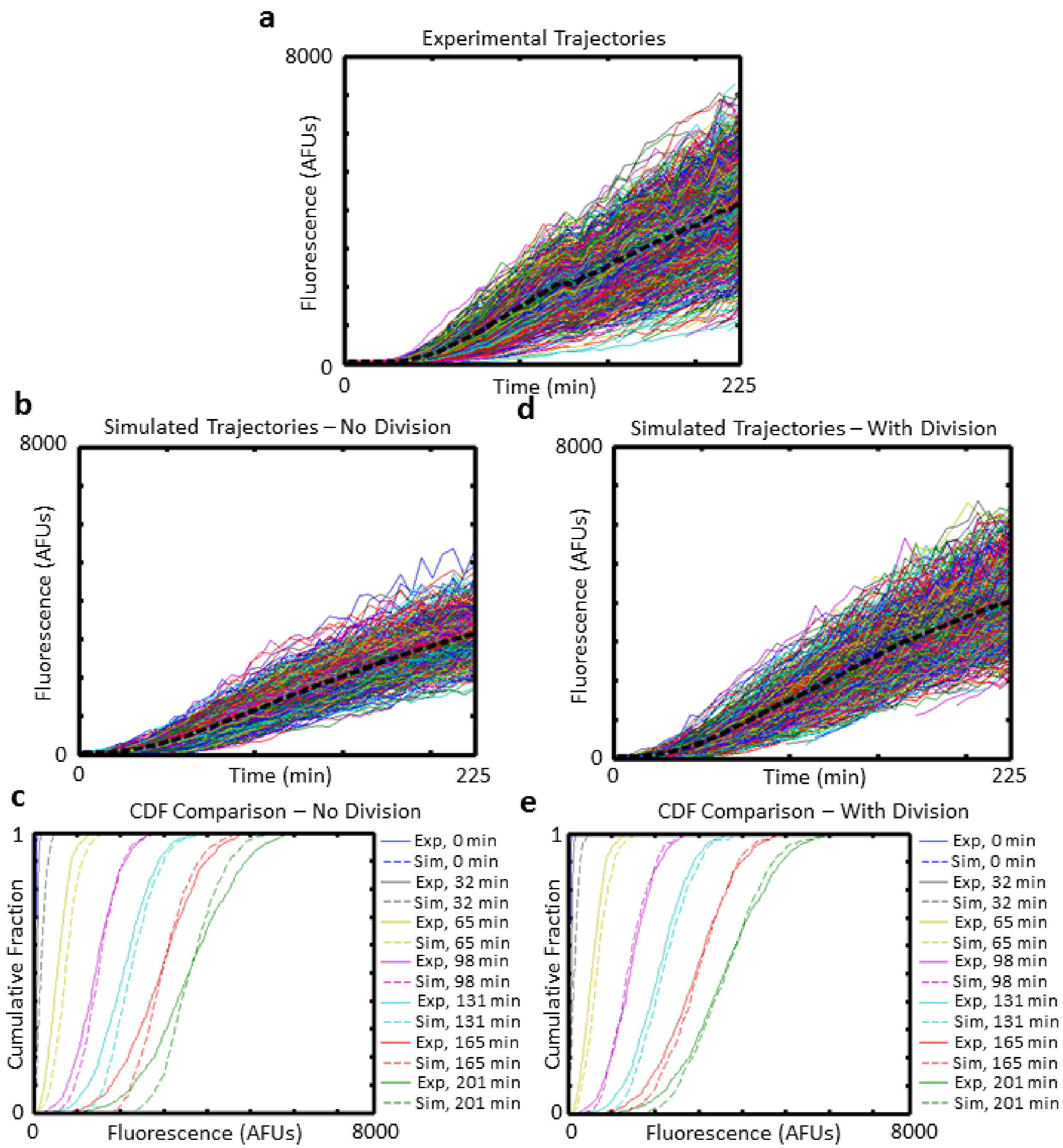

Figure B.1: Division noise.
The observed cell-cell variability in experimental data (a) is larger than predicted by the mass-action model alone (b). This difference contributes to the objective function in the form of CDF's of simulated cells that are too steep (d, only a subset of time points shown for presentation clarity). Modeling cell division increases cell-cell variability (c) to levels similar to those observed, which is reflected in the objective function by CDF's of simulated cells that are more similar to experimental cells (e).

B.2 Experimental Considerations

B.2.1 Experimental replicates

We performed 3 experimental replicates of each experiment (inhibition and induction), each giving similar results. Figure B.2 shows the datasets (inhibition: a-c, induction: d-f). Figure B.3 shows a comparison of the mean and standard deviations of the replicates.

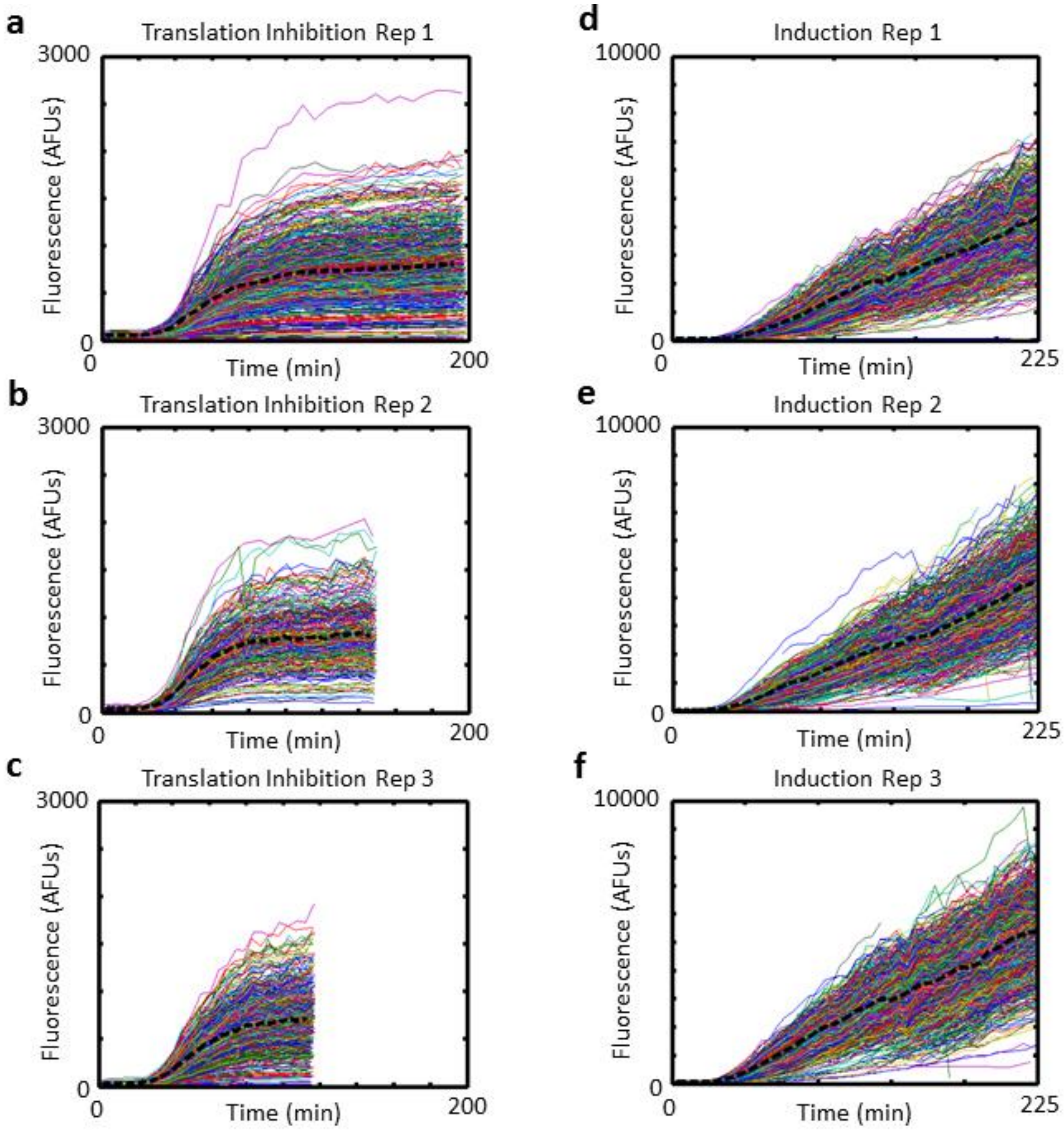

Figure B.2: Experimental replicates.
Three experimental replicates of the translation inhibition experiment (a-c) and induction experiment (e-g) are shown. Translation inhibition experiments have different end times corresponding to when the fluorescence values reached steady state.

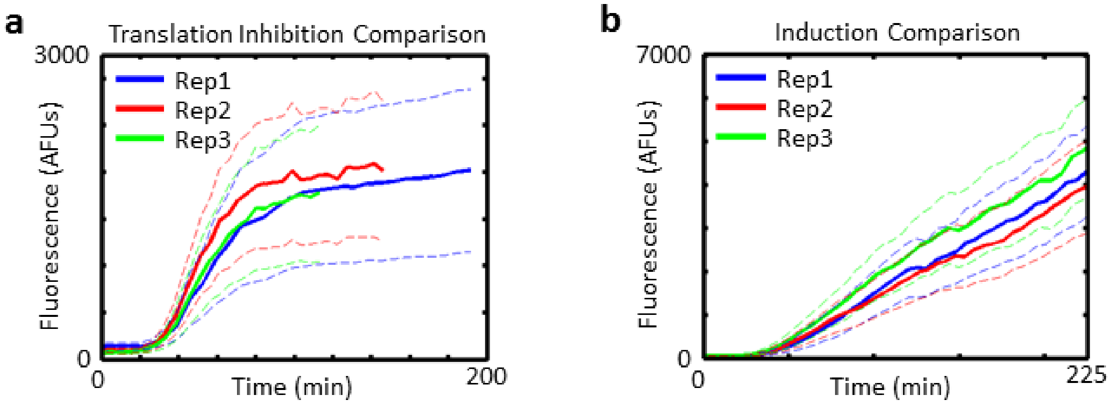

Figure B.3: Comparison of experimental replicates.
Comparison of the mean (solid lines) plus or minus one standard deviation (dashed lines) for each experimental replicate (colors; see legend) are compared (inhibition: a, induction: b).

B.2.2 Growth rates and cell counts

In order to validate that our cells are growing normally during our experiments, we fit a growth curve to the cell count over time as collected by GenoSIGHT (Figure B.4). We only considered growth rates after induction, which excludes the time that we allow cells to adjust to the new environment. We note that the estimated doubling times of ~160 mins are reasonable for growth in synthetic media with a non-preferred carbon source, galactose. For the first few timepoints after induction, the estimated growth rate does not match the observed data. This result is expected because the cells are adjusting the change of carbon source upon introduction of galactose, the inducer molecule for the system. Experimental replicates of the induction experiment give similar growth rate fits (Table B.3). We do not estimate growth rates for the inhibition experiment because the inhibitor is added before the cells have time to adjust to the carbon source change. Cell counts are consistent across experimental replicates, with one exception where a smaller number of fields of view were analyzed. Cell counts are higher for non-inhibition experiments because the cells continue to grow throughout, unlike in the inhibition experiments.

Experiment	Cell Count			Doubling Time (min)		
	Rep1	Rep2	Rep3	Rep1	Rep2	Rep3
Translation Inhibition	740	300*	728	n/a	n/a	n/a
Induction	1000	924	1281	163	167	158

Table B.3: Growth rates for experimental replicates.
**Inhibition Rep 2 used fewer fields of view, resulting in fewer cells.*

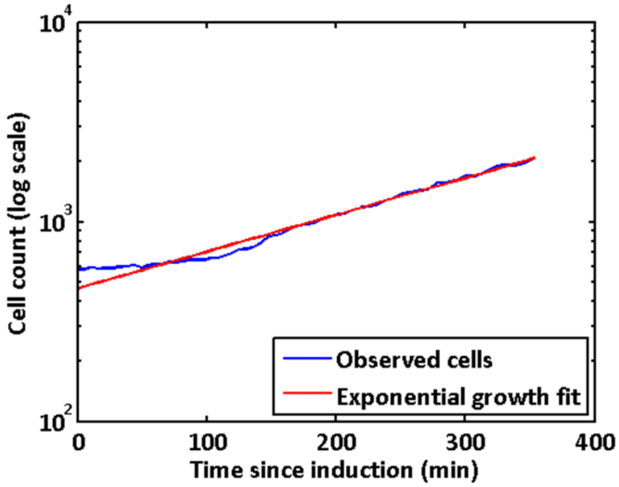

Figure B.4: Fit of growth curve to experimental cell counts.
Time zero represents the time that the inducer (galactose) is added during a representative non-inducing experiment. After an expected period of adjustment to the new carbon source (raffinose to galactose), the cells grow exponentially.

B.2.3 FISH Probe Sequences

Probes for Venus FISH experiments were designed as described in Online Methods. Table B.4 lists the sequences of each probe used.

Probe	Sequence
Probe 1	CCG-T-ATGTTGCA-T-CACCTTCACCC-T-CTCCAC-T-GACAGAAAA-T-TTGTGCCC
Probe 2	G-T-AGTGACAAG-T-GTTGGCCA-T-GGAACAGGTAG-T-TTTCCAGTAG-T-GC
Probe 3	GGG-T-ATCACCT-T-CAAACTTGACT-T-CAGCACGTGTCT-T-GTAGTTCCCG-T-C
Probe 4	AAAGGGCAGA-T-TGTGTGGACAGG-T-AATGGTTGTC-T-GGTAAAAGGACAGGGCC
Probe 5	CCCAGCAGC-T-GTTACAAAC-T-CAAGAAGGACCA-T-GTGGTCTCTC-T-TTTCGTTGGG

Table B.4: FISH Probe Sequences.
The 5 probes used for FISH analysis of Venus mRNA are listed. T's delineated by dashes represent bases that contained an amine group for fluorescent die labeling (see Online Methods).

B.3 Simulations

B.3.1 Matching simulated data sets to experimental replicates

We match our model to a given data set by hand in a systematic way. Starting with biologically reasonable parameter values, we manually adjust all parameters to obtain a decent first-cut match, and then tune individual parameters to key features. First, in the translation inhibition experiment there are some cells that never exhibit an increase in fluorescence, presumably because they did not express any unmature FP copies before inhibition. Noting this observation, we tune *kACT* such that the fraction of cells that never exhibit increased fluorescence matches between simulations and experimental data. Second, the sole determining factor in how the distributions evolve after translation inhibition is *kMAT*, allowing it to be determined in isolation by matching the transition of the cdf's from translation inhibition until steady state. Third, changes to *kMDEG* impact inhibition and induction experiments differently, which can be leveraged to hone in on an appropriate value. We start with a parameter set that matches the inhibition data well, then simulate the induction experiment with the same parameter set. We then adjust *kMDEG* up or down and tune *kTX*, *kTL*, and *A* to again match the translation inhibition data. Finally, we check how this new parameter set matches the induction data set and iterate the process until simulations match both data sets well. For the remaining parameters (*kTX*, *kTL*, and *A*), we found that multiple combinations can be found by hand with indistinguishable fits. See the main text and below for discussions of these parameters.

We repeated this process for each of the 3 experimental replicates of the translation inhibition experiment. Since the induction experiment replicates are in no way paired to the inhibition replicates, we chose to simply use what appeared to be the most representative of the 3 induction replicates (Rep 1, Figure B.2e; see Figure B.3b for a comparison of the 3 replicates). We use this individual replicate to choose *kMDEG* for each of the 3 translation inhibition replicates. Our simulations match the experimental data well for both the translation inhibition (Figure B.5a-c vs. Figure B.2a-c; Figure B.3a) and induction (Figure B.5d-f vs. Figure B.2d-f; Figure B.3b) experiments.

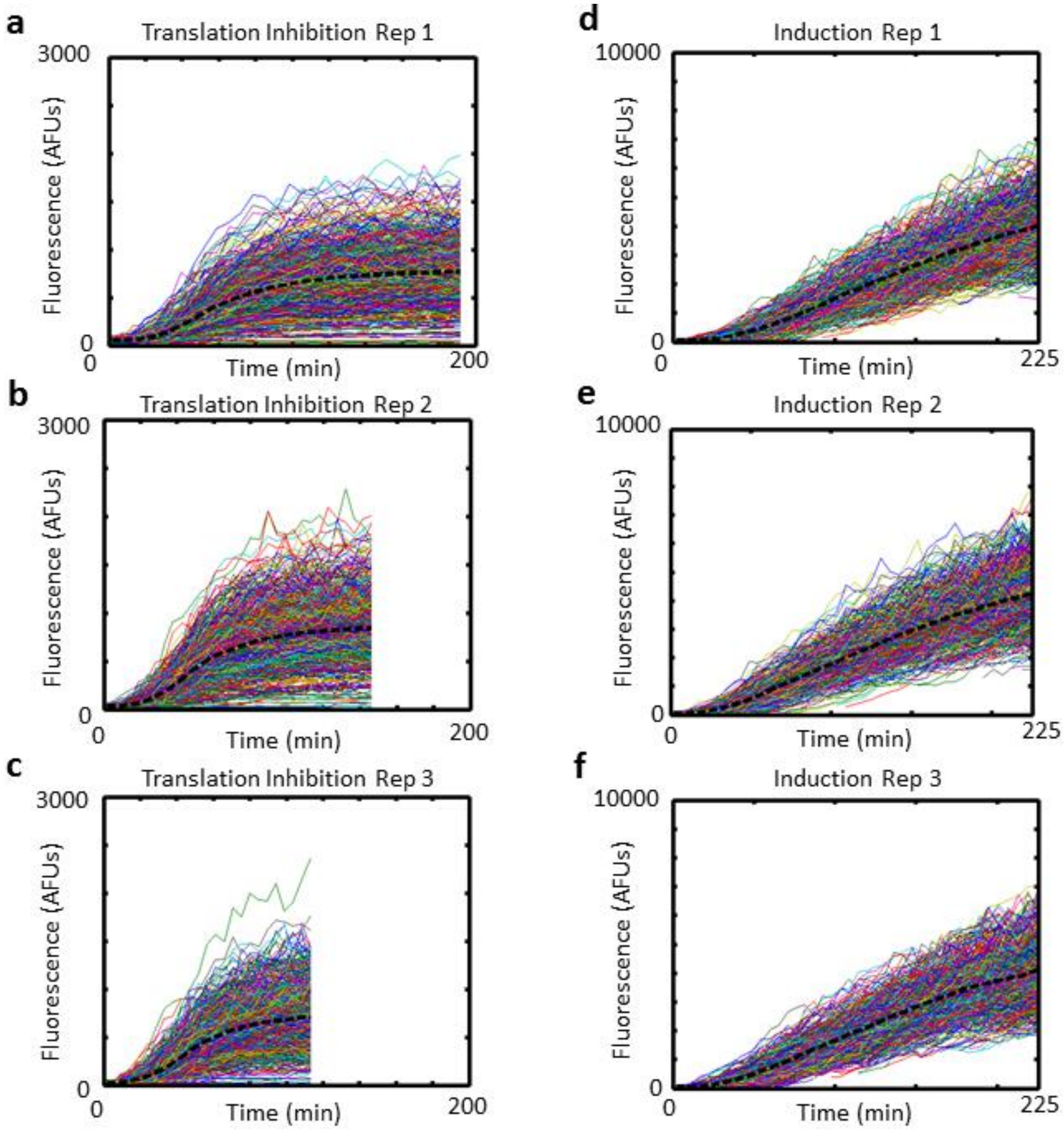

Figure B.5: Simulations for each replicate.
We tune parameters so that simulations match replicates of the translation inhibition experiments (a-c). Since kMDEG is tuned based on induction data, we use a single representative experimental replicate of the induction experiment (Rep 1, see Figure B.3) to match each parameter set from the translation inhibition replicates. Inhibition simulations matching each experimental replicate are shown (a-c). Induction simulations using parameters consistent with each inhibition experimental replicates are also shown (d-f; parameters consistent horizontally across panels, i.e. pairwise: a&d, b&e, and c&f).

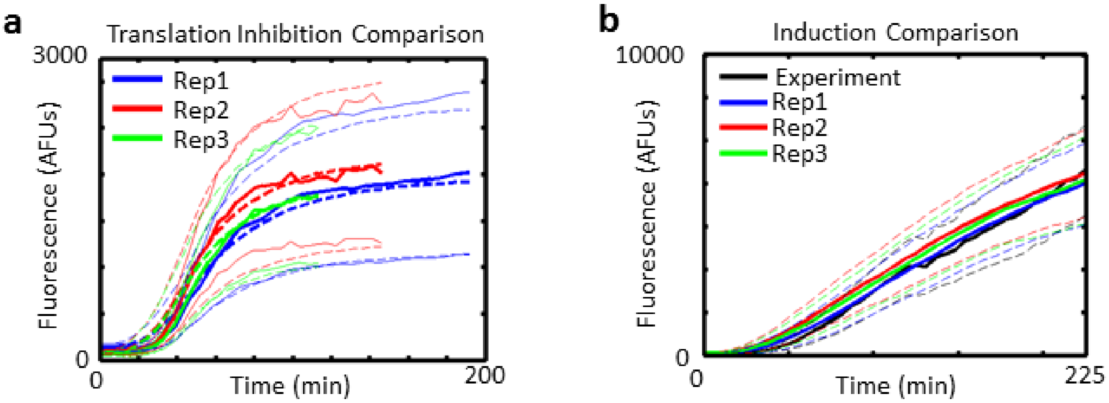

Figure B.6: Comparison of simulated and experimental datasets.
In each case, the translation inhibition simulations match the experimental data well (a – thick lines represent the mean, thin lines are plus or minus 1 standard deviation; solid lines are experimental, dashed lines are simulated). These parameters also mimic the chosen representative induction datasets (b – solid lines are mean, dashed are plus or minus 1 standard deviation; black is experimental, colors are simulations from parameter sets fit to inhibition datasets).

B.3.2 Sensitivity to individual parameters

We performed sensitivity analyses for each of the parameter sets fit to the experimental replicates. The plots shown here are for the parameter sets obtained prior to FISH experiments (Table B.5). It is important to note that the two experimental setups exhibit different sensitivities to each parameter. Since we want to consider the overall sensitivity, in the main text we only presented the average of the fold change in objective function value as the measure of sensitivity. Here we present the sensitivity of each objective function value individually and combined, and for parameter sets corresponding to each of the 3 inhibition replicates (Figure B.7).

Note that in each case, the sensitivity plot for *kACT* is not at its minimum at the no-perturbation point for the induction experiments. We chose this value as described above (briefly, we use the fraction of cells that never “turn on” before inhibition in addition to the CDF’s). For this parameter in the inhibition experiment there is a tradeoff between accurately matching the fraction of cells that never fluoresce and adequately modeling the delay between induction and observed fluorescence. Since the inhibition experiment contains more information in this case, we ignore the location of the minimum suggested by the induction experiment. Moreover, since the induction experiment is not very sensitive to the value of *kACT*, the impact on the average sensitivity is small, though present, especially in Rep 2.

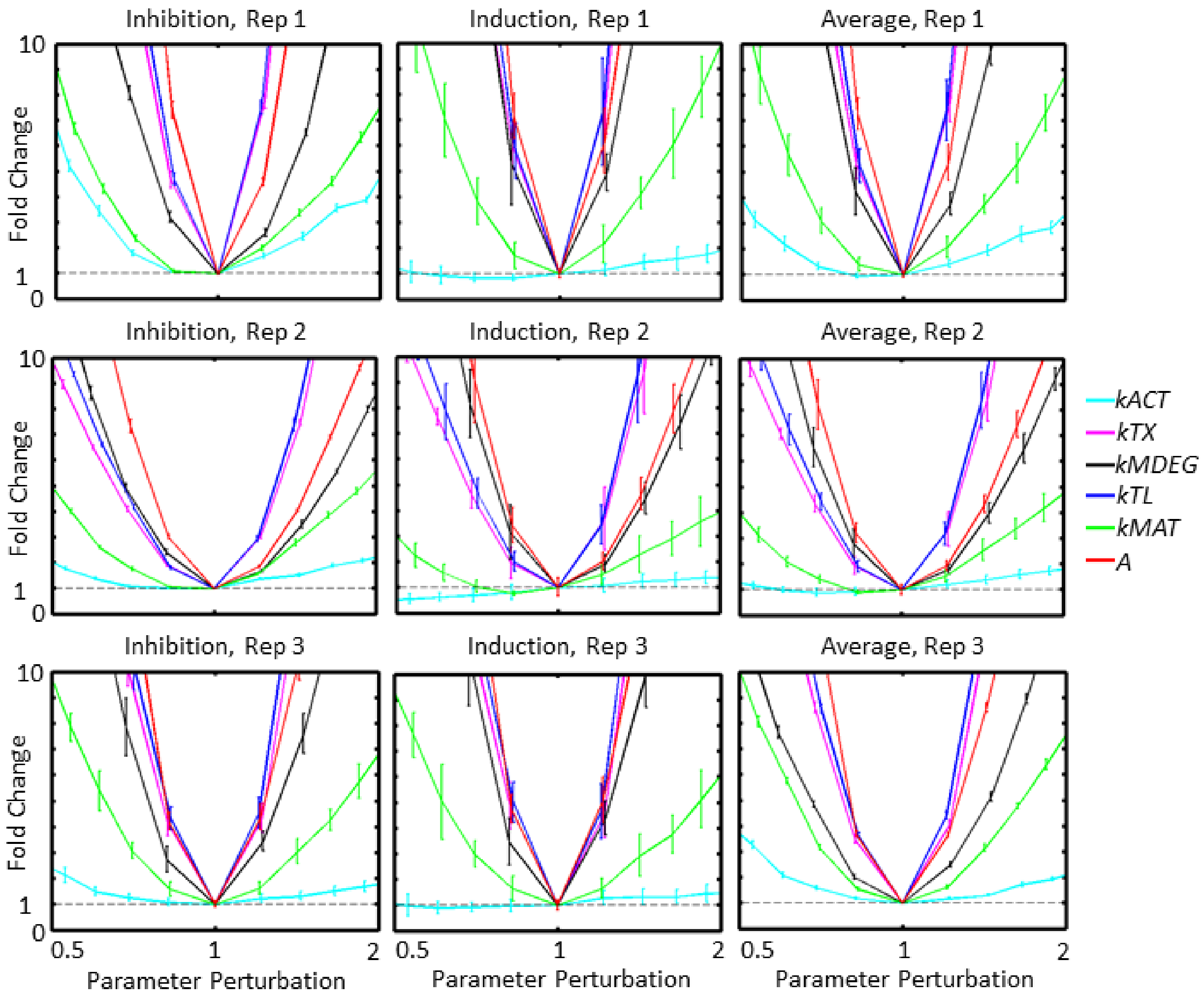

Figure B.7: Sensitivity analyses.

We computed the sensitivity of the objective function to parameter perturbations for inhibition simulations, induction simulations, and combined for parameter sets fit to each of 3 inhibition experiment replicate.

Parameter	Unit	Function	Rep 1	Rep 2	Rep 3
kACT	[min^{-1}]	Activation	0.12	0.14	0.15
kTX	[min^{-1}]	Transcription	0.71	0.73	0.69
B	[]	Burst Size	1	1	1
kMDeg	[min^{-1}]	mRNA Decay	0.087	0.23	0.099
kTL	[min^{-1}]	Translation	0.11	0.27	0.10
kMAT	[min^{-1}]	Maturation	0.025	0.030	0.027
A	[AFU · molecule^{-1}]	Conversion to AFUs	35	33	42

Table B.5: Parameter values used for sensitivity analysis.

B.3.3 Simulation uncertainty

Here, we demonstrate the concept that increasing *n* decreases the simulation uncertainty. Figure B.8a (solid lines) shows that the magnitude of the perturbation size that can be statistically distinguished decreases with increasing *n* for each parameter. The simulation uncertainties start or quickly fall below the normalized standard error for most parameters (Figure B.6a – dashed lines). For *kACT* and *kMAT*, the larger required *n* is unsurprising as the system is much less sensitive to these parameters (see Figure B.7). Figure B.8b shows that increasing *n* constrains *kTX* and *kTL* to a tighter region, but that the region still only has a lower bound on *kTX* and no bounds on *kTL* (see Figure 5.2c and corresponding discussion in Chapter 5).

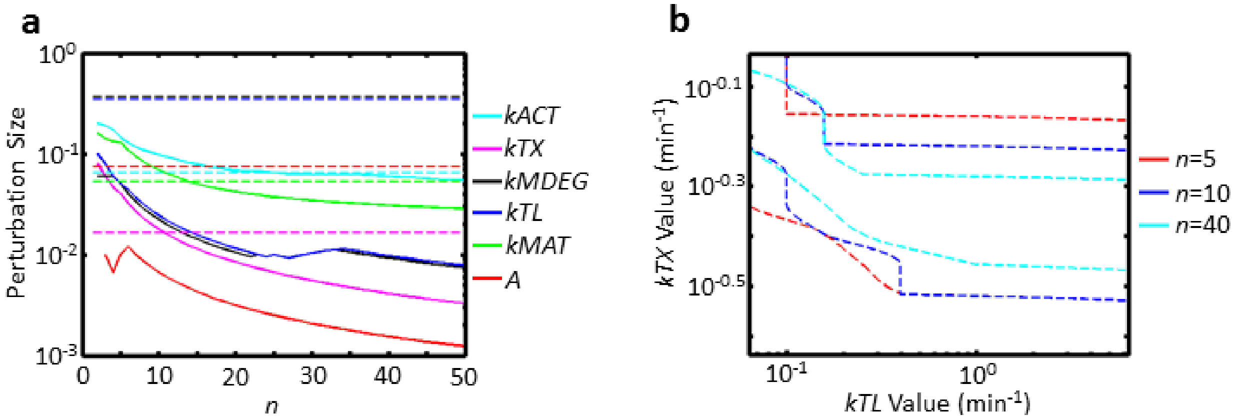

Figure B.8: Simulation uncertainty scales with n.
Larger n decreases the magnitude of the perturbation size that can be distinguished by t-test (a – solid lines), which can be compared to the relative size of the standard error for 3 experimental replicates (a – dashed lines). The size of the region of kTX/kTL pairs that maintain constant P also shrinks with increasing n (b).

B.3.4 Sensitivity analysis of lumped parameters

We investigated the sensitivity of the objective function to perturbations in *kTX*, *kTL*, and *A* with and without *P=kTX×kTL×A* fixed. Figure 5.2b-c in Chapter 5 shows the results for Rep 1. Here we show the corresponding results for each replicate broken down into inhibition, induction, and average conditions (Figure B.9). These results seem to indicate a structural non-identifiability in our system; however, different values of the parameters can produce different distribution shapes, which have a subtle impact on the objective function (see Main Text for more detail). We demonstrate this concept by simulating a population with very large perturbations in *kTL* and *A* that maintain *P* constant (Figure B.10). For smaller perturbations, the impact on the objective function becomes small compared to the noise and requires large values of *n* to detect the impact of small perturbations (see Figure B.8b for an example).

We also considered whether or not the structural non-identifiability observed in Figure B.9 and Figure B.11 were the result of the comparatively slow time-scale maturation process obscuring the details of the transcription and translation steps. We generated a simulated data set with the parameters found for Rep 1 but with the maturation rate very fast (*kMAT*=.001). Repeating the sensitivity analysis, we found that the same picture emerged, indicating that the non-identiability is not caused by slow maturation alone (Fig. A.12).

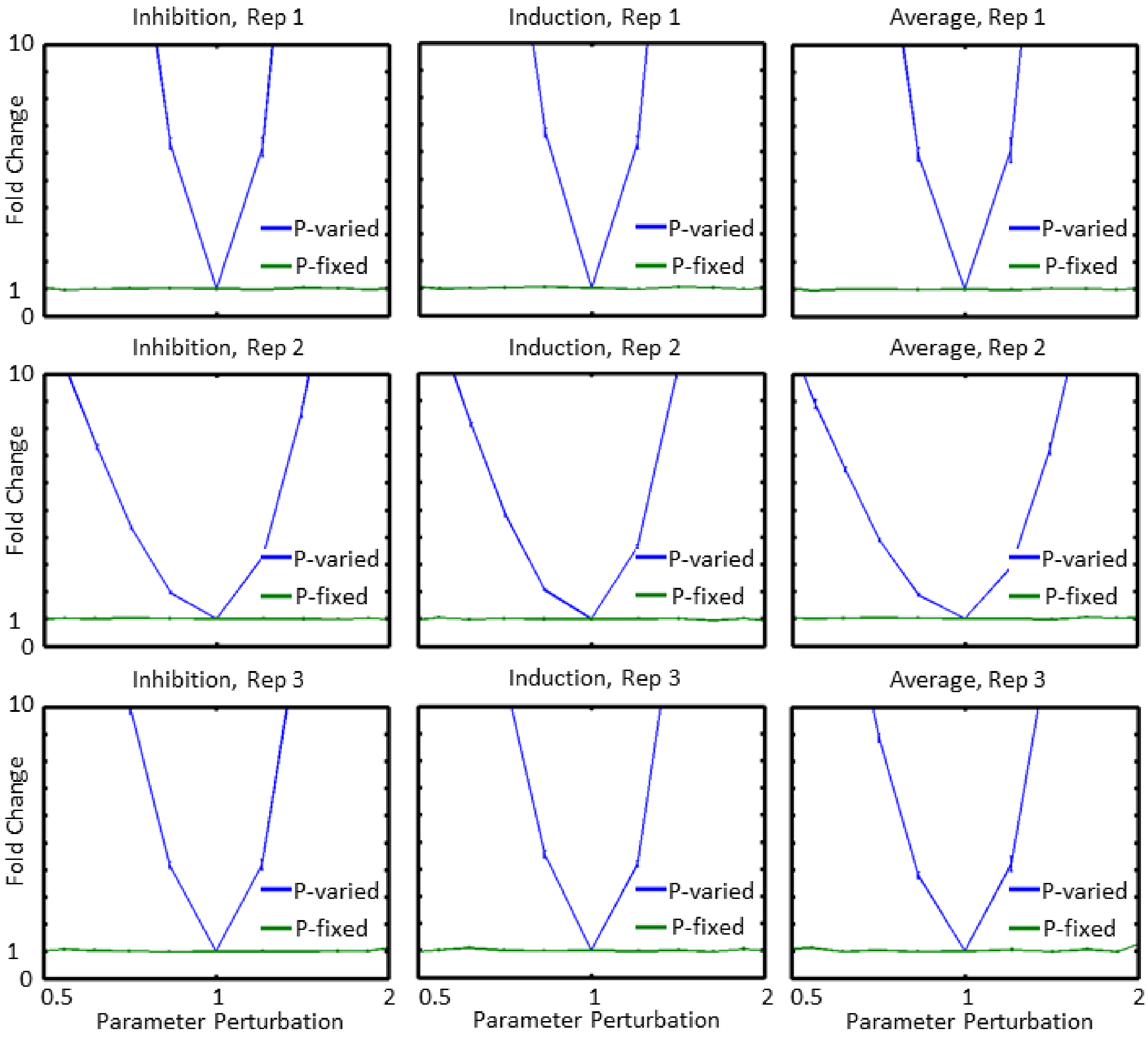

Figure B.9: Sensitivity to lumped parameters with P-fixed and P-varied.
Sensitivity of the inhibition, induction, and combined objective functions to perturbations in kTX, kTL, and A with and without P=kTX×kTL×A fixed for parameters fit to each experimental replicate.

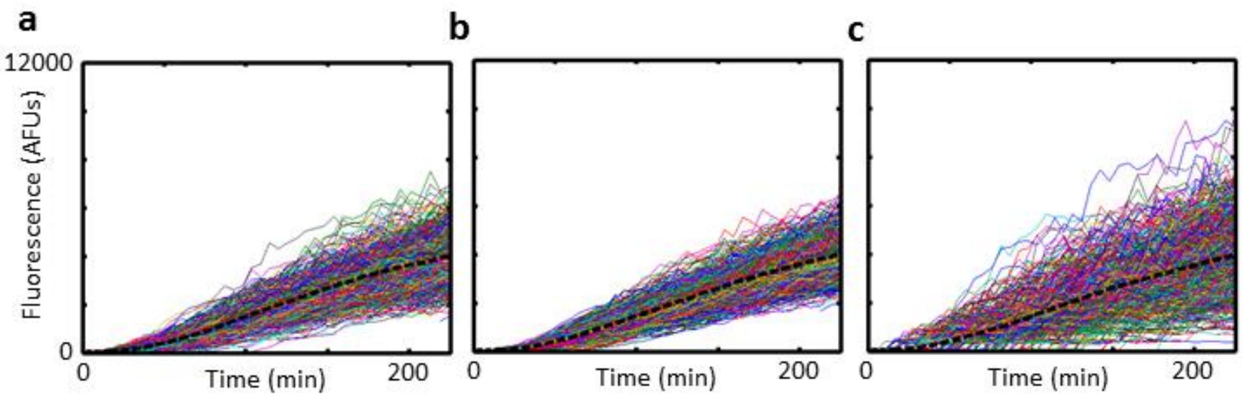

Figure B.10: Cell-cell variability changes even with P-fixed.
*Simulations of a population under induction conditions with related parameter sets (a – no perturbations; b –kTL*16, A/16; c – kTL/8, A*8). These perturbations are very large so that the differences can be observed visually.*

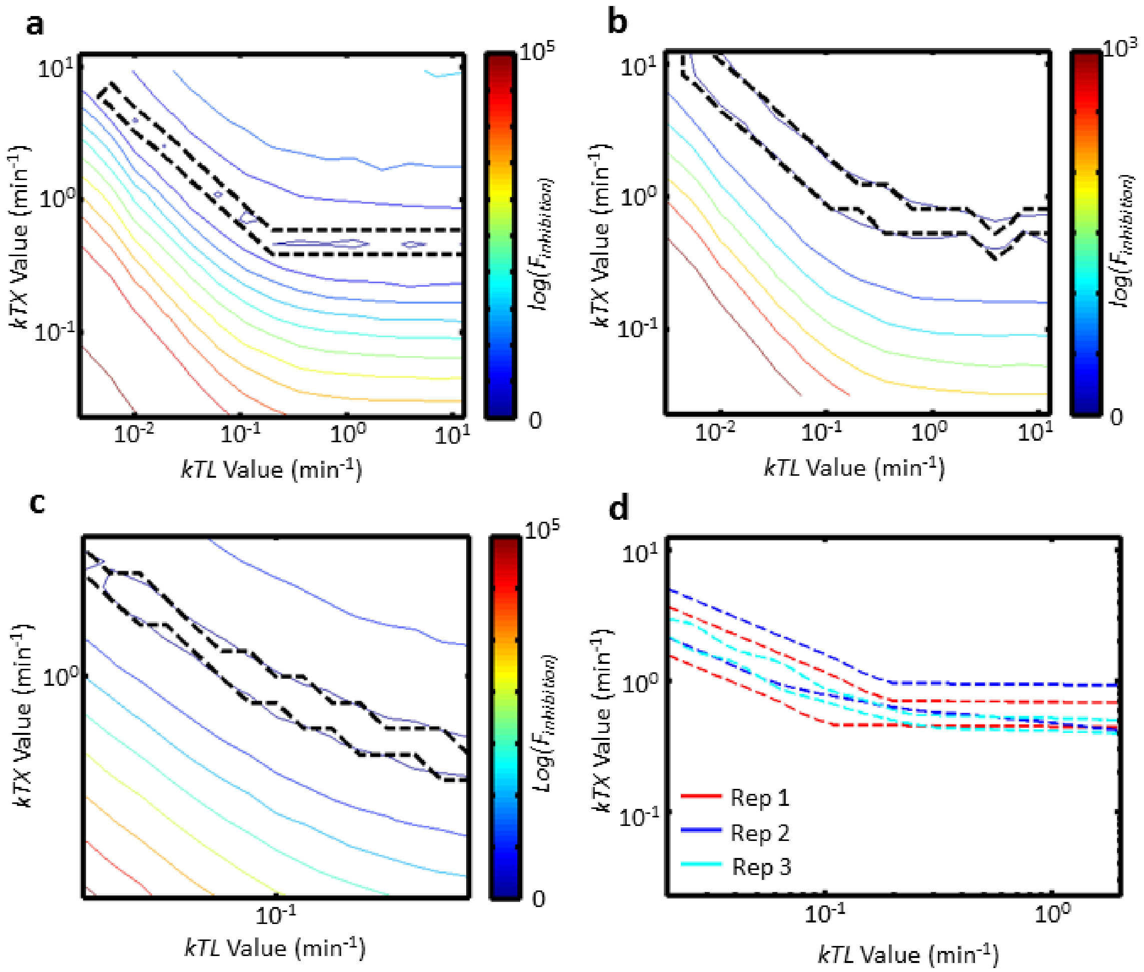

Figure B.11: Sensitivity to kTX and kTL with P fixed.
Parameter sets for each experimental replicate (a-c for Rep 1-3, respectively) result in highly similar regions where different kTX/kTL pairs are indistinguishable by t-test (n=30). These regions are overlapping (d).

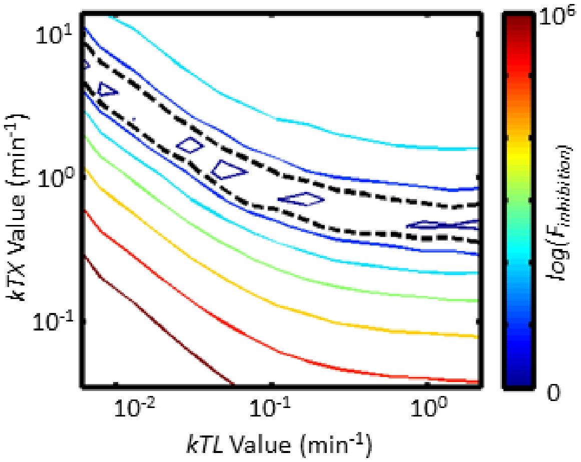

Figure B.12: kTX, kTL, and A are non-identifiable even when maturation dynamics are fast.

Using a simulated dataset with fast maturation dynamics, there is still a locus where different parameter values are indistinguishable by t-test (n=30).

B.3.5 Sensitivity analysis for FISH data

In the main text, we show the ability of the model to fit the FISH data set uniquely given values for *kACT* and *kMDEG* determined from the fluorescence data; however, we only display the results for Rep 1 (Figure 5.3b, Chapter 5). Here we show the corresponding plots for all reps and the overlapping regions (Figure B.13). Similarly, Figure 5.3c in Chapter 5 only depicts results for Rep 1 of the identifiability of *kTL/A* pairs given the transcriptional parameters determined from FISH. Figure B.14 depicts all 3 reps overlaid.

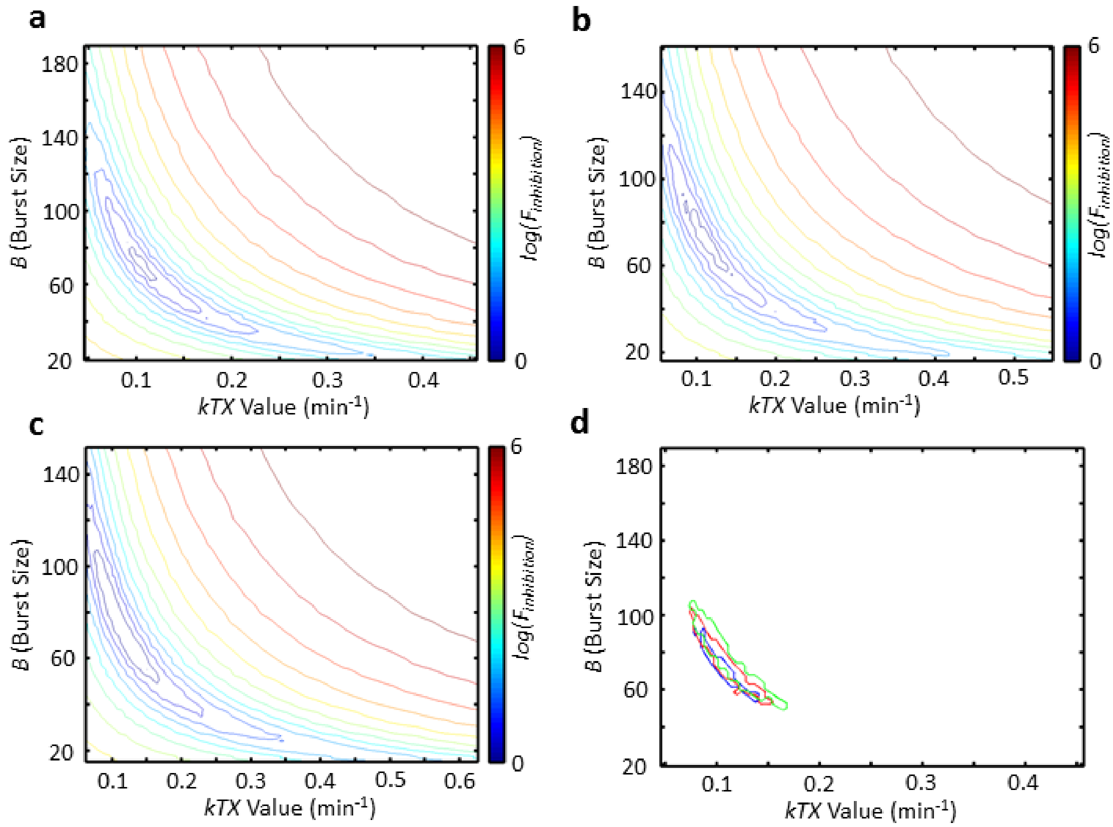

Figure B.13: Transcription rate and burst sizes for each replicate.

The sensitivity landscape of the fit to FISH data for perturbations in kTX and B with kACT and kMDEG fixed from each inhibition experimental replicate indicates a minimum in each case (a-c for reps 1-3, respectively). For each replicate, the region where solutions are indistinguishable from the minimum by t-test (see Main Text and Online Methods; n=5) are overlapping (d).

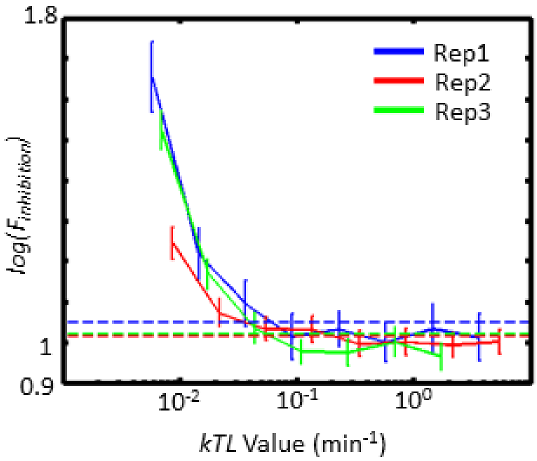

Figure B.14: Sensitivity to kTL and A with transcription parameters known.

After a certain threshold (dashed lines), kTL/A pairs become indistinguishable for the parameter sets corresponding to all 3 experimental replicates. Solid lines are the fold change for specific values of kTL with A scaled to keep P constant

B.3.6 Matching activation delay

We were unable to find parameter sets that matched the delay between induction and observed increases in fluorescence without resulting in a large number of cells that never produce fluorescence in the inhibition experiment. Indeed, the contribution of each time point to the total objective function value is consistently dominated by the points around where the cells begin to exhibit fluorescence (Figure B.15). The simulated cells tend to show fluorescence a bit earlier and undergo a less dramatic rise, as compared to the experimental data, before settling into behavior that mimics the data well. Additionally, the fit to the early time point in the FISH data (t=20 min) is poor, and the fit for the later time points (t=40 min and t=60 min) does not match the proportion of cells that are still "off" (i.e. zero or a few mRNAs) observed in the experimental data (Figure 5.3a, Chapter 5). It seems likely that these discrepancies result from an inadequacy of our single mass-action equation to model the complex process of activation of the GAL promoter (see Main Text for further discussion).

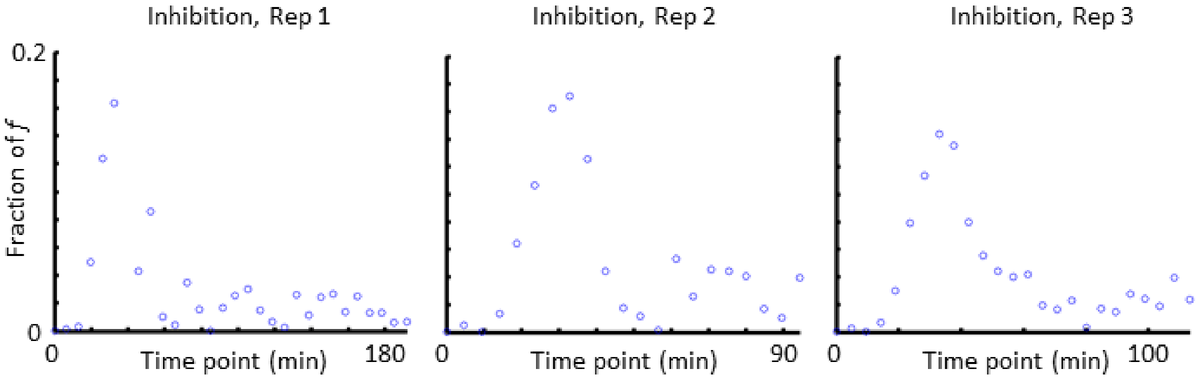

Figure B.15: The model fits the initial stage of induction poorly.
The contribution of each time point to the total objective function is displayed for each inhibition replicate. The largest contributors to the overall value of the objective function are those near where an increase in fluorescence begins to be detected.

B.4 References

1. Huh, D. & Paulsson, J. Non-genetic heterogeneity from stochastic partitioning at cell division. Nature genetics 43, 95-100 (2011).
2. Rosenfeld, N., Young, J.W., Alon, U., Swain, P.S. & Elowitz, M.B. Gene regulation at the single-cell level. Science 307, 1962-1965 (2005).