# Synthetic Biology: A New Application Area for Design Automation Research

Chris Myers

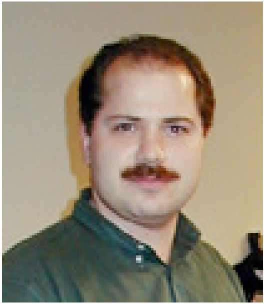
University of Utah

Carnegie Mellon University
December 10, 2009

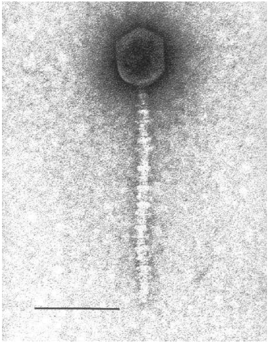
# James Watson

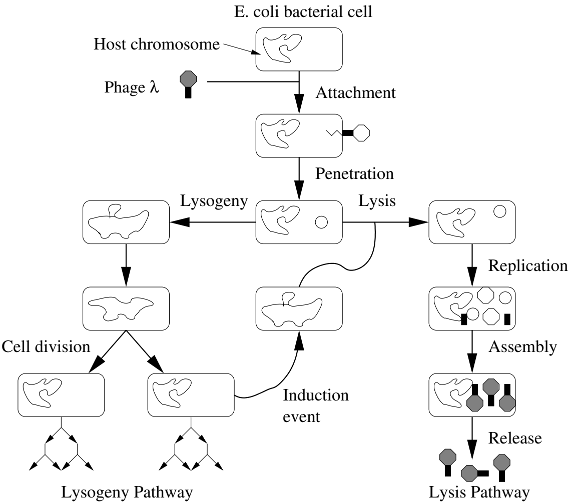
*Biology has at least 50 more interesting years (1984).*

# Michael Samoilov and Adam Arkin

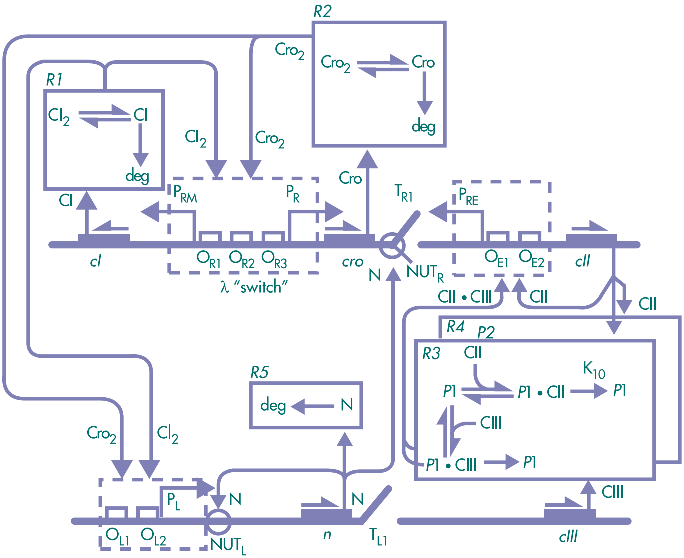


# Phage λ Virus

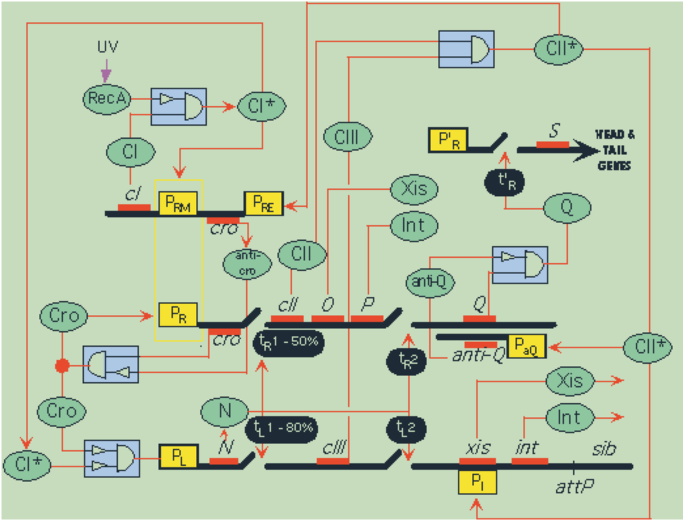




# Phage λ Decision Circuit

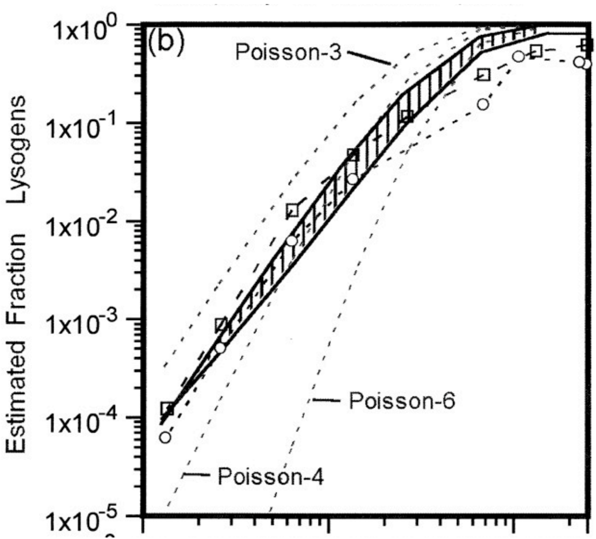


# Asynchronous Circuit?



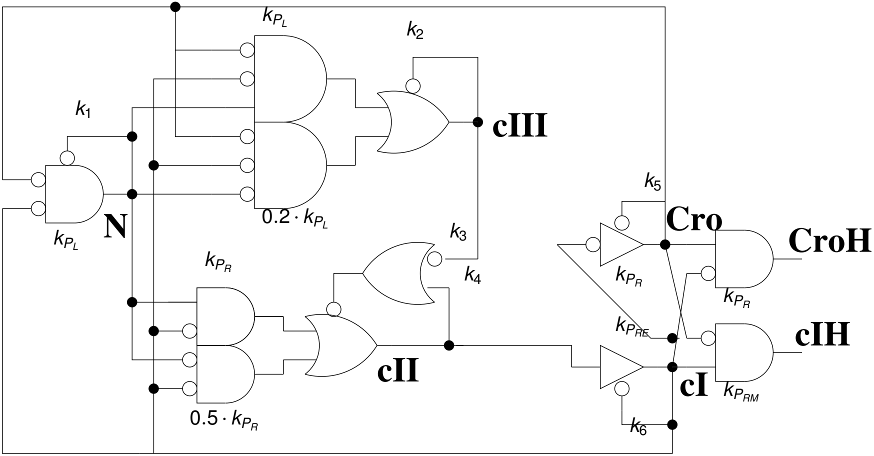

McAdams/Shapiro, Science (1995)

# Stochastic Circuit?

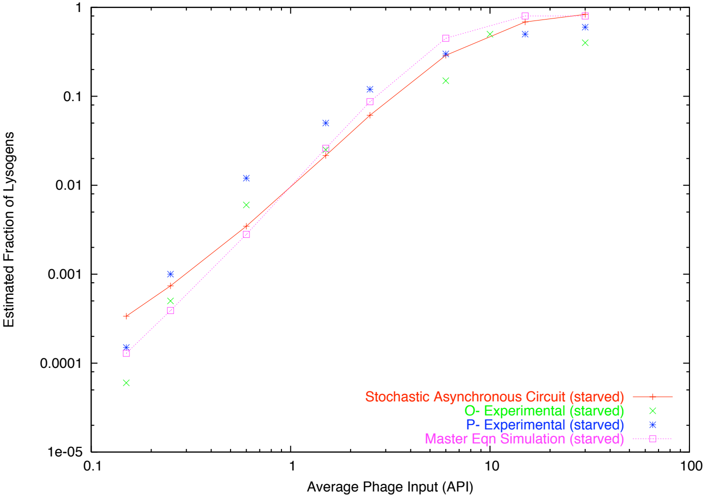



Arkin/Ross/McAdams, Genetics (1998)

# Stochastic Asynchronous Circuit?



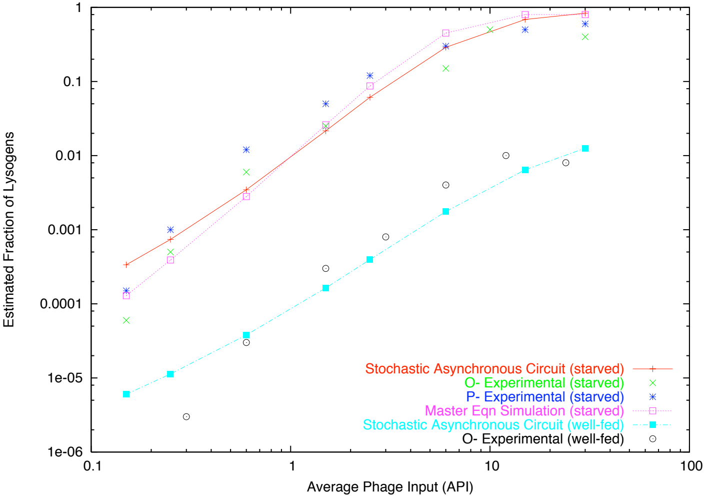
# Stochastic Asynchronous Circuit Results




SAC results generated in only 7 minutes.

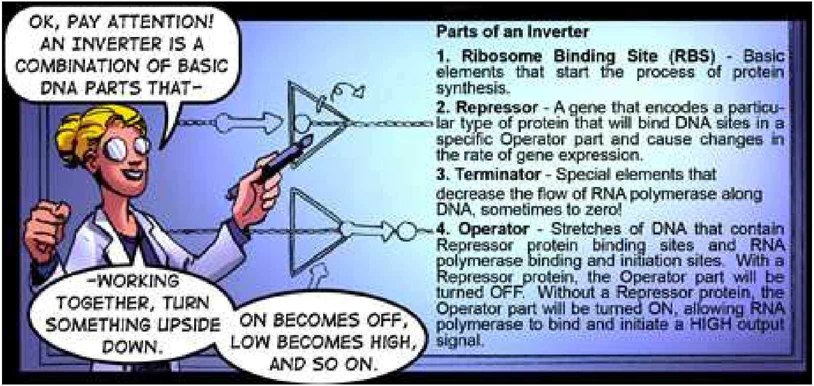
Kuwahara et al., Trans. on Comp. Sys. Bio. (2006)

# Stochastic Asynchronous Circuit Results




SAC results generated in only 7 minutes.

Kuwahara et al., Trans. on Comp. Sys. Bio. (2006)




(From "Adventures in Synthetic Biology" - Endy et al.)

# Genetic Engineering vs. Synthetic Biology

- *Genetic engineering* (last 30 years):
  - *Recombinant DNA* - constructing artificial DNA through combinations.
  - *Polymerase Chain Reaction* (PCR) - making many copies of this new DNA.
  - *Automated sequencing* - checking the resulting DNA sequence.
- Synthetic biology adds:
  - *Standards* - create repositories of parts that can be easily composed.
  - *Abstraction* - high-level models to facilitate design.
  - *Automated construction* - separate design from construction.

(source: Drew Endy)

# Genetic Design Automation (GDA)

- Standards, abstraction, and automated construction are the cornerstones of *Electronic Design Automation* (EDA).
- EDA facilitates the design of more complex integrated circuits each year.
- Crucial to the success of synthetic biology is an improvement in methods and tools for *Genetic Design Automation* (GDA).
- Experiences with EDA can jump start the development of GDA.

# Current State of GDA (Standards)

- Registry of standard biological parts used to design synthetic genetic circuits (`http://partsregistry.org`).
- Adequate characterization of these parts is an ongoing effort.
- *Systems Biology Markup Language* (SBML) has been proposed as a standard representation for the simulation of biological systems.
- Many simulation tools have been developed that accept models in the SBML format (Copasi, Jarnac, CellDesigner, SimBiology, iBioSim, etc.).

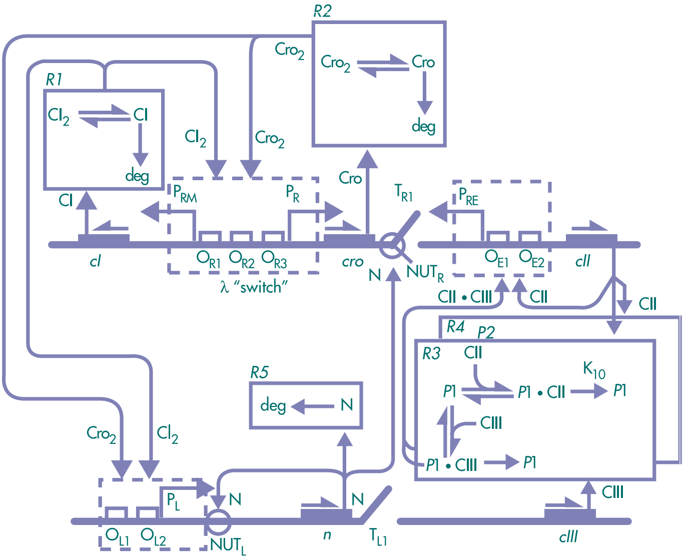
# Current State of GDA (Abstraction)

- Existing SBML-based GDA tools model biological systems at the molecular level.
- A typical SBML model is composed of a number of chemical *species* (i.e., proteins, genes, etc.) and *reactions* that transform these species.
- This is a very low level representation which is roughly equivalent to the layout level for electronic circuits.
- Designing and simulating genetic circuits at this level of detail is extremely tedious and time-consuming.

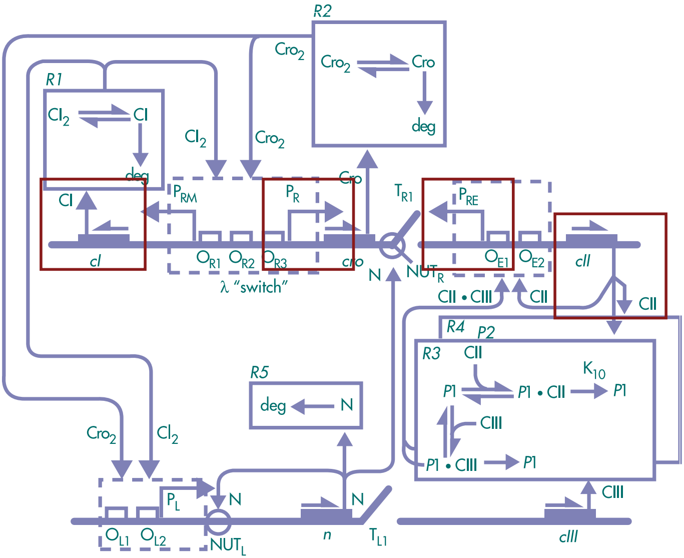
# Current State of GDA (Automated Construction)

- Several companies have formed that will construct a plasmid from an arbitrary DNA sequence.
- It is still difficult, however, to separate design and construction issues.
- To achieve this, a GDA tool that supports higher-levels of abstraction for modeling, analysis, and design of genetic circuits is essential.

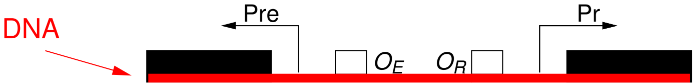
# Phage λ Decision Circuit



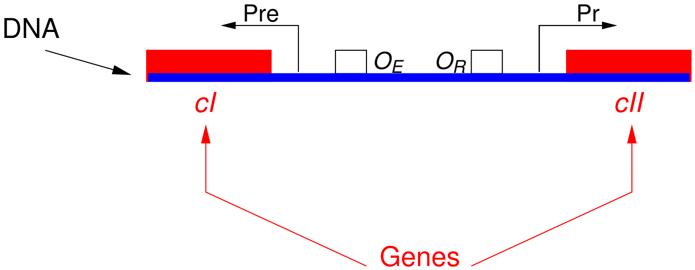
# Phage λ Decision Circuit



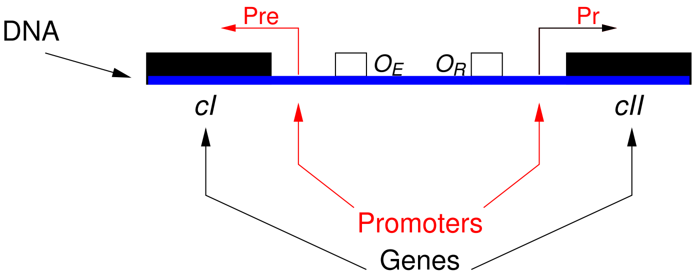
# Genetic Circuits



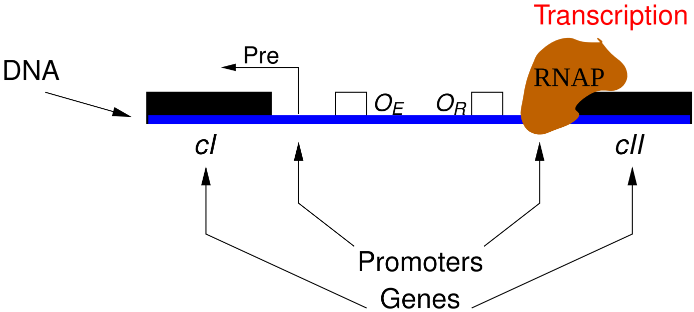
# Genetic Circuits

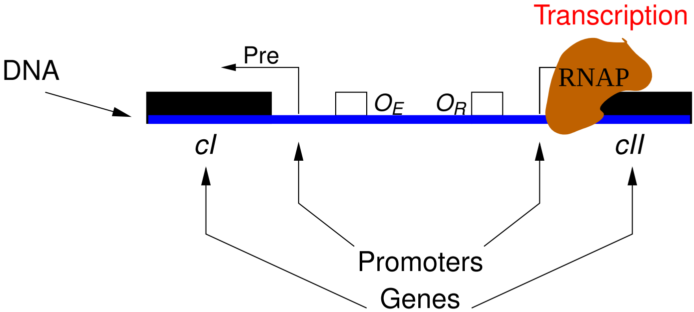


# Genetic Circuits

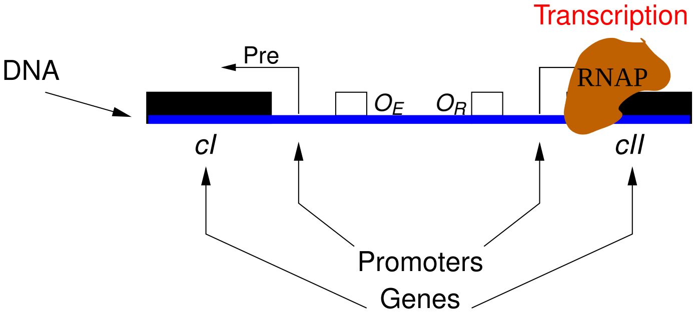


# Genetic Circuits

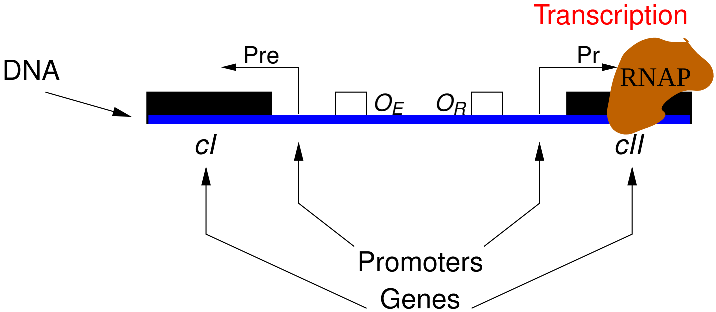



Transcription
Pre
DNA
RNAP
$O_E$
$O_R$
*cI*
*cII*
Promoters
Genes

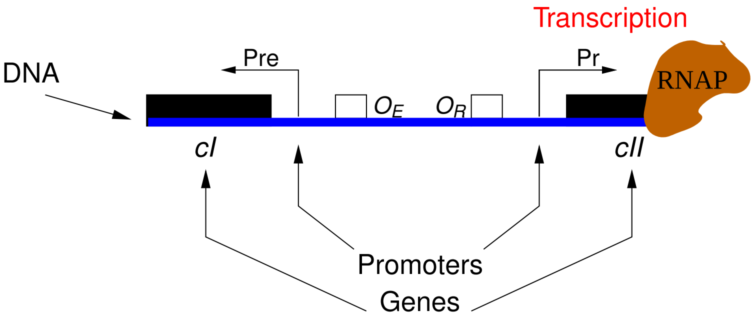
# Genetic Circuits



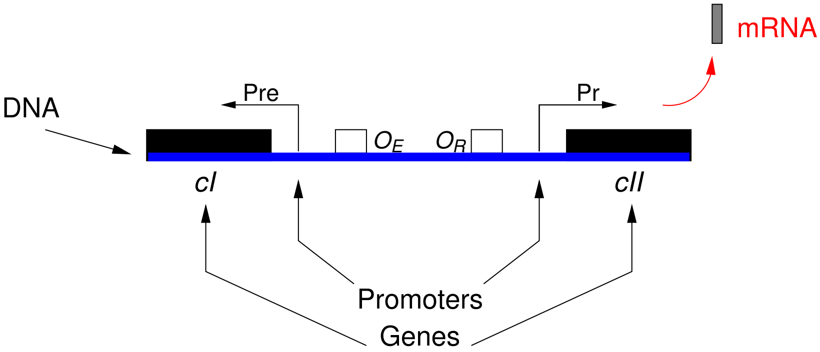
# Genetic Circuits



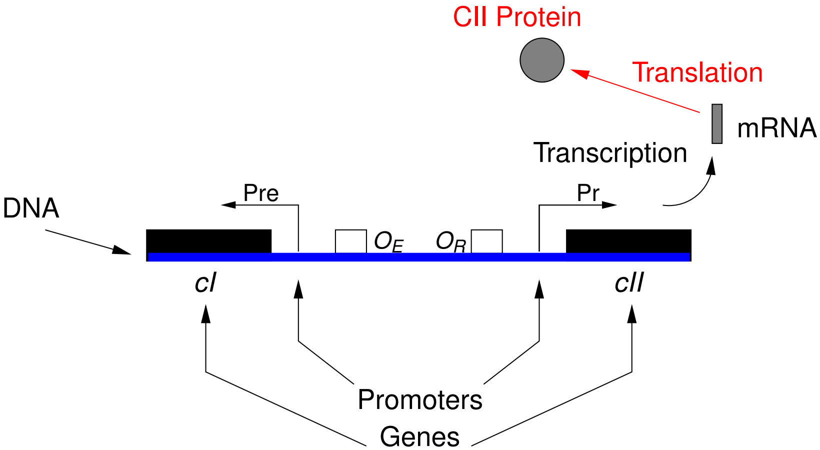
# Genetic Circuits



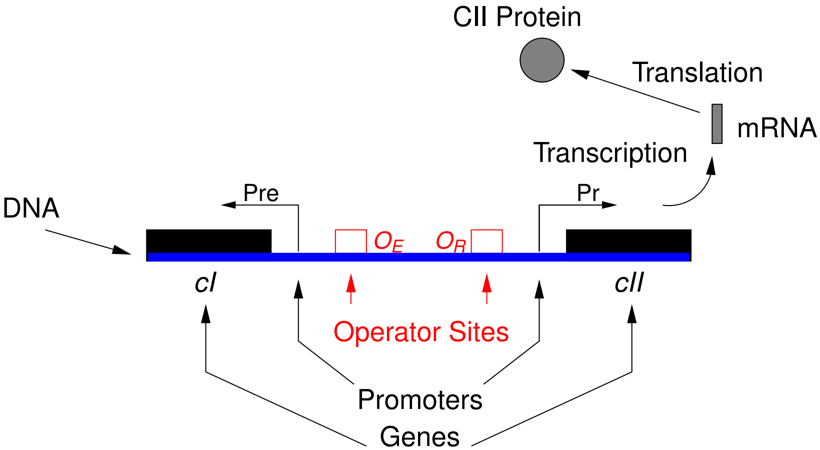
# Genetic Circuits

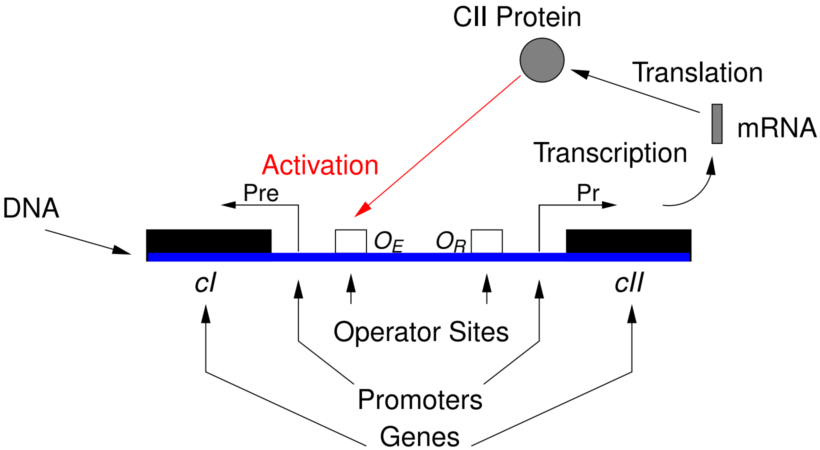


# Genetic Circuits

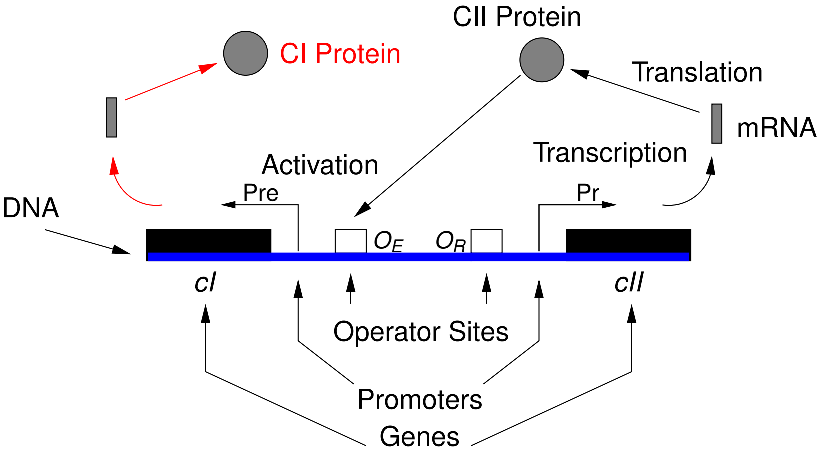


# Genetic Circuits

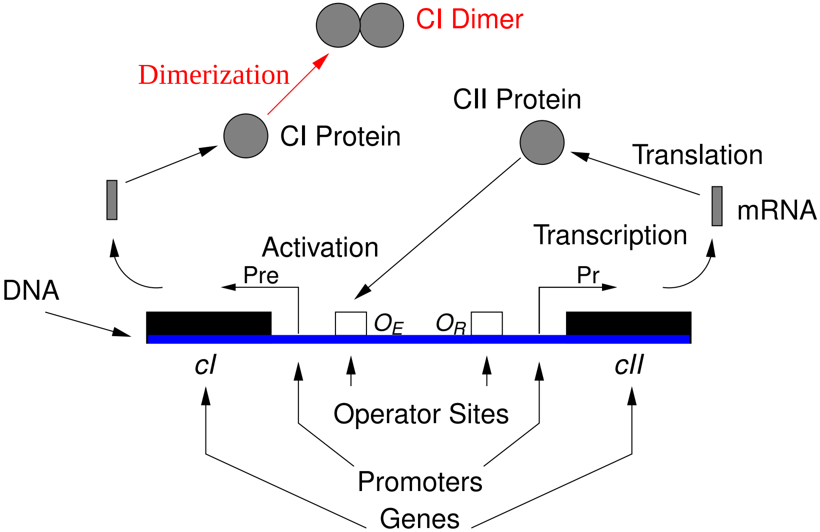



CII Protein
Translation
mRNA
Transcription
Activation
Pre
Pr
DNA
$O_E$
$O_R$
*cI*
*cII*
Operator Sites
Promoters
Genes

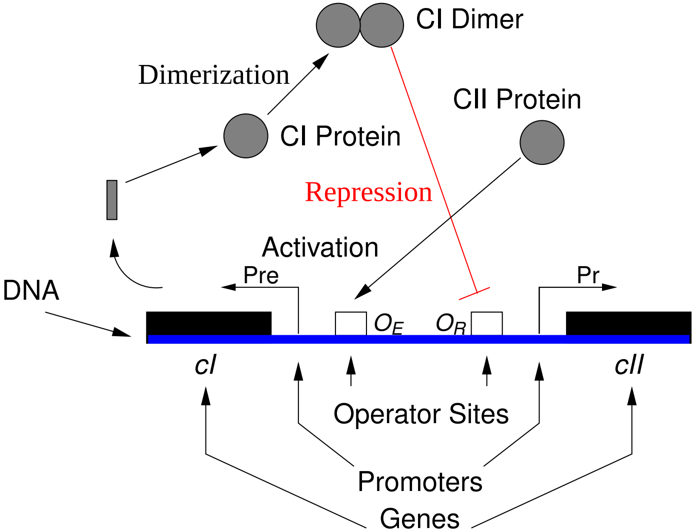
# Genetic Circuits

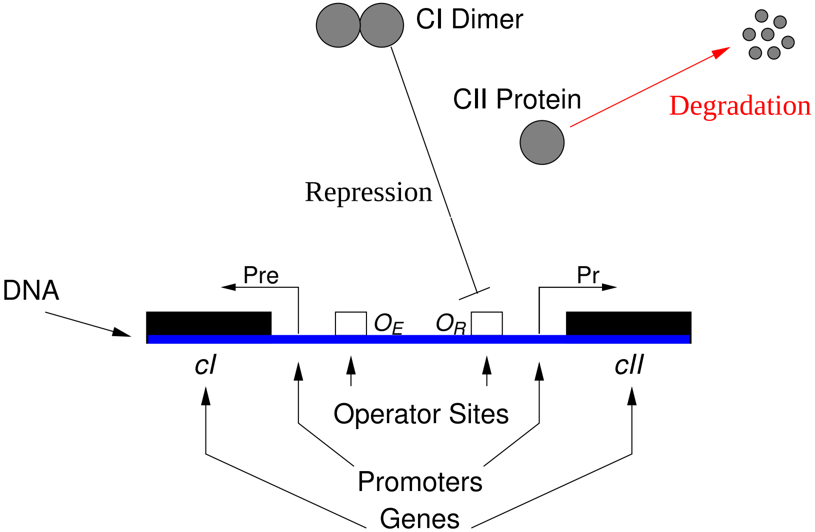


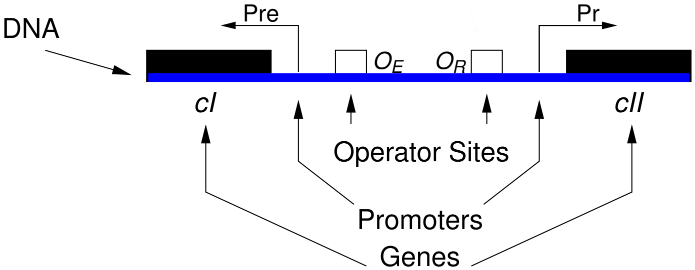
# Genetic Circuits



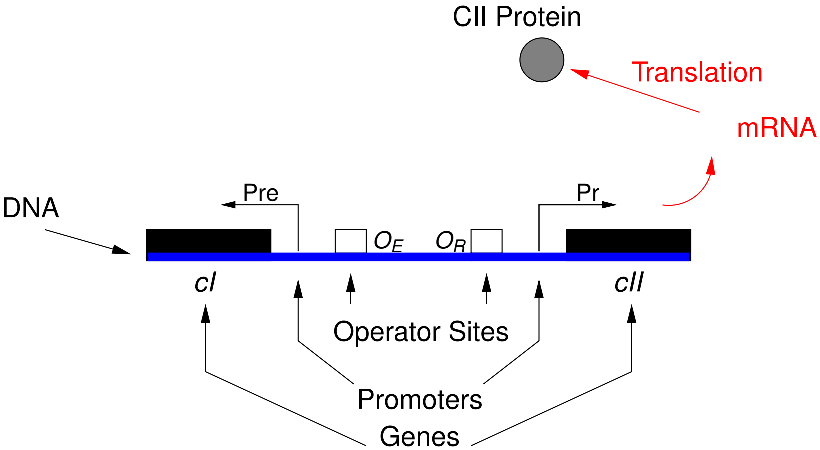
# Genetic Circuits




CI Dimer
CII Protein
Degradation
Repression
DNA
Pre
Pr
$O_E$
$O_R$
cI
cII
Operator Sites
Promoters
Genes


DNA
Pre
Pr
$O_E$
$O_R$
cI
cII
Operator Sites
Promoters
Genes

# Genetic Circuits



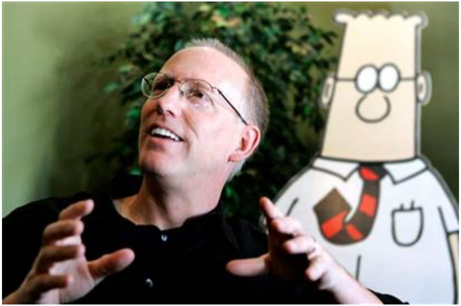
# Francis Crick

*DNA makes RNA, RNA makes protein, and proteins make us.*

# Johann Von Neumann

*The sciences do not try to explain, they hardly even try to interpret, they mainly make models. By a model is meant a mathematical construct which, with the addition of certain verbal interpretations, describes observed phenomena. The justification of such a mathematical construct is solely and precisely that it is expected to work.*

# Scott Adams

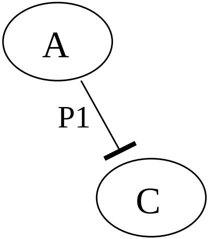


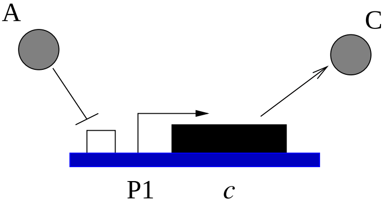
*There are many methods for predicting the future. For example, you can read horoscopes, tea leaves, tarot cards, or crystal balls. Collectively, these methods are known as "nutty methods." Or you can put well-researched facts into sophisticated computer models, more commonly referred to as "a complete waste of time."*

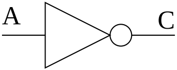
# Genetic Circuit Model (GCM)

- Provides a higher level of abstraction than SBML.
- Includes only important species and their influences upon each other.
- GCMs also include structural constructs that allow us to connect GCMs for separate modules through species ports.

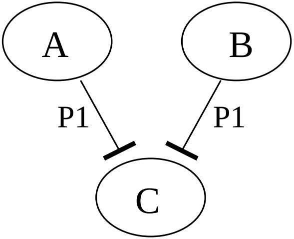
# A Genetic Not Gate

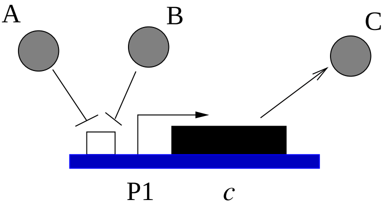


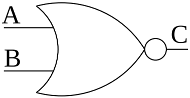



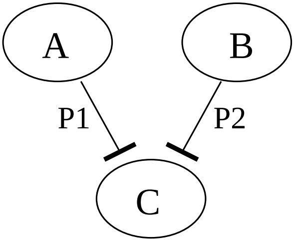
# A Genetic Nor Gate

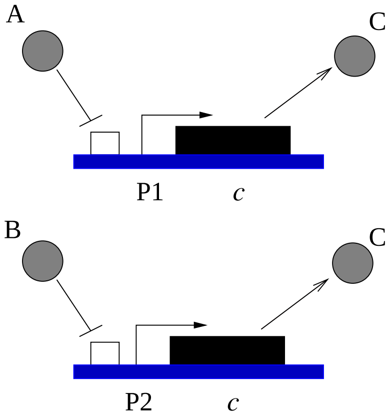


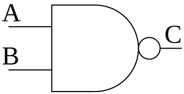



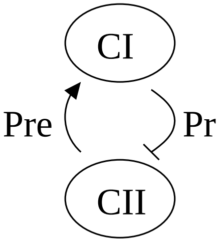
# A Genetic Nand Gate

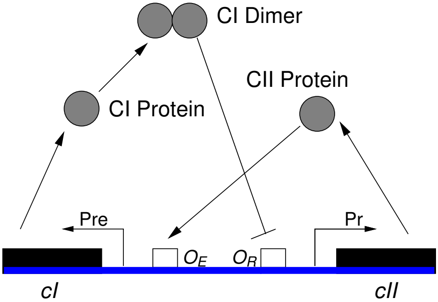


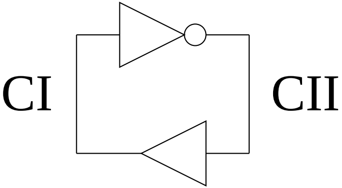



# A Genetic Oscillator

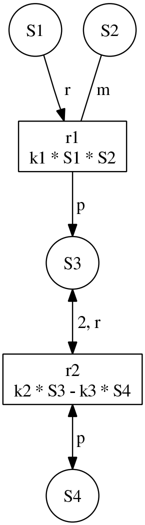


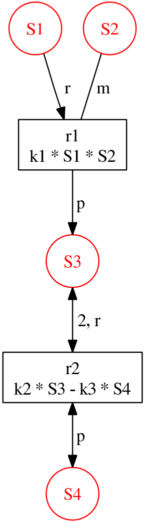
# SBML: Main Elements

- Species
- Global parameters (ex. k1=0.1)
- Reactions
  - Reactants
  - Products
  - Modifiers
  - Stoichiometry
  - Reversible
  - Kinetic laws



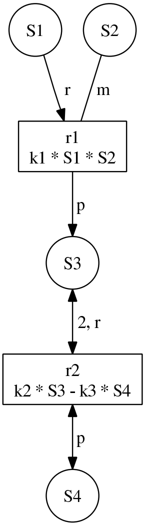
# SBML: Main Elements

- Species
- Global parameters (ex. k1=0.1)
- Reactions
  - Reactants
  - Products
  - Modifiers
  - Stoichiometry
  - Reversible
  - Kinetic laws



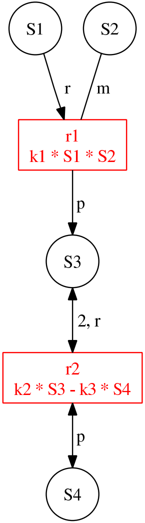
# SBML: Main Elements

- Species
- Global parameters (ex. k1=0.1)
- Reactions
  - Reactants
  - Products
  - Modifiers
  - Stoichiometry
  - Reversible
  - Kinetic laws



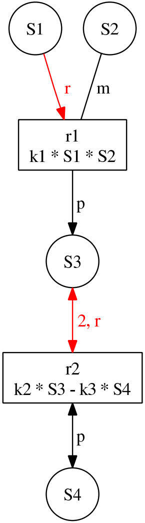
# SBML: Main Elements

- Species
- Global parameters (ex. k1=0.1)
- Reactions
  - Reactants
  - Products
  - Modifiers
  - Stoichiometry
  - Reversible
  - Kinetic laws



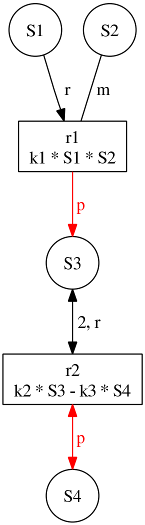
# SBML: Main Elements

- Species
- Global parameters (ex. k1=0.1)
- Reactions
  - Reactants
  - Products
  - Modifiers
  - Stoichiometry
  - Reversible
  - Kinetic laws



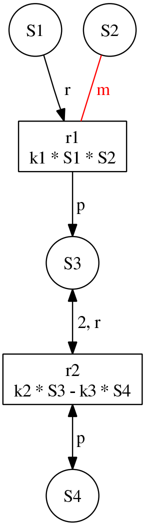
# SBML: Main Elements

- Species
- Global parameters (ex. k1=0.1)
- Reactions
  - Reactants
  - Products
  - Modifiers
  - Stoichiometry
  - Reversible
  - Kinetic laws



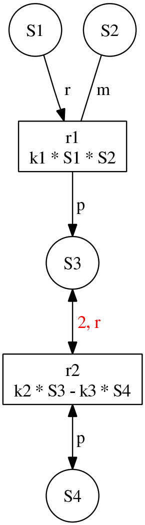
# SBML: Main Elements

- Species
- Global parameters (ex. k1=0.1)
- Reactions
  - Reactants
  - Products
  - Modifiers
  - Stoichiometry
  - Reversible
  - Kinetic laws



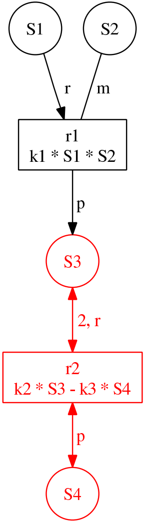
# SBML: Main Elements

- Species
- Global parameters (ex. k1=0.1)
- Reactions
  - Reactants
  - Products
  - Modifiers
  - Stoichiometry
  - Reversible
  - Kinetic laws



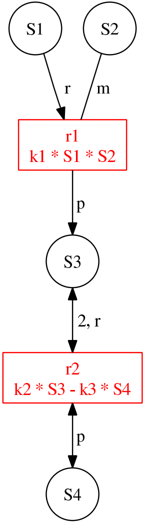
# SBML: Main Elements

- Species
- Global parameters (ex. k1=0.1)
- Reactions
  - Reactants
  - Products
  - Modifiers
  - Stoichiometry
  - Reversible
  - Kinetic laws



# SBML: Main Elements

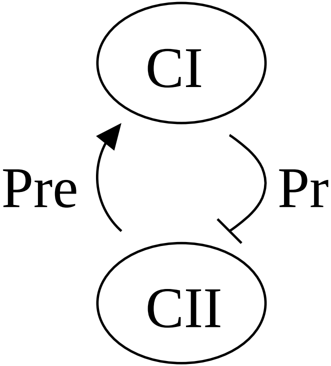
- Species
- Global parameters (ex. k1=0.1)
- Reactions
  - Reactants
  - Products
  - Modifiers
  - Stoichiometry
  - Reversible
  - Kinetic laws



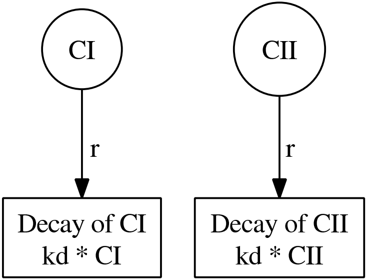
# Synthesizing SBML from a GCM Representation

- Create degradation reactions
- Create open complex formation reactions
- Create dimerization reactions
- Create repression reactions
- Create activation reactions

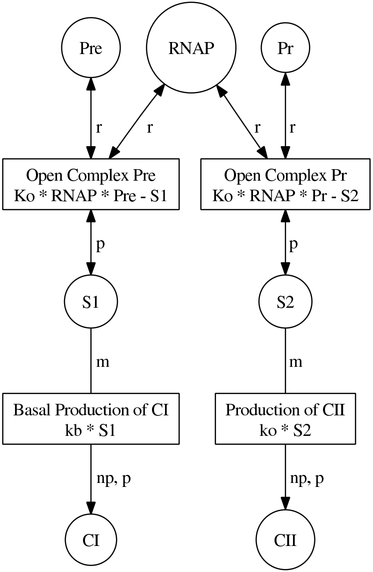
# GCM Example



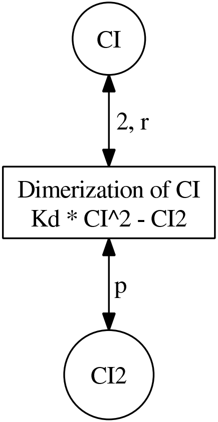
# Degradation Reactions



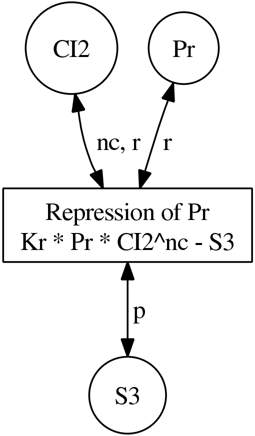
# Open Complex Formation Reactions



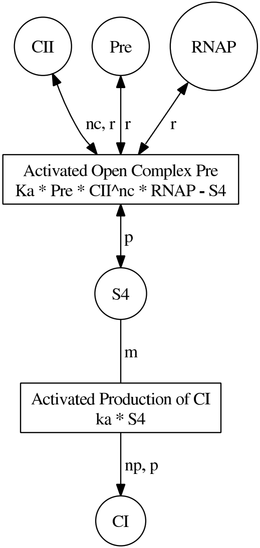
# Dimerization Reactions



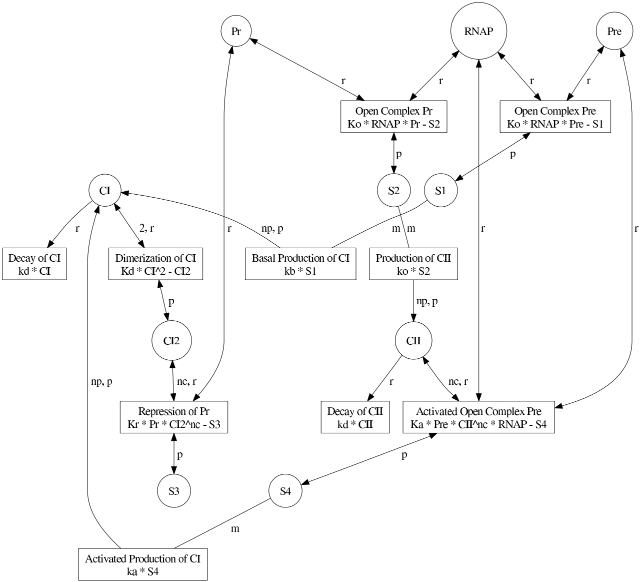
# Repression Reactions



# Activation Reactions



# Complete SBML Model



# Classical Chemical Kinetics

- Uses *ordinary differential equations* (ODE) to represent the system to be analyzed, and it assumes:
  - Molecule counts are high, so concentrations can be continuous variables.
  - Reactions occur continuously and deterministically.
- Genetic circuits have:
  - Small molecule counts which must be considered as discrete variables.
  - Gene expression reactions that occur sporadically.
- ODEs do not capture non-deterministic behavior.

# Richard Feynmann

*A philosopher once said "It is necessary for the very existence of science that the same conditions always produce the same results." Well, they do not. You set up the circumstances, with the same conditions every time, and you cannot predict behind which hole you will see the electron.*

# NYTimes: Expressing Our Individuality, the Way E. Coli Do

The New York Times

Research

Health All NYT Search



WORLD U.S. N.Y. / REGION BUSINESS TECHNOLOGY SCIENCE HEALTH SPORTS OPINION ARTS STYLE TRAVEL JOBS REAL ESTATE AUTOS

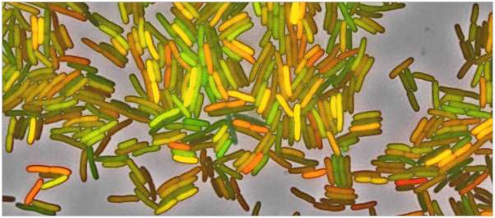
RESEARCH FITNESS & NUTRITION MONEY & POLICY VIEWS HEALTH GUIDE



Search Health 3,000+ Topics

Go

## Expressing Our Individuality, the Way E. Coli Do



Dr. Michael Elowitz

A colony of genetically identical E. coli is actually a mob of individuals. Under identical conditions, they behave in different ways.

By CARL ZIMMER
Published: April 22, 2008

SIGN IN TO E-MAIL

**Well**
Tara Parker-Pope on Health

Socializing Appears to Delay Memory Problems
June 4, 2008, 12:34 PM

Colon Cancer in Family Predicts Better Survival
June 3, 2008

Jane Brody's New Knees
June 3, 2008

Brain Surgeons and Cellphones
June 2, 2008

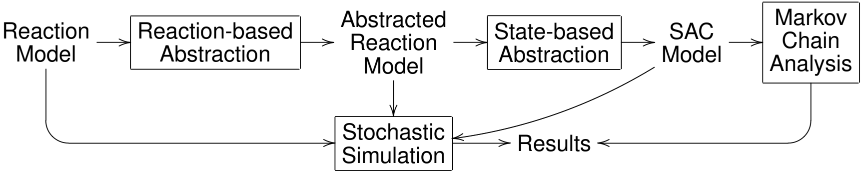
Cleaner Classrooms Mean Fewer Sick Kids
June 2, 2008

# Rainbow and CC

# Stochastic Chemical Kinetics

- To more accurately predict the temporal behavior of genetic circuits, *stochastic chemical kinetics* formalism can be used.
- Use Gillespie's *Stochastic Simulation Algorithm* which tracks the quantities of each molecular species and treats each reaction as a separate random event.
- Only practical for small systems with no major time-scale separations.
- Abstraction is essential for efficient analysis of any realistic system.

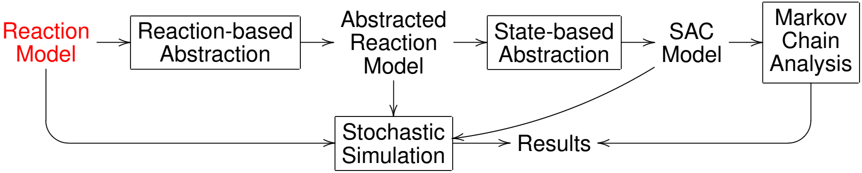
# Automatic Abstraction




- Begins with a *reaction-based model* in SBML.
- Automatically abstracts this model leveraging the quasi-steady state assumption, whenever possible.
- Encodes chemical species concentrations into Boolean (or n-ary) levels to produce a *stochastic asynchronous circuit* (SAC) model.
- Can now utilize Markov chain analysis.

Kuwahara et al., Trans. on Comp. Sys. Bio. (2006)

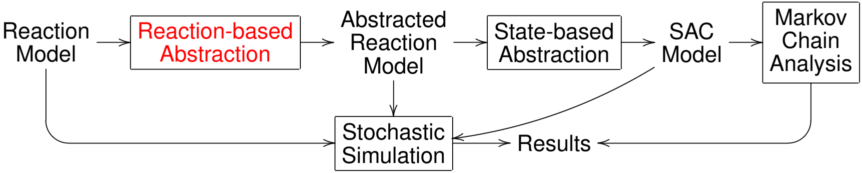
# Automatic Abstraction




- Begins with a *reaction-based model* in SBML.
- Automatically abstracts this model leveraging the quasi-steady state assumption, whenever possible.
- Encodes chemical species concentrations into Boolean (or n-ary) levels to produce a *stochastic asynchronous circuit* (SAC) model.
- Can now utilize Markov chain analysis.

Kuwahara et al., Trans. on Comp. Sys. Bio. (2006)

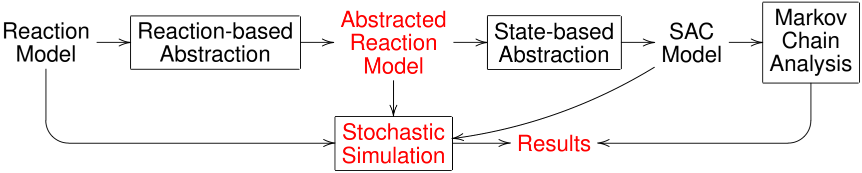
# Automatic Abstraction




- Begins with a *reaction-based model* in SBML.
- Automatically abstracts this model leveraging the quasi-steady state assumption, whenever possible.
- Encodes chemical species concentrations into Boolean (or n-ary) levels to produce a *stochastic asynchronous circuit* (SAC) model.
- Can now utilize Markov chain analysis.

Kuwahara et al., Trans. on Comp. Sys. Bio. (2006)

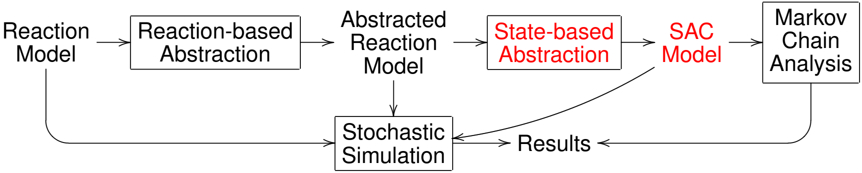
# Automatic Abstraction




- Begins with a *reaction-based model* in SBML.
- Automatically abstracts this model leveraging the quasi-steady state assumption, whenever possible.
- Encodes chemical species concentrations into Boolean (or n-ary) levels to produce a *stochastic asynchronous circuit* (SAC) model.
- Can now utilize Markov chain analysis.

Kuwahara et al., Trans. on Comp. Sys. Bio. (2006)

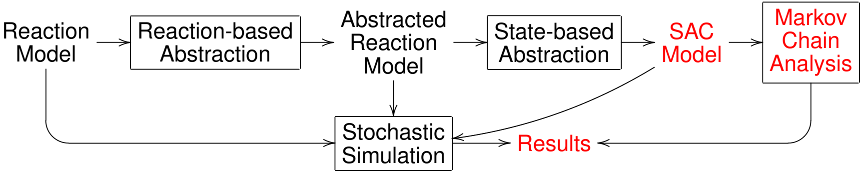
# Automatic Abstraction




- Begins with a *reaction-based model* in SBML.
- Automatically abstracts this model leveraging the quasi-steady state assumption, whenever possible.
- Encodes chemical species concentrations into Boolean (or n-ary) levels to produce a *stochastic asynchronous circuit* (SAC) model.
- Can now utilize Markov chain analysis.

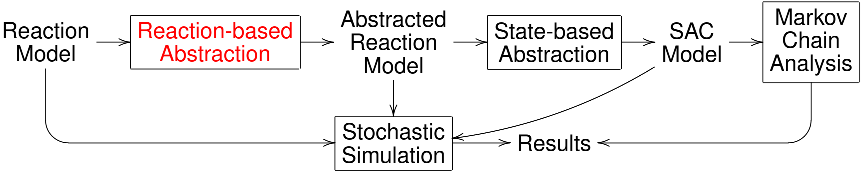
Kuwahara et al., Trans. on Comp. Sys. Bio. (2006)

# Automatic Abstraction




- Begins with a *reaction-based model* in SBML.
- Automatically abstracts this model leveraging the quasi-steady state assumption, whenever possible.
- Encodes chemical species concentrations into Boolean (or n-ary) levels to produce a *stochastic asynchronous circuit* (SAC) model.
- Can now utilize Markov chain analysis.

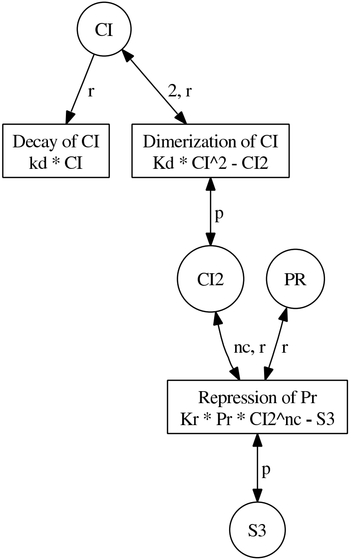
Kuwahara et al., Trans. on Comp. Sys. Bio. (2006)



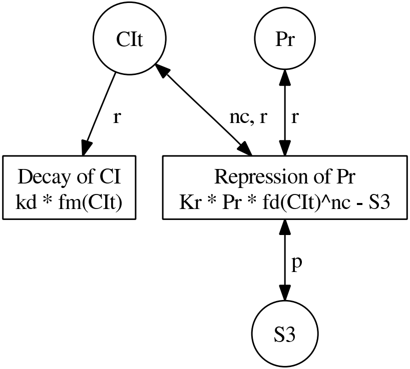

- Begins with a *reaction-based model* in SBML.
- Automatically abstracts this model leveraging the quasi-steady state assumption, whenever possible.
- Encodes chemical species concentrations into Boolean (or n-ary) levels to produce a *stochastic asynchronous circuit* (SAC) model.
- Can now utilize Markov chain analysis.

Kuwahara et al., Trans. on Comp. Sys. Bio. (2006)

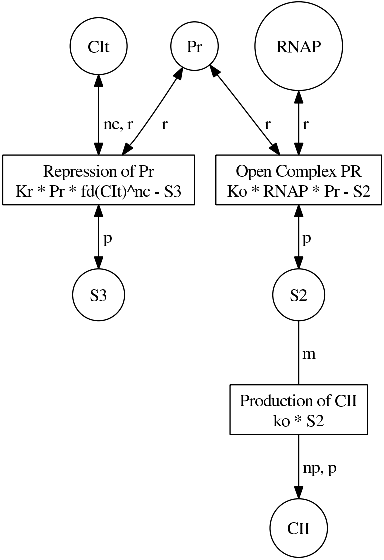
# Dimerization Reduction



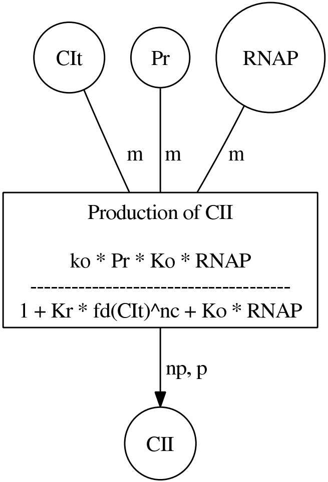
# Dimerization Reduction



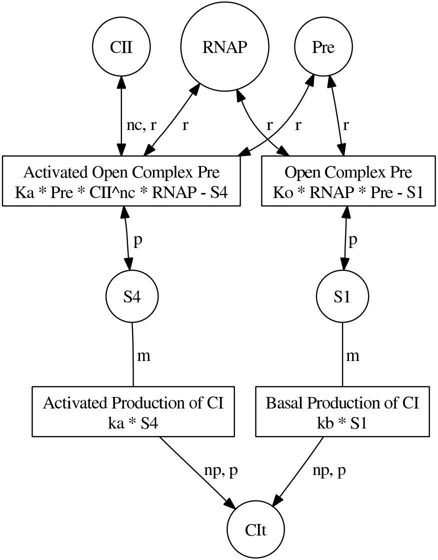
# Operator Site Reduction (PR)



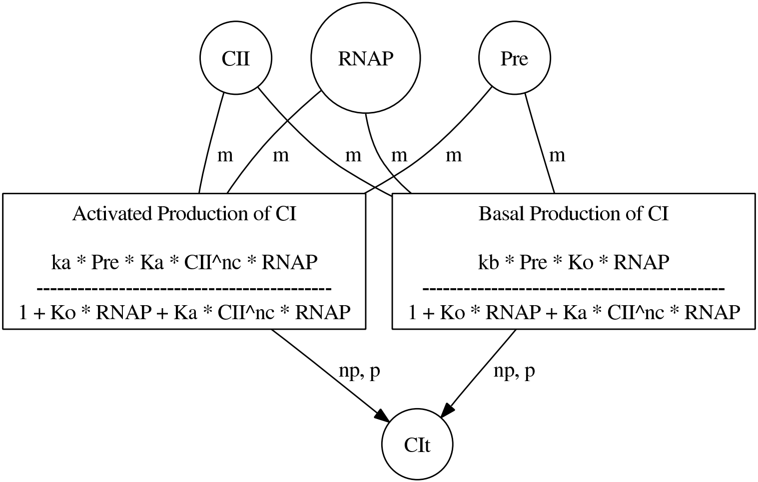
# Operator Site Reduction (PR)



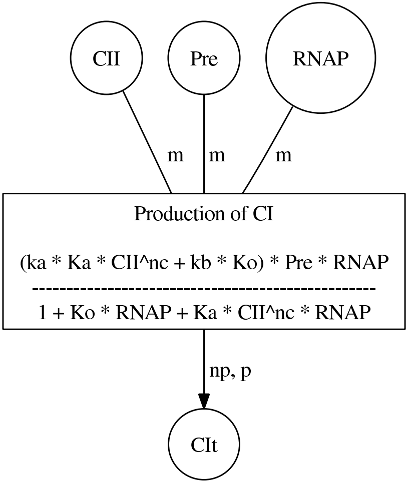
# Operator Site Reduction (PRE)



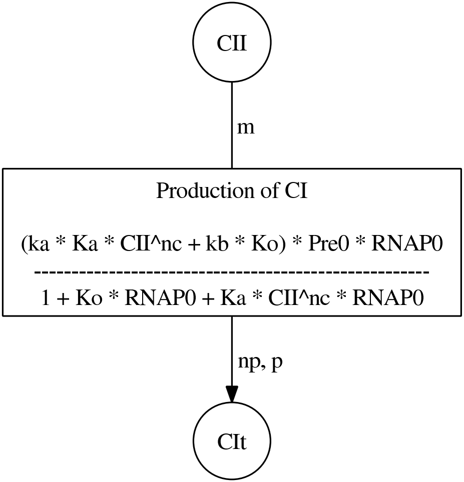
# Operator Site Reduction (PRE)



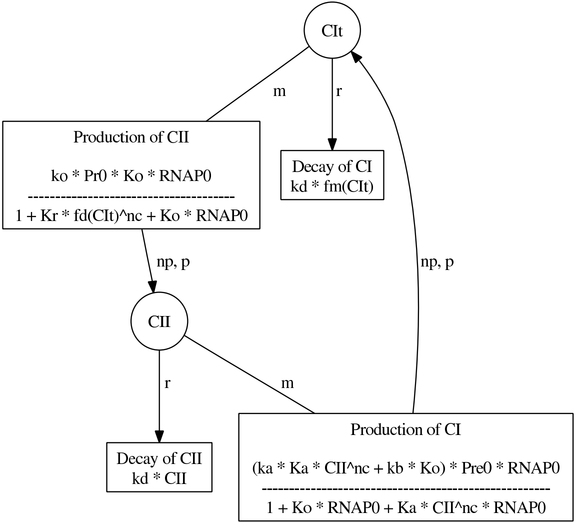
# Similar Reaction Combination



# Modifier Constant Propagation



# Final SBML Model




10 species and 10 reactions reduced to 2 species and 4 reactions

# GCM Advantages

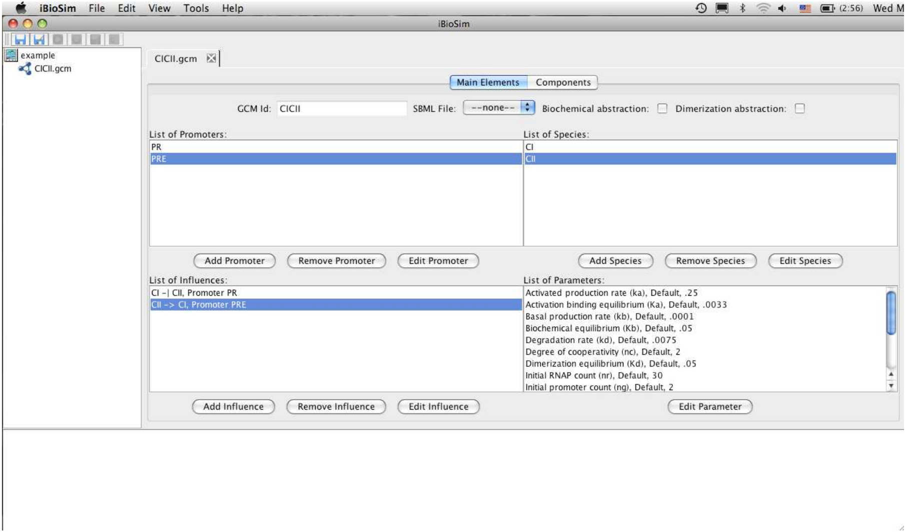
- Greatly increases the speed of model development and reduces the number of errors in the resulting models.
- Allows efficient exploration of the effects of parameter variation.
- Constrains SBML model such that it can be more easily abstracted resulting in substantial improvement in simulation time.

# iBioSim: The Intelligent Biological Simulator

- Project management support.
- GCM Editor - creates *Genetic Circuit Models* (GCM).
- SBML Editor - creates models using the *Systems Biology Markup Language* (SBML).
- `reb2sac` - abstraction-based ODE, Monte Carlo, and Markov analysis.
- TSD Graph Editor - visualizes *time series data* (TSD).
- Probability Graph Editor - visualizes probability data.
- `GeneNet` - learns GCMs from TSD.

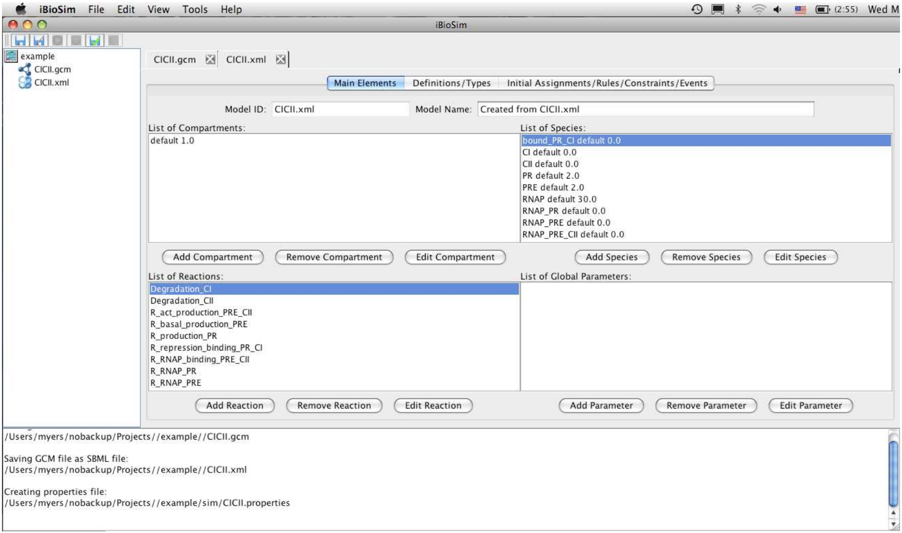
Myers et al., Bioinformatics (2009)

# iBioSim: Genetic Circuit Editor



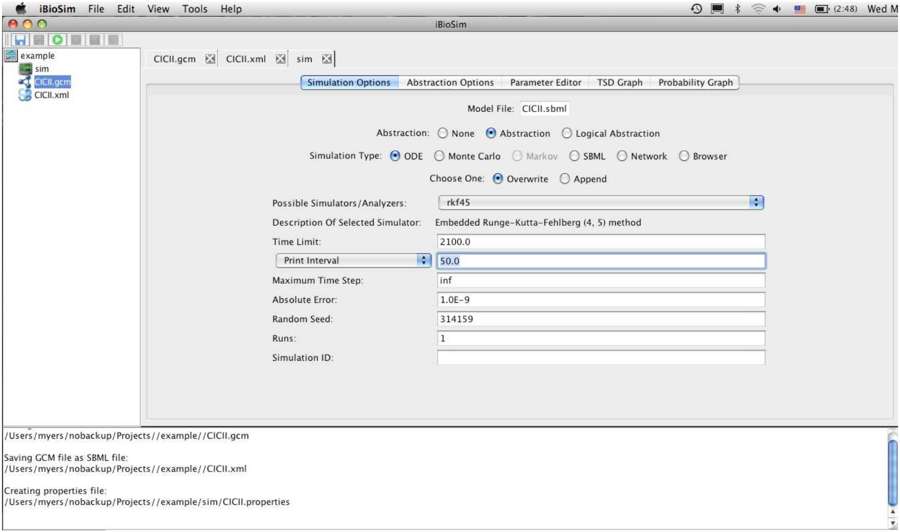

Myers et al., Bioinformatics (2009)

# iBioSim: SBML Editor



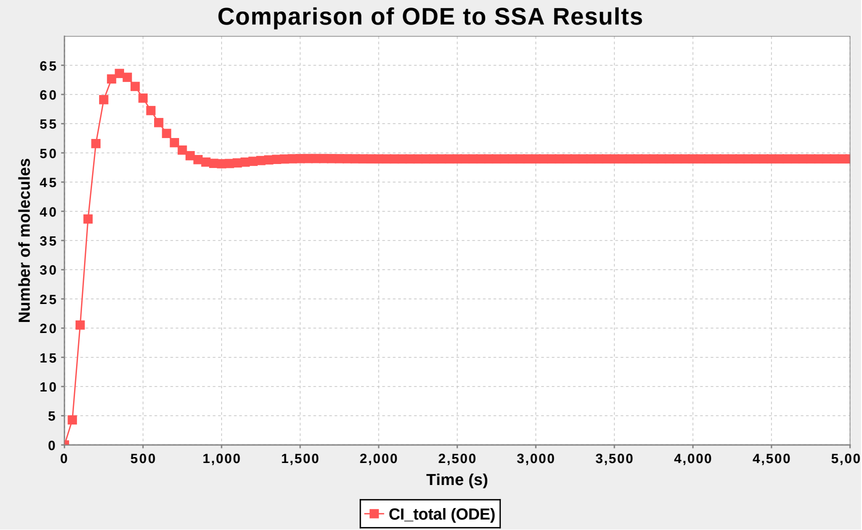

Myers et al., Bioinformatics (2009)

# iBioSim: Analysis Engine

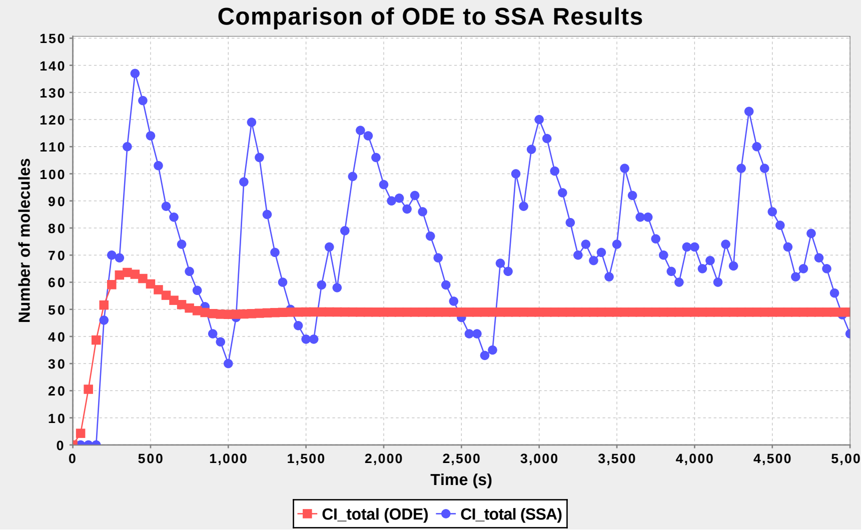



Myers et al., Bioinformatics (2009)

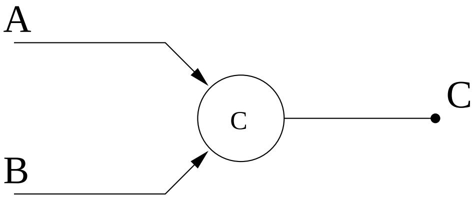
# ODE Results for the Simple Genetic Oscillator



# SSA Results for the Simple Genetic Oscillator

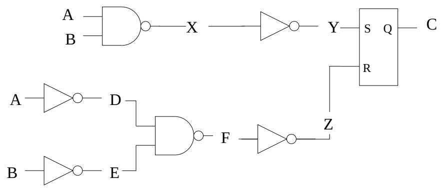


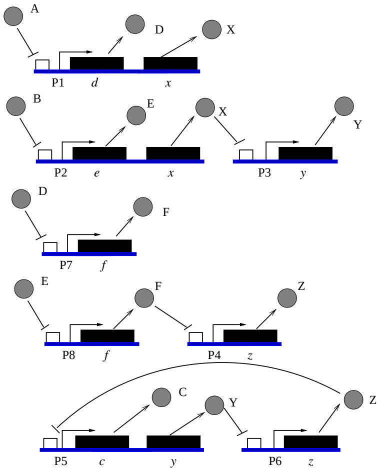
# Genetic Muller C-Element




| A | B | C |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | C |
| 1 | 0 | C |
| 1 | 1 | 1 |

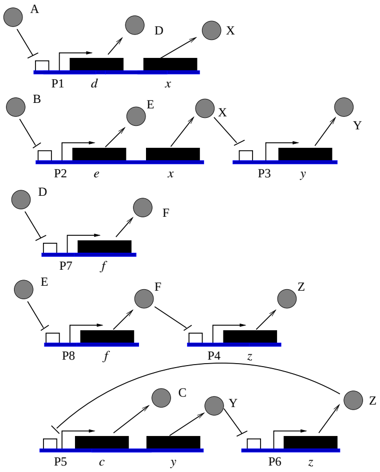
# Toggle Switch C-Element (Genetic Circuit)

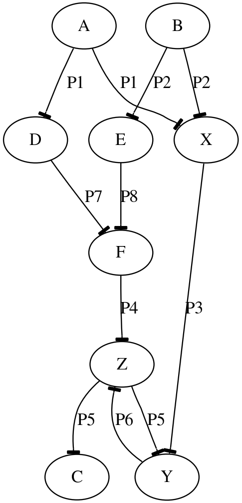


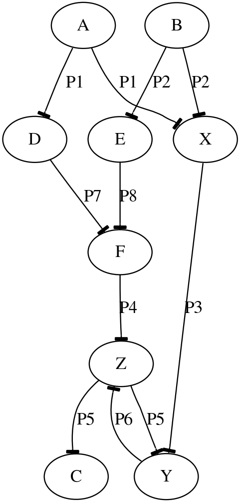

Nguyen et al., 13th Symposium on Async. Ckts. & Sys., 2007 (**best paper**)
Nguyen et al., to appear in the Journal of Theoretical Biology

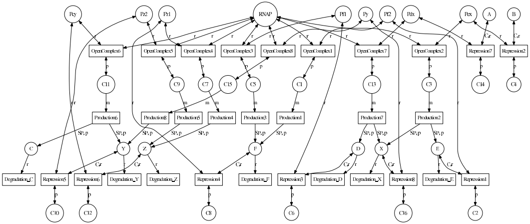
# Toggle Switch C-Element (GCM)



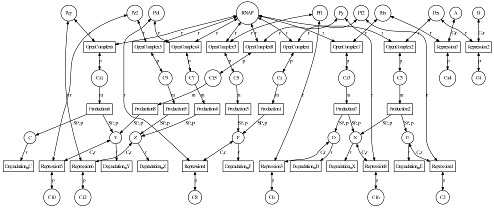



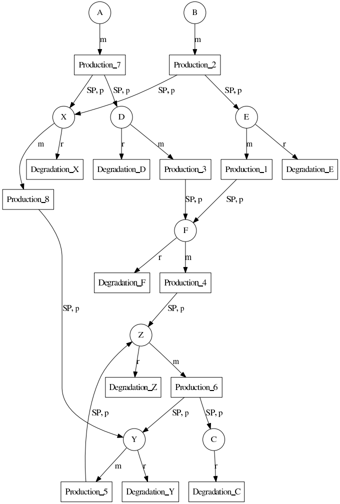
# Toggle Switch C-Element (SBML)



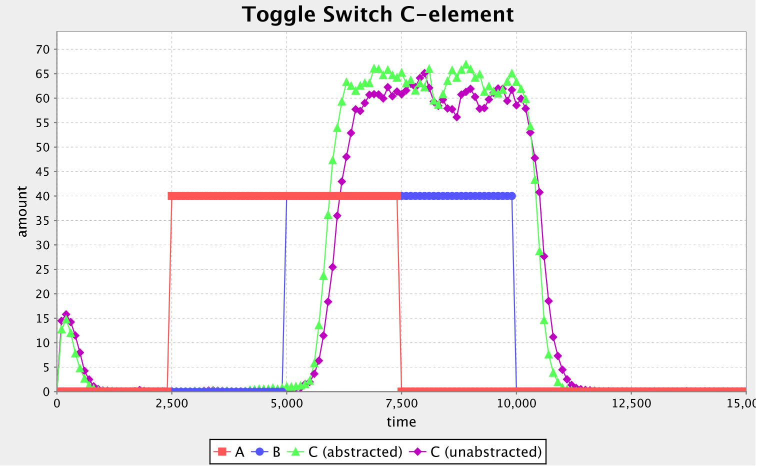
# Toggle Switch C-Element (Abstracted)





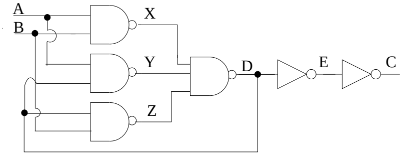

Reduced from 34 species and 31 reactions to 9 species and 15 reactions.

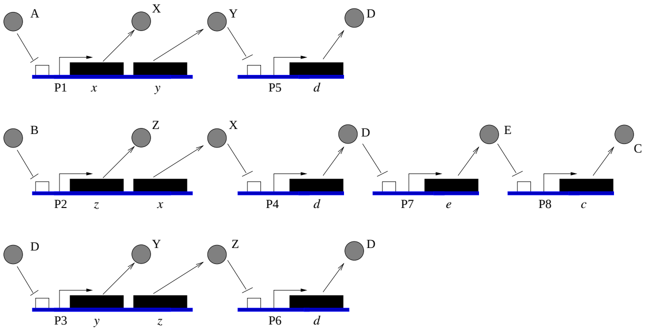
# Toggle Switch C-Element (Simulation)



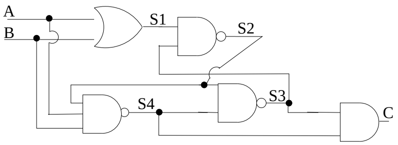

Simulation time improved from 312 seconds to 20 seconds.

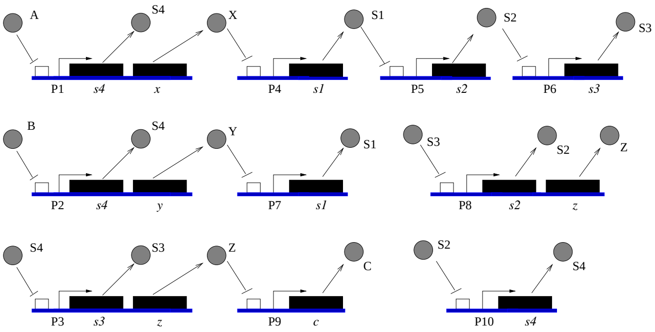
# Majority Gate C-Element (Genetic Circuit)



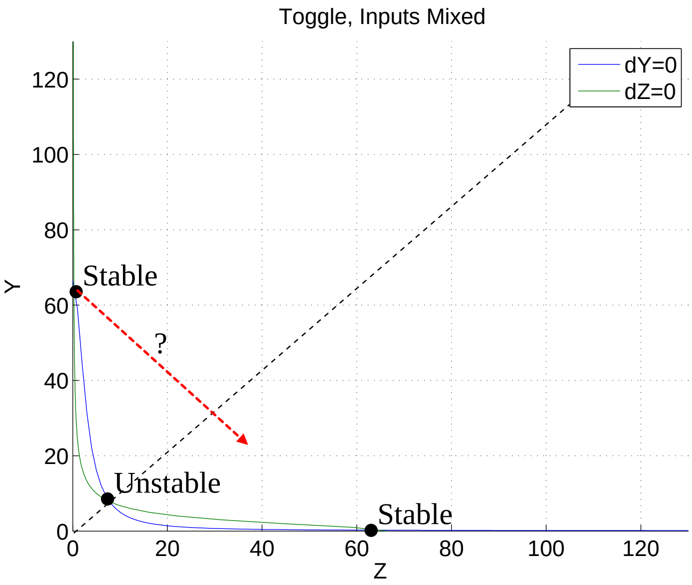



# Speed-Independent C-Element (Genetic Circuit)

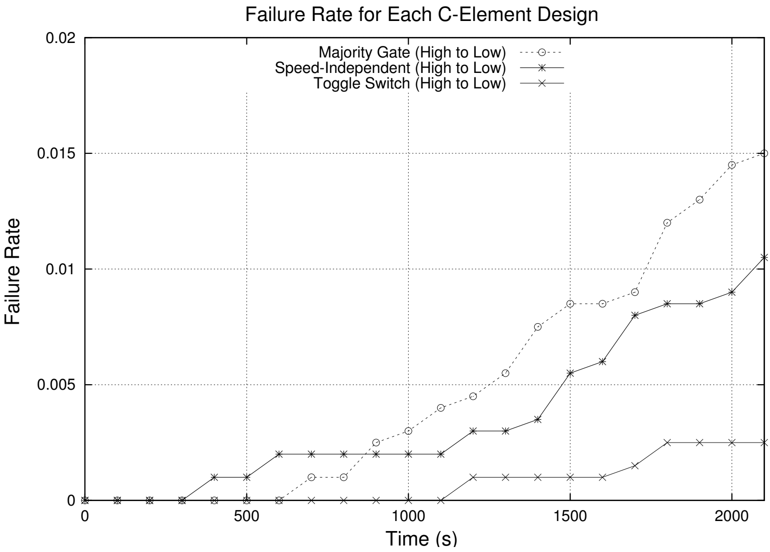


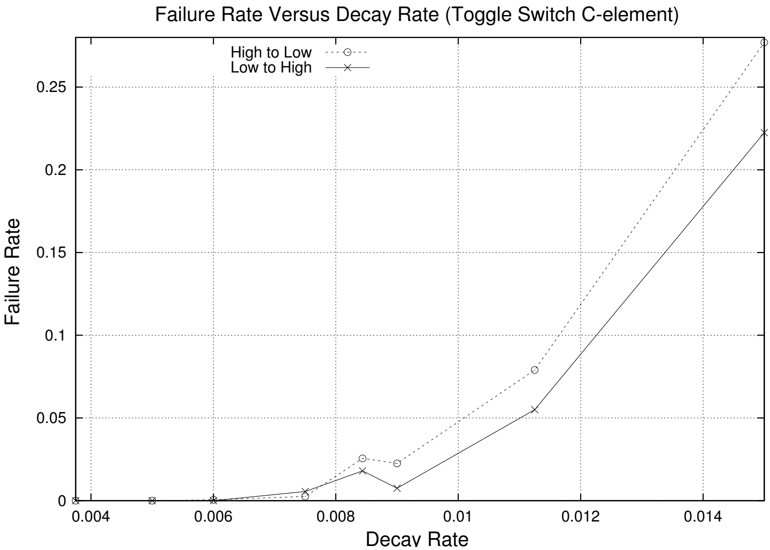
# Genetic C-element Failures



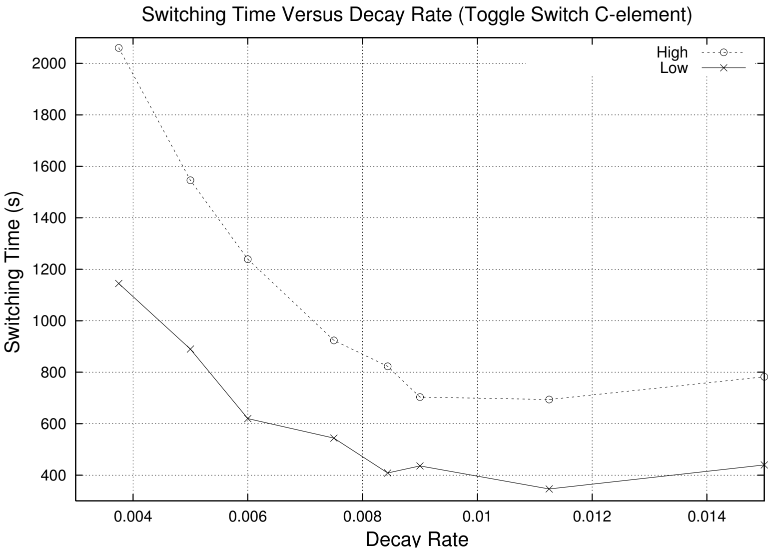
# Comparison of Failure Rates for the C-element Designs



# Effects of Decay Rates on Failure Rates

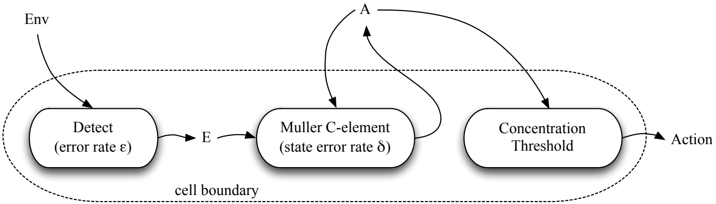


# Effects of Decay Rates on Switching Time



# Application: Bacterial Consensus

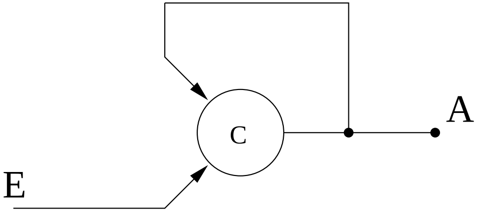
- One interesting application for synthetic biology is the design of bacteria that can hunt and kill tumor cells (Anderson et al.).
- Care must be taken in determining when to attack potential tumor cells.
- Can use a genetic Muller C-element and a bacterial consensus mechanism known as *quorum sensing*.
- C-element combines a noisy environmental trigger signal and a density dependent quorum sensing signal.
- Activated bacteria signal their neighbors to reach consensus.




Winstead et al., IBE Conference (2008)

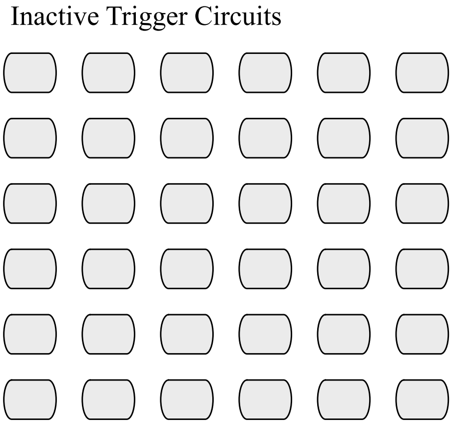
# Confidence Amplifier

- A noisy C-element with a confidence-feedback loop:

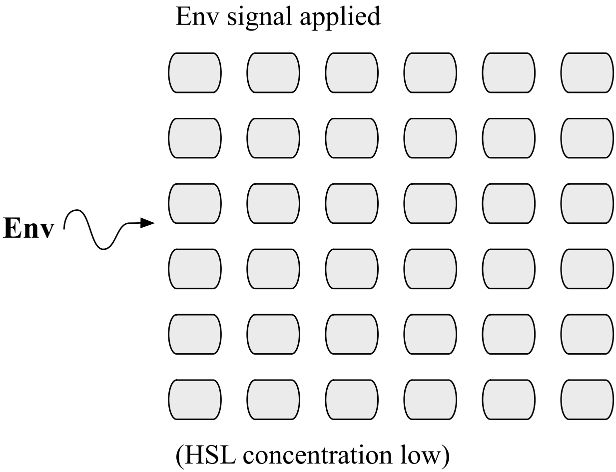



- The output “rails” to maximum confidence, even if *E* has low confidence.
- This configuration only works if the C-element is “noisy”. Otherwise, the circuit is permanently stuck in its initial state.

# Quorum Trigger Population Dynamics

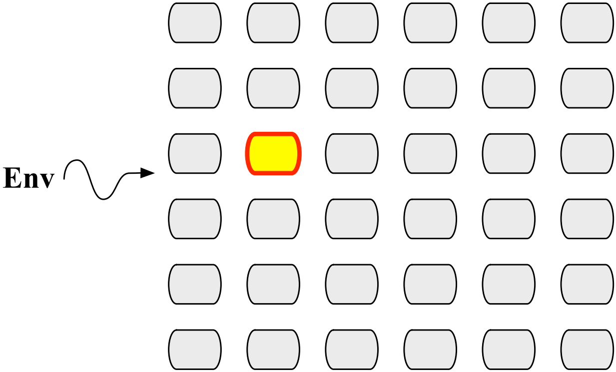


# Quorum Trigger Population Dynamics



# Quorum Trigger Population Dynamics

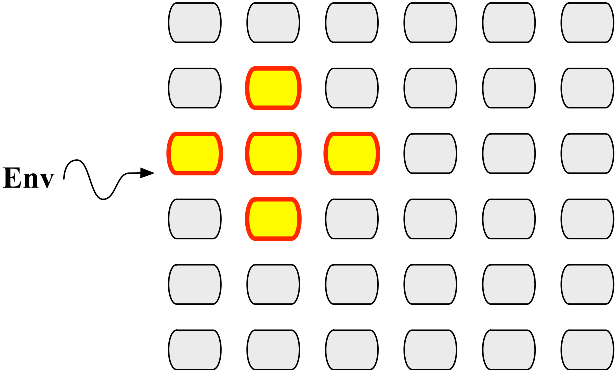
One circuit randomly activates




(HSL concentration increases)

# Quorum Trigger Population Dynamics

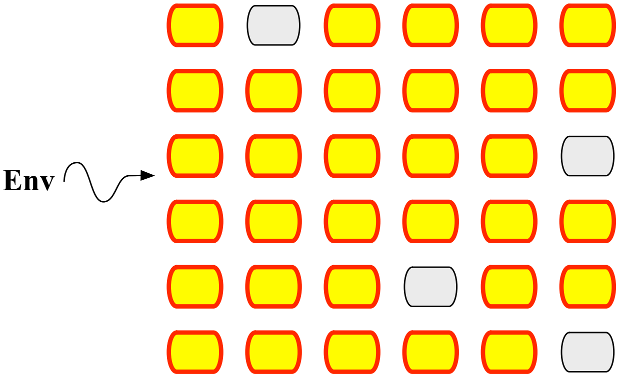
More circuits activate due to HSL




(HSL concentration increases sharply)

# Quorum Trigger Population Dynamics

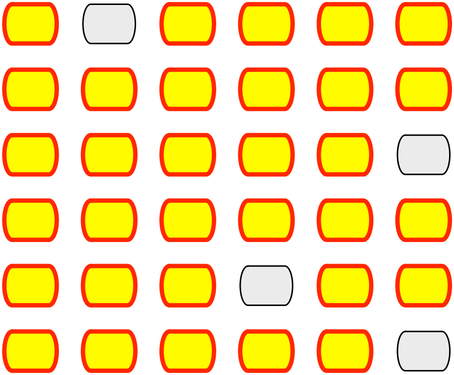
Avalanche effect: most cells activate




(HSL concentration saturates)

# Quorum Trigger Population Dynamics

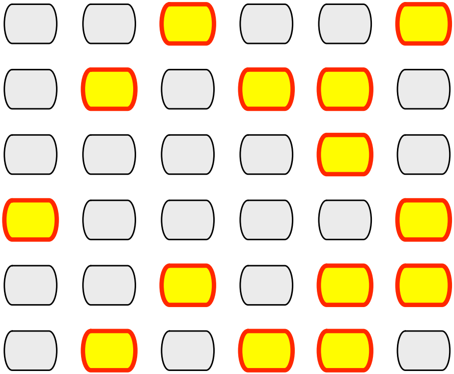
Env signal is removed.



(Circuits stay active)

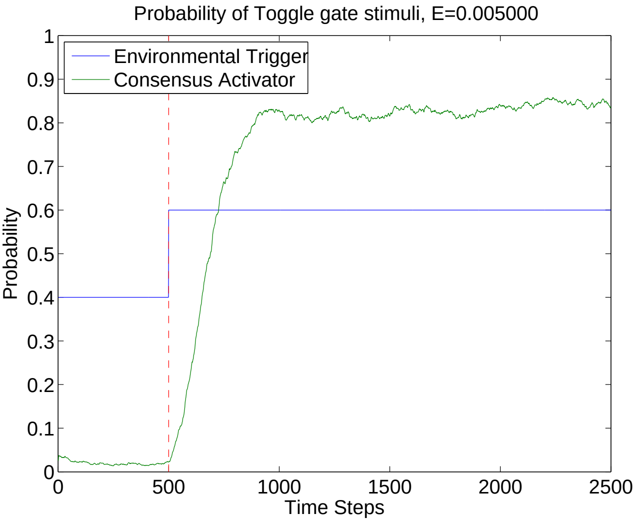
# Quorum Trigger Population Dynamics

Time passes.

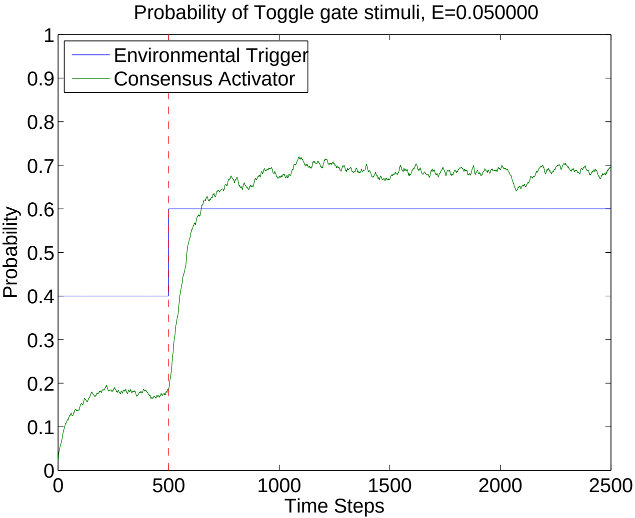


(Circuits randomly switch off)

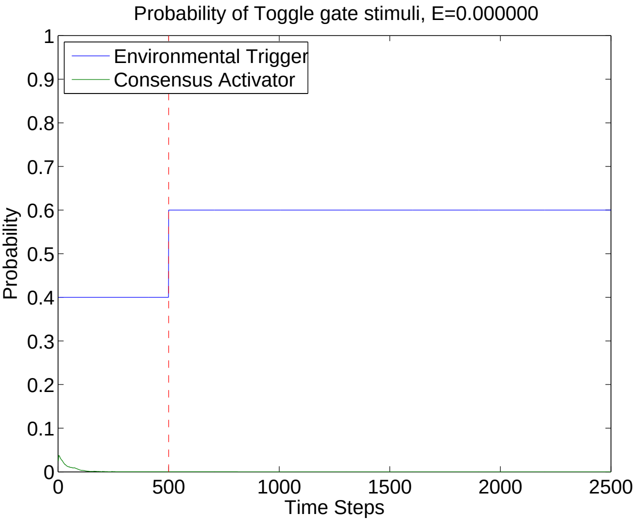
# Quorum Trigger Simulation Results



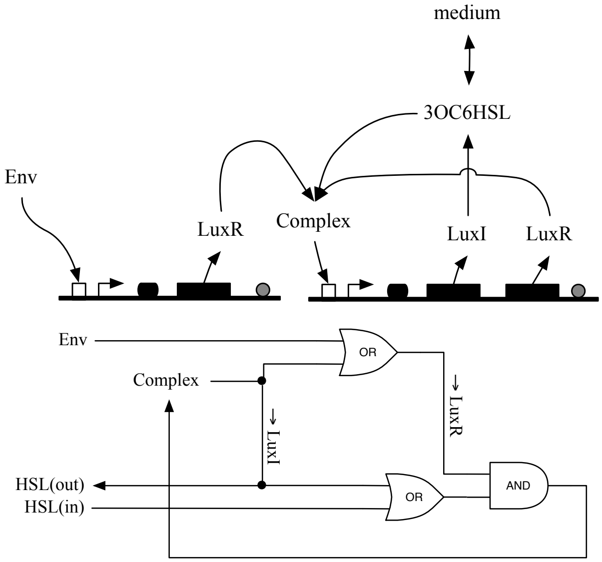
# Quorum Trigger Simulation Results



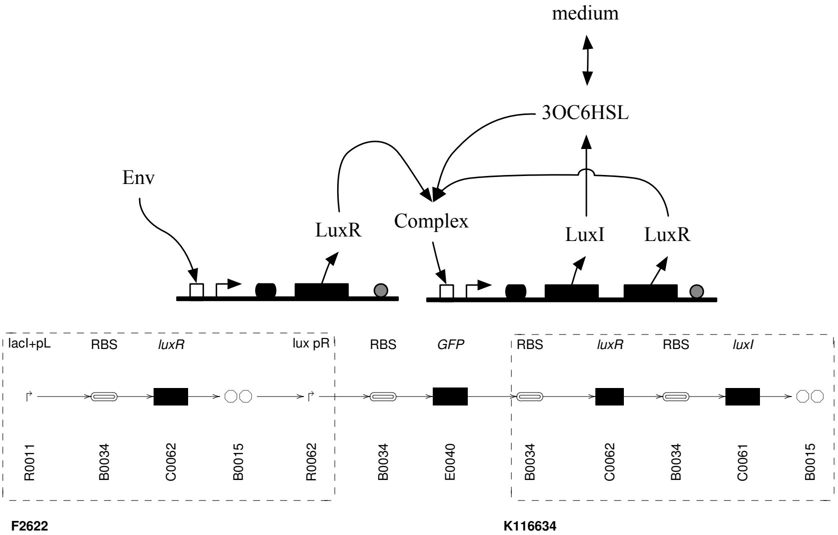
# Quorum Trigger Simulation Results



# Quorum Trigger Design



# Quorum Trigger Design



# Quorum Trigger Design

# Future GDA Research Directions

- Genetic circuits have no signal isolation.
- Circuit products may interfere with each other and host cell.
- Gates in a genetic circuit library usually can only be used once.
- Behavior of circuits are non-deterministic in nature.
- No global clock, so timing is difficult to characterize.
- To address these challenges, we are investigating soft logic models based on *factor graphs* and adapting asynchronous synthesis tools to a genetic circuit technology.

# Biologically Inspired Circuit Design

- Human inner ear performs the equivalent of one billion floating point operations per second and consumes only 14 $\mu$W while a game console with similar performance burns about 50 W (Sarpeshkar, 2006).
- We believe this difference is due to over designing components in order to achieve an extremely low probability of failure in every device.
- Future silicon and nano-devices will be much less reliable.
- For Moore's law to continue, future design methods should support the design of reliable systems using unreliable components.
- Biological systems constructed from very noisy and unreliable devices.
- GDA tools may be useful for future integrated circuit technologies.

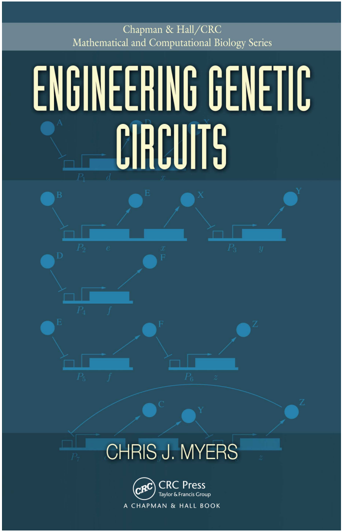
# Adam Arkin

*Since the engineering principles by which such circuitry is constructed in cells comprise a super-set of that used in electrical engineering, it is, in turn, possible that we will learn more about how to design asynchronous, robust electronic circuitry as well.*

# More Information

- Linux/Windows/Mac versions of `iBioSim` are freely available from:
  `http://www.async.ece.utah.edu/iBioSim/`
- Publications:
  `http://www.async.ece.utah.edu/publications/`
- Course materials:
  `http://www.async.ece.utah.edu/∼myers/ece6760/`
  `http://www.async.ece.utah.edu/∼myers/math6790/`

# Engineering Genetic Circuits



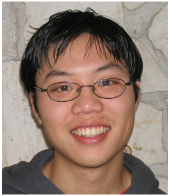
# Acknowledgments

Nathan Barker

Keven Jones

Hiroyuki Kuwahara

Curtis Madsen



Nam Nguyen

Chris Winstead

This work is supported by the National Science Foundation under Grants No. 0331270, CCF-07377655, and CCF-0916042.