AI for Synthetic Biology

Fusun Yaman, Aaron Adler
fusun, aadler @bbn.com
BBN Technologies, Cambridge, MA

July 9, 2016
The Vision – Programming Biology Like Computers

In the 1950s, computers were programmed by flipping switches and moving cables...

Synthetic Biology is at that point today...

• Manual creation of simple “parts”
• Tedious, ad hoc creation of simple systems

• Many opportunities in Design – Build – Test loop to improve process
• Goal: high-level program to functioning cells

Many iterations

>1 year/system

Simple systems

If detect explosives: emit signal
If signal > threshold: glow red

Many

iterations

Simple systems

>1 year/system
Why is this Important?

- DNA synthesis has been growing exponentially, but circuit (behavior) complexity has plateaued:

![Graph showing the increase in DNA synthesis and circuit size over the years.](image)

**DNA synthesis**
- 207 base pairs in 1975
- 2,100 base pairs in 1985
- 2,700 base pairs in 1990
- 7,500 base pairs in 2000
- 14,600 base pairs in 2010
- 1,080,000 base pairs in 2012

**Circuit size**
- 11 promoters in 2012

Designing complex circuits is hard!

*Sampling of systems in publications with experimental circuits*
How to break the complexity barrier?

Organism Level Description

Cells
Answer: Tool Chain

Collaborators:
- Ron Weiss
- Douglas Densmore

Desperately Needs AI

Collaborators: Ron Weiss, Douglas Densmore

Synthetic Biology Workflow

Design

Map behavior specification to nucleic acid sequence(s)

Test

Transfect/transform cells, assay behavior

Build

Synthesize/assemble nucleic acid sequence(s)
Knowledge Representation (KR)

• **Why:**
  – Capture details about system designs for reproducibility
  – Curate useful part databases
    • Qualitative and quantitative information enabling automation
  – Data exchange formats for collaboration

• **Challenges**
  – What is the minimum information required?
  – Detailed protocols descriptions required to reproduce results
  – Comparable measurement units and techniques are just emerging
  – Results are stochastic and vary with time
  – Many biological processes poorly understood
Solutions: SBOL

- Synthetic Biology Open Language (SBOL), standard describing genetic parts, devices, modules, systems
  - Community effort much like W3C
  - Ontology based approach
  - Defines a standard visualization for designs
  - Slowly gaining recognition and acceptance

- ACS Synthetic Biology adopt SBOL for depiction and representation of genetic construct for recording and sharing
Solutions: Part Characterization

TASBE characterization

- Target system for characterization
- Fluorescent color experiments
- FACS data
- Cell autofluorescence compensation
- Fluorescent color selection
- Fluorescent color compensation
- Segmentation and subgroup statistics
- Normalization to per-plasmid behavior

Transfer Curve

- Normalized Dox transfer curve, colored by plasmid bin

Cell autofluorescence compensation

- IFP MEFL/plasmid Normalized Dox transfer curve, colored by plasmid bin

Details of these steps can be found in Section 2.

The statistical distribution of single-cell output levels for each input level, in order to estimate the variability of behavior.

The reason for these requirements is that much of the interaction between parts takes place within individual cells, rather than averaged across the whole population (except for special case systems involving intercellular communication). Thus, we need to measure the behavior of single cells. Cells exhibit a high degree of behavioral variability within a population, so to control for variability we acquire a large number of single-cell measurements.

Finally, when evaluating DNA parts in a digital logic context, it is typically required for parts to have an approximately sigmoidal response with well-defined high and low input and output signal levels. Different DNA parts have different output signal levels and make their high-to-low transitions at different input signal levels. Thus predictable composition requires knowledge of the full input/output relation, particularly when working with parts whose transition between low and high expression has a slope that is not very steep.

Additionally we need to know the per-copy (per-plasmid) behavior of the constructs in order to predict overall signal levels, since the number of copies of a system in a cell can vary greatly both by design (e.g., multiple copies, plasmids with different mean copy numbers) and through natural cell dynamics (e.g., copy number variation, transient transfection).
Knowledge-Based Systems

- **Why**
  - Design is knowledge intensive
  - Understand what might be going wrong
  - What to do and how to correct it

- **Challenges**
  - Greater precision required for computation
  - Managing complexity
Solutions

- Formal grammars (GenoCad) for verification
  - Is this a valid DNA sequence?
- Black listed bad sequences (Eugene)
- BioCompiler: design motifs

GenoCAD CFG [Cai et al., ‘07]
Motif-Based compilation where operators are translated to motifs:

Solutions: Bio Compiler
Complex Example: 4-bit Counter

Optimized compiler already outperforms human designers
Constraints and Satisfiability

• Why
  – Transforming high-level organism descriptions to DNA sequences involves solving several constraint satisfaction problems
    • Which parts to use
      – Parts are not compatible with each other, interact and interfere
    • Which method to use
      – Better noise reduction at the cost of higher metabolosmic load

• Challenges
  – The data is missing or incomplete
  – Formulation of biological requirements as constraints
  – Domain is complex: identify necessary and sufficient conditions
Solutions: MatchMaker

How can we map the abstract parts in an AGRN to real parts?

Feature Mapping: Satisfy the constraints on the edges of the AGRN

Signal Matching: Pick parts that are signal compatible accounting for noise and preserving digital behavior

Design

Feature Database

AGRN
Transcription Factors
Promoters

Find Match

Dox
rtTA
CFP
EYFP

Dox
rtTA
CFP
lacI
pHef1a-LacO1Oid

pHef1a
pTre
pTre

Transcription Factors
Promoters

[Y]
[X]

[Y]
[X]

[Y]
[X]

[Y]
[X]

[y]
[x]

[z]
[x]

[z]
[x]

[z]
[x]

[z]
[x]

[z]
[x]

[z]
[x]

[z]
[x]
Optimization and Heuristic Search

• Why:
  – Most of the constraint satisfaction problems at the design stage are at least NP-Hard.
    • Feature matching, Signal Matching, Part Matching
  – Optimization is necessary at every stage
    • To manage complexity and the viability of the design
    • To make the most of shared resources that degrade over time: e.g., chemicals, DNA

• Challenges:
  – Objective functions for optimization can be very complex
  – Domain specific heuristics require a lot of biology expertise
Solutions

• Assembly Planner: Optimize BioBrick assembly across multiple designs (Densmore)
  – Determines subpart sharing across goal parts.
  – Utilize three different heuristics to find an assembly sequence that minimize assembly steps while maximizing the sharing of intermediate products.
Machine Learning

• **Why:**
  – Wide range of applications need small synthetic biology circuit to classify cell state.
    • Infected or not?
  – Learning from experience – success as well as failure
    • Currently just eyeballing Excel spreadsheets

• **Challenges:**
  – Big data
  – Incomplete and noisy data
  – Synthetic circuits might not be viable
Solutions

- Goal: Logic formula to identify HeLa cancer cells
- Sensors: Micro RNA

\[
\text{HeLa} = \text{miR-1H} \land (\text{miR-2H} + \text{miR-3H}) \land \\
\sim\text{miR-4L} \land \sim\text{miR-5L} \land \sim\text{miR-6L}
\]

- Supervised learning and information-based approach to design of cell-state detectors (Beal & Yaman 2012)
  - Which of the 600 markers are more informative?
  - What is the threshold?
- Identified a smaller set of markers
- No false positives & moderate false negatives
Planning and Scheduling

• Why
  – Complex dependencies with temporal and resource constraints, non-deterministic outcomes at assembly and test
  – Effective use of shared wet-lab, shared supplies, shared equipment
  – Increasing precision of executed protocols

• Challenges:
  – Reliance on existing paper lab notebooks, software
  – Closed source lab management software
  – Complex scheduling including living cells that must be maintained and measured at certain times
Solutions

- **Puppeteer (Vasilev et al. 2011)**
  - Suite of tools for defining
    - robot specific resources
    - robot specific protocols
    - producing executable scripts.

1. **Assignment to wells**
   - Stock Plate
   - Dilution Plate
   - Reaction Plate

   - ab + cd
     - ab
     - b
     - cd
     - d

2. **Dilute with buffer**
   - ab cd

3. **Combine assembly step**
Reasoning under Uncertainty

• Why
  – Many biological processes are inherently stochastic and not synchronized
  – Still many unknowns and noise in the data/models

• Challenges:
  – Some data collection methods destroy the cells
    • Can’t be used for data collection at later time points
  – Different trials of the same experiment may have different results
  – Modeling enough to guarantee predictability
Solutions

- **Empirical Quantitative Incremental Prediction (EQuIP)**
  - accurate prediction of genetic regulatory network behavior from detailed characterizations of their components. (Davidsohn 2015)
Multi-Agent Systems

Why

- Each cell can be viewed as an “agent”
  - Cells can communicate using small molecules
- Complex applications rely on interaction between cells
  - Tissue differentiation, multi-cellular organisms

Challenges

- Communication rate and reaction time are slow
- Need to model cell-to-cell communication for complex aggregate behavior
- Test and validation of differentiated cells
Spatial Computing/Aggregate programming
– Cells are seen as spatial computers: Executing the same program in reaction to the local sensor information.

```lisp
(def band-detector (signal lo hi)
  (and (> signal lo) (< signal hi)))
(let ((v (diffuse (aTc) 0.8 0.05)))
  (green (band-detect v 0.2 1)))
```

High-Level Bio-Focused Language

Genetic Regulatory Network

[Beal & Bachrach, '08]

[Weiss]
Robotics

• Why
  – Complex assembly techniques, error prone
    • “Magic touch” by some practitioners
  – Schedule flexibility and continuous availability
  – Increase consistency of results for repetitive, error-prone, time-sensitive procedures

• Challenges:
  – Sampling cells and pipetting small volumes
  – Effective human-machine collaborations
  – Assembly techniques that will require more automation
Solutions

• Automated Assembly (Densmore Lab)

Assembly Planning Tools

Robot

Build

Biomek 3000
• 1-way and 8-way pipette tools
• gripper to move plates to magnet
• heat block, shaker and -20C on deck
Summary & Conclusions

- Synthetic biology has exciting applications:
  - curing cancer, diabetes, neglected diseases
  - environmental remediation
  - biofuels, nanofabrication
- Current techniques and tools are very limited
- AI has many techniques that can be applied to Synthetic Biology
- Workshop Goal: Build connections between the AI community and the Synthetic Biology community.

Hopefully the first in a series of workshops!