Towards automated selection of parts for genetic regulatory networks

(BBN) F. Yaman, A. Adler, J. Beal
(BU) S. Bhatia, D. Densmore
(MIT) R. Weiss, N. Davidsohn

Work sponsored by DARPA I2O under contract HR0011-10-C-0168; the views and conclusions contained in this document are those of the authors and not DARPA or the U.S. Government.
Designing Transcriptional Networks

Genetic Regulatory Network (GRN)
- Specific design
- Ready to assemble

Behavior description
- High-level design
- Must be mapped to biology

if (sense(Dox))
then fluoresce(cyan)
else fluoresce(yellow)
Abstract Genetic Regulatory Networks

Genetic Regulatory Network (GRN)
- Specific design
- Ready to assemble

Abstract Genetic Regulatory Network (AGRN)
- Template representing multiple GRNs
- Contains necessary constraints

Behavior description
- High-level design
- Must be mapped to biology

\[
\text{if (sense(Dox))}
\]
\[
\text{then fluoresce(cyan)}
\]
\[
\text{else fluoresce(yellow)}
\]
From AGRNs to GRNs

Genetic Regulatory Network (GRN)
- Specific design
- Ready to assemble

Abstract Genetic Regulatory Network (AGRN)
- Template representing multiple GRNs
- Contains necessary constraints

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

**Solution:** Feature Mapping + Signal Matching
Feature Mapping

• **Feature**: a DNA sequence responsible for a specific biochemical behavior, e.g., promoter

• **Feature database**: a collection of features with the regulatory relationships between them

![Diagram]

- TetR → pTet
- rtTA → pTRE

• **Feature mapping**: assign features from the database to the variables in AGRN
  - The mapping should satisfy all edges in AGRN and not entail additional interactions.
  - Given an AGRN G and a feature database H, find a network of promoters and transcription factors in H that is isomorphic to G.
Feature Mapping: assign features to variables

- **Feature mapping**: Given bipartite graphs $G$ and $H$, find a subgraph of $H$ that is strictly isomorphic to $G$. 

AGRN

<table>
<thead>
<tr>
<th>Transcription Factors</th>
<th>Promoters</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_0$</td>
<td>$P_0$</td>
</tr>
<tr>
<td>$a_1$</td>
<td>$P_1$</td>
</tr>
<tr>
<td>$x$</td>
<td>$P_2$</td>
</tr>
<tr>
<td>$y$</td>
<td></td>
</tr>
</tbody>
</table>

Feature Database

<table>
<thead>
<tr>
<th>Transcription Factors</th>
<th>Promoters</th>
</tr>
</thead>
<tbody>
<tr>
<td>$q_0$</td>
<td>$p_0$</td>
</tr>
<tr>
<td>$q_1$</td>
<td>$p_1$</td>
</tr>
<tr>
<td>$q_2$</td>
<td>$p_2$</td>
</tr>
<tr>
<td>$q_3$</td>
<td>$p_3$</td>
</tr>
<tr>
<td>$q_4$</td>
<td>$p_4$</td>
</tr>
<tr>
<td>$q_5$</td>
<td>$p_5$</td>
</tr>
</tbody>
</table>
Feature Mapping: assign features to variables

**AGRN**

- **Transcription Factors**
  - $a_0$
  - $a_1$
  - $x$
  - $y$

- **Promoters**
  - $P_0$
  - $P_1$
  - $P_2$

---

**Feature Database**

- **Transcription Factors**
  - $q_0$
  - $q_1$
  - $q_2$
  - $q_3$
  - $q_4$

- **Promoters**
  - $p_0$
  - $p_1$
  - $p_2$
  - $p_3$
  - $p_4$
  - $p_5$
Feature Mapping: assign features to variables

AGRN

Transcription Factors      Promoters

a₀ → P₀
a₁ → P₁
x → P₂
y

Subgraph of Feature Database

Promoters      Transcription Factors

q₀ → p₁
q₁
q₂ → p₃
q₄ → p₄
Feature Mapping: assign features to variables

- This problem is NP-complete: There is no fast algorithm for solving every instance of this problem*.
Solution: Heuristc guided search

Solve (AGRN, FDB, Assignments)
   Pick a node v in AGRN that is not in Assignments
   If no such node exists
      Return Assignments
   For every node f in FDB do
      NewAssignments = Assignments + (v,f)
      result =Solve (AGRN, FDB, NewAssignments )
      If result is not FAIL
         Return result
   Return FAIL;
Solution: Heuristic guided search

Solve (AGRN, FDB, Assignments)
Pick a node \( v \) in AGRN that is not in Assignments
If no such node exists
    Return Assignments
For every node \( f \) in FDB do
    NewAssignments = Assignments + (v,f)
    result =Solve (AGRN, FDB, NewAssignments )
    If result is not FAIL
        Return result
Return FAIL;

Same type as \( v \)
Have at least same or more in/out degree as \( v \)
Can support every edge of \( v \)
Solution: Heuristic guided search

Rank the nodes: small domain --> higher rank
Pick the highest ranking node v

Solve (AGRN, FDB, Assignments)
Pick a node v in AGRN that is not in Assignments
If no such node exists
    Return Assignments
For every node f in FDB do
    NewAssignments = Assignments + (v,f)
    result = Solve (AGRN, FDB, NewAssignments)
    If result is not FAIL
        Return result
Return FAIL;
Solution: Heuristic guided search

Solve (AGRN, FDB, Assignments)
    Pick a node $v$ in AGRN that is not in Assignments
    If no such node exists
        Return Assignments
    For every node $f$ in FDB do
        NewAssignments = Assignments + ($v$, $f$)
        result =Solve (AGRN, FDB, NewAssignments)
        If result is not FAIL
            Return result
    Return FAIL;

Rank nodes $f$; (appears in more variable domains --> higher rank)
Try the nodes in ranking order
Feature Mapping produces a GRN

AGRN
- rtTA
- CFP
- EYFP

GRN
- rtTA
- pHef1a
- CFP
- Lacl
- EYFP
- pHef1a-LacO1Oid

Feature Mapping

Dox
Signal Matching

- Feature mapping ensures the pair wise logical relationships but there is also signal ranges to consider.
Signal Matching

• Composition should preserve digital behavior

Ideally…
Signal Matching

- Composition should preserve digital behavior

**Signal Matching Problem:** How do we pick the parts that have compatible interpretations for on/off so that when composed will preserve digital behavior?
Solution

- Pick the parts that are **signal compatible**
  - operate in same signal range where Signal = Concentration

- Parts are signal compatible iff noisy output range is contained in valid input range
Feature Mapping + Signal Matching

• First do a feature mapping
  – Convert AGRN to GRN

• Check signal compatibility for every pair of bio-device in the GRN
  – If fails go back to previous step, find another GRN
Implementation: MatchMaker

www.clothocad.org

“MatchMaker” App
tasbe-team@bbn.com
Empirical Results

- Experiments with random feature graphs (up to 200 nodes) and AGRNs (up to 60 nodes)

When there is a solution we find it fast, suggesting timeouts might work well.
Conclusions

• Contribution
  – Fill in the big gap for going from AGRNs to GRNs

• Future plans and on going work
  – Hierarchical feature mapping
    • Search over families of features instead of individuals
  – Finding the most noise-tolerant network
    • Greedy search over signal data
Questions?
BackUp Slides
Abstract GRN to Sequence of Parts

Step 1: Feature Matching

Feature DB

A
B
C
D
E
F
G
H
I
K

Regulating Proteins

Promoters

Y_1
Y_2

OR

Problem: Find a set of nodes in the DB that is isomorphic to input.
Challenge: NP-Complete

Step 2: Signal Matching

Characterization DB

A
B
C
D
E
F
G
H
I
K

GRN

G--A--K--E
K--E--G--A

Problem: Verify that components operate in the same range and combination is noise resilient.
Challenge: Getting & interpreting data.

Step 3: Parts Matching

Parts DB

G--A--K--E
K--E--G--A

Problem: Find minimum number of parts to implement a linearization of the network.
Challenge: NP-Complete

BEST!

Input to the Assembly Manager (next step in the tool-chain)
Linearization

- Features in the A-GRN are loosely ordered
  - Y1 should be next to X1
  - Y2 should be next to X2
- Intuitively any total order that will satisfy these orderings is equivalently good.
  - Trivial solution: Linear time algorithm
  - [X₁, Y₁, X₂, Y₂] or [X₂, Y₂, X₁, Y₁]
- Other design constraints may eventually affect orderings
Part Mapping

- **Basic**: Given a sequence of features, what are the parts that can cover the sequence?
- **Enhanced**: Optimization problem
  - Minimize the number of parts used
  - Maximize the use of existing samples
- **Current implementation** addresses basic problem
  - Greedy search with preference on larger parts.
Next Steps

- Implementation of Families in Clotho to support hierarchical feature matching
- Multidimensional signal matching
- Optimal parts matching algorithm
Truth Table for a Hybrid Promoter?

• Just seeing this symbol does not tell us what the intended behavior is
• Most of the time behavior depend on the type of the promoter and it should be part of the notation

<table>
<thead>
<tr>
<th>R</th>
<th>P_OUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

• Implicit Assumption: P is constitutently high

<table>
<thead>
<tr>
<th>R</th>
<th>P_OUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

• Implicit Assumption: P is constitutently low

<table>
<thead>
<tr>
<th>R</th>
<th>P_OUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

tasbe-team@bbn.com
Truth Table for a Hybrid Promoter?

<table>
<thead>
<tr>
<th>R</th>
<th>I</th>
<th>P_OUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>R</th>
<th>I</th>
<th>P_OUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Proposed Hierarchy of Promoters

- Promoters
  - High
    - Repressible
      - Semi-Repressible
    - Inducible
      - Semi-Inducible
  - Low