Notes on a Bio-automaton

Tao / Bhatia

1 Introduction

We propose an algorithm for generating a plasmid design that implements a given deterministic acyclic finite automaton using cellular transcriptional machinery. We propose a composable primitive, which we call the τ-module, to implement our design.

Definition 1. *A τ-module(α, i, j, S) is a segment of DNA comprising the following parts in the following order:*

1. *A strong bidirectional terminator;*
2. *A promoter induced by α flanked by co-oriented recombinase i recognition sites;*
3. *A strong bidirectional terminator flanked by co-oriented recombinase j recognition sites, where $i \neq j$;*
4. *A set S of sequences coding for recombinases; and*
5. *A strong bidirectional terminator.*

Definition 2. *A deterministic acyclic finite automaton is a tuple $\mathcal{A} = (Q, \Sigma, \delta, s, F)$, where Q is a finite nonempty set of states, Σ is a finite nonempty set of symbols, $\delta : Q \times \Sigma \to Q$ is a transition function, s is the initial state, and F is the set of final states.*

For our bio-automaton, we interpret Σ to be a set of inducers and F to be the set of states in which a reporter is produced. The various states of this bio-automaton can be visually represented with an acyclic graph in which states are represented by vertices and inducers are represented by arcs connecting two vertices. A bio-automaton described by Definition 2 can be constructed out of n τ-modules where n corresponds to the number of of arcs in the graphs. With two exceptions, a bio-automaton can be constructed out of n τ-modules.

1. Promoters 'active' in the initial state of the promoter should not be followed by a strong bidirectional terminator flanked by co-oriented matching recombinase recognition sites. The term active describes a promoter that will allow for the translation of a meaningful product before encountering a terminator.
2. A τ-module corresponding to a final state need not contain recombinase coding regions. Recombinase coding regions may be substituted with reporter gene coding regions. A final state refers to any state that corresponds to the last vertice encountered in a depth first traversal of the graph corresponding the bio-automaton described here.

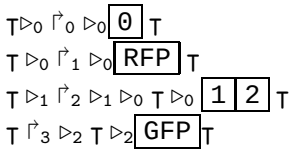
Algorithm 1 BIO-AUTOMATON GENERATOR

Require: Deterministic acyclic finite automaton $\mathcal{A} = (Q, \Sigma, \delta, s, F)$

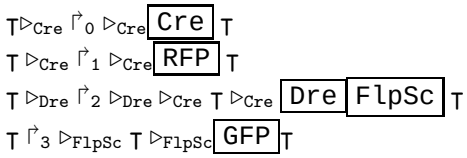
$r \leftarrow 0$, $i \leftarrow 0$
$A \leftarrow \epsilon$
for each state s in Q **do**
  for each arc $e_i = (s, y, \alpha)$ **do**
    $p_i \leftarrow r$, $t_i \leftarrow r + 1$
    $C_i \leftarrow C_i \cup p_i$
    $r \leftarrow r + 2$
    $T_i \leftarrow \tau$-*module*(α, p_i, t_i, C_i)
  end for
  for each arc $e_i = (s, y, \alpha)$ **do**
    for each arc $e_j = (s, w, \alpha) \neq e_i$ **do**
      $C_i = C_i \cup \{j\}$
    end for
  end for
  for each pair of arcs $e_i = (x, s, \alpha)$ and $e_j = (s, y, \beta)$ **do**
    $C_i \leftarrow C_i \cup \{t_j\}$
  end for
end for
Output $T_1 \cdot T_2 \cdot \cdots \cdot T_{|\delta|}$

2 A Proof of Concept

The translation of abstract computer science to physical constructs composed of nucleic acids is an exciting capability that may allow synthetic biologists to leverage the extensive work of computer scientists. However, the bio-automatons proposed here remain abstract and no more than a fanciful dream unless sufficiently grounded by a proof of concept construct. Such a proof of concept is described below using symbology given in Table 1.



The numbers in this design can be substituted for actual recombinase and promoters of course. All producers in the design is an inducible promoter. Given that inducible promoters are better characterized than recombinase and that there is a large variety of promoters, specific promoter names will not be provided below. The following design replaces the numerical subscripts for recombinase recognition sites and recombinase coding regions with the actual recombinase. The recombinases used in the design are Cre, Dre, and FlpSc. Each of these recombinase will recognize a pair of recognition sites, and then excise the content in between the sites; one recognition site is removed in the process. Dre and FlpSc can recognize one pair of sites each; Cre and recognize as many as three.



Symbol	Meaning
$\vec{\Gamma}_0$	Promoter
T	Terminator
$\triangleleft_0, \triangleright_0$	Recombinase recognition site
0	Recombinase coding region
GFP	Gene coding region

Table 1: Table 1- Bio-automaton design symbology

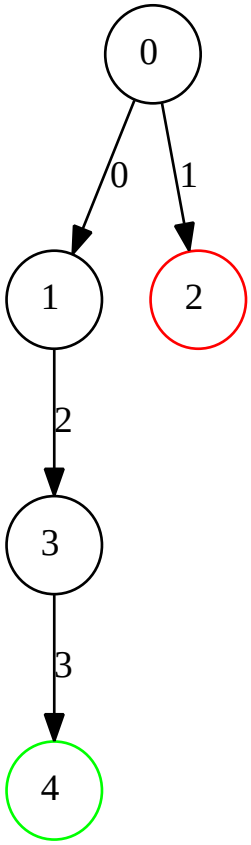


Figure 1: The bioautomaton described by the proof of concept construct is described by the following graph. States are represented by vertices and each arc corresponds to an inducible promoter

The construct will demonstrate the following:

1. The design pattern, whereby a state machine is refactored as a set of τ-modules, is feasible, and has behavior that corresponds to the automaton described in Definition 1.
2. The design pattern is potentially scalable given sufficient components such as recombinase and inducible promoters.
3. The τ-module offers clear advantages over the Single Invertase Memory Module created by Collins et al. These advantages are detailed in the advantages section.
4. The various states of the bio-automaton can be distinguished from one another by observing responses from various reporter genes.

3 Positive Points

1. Arrangement of modules is trivial. Any permutation of the constituent modules of a design will give a bio-automaton with behavior equivalent to any other permutation; this may be leveraged to produce constructs that reside on multiple plasmids.
2. Provides a clear way to produce excusive arcs. Suppose that several arcs branch out from a node in the state machine graph. An exclusive arc should render it impossible to traverse the other arcs from the given node.
3. Allows for the repeat use of promoters while still allowing for distinct final words; a final word is a set of inputs that result in a reporter response
4. Easily allows for branching designs
5. Supposing only full words need to be recognized, intermediate states may share recombinase, allowing for didesigns that use fewer types of recombinase

Assuming that constructs built using paradigms outlined here possess physical behavior akin to the behavior asserted here, then τ-module can be a boon to synthetic biology. Our software posesses capability to translate a directed graph to a bio-automaton design. Each arc is translated individually without information beyond which arcs intersect with a given arc. The minimal use of contextual information during each arc translation indicates that each operation is essentially, independent. Use of the τ-module offers composability. Composability is a valuable characteristic to automated work flows. If the software that generates the bio-automaton can perform independent optimization operations on each module, then the software can deliver scalable, complex designs, which are difficult to generate by hand. Such tools and design architectures will be common in the next generation of synthetic biological work. No longer will sofware merely mimic human chores, but software tools will enhance and further human aims and efforts.

4 Optimization

The composable nature of designs using the τ-module described here offers the opportunity to perform optimizations on a design on an arc to arc basis; each arc corresponds to one τ-module. Such an optimization would examine each module iteratively and perform the appropriate optimizations. The only necessary contextual would be the module itself, and its immediate neighbors. Such an optimization method would scale easily with any design, regardless of the complexity.

The first of these optimizations involve a merging of modules that represent a set of forwardly-oriented arcs diverging from one common vertex. Each τ-module diverging from this node uses one recombinase to excise its own promoter to prevent excessive translation of recombinase; another recombinase is used to remove the promoter of each τ-module that diverge from the aforementioned

common vertex. The use of a recombinase in one τ-module removing a promoter in another τ-module allows for the exclusive paths when traversing the graph corresponding to our bioautomaton. Such a use of recombinase is nontrivial, and can become expensive in terms of recombinase use in a bioautomaton with widely branching states. Examine the following graph, which contains just one fork.

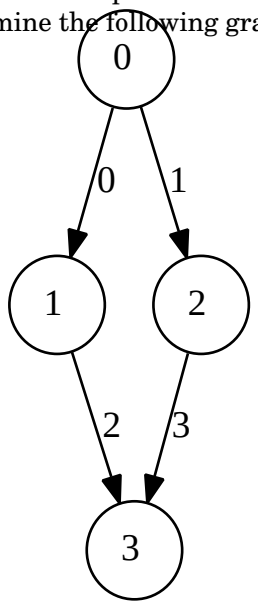


Figure 2: A graph describing an automaton that has two states that fork from the initial state

The unoptimized construct given by the algorithm described in the introduction is as follows:

T $\triangleright_0$ $\triangleright_4$ $\overset{\curvearrowright}{\Gamma}_0$ $\triangleright_4$ $\triangleright_0$ [0][1][2] T

T $\triangleright_1$$\triangleright_3$ $\overset{\curvearrowright}{\Gamma}_1$ $\triangleright_3$ $\triangleright_1$ [3][4][5] T

T $\overset{\curvearrowright}{\Gamma}_2$ $\triangleright_2$ T $\triangleright_2$ [R0] T

T $\overset{\curvearrowright}{\Gamma}_3$ $\triangleright_5$ T $\triangleright_5$ [R1] T

The design uses a total of 6 different recombinase, assuming that recombinases use can recognize only one set of recognition sites.

The simple, unoptimized designs can be improved upon by 'merging' several τ-modules together. Each of the merged modules must diverge from the same vertice and share the same orientation; we will cause these modules 'neighbors'. Each of these modules contains a recombinase that removes the modules own promoter, and another recombinase that excises the promoter of τ-modules that also diverge from the same state. In the particular case of several arcs that diverge from the same vertice, the two recombinase described, are redundant. By enclosing a module and each of its neighbors in the recognition sites that would be used to cut out the module's own promoter, no additional recombinase need to be used to excise the promoter's of neighbors. Instead neighboring modules are excised in their entirety, producing the excusive paths described previously. The reduced use of recombinase should scale exponentially with the number of τ modules with neighbors, and the number of neighbors each of these modules has. One recombinase for each neighbor is used in the unoptimized design to remove the promoters of neighbors. And so supposing there are n modules with neighbors.... ADD MATH TO DESCRIBE HOW MUCH IS SAVED

$\triangleright_0$$\triangleright_3$ $\overset{\curvearrowright}{\Gamma}_1$ $\triangleright_3$ $\triangleright_5$ T $\overset{\curvearrowright}{\Gamma}_0$ $\triangleright_0$ $\triangleright_2$ T $\triangleright_3$ $\triangleright_0$

T $\overset{\curvearrowright}{\Gamma}_2$ $\triangleright_2$ T $\triangleright_2$ [R0] T

T $\overset{\curvearrowright}{\Gamma}_3$ $\triangleright_5$ T $\triangleright_5$ [R1] T

5 Future Work

Here are a few items to pursue in the future:

1. Construction and characterization of a physical sample
2. Generation of permutations of parts.
3. Control module for two cells that talk to each other.
4. DNA security and authentication
5. NOR gates and other logic gates