Lecture 8

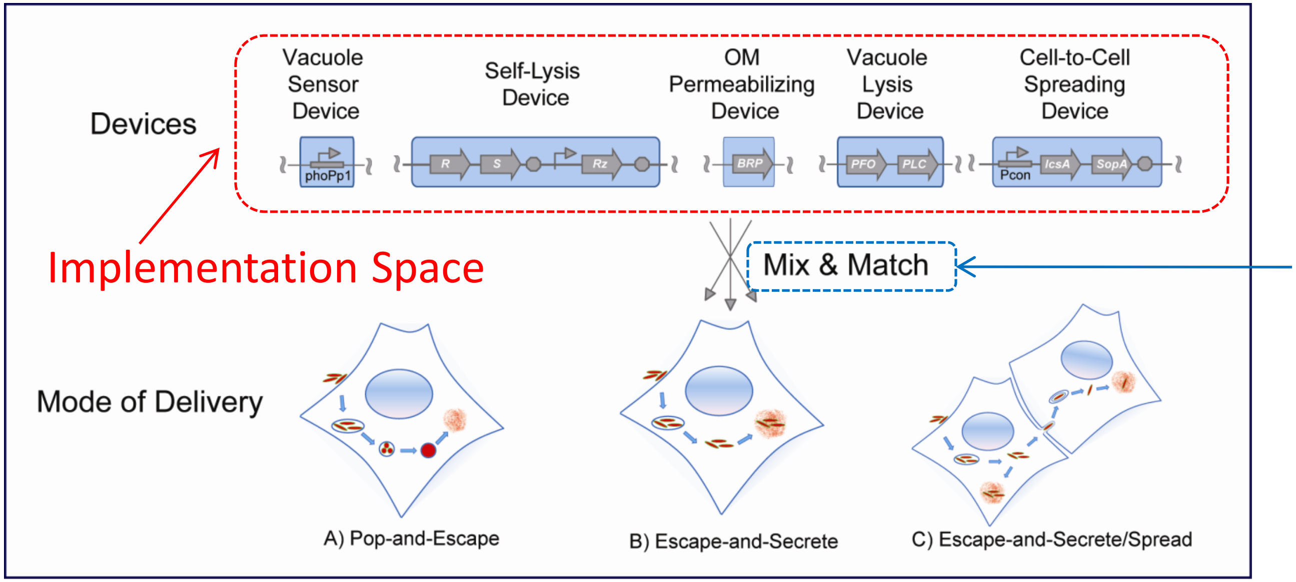
Design – Overview, Approaches, Challenges

Prof. Douglas Densmore

EC/BE552

Computational Synthetic Biology for Engineers

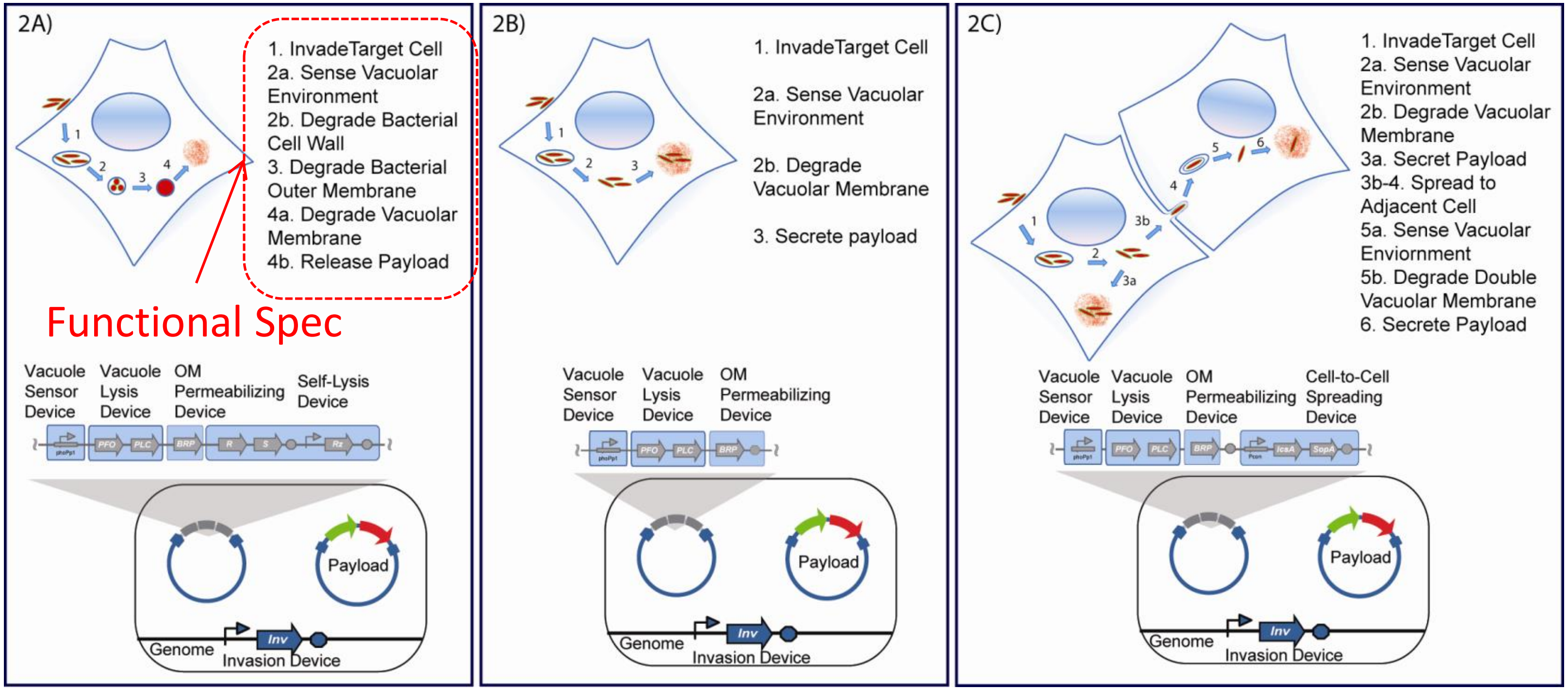
Synthetic Biology: Tremendous Potential



The disciplined design of biological systems using engineering principles

Design Activities

How do you capture, specify, and validate this design? How do you share it? Reason about it?



2A)

1. InvadeTarget Cell
2a. Sense Vacuolar Environment
2b. Degrade Bacterial Cell Wall
3. Degrade Bacterial Outer Membrane
4a. Degrade Vacuolar Membrane
4b. Release Payload

2B)

1. InvadeTarget Cell

2a. Sense Vacuolar Environment

2b. Degrade Vacuolar Membrane

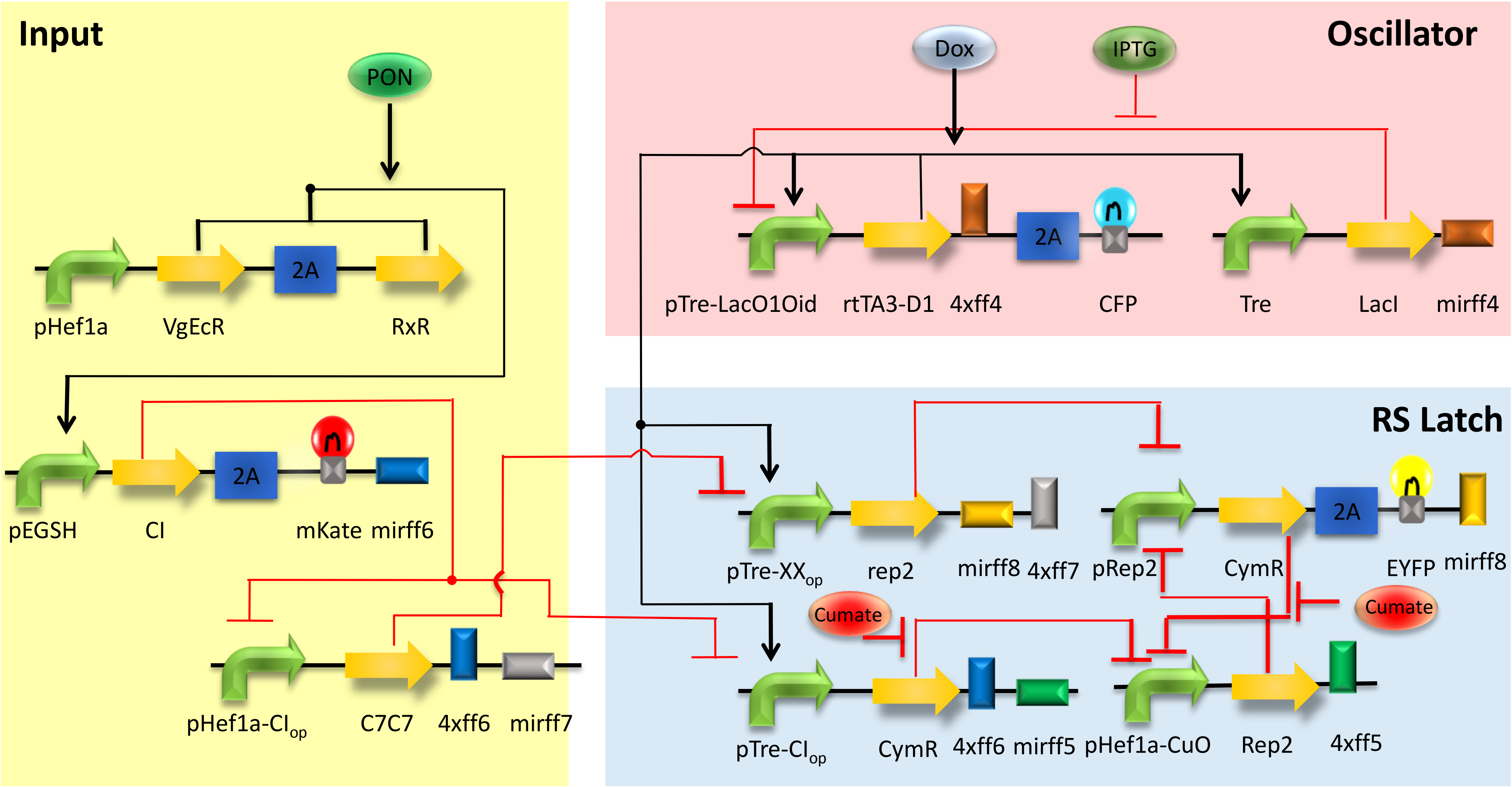
3. Secrete payload

2C)

1. InvadeTarget Cell
2a. Sense Vacuolar Environment
2b. Degrade Vacuolar Membrane
3a. Secret Payload
3b-4. Spread to Adjacent Cell
5a. Sense Vacuolar Enviornment
5b. Degrade Double Vacuolar Membrane
6. Secrete Payload

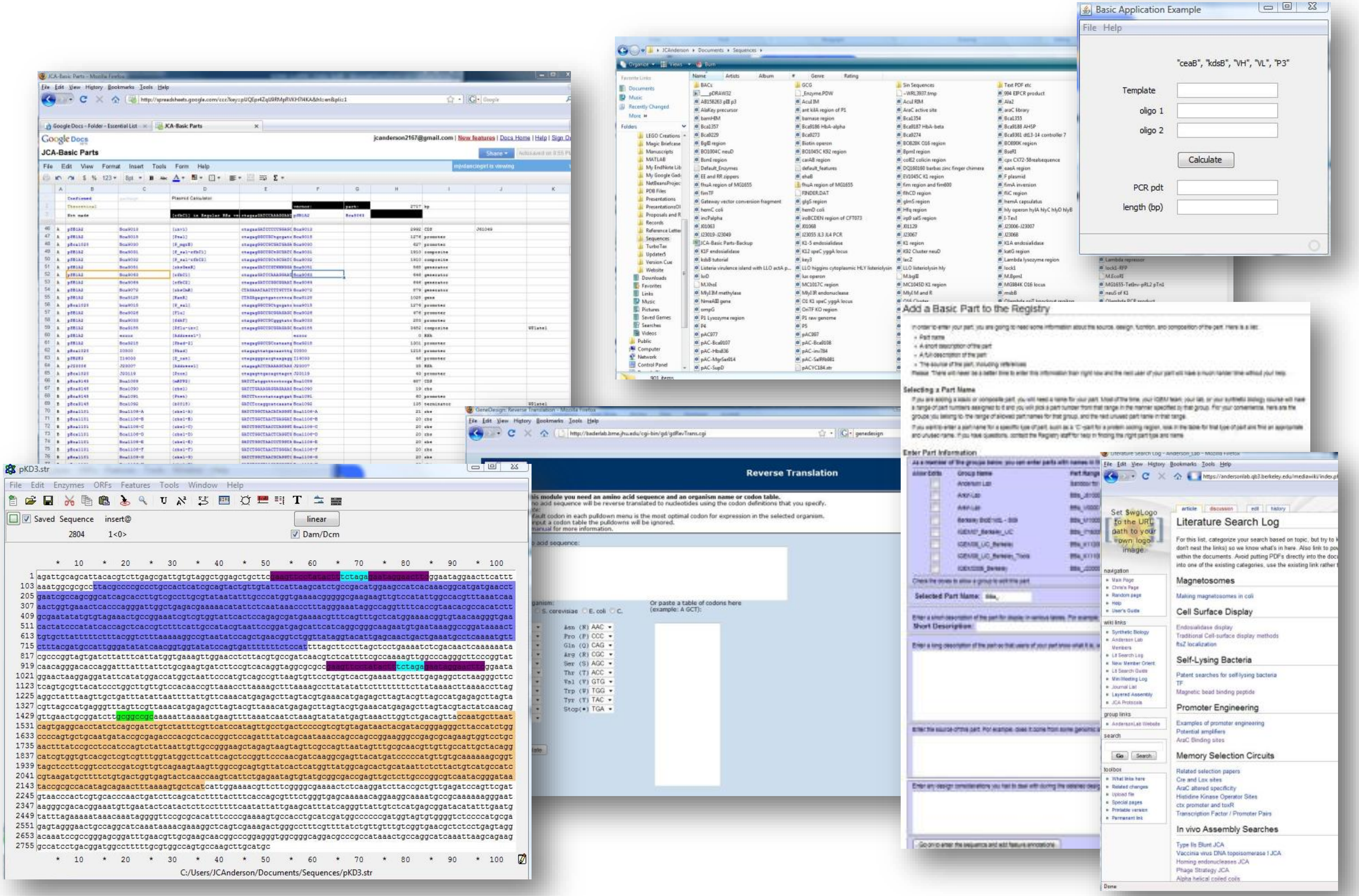
Courtesy: Jin Huh and J.C. Anderson (UCB)

Potential Complexity?



Circuit size: 9 promoters, 13 genes

State of the Art??



Basic Application Example

File Help

"ceaB", "kdsB", "VH", "VL", "P3"

Template

oligo 1

oligo 2

Calculate

PCR pdt

length (bp)

JCA-Basic Parts

pKD3.str

File Edit Enzymes ORFs Features Tools Window Help

☑ Saved Sequence insert@

2804 1<0>

linear

☑ Dam/Dcm

C:/Users/JCAnderson/Documents/Sequences/pKD3.str

Reverse Translation

Add a Basic Part to the Registry

Selecting a Part Name

Enter Part Information

Literature Search Log

Magnetosomes

Making magnetosomes in coli

Cell Surface Display

Endosialidase display

Traditional Cell-surface display methods

ftsZ localization

Self-Lysing Bacteria

Patent searches for self-lysing bacteria

TF

Magnetic bead binding peptide

Promoter Engineering

Examples of promoter engineering

Potential amplifiers

AraC Binding sites

Memory Selection Circuits

Related selection papers

Cre and Lox sites

AraC altered specificity

Histidine Kinase Operator Sites

ctx promoter and toxR

Transcription Factor / Promoter Pairs

In vivo Assembly Searches

Type IIs Blunt JCA

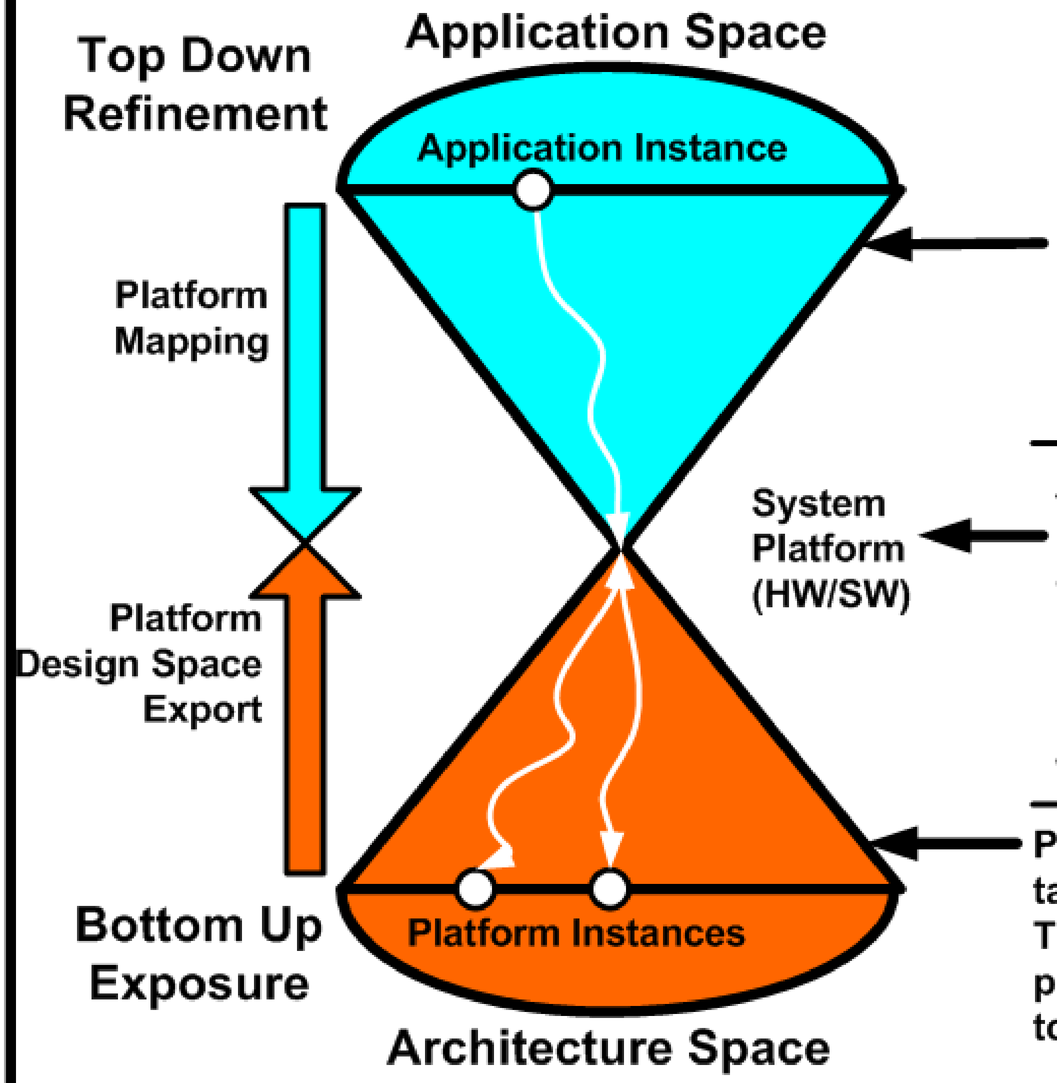
Vaccinia virus DNA topoisomerase I JCA

Homing endonucleases JCA

Phage Strategy JCA

Alpha helical coiled coils

Platform-Based Design Methodology



Functionality

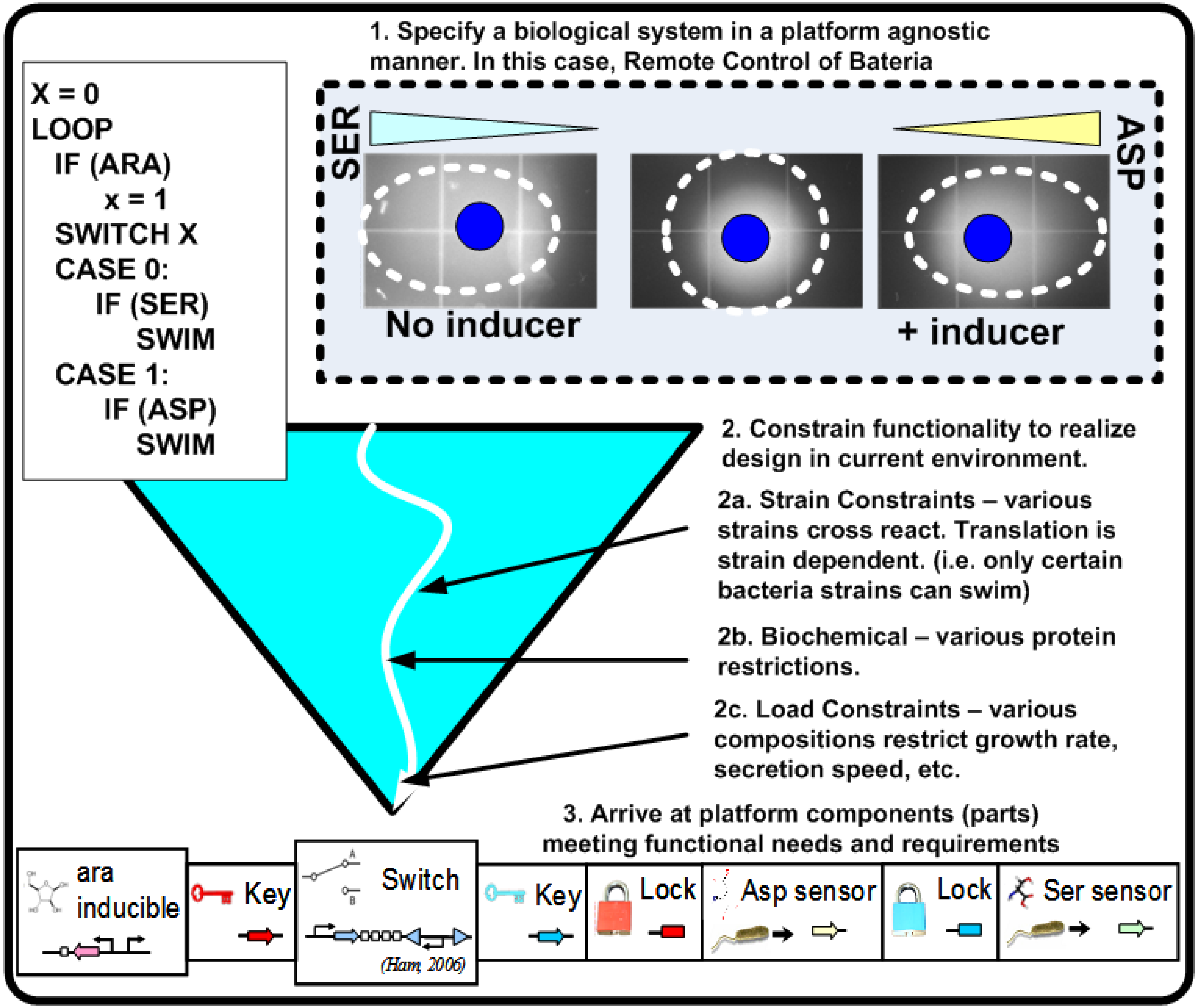
Describes the functionality of the design. Restricts functionality as necessary to make design less abstract so it can physically be realized.

A common semantic domain in which the application and architectural space meet. Here mapping tradeoffs can be explored and potential performance estimations be examined.

Provides various architectural targets for implementation. These targets should allow performance exportation up to the platform for estimation.

Cost

Circa 2008



```
X = 0
LOOP
  IF (ARA)
      x = 1
  SWITCH X
  CASE 0:
     IF (SER)
        SWIM
  CASE 1:
      IF (ASP)
        SWIM
```

1. Specify a biological system in a platform agnostic manner. In this case, Remote Control of Bateria

2. Constrain functionality to realize design in current environment.

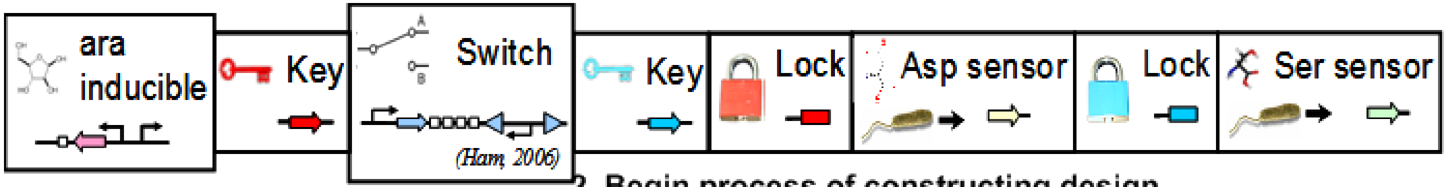
2a. Strain Constraints – various strains cross react. Translation is strain dependent. (i.e. only certain bacteria strains can swim)

2b. Biochemical – various protein restrictions.

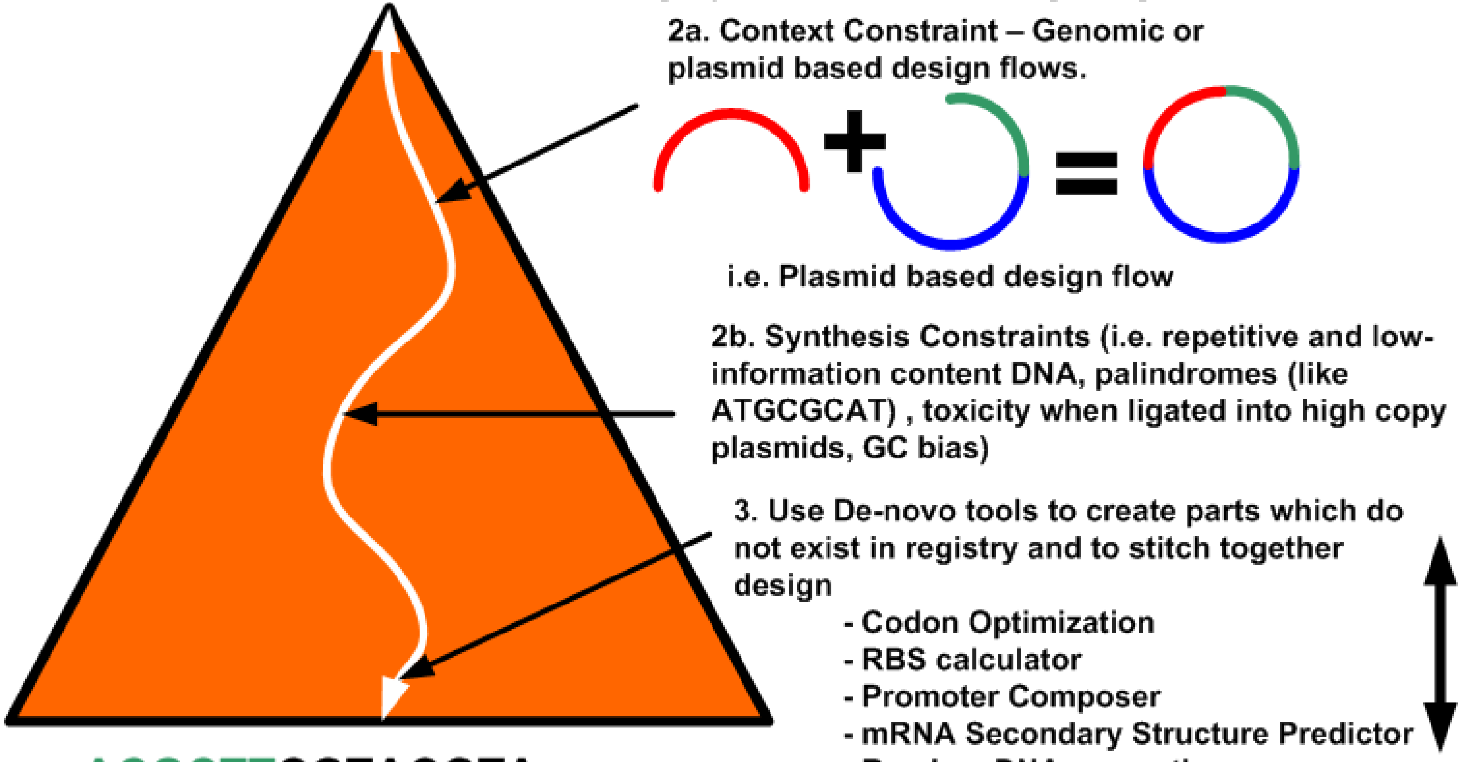
2c. Load Constraints – various compositions restrict growth rate, secretion speed, etc.

3. Arrive at platform components (parts) meeting functional needs and requirements

1. Start with a collection of standardized parts which have the potential to implement the functionality needed.



2. Begin process of constructing design.



2a. Context Constraint – Genomic or plasmid based design flows.

i.e. Plasmid based design flow

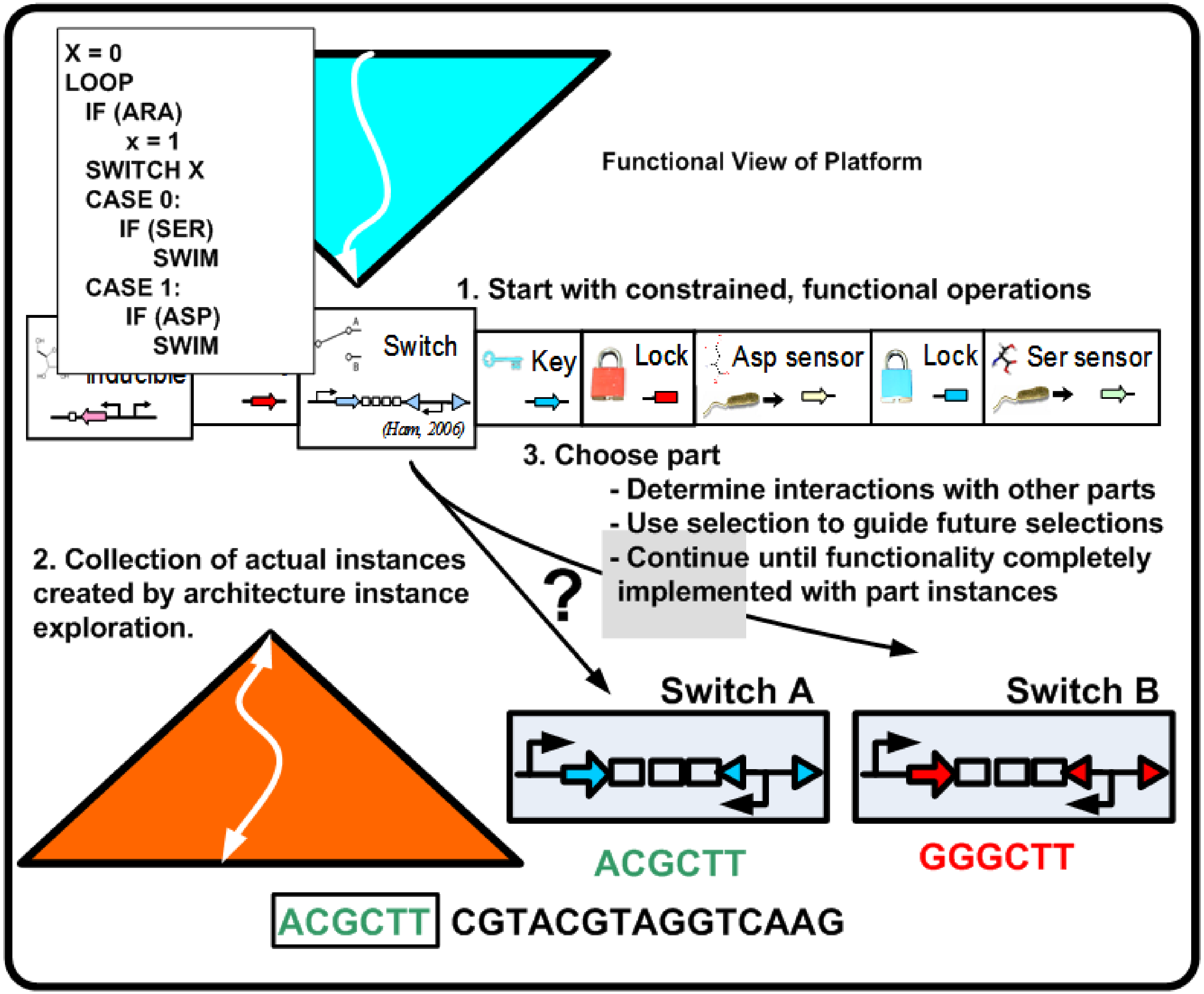
2b. Synthesis Constraints (i.e. repetitive and low-information content DNA, palindromes (like ATGCGCAT) , toxicity when ligated into high copy plasmids, GC bias)

3. Use De-novo tools to create parts which do not exist in registry and to stitch together design

- Codon Optimization
- RBS calculator
- Promoter Composer
- mRNA Secondary Structure Predictor
- Random DNA generation

ACGCTTCGTACGTA
GGTCAAGGGGCTT

These tools give info about the design (cost) which determine if this part(s) is amenable or not



```
X = 0
LOOP
  IF (ARA)
      x = 1
  SWITCH X
  CASE 0:
     IF (SER)
        SWIM
  CASE 1:
      IF (ASP)
        SWIM
```

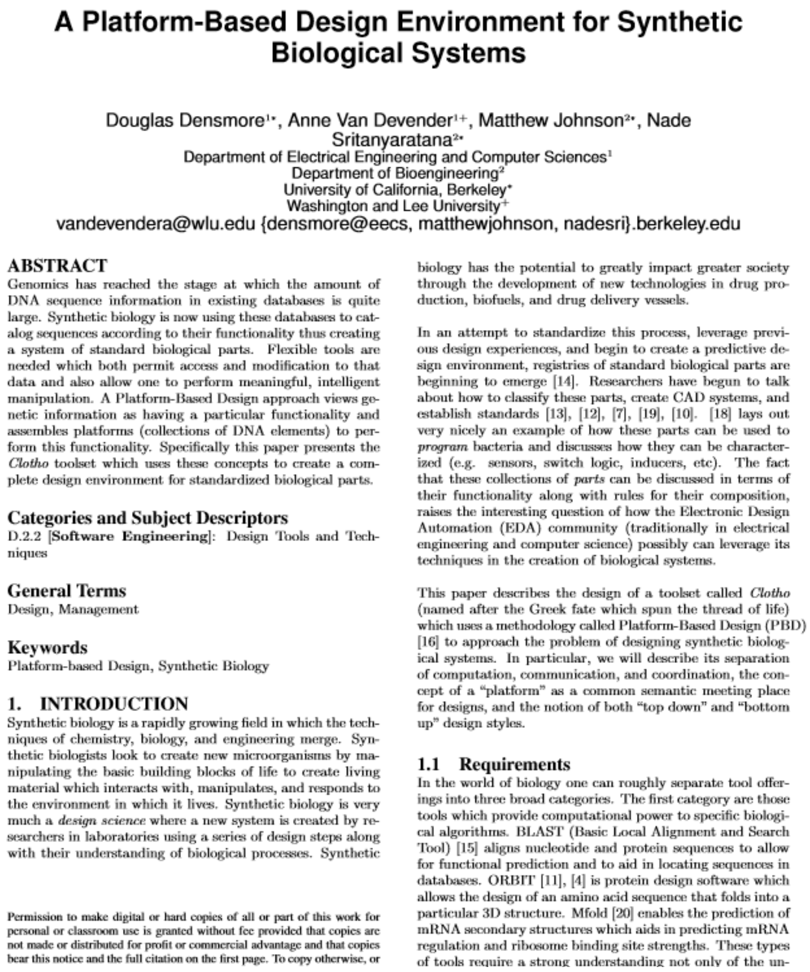
Functional View of Platform

1. Start with constrained, functional operations

2. Collection of actual instances created by architecture instance exploration.

3. Choose part
   - Determine interactions with other parts
   - Use selection to guide future selections
   - Continue until functionality completely implemented with part instances

Clotho 0.0: 2007-2009



A Platform-Based Design Environment for Synthetic Biological Systems

Douglas Densmore[1*], Anne Van Devender[1+], Matthew Johnson[2*], Nade Sritanyaratana[2*]
Department of Electrical Engineering and Computer Sciences[1]
Department of Bioengineering[2]
University of California, Berkeley[*]
Washington and Lee University[+]
vandevendera@wlu.edu {densmore@eecs, matthewjohnson, nadesri}.berkeley.edu

ABSTRACT

Genomics has reached the stage at which the amount of DNA sequence information in existing databases is quite large. Synthetic biology is now using these databases to catalog sequences according to their functionality thus creating a system of standard biological parts. Flexible tools are needed which both permit access and modification to that data and also allow one to perform meaningful, intelligent manipulation. A Platform-Based Design approach views genetic information as having a particular functionality and assembles platforms (collections of DNA elements) to perform this functionality. Specifically this paper presents the *Clotho* toolset which uses these concepts to create a complete design environment for standardized biological parts.

Categories and Subject Descriptors

D.2.2 [**Software Engineering**]: Design Tools and Techniques

General Terms

Design, Management

Keywords

Platform-based Design, Synthetic Biology

1. INTRODUCTION

Synthetic biology is a rapidly growing field in which the techniques of chemistry, biology, and engineering merge. Synthetic biologists look to create new microorganisms by manipulating the basic building blocks of life to create living material which interacts with, manipulates, and responds to the environment in which it lives. Synthetic biology is very much a *design science* where a new system is created by researchers in laboratories using a series of design steps along with their understanding of biological processes. Synthetic biology has the potential to greatly impact greater society through the development of new technologies in drug production, biofuels, and drug delivery vessels.

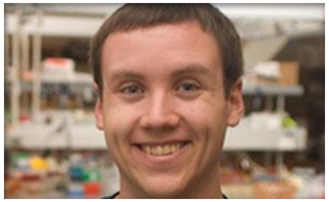
In an attempt to standardize this process, leverage previous design experiences, and begin to create a predictive design environment, registries of standard biological parts are beginning to emerge [14]. Researchers have begun to talk about how to classify these parts, create CAD systems, and establish standards [13], [12], [7], [19], [10]. [18] lays out very nicely an example of how these parts can be used to *program* bacteria and discusses how they can be characterized (e.g. sensors, switch logic, inducers, etc). The fact that these collections of *parts* can be discussed in terms of their functionality along with rules for their composition, raises the interesting question of how the Electronic Design Automation (EDA) community (traditionally in electrical engineering and computer science) possibly can leverage its techniques in the creation of biological systems.

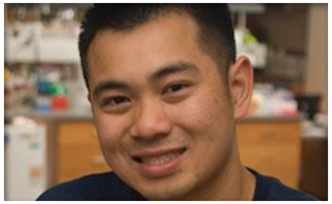
This paper describes the design of a toolset called *Clotho* (named after the Greek fate which spun the thread of life) which uses a methodology called Platform-Based Design (PBD) [16] to approach the problem of designing synthetic biological systems. In particular, we will describe its separation of computation, communication, and coordination, the concept of a "platform" as a common semantic meeting place for designs, and the notion of both "top down" and "bottom up" design styles.

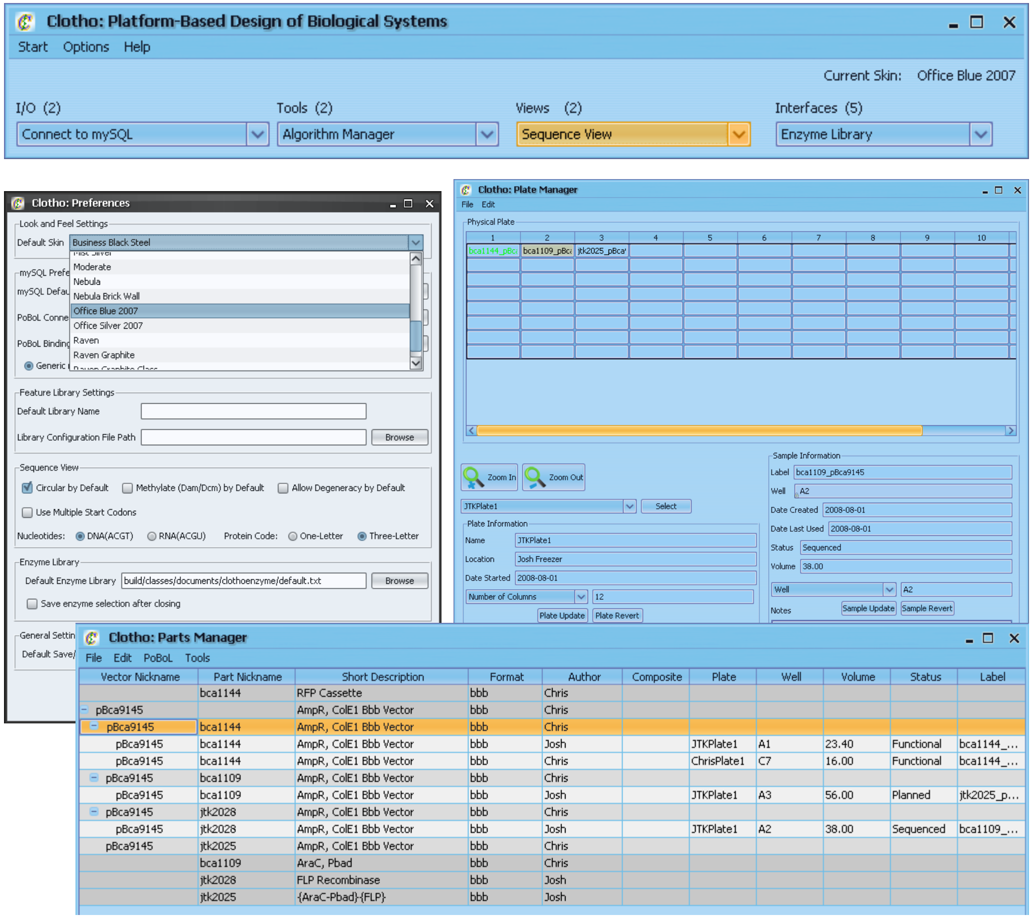
1.1 Requirements

In the world of biology one can roughly separate tool offerings into three broad categories. The first category are those tools which provide computational power to specific biological algorithms. BLAST (Basic Local Alignment and Search Tool) [15] aligns nucleotide and protein sequences to allow for functional prediction and to aid in locating sequences in databases. ORBIT [11], [4] is protein design software which allows the design of an amino acid sequence that folds into a particular 3D structure. Mfold [20] enables the prediction of mRNA secondary structures which aids in predicting mRNA regulation and ribosome binding site strengths. These types of tools require a strong understanding not only of the un-

"A platform-based design environment for synthetic biological systems," in *The Fifth Richard Tapia Celebration of Diversity in Computing Conference: Intellect, Initiatives, Insight, and Innovations*, New York, NY, USA, 2009, pp. 24-29.







Clotho: Platform-Based Design of Biological Systems

Current Skin: Office Blue 2007

I/O (2): Connect to mySQL
Tools (2): Algorithm Manager
Views (2): Sequence View
Interfaces (5): Enzyme Library

Clotho: Preferences

Look and Feel Settings
Default Skin: Business Black Steel

mySQL Prefe
mySQL Defau
PoBoL Conne
PoBoL Binding
● Generic

Feature Library Settings
Default Library Name
Library Configuration File Path

Sequence View
☑ Circular by Default ☐ Methylate (Dam/Dcm) by Default ☐ Allow Degeneracy by Default
☐ Use Multiple Start Codons
Nucleotides: ● DNA(ACGT) ○ RNA(ACGU) Protein Code: ○ One-Letter ● Three-Letter

Enzyme Library
Default Enzyme Library: build/classes/documents/clothoenzyme/default.txt

☐ Save enzyme selection after closing

General Settin
Default Save/

Clotho: Plate Manager

Physical Plate

1	2	3	4	5	6	7	8	9	10
bca1144_pBc	bca1109_pBc	jtk2025_pBca							

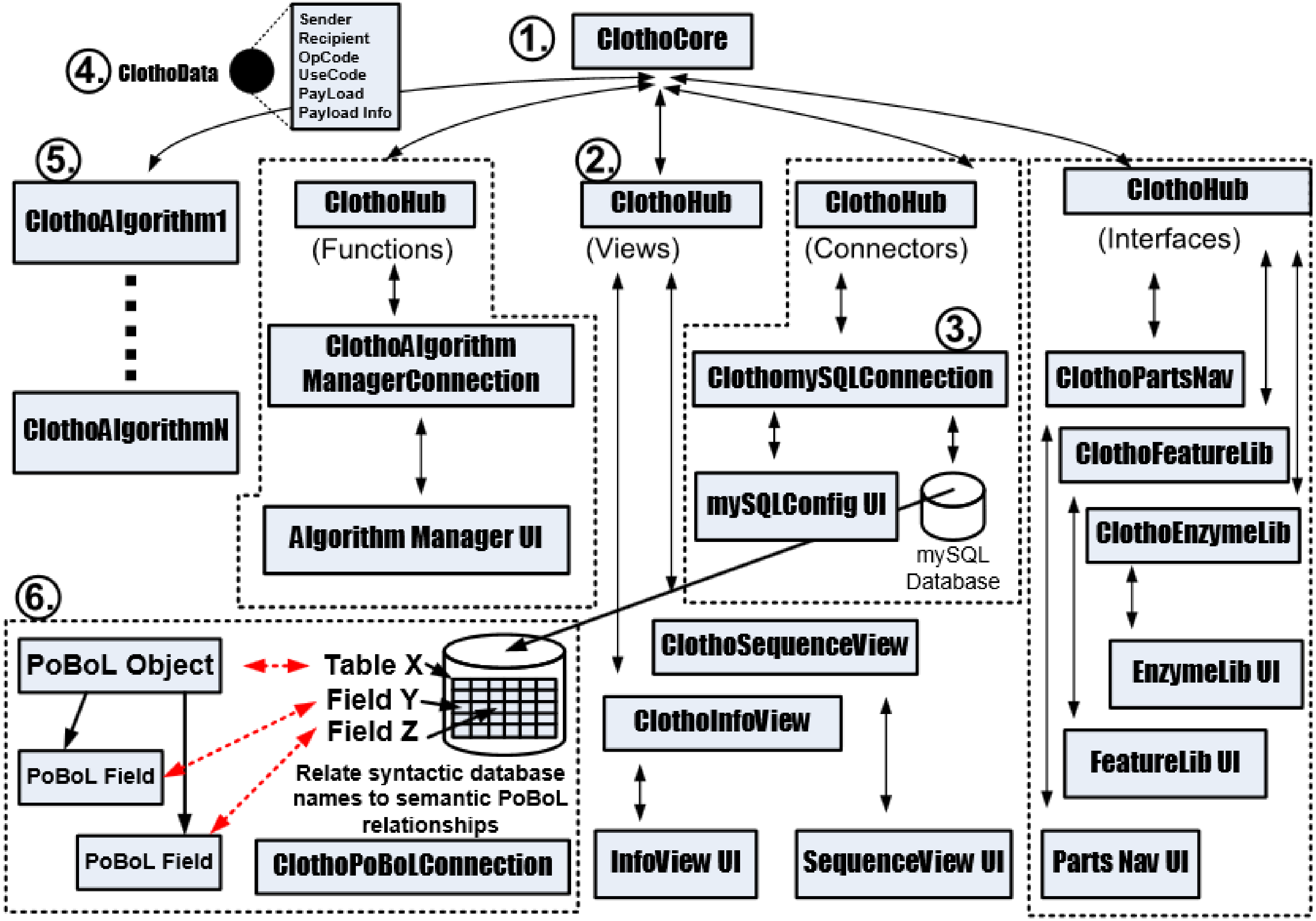
JTKPlate1

Plate Information
Name: JTKPlate1
Location: Josh Freezer
Date Started: 2008-08-01
Number of Columns: 12

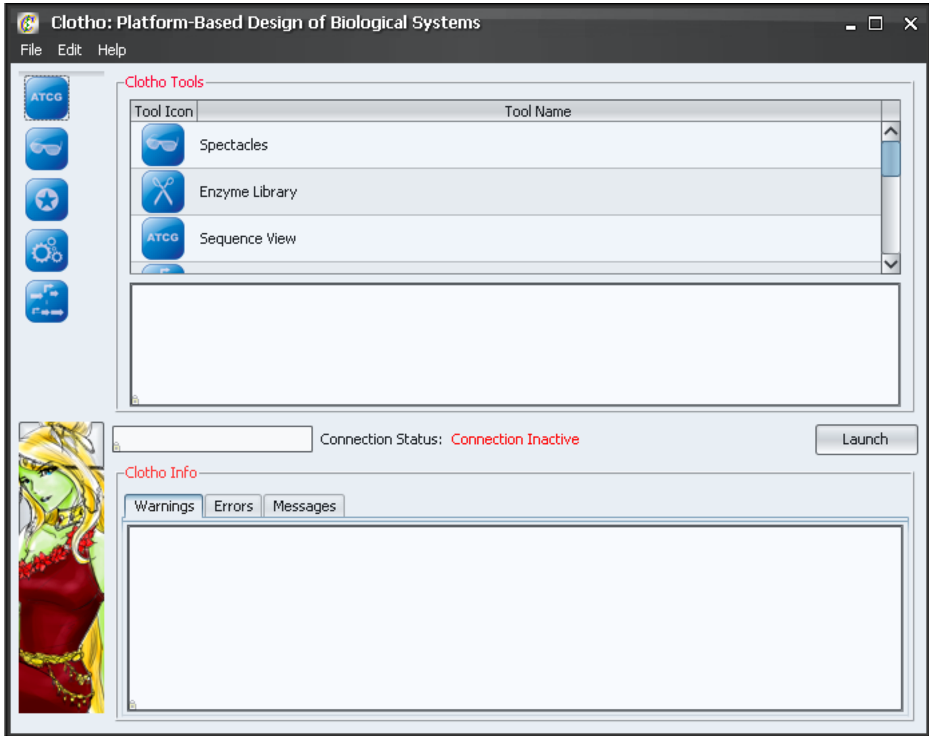
Sample Information
Label: bca1109_pBca9145
Well: A2
Date Created: 2008-08-01
Date Last Used: 2008-08-01
Status: Sequenced
Volume: 38.00
Well: A2
Notes

Clotho: Parts Manager

Vector Nickname	Part Nickname	Short Description	Format	Author	Composite	Plate	Well	Volume	Status	Label
	bca1144	RFP Cassette	bbb	Chris						
pBca9145		AmpR, ColE1 Bbb Vector	bbb	Chris						
pBca9145	bca1144	AmpR, ColE1 Bbb Vector	bbb	Chris						
pBca9145	bca1144	AmpR, ColE1 Bbb Vector	bbb	Josh		JTKPlate1	A1	23.40	Functional	bca1144_...
pBca9145	bca1144	AmpR, ColE1 Bbb Vector	bbb	Chris		ChrisPlate1	C7	16.00	Functional	bca1144_...
pBca9145	bca1109	AmpR, ColE1 Bbb Vector	bbb	Chris						
pBca9145	bca1109	AmpR, ColE1 Bbb Vector	bbb	Josh		JTKPlate1	A3	56.00	Planned	jtk2025_p...
pBca9145	jtk2028	AmpR, ColE1 Bbb Vector	bbb	Chris						
pBca9145	jtk2028	AmpR, ColE1 Bbb Vector	bbb	Josh		JTKPlate1	A2	38.00	Sequenced	bca1109_...
pBca9145	jtk2025	AmpR, ColE1 Bbb Vector	bbb	Chris						
	bca1109	AraC, Pbad	bbb	Chris						
	jtk2028	FLP Recombinase	bbb	Josh						
	jtk2025	{AraC-Pbad}{FLP}	bbb	Josh						


1.
ClothoCore
4. ClothoData
Sender
Recipient
OpCode
UseCode
PayLoad
Payload Info
5.
ClothoAlgorithm1
ClothoAlgorithmN
ClothoHub
(Functions)
ClothoAlgorithm ManagerConnection
Algorithm Manager UI
2.
ClothoHub
(Views)
ClothoHub
(Connectors)
3.
ClothomySQLConnection
mySQLConfig UI
mySQL Database
ClothoHub
(Interfaces)
ClothoPartsNav
ClothoFeatureLib
ClothoEnzymeLib
EnzymeLib UI
FeatureLib UI
Parts Nav UI
6.
PoBoL Object
Table X
Field Y
Field Z
PoBoL Field
PoBoL Field
Relate syntactic database names to semantic PoBoL relationships
ClothoPoBoLConnection
ClothoSequenceView
ClothoInfoView
InfoView UI
SequenceView UI

Clotho 1.0: 2009-2010



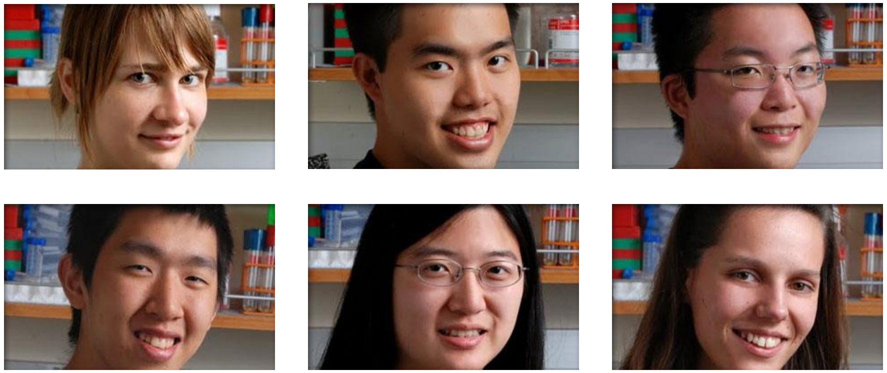
Clotho: Platform-Based Design of Biological Systems

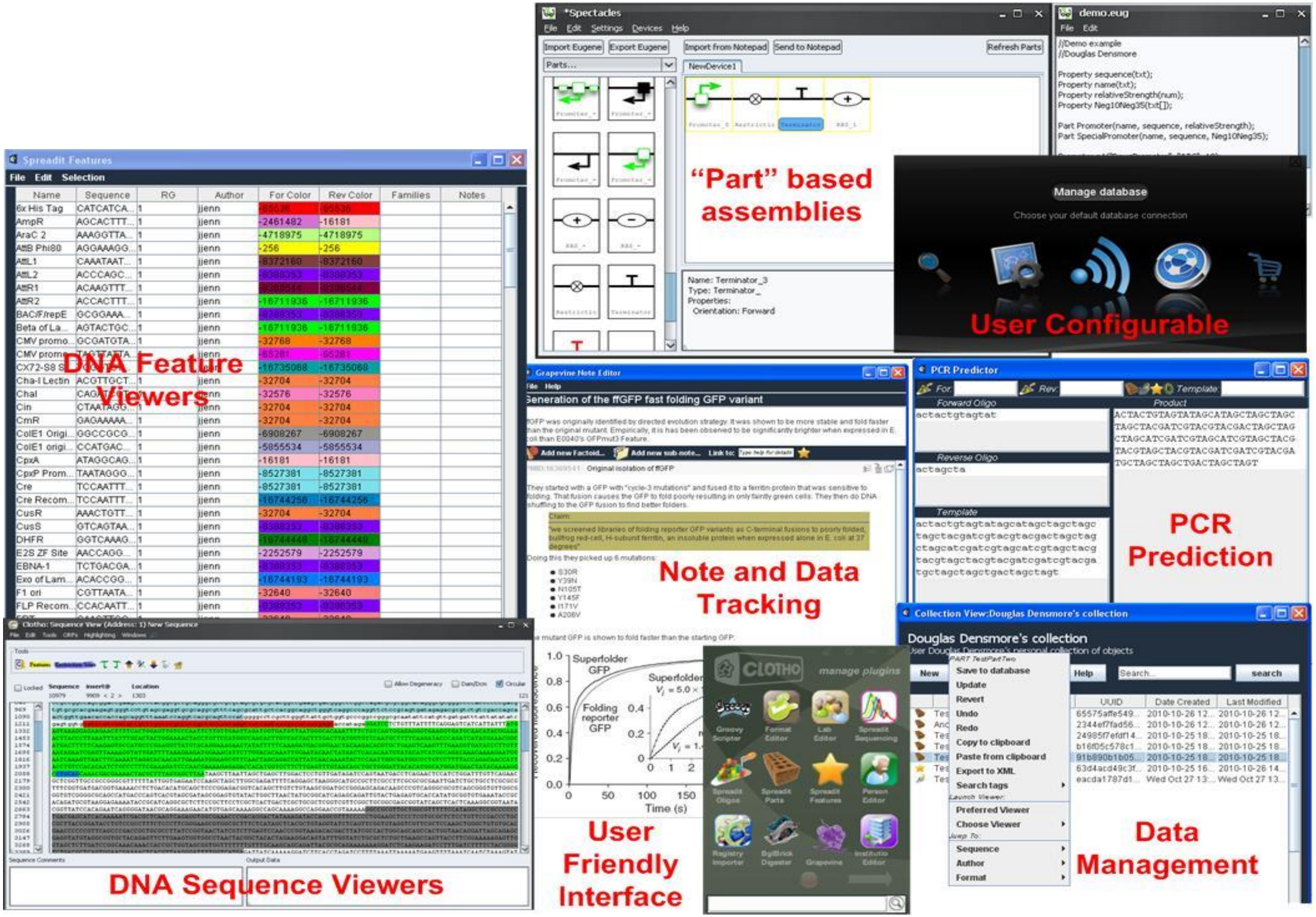
Clotho Tools

Tool Icon	Tool Name
	Spectacles
	Enzyme Library
	Sequence View

Connection Status: Connection Inactive

Clotho Info





Name	Sequence	RG	Author	For Color	Rev Color	Families	Notes
6x His Tag	CATCATCA...	1	jjenn	-65536	-65536		
AmpR	AGCACTTT...	1	jjenn	-2461482	-16181		
AraC 2	AAAGGTTA...	1	jjenn	-4718975	-4718975		
AttB Phi80	AGGAAAGG...	1	jjenn	-256	-256		
AttL1	CAAATAAT...	1	jjenn	-8372160	-8372160		
AttL2	ACCCAGC...	1	jjenn	-8388353	-8388353		
AttR1	ACAAGTTT...	1	jjenn	-8388544	-8388544		
AttR2	ACCACTTT...	1	jjenn	-16711936	-16711936		
BAC/F/repE	GCGGAAA...	1	jjenn	-8388353	-8388353		
Beta of La...	AGTACTGC...	1	jjenn	-16711936	-16711936		
CMV promo...	GCGATGTA...	1	jjenn	-32768	-32768		
CMV prom...	TAGTTATTA...	1	jjenn	-65281	-65281		
CX72-S8 S...	[illegible]	1	jjenn	-16735068	-16735068		
Cha-I Lectin	ACGTTGCT...	1	jjenn	-32704	-32704		
Chal	CAG...	1	jjenn	-32576	-32576		
Cin	CTAATAGG...	1	jjenn	-32704	-32704		
CmR	GAGAAAAA...	1	jjenn	-32704	-32704		
ColE1 Origi...	GGCCGCG...	1	jjenn	-6908267	-6908267		
ColE1 origi...	CCATGAC...	1	jjenn	-5855534	-5855534		
CpxA	ATAGGCAG...	1	jjenn	-16181	-16181		
CpxP Prom...	TAATAGGG...	1	jjenn	-8527381	-8527381		
Cre	TCCAATTT...	1	jjenn	-8527381	-8527381		
Cre Recom...	TCCAATTT...	1	jjenn	-16744256	-16744256		
CusR	AAACTGTT...	1	jjenn	-32704	-32704		
CusS	GTCAGTAA...	1	jjenn	-8388353	-8388353		
DHFR	GGTCAAAG...	1	jjenn	-16744448	-16744448		
E2S ZF Site	AACCAGG...	1	jjenn	-2252579	-2252579		
EBNA-1	TCTGACGA...	1	jjenn	-8388353	-8388353		
Exo of Lam...	ACACCGG...	1	jjenn	-16744193	-16744193		
F1 ori	CGTTAATA...	1	jjenn	-32640	-32640		
FLP Recom...	CCACAATT...	1	jjenn	-8388353	-8388353		

DNA Feature Viewers

DNA Sequence Viewers

Name: Terminator_3
Type: Terminator_
Properties:
Orientation: Forward

"Part" based assemblies

```
//Demo example
//Douglas Densmore

Property sequence(txt);
Property name(txt);
Property relativeStrength(num);
Property Neg10Neg35(txt[]);

Part Promoter(name, sequence, relativeStrength);
Part SpecialPromoter(name, sequence, Neg10Neg35);
```

User Configurable

Generation of the ffGFP fast folding GFP variant

ffGFP was originally identified by directed evolution strategy. It was shown to be more stable and fold faster than the original mutant. Empirically, it is has been observed to be significantly brighter when expressed in E. coli than E0040's GFPmut3 Feature.

PMID:16369541 Original isolation of ffGFP

They started with a GFP with "cycle-3 mutations" and fused it to a ferritin protein that was sensitive to folding. That fusion causes the GFP to fold poorly resulting in only faintly green cells. They then do DNA shuffling to the GFP fusion to find better folders.

Claim:

"we screened libraries of folding reporter GFP variants as C-terminal fusions to poorly folded, bullfrog red-cell, H-subunit ferritin, an insoluble protein when expressed alone in E. coli at 37 degrees"

Doing this they picked up 6 mutations:

- S30R
- Y39N
- N105T
- Y145F
- I171V
- A206V

...e mutant GFP is shown to fold faster than the starting GFP:

Note and Data Tracking

User Friendly Interface

Forward Oligo

```
actactgtagtat
```

Reverse Oligo

```
actagcta
```

Template

```
actactgtagtatagcatagctagctagc
tagctacgatcgtacgtacgactagctag
ctagcatcgatcgtagcatcgtagctacg
tacgtagctacgtacgatcgatcgtacga
tgctagctagctgactagctagt
```

Product

```
ACTACTGTAGTATAGCATAGCTAGCTAGC
TAGCTACGATCGTACGTACGACTAGCTAG
CTAGCATCGATCGTAGCATCGTAGCTACG
TACGTAGCTACGTACGATCGATCGTACGA
TGCTAGCTAGCTGACTAGCTAGT
```

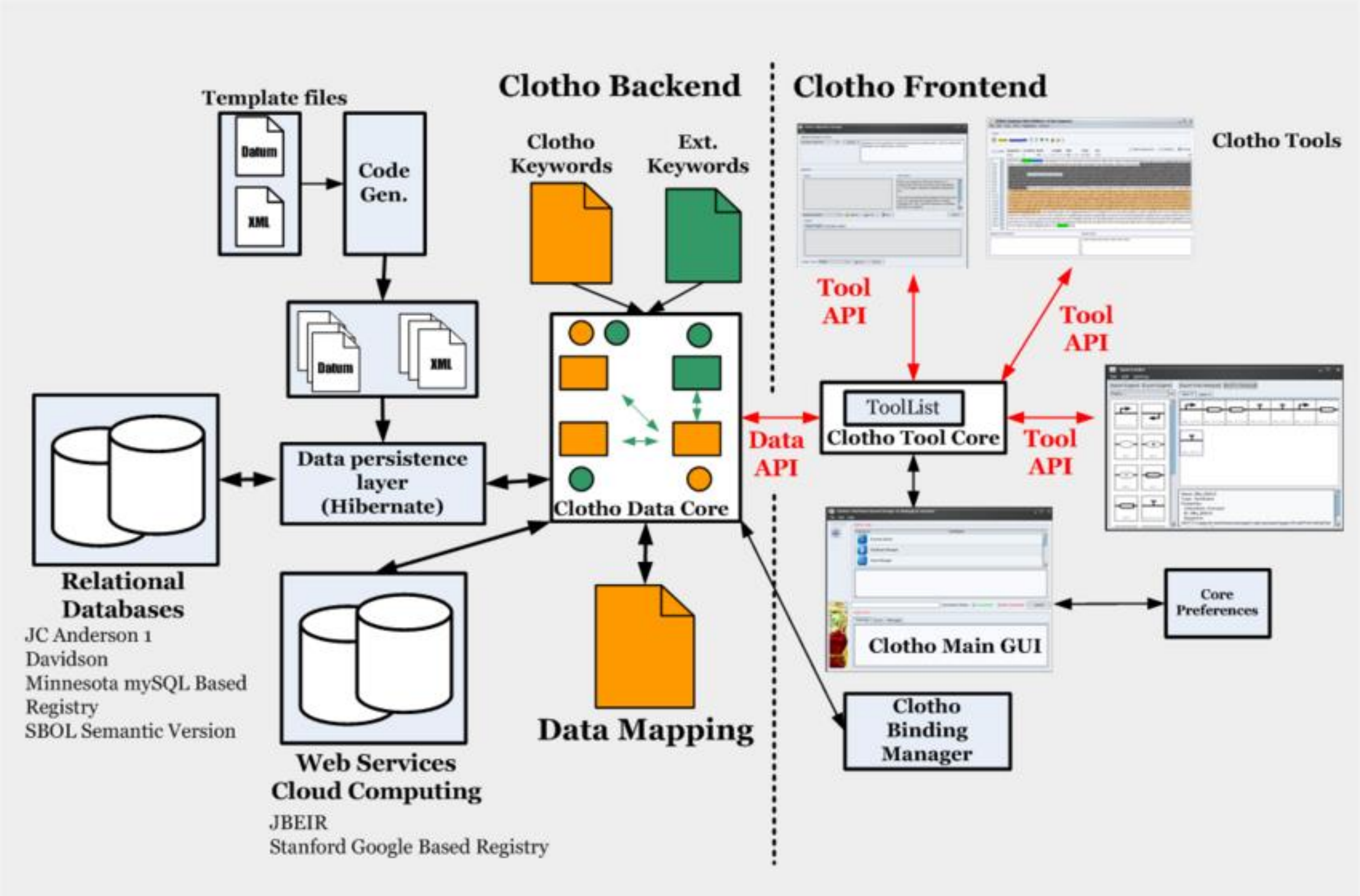
PCR Prediction

Douglas Densmore's collection

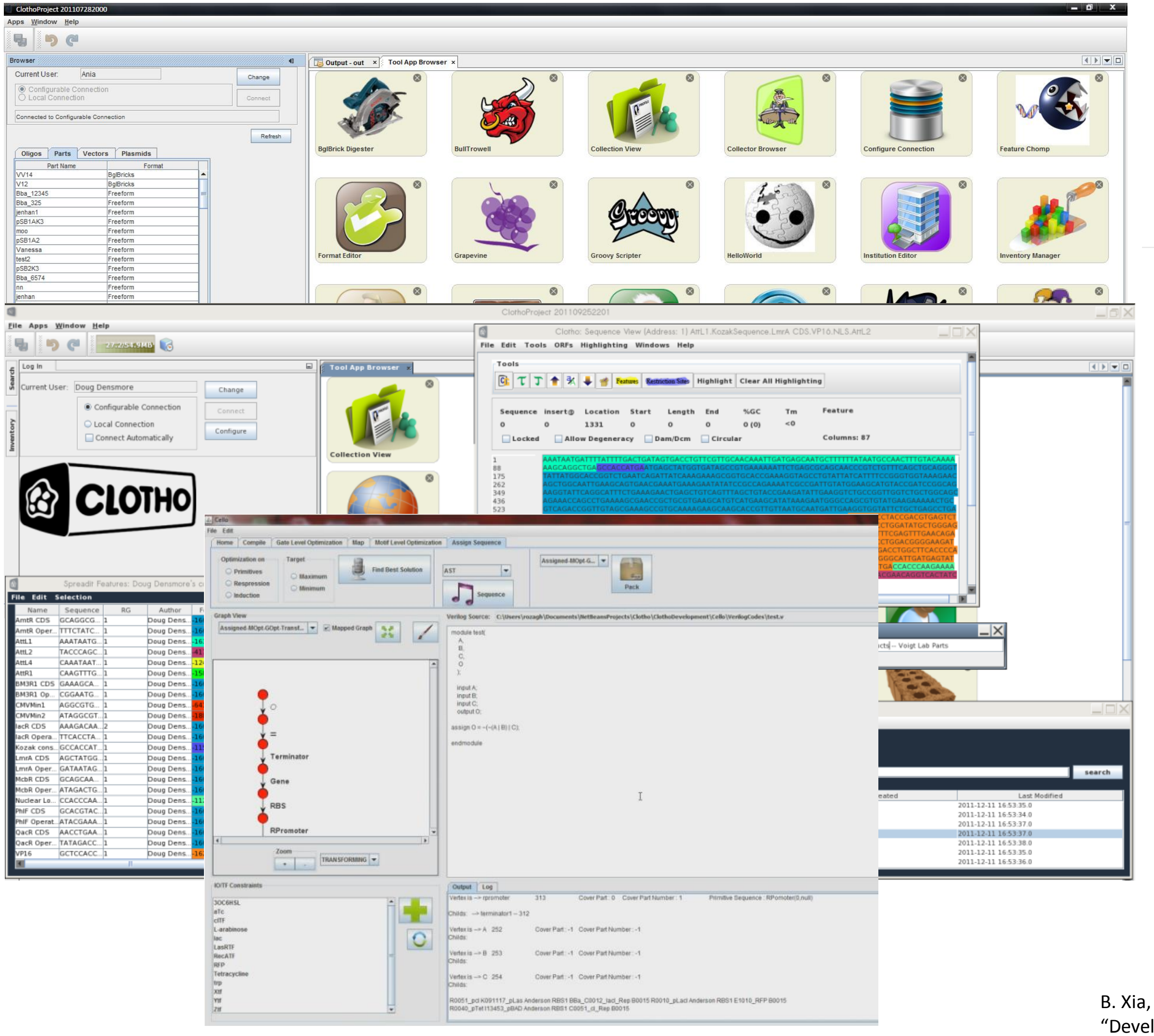
User Douglas Densmore's personal collection of objects

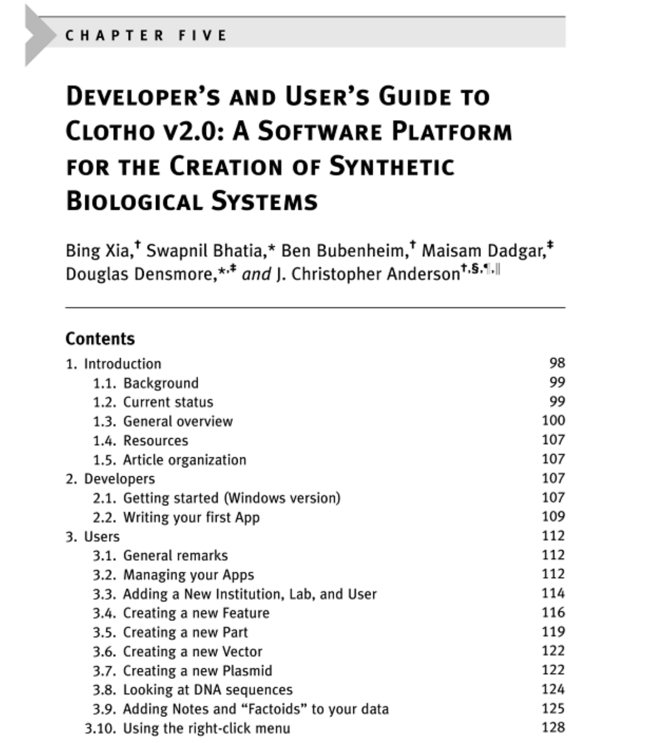
	UUID	Date Created	Last Modified
Tes	65575affe549...	2010-10-26 12...	2010-10-26 12...
And	2344ef7fad56...	2010-10-26 12...	2010-10-26 12...
Tes	24985f7efdf14...	2010-10-25 18:...	2010-10-25 18...
Tes	b16f05c578c1...	2010-10-25 18...	2010-10-25 18...
Tes	91b890b1b05...	2010-10-25 18...	2010-10-25 18...
Tes	63d4acd49c3f...	2010-10-25 16...	2010-10-26 14...
Tes	eacda1787d1...	Wed Oct 27 13:...	Wed Oct 27 13...

Data Management


Template files
Datum
XML
Code Gen.
Datum
XML
Data persistence layer (Hibernate)
Relational Databases
JC Anderson 1
Davidson
Minnesota mySQL Based Registry
SBOL Semantic Version
Web Services Cloud Computing
JBEIR
Stanford Google Based Registry
Clotho Backend
Clotho Keywords
Ext. Keywords
Clotho Data Core
Data Mapping
Clotho Frontend
Clotho Tools
Tool API
Tool API
Data API
ToolList
Clotho Tool Core
Tool API
Clotho Main GUI
Core Preferences
Clotho Binding Manager

Clotho 2.0: 2010-2012





CHAPTER FIVE

Developer's and User's Guide to Clotho v2.0: A Software Platform for the Creation of Synthetic Biological Systems

Bing Xia,[†] Swapnil Bhatia,[*] Ben Bubenheim,[†] Maisam Dadgar,[‡] Douglas Densmore,[*,‡] *and* J. Christopher Anderson[†,§,¶,||]

Contents

B. Xia, S. Bhatia, B. Bubenheim, M. Dadgar, D. Densmore, and J. C. Anderson, "Developer's and user's guide to Clotho v2.0 A software platform for the creation of synthetic biological systems," *Meth. Enzymol.*, vol. 498, pp. 97-135, 2011.

What was wrong? What could be right?

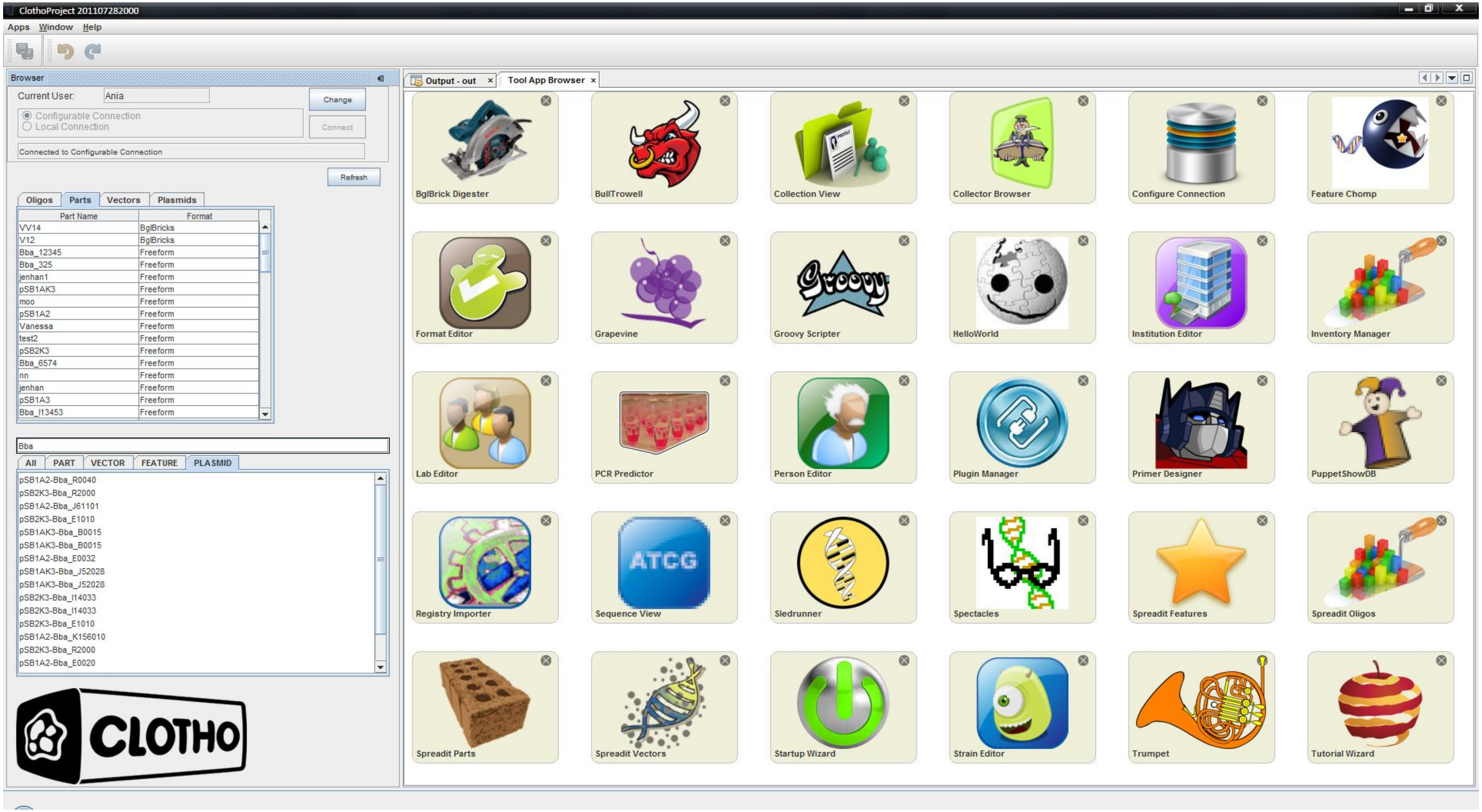
In 2.0

- Rich client – physically had to install Clotho; Java/Swing/Hibernate
  - Updates to code base not globally propagated
  - Non-uniform development platforms
  - Code bloat/registry reminants/etc
- Hard-coded bio objects
  - Provided guideline but field still immature
  - Meta-data lost
- 22+ GUIs
  - What about the 23rd? “Can you make that button green?”
- Fragmented documentation
  - No centralized body to maintain this

In 3.0

- Web-based – “only” need a browser
  - Centralized updating
  - Fewer development platforms
  - More inline with current software development practices
- Abstracted API and Datamodel
  - Comes with 2.0esque model standard
  - Can be modified by the user(s)
  - Multiple UI environment API support
- Just one editor
  - Simplify the Clotho workflow
- Embedded interactive training
  - Clotho actually stores its own documentation
  - Create organization to maintain documentation going forward

Clotho...there's an app for that!



Current User: Ania

● Configurable Connection
○ Local Connection

Connected to Configurable Connection

Oligos | Parts | Vectors | Plasmids

Part Name	Format
VV14	BglBricks
V12	BglBricks
Bba_12345	Freeform
Bba_325	Freeform
jenhan1	Freeform
pSB1AK3	Freeform
moo	Freeform
pSB1A2	Freeform
Vanessa	Freeform
test2	Freeform
pSB2K3	Freeform
Bba_6574	Freeform
nn	Freeform
jenhan	Freeform
pSB1A3	Freeform
Bba_I13453	Freeform

Bba

All | PART | VECTOR | FEATURE | PLASMID

pSB1A2-Bba_R0040
pSB2K3-Bba_R2000
pSB1A2-Bba_J61101
pSB2K3-Bba_E1010
pSB1AK3-Bba_B0015
pSB1AK3-Bba_B0015
pSB1A2-Bba_E0032
pSB1AK3-Bba_J52028
pSB1AK3-Bba_J52028
pSB2K3-Bba_I14033
pSB2K3-Bba_I14033
pSB2K3-Bba_E1010
pSB1A2-Bba_K156010
pSB2K3-Bba_R2000
pSB1A2-Bba_E0020

Specification Apps

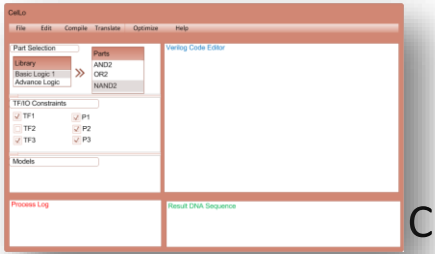
```
Property sequence(txt);
Property strength(num);
Property toxicity(txt);
Property uniqueID(num);

Part Promoter(sequence);
Part ORF(sequence, toxicity);
Part Terminator(sequence, strength);
Part RBS(sequence, uniqueID);

Rule rule2a(p BEFORE r1);
Rule rule2b(p NEXTTO r1);
Rule rule2c(p BEFORE r2);
Rule rule2d(p NEXTTO r2);
Note((rule2a AND rule2b)

//Device Generation
permute(deviceType4, 25, strict)
```

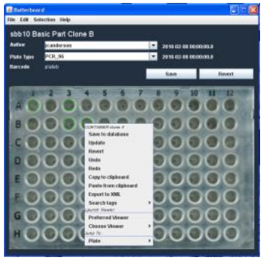
Eugene Scripter

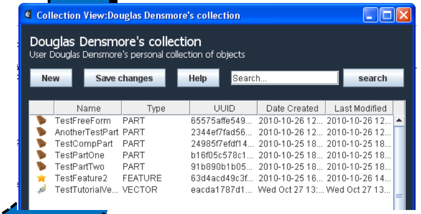


Cello

Data Management Apps

Batterboard





Collection View:Douglas Densmore's collection

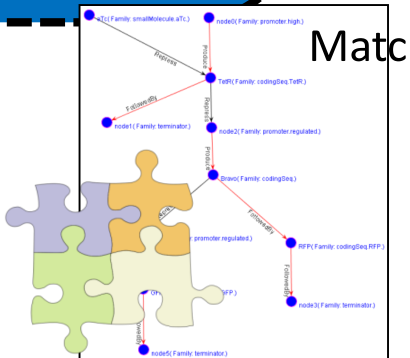
Douglas Densmore's collection

User Douglas Densmore's personal collection of objects

New | Save changes | Help | Search... | search

	Name	Type	UUID	Date Created	Last Modified
	TestFreeForm	PART	65575affe549...	2010-10-26 12...	2010-10-26 12...
	AnotherTestPart	PART	2344ef7fad56...	2010-10-26 12...	2010-10-26 12...
	TestCompPart	PART	24985f7efdf14...	2010-10-25 18...	2010-10-25 18...
	TestPartOne	PART	b16f05c578c1...	2010-10-25 18...	2010-10-25 18...
	TestPartTwo	PART	91b890b1b05...	2010-10-25 18...	2010-10-25 18...
★	TestFeature2	FEATURE	63d4acd49c3f...	2010-10-25 16...	2010-10-26 14...
	TestTutorialVe...	VECTOR	eacda1787d1...	Wed Oct 27 13:...	Wed Oct 27 13...

Part Management



aTc(Family: smallMolecule.aTc.)
node0(Family: promoter.high.)
Repress
Produce
TetR(Family: codingSeq.TetR.)
FollowedBy
Repress
node1(Family: terminator.)
node2(Family: promoter.regulated.)
Produce
Bravo(Family: codingSeq.)
promoter.regulated.)
FollowedBy
RFP(Family: codingSeq.RFP.)
FollowedBy
node3(Family: terminator.)
FollowedBy
node5(Family: terminator.)

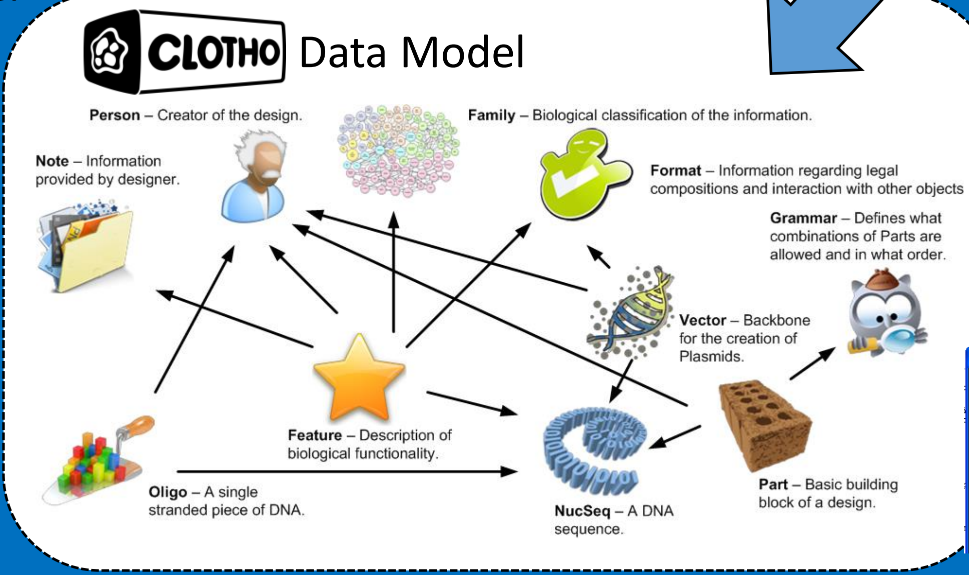
Matchmaker

API

API

API

API



CLOTHO Data Model

Person – Creator of the design.

Family – Biological classification of the information.

Note – Information provided by designer.

Format – Information regarding legal compositions and interaction with other objects

Grammar – Defines what combinations of Parts are allowed and in what order.

Vector – Backbone for the creation of Plasmids.

Feature – Description of biological functionality.

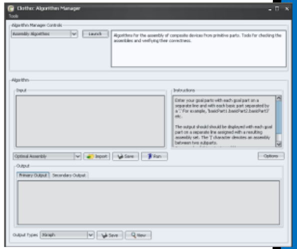
Oligo – A single stranded piece of DNA.

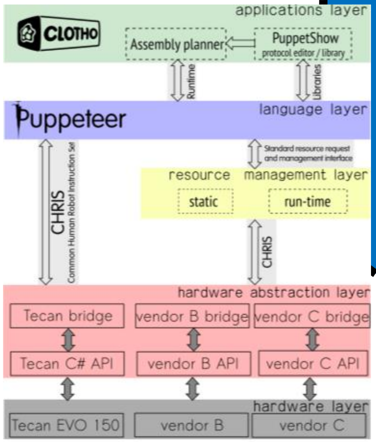
NucSeq – A DNA sequence.

Part – Basic building block of a design.

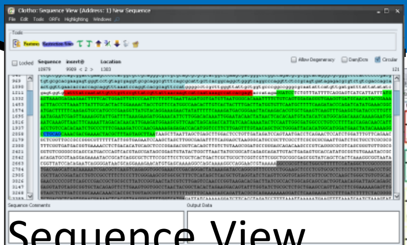
Assembly Apps

Assembly Manager



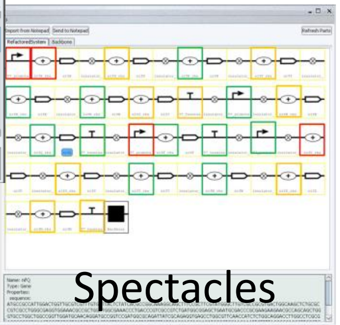


PuppetShow



Sequence View

Design Apps

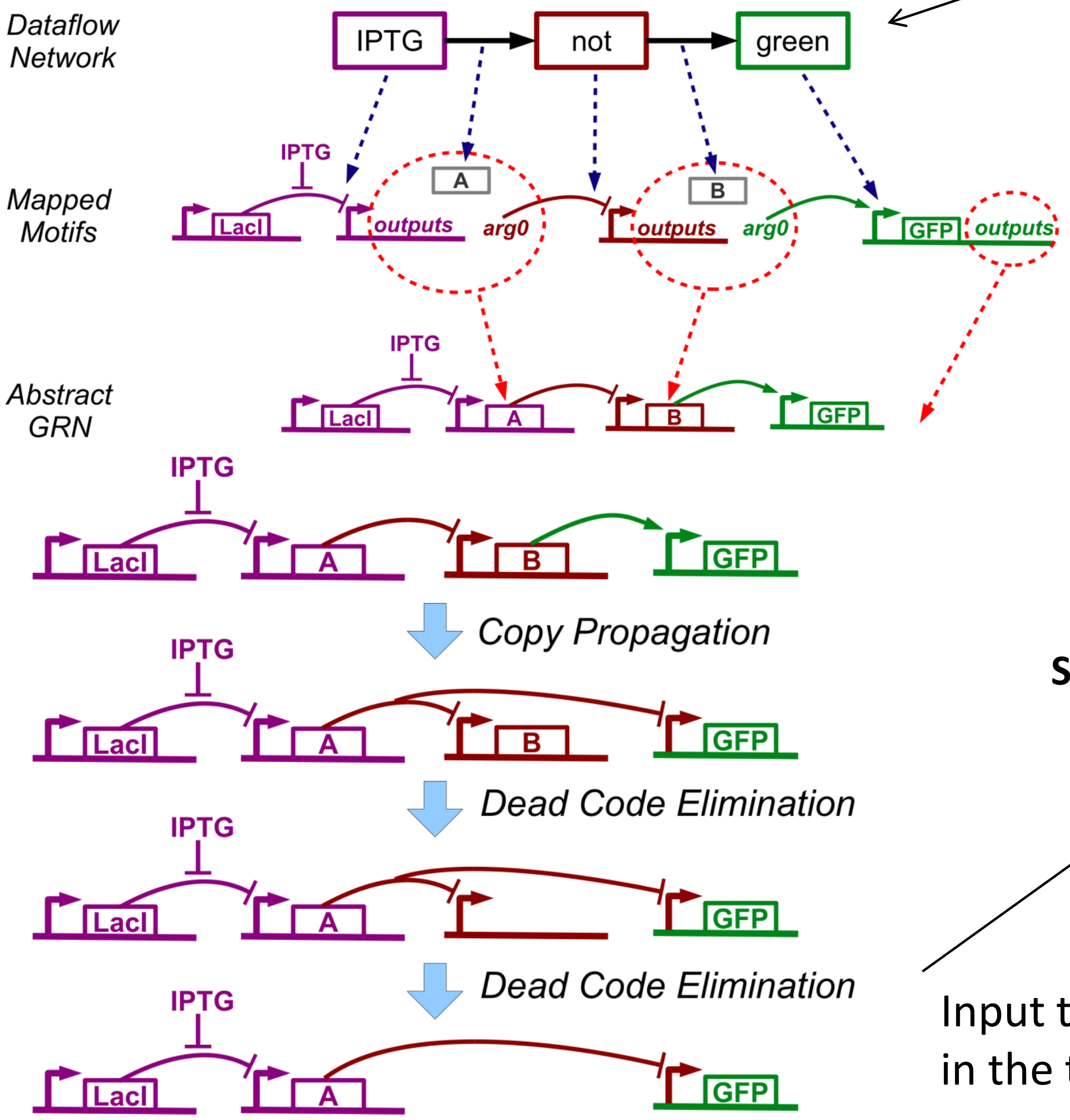


Spectacles

Raytheon
BBN Technologies

Proto

Input Specification



SBOL Compliant Output

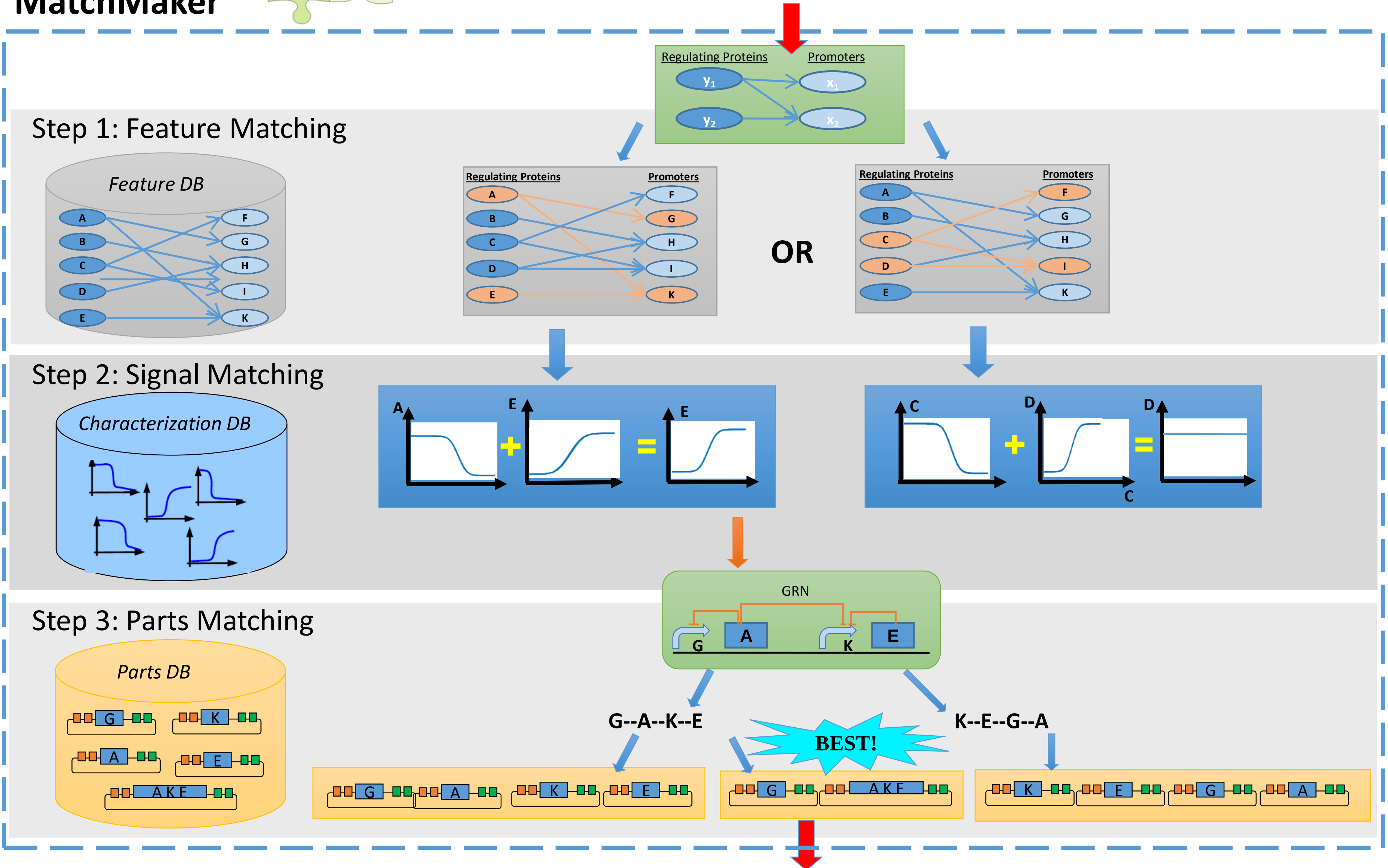
XML

Input to **MatchMaker** (next step in the tool-chain)

CLOTHO

MatchMaker

BioCompiler (previous step in tool-chain) **output**



Step 1: Feature Matching

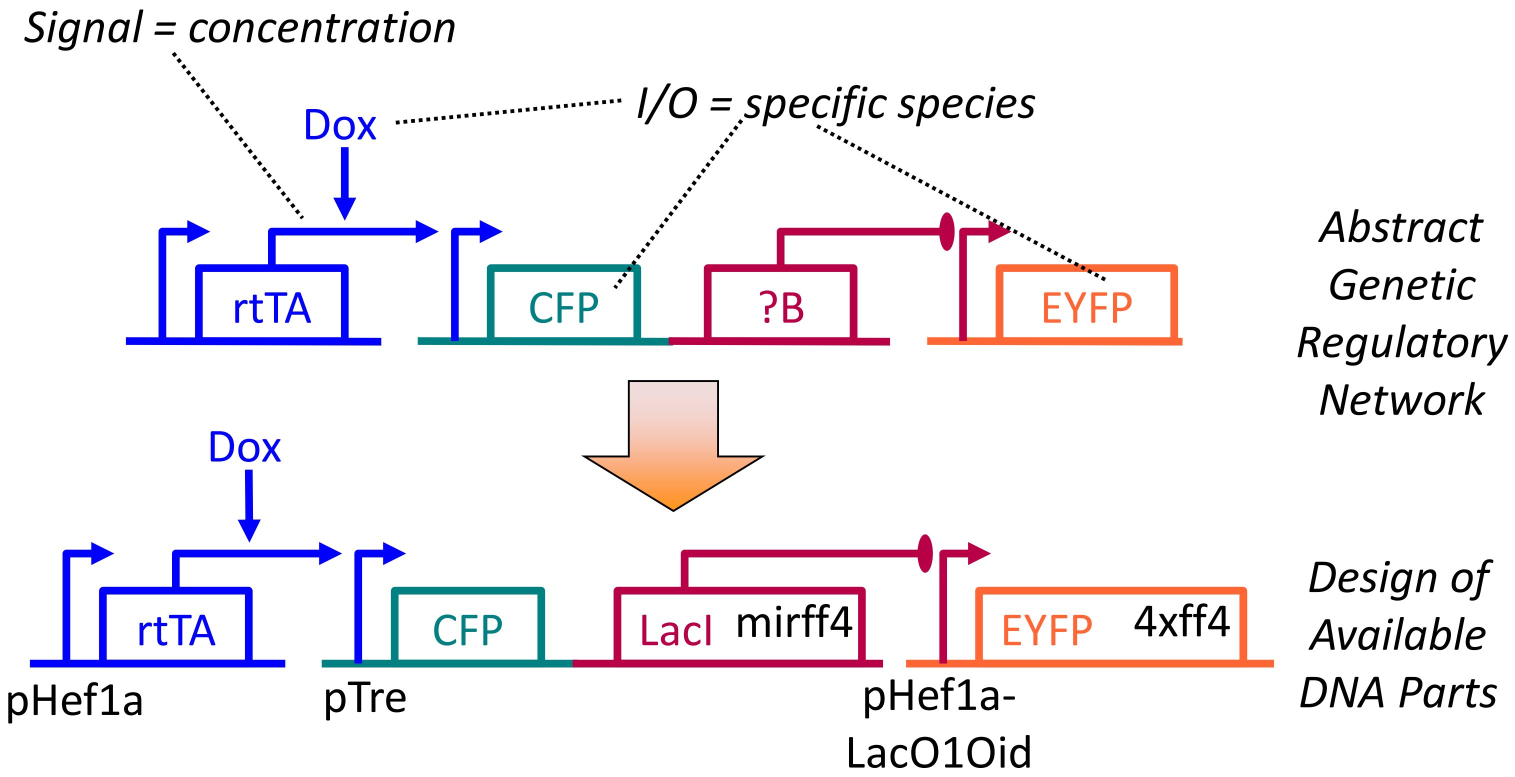
Step 2: Signal Matching

Step 3: Parts Matching

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

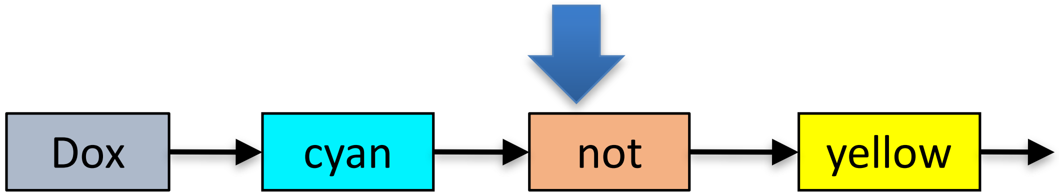
Problem: DNA Part Selection

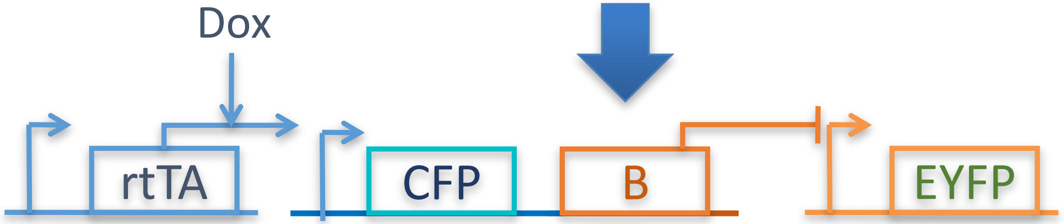
Transcriptional boolean logic networks:

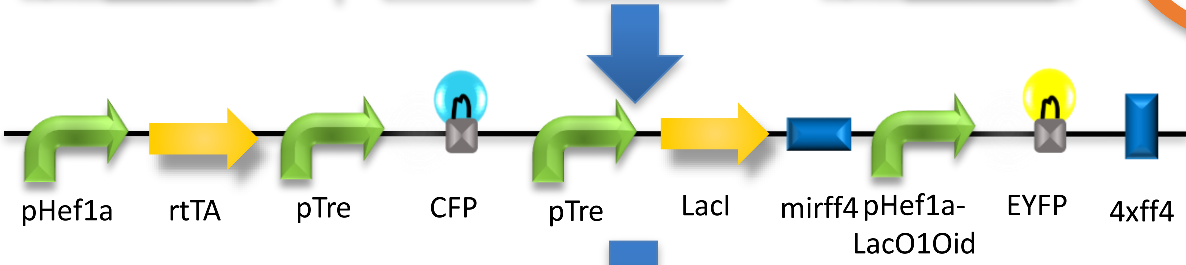


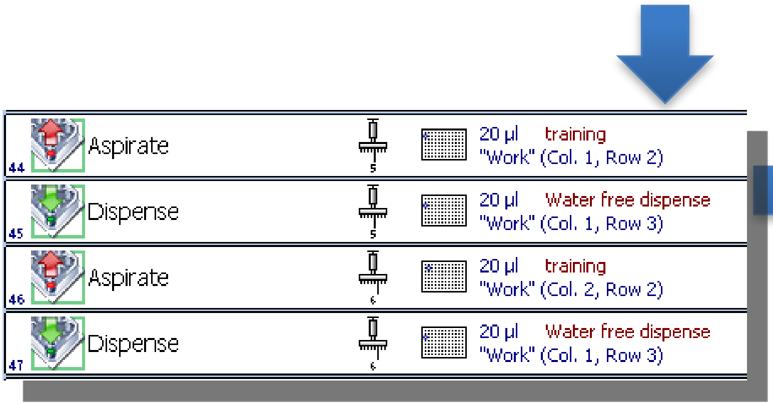
Solution: Feature Mapping + Signal Matching

```
(yellow (not (cyan (Dox))))
```

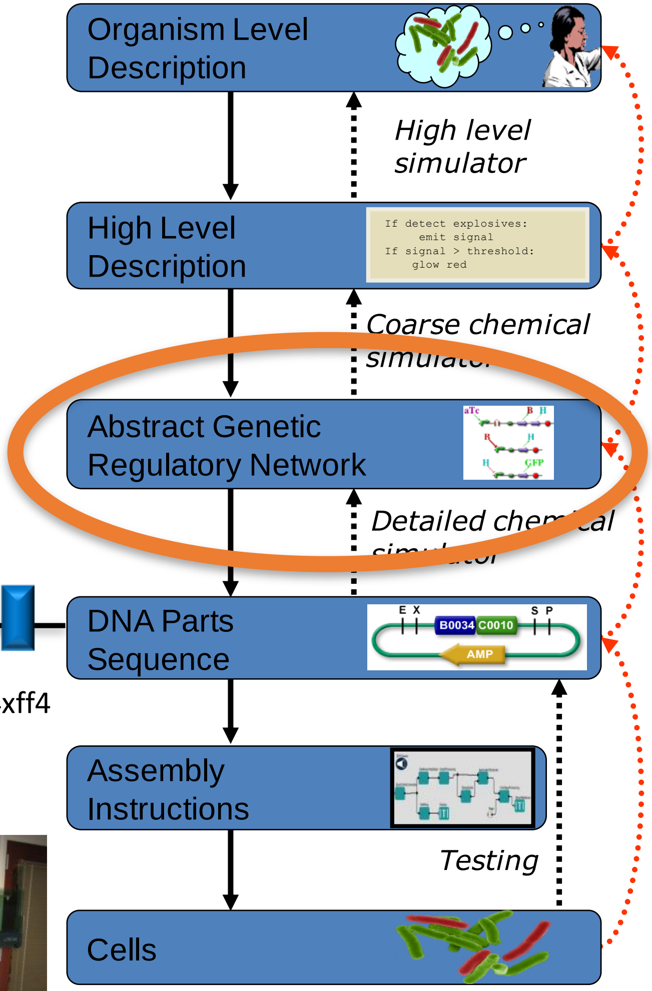






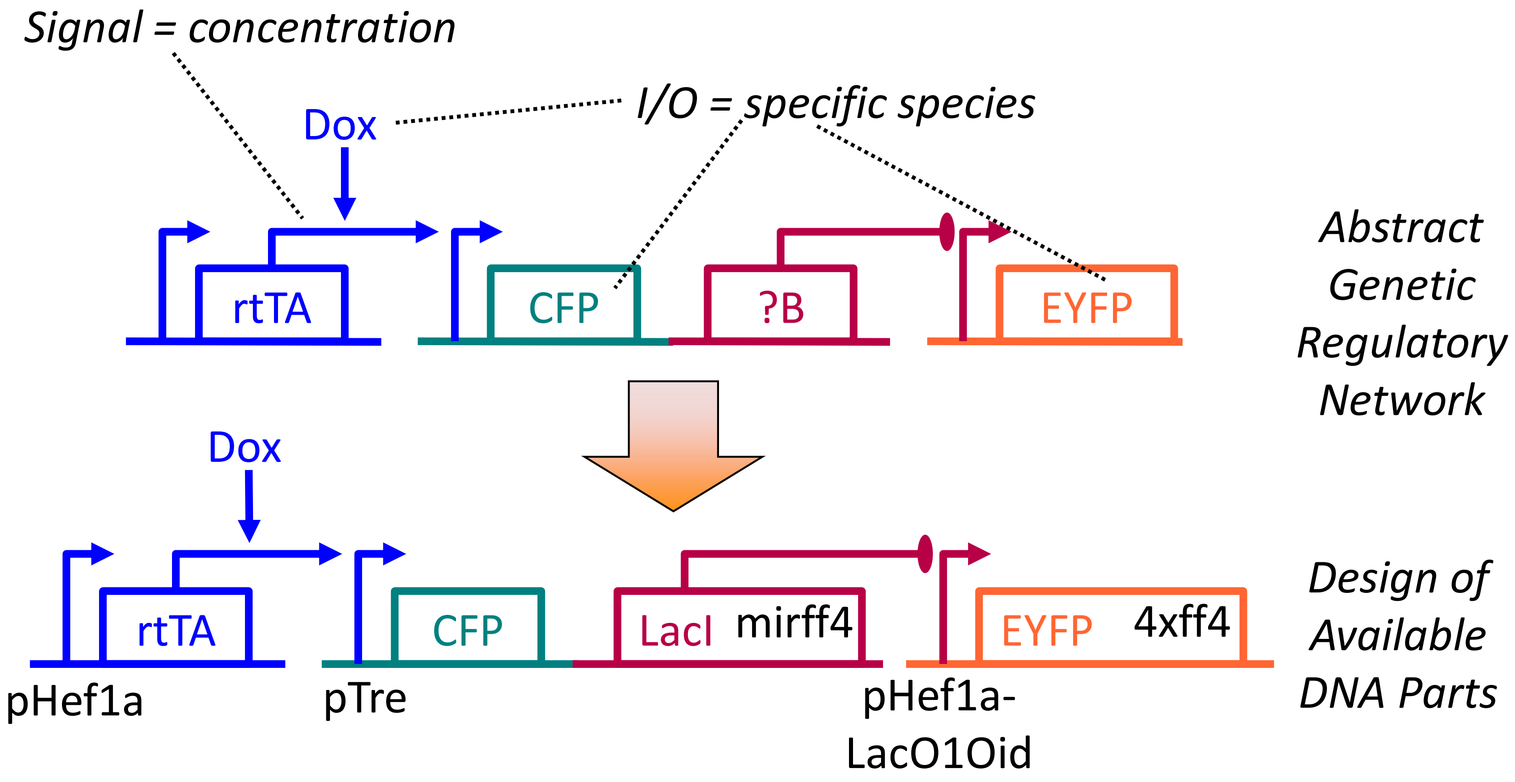


44	Aspirate	20 µl training	"Work" (Col. 1, Row 2)
45	Dispense	20 µl Water free dispense	"Work" (Col. 1, Row 3)
46	Aspirate	20 µl training	"Work" (Col. 2, Row 2)
47	Dispense	20 µl Water free dispense	"Work" (Col. 1, Row 3)



Problem: DNA Part Selection

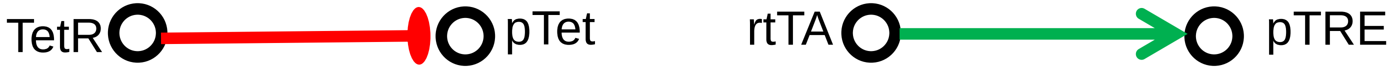
Transcriptional boolean logic networks:



Solution: Feature Mapping + Signal Matching

Feature Mapping: assign features to variables

- Feature: a DNA sequence responsible for a specific biochemical behavior
- Feature database: a collection of features and transcription factors with the regulatory relationships between them



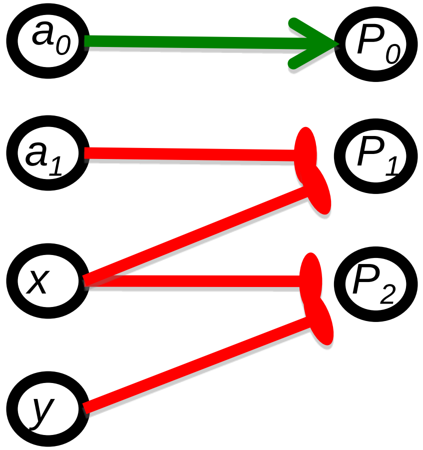
- Feature mapping: Given an AGRN G and a feature database H, find a network of promoters and transcription factors in H that is isomorphic to G.
  - **isomorphism: a strict correspondence between arcs in G and the solution**

Feature Mapping: assign features to variables

AGRN

Transcription Factors

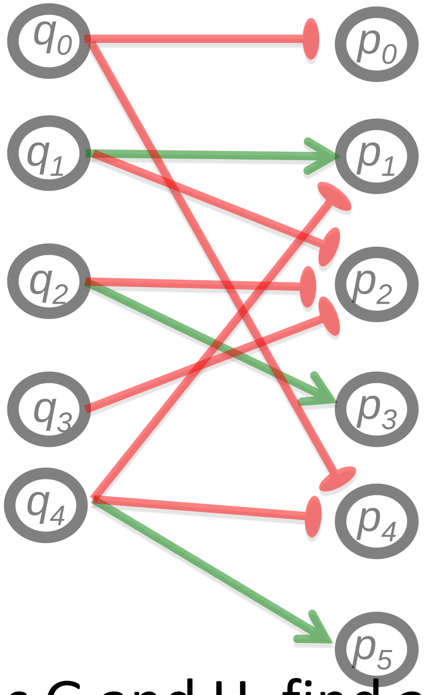
Promoters



Feature Database

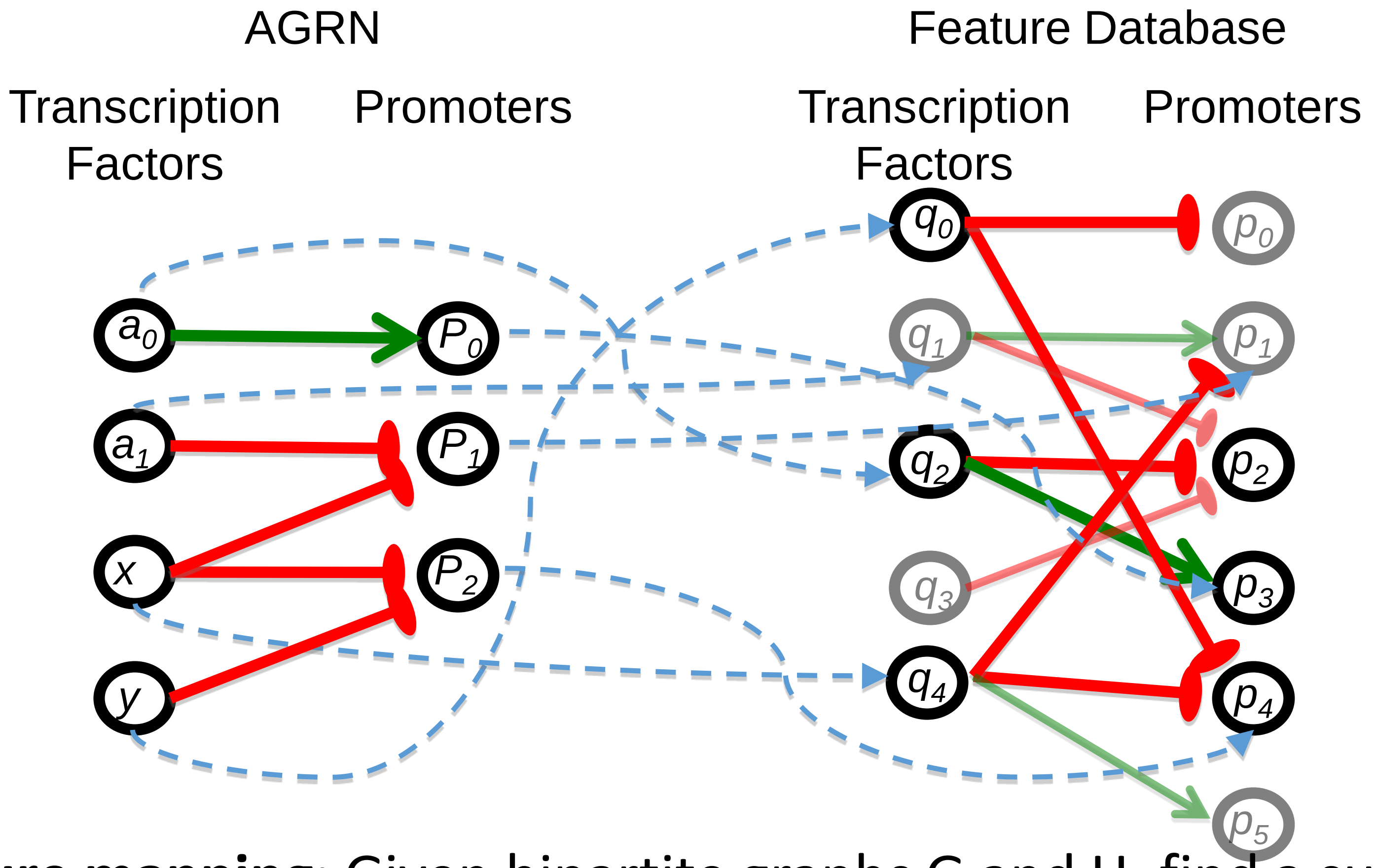
Transcription Factors

Promoters



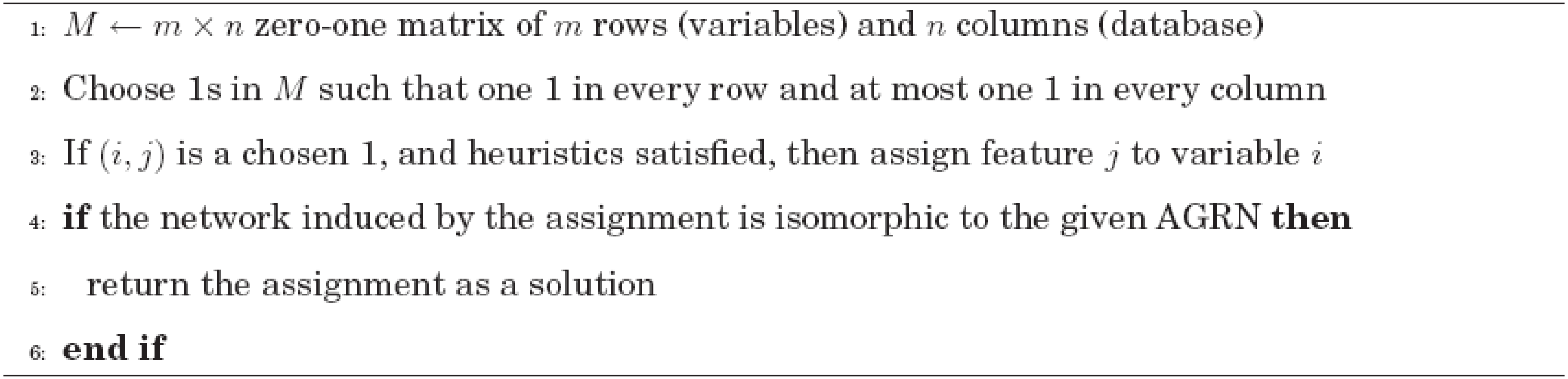
- **Feature mapping**: Given bipartite graphs G and H, find a subgraph of H that is strictly isomorphic to G.

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.
- This problem is NP-complete: unless an astonishing mathematical hypothesis—P=NP—is true, there is no fast algorithm for solving this problem.

Solution: Feature Mapping with heuristic guided search

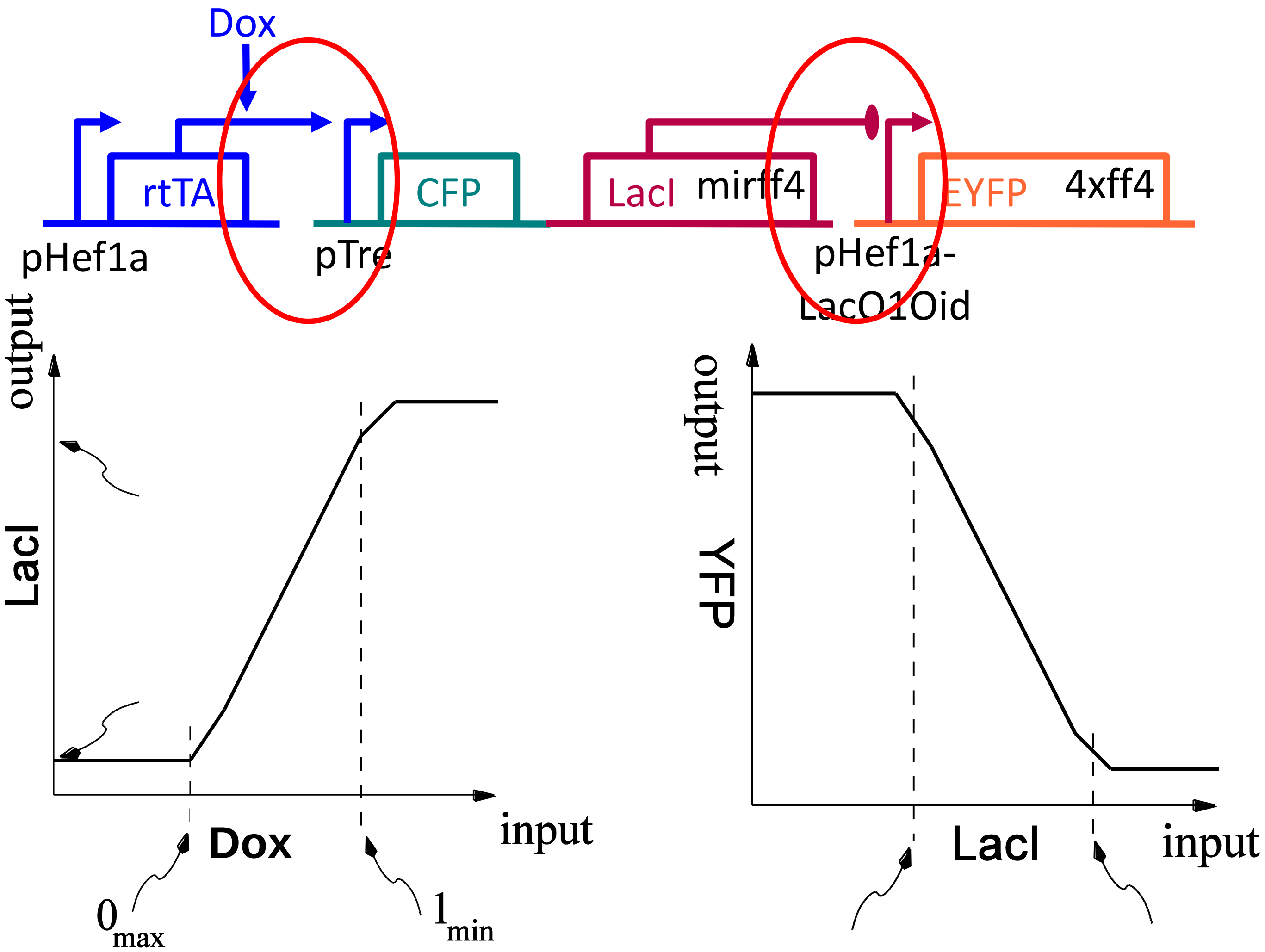


```
$M \leftarrow m \times n$ zero-one matrix of $m$ rows (variables) and $n$ columns (database)
Choose 1s in $M$ such that one 1 in every row and at most one 1 in every column
If $(i, j)$ is a chosen 1, and heuristics satisfied, then assign feature $j$ to variable $i$
if the network induced by the assignment is isomorphic to the given AGRN then
  return the assignment as a solution
end if
```

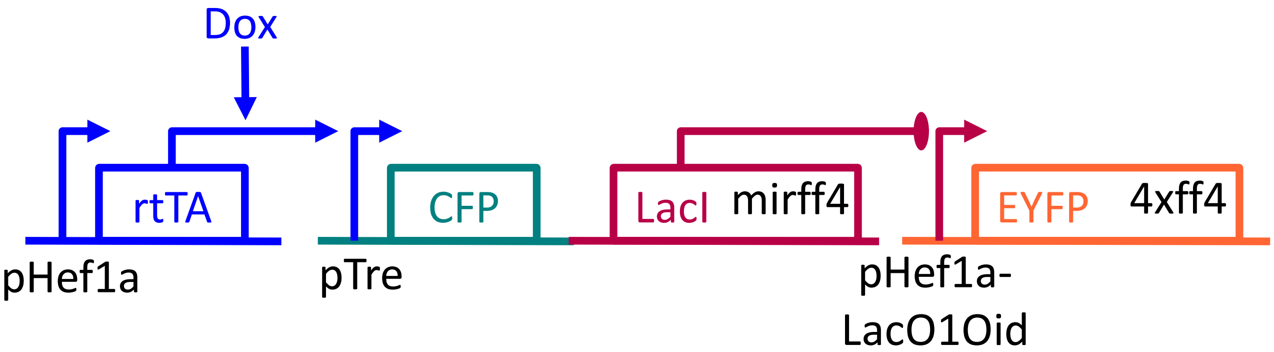
- Two heuristics to reduce search space (Ullman, *JACM 1976)*
- Number of arcs on the variable ≤ number of arcs on the feature
- Number of arcs on neighbors

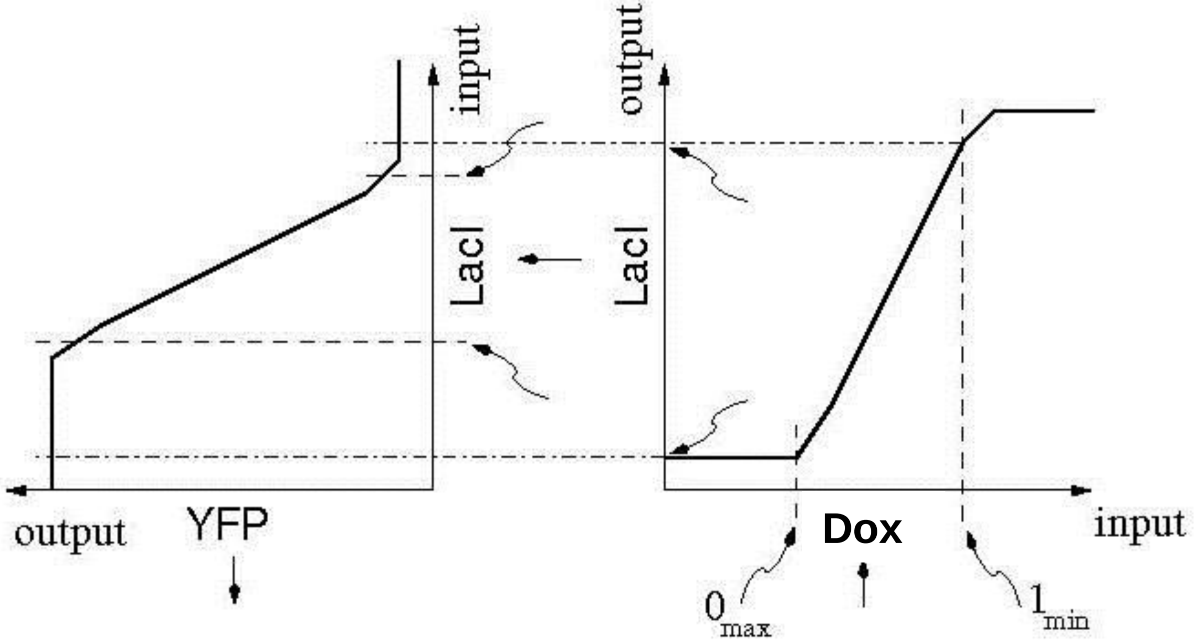
(types must match)

Signal Matching

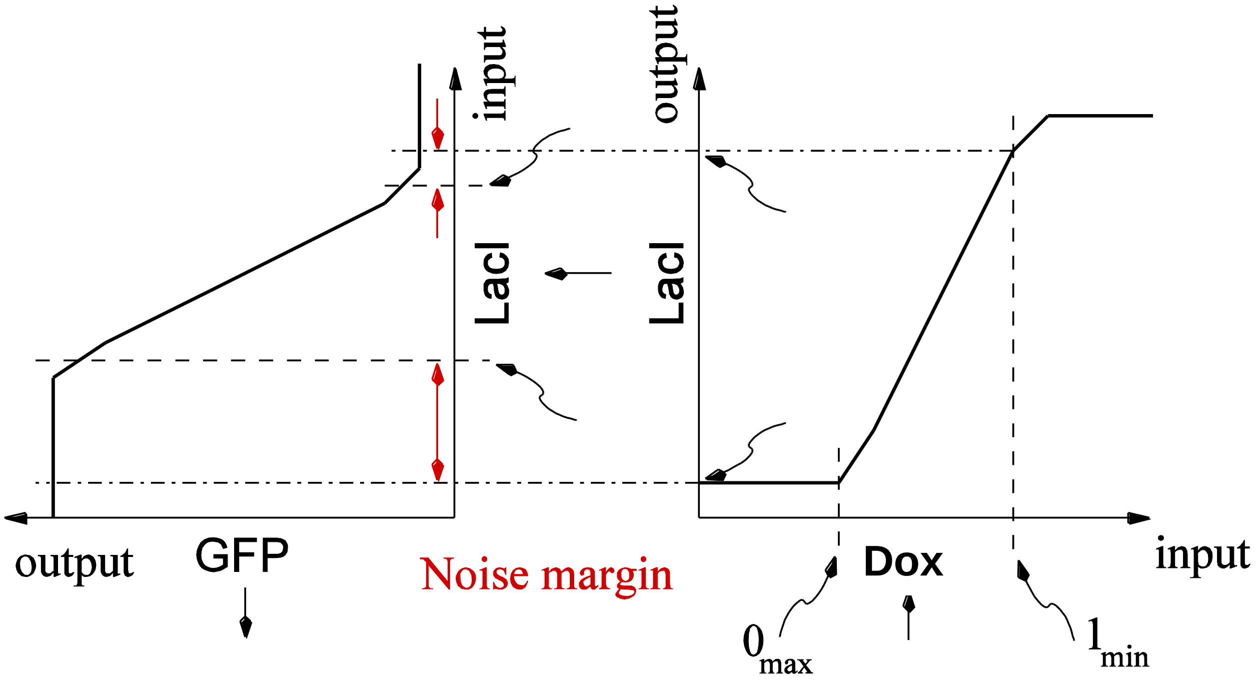


Signal Matching





Signal Matching



Output 0_{max} of Dox/LacI ≤ input 0_{max} of LacI/YFP

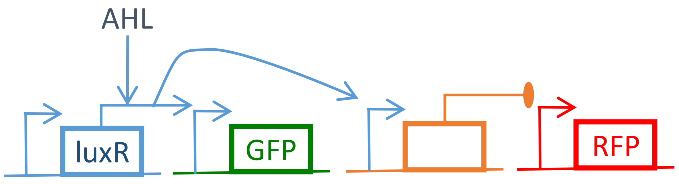
Output 1_{min} of Dox/LacI ≥ input 1_{min} of LacI/YFP

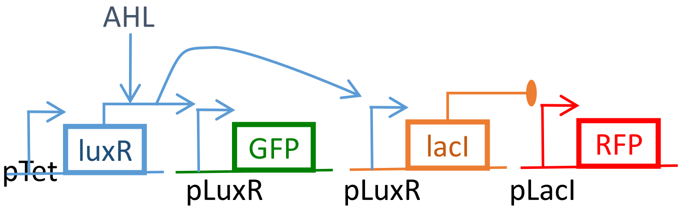
Noise margin of a circuit is the minimum noise margin over all junctions
Extend earlier algorithm with greedy search for noise margin maximization

“Cross Compilation” Possibilities

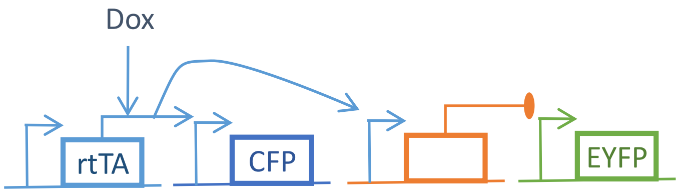
(color1 (not (color2 (sense 1))))

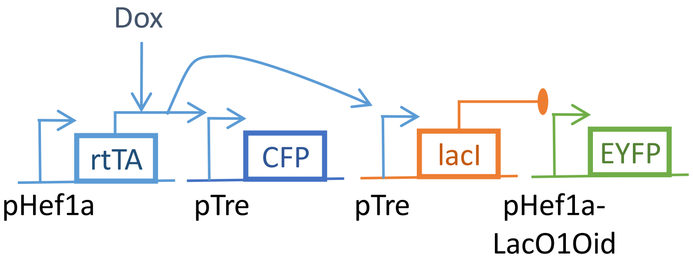
(red (not (green (AHL))))





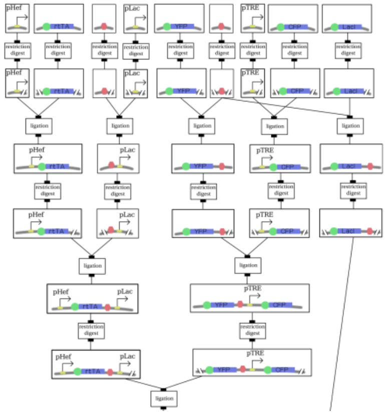
(yellow (not (cyan (Dox))))





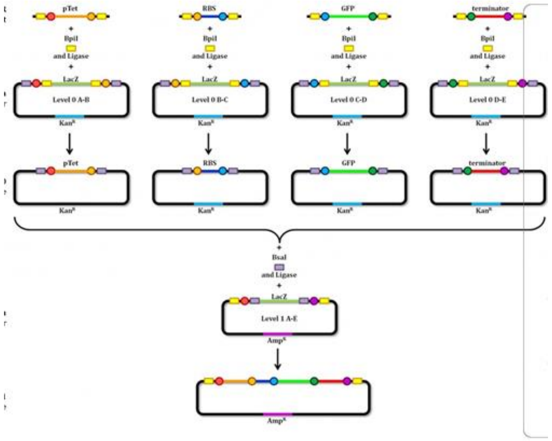
BioCompiler (high level To AGRN)

MatchMaker (AGRN to GRN)



BioBrick (GRN to assembly Tree)

Gateway/ Gibson/MoClo (GRN to multiway Assembly)


pTet
RBS
GFP
terminator
BpiI and Ligase
LacZ
Level 0 A-B
Level 0 B-C
Level 0 C-D
Level 0 D-E
BsaI and Ligase
Level 1 A-E