Defining Domain-Specific DNA Programming Languages Using GenoCAD 2.2 Grammar Editor

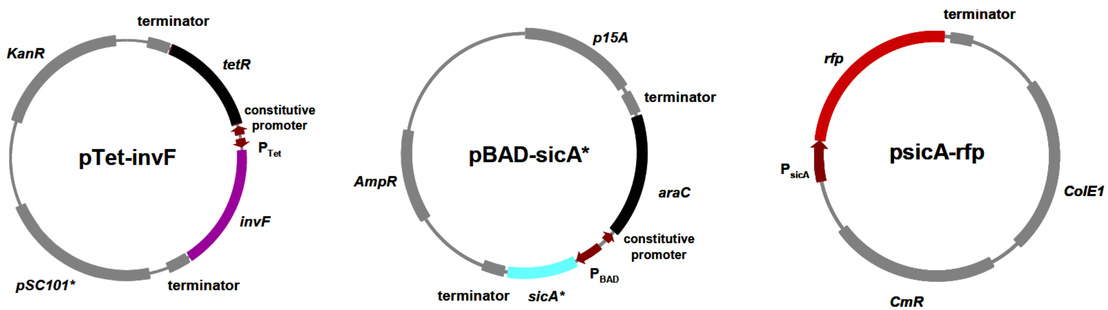
Jean Peccoud, Laura Adam, Mandy Wilson

Virginia Bioinformatics Institute

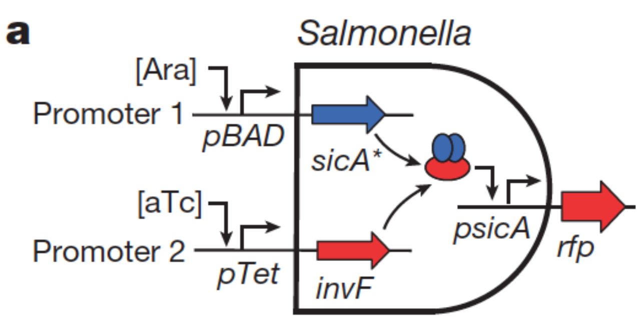
Virginia Tech

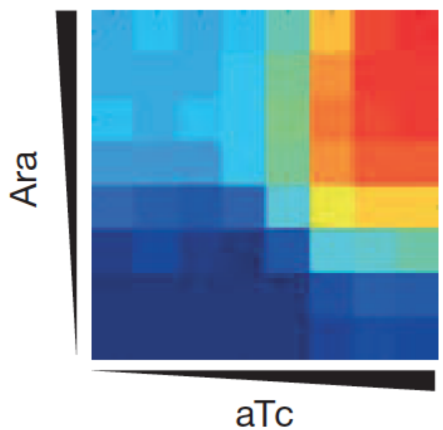
Synthetic DNA molecules as programs

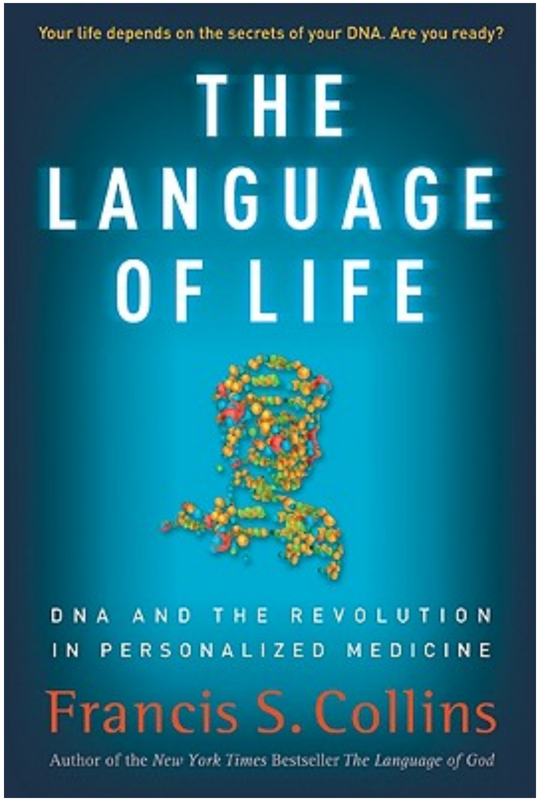
Program



Behavior

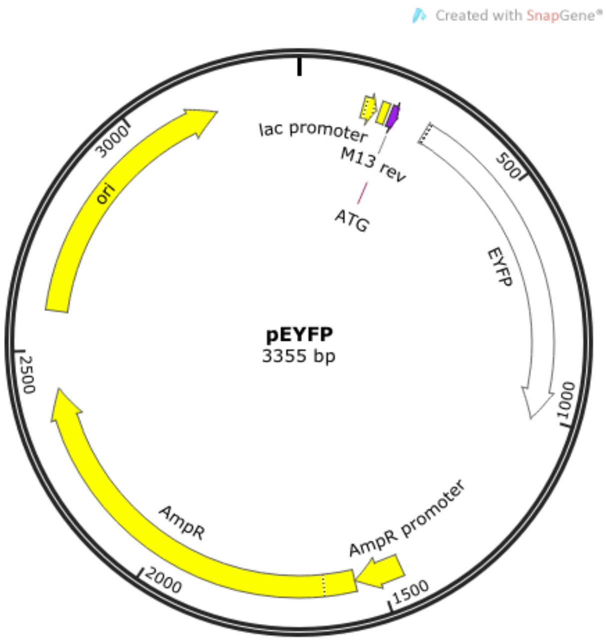




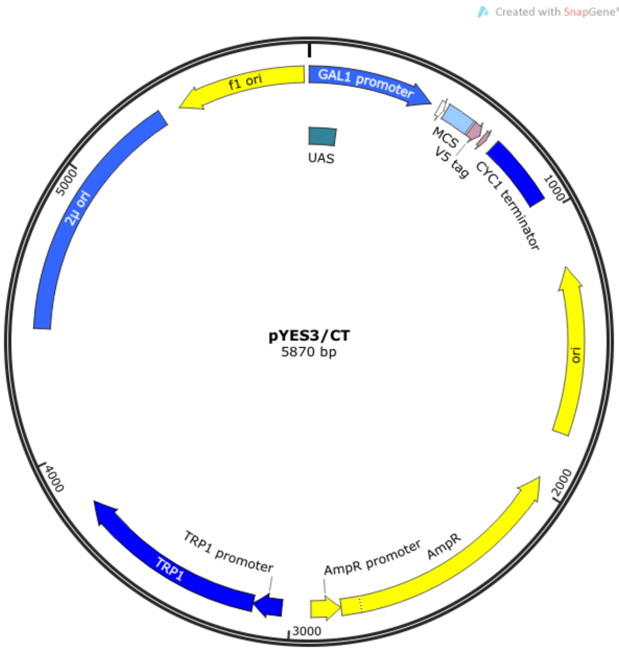


Moon et all (2012) **Genetic programs constructed from layered logic gates in single cells** Nature (491) 249-53

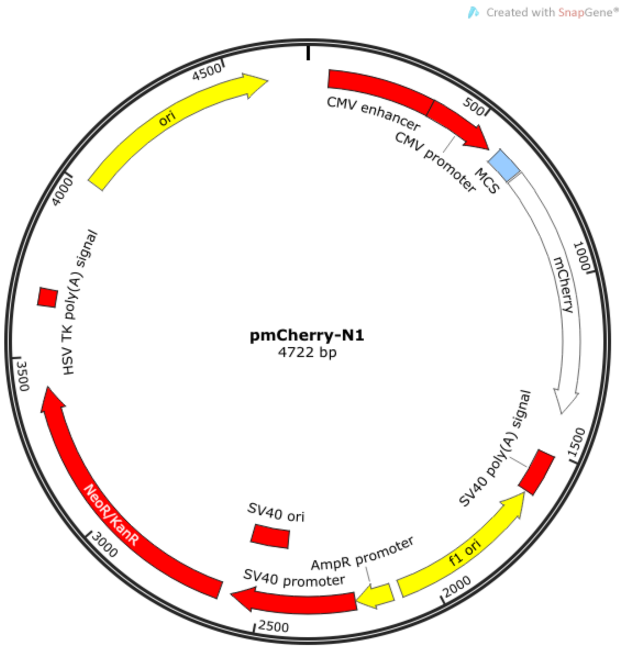
Different Dialects
Plasmids for different purposes



Bacterial



Yeast



Mammalian

Domain Specific Languages

No universal language

- C, SQL, HTML, Flash

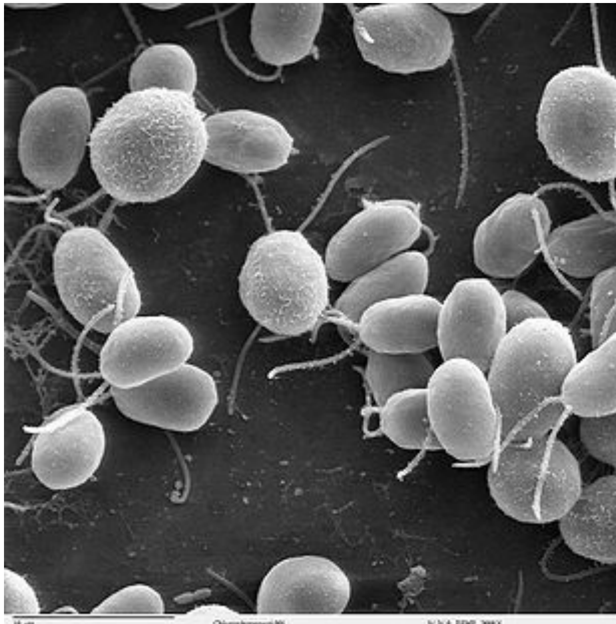
Languages express design strategies

- Domain-specific
  - bacterial, yeast, mammalian
- Project-specific
  - gene therapy, synthetic biology
- Organization-specific
  - know-how, intellectual property

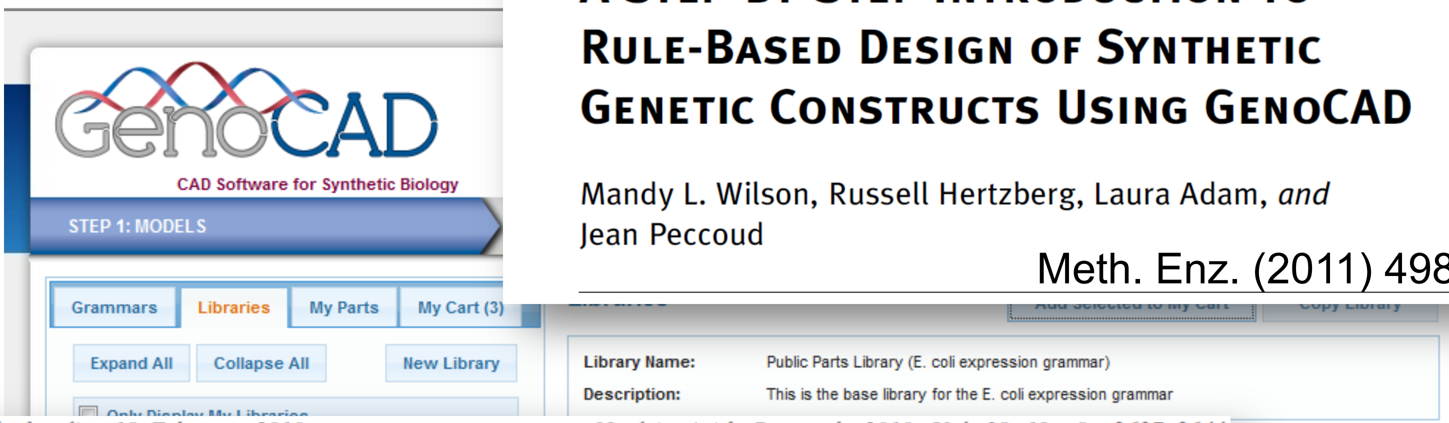
Empower end-users to develop their own DSL

- Gene expression in the chloroplast of microalgae

Pieter Bruegel the Elder - The Tower of Babel



Chlamydomonas reinhardtii



CHAPTER EIGHT

A Step-by-Step Introduction to Rule-Based Design of Synthetic Genetic Constructs Using GenoCAD

Mandy L. Wilson, Russell Hertzberg, Laura Adam, *and* Jean Peccoud

Meth. Enz. (2011) 498

Published online 18 February 2010 *Nucleic Acids Research, 2010, Vol. 38, No. 8* ***2637–2644***
doi:10.1093/nar/gkq086

GenoCAD for iGEM: a grammatical approach to the design of standard-compliant constructs

Yizhi Cai, Mandy L. Wilson and Jean Peccoud*

Virginia Bioinformatics Institute, Virginia Polytechnic Institute and State University, Washington St MC 0477, Blacksburg VA 24061, USA

Received December 13, 2009; Revised January 28, 2010; Accepted February 1, 2010

W40–W47 Nucleic Acids Research, 2009, Vol. 37, Web Server issue *Published online 8 May 2009*
doi:10.1093/nar/gkp361

Writing DNA with GenoCAD™

Michael J. Czar, Yizhi Cai and Jean Peccoud*

Virginia Bioinformatics Institute at Virginia Polytechnic and State University, Blacksburg, VA 24061, USA

Received March 13, 2009; Revised April 21, 2009; Accepted April 22, 2009

2013/07/24

Example of formal grammar

A **grammar** is a:

- Set of **rules** describing how to form **sentences** from a **language's vocabulary**

R1: Sentence ⟶ Subject + Verb + Object

R2: Subject ⟶ NounPhrase

R3: Object ⟶ NounPhrase

R4: NounPhrase ⟶ NounPhrase + Modifier

R5: Modifier ⟶ PrepositionalPhrase

How to build a sentence?

Sentence

Subject | Verb | Object R1

R2 NounPhrase | NounPhrase R3

NounPhrase | Modifier R4

Prepositional Phrase R5

R1: Sentence ⟶ Subject + Verb + Object

R2: Subject ⟶ NounPhrase

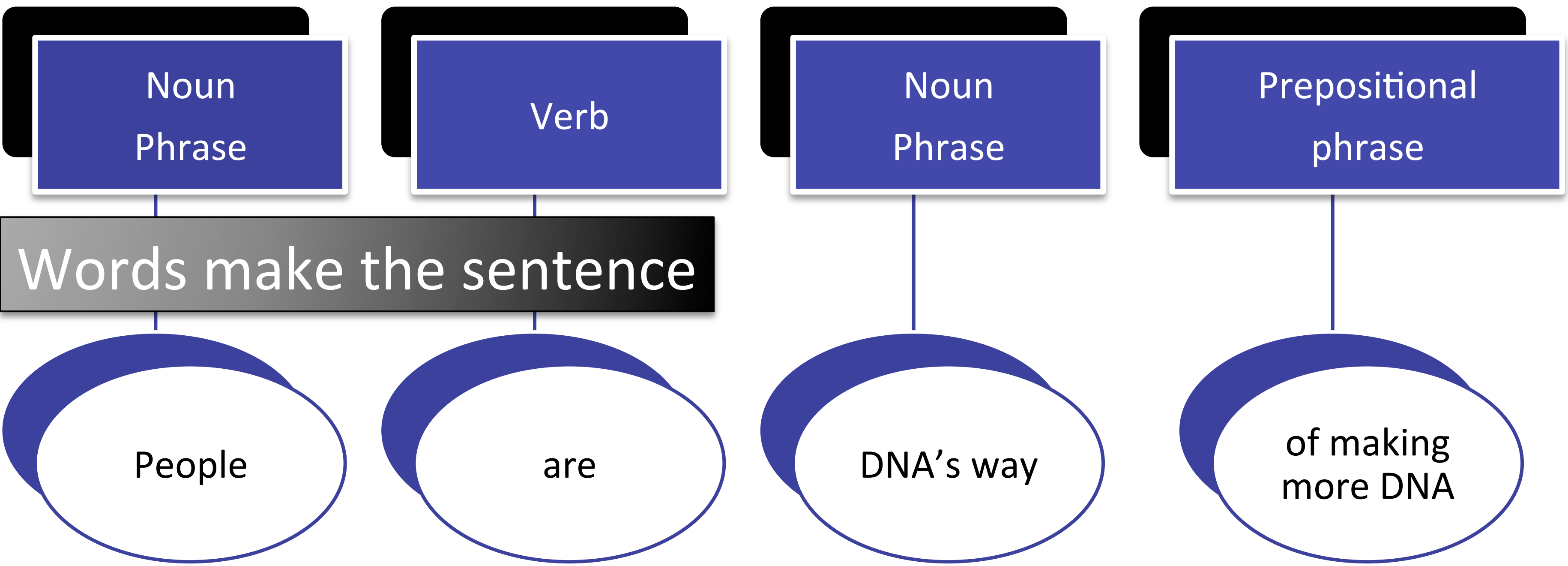
R3: Object ⟶ NounPhrase

R4: NounPhrase ⟶ NounPhrase + Modifier

R5: Modifier ⟶ PrepositionalPhrase

Rules make the structure

Words make the sentence

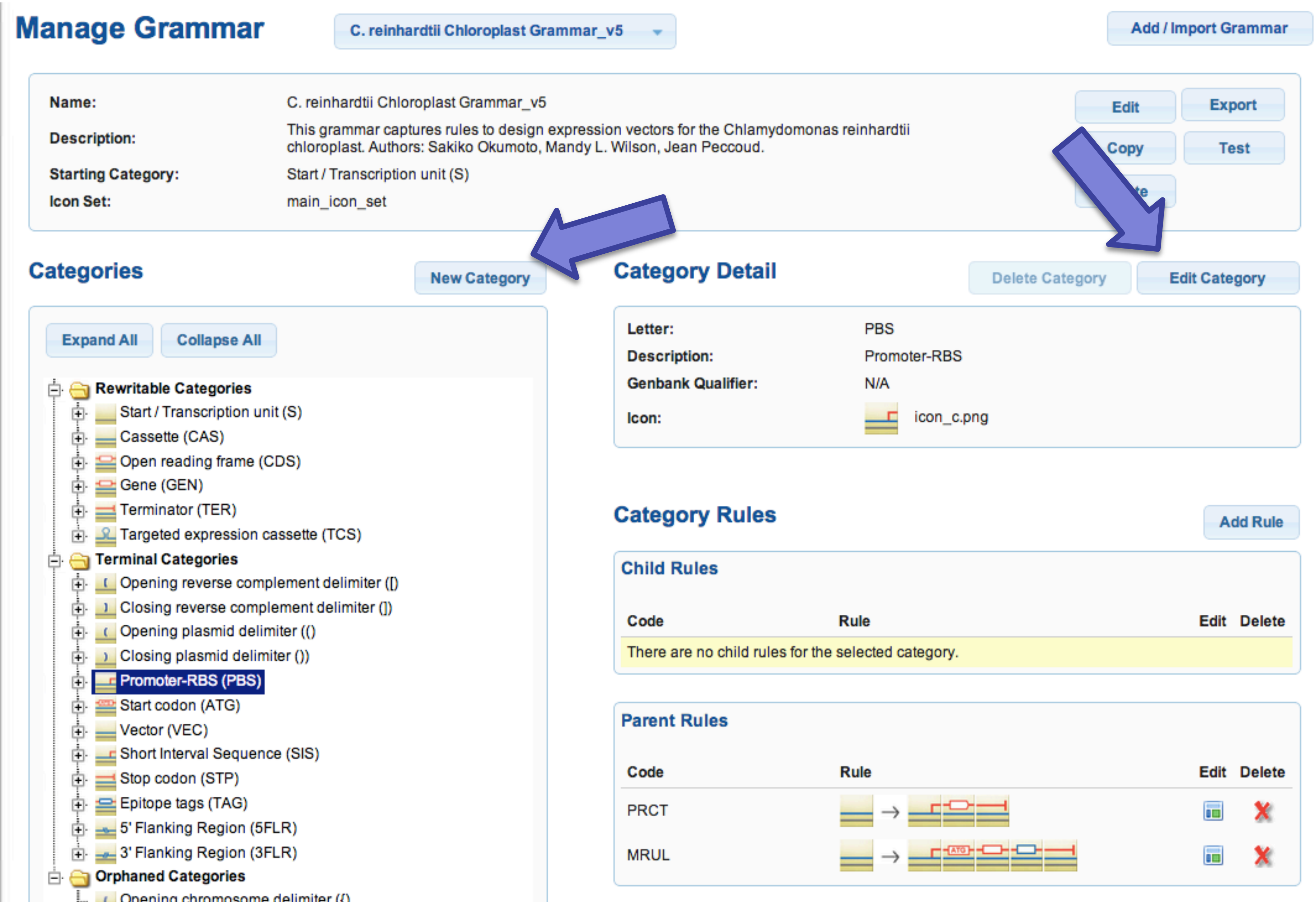


~ Edward O. Wilson, 1975

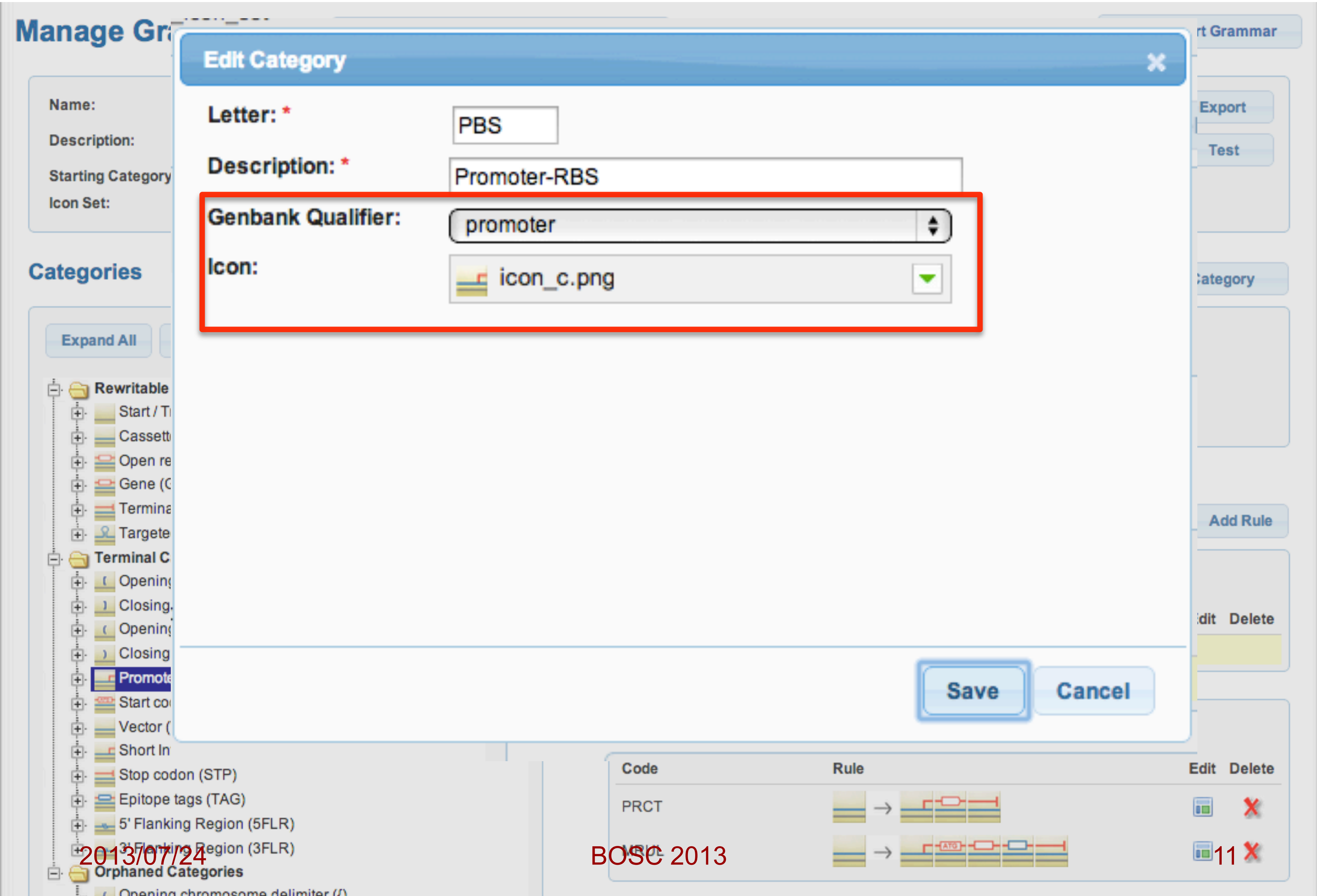
Identify and Define Categories

Category	Definition
5FLR / 3FLR	5’ / 3’ Flanking region for homologous recombination
SIS	Short Interval Sequences used to make polycistronic cassettes
STP	Stop codon
ATG	Start codon
GEN	Gene or protein domain. By convention does not include start and stop codons.
CDS	Open reading frame composed of several protein domains. Does not include start and stop codons.
TAG	Epitope tags. By convention does not include Start or Stop codons.
PBS	Sequence associated with the initiation of transcription and translation.
TCS	Targeted expression cassette. Expression cassette flanked with two adjacent genomic sequences for homologous recombination.
CAS	Expression cassette delimited by a promoter in 5’ and a transcription terminator in 3’.

Grammar Editor – Add/Edit Categories

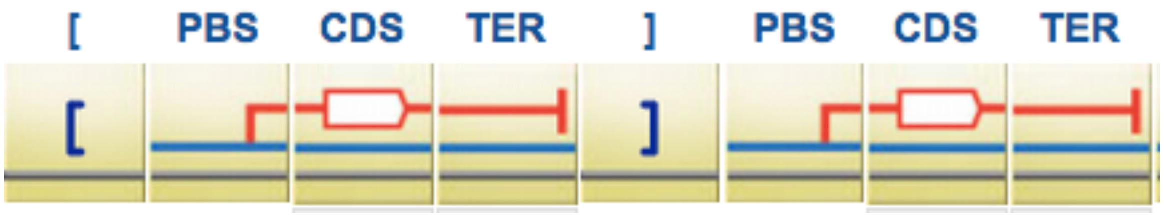


Grammar Editor – Add/Edit Categories



Reserved Categories

Category	Definition
[and]	Negative orientation delimiters
(and)	Plasmid delimiters
{ and }	Chromosome delimiters

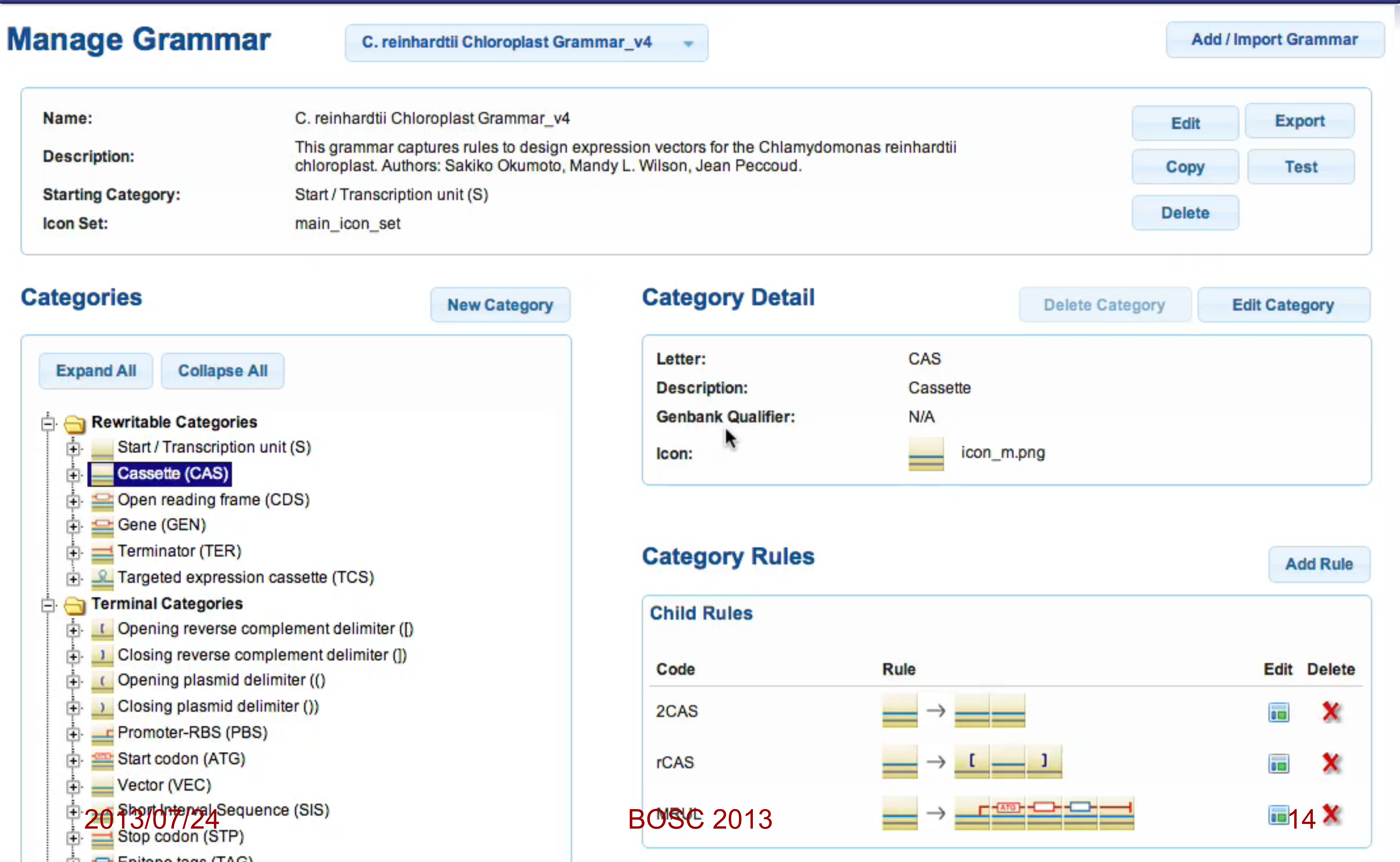


Define rewriting rules

Code	Rule	Comment
CAS	S -> TCS	This rule is used to design only one expression cassette
1PLAS	S -> (VEC TCS)	This rule is used to specify the expression cassette along with the vector where it is inserted. The output is the entire plasmid sequence.
2PLAS	S -> (VEC TCS) (VEC TCS)	This rule is for designs that involve two plasmids.
TGS	TCS -> 5FLR CAS 3FLR	Specifies the flanking regions for homologous recombination.
PRCT	CAS-> PBS CDS TER	A gene expression cassette is composed of a promoter, open reading frame, and a transcription terminator.
2CAS	CAS -> CAS CAS	This rule makes it possible to have more than one expression cassette on a construct.
rCAS	CAS -> [CAS]	This rule is used to specify that the cassette is coded on the negative strand.
2CDS	CDS -> CDS SIS CDS	This rule makes it possible to design polycistronic constructs.
SGEN	CDS -> ATG GEN STP	The open reading frame is composed of a single gene flanked by a start and stop codon.
TGEN	GEN → GEN TAG	This rule is used to add a tag to a coding sequence. It can be used iteratively to add more than one tag.
2GEN	GEN-> GEN GEN	This rule can be used to fuse two coding sequences that are not tags.

Grammar Editor – Add/Edit Rules

PRCT	CAS-> PBS CDS TER	A gene expression cassette is composed of a promoter, open reading frame, and a transcription terminator.

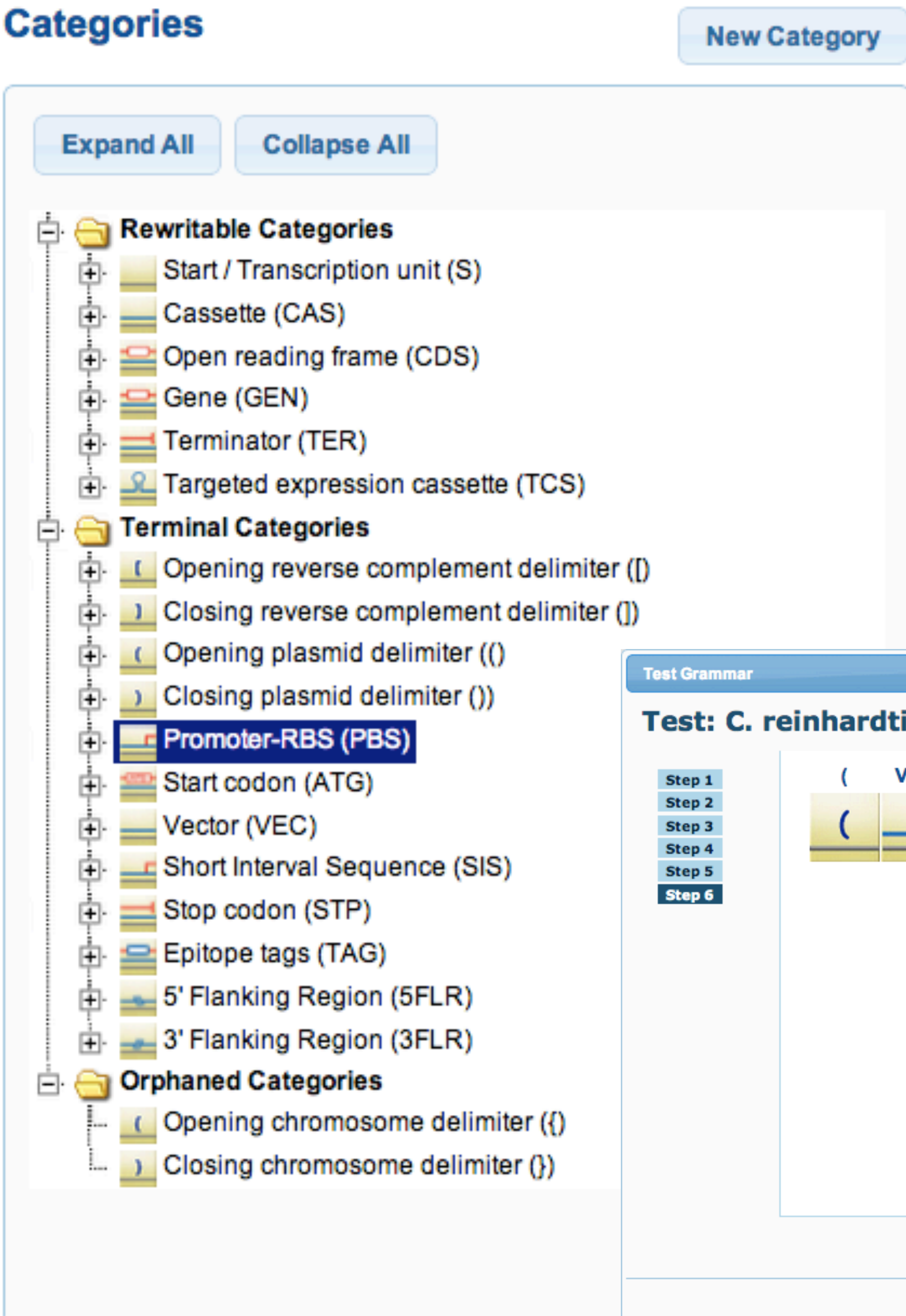


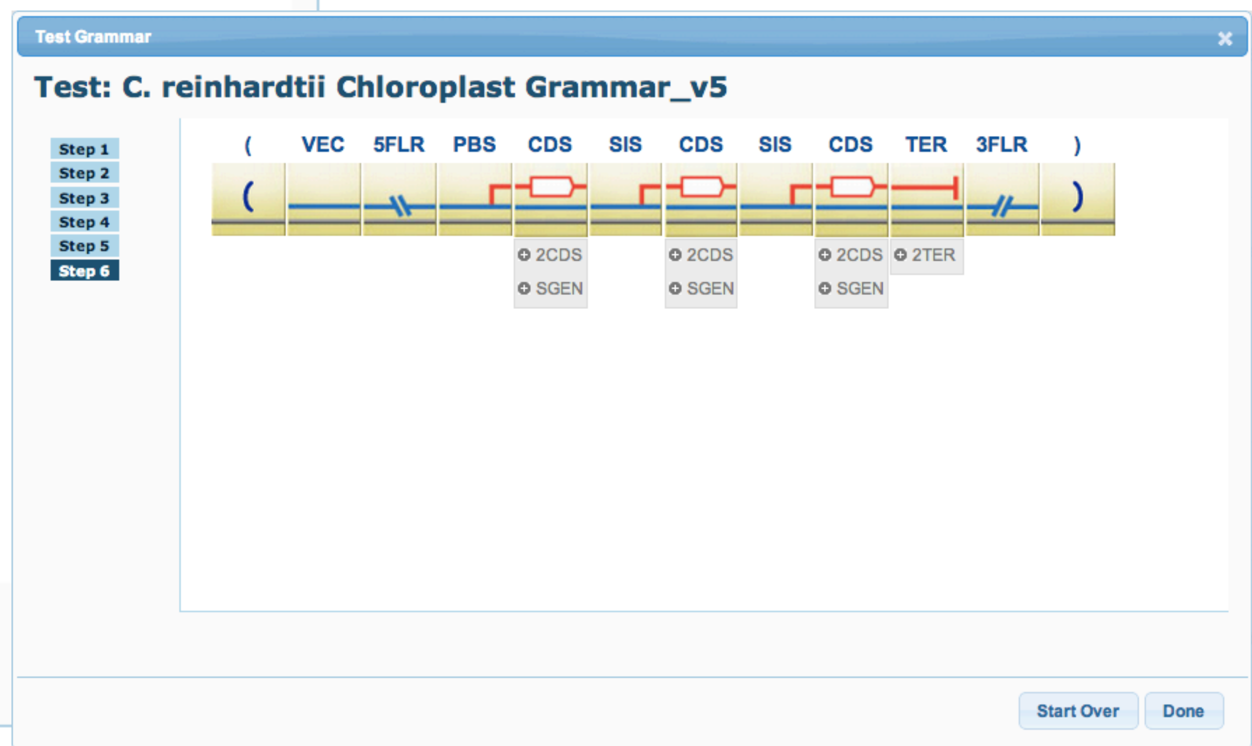
Debugging Tools

Rewritable: used on the left side

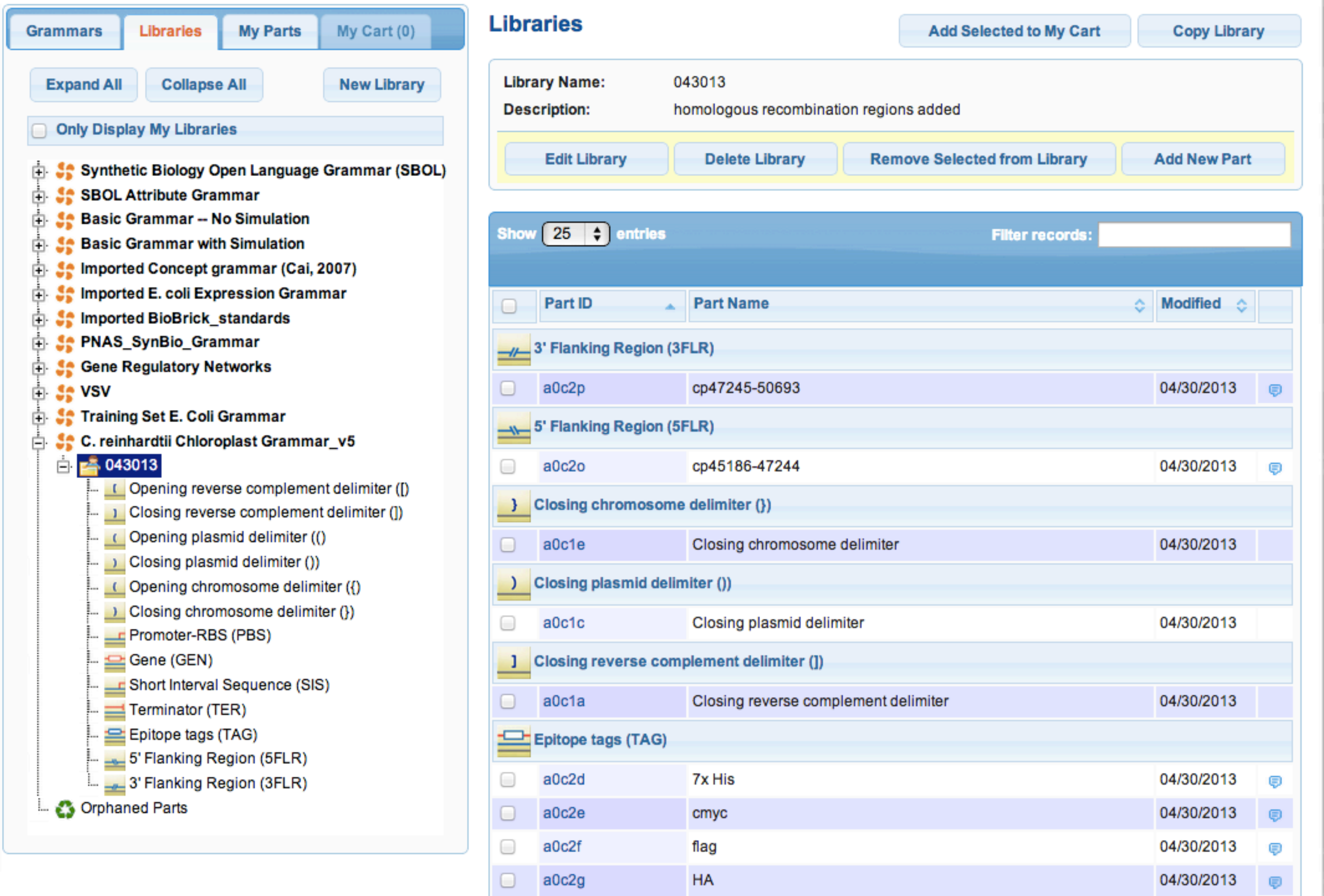
Terminal: used only on the right side

Orphan: not used

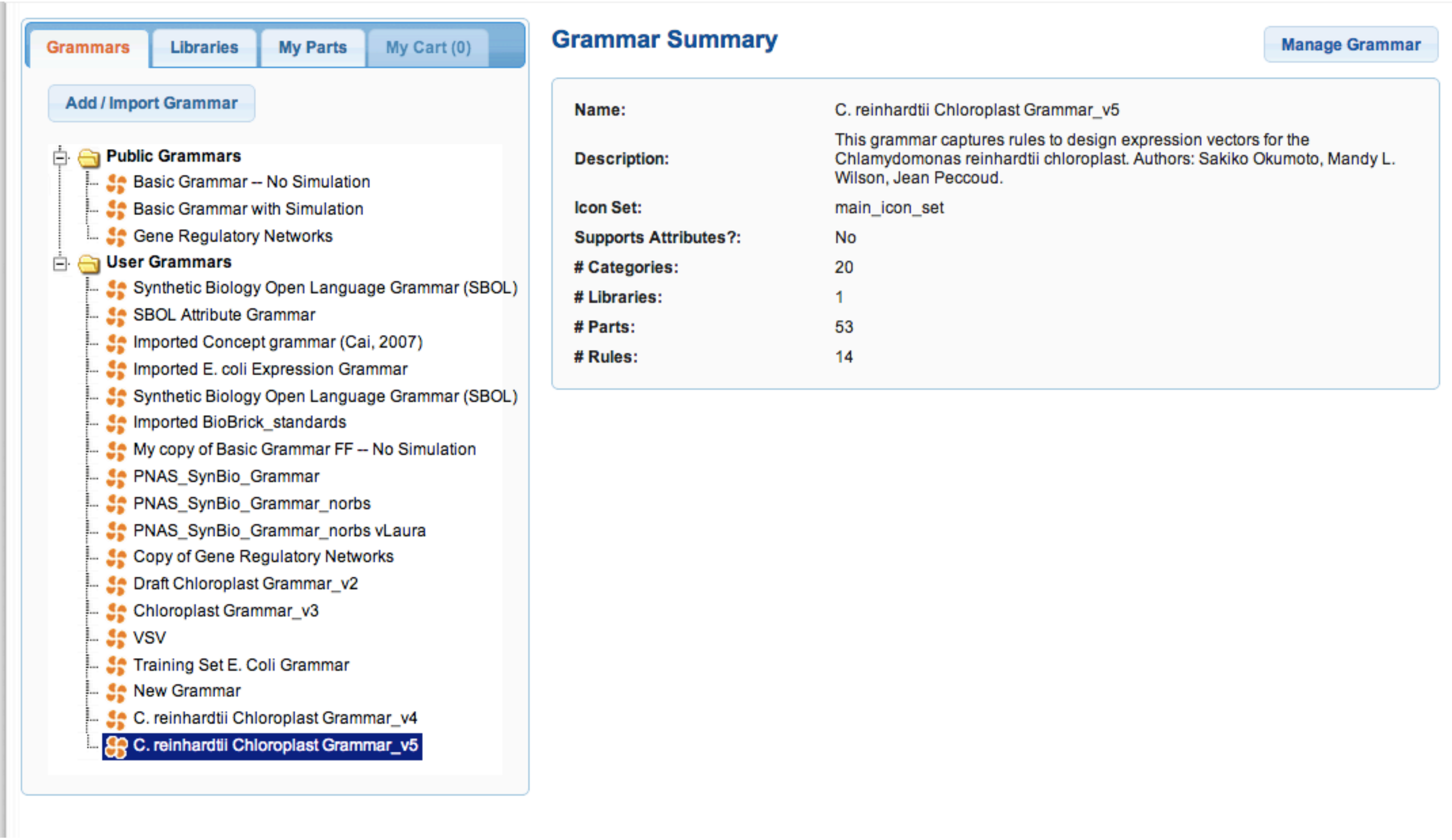




Add parts in libraries



Grammar Summary



Rule-based design of DNA sequences

Part:BBa_J13004

Composite | DNA Available | Experience: Works | Get This Part

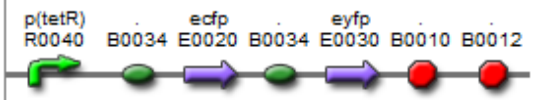
Designed by Jeff Tabor Group: iGEM_UTAustin (2005-06-28)

polycistronic CFP/YFP expression cassette

A single mRNA containing CFP and YFP under the strong RBS B0034 is driven by the tet repressible promoter, R0040.

Sequence and Features

Format: Subparts | Ruler | SS | DS Search: Length: 1689 bp Context: Part only Get selected sequence



Construct

Promoter | Cistron | Terminator

Cistron | Cistron | Terminator

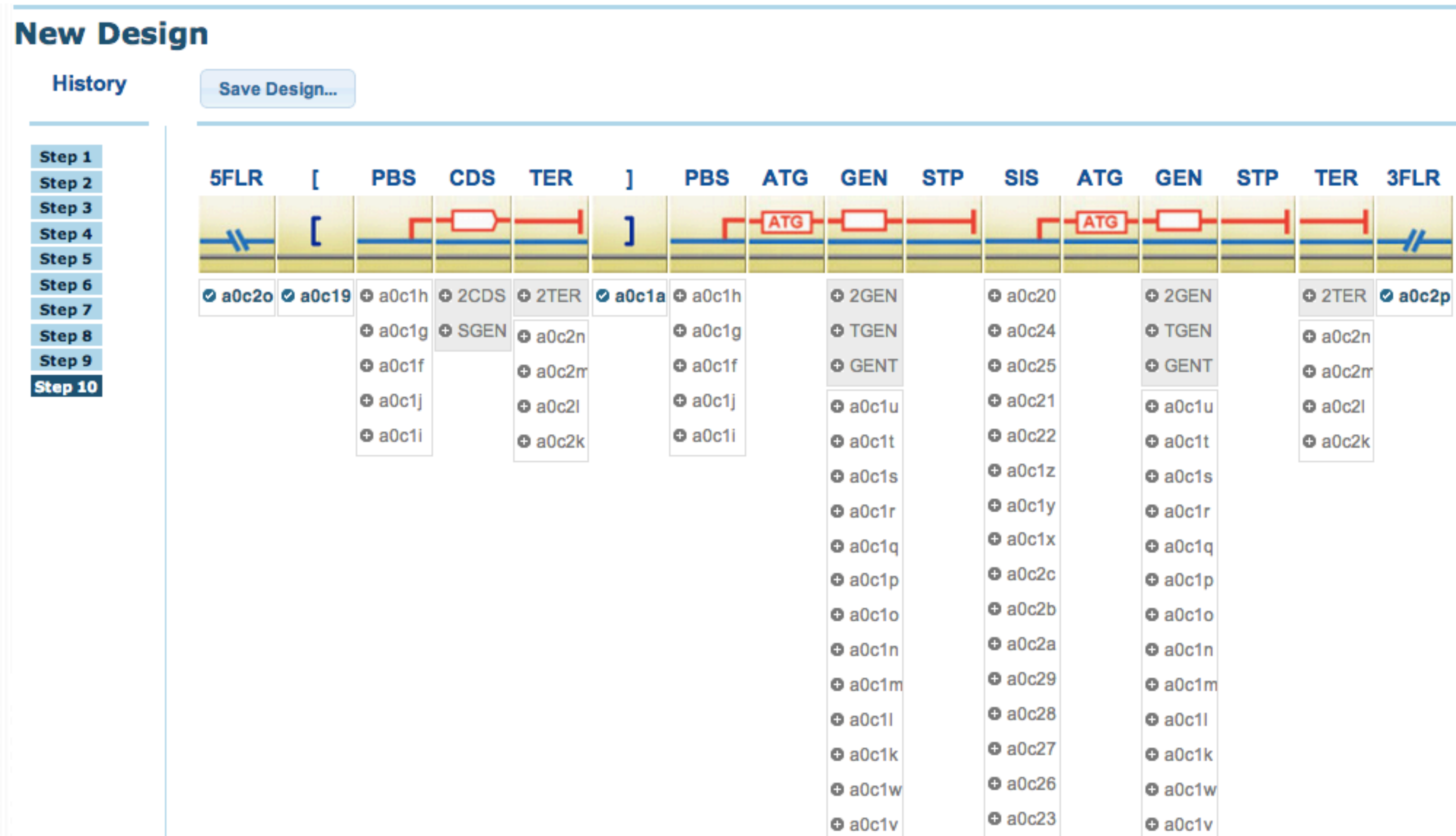
BIOINFORMATICS ORIGINAL PAPER Vol. 23 no. 20 2007, pages 2760–2767
doi:10.1093/bioinformatics/btm446

Systems biology

A syntactic model to design and verify synthetic genetic constructs derived from standard biological parts

Yizhi Cai[1], Brian Hartnett[1], Claes Gustafsson[2] and Jean Peccoud[1,*]

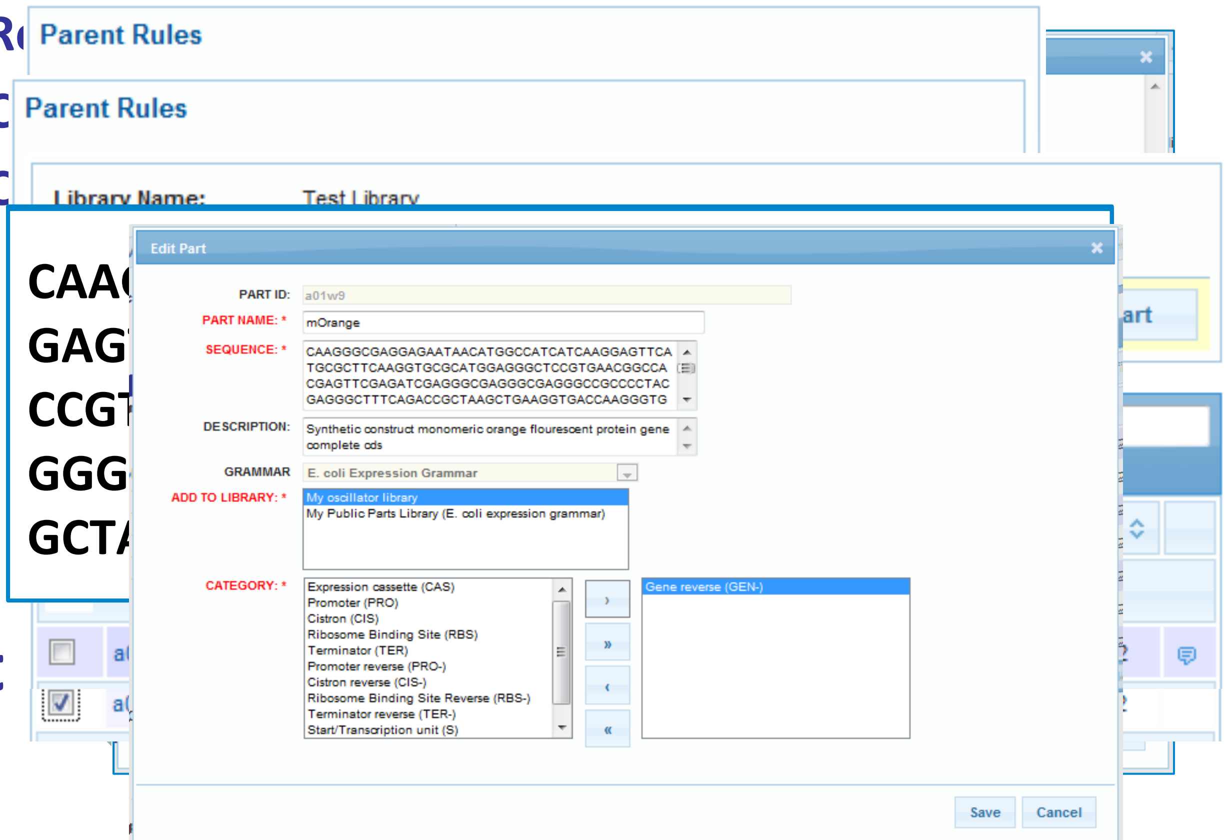
Point-and-click design tool



A dynamic language

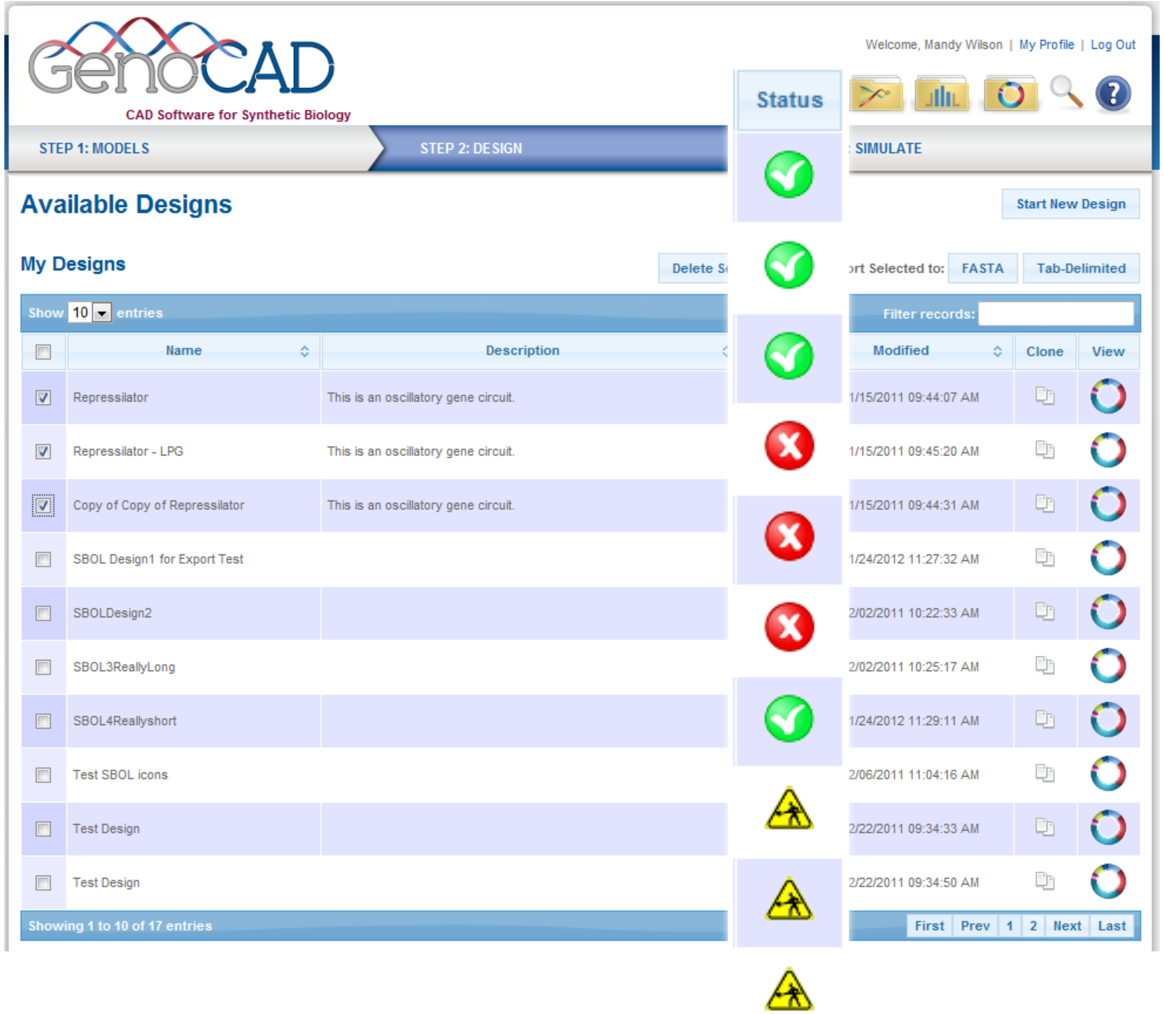
You can change the model …

- Add or Change a Rule in a Grammar
- Delete a Rule from a Grammar
- R
- C
- C



… but … signs?

Available Designs



Different design statuses

Valid – the sequence could be decomposed into its parts, and the parts' categories make up a grammar-sanctioned framework.

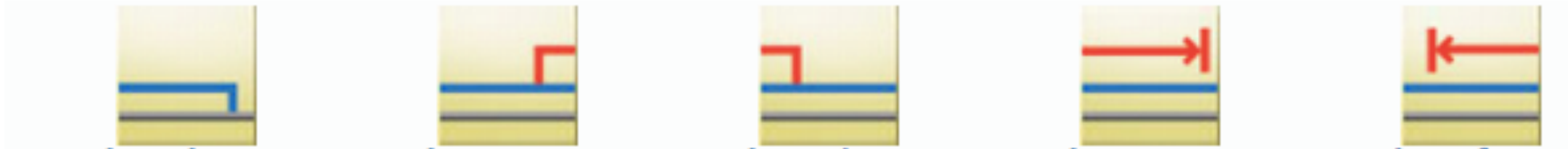
Needs validation – either grammar, part, or library has changed, and the sequence has not been validated since

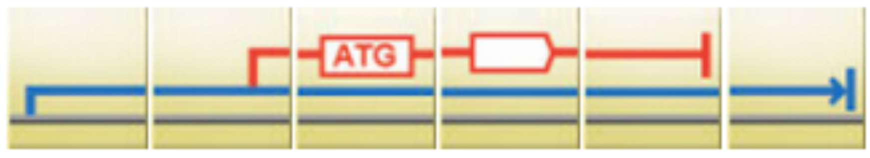
Under construction – design is unfinished, so cannot be compiled.

Out of Date – although design is still valid with respect to grammars and libraries, the parts have changed.

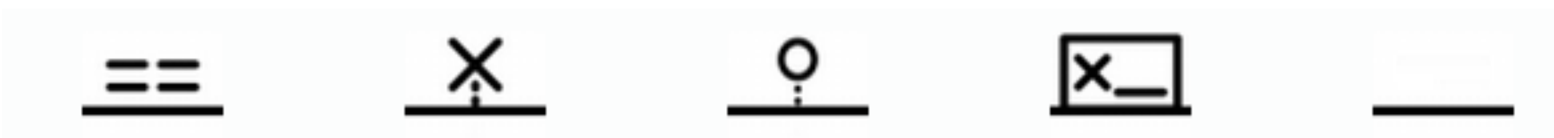
Invalid – the sequence cannot be resolved.

Graphical Language Icons



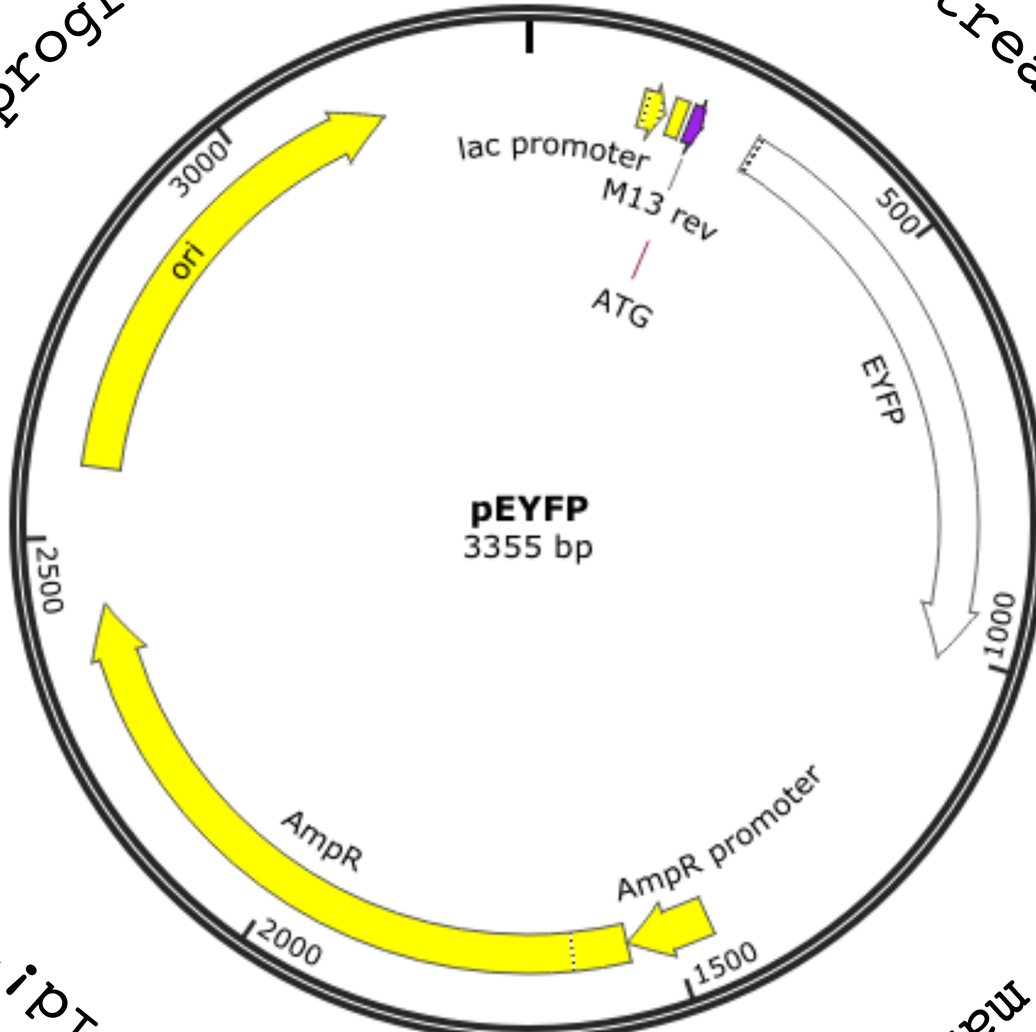


Icon sets: visual language – color – structure

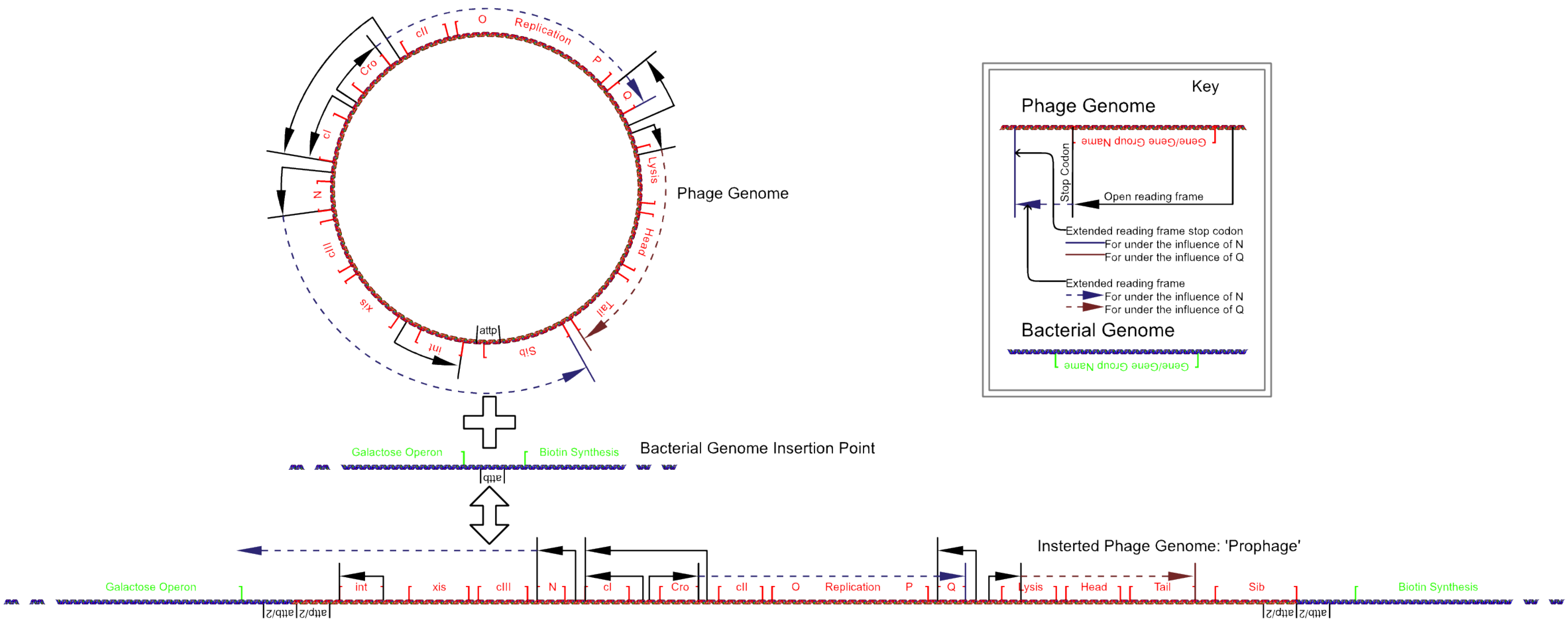


Challenges

```
#include <iostream> using namespace std;int main (){ cout << "Hello World!"; return 0;}// my first program in C++
```



Never odd or even.
No lemon, no melon.



Richard Wheeler (Zephyris) 2005; Schematic representation of the insertion of the bacteriophage lambda

Custom languages vs. Standardization

DSL in a standardized world

- GenBank
- Sequence Ontology
- Synthetic Biology Open Language

Need to map custom concepts to standards

- Data exchange
- Graphical representation
- Sequence annotation

Can we standardize customization?

SBLE 2013 – October 26 - Indianapolis

Workshop Synthetic Biology and Language Engineering

- Bring together language designers and synthetic biologists with the goal of analyzing the different programming paradigms that have been or could be explored to write these biological programs more effectively
- Part of:
  - International Conference on Software Language Engineering (SLE)
  - International Conference on Generative Programming: Concepts & Experiences (GPCE)
  - Systems, Programming, Languages and Applications: Software for Humanity (SPLASH)

http://planet-sl.org/sble-at-sle2013/

Acknowledgements

VBI SynBio Group

- M. Wilson
- D. Ball
- M. Lux
- L. Adam
- C. Overend

GenoCAD Alumni

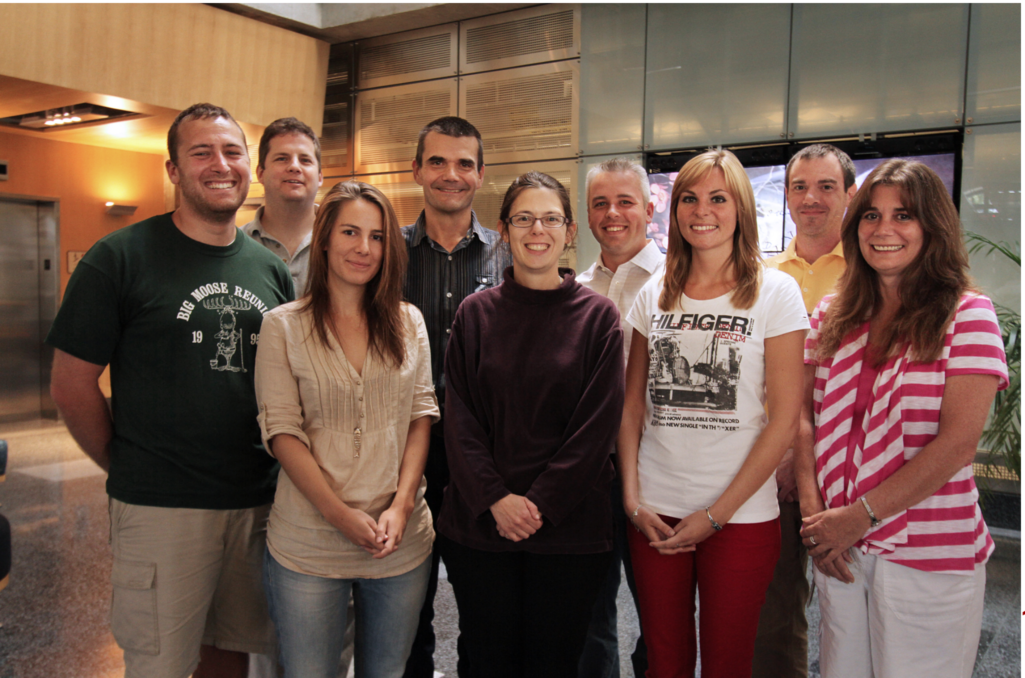
- Yizhi Patrick Cai (JHU)
- Mike Czar (Carillon)

GenoCAD Collaborators

- VBI: S. Hoops, J. Lewis
- SBOL: H. Sauro, C.Myers, D. Densmore, C. Rodriguez, M. Galdzicki and many more
- Language: Eric Van Wyck

Find GenoCAD

- www.genocad.org
- www.facebook.com/genocad/
- @genocad
- Google +
- LinkedIn



Questions?

CAD Model of VBI

Photo of VBI