Genetic Systems Engineering

Prashant Vaidyanathan, Evan Appleton, Curtis Madsen, Cristian-Ioan Vasile, Alan Pacheco, Iman Haghighi, Nicholas Roehner, Rachael Ivison, Junmin Wang, Yash Agarwal, Zachary Chapasko, Calin Belta, and Douglas Densmore
Boston University, Boston, Massachusetts, USA

Motivation

Systems engineering is an interdisciplinary field that focuses on building and maintaining complex engineering systems over their entire life cycles. Applying standards and core concepts of related sub-fields such as **performance engineering** (ensuring that a system meets its expected performance requirements) and **reliability engineering** (a system does not fail more than expected during its life cycle) is a promising way to advance synthetic biology's potential applications in research areas.

Approach

We have created a software platform (Phoenix) that encapsulates the procedures required during the **specify**-**design**-**build**-**test** work flow to enable iterative design of complex genetic systems. The framework is designed to ensure that the **functional**, **performance**, and **structural** specification of the genetic system is well defined, reproducible, and reliable. The iterative process will use principles of **reinforcement learning** to improve the performance of the synthesis and assignment algorithms.

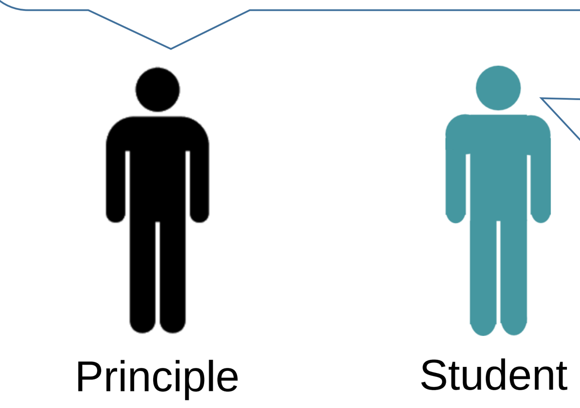
References

[1] E. Appleton et al. Interactive assembly algorithms for molecular cloning. Nat. Methods, 11:657-662, 2014.
[2] E. Oberortner et al. A rule-based design specification language for synthetic biology. ACM Journal on Emerging Technologies in Computing Systems (JETC), 11(3):25, 2014.
[3] C.-I. Vasile et al. Compositional signal temporal logic with applications to synthetic biology. In progress.
[4] P. Vaidyanathan et al. A framework for genetic logic synthesis. Proceedings of the IEEE, 103(11):2196-2207, 2015.

Phoenix Workflow

Can you build a genetic NOT gate, where if the level of the inducer is less than 5 units for 60 mins, the output signal must be greater than 10 units within 90 mins and remain greater than 10 units for 30 mins ?

I could use Phoenix to specify the structural, performance, and functional constraints and see if we can build such a construct with our library of DNA components!



Specify STL Formula

$$((G_{[0,60]}\ \text{inducer} <= 5) \Rightarrow (F_{[0,90]}G_{[0,30]}\ \text{output} > 10) \wedge (G_{[0,60]}\ \text{inducer} > 5) \Rightarrow (F_{[0,90]}G_{[0,30]}\ \text{output} <= 10))$$

Key
Purple Text : Linear Predicates
Gold Text : Temporal Operators

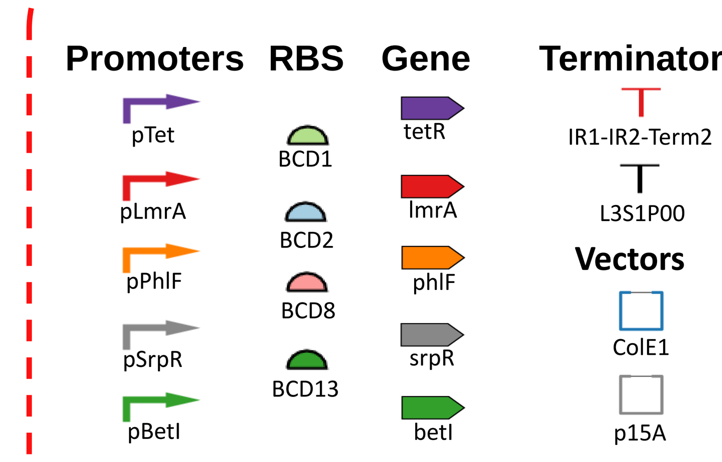
Structural Constraints

```
// COUNTING
ip1 exactly 1
rp1 exactly 1
g1 exactly 1
unk2 exactly 1
r1 exactly 1
r2 exactly 1
t1 exactly 1
t2 exactly 1

// INTERACTIONS
g1 represses rp1
ip1 drives g1
not ip1 drives unk2
rp1 drives unk2
not rp1 drives g1
```

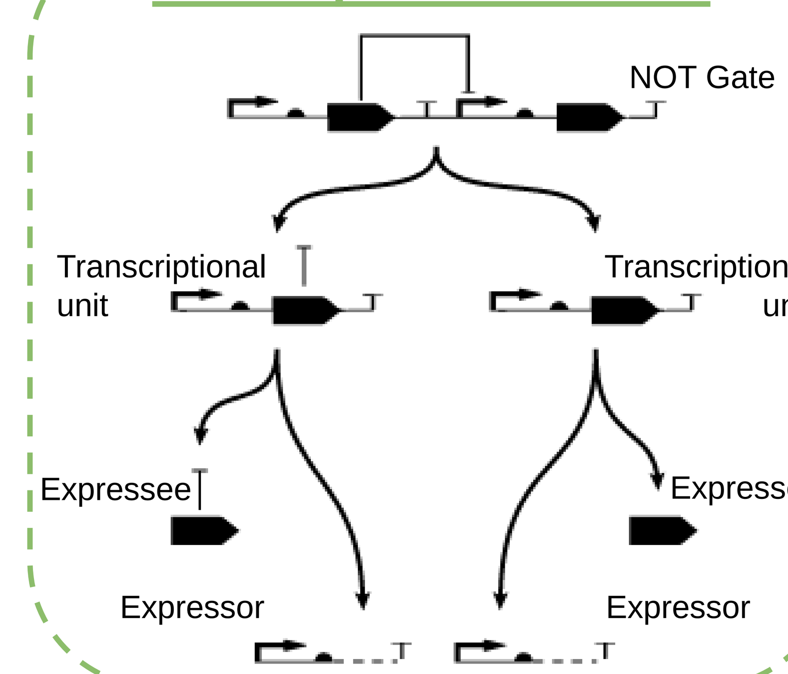
Rules specified in Eugene

Library of Components

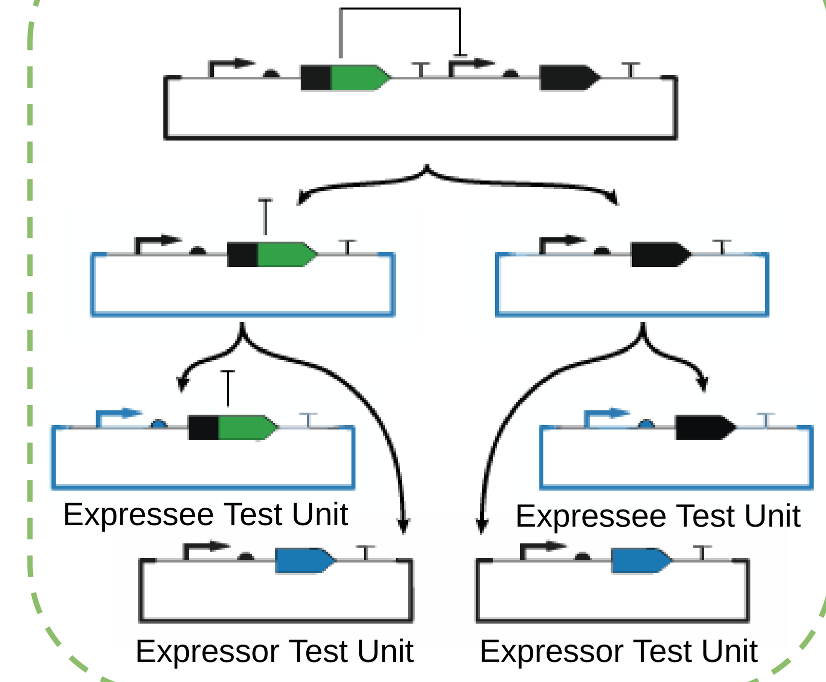


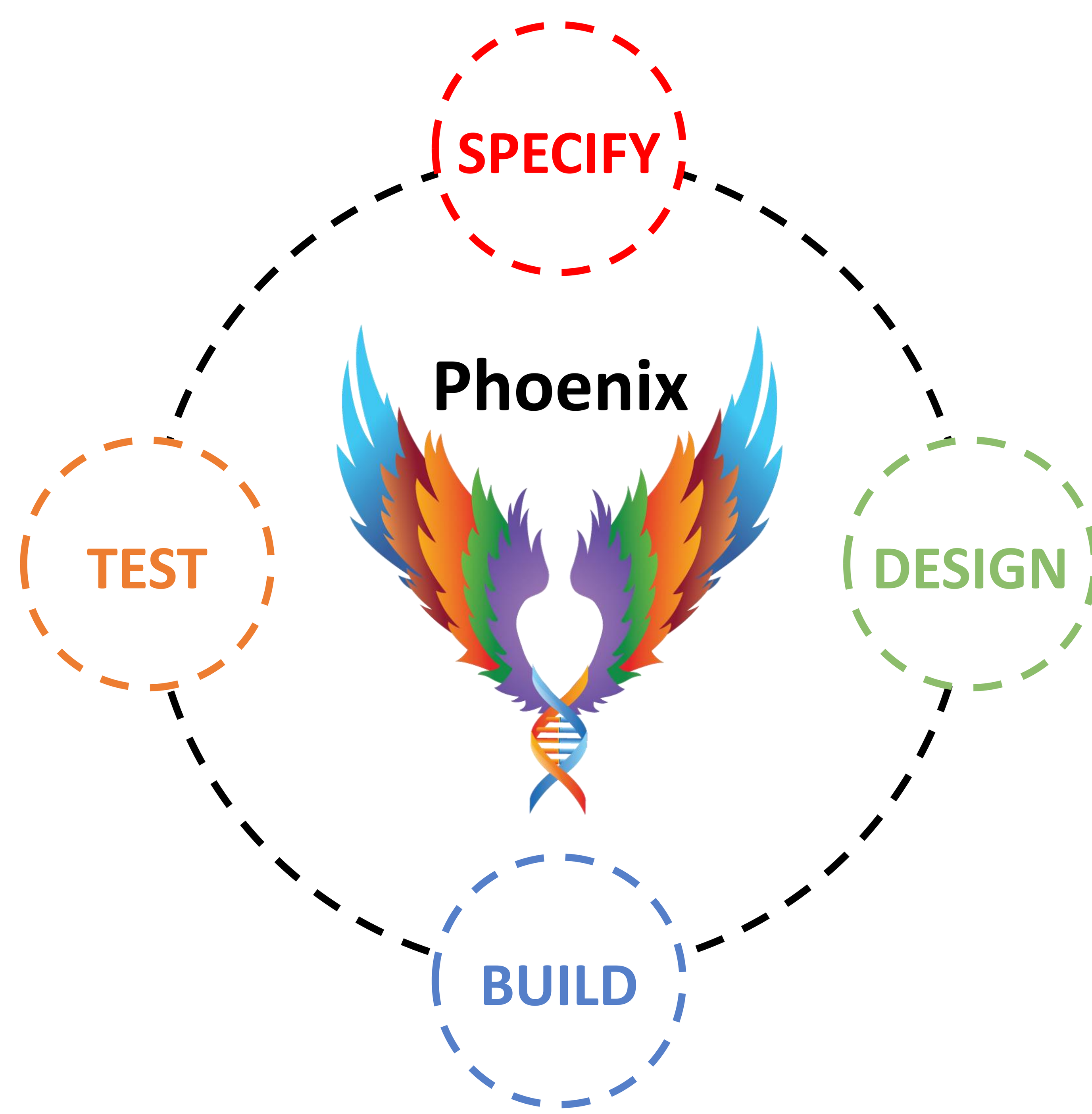
Components stored in Clotho

Decompose Structure



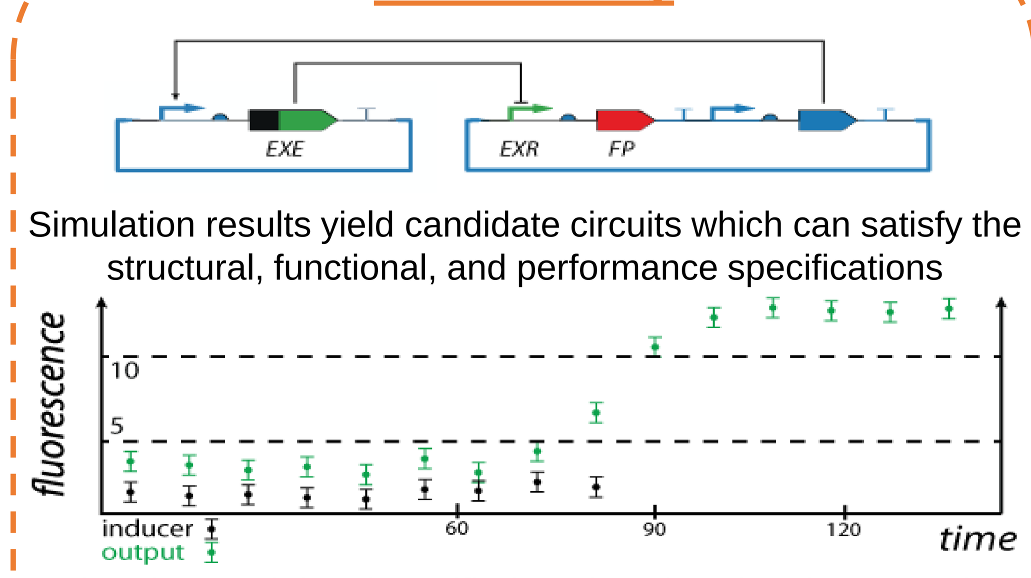
Create Testing Units





The Phoenix tool map and flow. This diagram describes the **Specify**, **Design**, **Build**, and **Test** steps to build a robust genetic inverter using principles of Systems Engineering.

Test & Verify

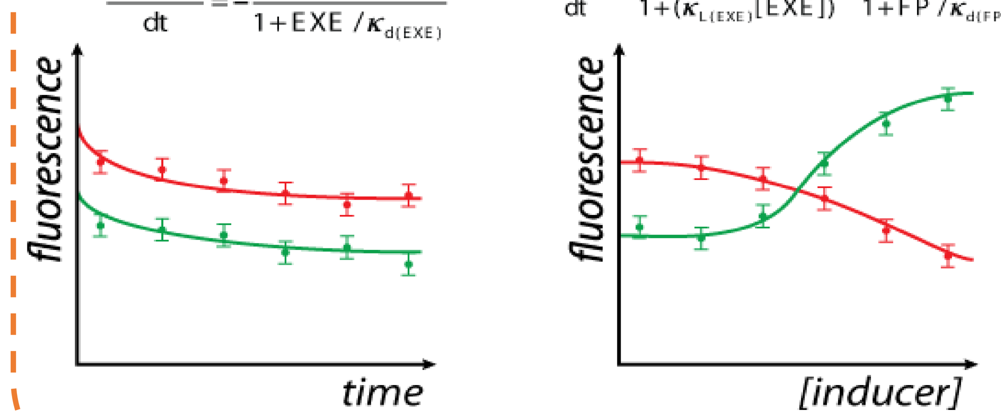


Simulation results yield candidate circuits which can satisfy the structural, functional, and performance specifications

The observed data is verified against the Signal Temporal Logic formula to confirm the robustness of the system

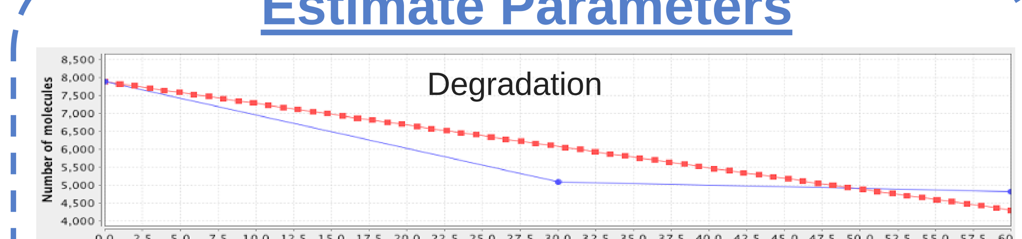
Simulate

$$\frac{d[EXE]}{dt} = -\frac{\gamma(EXE/\kappa_{d(EXE)})}{1+EXE/\kappa_{d(EXE)}} \qquad \frac{d[FP]}{dt} = \frac{k_{EXR}}{1+(\kappa_{L(EXE)}[EXE])} - \frac{\gamma(FP/\kappa_{d(FP)})}{1+FP/\kappa_{d(FP)}}$$



Simulation data is generated by iBioSim

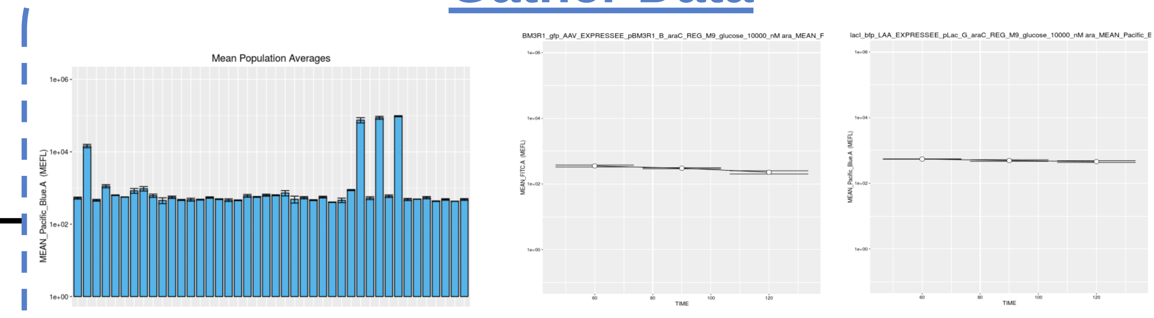
Estimate Parameters



K_d = 0.9999, y = 59.6013

Parameters are estimated by iBioSim for SBML models of the biochemical equations describing degradation, expression, and small molecule regulation of Expressors and Expressees

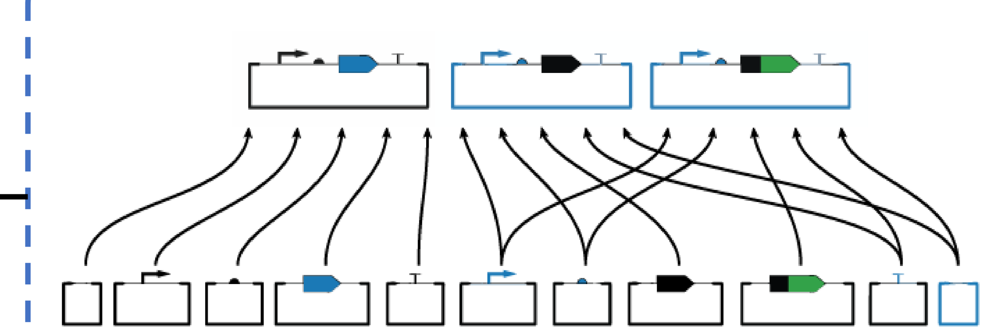
Gather Data



(Expression data) (Degradation curves for Expressees)

FCS data processed by an R script within the Phoenix framework

Generate Assembly Instructions



Assembly instructions generated by Raven

Research reported on this poster was supported by the National Science Foundation under grant CPS Frontier 1446607 and the Office of Naval Research MURI Award.