How to build an n-input circuit library

Swapnil Bhatia and Douglas Densmore
Joint work with Alec Nielsen, Michael Smanski, and Christopher Voigt

**Goal**
To build a library of genetic circuits implementing all three-input one-output Boolean functions

**Plan**
- Choose a **basis set** of logic gates
- For each possible Boolean function, generate **optimal** logic circuits
- For each logic circuit, generate abstract genetic circuit
- Assign parts to all abstract genetic circuits from a parts library
- Compute an **optimal** assembly plan
- Compile to **robot** code and execute

**Approach**
- Basis set: **two-input NOR gate**
- Optimal circuits via **exhaustive search**
- Abstract genetic circuits by **motif mapping**
- Part assignment for **maximal reuse**
- Modular cloning (MoClo) with **primer-minimizing** assembly plan
- MoClo protocols in **Puppeteer** on Tecan

**Optimal circuits**
A logic circuit is **optimal** if it uses the fewest gates
Define grammar: \( S \rightarrow a|b|c|0|(S \star S) \)

**Algorithm**
Generate all **distinct expressions** \( e \) from \( S \)
Compute **truth table** of the logic circuit of \( e \), \( C(e) \)
Retain, if size of \( C(e) \) < size of best expression so far

**Motif mapping**
\[
(x \star y) \quad \rightarrow \quad p(x) \quad p(y) \quad g(x \star y) \quad p(x \star y)
\]

**Part assignment as bipartite graph coloring**
Motif instances

**Algorithm**
Heuristic constraint solver:
Color all motif instances such that **all colors in a circuit are distinct**

**Primer optimal MoClo assembly**
Given a collection of sets of transcriptional units (TUs), compute a **linearization** for each set such that the number of **distinct vectors per TU** is minimized.

**Heuristic algorithm**
Linearize TUs by constituent part usage count

**Illustrative results**

**Development in progress**
Puppeteer implementation of MoClo assembly
Robot liquid class optimization
Circuit-level RBS tuning algorithm