(mini)Eugene:
Rule-based Design of Synthetic Biological Systems

Ernst Oberortner

Postdoctoral Research Associate

ernstl@bu.edu

Agilent Technologies

Combinatorial Design Problem

Given: - Part Library ***P*** consisting of ***k*** Parts

$$P = \{ p_1 , \ldots , p_k \}$$

Infinite number of designs!

- The length of the design ***N***

k^N designs

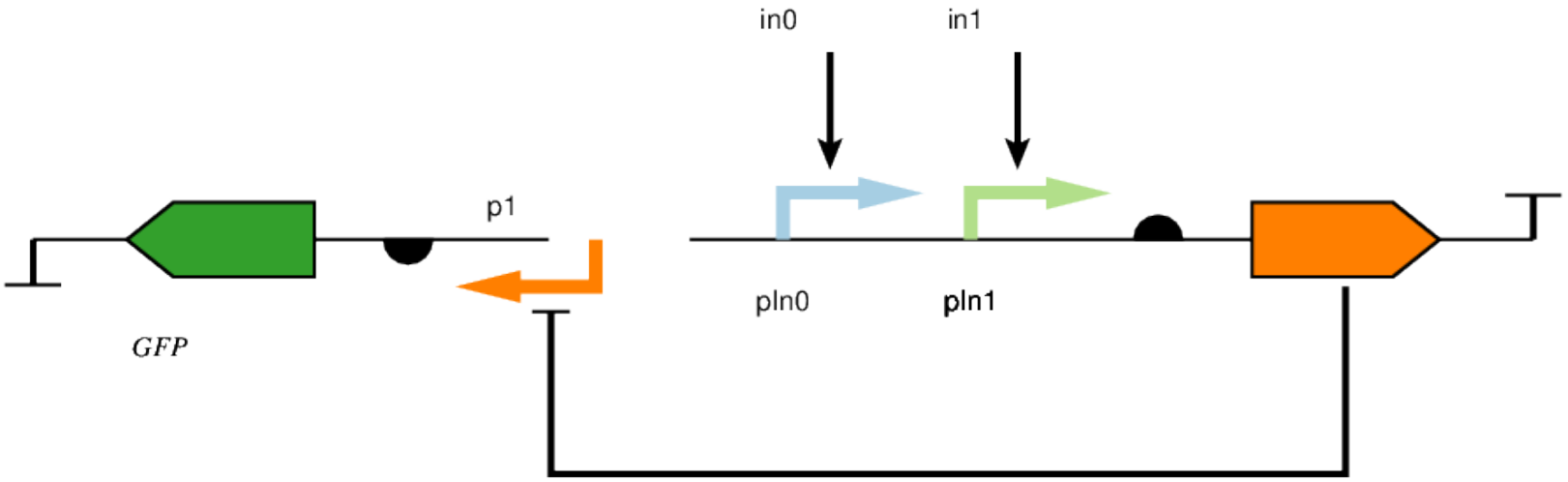
Orientation:

In a design, every part can have two orientations {Forward, Reverse}

$(k^N * 2^N)$ designs

Example: Transcriptional NOR Gate

Given: P = {pIn0, pIn1, p1, r, c1, t, GFP}, N=9



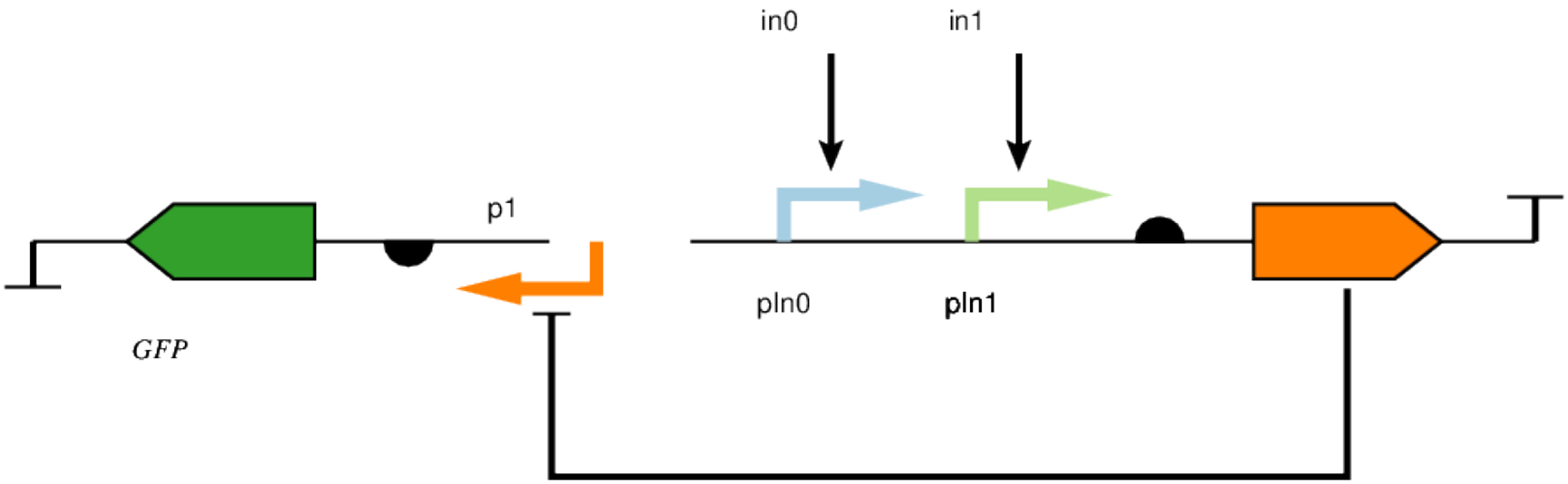
> 20 billion combinations ($7^9 * 2^9$)

However! not all possible combinations will function

Biology follows certain Rules

``Biology'' follows certain Rules

Counting … number of occurrences
Positioning … place of occurrence
Pairing … pair-wise occurrence
Interactions … regulatory interactions
Orientation … forward or reverse



Eugene

Finds **ALL** rule-compliant designs

- **Imperative** and **Declarative** Language
- Design Specification
  - Design **Templates**
  - **Rules** on the Design Templates
  - Automated **Generation** of Rule-Compliant Designs
- Populating the **Part Library**
  - Manually, Data Exchange Standards (SBOL)
- Design **Representation**:
  - Eugene, SBOL, Pigeon
- **Available** as
  - Web-based IDE, XML-RPC web service, Embeddable JAR, Runnable JAR

Eugene is **comprehensive** and
has step learning curve

miniEugene

Basic Language offering only **Rule Conjunctions**
Purely **declarative**
Design **Representation**: Pigeon, Textual, SBOL
Statistics

TASK:
Formulate the word SBOL by only using rules

Thanks!

Ernst Oberortner

Postdoctoral Research Associate

ernstl@bu.edu

CIDAR

BOSTON UNIVERSITY

Agilent Technologies