Synthetic Biology in a Nutshell

- Multi disciplinary approach
- design of synthetic bio-systems
- Engineering principles

Design ➔ Assembly of BIOBRICKS

Kwok - Nat. 463 Jan. 2010

FIVE HARD TRUTHS FOR SYNTHETIC BIOLOGY

Can engineering approaches tame the complexity of living systems? Roberta Kwok explores five challenges for the field and how they might be resolved.
Modeling 

Design
What I cannot compute, I cannot understand ...

Computing = Understanding

HOPE SO!
computation ↔ Biology

Bio-inspired

Life as model of computation

Computing with Life

Synthetic Biology

Programming Life

Biology

computation model as a model of Life

Systems biology
BIO-INSPIRED ALGORITHMS & MODELS

Life as model of computation
Bio-Inspired algorithms

• Computation Framework
  • For a large class of problems
  • Meta heuristic

• For complex problems
  • Intractable $\rightarrow$ NP complete
  • IA problems (Design)

• Common Features
  • Population, Society
  • Local operations
  • Global evaluation
  • Randomness

Genetic Algorithm
J. Holland - Goldberg

Swarm algorithm
M. Dorigo, V. Maniezzo, et A. Colorni

Neural networks
Hopfield – Rosenblatt
Genetic algorithm - Bridge Design

Darwinian Paradigm

Mutation  
Fitness ranking  
Cross over  
Selection


Agent = Genome coding for a bridge
Bio-Inspired models

Turing machine = “common ancestor”

- New computing model
  - Alternative computation medium
  - New computing rules
- Computational power
  - Turing universality: Simulation of a computer (universal Turing machine)
  - Massive parallelism: combinatorial resolution of NP complete problems

Membrane computing
G. Paun

L – Systems
A. Lindenmayer, P. Prusinkiewicz

Cellular Automata
J. Von Neuman - S. Wolfram

DNA computing
Adleman – G. Paun, G. Rozenberg
Example Membrane computing

- P-System
- Non determinism
- Evolution strategy $\rightarrow$ maximal parallel
- Application
  - Sorting
  - NP complete problem (TSP)
  - Verification of cryptography protocol

Skin
Membrane
Rules
Objects

\[ e \rightarrow d_{out} \]
\[ x \rightarrow y \]
\[ d \rightarrow \delta \]
\[ a \rightarrow ab \rightarrow c \]
SYSTEMS BIOLOGY

computation as a model of biological processes
computation-inspired framework

« De toute évidence, la cohérence fonctionnelle d'une machine chimique aussi complexe, et en outre autonome, exige l'intervention d'un système cybernétique gouvernant et contrôlant l'activité chimique en de nombreux points. » (p. 59)

• “Cybernétique moléculaire”
• Circuit / Network explanatory framework of biological process
• Logical Network (R.Thomas)
Formal models

- Discrete dynamics
  - State based
- Formal property analysis
  - Safety
  - Reachability / invariance
- Explanatory framework
  - Formalization of Biological process
  - Phenotype = molecular signature \( \rightarrow \) equilibria

Images of Automata, Petri net, Process algebra, Game theory, and Membrane comp.
Example - Automata based model

\[ a = \text{NOT } c \]
\[ b = a \]
\[ c = \text{NOT } b \]

Specification

Dynamics
SYNTHETIC BIOLOGY

Programming/designing living organism
Design-manufacturing $\rightarrow$ compilation

```c
main(){
    Printf(“Hello World”);
}
```

Program $\rightarrow$ Compilation $\rightarrow$ Execution

Computer

Synthetic Biology $\rightarrow$ Synthesis $\rightarrow$ SB function
CAD Environment (ideal) Overview

Interface

High level Program

Translator/Compiler

Low level Program

Validation/Optimization

Sequence

Synthesis

Genetic Engineering

Database

Parts

Dynamics Analysis

Safety Security Checking

Simulator

Trace Report

Integrated view from the current states of art
Compilation principles in a nutshell

Syntactic tree

Abstract machine

MEMORY

LD V,Rx

ST V,Ry

REGISTER

UAL

OP R1, R2, R3

LDC 3, R5
LDC 4, R2
LD &K, R1
MUL R2, R1, R4
LD R3, &J
ADD R5, R6, R7
ADD R5, R6, R7
DIV R4, R3, R6
LD R3, &J
MUL R2, R1, R4
LD &K, R1
LDC 4, R2
LDC 3, R5
LD &K, R1
LDC 4, R2

LDC 3, R5

LDC 4, R2

LD &K, R1

LDC 4, R2

LD &K, R1

LDC 4, R2

LDC 4, R2

LDC 4, R2
Languages for SB ➔ Structural description

- Program = sequence description
- Usual in language & CAD Env.
  - Genocad, GEC, Kera
  - Clotho, Eugène, Tinker Cell, ...
- Grammar rules = guide of design
- Structural description
- Low level (DNA sequence)

(GEC [Pedersen, Plotkin])

r0040:prom;b0034:rbs;c0040:prc;X:ter

(Genocad [Peccoud])

19
Franck Delaplace - Berder 2012
Structured vs. behavioral description

• Structure
  • component assembly description
  • Low level of description
  • Back end

• Behavior/function
  • Process design
  • Functional → safety
  • High level of description
  • Specification/document

Hardware Description Language
Verilog - VHDL

Behavioral program

Generate Oil
Y when X

Structural program

X

Y

ATG|AAA|TTG|...

Synthesis

Toy example
Compilation ➔

Part assembly = behavior assembly

• Component description = behavior description
• Bio system design = behavior description
  • The program describes the expected “function”
• The issue is to define a compilation method assembling parts such that the behavior of the assembly is “similar” to the behavior of the designed function.
• Problem - Reliability ➔ guaranty on the assembly
  • i.e. formal guaranty that each step is correct w.r.t. to the behavioral “similarity”.

Correct translation (compilation) ?
Behavior compilation for SB - 2 possible ways

Computational description

- Behavior = Abstract machine instruction
- Synthesis = Semantic rules

Logical Specification

- Behavior = Specification
- Synthesis = Proof rules

State = Situation

Event-driven transition

State = Situation

Axioms

Behavior of components

Proof

Theorem

Behavior of the function

Dessine moi un canard !
**Model & theory in logic**

Is a formula true?

**Model**

- Interpretation
  
  \[ I(b) = 1, I(c) = 1 \]
  
  \[ I(b \lor c) = \max(I(b), I(c)) \]
  
  \[ I(b \land c) = \min(I(b), I(c)) \]

<table>
<thead>
<tr>
<th>OR</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

**Theory**

- Deduction system
  
  \[ c \text{ is proved} \]
  
  \[ b \lor c \text{ is proved} \]

  \[ b \text{ is proved} \]
  
  \[ b \lor c \text{ is proved} \]

  \[ b \land c \text{ is proved} \]
Correction of the assembly

Correction of assembly = observational behavior inclusion

Compilation = Proof ➞ find a behavior assembly s.t. the behavior of the designed function is included

Part A

Part B

AATTGGAAGCC + AATGCGTTTATAGCCCCATGG..
Remarks

• Functional/behavioral programming
  • 1 function ➔ n structures – e.g. inhibition -
  • Document function ➔ Safety analysis capability

• Proof framework
  • Safe design ➔ formal method + safety analysis
  • Functional/qualitative description ➔ Specification, resolution principles
  • Quantitative description ➔ Strategy of the resolution, tuning

• Hierarchy of the components ≠ Organization for organisms
  • Functionality
  • Inter-operability

• GUBS project - Adrien Basso Blandin – Franck Delaplace
  • Behavioral language
  • Compiler
Synbiotic: Tower of languages

- Global specification
  - Population level
  - Global programming

- Local specification
  - Agent centric
  - Elementary behavior
    - GUBS

- Implementation
  - Interface in-silico / In vivo
  - Regulatory network

Conclusion

- Function
- Process

Behavioral Language for synthetic biology
THANK YOU !