Human Comprehensible Active Learning of Genome-Scale Metabolic Networks

Lun Ai\(^1\), Shishun Liang\(^2\), Wang-Zhou Dai\(^3\), Liam Hallett\(^2\), Stephen H. Muggleton\(^1\), Geoff S. Baldwin\(^2\)

\(^1\)Department of Computing, Imperial College London, UK
\(^2\)Department of Life Science, Imperial College London, UK
\(^3\)School of Intelligence Science and Technology, Nanjing University, China

lun.ai15@imperial.ac.uk
Outline

- Scientific problem
- Framework
- Novel matrix approach
- Results
- Summary
Importance to Synthetic Biology

- **Aim:**
  - Synthesise **useful compounds**

- **Method:**
  - Tuning for correct precursors
  - Engineering of **metabolic networks**
  - Examination of **phenotypes**
Exemplary metabolic network

\[ m_1 + m_2 \leftrightarrow m_3 + m_4 \]
Efficiently learn and navigate genome-scale metabolic networks?

iML1515 (Monk et al. 2017), **100 times increase in model complexity**
Reduce cost and design space?

1515 genes + 2719 reactions
● Scientific problem

● **Framework**
  ○ Model-Comprehend

● Novel matrix approach

● Results

● Summary
Model-Comprehend

- Automate Design, Build, Test, Learn
  - Rapid inferences
  - Hypothesis space reduction
  - Cost minimisation
Online source

Phenotype datasets

iML1515

Compile

Simulate

Classifications

Supply

Hypothesise

Update

Active learning

Select

Logical model

Abductive reasoning

codes(gene_b, e_b).
reaction(m1, m3).
enzyme(e_b, m1, m4).
metabolite(m1).
Model

Design → Build → Test → Learn

Human comprehension tests

Comprehend
• Scientific problem
• Framework
• **Novel matrix approach**
  1. **Abduction**
  2. **Active learning**
• Results
• Summary
Abduction

\[ p \leftarrow q \land r \]

- Phenotypes in various nutrient media
- Simulation based on metabolite saturation
- Hypothesise gene functions to explain data

Experiment data:

\text{phenotypic\_effect}(\text{Gene, Nutrients}).

Hypothesis:

\text{codes}(\text{Gene, Enzyme}).
phenotypic_effect(Gene, Medium):-
  % abduced fact
codes(Gene, Enzyme),
cant_use_enzyme(Enzyme),
%metabolic pathways
metabolic_path(Medium, Metabolites),
no_essential_molecule(Metabolites).

Metabolite saturation
Novel matrix encoding

Knowledge base

Reactant/product matrices

\[
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

Metabolites

Binary matrix operations

Metabolites

Metabolites
Active learning

a) Binary discrimination of hypotheses

b) Approx. minimum cost binary decision tree
● Scientific problem

● Framework

● Novel matrix approach

● **Results**
  ○ *Runtime improvement*
  ○ *Cost reduction*

● Summary
### Time per simulation

<table>
<thead>
<tr>
<th></th>
<th>Robot Scientist (King et al. 2004)</th>
<th>MC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Without multi-threads</td>
<td>≈250s</td>
<td>≈0.6s</td>
</tr>
<tr>
<td>With multi-threads 20 cpus</td>
<td>≈27s</td>
<td>≈0.06s</td>
</tr>
</tbody>
</table>

> 4000 times better in runtime
10 times saving in cost
- Scientific problem
- Framework
- Novel matrix approach
- Results
- Summary
Summary

- To automate metabolic network engineering
- Overcome network complexity
  - 4000+ times better runtime
- Experimental design
  - 10 times lower cost
  - Most informative trials
Future work

- Generalisation of framework
  - Quantification
  - Multi-clause theories

- Optimisation of metabolic network
  - Validate hypotheses (CRISPRi)
  - Multiple gene loci

- Hypothesis comprehensibility (Ai et al. 2021)
Lun Ai

Email: lun.ai15@imperial.ac.uk

Website: https://lai1997.github.io/

Linkedin: https://www.linkedin.com/in/lun-ai-46481a128/
Logical knowledge base

% static knowledge
codes(gene_b, e_b).
codes(gene_c, e_c).
codes(gene_e, e_e).
...
metabolic_step(n1, m3).
metabolic_step(n1, m4).
...
enzyme(e_a, n1, m3).
enzyme(e_b, n1, m4).
essential molecule(m7).
essential molecule(m8).

% description of effect using metabolic network
phenotypic_effect(Gene, Medium):-
   % abduced fact
codes(Gene, Enzyme),
cant_use_enzyme(Enzyme),
%metabolic pathways
metabolic_path(Medium, Metabolites),
no_essential_molecule(Metabolites).
<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Four metabolites:</td>
<td>m1, m2, m3, m4</td>
</tr>
<tr>
<td>Two orfs:</td>
<td>g1, g2</td>
</tr>
<tr>
<td>Initial metabolic state:</td>
<td>m1, m2, m3</td>
</tr>
<tr>
<td>Representation:</td>
<td>[1, 1, 1, 0]</td>
</tr>
<tr>
<td>Logic encoding:</td>
<td>mstate(0, 14)</td>
</tr>
</tbody>
</table>
Reactions:

enz1: m1 + m2 -> m3 + m4
enz2: m3 <-> m4

Genes:

codes(g1, enz1)
codes(g2, enz2)
State S: [1,1,1,0]

Reactions: [1,1,0]

(1) Im subset

[1,1,0,0]
[0,0,1,0]
[0,0,0,1]

(2) KO enzyme Enz1

[1,0,0]

Reactions: [0,1,0]

(3) Im product

[0,0,1,1]
[0,0,0,1]
[0,0,1,0]

KO g1

New metabolite(s): [0,0,0,1]

New state S': [1,1,1,1]
KO g1 + g2

State S:
\[ [1,1,1,0] \]

Reactions:
\[ [1,1,0] \]

(1) Im subset
\[ 1,1,0,0 \]
\[ 0,0,1,0 \]
\[ 0,0,0,1 \]

Rm

(2) KO enzyme Enz1
\[ [1,1,0] \]

Reactions:
\[ [0,0,0] \]

New metabolite(s):
\[ [0,0,0,0] \]

State unchanged:
\[ [1,1,1,0] \]

(3) Im product
\[ 0,0,1,1 \]
\[ 0,0,0,1 \]
\[ 0,0,1,0 \]