ORIENTING BIOCHEMICAL REACTIONS IN METABOLIC PATHWAYS AND NETWORKS

Presented by Zephyr Serret Verbist

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat d'Informàtica de Barcelona

FIB

22nd of January 2024

Table of Contents :

1.a. Terms and definitions - Hypergraphs

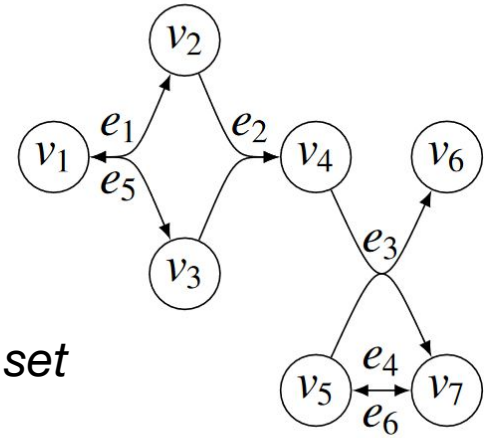
Hypergraphs are an extension to graphs

They are defined as $H=(V,E)$

where

- *V is the set of vertices*
- *E is the set of hyperedges*

*Each hyperedge has a **tail** set and a **head** set*



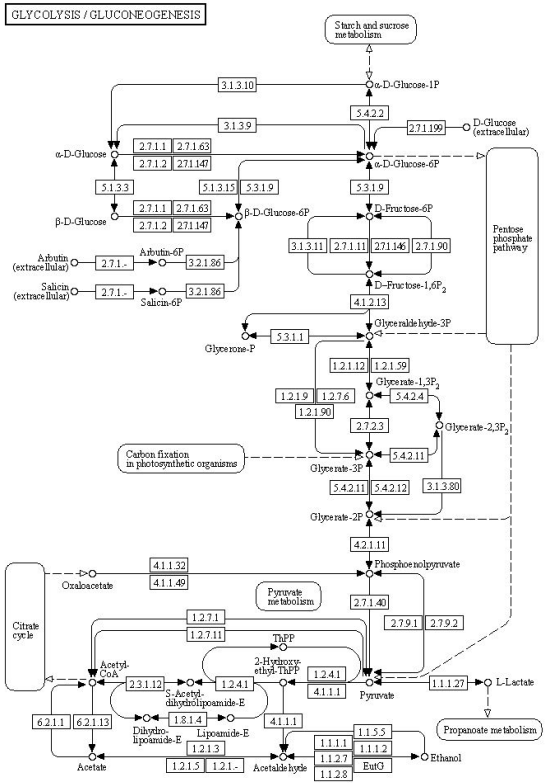
1. $v_1 \rightarrow v_2 + v_3$
2. $v_1 + v_2 \rightarrow v_4$
3. $v_4 + v_5 \rightarrow v_6 + v_7$
4. $v_5 \rightarrow v_7$
5. $v_2 + v_3 \rightarrow v_1$
6. $v_7 \rightarrow v_5$

$$E=\{\,(T(e),H(e))\mid T(e)\subseteq V,\ H(e)\subseteq V,\ T(e)\neq\emptyset,\ H(e)\neq\emptyset\}.$$

1.a. Terms and definitions - Metabolic Pathway

Metabolic pathways are sets of linked chemical reactions that occur in the cell.

In this study, we model them using hypergraphs

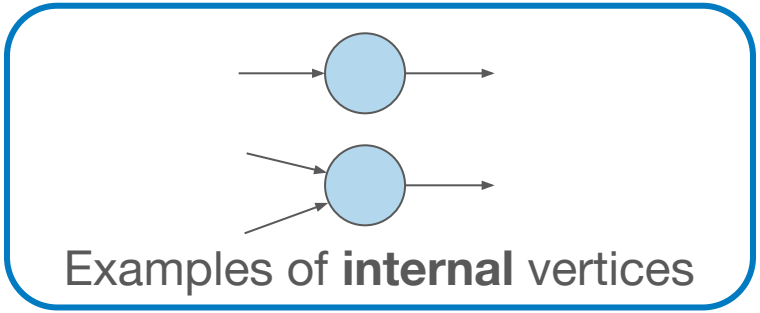


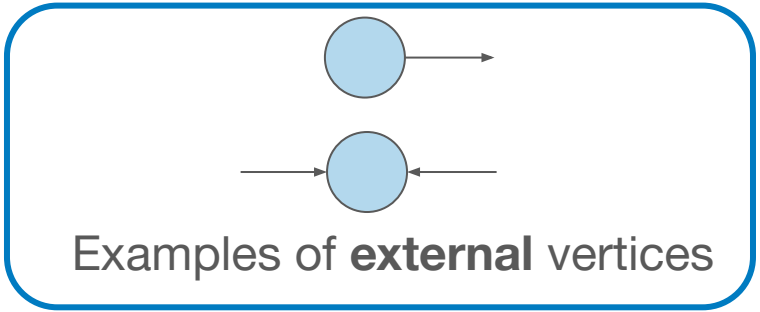
1.a. Terms and definitions - Internal/external vertices

A vertex is **internal** if and only if

- It has one or more incoming edges
- It has one or more outgoing edges

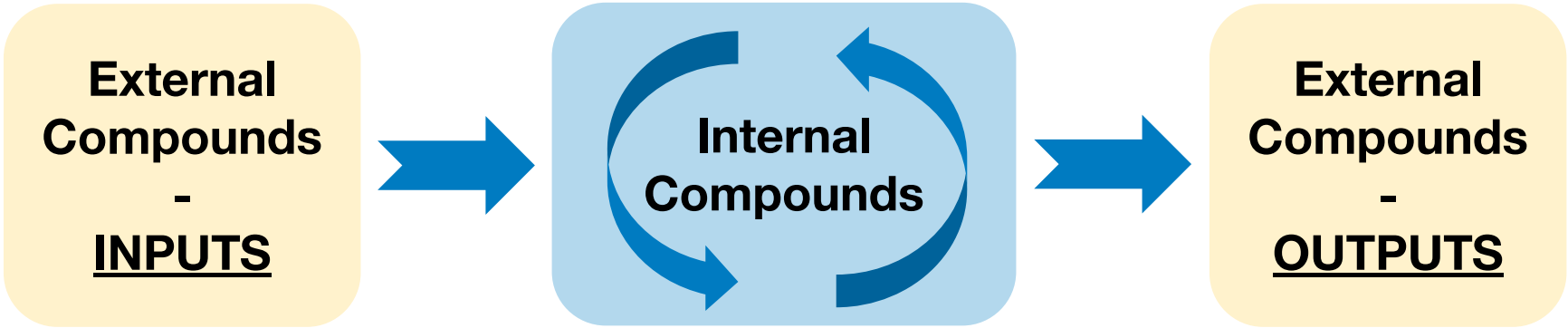
Otherwise, it is **external**.



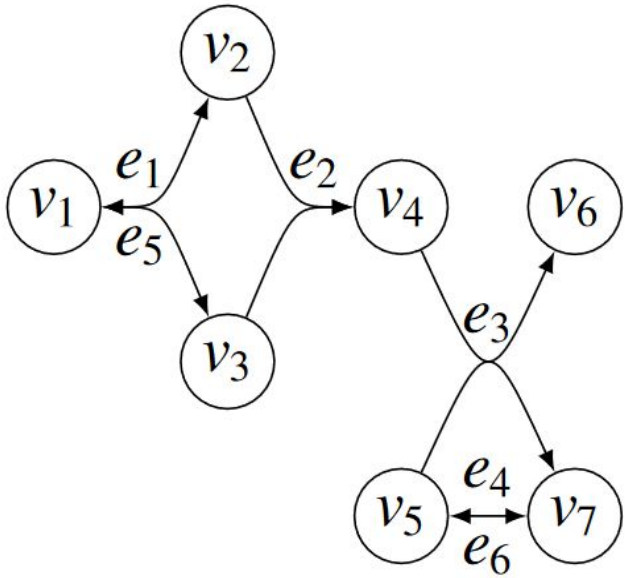


1.b. Problem definition

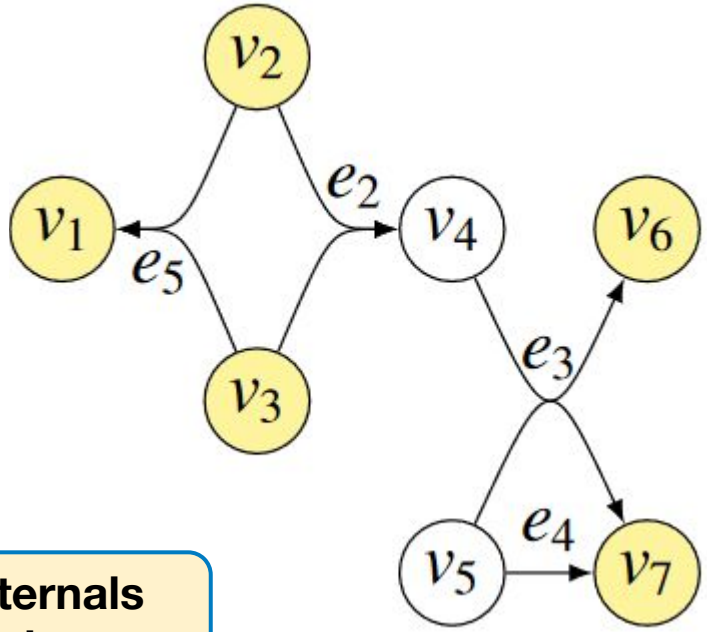
Given a **metabolic** pathway, orient the reactions in order to minimize the number of **external** compounds.



1.b. Problem definition

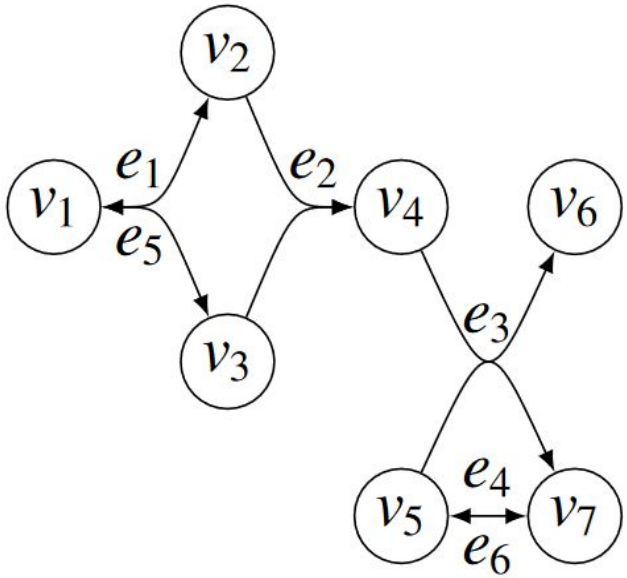


1. $v_1 \rightarrow v_2 + v_3$
2. $v_1 + v_2 \rightarrow v_4$
3. $v_4 + v_5 \rightarrow v_6 + v_7$
4. $v_5 \rightarrow v_7$
5. $v_2 + v_3 \rightarrow v_1$
6. $v_7 \rightarrow v_5$

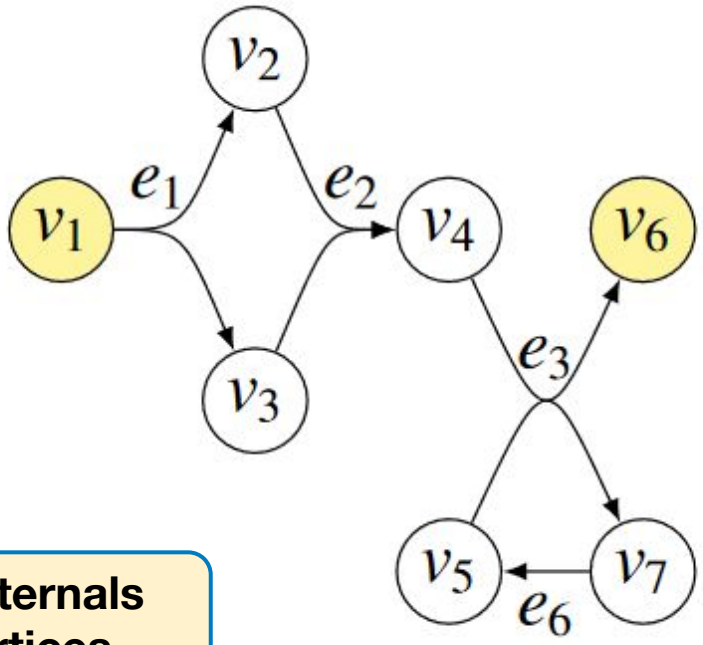


5 externals vertices

1.b. Problem definition



1. $v_1 \rightarrow v_2 + v_3$
2. $v_1 + v_2 \rightarrow v_4$
3. $v_4 + v_5 \rightarrow v_6 + v_7$
4. $v_5 \rightarrow v_7$
5. $v_2 + v_3 \rightarrow v_1$
6. $v_7 \rightarrow v_5$



2 externals vertices

1.c. Objectives and Scope

- Develop a **ILP** (Integer Linear Programming) model that, given a **hypergraph**, **orients** all **hyperedges** in order to **minimize** the number of **external** vertices.

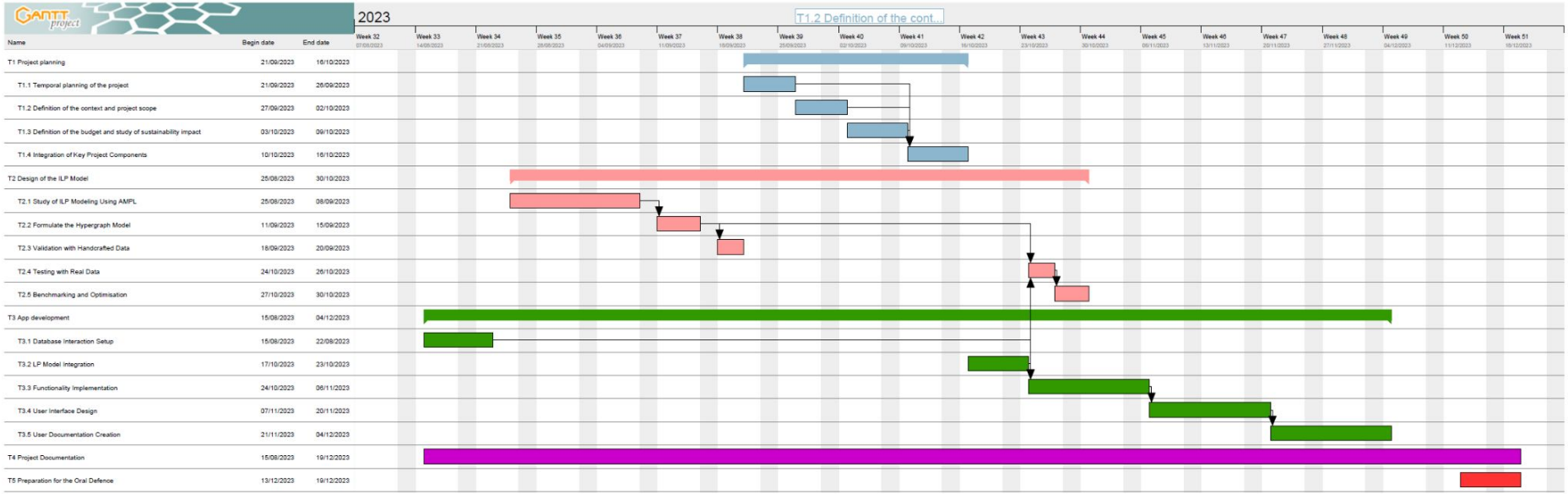
- Create an application that connects to **KEGG** and allows anyone to try out the models.

2. Planning

- Started around July
- ~ 25 hrs per week

ID	Task	Dependencies	Duration (hours)
T1	Project Management		85
T1.1	Definition of the context and project scope		25
T1.2	Temporal planning of the project		15
T1.3	Budget and sustainability		15
T1.4	Integration of Key Project Components	T1.1, T1.2, T1.3	15
T1.5	Meetings with the tutor		15
T2	Design of the ILP Model		90
T2.1	Study of ILP Modeling Using AMPL		40
T2.2	Formulate the Hypergraph Model	T2.1	15
T2.3	Validation with Handcrafted Data	T2.2	10
T2.4	Testing with Real Data	T2.2, T3.1	15
T2.5	Benchmarking and Optimization	T2.4	10
T3	App development		210
T3.1	Database Interaction Setup		35
T3.2	ILP Model Integration	T2.2	25
T3.3	Functionality Implementation	T3.1, T3.2	50
T3.4	User Interface Design	T3.3	50
T3.5	User Documentation Creation	T3.4	50
T4	Project documentation		50
T5	Preparation for the oral defence	T1, T2, T3, T4	30
		Total:	435

2. Planning


GANTT project
2023
T1.2 Definition of the cont...
Week 32 07/08/2023
Week 33 14/08/2023
Week 34 21/08/2023
Week 35 28/08/2023
Week 36 04/09/2023
Week 37 11/09/2023
Week 38 18/09/2023
Week 39 25/09/2023
Week 40 02/10/2023
Week 41 09/10/2023
Week 42 16/10/2023
Week 43 23/10/2023
Week 44 30/10/2023
Week 45 06/11/2023
Week 46 13/11/2023
Week 47 20/11/2023
Week 48 27/11/2023
Week 49 04/12/2023
Week 50 11/12/2023
Week 51 18/12/2023
Name
Begin date
End date
T1 Project planning 21/09/2023 18/10/2023
T1.1 Temporal planning of the project 21/09/2023 28/09/2023
T1.2 Definition of the context and project scope 27/09/2023 02/10/2023
T1.3 Definition of the budget and study of sustainability impact 03/10/2023 09/10/2023
T1.4 Integration of Key Project Components 10/10/2023 18/10/2023
T2 Design of the ILP Model 25/08/2023 30/10/2023
T2.1 Study of ILP Modeling Using AMPL 25/08/2023 08/09/2023
T2.2 Formulate the Hypergraph Model 11/09/2023 15/09/2023
T2.3 Validation with Handcrafted Data 18/09/2023 20/09/2023
T2.4 Testing with Real Data 24/10/2023 26/10/2023
T2.5 Benchmarking and Optimisation 27/10/2023 30/10/2023
T3 App development 15/08/2023 04/12/2023
T3.1 Database Interaction Setup 15/08/2023 22/08/2023
T3.2 LP Model Integration 17/10/2023 23/10/2023
T3.3 Functionality Implementation 24/10/2023 06/11/2023
T3.4 User Interface Design 07/11/2023 20/11/2023
T3.5 User Documentation Creation 21/11/2023 04/12/2023
T4 Project Documentation 15/08/2023 19/12/2023
T5 Preparation for the Oral Defence 13/12/2023 19/12/2023

3. Budget

Category	Cost
Personnel Cost per Activity	€19,425.69
Generic Costs	€146.12
Incidental Costs	€843.75

Total	€20,415.56
Total with Contingency	€22,457.12

4. Sustainability

Difficult to assess because:

- It's a novel problem, there are no alternatives or prior studies.
- It isn't clear whether the results from our algorithm bear any **biological significance**.

5.a. Basics of AMPL

AMPL = “A Mathematical Programming Language”

5.b. Our models

First Prototype

```
1  ######################
2  ## Inputs
3  ######################
4
5  set V; # nodes
6  set E; # hyperedges
7
8  set X{E}; # for each hyperedge, its tail set
9  set Y{E}; # for each hyperedge, its head set
10
11 param invertible{E} binary; # determines whether an edge is invertib
12
13 ######################
14 ## Helpers
15 ######################
16
17 set substrate_in{i in V} := {j in E: i in X[j]};
18 set product_in{i in V} := {j in E: i in Y[j]};
19
20 ######################
21 ## Variables
22 ######################
23
24 var inverted {E} binary; #determines whether an edge is inverted
25
26 var has_out{V} binary;
27 var has_in{V} binary;
28
29 var is_internal{V} binary;
```

```
31 ######################
32 ## Rules
33 ######################
34
35 maximize obj: sum{i in V} is_internal[i];
36
37 subject to compute_is_internal{i in V}:
38         is_internal[i] <= has_in[i] + has_out[i] - 1;
39
40 subject to substrates_not_inverted{i in V, j in substrate_in[i]}:
41         has_in[i] >= 1-inverted[j];
42
43 subject to substrates_inverted{i in V, j in substrate_in[i]}:
44         has_out[i] >= inverted[j];
45
46 subject to products_not_inverted{i in V, j in product_in[i]}:
47         has_out[i] >= 1-inverted[j];
48
49 subject to products_inverted{i in V, j in product_in[i]}:
50         has_in[i] >= inverted[j];
51
52 subject to not_substrate_at_all{i in V}:
53         has_in[i] <=
54             sum{j in substrate_in[i]} (1-inverted[j]) +
55             sum{j in product_in[i]} inverted[j];
56
57 subject to not_product_at_all{i in V}:
58         has_out[i] <=
59             sum{j in product_in[i]} (1-inverted[j]) +
60             sum{j in substrate_in[i]} inverted[j];
61
62 subject to respect_invertability {i in E}:
63         inverted[i] <= invertible[i];
```

5.b. Our models

MODEL A

```
1  ######################
2  ## Parameters
3  ######################
4  set V;
5  set E;
6
7  set X{E} within V;
8  set Y{E} within V;
9
10 ######################
11 ## Variables
12 ######################
13
14 var inverted{E} binary;
15 var is_internal{V} binary;
16
17
```

```
18 ######################
19 ## Rules
20 ######################
21
22 maximize obj: sum {j in V} is_internal[j];
23
24 subject to outgoing_half_implies_internal {j in V}:
25     sum {i in E: j in X[i]} (1-inverted[i]) +
26     sum {i in E: j in Y[i]} inverted[i]
27     >= is_internal[j];
28     # will only allow is_internal to be one if there's an outgoing edge.
29
30 subject to incoming_half_implies_internal {j in V}:
31     sum {i in E: j in Y[i]} (1-inverted[i]) +
32     sum {i in E: j in X[i]} inverted[i]
33     >= is_internal[j];
34     # will only allow is_internal to be one if there's an incoming edge.
```

5.b. Our models

MODEL B

```
set V;
set E;

param M = card(V);

set X {E} within V;
set Y {E} within V;


var inverted {E} binary;
var source {V} binary;
var sink {V} binary;
var is_internal{V} binary;

maximize internal: sum {j in V} is_internal[j];

#############

s.t. outgoing_1{j in V}: #for every vertex
    sum{i in E: j in Y[i]} inverted[i]  +
    sum{i in E: j in X[i]} (1-inverted[i])
    - M * source[j]
    <= 0;

s.t. outgoing_2{j in V}:
    sum{i in E: j in Y[i]} inverted[i]  +
    sum{i in E: j in X[i]} (1-inverted[i])
    - 0.5 * source[j]
    >= 0;
```

```
##############

s.t. incoming_1{j in V}:
    sum{i in E: j in Y[i]} (1-inverted[i]) +
    sum{i in E: j in X[i]} inverted[i]
    - M * sink[j]
    <= 0;

s.t. incoming_2{j in V}:
    sum{i in E: j in Y[i]} (1-inverted[i]) +
    sum{i in E: j in X[i]} inverted[i]
    - 0.5 * sink[j]
    >= 0;

###############

s.t. internal_1 {j in V}:
    source[j] + sink[j] -1 - M * is_internal[j] <= 0;

s.t. internal_2 {j in V}:
    source[j] + sink[j] -1 - 0.5 * is_internal[j] >= 0;
```

5.b. Our models

Serret's DualImply model

```
# ####################
## Parameters
# ####################
set V;
set E;

param M = card (E);

set X {i in E} within V;
set Y {i in E} within V;

# ####################
## Variables
# ####################
var inverted {E} binary ;
var has_outgoing {V} binary ;
var has_incoming {V} binary ;
var is_internal {V} binary;
```

```
# ####################
## Rules
# ####################
maximize internal : sum {j in V} is_internal[j];

subject to is_internal_1 {i in V}:
        is_internal[i] * 2  <= has_incoming[i] + has_outgoing[i];

subject to outgoing_implies_has_outgoing {j in V}:
        sum {i in E: j in X[i]} (1- inverted [i]) +
        sum {i in E: j in Y[i]} inverted [i]
        >= has_outgoing [j];

subject to has_outgoing_implies_outgoing {j in V}:
        sum {i in E: j in X[i]} (1- inverted [i]) +
        sum {i in E: j in Y[i]} inverted [i]
        <= M * has_outgoing [j];

subject to incoming_implies_has_incoming {j in V}:
        sum {i in E: j in X[i]} inverted [i] +
        sum {i in E: j in Y[i]} (1- inverted [i])
        >= has_incoming [j];

subject to has_incoming_implies_incoming {j in V}:
        sum {i in E: j in X[i]} inverted [i] +
        sum {i in E: j in Y[i]} (1- inverted [i])
        <= M * has_incoming [j];
```

5.b. Our models

Serret's DualImply model
Extra restrictions

```
12 set uninvertibles within E; # set of edges that cannot be inverted
13 set forced_internals within V; # set of nodes that must be sinks
14 set forced_externals within V; # set of nodes that must be sources
```

```
52 # ####################
53 ## Extra Restrictions
54 # ####################
55
56 subject to forced_externals_rule {i in forced_externals }:
57         has_incoming [i] + has_outgoing [i] <= 1;
58
59 subject to forced_internals_rule {i in forced_internals }:
60         has_incoming [i] + has_outgoing [i] = 2;
61
62 subject to respect_invertability {i in uninvertibles }:
63         inverted [i] = 0;
```

5.c. Mathematical grounding

```
16 # ####################
17 ## Variables
18 # ####################
19 var inverted {E} binary
20 var has_outgoing {V} binary
21 var has_incoming {V} binary
22 var is_internal {V} binary;
23
```

inverted_j: True if the solution inverts the hyperedge j.

$\textbf{inTailSet}(i)$: returns the set of hyperedges that have the vertex i in their tail set.

$\textbf{inHeadSet}(i)$: returns the set of hyperedges that have the vertex i in their head set.

$$\text{is\_internal}_i \equiv \text{has\_incoming}_i \wedge \text{has\_outgoing}_i$$

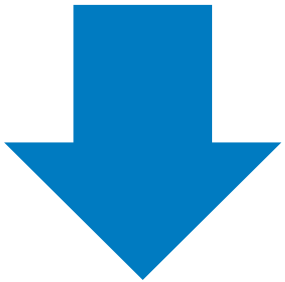
$$\text{has\_outgoing}_i \iff (\exists\, j \in \textbf{inTailSet}(i) : \neg \text{inverted}_j) \vee (\exists\, j \in \textbf{inHeadSet}(i) : \text{inverted}_j)$$

$$\text{has\_incoming}_i \iff (\exists\, j \in \textbf{inTailSet}(i) : \text{inverted}_j) \vee (\exists\, j \in \textbf{inHeadSet}(i) : \neg \text{inverted}_j)$$

5.b. Mathematical grounding

MODEL A

$$\text{has\_outgoing}_i \implies (\exists\, j \in \textbf{inTailSet}(i) : \neg\text{inverted}_j) \vee (\exists\, j \in \textbf{inHeadSet}(i) : \text{inverted}_j)$$
$$\text{has\_incoming}_i \implies (\exists\, j \in \textbf{inTailSet}(i) : \text{inverted}_j) \vee (\exists\, j \in \textbf{inHeadSet}(i) : \neg\text{inverted}_j)$$



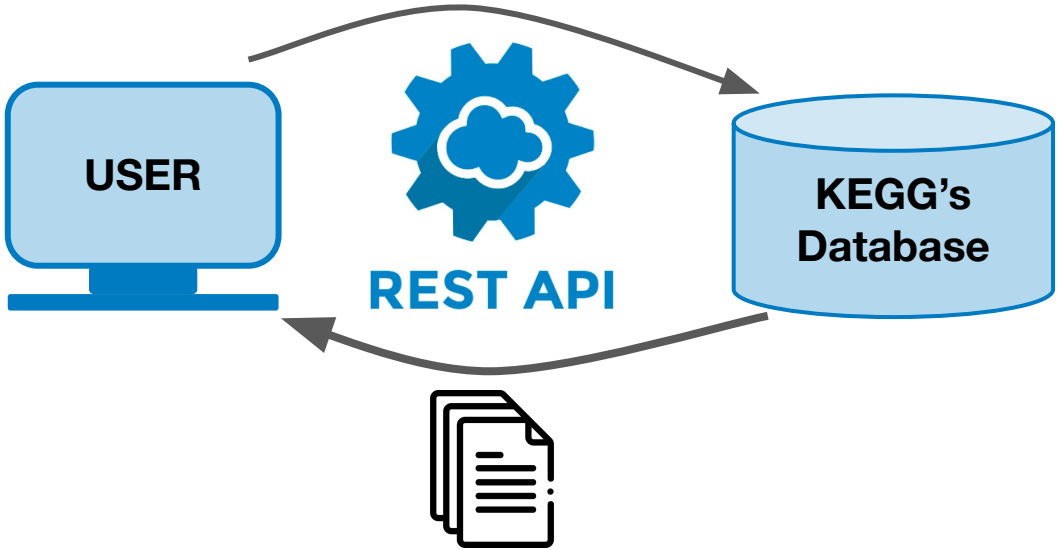
$$\text{is\_internal}_i \implies (\exists\, j \in \textbf{inTailSet}(i) : \neg\text{inverted}_j) \vee (\exists\, j \in \textbf{inHeadSet}(i) : \text{inverted}_j)$$
$$\text{is\_internal}_i \implies (\exists\, j \in \textbf{inTailSet}(i) : \text{inverted}_j) \vee (\exists\, j \in \textbf{inHeadSet}(i) : \neg\text{inverted}_j)$$

5.c. Benchmarks

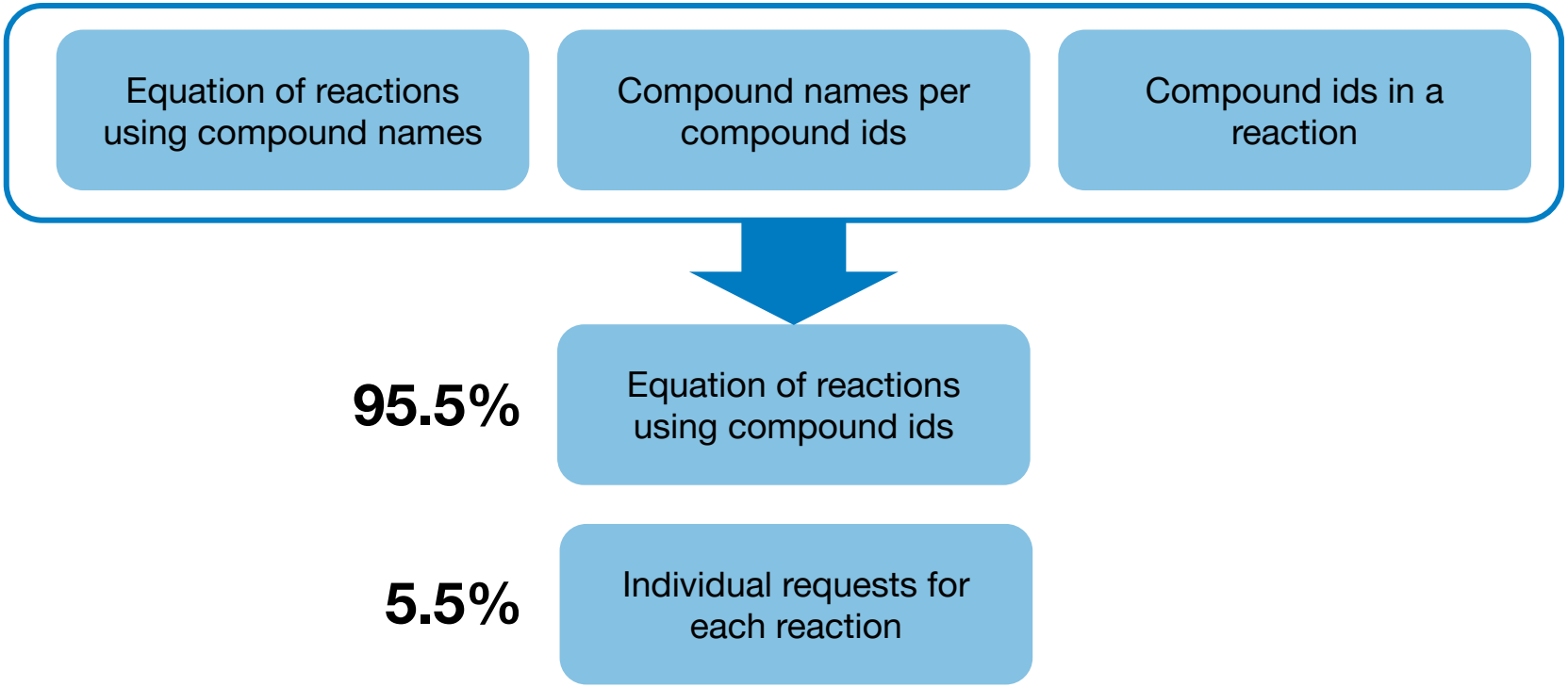
entry	solver	First Prototype	Serret DualImply	Serret UniImply	Model A	Model B
map01100	cbc	74.960s	24.274s	9.383s	8.572s	24.458s
map01110	cbc	25.927s	9.704s	4.631s	3.938s	10.407s
map01120	cbc	9.042s	3.279s	1.460s	1.186s	4.058s
map01200	cbc	0.474s	0.485s	0.266s	0.258s	0.521s
map01210	cbc	0.462s	0.507s	0.239s	0.343s	0.604s
map01212	cbc	0.467s	0.329s	0.202s	0.211s	0.385s
map01230	cbc	0.432s	0.389s	0.260s	0.243s	0.419s
map01232	cbc	0.350s	0.331s	0.224s	0.244s	0.424s
map01250	cbc	0.401s	0.382s	0.257s	0.245s	0.710s
map01240	cbc	1.148s	1.080s	0.476s	0.301s	1.373s
map01220	cbc	0.769s	0.539s	0.260s	0.276s	0.479s
map00010	cbc	0.357s	0.307s	0.273s	0.275s	0.339s
map00020	cbc	0.351s	0.299s	0.312s	0.281s	0.308s
map00030	cbc	0.351s	0.313s	0.265s	0.256s	0.365s

7. Obtaining data from KEGG

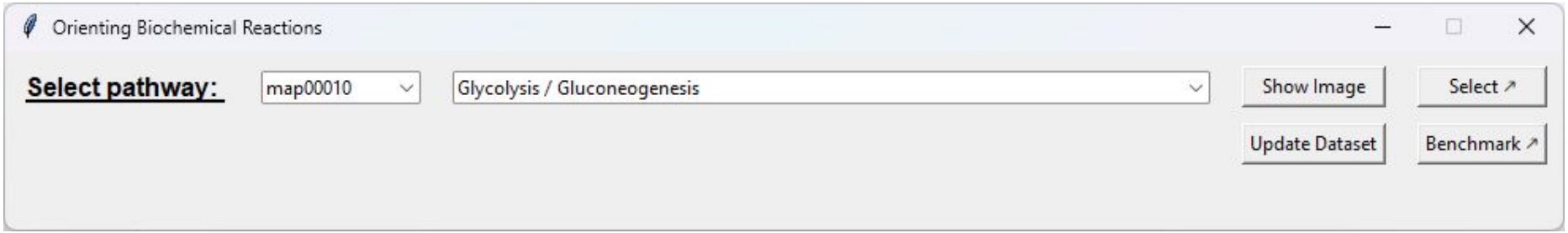
https://rest.kegg.jp/get/map00533/

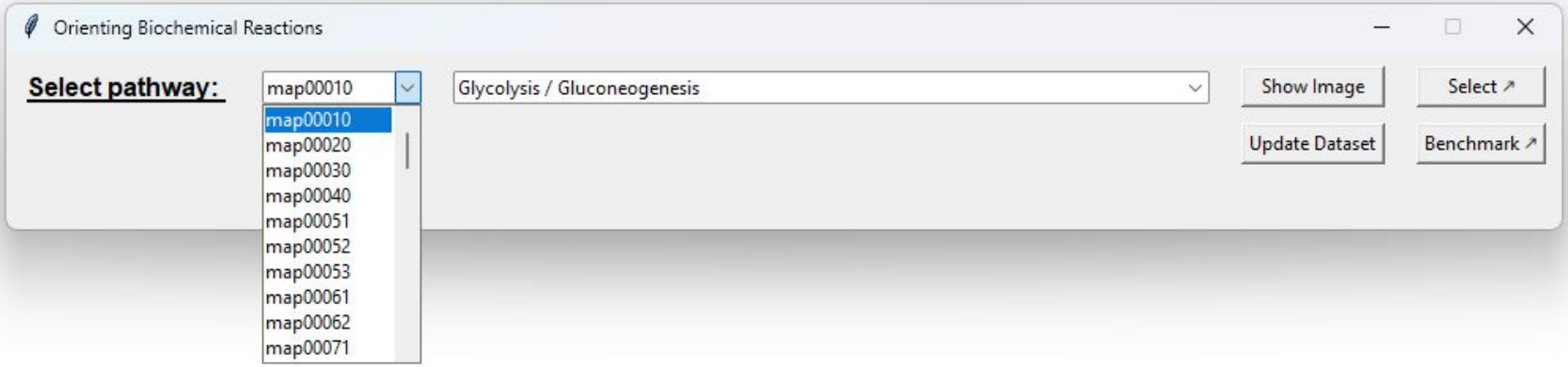


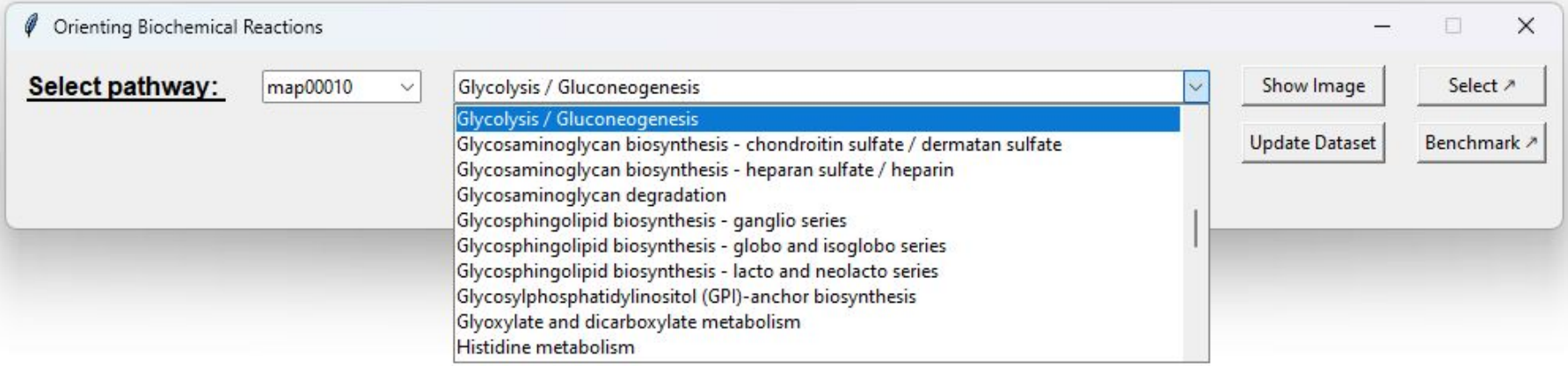
7. Obtaining data from KEGG

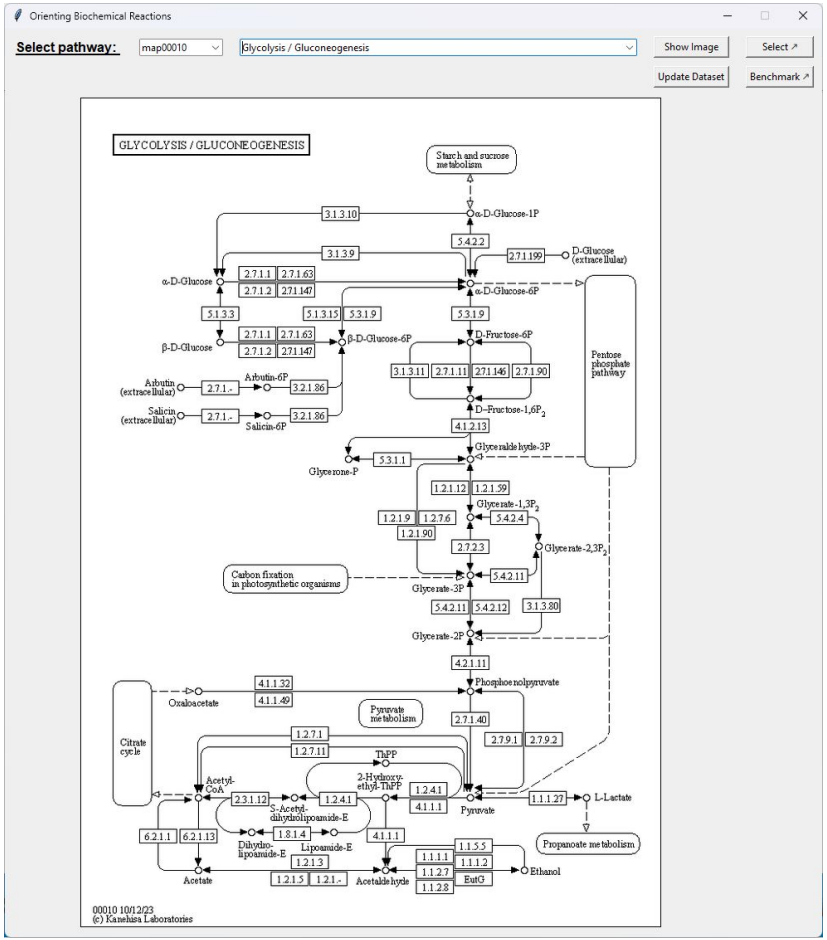


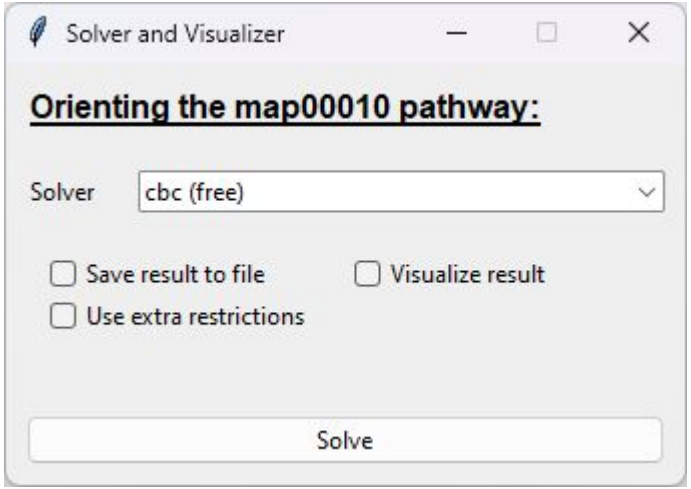
DEMONSTRATION

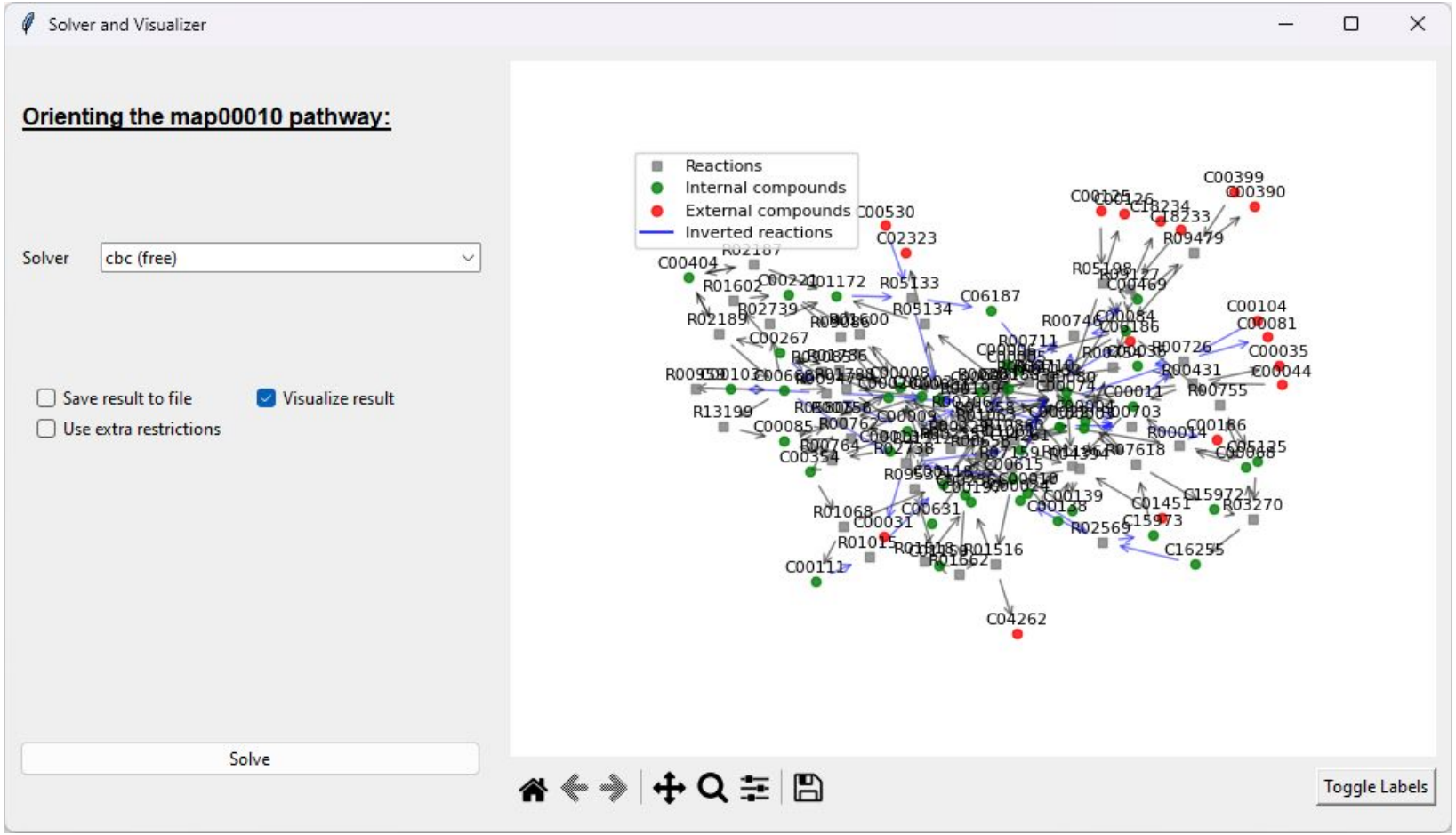

Orienting Biochemical Reactions
Select pathway:
map00010
Glycolysis / Gluconeogenesis
Show Image
Select ↗
Update Dataset
Benchmark ↗

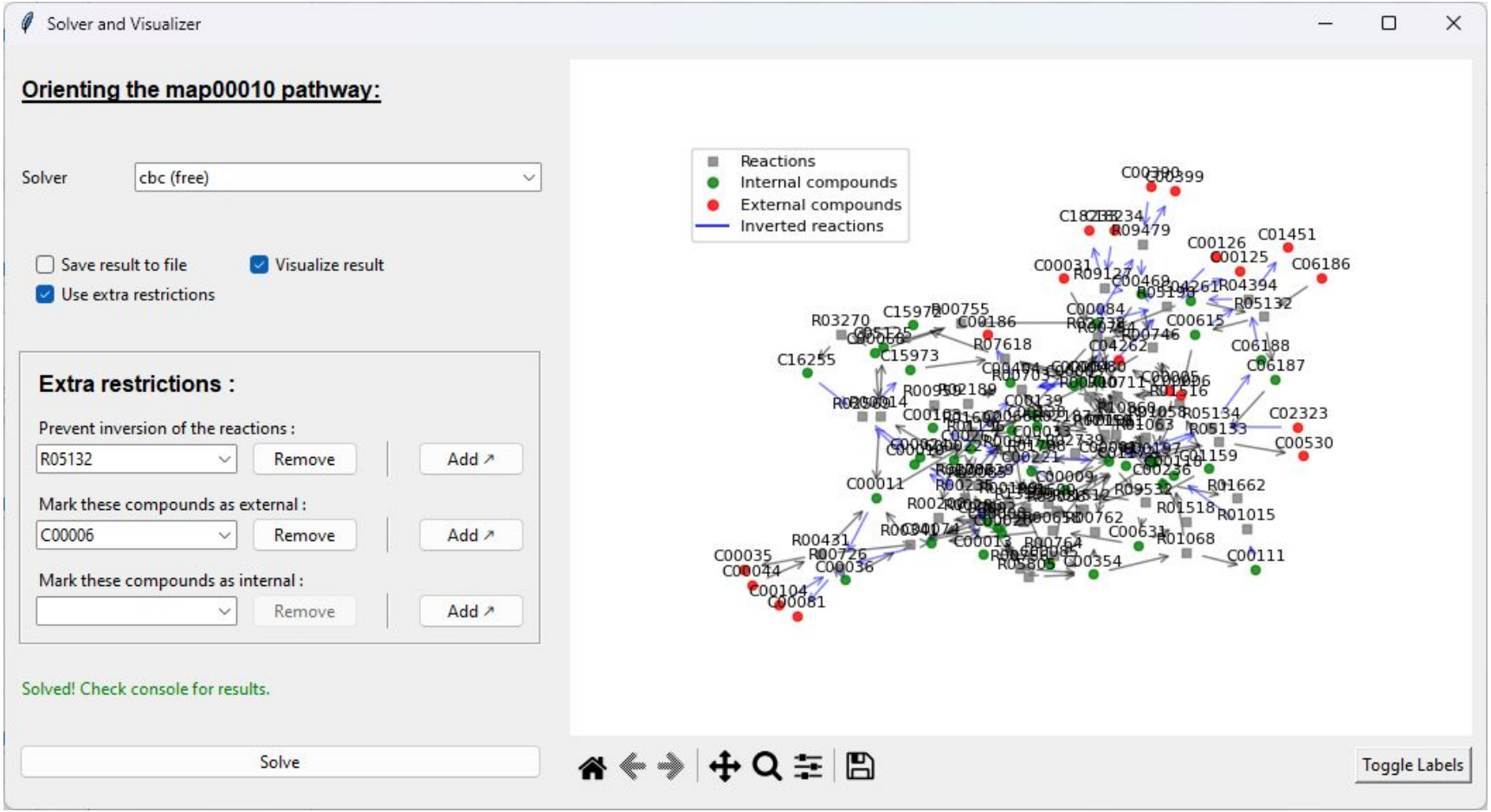

Orienting Biochemical Reactions
Select pathway:
map00010
map00010
map00020
map00030
map00040
map00051
map00052
map00053
map00061
map00062
map00071
Glycolysis / Gluconeogenesis
Show Image
Select ↗
Update Dataset
Benchmark ↗

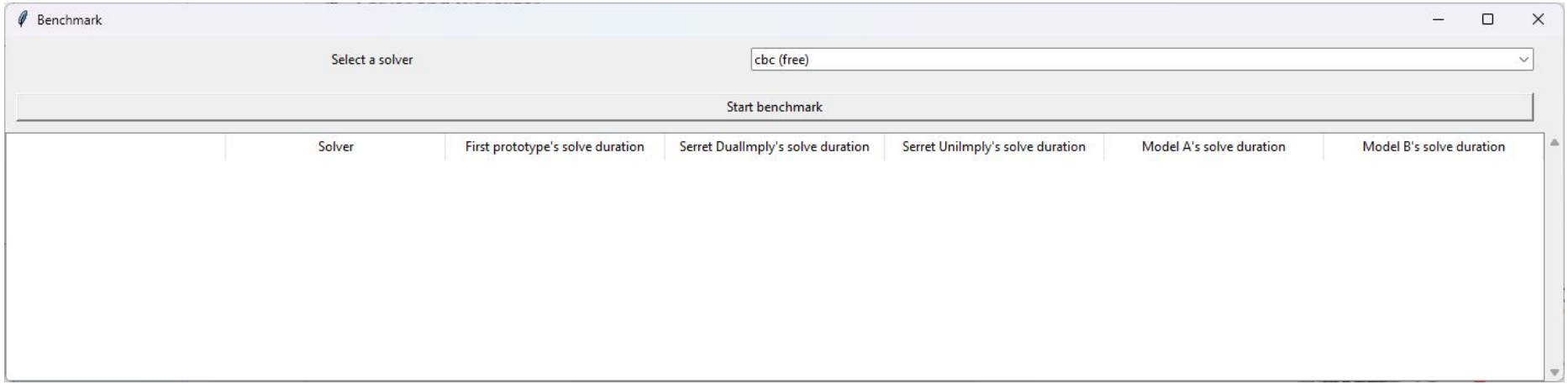

Orienting Biochemical Reactions
Select pathway:
map00010
Glycolysis / Gluconeogenesis
Glycolysis / Gluconeogenesis
Glycosaminoglycan biosynthesis - chondroitin sulfate / dermatan sulfate
Glycosaminoglycan biosynthesis - heparan sulfate / heparin
Glycosaminoglycan degradation
Glycosphingolipid biosynthesis - ganglio series
Glycosphingolipid biosynthesis - globo and isoglobo series
Glycosphingolipid biosynthesis - lacto and neolacto series
Glycosylphosphatidylinositol (GPI)-anchor biosynthesis
Glyoxylate and dicarboxylate metabolism
Histidine metabolism
Show Image
Select ↗
Update Dataset
Benchmark ↗

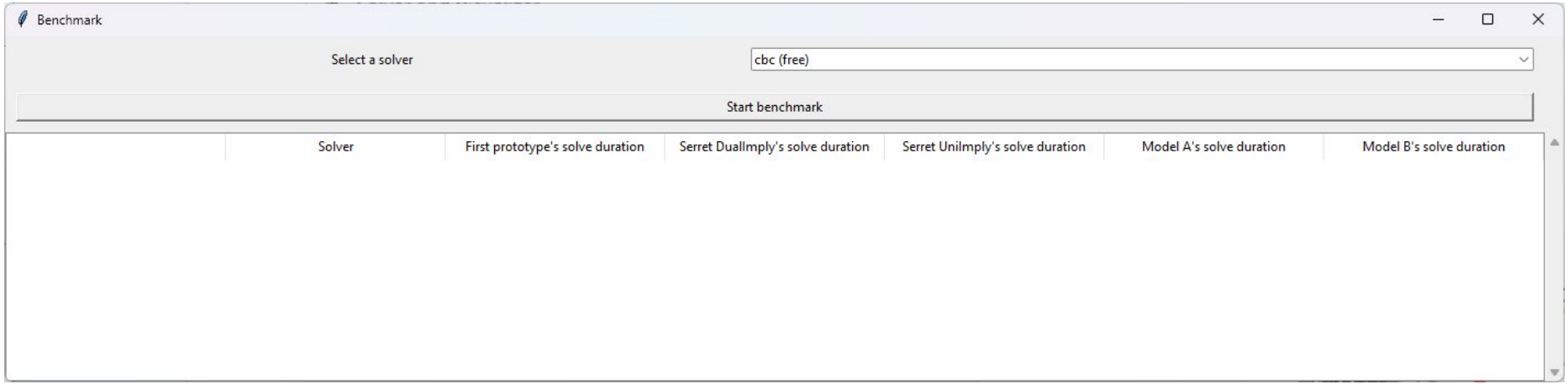

Orienting Biochemical Reactions
Select pathway:
map00010
Glycolysis / Gluconeogenesis
Show Image
Select
Update Dataset
Benchmark
GLYCOLYSIS / GLUCONEOGENESIS
Starch and sucrose metabolism
α-D-Glucose-1P
D-Glucose (extracellular)
α-D-Glucose
α-D-Glucose-6P
β-D-Glucose
β-D-Glucose-6P
D-Fructose-6P
Pentose phosphate pathway
Arbutin (extracellular)
Arbutin-6P
Salicin (extracellular)
Salicin-6P
D-Fructose-1,6P2
Glyceraldehyde-3P
Glycerone-P
Glycerate-1,3P2
Glycerate-2,3P2
Carbon fixation in photosynthetic organisms
Glycerate-3P
Glycerate-2P
Phosphoenolpyruvate
Oxaloacetate
Pyruvate metabolism
Citrate cycle
ThPP
2-Hydroxy-ethyl-ThPP
Acetyl-CoA
S-Acetyl-dihydrolipoamide-E
Pyruvate
L-Lactate
Propanoate metabolism
Dihydro-lipoamide-E
Lipoamide-E
Acetate
Acetaldehyde
Ethanol
EutG
00010 10/12/23
(c) Kanehisa Laboratories


Solver and Visualizer
Orienting the map00010 pathway:
Solver
cbc (free)
Save result to file
Visualize result
Use extra restrictions
Solve


Solver and Visualizer
Orienting the map00010 pathway:
Solver
cbc (free)
Save result to file
Visualize result
Use extra restrictions
Solve
Reactions
Internal compounds
External compounds
Inverted reactions
Toggle Labels


Solver and Visualizer
Orienting the map00010 pathway:
Solver
cbc (free)
Save result to file
Visualize result
Use extra restrictions
Extra restrictions :
Prevent inversion of the reactions :
R05132
Remove
Add ↗
Mark these compounds as external :
C00006
Remove
Add ↗
Mark these compounds as internal :
Remove
Add ↗
Solved! Check console for results.
Solve
Reactions
Internal compounds
External compounds
Inverted reactions
Toggle Labels


Benchmark
Select a solver
cbc (free)
Start benchmark
Solver
First prototype's solve duration
Serret DualImply's solve duration
Serret UniImply's solve duration
Model A's solve duration
Model B's solve duration


Benchmark
Select a solver
cbc (free)
Start benchmark
Solver
First prototype's solve duration
Serret DualImply's solve duration
Serret UniImply's solve duration
Model A's solve duration
Model B's solve duration