Learning Boolean controls in regulated metabolic networks: a case-study

Kerian Thuillier[1] Caroline Baroukh[2] Alexander Bockmayr[3]
Ludovic Cottret[2] Loïc Paulevé[4] Anne Siegel[1]

[1]Univ Rennes, Inria, CNRS, IRISA, F-35000 Rennes, France

[2]LIPME, INRAE, CNRS, Université de Toulouse, Castanet-Tolosan, France

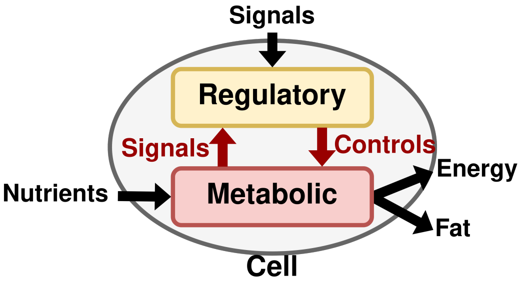
[3]Freie Universität Berlin, Institute of Mathematics, D-14195 Berlin, Germany

[4]Univ. Bordeaux, Bordeaux INP, CNRS, LaBRI, UMR5800, F-33400 Talence, France

September 2021

Multilayered structure

Context: Cells modelled as multi-layered structures



Focus

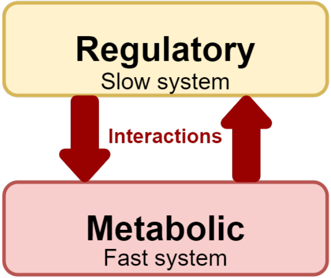
2 classes of processes

- Regulatory system
- Metabolic system

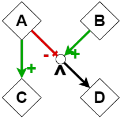
From simulation to learning

Multiplicity of formalisms

2 systems with 2 different dynamics



Discrete dynamics[1]



Model

$f_A(x_A, x_B, x_C, x_D) = x_A$

$f_B(x_A, x_B, x_C, x_D) = x_B$

$f_C(x_A, x_B, x_C, x_D) = x_A$

$f_D(x_A, x_B, x_C, x_D) = \neg x_A \wedge x_B$

Logical rules (Boolean network)

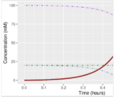
Inputs				Outputs			
x_A	x_B	x_C	x_D	x_A	x_B	x_C	x_D
0	0	x	x	0	0	0	0
0	1	x	x	0	1	0	1
1	0	x	x	1	0	1	0
1	1	x	x	1	1	1	0

Truth table (simulation)

Flow dynamics[2] *(MILP problem)*

$$\begin{aligned} \text{maximise} \quad & x_{\text{Growth}} \\ \text{such that:} \quad & S \cdot x = 0 \\ & l_r \cdot f_r \leq x_r \leq u_r \cdot f_r \qquad \forall r \in \mathcal{R} \end{aligned}$$

Regulatory flux balance analysis (rFBA)



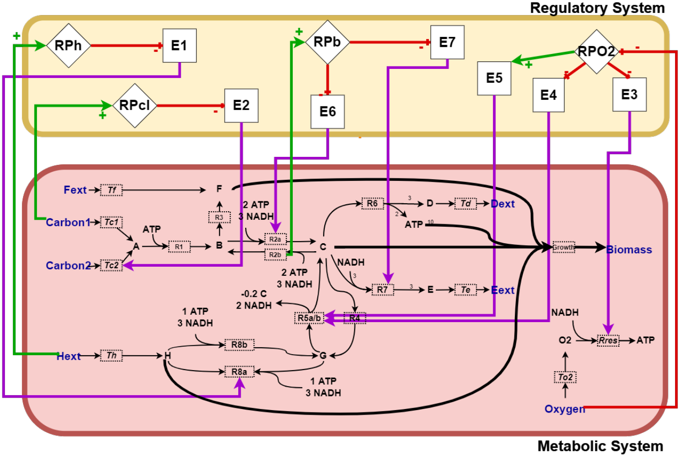
Metabolic traces

Bottleneck: coupling both system dynamics

[1] S. Videla et al., Bioinformatics, 2016

[2] M. W. Covert et al., Journal of theoretical biology, 2001

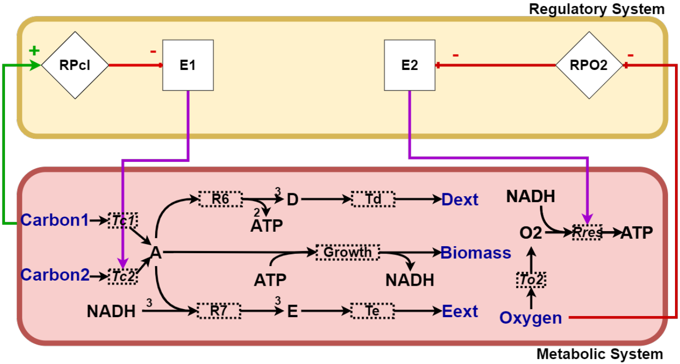
Our case study: Minitoy



Covert's regulated metabolic network of diauxic shift[1]

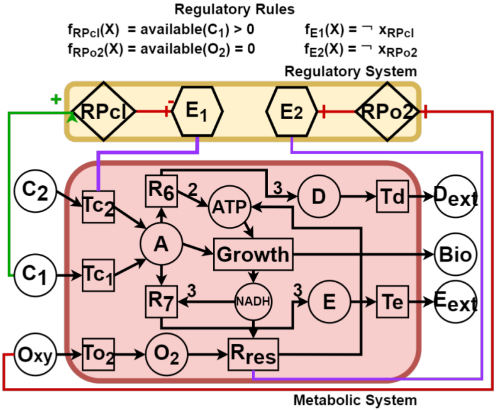
[1] M. W. Covert et al., Journal of theoretical biology, 2001

Our case study: Minitoy



Minitoy: simplified version of Covert's network

Formalism: regulated metabolic networks $\mathcal{N}=(\mathcal{M},\mathcal{R},\mathcal{S},f)$



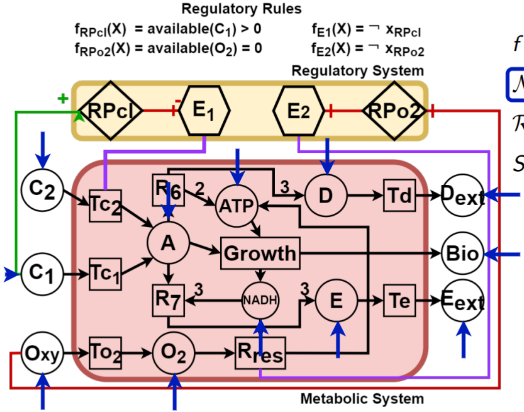
$f : \{0,1\}^n \rightarrow \{0,1\}^n$: regulatory rules

$\mathcal{M}$: metabolites

$\mathcal{R}$: chemical reactions

$\mathcal{S}$: stoichiometric matrix

Formalism: regulated metabolic networks $\mathcal{N}=(\mathcal{M},\mathcal{R},S,f)$



$f : \{0,1\}^n \rightarrow \{0,1\}^n$: regulatory rules

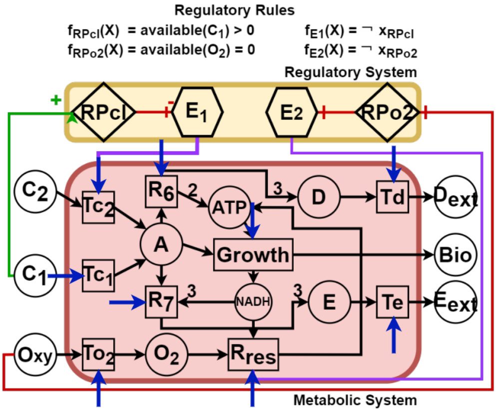
$\mathcal{M}$: metabolites

$\mathcal{R}$: chemical reactions

S: stoichiometric matrix

Metabolites = Chemical components

Formalism: regulated metabolic networks $\mathcal{N}=(\mathcal{M},\mathcal{R},S,f)$



$f : \{0,1\}^n \rightarrow \{0,1\}^n$: regulatory rules

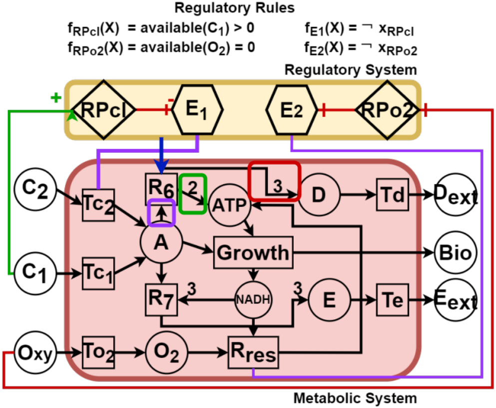
$\mathcal{M}$: metabolites

$\mathcal{R}$: chemical reactions

S: stoichiometric matrix

Chemical reactions are modelled using stoichiometric matrix

Formalism: regulated metabolic networks $\mathcal{N}=(\mathcal{M},\mathcal{R},S,f)$



$f:\{0,1\}^n \rightarrow \{0,1\}^n$: regulatory rules

$\mathcal{M}$: metabolites

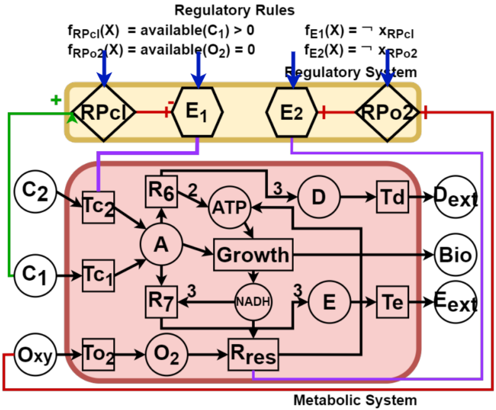
$\mathcal{R}$: chemical reactions

S: stoichiometric matrix

	T_{C1}	T_{C2}	T_{O2}	R_{res}	Growth	R_6	R_7	T_d	T_e
C1	−1	0	0	0	0	0	0	0	0
C2	0	−1	0	0	0	0	0	0	0
Oxy	0	0	−1	0	0	0	0	0	0
O2	0	0	1	−1	0	0	0	0	0
ATP	0	0	0	1	−1	2	0	0	0
NADH	0	0	0	−1	1	0	−3	0	0
A	1	1	0	0	−1	−1	−1	0	0
D	0	0	0	0	0	3	0	−1	0
E	0	0	0	0	0	0	3	0	−1
Dext	0	0	0	0	0	0	0	1	0
Eext	0	0	0	0	0	0	0	0	1
Bio	0	0	0	0	1	0	0	0	0

$$R_6: \quad 1 \cdot A \rightarrow 2 \cdot ATP + 3 \cdot D$$

Formalism: regulated metabolic networks $\mathcal{N}=(\mathcal{M},\mathcal{R},\mathcal{S},f)$



$f : \{0,1\}^n \rightarrow \{0,1\}^n$: regulatory rules

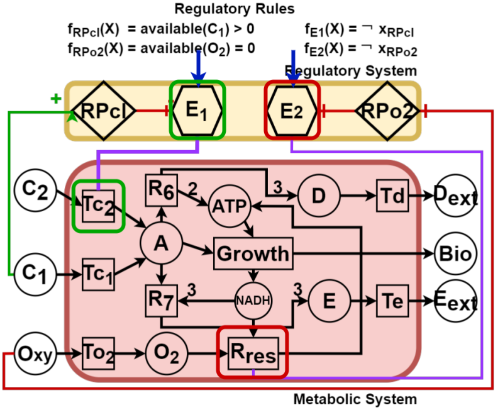
$\mathcal{M}$: metabolites

$\mathcal{R}$: chemical reactions

$\mathcal{S}$: stoichiometric matrix

Regulatory system contains: proteins + enzymes

Formalism: regulated metabolic networks $\mathcal{N}=(\mathcal{M},\mathcal{R},S,f)$



$f : \{0,1\}^n \rightarrow \{0,1\}^n$: regulatory rules

$\mathcal{M}$: metabolites

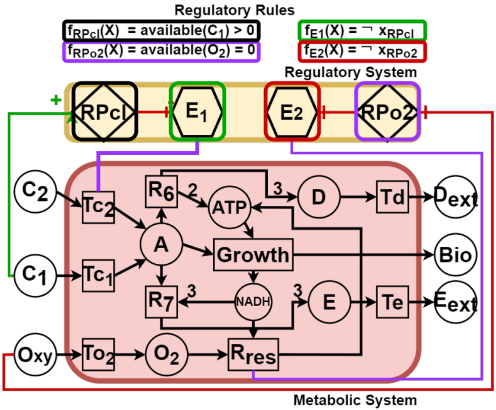
$\mathcal{R}$: chemical reactions

S: stoichiometric matrix

Enzymes control reactions

Regulatory system contains: proteins + enzymes

Formalism: regulated metabolic networks $\mathcal{N}=(\mathcal{M},\mathcal{R},S,f)$



$f : \{0,1\}^n \rightarrow \{0,1\}^n$: regulatory rules

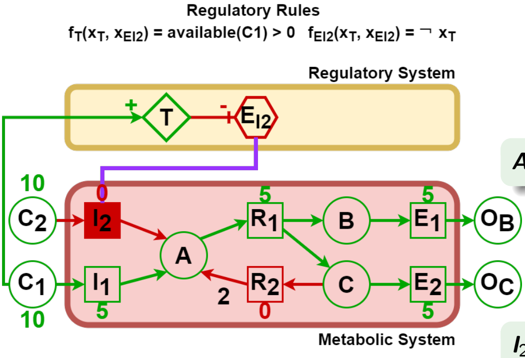
$\mathcal{M}$: metabolites

$\mathcal{R}$: chemical reactions

S: stoichiometric matrix

Regulatory system contains: proteins + enzymes

Formalism: metabolic steady states – MSS: $\boldsymbol{v} \in \mathbb{R}^{|\mathcal{R}|}$



Input and output fluxes are equals

$$\forall m \in \mathcal{M}, \sum_{\substack{r \in \mathcal{R} \\ S_{mr} > 0}} v_r = \sum_{\substack{r \in \mathcal{R} \\ S_{mr} < 0}} v_r \quad (1)$$

A: $v_{I_1} + v_{I_2} + 2 \times v_{R_2} = v_{R_1}$

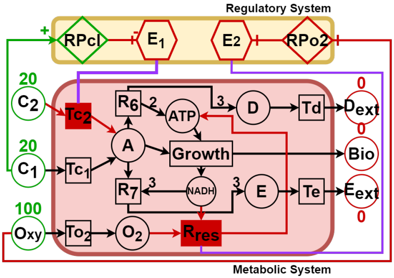
Inhibited reactions have a zero flux

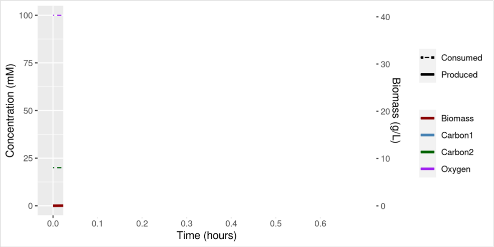
$$\forall r \in \mathcal{R}, x_{E_r} = 0 \implies v_r = 0 \quad (2)$$

I_2: $x_{E_{I_2}} = 0 \implies v_{I_2} = 0$

Steady-state assumption: no components are produced/consumed in excess

Dynamic regulatory flux balance analysis – d-rFBA

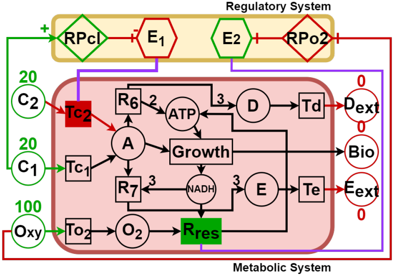


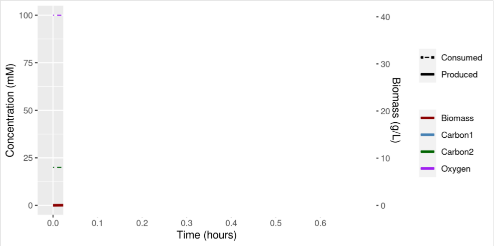


Iterating over the 3 steps:

1. Updating the regulatory system
2. Computing an optimal MSS
3. Updating the input/output

Dynamic regulatory flux balance analysis – d-rFBA





Iterating over the 3 steps:

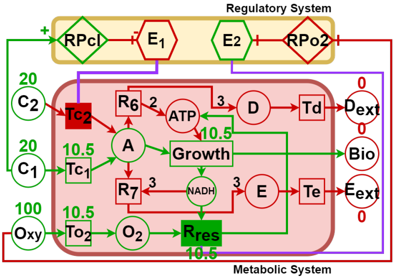
1. Updating the regulatory system
2. Computing an optimal MSS
3. Updating the input/output

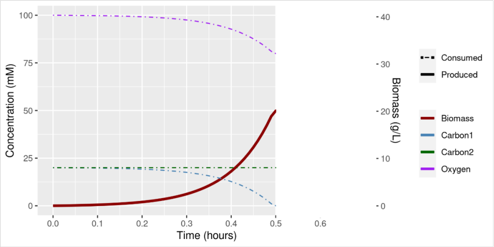
Applying synchronously the Boolean rules on previous regulatory state

$f_{\mathbf{RPcl}}(X) = \mathbf{available}(x_{C1}) > 0 \qquad f_{\mathbf{E1}}(X) = \neg x_{\mathbf{RPcl}}$

$f_{\mathbf{RPo2}}(X) = \mathbf{available}(x_{Oxy}) = 0 \qquad f_{\mathbf{E2}}(X) = \neg x_{\mathbf{RPo2}}$

Dynamic regulatory flux balance analysis – d-rFBA





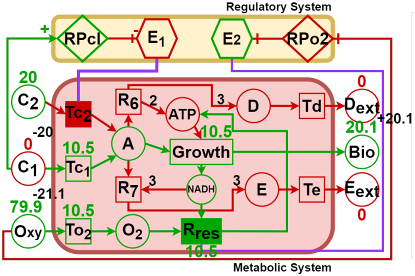
Iterating over the 3 steps:

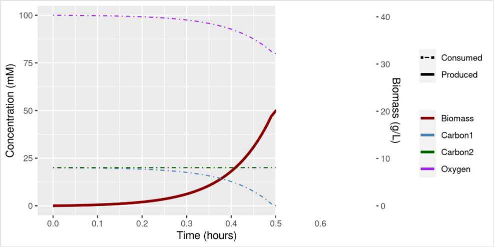
1. Updating the regulatory system
2. Computing an optimal MSS
3. Updating the input/output

Maximising an objective function

$$o(v) = v_{\text{Growth}}$$

Dynamic regulatory flux balance analysis – d-rFBA



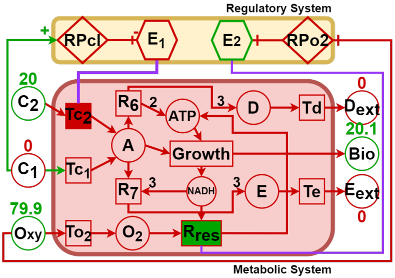


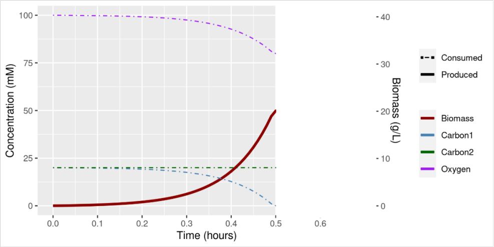
Iterating over the 3 steps:

1. Updating the regulatory system
2. Computing an optimal MSS
3. Updating the input/output

Wait that metabolites are consumed/produced before updating

Dynamic regulatory flux balance analysis – d-rFBA



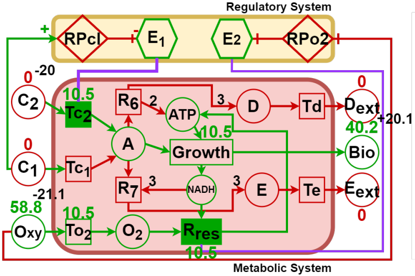


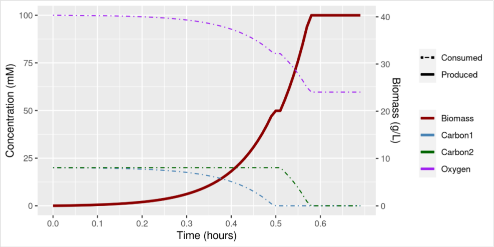
Iterating over the 3 steps:

1. Updating the regulatory system
2. Computing an optimal MSS
3. Updating the input/output

Repeat!

Dynamic regulatory flux balance analysis – d-rFBA





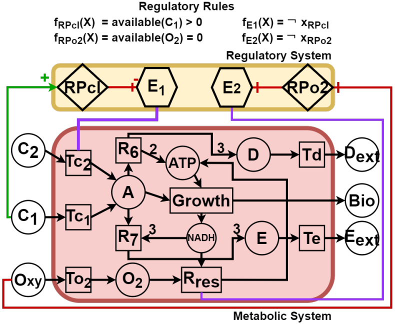
Iterating over the 3 steps:

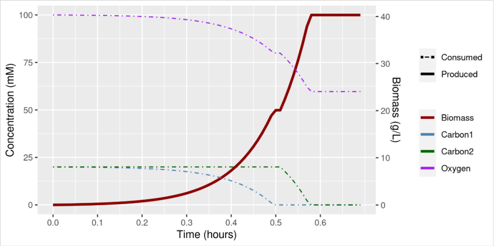
1. Updating the regulatory system
2. Computing an optimal MSS
3. Updating the input/output

Repeat!

d-rFBA: a scalable simulation framework

A coupled model can be simulated as soon as it is built





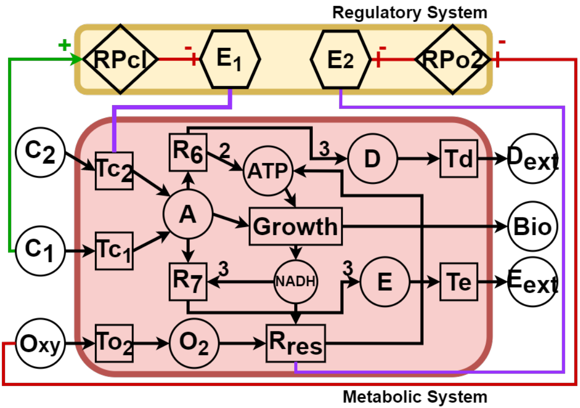
Model	Metabolic		Regulatory	
	Metabolites	Reactions	Regulatory proteins	Regulations
Toy	12	9	2	12
Covert[1]	19	20	4	20
E.coli – genome scale[2]	761	1 075	104	479

No scaling issues

[1] M. W. Covert et al., Journal of theoretical biology, 2001

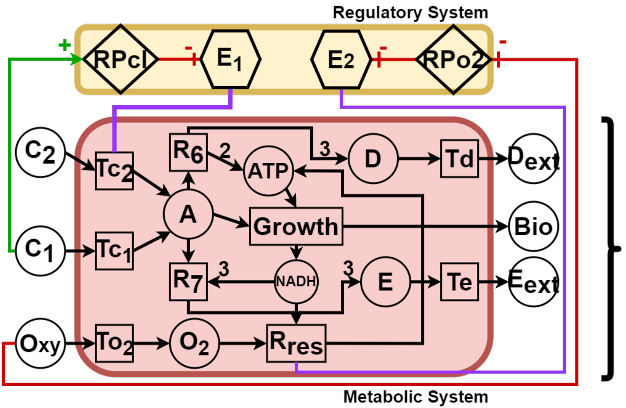
[2] M. W. Covert et al., Nature, 2004

Bottleneck: learning regulation rules of coupled systems



How to build regulatory metabolic networks ?

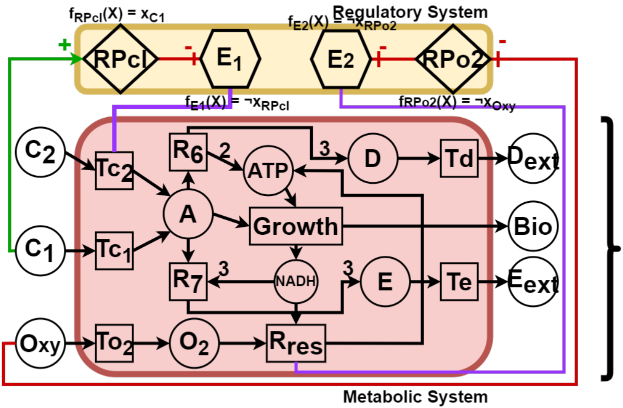
Bottleneck: learning regulation rules of coupled systems



Inferred through genome tagging

How to build regulatory metabolic networks ?

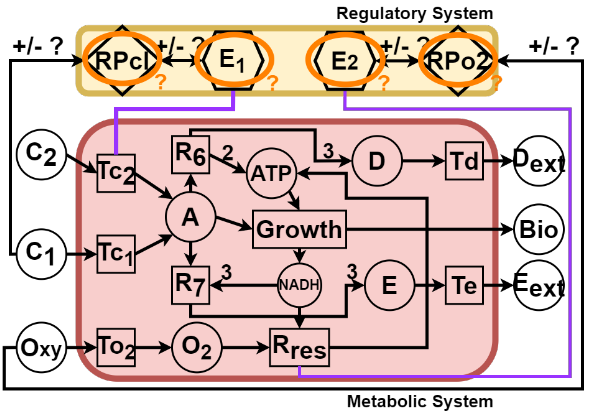
Bottleneck: learning regulation rules of coupled systems



Inferred through genome tagging

How to build regulatory metabolic networks ?

Bottleneck: learning regulation rules of coupled systems

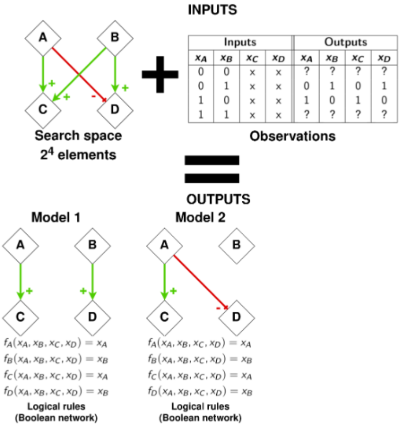


Manually inferred by researchers

Inferred through genome tagging

How to build regulatory metabolic networks ?

Reverse-engineering regulatory rules from observations



Inputs:

1. Observations
2. Domain constraints: finite set of usable interactions

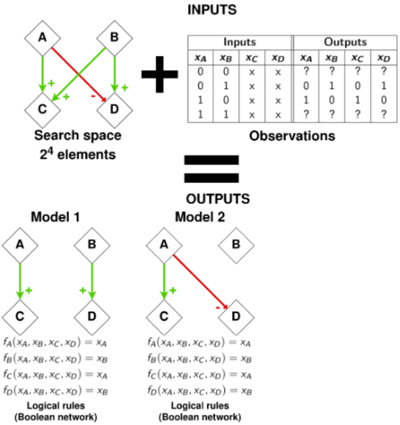
Search space: *all the Boolean networks respecting the domain constraints*

Outputs: sets of logical rules s.t.:

1. in the search domain
2. its simulations match the observations for a given semantics

Size of the search domain in $O(2^{2^n})$ with n number of domain constraints

Reverse-engineering regulatory rules from observations



Inputs:

1. Observations
2. Domain constraints: finite set of usable interactions

Search space: *all the Boolean networks respecting the domain constraints*

Outputs: sets of logical rules s.t.:

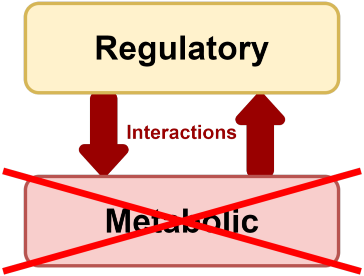
1. in the search domain
2. its simulations match the observations for a given semantics

Size of the search domain in $O(2^{2^n})$ with n number of domain constraints

State of the art: learning regulatory systems

State of the art relies on logical programming and combinatorial problem formulation

ASP: declarative programming (1^{st} order logic + SAT-based solvers)



State of the art **only** learns regulatory systems from **Boolean traces of the regulatory system**

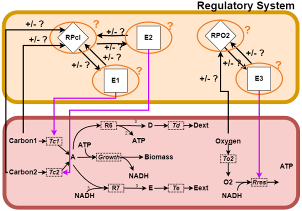
Examples: CASPO[1], CASPOTS[2] or BoNeSis[3]

1 S. Videla et al., Theoretical Computer Science, 2015

2 M. Ostrowski et al., BioSystems, 2016

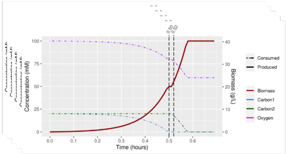
3 L. Paulevé et al., Nature Communications, 2020

Our issue: reverse-engineering coupled system from metabolic traces



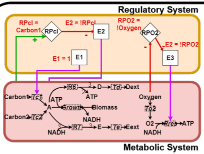
Search domain

+



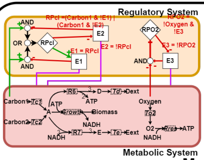
Traces of regulated metabolic network

=



RPcl = Carbon1
RPO2 = ¬Oxygen
E1 = 1
E2 = ¬RPcl
E3 = ¬RPO2

Model 1



RPcl = (Carbon1 ∧ ¬E1) ∨ (Carbon1 ∧ ¬E2)
RPO2 = ¬Oxygen ∧ ¬E3
E1 = RPcl
E2 = ¬RPcl
E3 = ¬RPO2

Model 2

Instance	Search domain size
Minitoy	1 944 320
Covert	2.9×10^{12}

Huge search space
Hybrid problem: combinatorial + linear

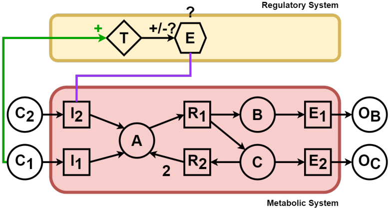
How to learn regulation rules from d-rFBA observations ?

Input metabolites:

$C_1 = C_2 = 5$

Objective function:

$o(v) = v_{E1}$



Observed value:

$V_{E1} = 5$

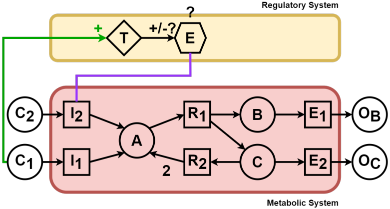
How to learn regulation rules from d-rFBA observations ?

Input metabolites:

$C_1 = C_2 = 5$

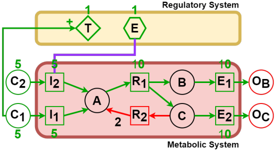
Objective function:

$o(v) = v_{E1}$



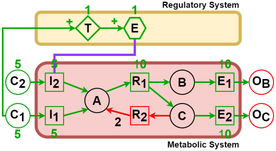
Observed value:

$V_{E1} = 5$



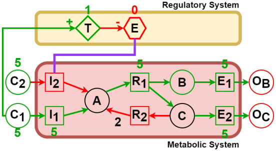
$f_E(x_T, x_E) = 1$

$v_{E1} = 10$



$f_E(x_T, x_E) = x_T$

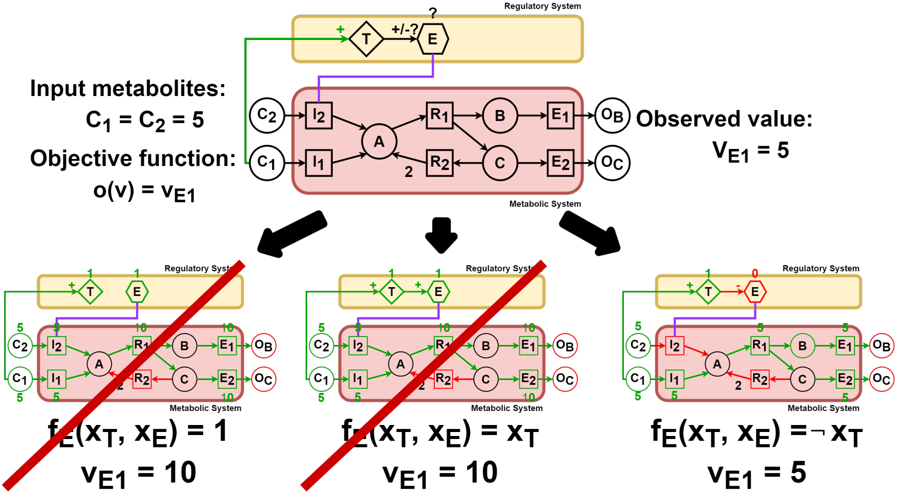
$v_{E1} = 10$



$f_E(x_T, x_E) = \neg x_T$

$v_{E1} = 5$

How to learn regulation rules from d-rFBA observations ?



According to the objective function maximisation assumption
$v_{E1} = 5$ **is observed** $\implies$ **the regulation rules do not allow to have** $v_{E1} > 5$

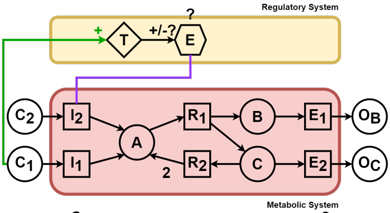
How to learn regulation rules from d-rFBA observations ?

Input metabolites:

$C_1 = C_2 = 5$

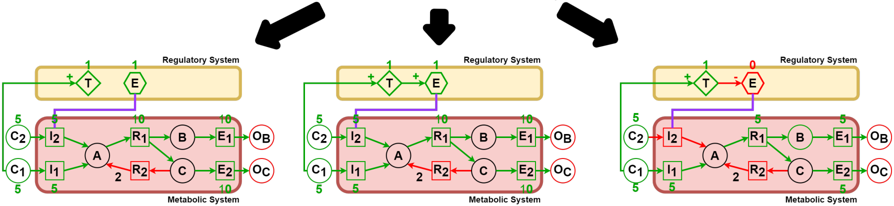
Objective function:

$o(v) = v_{I1}$



Observed value:

$V_{E1} = 5$



$f_E(x_T, x_E) = 1$

$v_{I1} = 5$

$f_E(x_T, x_E) = x_T$

$v_{I1} = 5$

$f_E(x_T, x_E) = \neg x_T$

$v_{I1} = 5$

The choice of objective function is really important

Formalisation of the inference problem – Definition

Inputs: metabolic network $\mathcal{N} = (\mathcal{M}, \mathcal{R}, S, l, u)$ + regulatory proteins $\mathcal{P}$ + search space $\mathbb{F}$ + time series T

Outputs: subset-minimal Boolean networks $f \in \mathbb{F}$ such that:

$$\forall (s, s') \in T, \forall (\hat{v}, \hat{w}, \hat{x}) \in \mathrm{rMSS}(\mathcal{N}, \mathcal{P}, f),$$
$$\hat{w}_{\mathrm{Inp}} = w'_{\mathrm{Inp}} \land \hat{x}_{\mathcal{P}} = x'_{\mathcal{P}} \implies \hat{v}_{\mathrm{Growth}} \leq v'_{\mathrm{Growth}}$$

Hybrid problem: combinatorial + linear

Formalisation of the inference problem – Definition

Inputs: metabolic network $\mathcal{N} = (\mathcal{M}, \mathcal{R}, S, l, u)$ + regulatory proteins $\mathcal{P}$ + search space $\mathbb{F}$ + time series T

Combinatorial part

Outputs: subset-minimal Boolean networks $f \in \mathbb{F}$ such that:

Linear part

$$\forall (s, s') \in T, \forall (\hat{v}, \hat{w}, \hat{x}) \in \text{rMSS}(\mathcal{N}, \mathcal{P}, f),$$
$$\hat{w}_{\text{Inp}} = w'_{\text{Inp}} \wedge \hat{x}_{\mathcal{P}} = x'_{\mathcal{P}} \implies \hat{v}_{\text{Growth}} \leq v'_{\text{Growth}}$$

Hybrid problem: combinatorial + linear

Formalisation of the inference problem – Definition

Inputs: metabolic network $\mathcal{N} = (\mathcal{M}, \mathcal{R}, S, l, u)$ + regulatory proteins $\mathcal{P}$ + search space $\mathbb{F}$ + time series T

Combinatorial part

Outputs: subset-minimal Boolean networks $f \in \mathbb{F}$ such that:

Combinatorial optimisation

Linear part

$$\forall (s, s') \in T, \forall (\hat{v}, \hat{w}, \hat{x}) \in \text{rMSS}(\mathcal{N}, \mathcal{P}, f),$$
$$\hat{w}_{\text{Inp}} = w'_{\text{Inp}} \wedge \hat{x}_{\mathcal{P}} = x'_{\mathcal{P}} \implies \hat{v}_{\text{Growth}} \leq v'_{\text{Growth}}$$

Linear optimisation

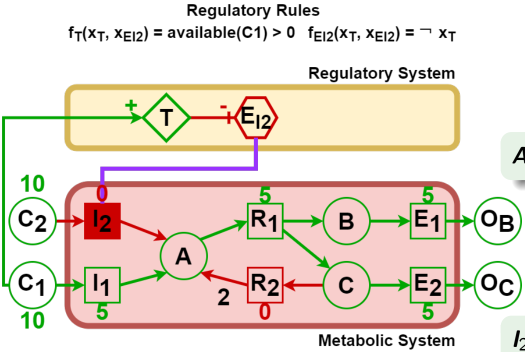
Hybrid problem: combinatorial + linear

Contributions on the inferring of regulatory rules

1. Boolean abstraction of the d-rFBA framework
2. Relaxation of the inferring problem as a combinatorial problem
3. ASP based resolution scheme for the relaxed problem

Solve the inferring problem as a purely combinatorial problem

Metabolic steady states – MSS: $v \in \mathbb{R}^{|\mathcal{R}|}$



Input and output fluxes are equals

$$\forall m \in \mathcal{M}, \sum_{\substack{r \in \mathcal{R} \\ S_{mr} > 0}} v_r = \sum_{\substack{r \in \mathcal{R} \\ S_{mr} < 0}} v_r \quad (1)$$

***A*:** $v_{I_1} + v_{I_2} + 2 \times v_{R_2} = v_{R_1}$

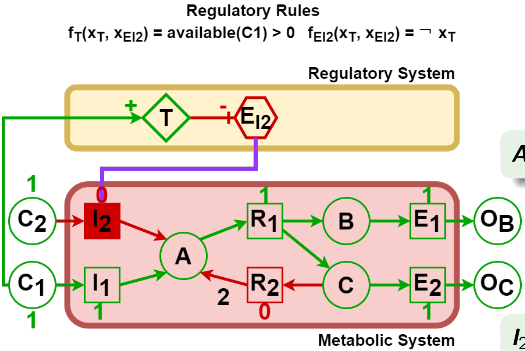
Inhibited reactions have a zero flux

$$\forall r \in \mathcal{R}, x_{E_r} = 0 \implies v_r = 0 \quad (2)$$

***I*₂:** $x_{E_{I_2}} = 0 \implies v_{I_2} = 0$

Abstracting elements according to whether they are present or not

Boolean metabolic steady states – $MSS^{\mathbb{B}}$: $\boldsymbol{v} \in \{0,1\}^{|\mathcal{R}|}$



***m* ∈ $\mathcal{M}$ is produced iff it is consumed**

$$\forall m \in \mathcal{M}, \bigvee_{\substack{r \in \mathcal{R} \\ S_{mr} > 0}} v_r \iff \bigvee_{\substack{r \in \mathcal{R} \\ S_{mr} < 0}} v_r \quad (1)$$

***A*:** $v_{I_1} \vee v_{I_2} \vee v_{R_2} \iff v_{R_1}$

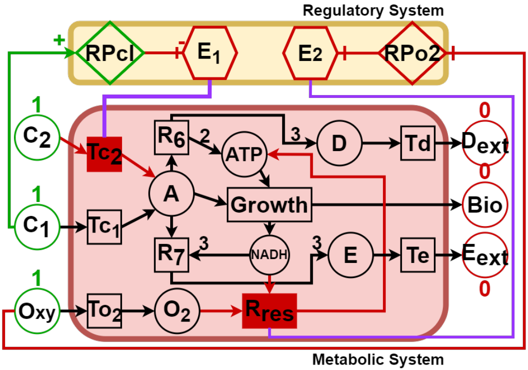
Inhibited reactions have a zero flux

$$\forall r \in \mathcal{R}, x_{E_r} = 0 \implies v_r = 0 \quad (2)$$

***I_2*:** $x_{E_{I_2}} = 0 \implies v_{I_2} = 0$

Abstracting elements according to whether they are present or not

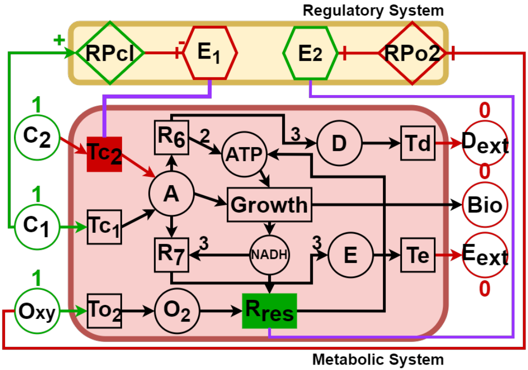
Boolean abstraction of d-rFBA



Iterating over the 3 steps:

1. Updating the regulatory system
2. Computing an optimal $MSS^{\mathbb{B}}$
3. Updating the input/output

Boolean abstraction of d-rFBA



Iterating over the 3 steps:

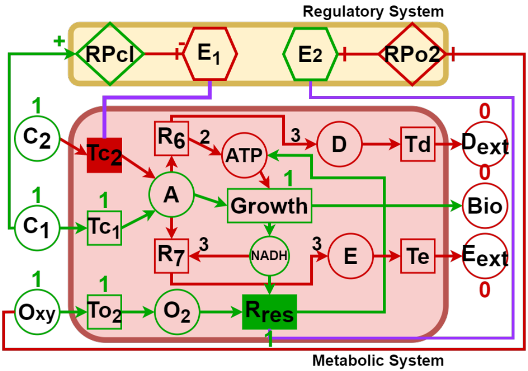
1. Updating the regulatory system
2. Computing an optimal $MSS^{\mathbb{B}}$
3. Updating the input/output

Applying synchronously the Boolean rules on previous regulatory state

$f_{\mathbf{RPcl}}(X')=x'_{C1}$ $\quad f_{\mathbf{E1}}(X')=\neg x'_{\mathbf{RPcl}}$

$f_{\mathbf{RPo2}}(X')=\neg x'_{Oxy}$ $\quad f_{\mathbf{E2}}(X')=\neg x'_{\mathbf{RPo2}}$

Boolean abstraction of d-rFBA



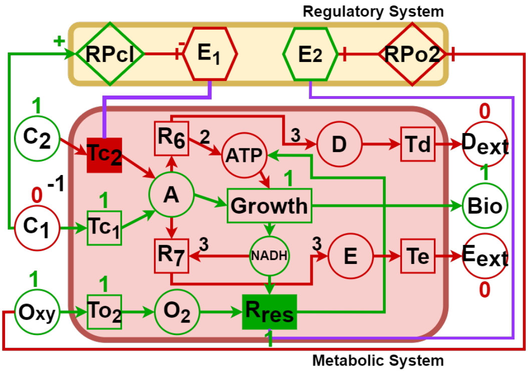
Iterating over the 3 steps:

1. Updating the regulatory system
2. Computing an optimal $MSS^{\mathbb{B}}$
3. Updating the input/output

Maximising an objective function

$$o(v) = x_{\mathbf{Tc1}} + x_{\mathbf{Tc2}} + \mathbf{To2}$$

Boolean abstraction of d-rFBA

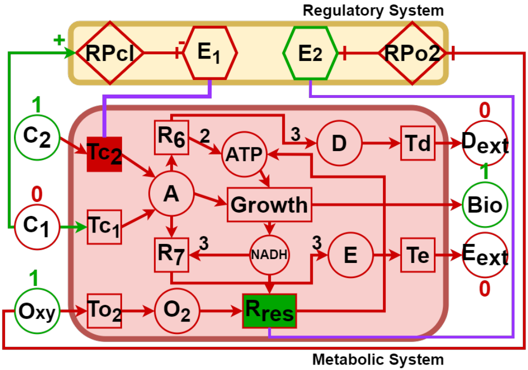


Iterating over the 3 steps:

1. Updating the regulatory system
2. Computing an optimal MSS$^{\mathbb{B}}$
3. Updating the input/output

Wait that metabolites are consumed/produced before updating

Boolean abstraction of d-rFBA

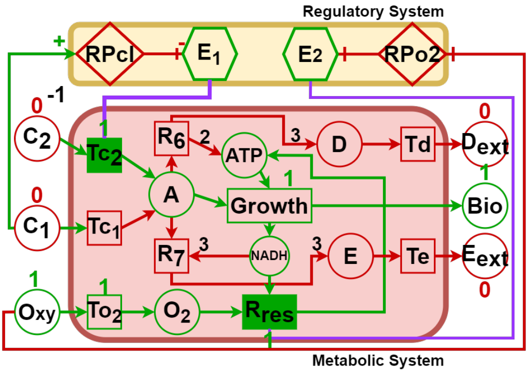


Iterating over the 3 steps:

1. Updating the regulatory system
2. Computing an optimal $MSS^{\mathbb{B}}$
3. Updating the input/output

Repeat!

Boolean abstraction of d-rFBA



Iterating over the 3 steps:

1. Updating the regulatory system
2. Computing an optimal $MSS^{\mathbb{B}}$
3. Updating the input/output

Repeat!

Boolean relaxation of the inference problem – Definition

Inputs: metabolic network $\mathcal{N} = (\mathcal{M}, \mathcal{R}, S, l, u)$ + regulatory protein $\mathcal{P}$ + search space $\mathbb{F}$ + time series $T^{\mathbb{B}}$ + objective function $\hat{o}$

Outputs: all the subset-minimal Boolean networks $f \in \mathbb{F}$ such that:

$$
\begin{aligned}
&\exists f \in \mathbb{F}, \forall (t_1, t_2) \in T^{\mathbb{B}},\\
&\qquad \exists x \in \mathrm{MSS}^{\mathbb{B}}(\mathcal{N}, t_2),\ x \preceq f(t_1),\\
&\qquad \forall y \in \mathrm{MSS}^{\mathbb{B}}(\mathcal{N}, t_2),\ y \not\preceq f(t_1) \lor \hat{o}(y) \leq \hat{o}(x)
\end{aligned}
$$

where $\forall x, y \in \mathbb{B}^n, x \preceq y \iff \forall i \in \{1, \cdots, n\}, x_i \leq y_i$
and $MSS^{\mathbb{B}}(\mathcal{N}, t)$ is the set of admissible metabolic state of $\mathcal{N}$ at time t

2-QBF problem $\Rightarrow$ Σ_2^P-complete[1]

[1] T. Eiter and G. Gottlob, Annals of Mathematics and Artificial Intelligence, 1995

Boolean relaxation of the inference problem – Definition

Inputs: metabolic network $\mathcal{N} = (\mathcal{M}, \mathcal{R}, S, l, u)$ + regulatory protein $\mathcal{P}$ + search space $\mathbb{F}$ + time series $T^{\mathbb{B}}$ + objective function $\hat{o}$

Outputs: all the subset-minimal Boolean networks $f \in \mathbb{F}$ such that:

$$\exists f \in \mathbb{F}, \forall (t_1, t_2) \in T^{\mathbb{B}},$$
$$\exists x \in \mathrm{MSS}^{\mathbb{B}}(\mathcal{N}, t_2),\ x \preceq f(t_1),$$
$$\forall y \in \mathrm{MSS}^{\mathbb{B}}(\mathcal{N}, t_2),\ y \npreceq f(t_1) \vee \hat{o}(y) \leq \hat{o}(x)$$

2-QBF form

$\exists x, \forall y, \Phi(x, y)$

where $\forall x, y \in \mathbb{B}^n,\ x \preceq y \iff \forall i \in \{1, \cdots, n\}, x_i \leq y_i$
and $MSS^{\mathbb{B}}(\mathcal{N}, t)$ is the set of admissible metabolic state of $\mathcal{N}$ at time t

2-QBF problem $\Rightarrow \Sigma_2^{\mathrm{P}}$-complete[1]

[1] T. Eiter and G. Gottlob, Annals of Mathematics and Artificial Intelligence, 1995

Answer Set Programming – ASP

Answer Set Programming – ASP[1]

```
b :- a. # Rule
a :-. # Fact
:- c. # Integrity rule
```

where *a*, *b*, *c* are atoms

Only solution: $\{a, b\}$

- Declarative framework
- Based on first-order logic predicates
  - Rules have the shape: <head> :- <body>
- Use to solve 2-QBF problems
  - Saturation technique[2]

Declarative framework allowing solving combinatorial satisfaction problems

[1] C. Baral, Cambridge University Press, 2003

[2] M. Gesber, Theory and Practice of Logic Programming, 2011

Modelling of the relaxed problem with ASP

```
{clause(N,1..C,L,S): in(L,N,S), maxC(N,C), node(N)}.

:- clause(N,_,L,S), clause(N,_,L,-S).
1 {constant(N,(-1;1)) } 1 :- node(N), not clause(N,_,_,_).
constant(N) :- constant(N,_).

size(N,C,X) :- X = #count{L, S: clause(N,C,L,S)}, clause(N,C,_,_).
:- clause(N,C,_,_), not clause(N,C-1,_,_), C > 1.
:- size(N,C1,X1), size(N,C2,X2), X1 < X2, C1 > C2.
clausediff(N,C1,C2,L) :- clause(N,C1,L,_), not clause(N,C2,L,_), clause(N,C2,_,_),
                         C1 != C2.
mindiff(N,C1,C2,L) :- clausediff(N,C1,C2,L),
                      L <= L': clausediff(N,C1,C2,L'), clause(N,C1,L',_), C1!=C2.
:- size(N,C1,X), size(N,C2,X), C1 > C2,
   mindiff(N,C1,C2,L1), mindiff(N,C2,C1,L2), L1 < L2.
:- size(N,C1,X1), size(N,C2,X2), C1 != C2, X1 <= X2,
   clause(N,C2,L,S): clause(N,C1,L,S).

update(T1,A) :- mode(T1,reg), node(A), not inp(A,_).
mode(T1,reg) :- next(T1,_).

constant(A,-1) :- inp(A,_).
:- constant(A), not inp(A,_).

eval(T,A,C,-1) :- update(T,A), clause(A,C,L,V), read(T,L,-V).
eval(T,A,C,1) :- read(T,L,V): clause(A,C,L,V); update(T,A), clause(A,C,_,_).
eval(T,A,1) :- eval(T,A,C,1), clause(A,C,_,_).
eval(T,A,-1) :- eval(T,A,C,-1): clause(A,C,_,_); update(T,A), clause(A,C,_,_).
eval(T,A,V) :- update(T,A), constant(A,V).

w(T2,A,V) :- inp(A,_), next(_,T2), obs(T2,A,V).
w(T2,A,V) :- next(T1,T2), not inp(A,_), not update(T1,A), v(T1,A,V).
w(T2,A,V) :- next(T1,T2), update(T1,A), eval(T1,A,V).

read(T,A,V) :- next(T,_), not inp(A,_), v(T,A,V).
read(T,A,V) :- next(T,T2), inp(A,_), obs(T2,A,V).

inp(X,R) :- reactant(X,R), not product(X,_).
r(r,A,R) :- reactant(A,R), product(A,_). r(p,A,R) :- product(A,R), reactant(A,_).
varm(A) :- r(_,A,_). varm(A) :- r(_,_,A). varm(A) :- inp(A,_).
time(T1) :- next(T1,_). time(T2) :- next(_,T2).

1 { v(T,A,(1;-1)) } 1 :- time(T), varm(A).
:- obs(T,A,V), v(T,A,-V).
:- time(T), r(S,A,_), v(T,A,1), v(T,R,-1): r(S,A,R).
:- time(T), r(_,A,R), v(T,R,1), v(T,A,-1).
:- time(T), inp(X,R), v(T,X,-1), v(T,R,1).

varx(A) :- node(A), not varm(A).
1{v(T,A,(-1;1))}1 :- varx(A), time(T).
:- varx(A), w(T,A,V), v(T,A,-V).
:- w(T,A,-1), v(T,A,1), node(A).

z(T,A,1);z(T,A,-1) :- time(T), varm(A).

no_rmss(T) :- inp(A,_), v(T,A,V), z(T,A,-V).
no_rmss(T) :- time(T), r(S,A,_), z(T,A,1), z(T,R,-1): r(S,A,R).
no_rmss(T) :- time(T), r(_,A,R), z(T,R,1), z(T,A,-1).
no_rmss(T) :- time(T), inp(X,R), z(T,X,-1), z(T,R,1).

no_rmss(T):- varx(A), w(T,A,V), z(T,A,-V).
n_rmss(T) :- w(T,A,-1), z(T,A,1), node(A).

valid(T) :- time(T), no_rmss(T).
valid(T) :- time(T), score(T,v,V), score(T,o,O), V <= O.

z(T,A,-V) :- time(T), varm(A), z(T,A,V), valid(T).
:- next(_, T), time(T), not valid(T).

#show.
#show clause/4.
```

Boolean network dynamics[1]

Observed Boolean metabolic steady state

Optimal Boolean metabolic steady state

Relaxed problem solved with ASP

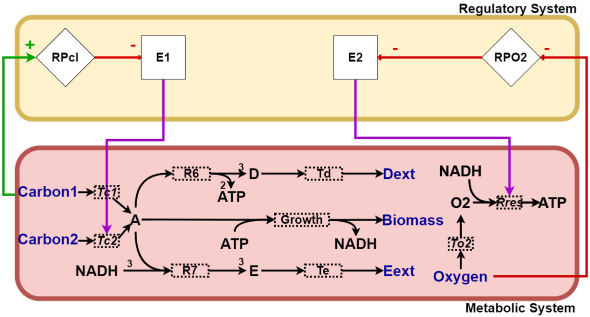
Model composed of 3 parts:

1. Boolean network discrete dynamics[1]
2. Boolean metabolic steady states
3. Computing the optimal Boolean metabolic steady states

– 2-QBF part –

[1] S. Chevalier et al., International Conference on Tools with Artificial Intelligence, 2019

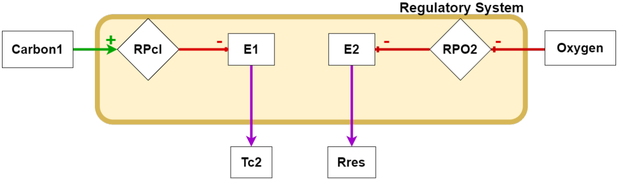
Minitoy: based on Covert's regulated metabolic network[1]



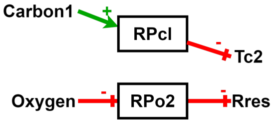
Our case study: Minitoy

[1] M. W. Covert et al., Journal of theoretical biology, 2001

Minitoy: based on Covert's regulated metabolic network[1]



Extended representation

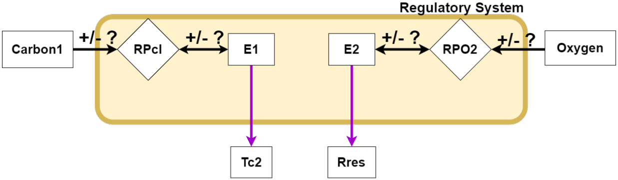


Simplified representation

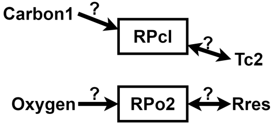
Set of regulations that must be retrieved

[1] M. W. Covert et al., Journal of theoretical biology, 2001

Minitoy: based on Covert's regulated metabolic network[1]



Extended representation

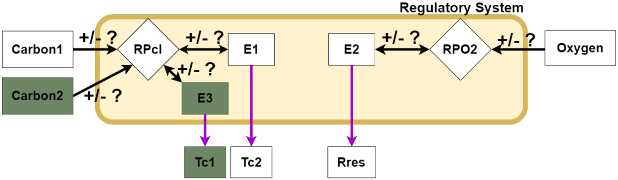


Simplified representation

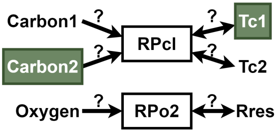
Remove the direction and the sign of each regulation

[1] M. W. Covert et al., Journal of theoretical biology, 2001

Minitoy: based on Covert's regulated metabolic network[1]



Extended representation



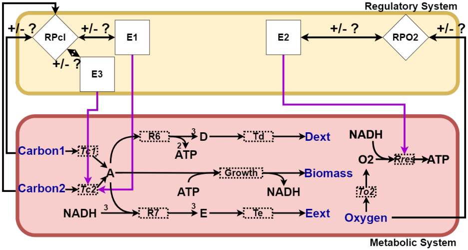
Simplified representation

Extend the search space by adding new domain constraints

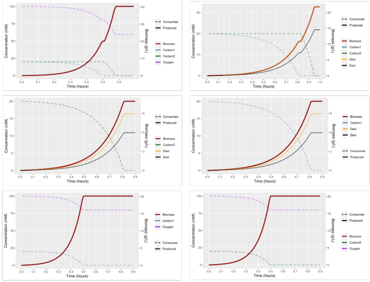
[1] M. W. Covert et al., Journal of theoretical biology, 2001

Instance of the relaxed problem

The 6 input simulations adapt from the litterature[1]



+



Boolean objective function $\hat{o}(v) = \sum_{r \in \text{Inputs}} v_r$

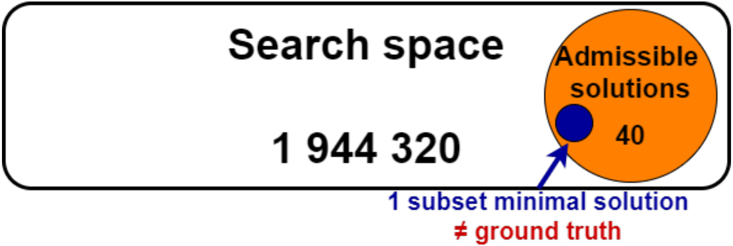
Search space contains 1 944 320 elements

[1] M. W. Covert et al., Journal of theoretical biology, 2001

Exact resolution of the relaxed problem

Exact resolution of the relaxed problem !

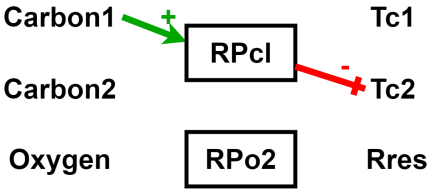
All the solutions have been found and enumerated



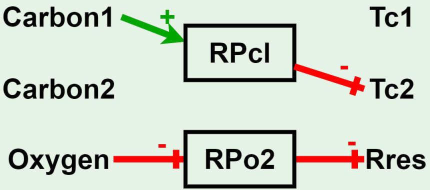
From 1 944 320 elements, 40 has been inferred
1 of which is subset minimal

Subset minimal results of the relaxed problem

40 admissible BNs of which 1 is subset minimal



Subset minimal BN



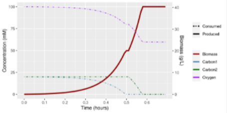
Ground truth BN

Subset minimal model is smaller than the ground truth model
Confirm an assumption made in the litterature[1]

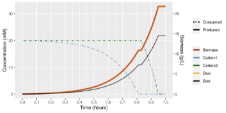
[1] M. W. Covert et al., Journal of theoretical biology, 2001

Results validation with respect to the hybrid problem

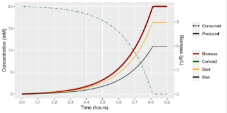
Results are validated by reproducing the input simulations



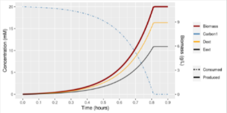
Experiment 1



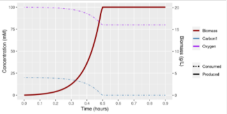
Experiment 2



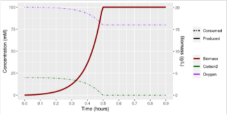
Experiment 3



Experiment 4



Experiment 5



Experiment 6

Subset minimal model allows retrieving the input simulations

Solution to **Hybrid and Combinatorial** problems

Conclusion: inferring regulatory rules from metabolic traces

Advantages:

1. Relaxation of the inferring problem as a combinatorial problem
2. Scale to bigger instances (ex: full Covert's model)
3. Correctly infer ground truth models
   - Find smaller models explaining the input data

Disadvantages:

1. Boolean d-rFBA leads to false negatives/positives results
2. Boolean objective functions are manually defined
   - **Futur works:** explore hybrid solving frameworks as SMT solvers[1]
3. Enzymatic and proteins costs are not considered
   - **Future works:** rely on regulatory dynamic enzyme-cost FBA framework[2]

[1] R. Kaminski et al., arXiv, 2020

[2] L. Liu et al., Journal of Theoretical Biology, 2020

References I

Baral, C. (2003).
Knowledge representation, reasoning and declarative problem solving.
Cambridge university press.

Chevalier, S., Froidevaux, C., Paulevé, L., and Zinovyev, A. (2019).
Synthesis of Boolean Networks from Biological Dynamical Constraints using Answer-Set Programming.
In 2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI), ICTAI, Portland, Oregon, United States. IEEE.

Covert, M. W., Knight, E. M., Reed, J. L., Herrgard, M. J., and Palsson, B. O. (2004).
Integrating high-throughput and computational data elucidates bacterial networks.
Nature, 429(6987):92–96.

Covert, M. W. and Palsson, B. O. (2002).
Transcriptional regulation in constraints-based metabolic models of escherichia coli *210.
Journal of Biological Chemistry, 277(31):28058–28064.

Covert, M. W., Schilling, C. H., and Palsson, B. (2001).
Regulation of gene expression in flux balance models of metabolism.
J Theor Biol, 213(1):73–88.

Eiter, T. and Gottlob, G. (1995).
On the computational cost of disjunctive logic programming: Propositional case.
Annals of Mathematics and Artificial Intelligence, 15(3-4):289–323.

Gebser, M., Kaminski, R., and Schaub, T. (2011).
Complex optimization in answer set programming.
Theory and Practice of Logic Programming, 11(4-5):821–839.

References II

Kaminski, R., Romero, J., Schaub, T., and Wanko, P. (2020).
How to build your own asp-based system?!
arXiv preprint arXiv:2008.06692.

Liu, L. and Bockmayr, A. (2020).
Regulatory dynamic enzyme-cost flux balance analysis: A unifying framework for constraint-based modeling.
J Theor Biol, 501:110317.

Marmiesse, L., Peyraud, R., and Cottret, L. (2015).
Flexflux: combining metabolic flux and regulatory network analyses.
BMC systems biology, 9(1):1–13.

Ostrowski, M., Paulevé, L., Schaub, T., Siegel, A., and Guziolowski, C. (2016).
Boolean network identification from perturbation time series data combining dynamics abstraction and logic programming.
Biosystems, 149:139–153.

Paulevé, L., Kolčák, J., Chatain, T., and Haar, S. (2020).
Reconciling qualitative, abstract, and scalable modeling of biological networks.
Nature communications, 11(1):1–7.

Videla, S., Guziolowski, C., Eduati, F., Thiele, S., Gebser, M., Nicolas, J., Saez-Rodriguez, J., Schaub, T., and Siegel, A. (2015).
Learning boolean logic models of signaling networks with asp.
Theoretical Computer Science, 599:79–101.