BioinformaticsDay@DAIS, Ca' Foscari University; July 07, 2016

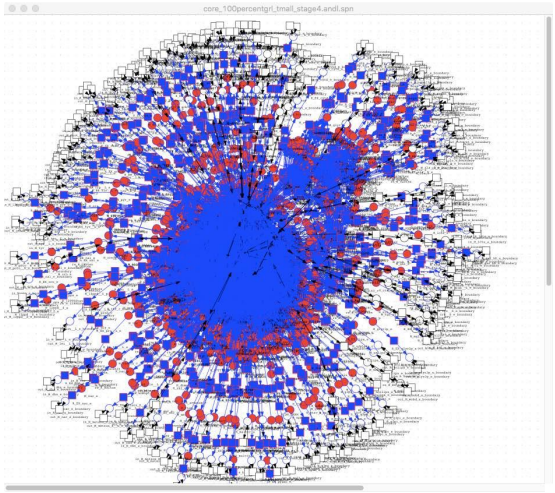
Model-driven design for Synthetic Systems Biology

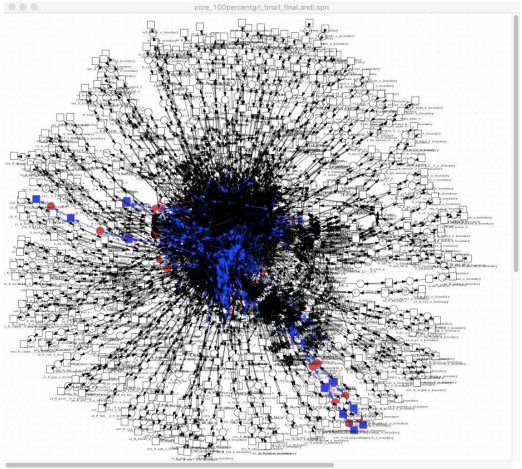
Monika Heiner[1,2] & David Gilbert[2]

[1] Brandenburg Technical University (BTU), Cottbus, Germany
[2] Brunel University London, UK,
Synthetic Biology Theme & Department of Computer Science

Outline

- Brunel University London:
  bacterial engineering / Synthetic Biology
- Whole genome metabolic models
  - engineering design templates
- Need for ‘correct’ initial template description
  - well behaved (dynamic behaviour)
  - based on (badly behaved) public domain models
- Structure based correction of initial models
  - graph analysis, graph editing,
  - dynamic simulation
  - model checking





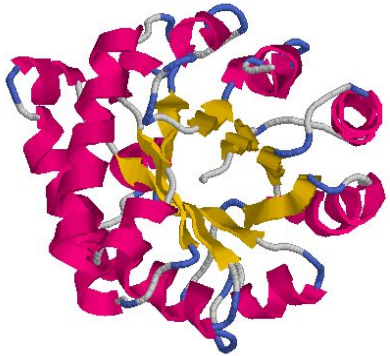
From Genes to Systems

DNA

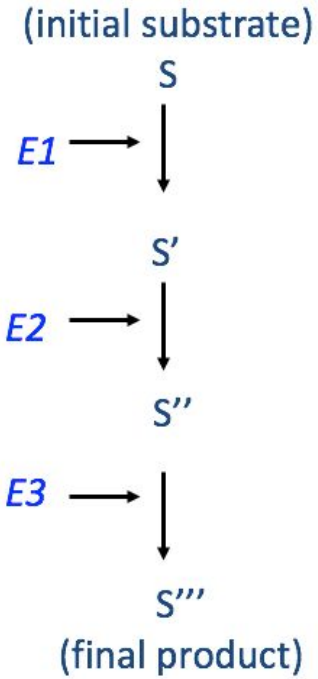
"gene"

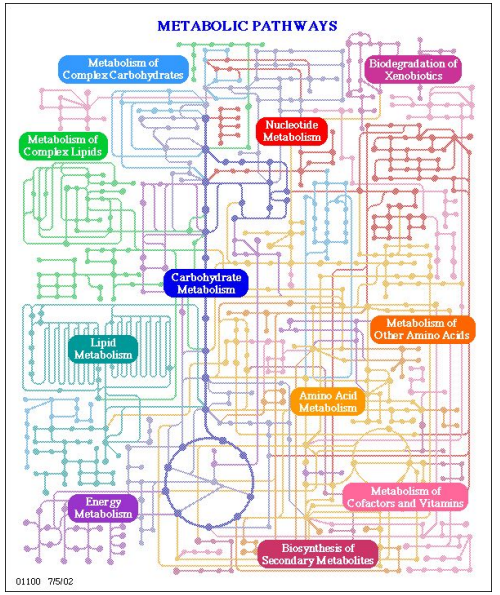
mRNA

Protein sequence

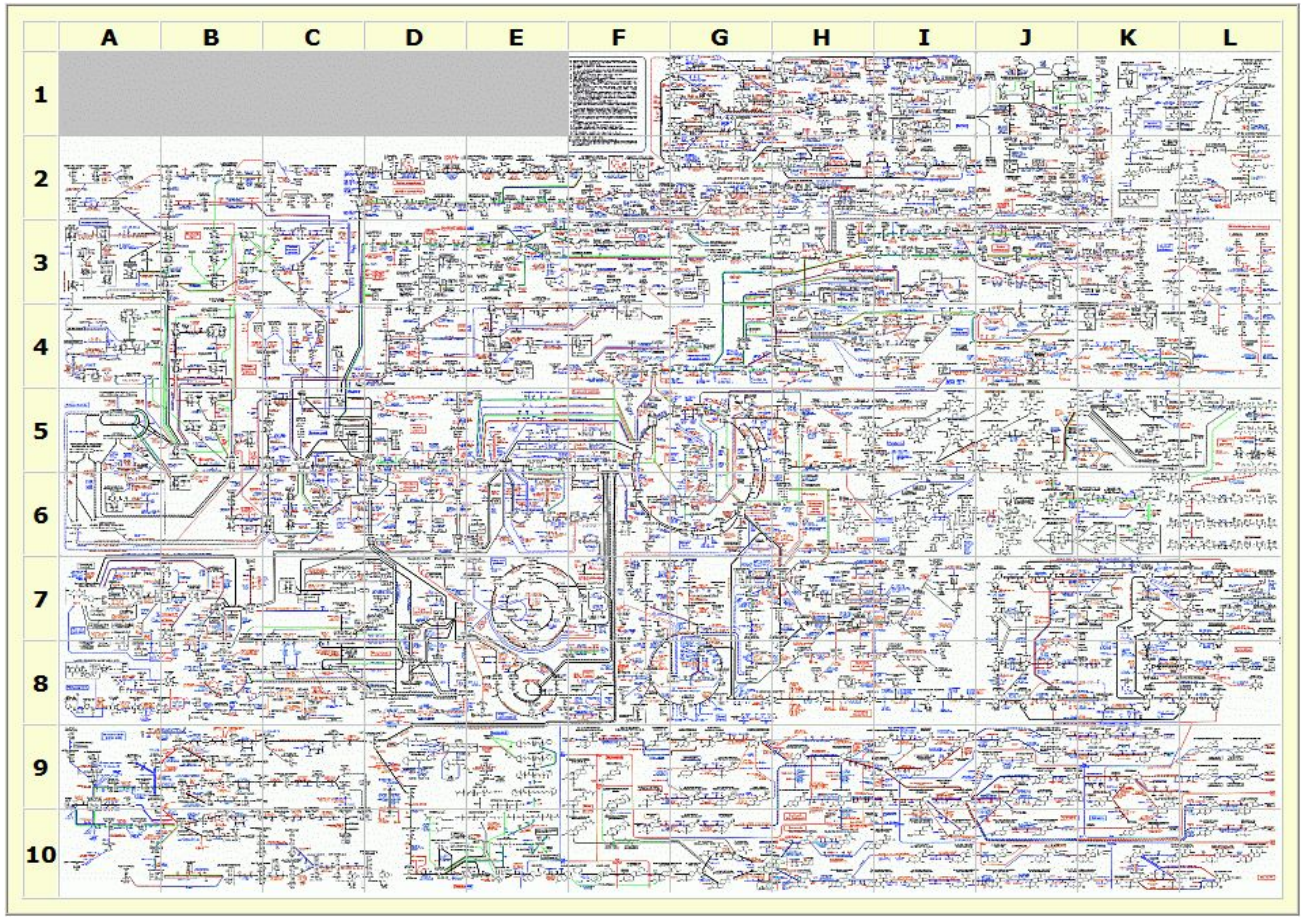


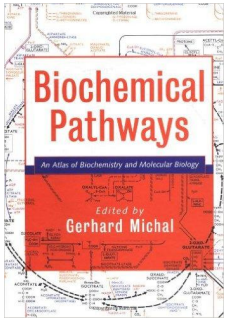
Folded Protein





Metabolic Pathways



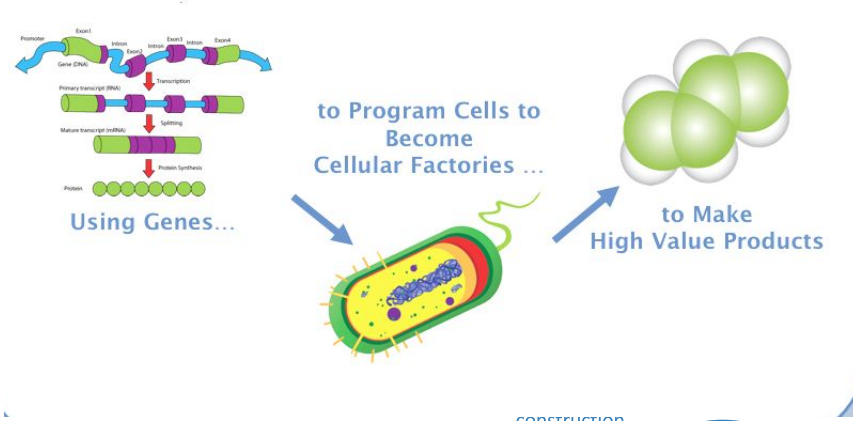


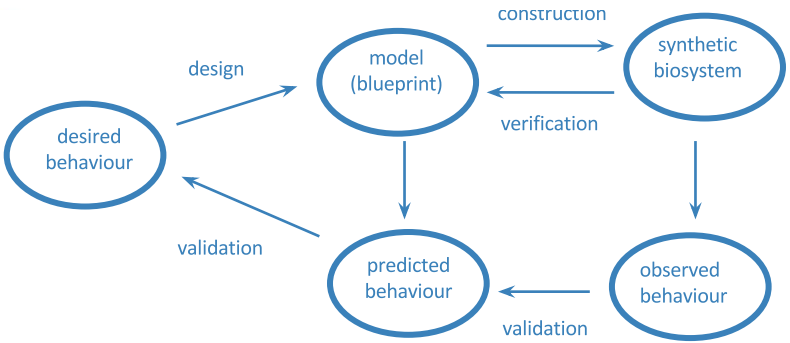
http://ca.expasy.org/tools/pathways/

Synthetic Biology / Bacterial Engineering

- **modify or make a new one**
  - system, or
  - product

- **Synthetic Biology**
  - the structured engineering of biological systems for useful purposes

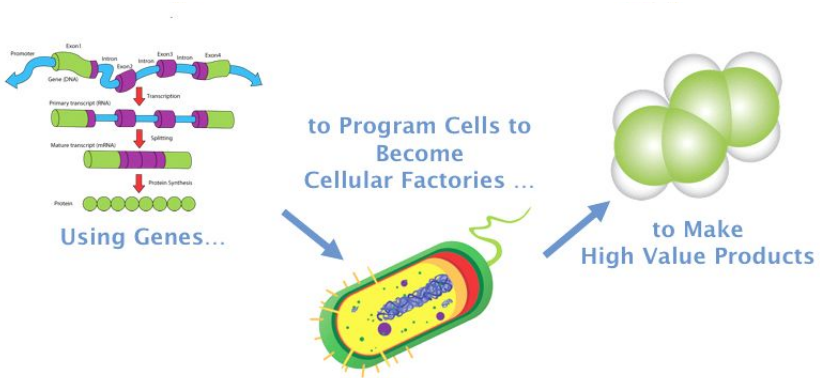




Bacterial Engineering

bacteria can be engineered to act as little factories for

- energy production
- drug production
- immune system booster(probiotics)
- pollution clean up
- environmental sensors



bacterial engineering

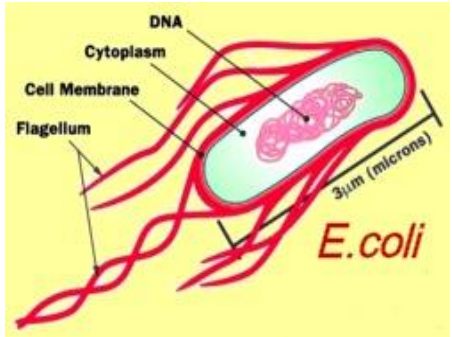
- genetic engineering -> target-driven genome modification
- metabolic engineering -> focus on metabolism

Model Organism: *Escherichia Coli* (*E. coli)*

- 1885 discovered by German-Austrian pediatrician Theodor Escherich
  -> named after him in 1919

- gram-negative, anaerobic, rod-shaped bacterium
  commonly found in lower intestine
  of warm-blooded organisms

- can be grown and cultured easily and inexpensively
  -> takes about 20' to reproduce
  (in favourable conditions)



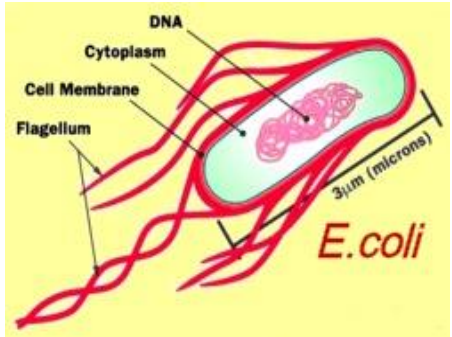
Model Organism: *Escherichia Coli* (*E. coli*)

- 1997 first complete DNA Sequence

- Today:
  Several hundred ‘complete’ genome sequences,
  Each individual genome: 4,000 - 5,500 genes
  (protein genes, RNA genes)

  How many protein genes control metabolism?

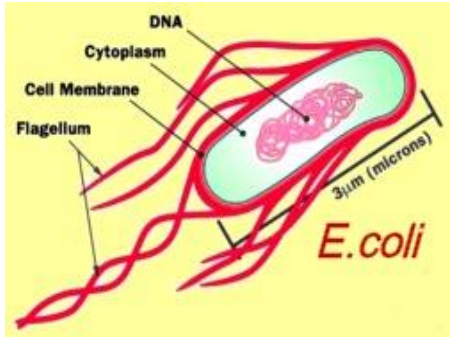
- The most widely studied organism
  -> EcoliWiki
  -> EcoCyc:
  scientific database for E.coli K-12 MG 1655



Model Organism: *Escherichia Coli* (*E. coli*)

- One of the most diverse bacterial species
- **Strain**
  A species' subgroup with unique characteristics that distinguish it from other strains
- > 4k protein coding genes, but
  only 20% of the genome common to all strains

  Compare:
  Genome of all humans differ by about 1%

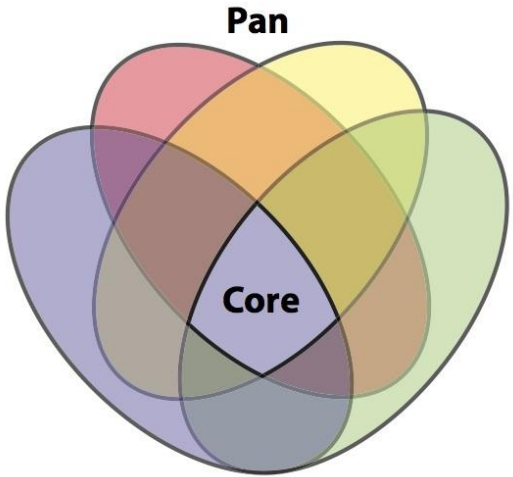


Model Organism: *Escherichia Coli* (*E. coli*)

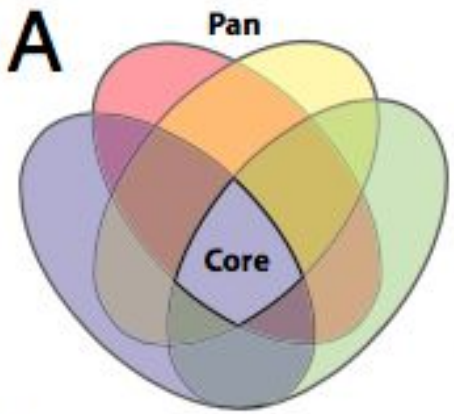
- **Core genome:** 800 - 1,100 genes
  -> genome common to all strains

- **Pangenome:** exceeds 16,000 genes
  -> Total number of different genes
  among all of the sequenced E. coli strains

  Possible explanation:
  Horizontal gene transfer

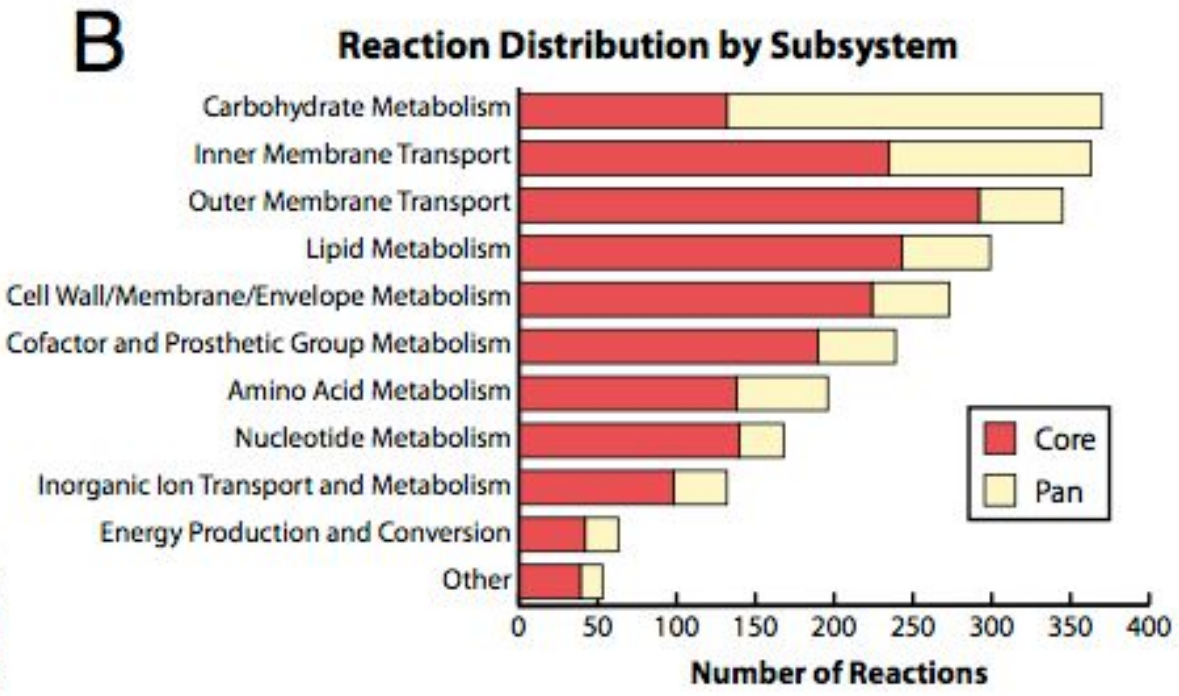


Monk Metabolic Models



For 55 unique *E. coli* strains *)

	Core:	Pan:
Genes:	965	1,460
Reactions:	1,773	2,501
Metabolites:	1,665	2,043



*) 47 E.coli, 8 Shigella

[Monk 2013]

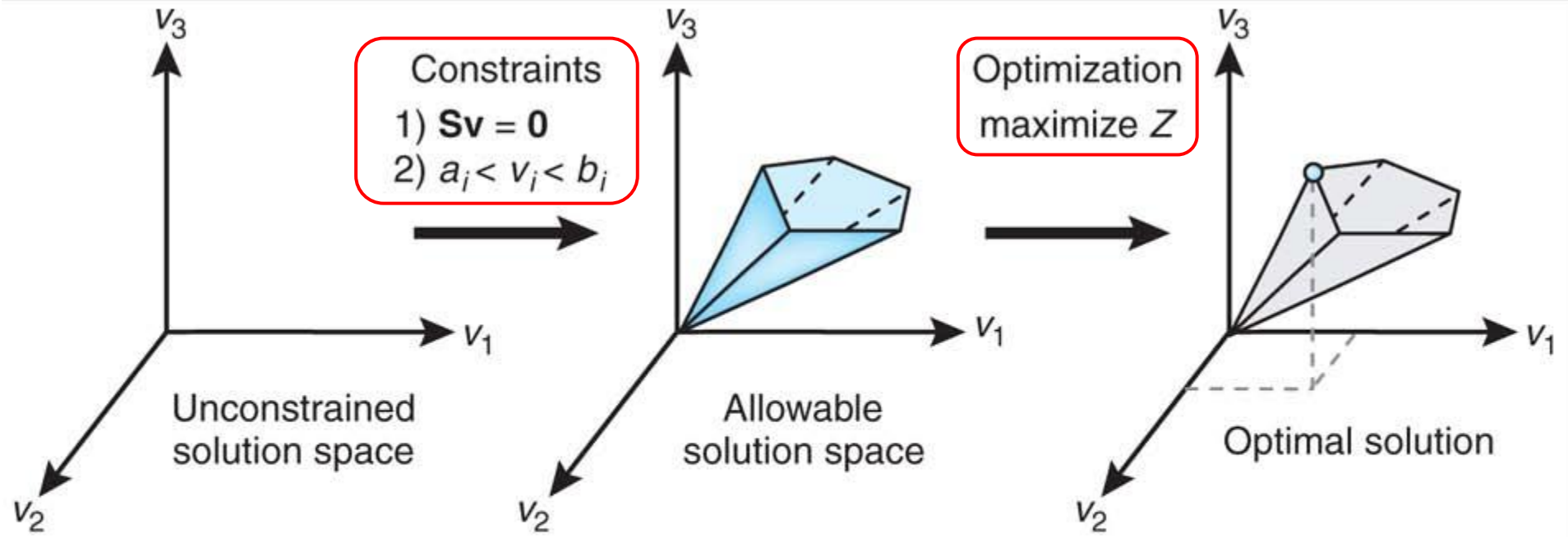
Modelling 4 Metabolic Engineering, State of the Art

- research subject for about 15 years;
  two categories of models
  - Static (structural) models (no kinetic info required) -> fast majority
  - Dynamic (kinetic) models -> computational models

- Standard graph algorithms
  - Eg, linear path from input A to output B,
    avoiding or passing specific intermediates C

- Linear programming techniques + steady state assumption
  - All minimal flows (elementary modes, T-invariants, …))
  - Flux balance analysis: “all minimal flows + target function”

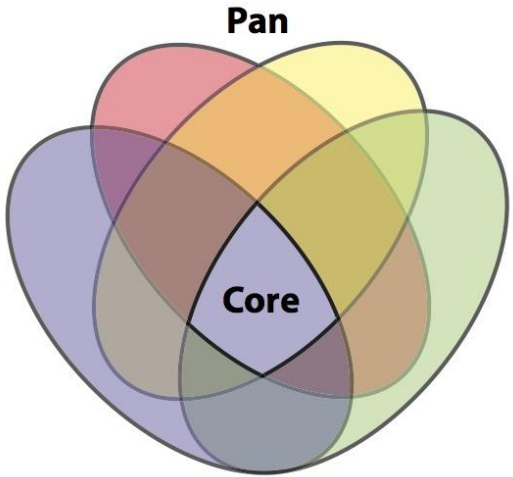
Flux Balance Analysis (FBA)



-> Engineering of single strains

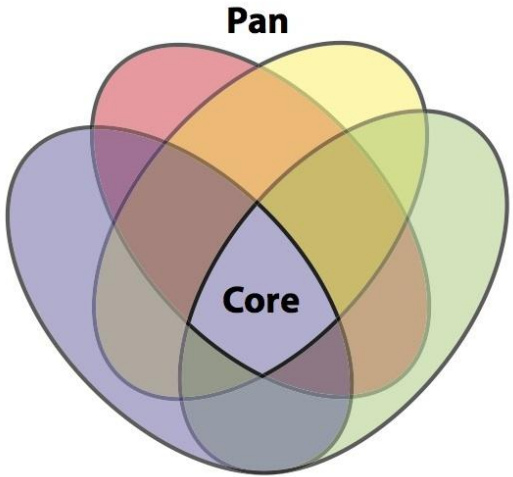
The PROJECT: Design Methods for Bacterial Engineering

- to develop computational techniques
  - dynamic simulation
    -> transient behaviour analysis
  - To deal with sets of models
- to build the Brunel Core Model
  - based on gene set from Nigel Saunder's group



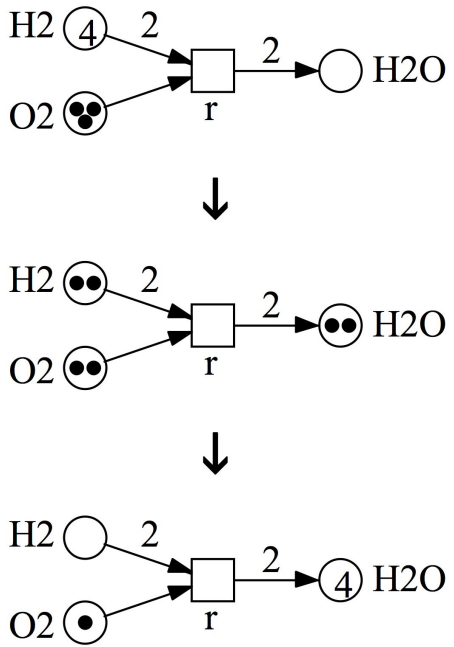
CHALLENGES

- How to generate strain-specific models?
  - Computational metabolic models
  - How to generate models for new strains?
- How to deal with sets of models?
  - To rank according to target behaviour
  - To identify genes crucial for performance
- How to select
  - Chassis strain: target for gene transfer
  - Donor strains: source of gene transfer



Biological Models

- reaction/metabolite graphs
  - bipartite graphs → Petri nets
- stoichiometry / arc weights
- no kinetic rates given
  - assume mass action, kinetic parameter=1
- boundary conditions
- model structure
  - cytoplasm, periplasm, external, boundary
- SBML (Systems Biology Markup Language)
  - → Petri nets



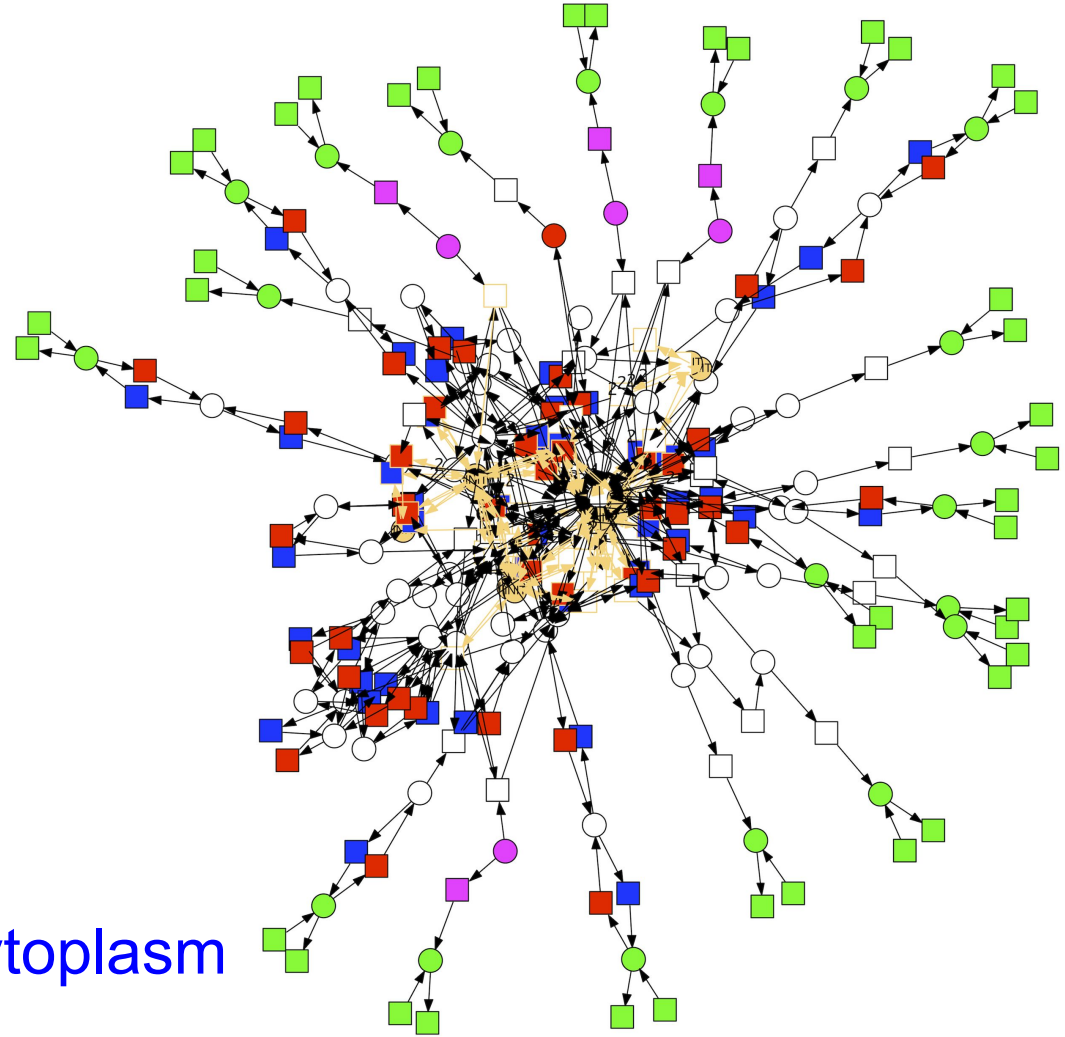
$2\ H_2 + O_2 \rightarrow 2\ H_2O$

Example
E. coli core

[Orth 2010]

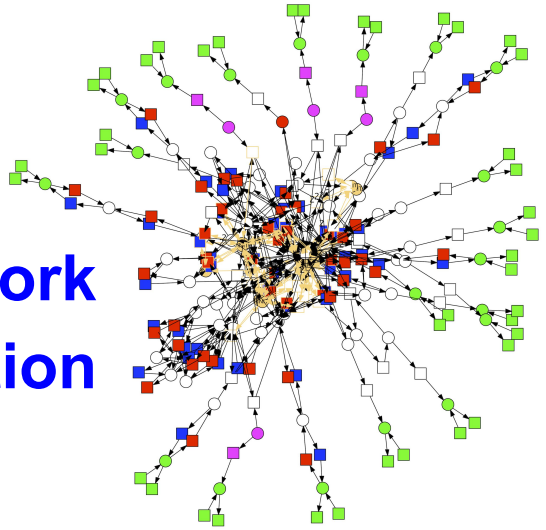
model structure:

- cytoplasm,
- periplasm,
- external,
- boundary



in/out flow through cytoplasm

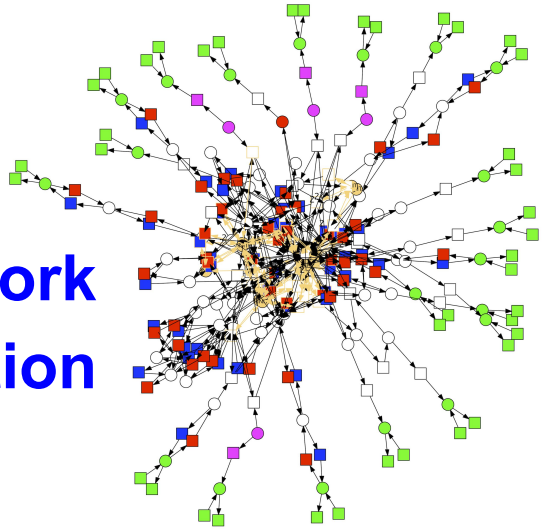
Assumption



We postulate that a **‘good’ metabolic network** is one in which **every metabolite and reaction** is (at least)

- weakly live (i.e. exhibits dynamic behaviour) at some point, and
- has a non-zero steady state.

Assumption

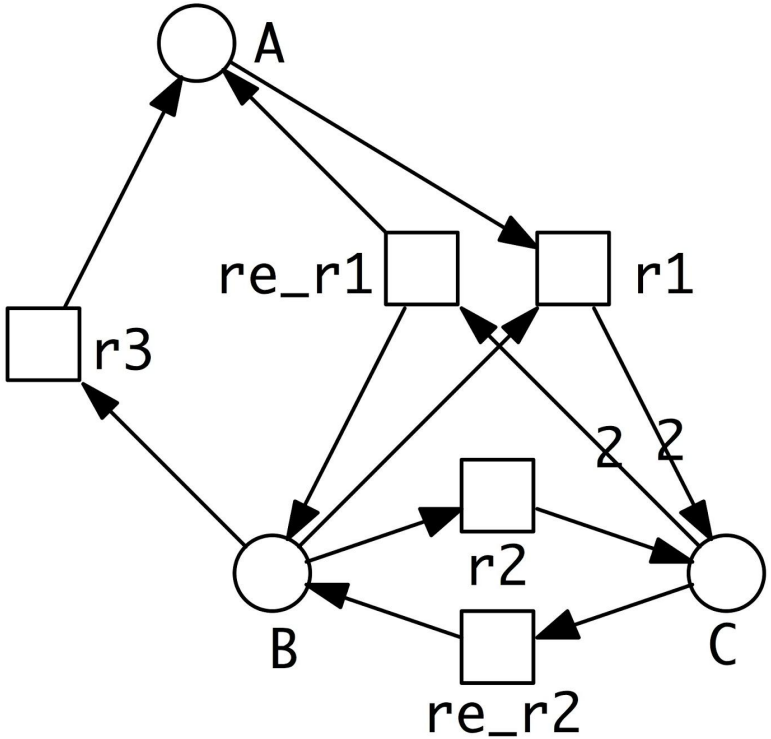


We postulate that a **‘good’ metabolic network** is one in which **every metabolite and reaction** is (at least)

- weakly live (i.e. exhibits dynamic behaviour) at some point, and
- has a non-zero steady state.

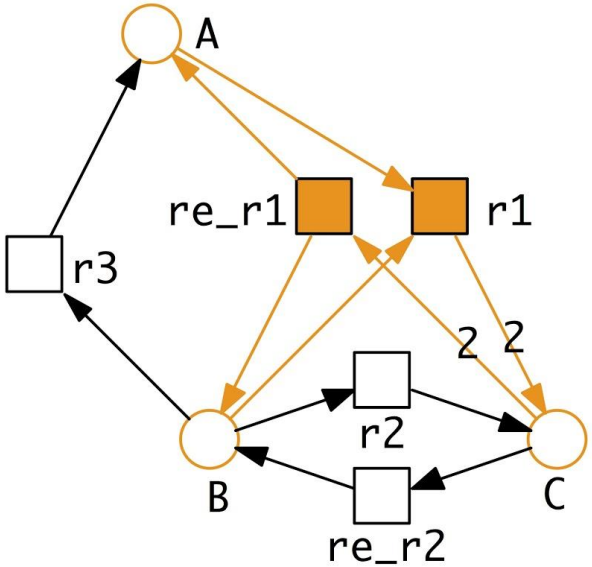
SOUNDS EASY, BUT ISN’T, BECAUSE . . .

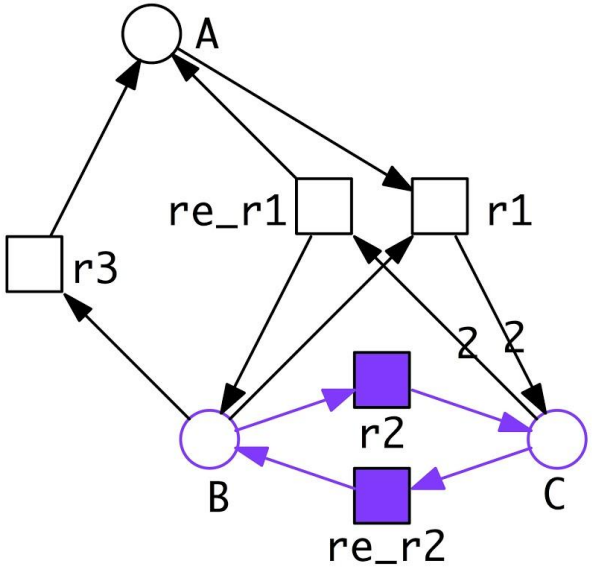
Example

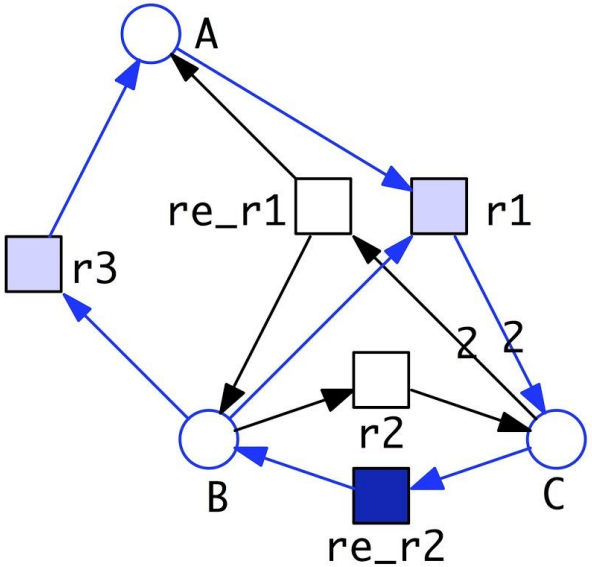



```
A+B <-> 2C
B   <-> C
B    -> A
```

Example - T-invariants

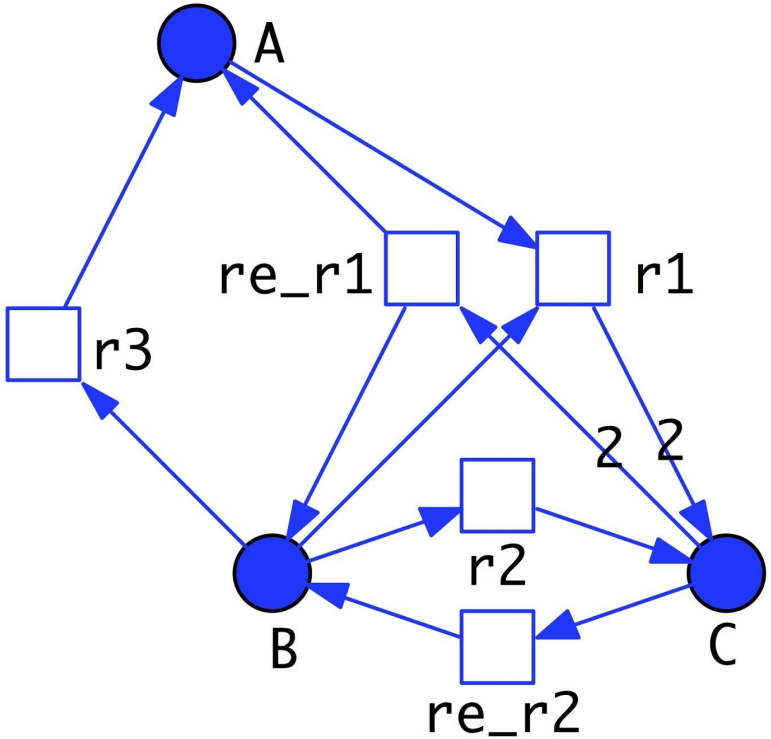






-> covered with T-invariants (CTI)

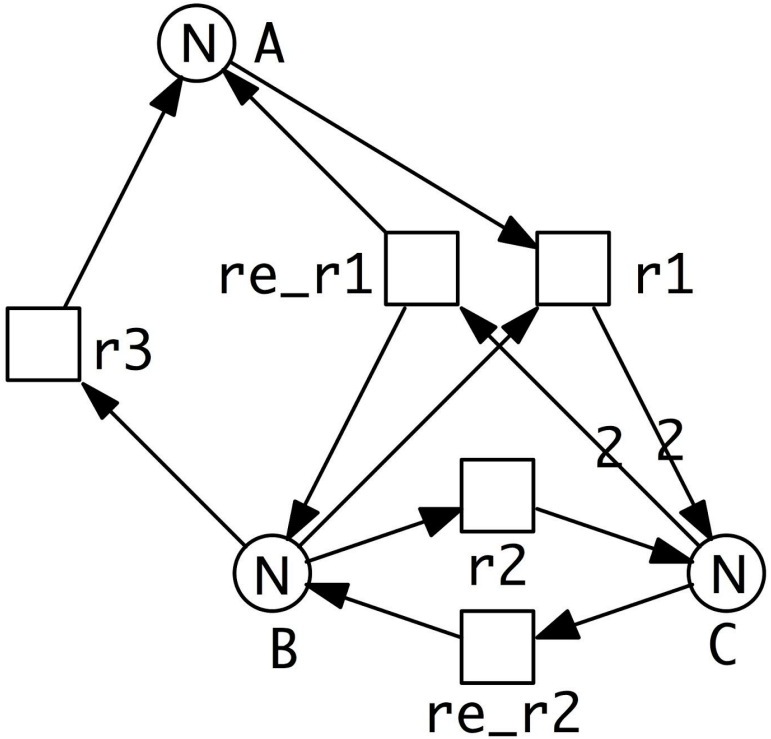
Example - P-invariants




```
A+B <-> 2C
B   <-> C
B    -> A
```

-> covered with P-invariants (CPI)

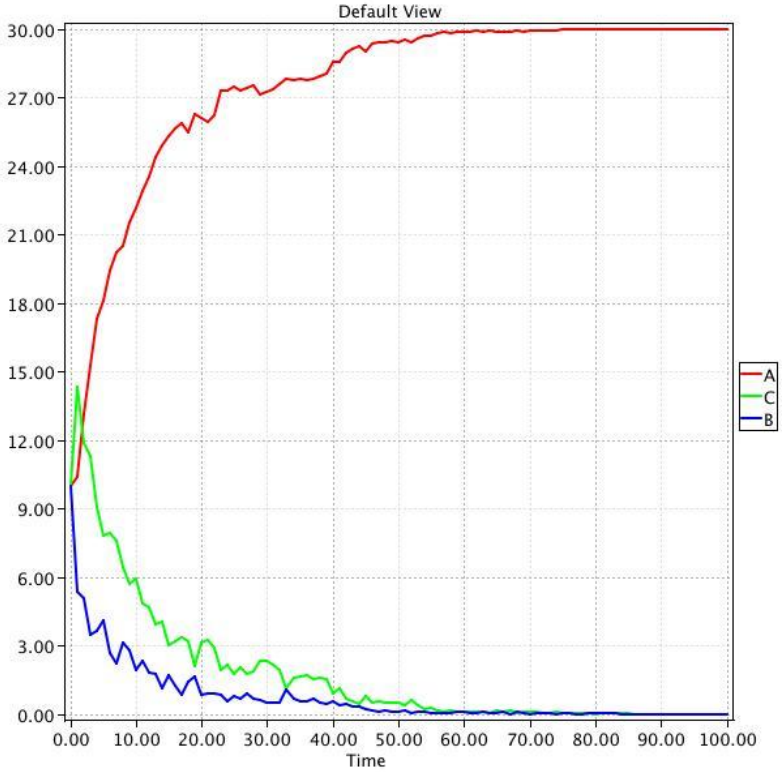
Example - Initialisation

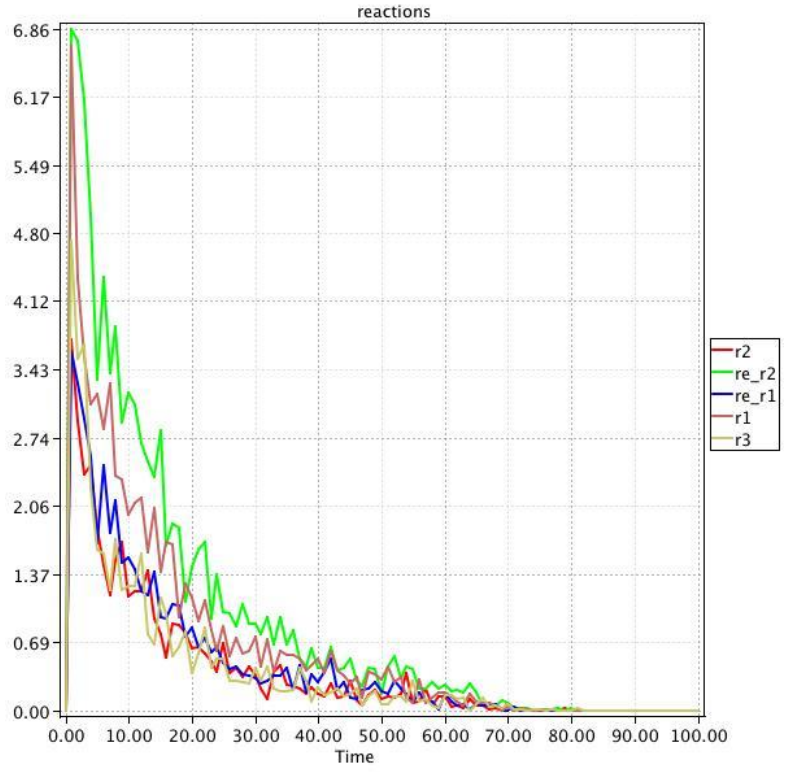



```
A+B <-> 2C
B   <-> C
B    -> A
```

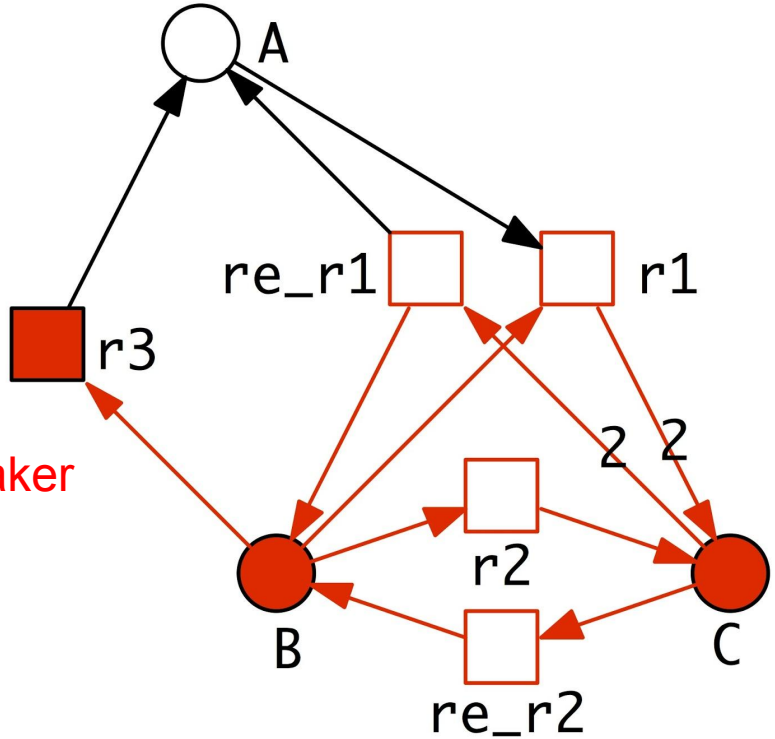
Const N = 1, 5, 10, 50, 100, . . .

Example - Simulation Results (N=10)





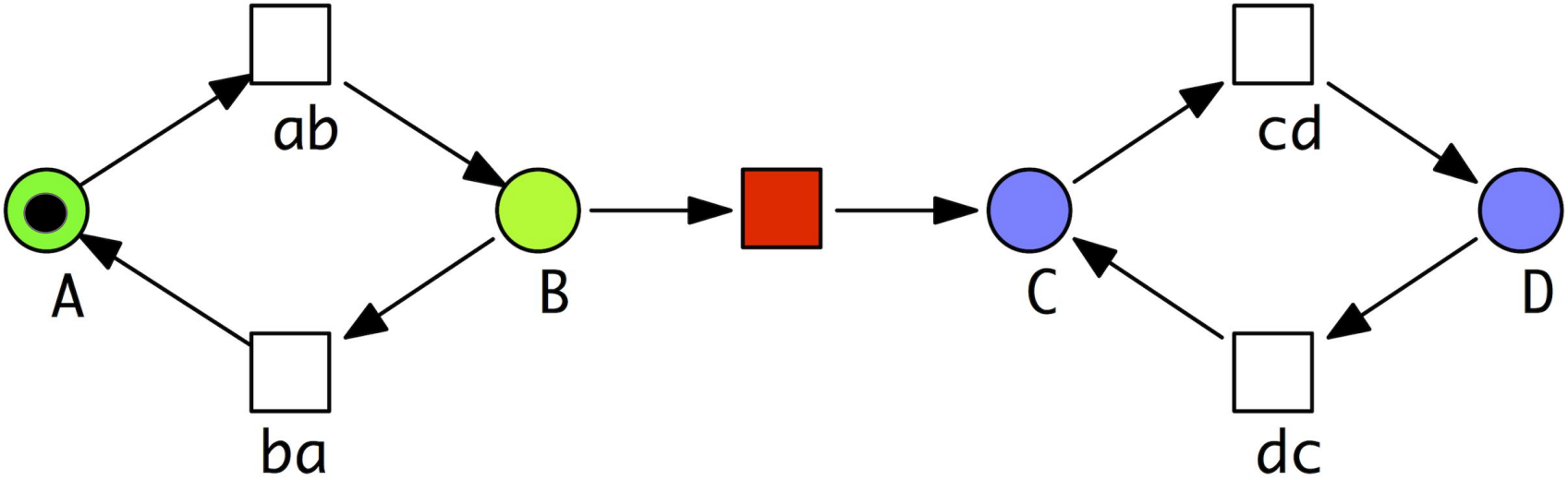
Example - Bad Siphon



troublemaker

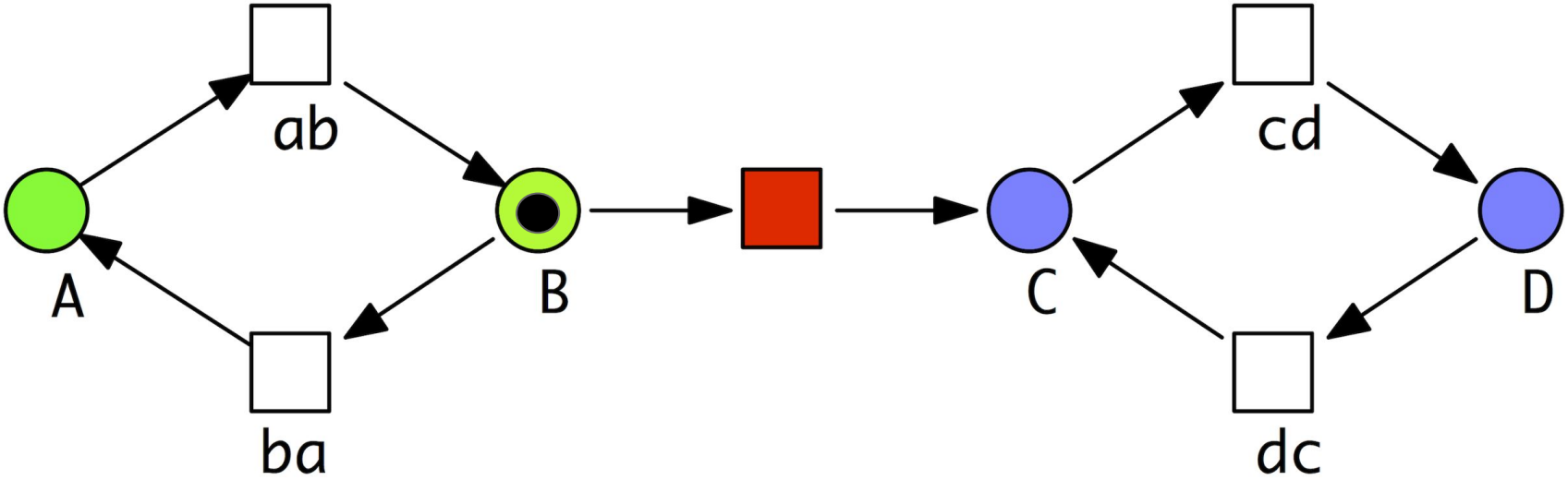
```
A+B <-> 2C
B   <-> C
B    -> A
```

Simple Siphon / Trap



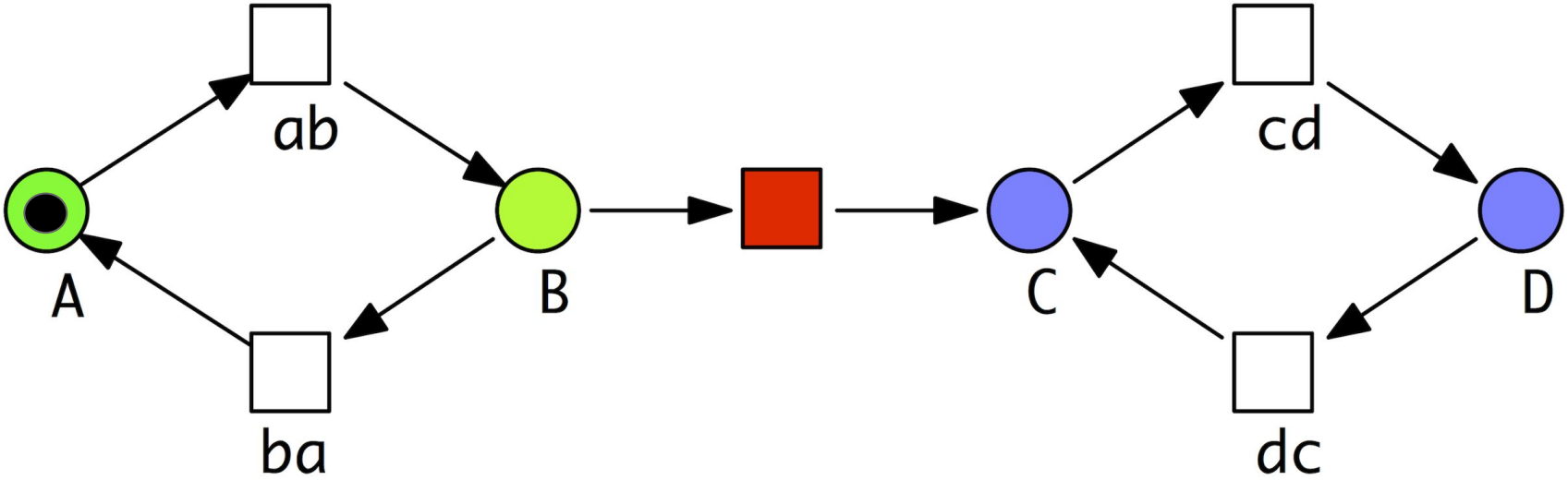
siphon places troublemaker transition trap places

Simple Siphon / Trap



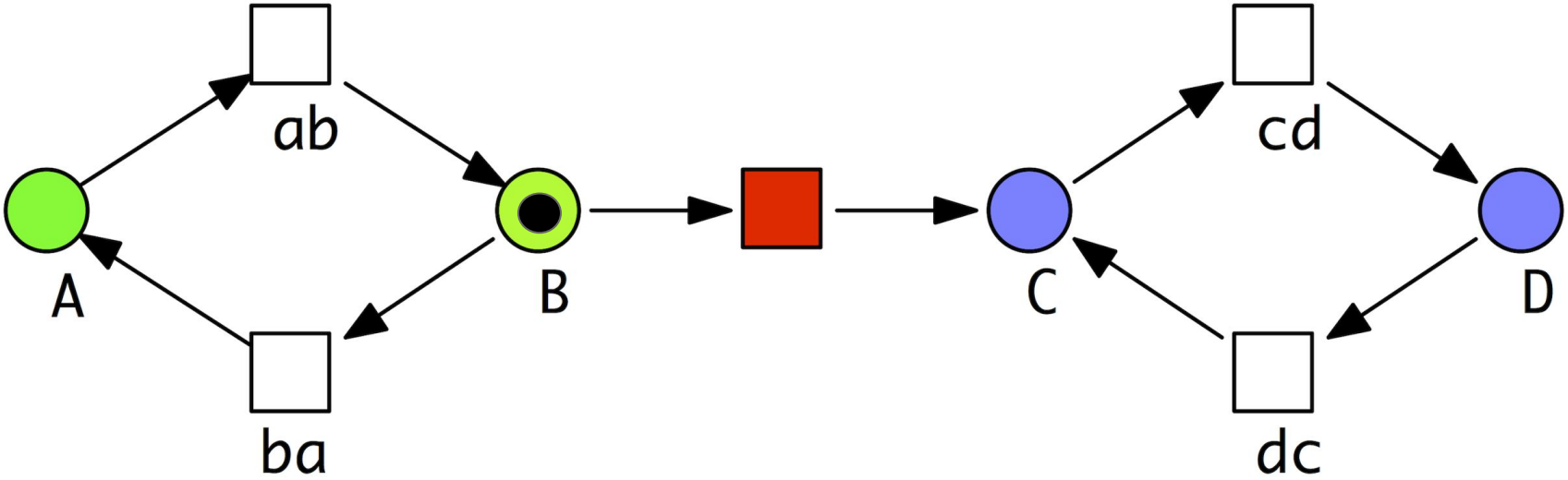
siphon places **troublemaker transition** **trap places**

Simple Siphon / Trap



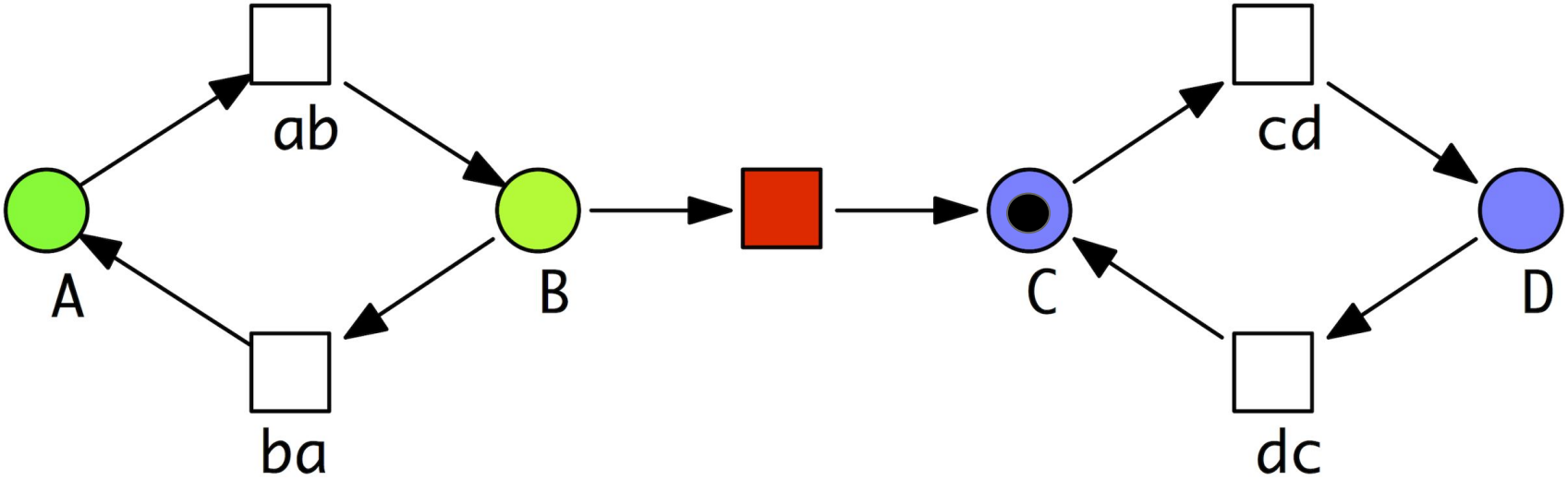
siphon places **troublemaker transition** **trap places**

Simple Siphon / Trap



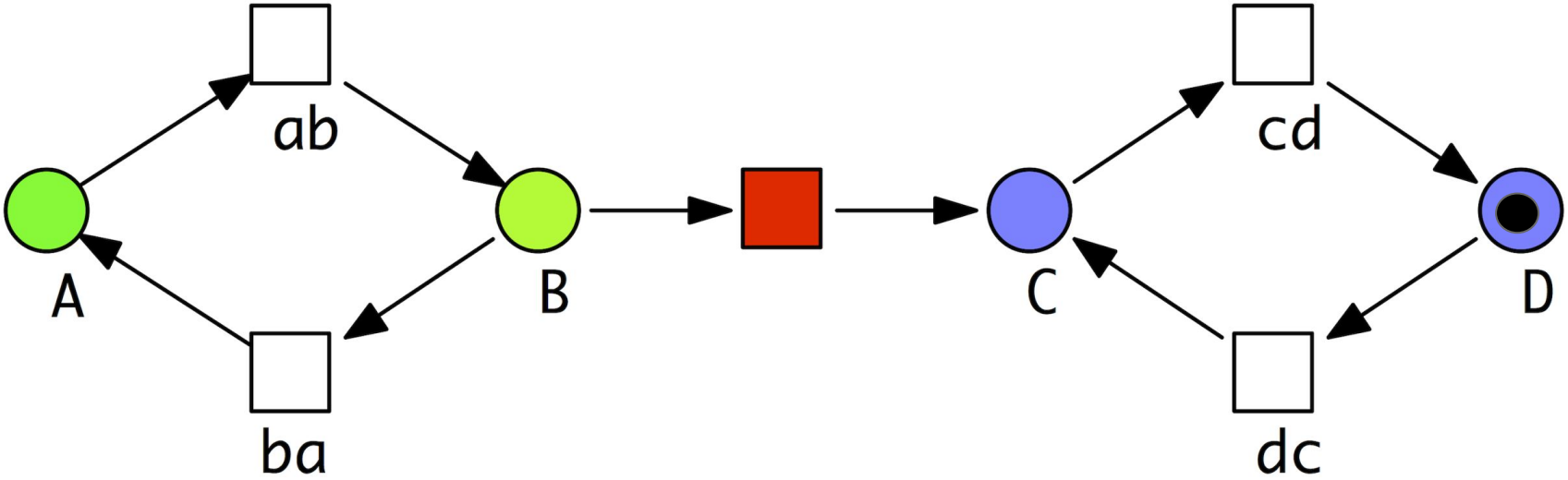
siphon places troublemaker transition trap places

Simple Siphon / Trap



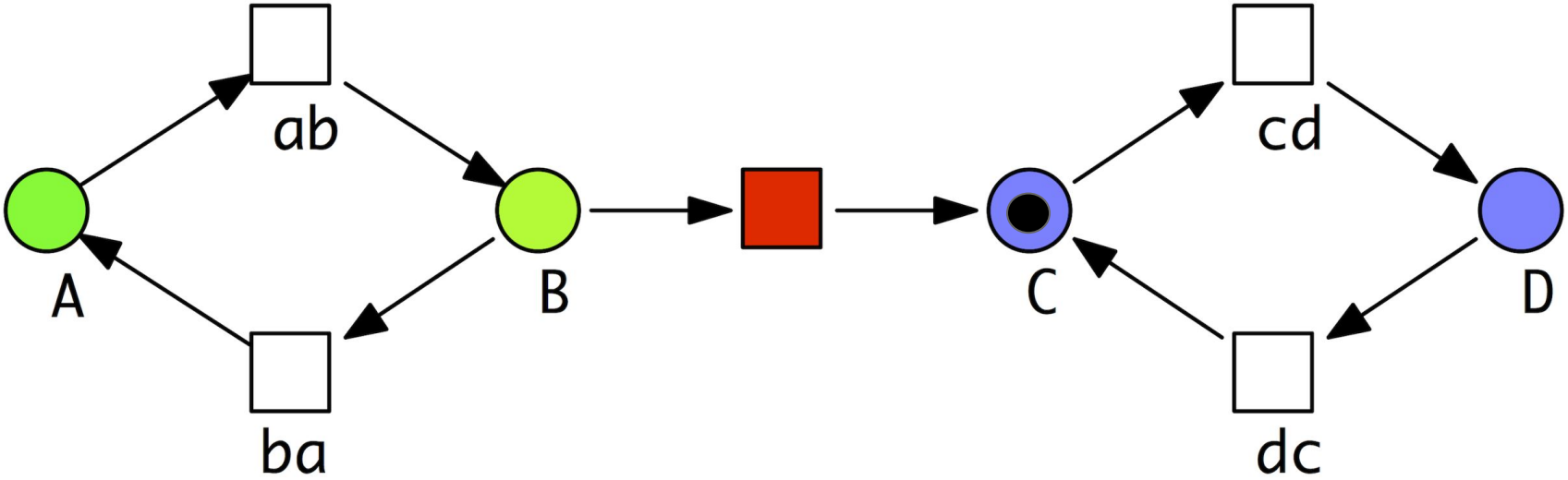
siphon places troublemaker transition trap places

Simple Siphon / Trap



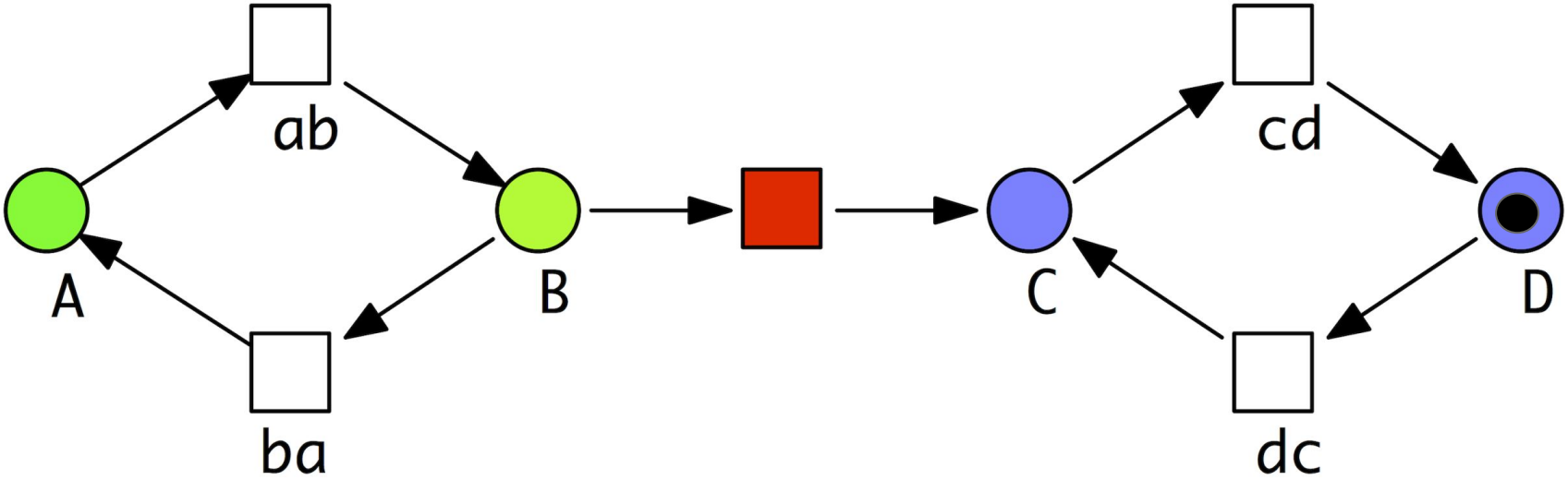
siphon places **troublemaker transition** **trap places**

Simple Siphon / Trap



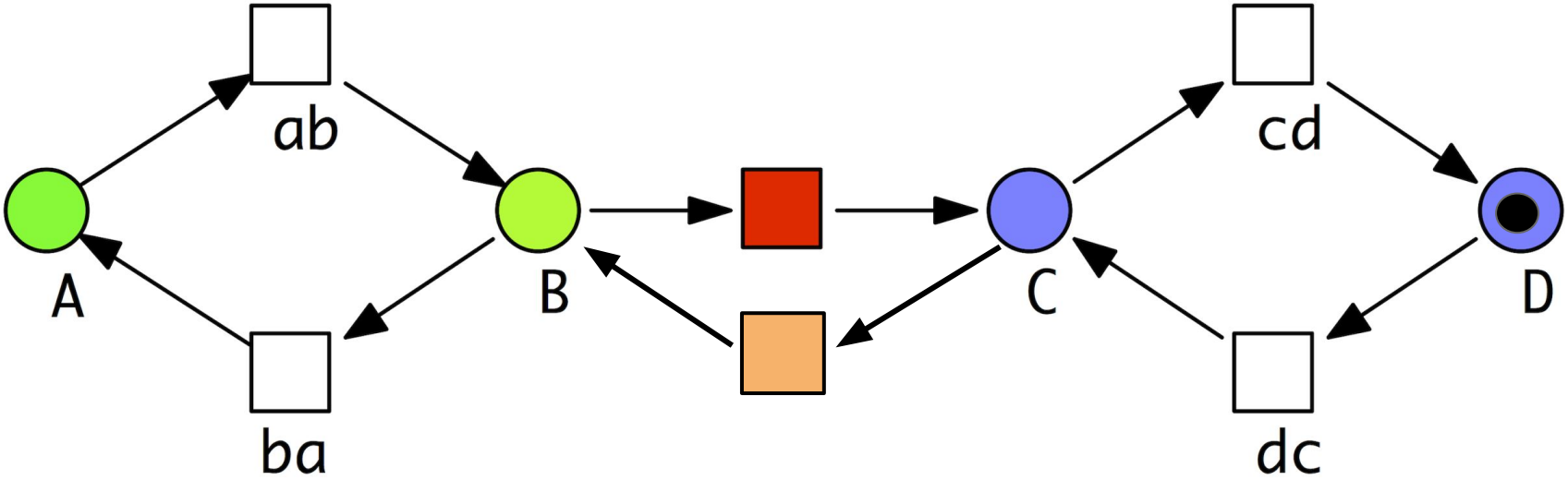
siphon places **troublemaker transition** **trap places**

Simple Siphon / Trap



siphon places troublemaker transition trap places

Simple Siphon / Trap

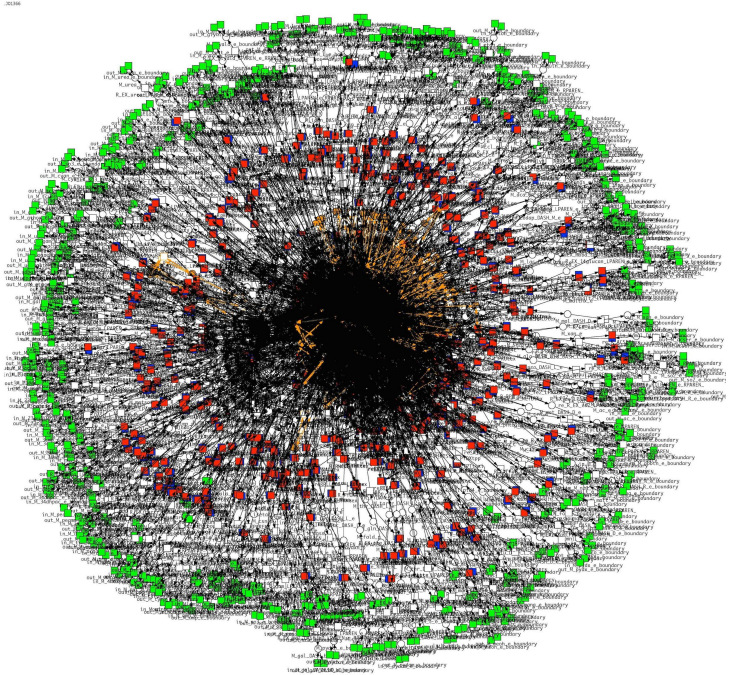


siphon places troublemaker transition trap places

repair transition

Computational Challenges (1)

- large size models
- example sizes
  - reactions > 4k
  - metabolites > 2k
  - connected by > 13k arcs
  - metabolite connectivity: 2-1200

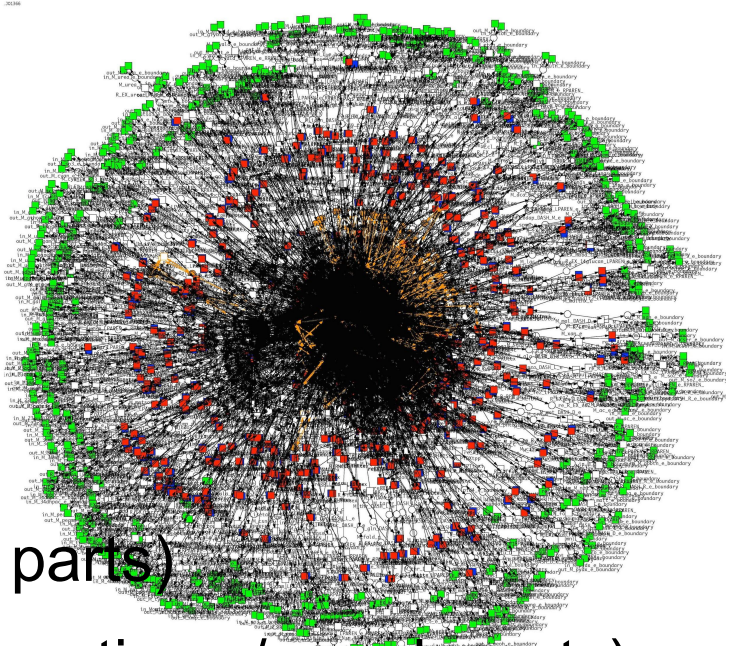


→ cannot perform visual analysis

→ need for automated tools for analysis & correction

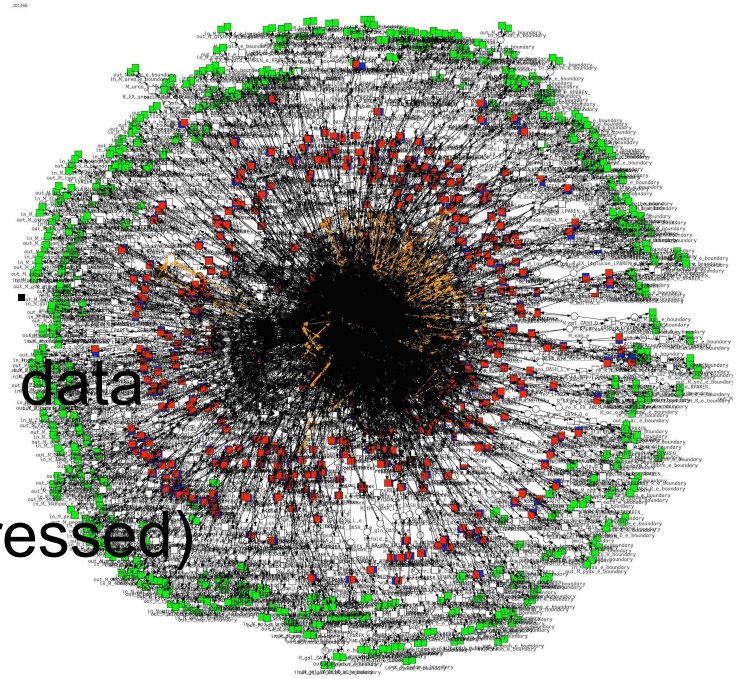
Computational Challenges (2)

- models constructed manually
  → **possibility of ‘errors’**
  - typos
  - wrong directions
  - missing information
    (reactions & metabolites / graph parts)
  - incorrect information (incorrect reactions / graph parts)
  - incorrect composition of parts (reactions) . . .



Computational Challenges (3)

- graph size & structure
  → computational complexity of
  structural and dynamic analysis, . . .
- large size of secondary (generated) data
  → simulation traces
  (30MB uncompressed/12MB compressed)
- design alternatives
  → generation of (very) many models (thousands) . . .

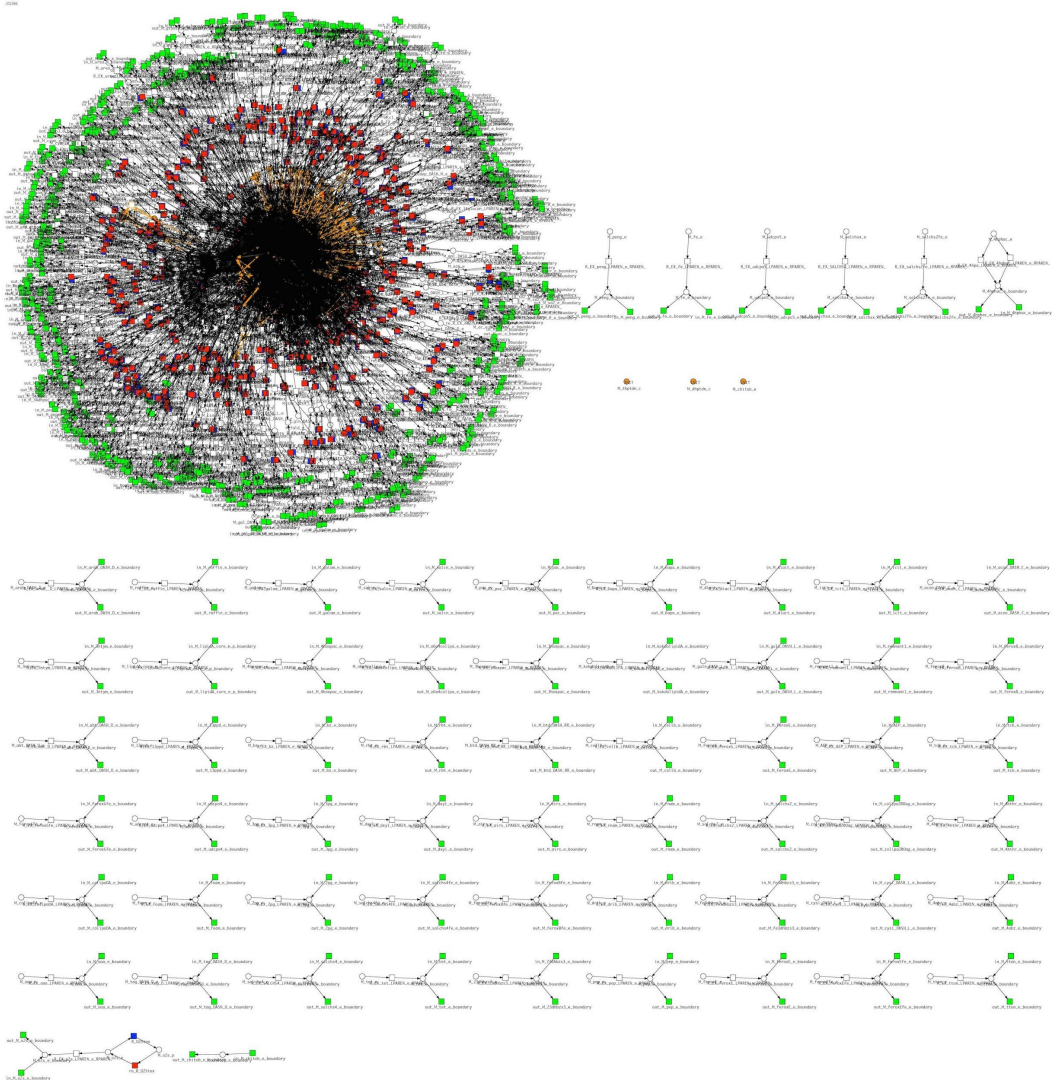


E.coli K-12, MG1655
Whole genome metabolic model

1367 genes
2123 enzymes
2257 metabolites
2645 reactions

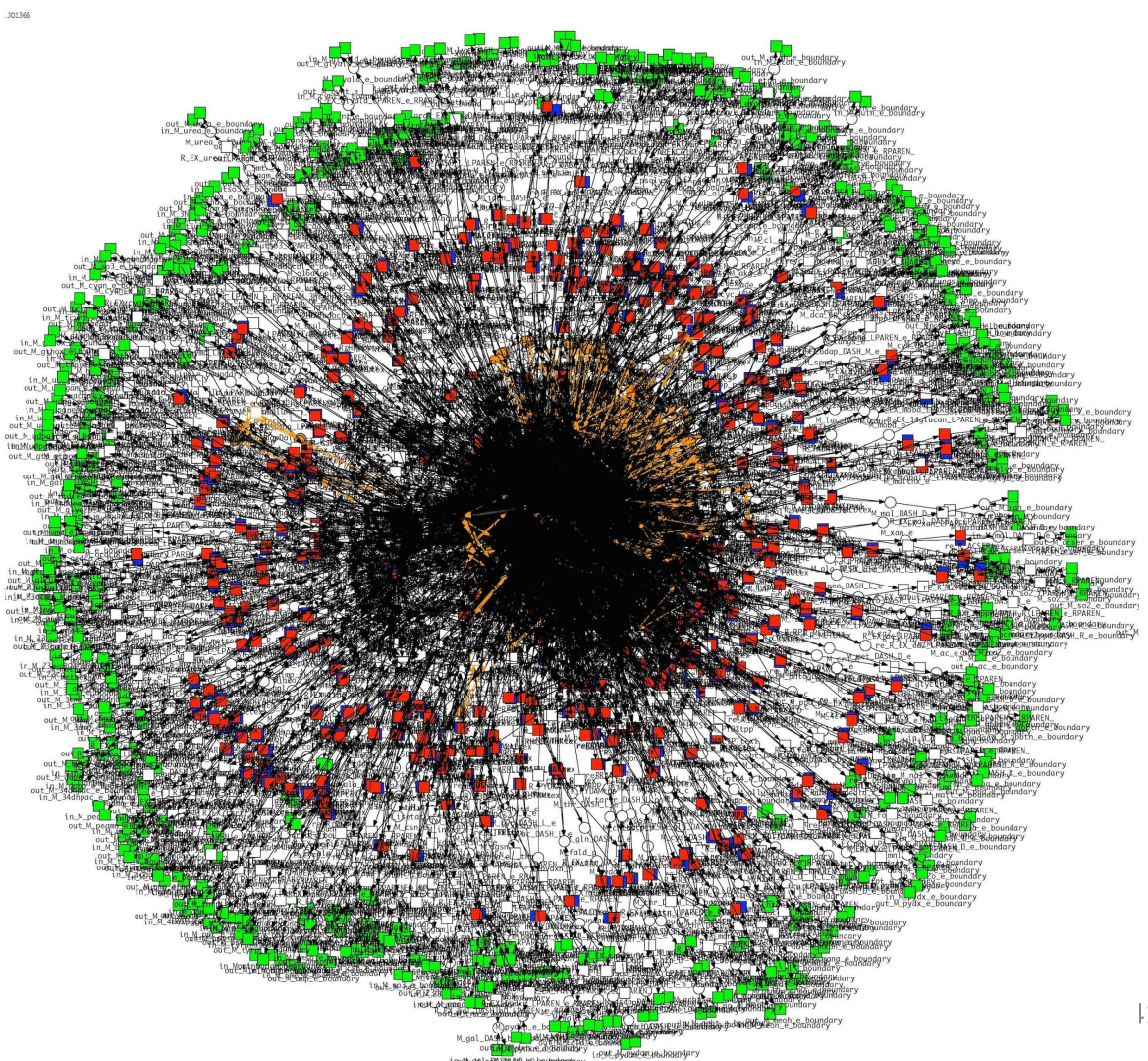
522 spontaneous reactions
  11 switched-off reactions
636 reversible reactions
391 boundary conditions

2257 places
4052 transitions

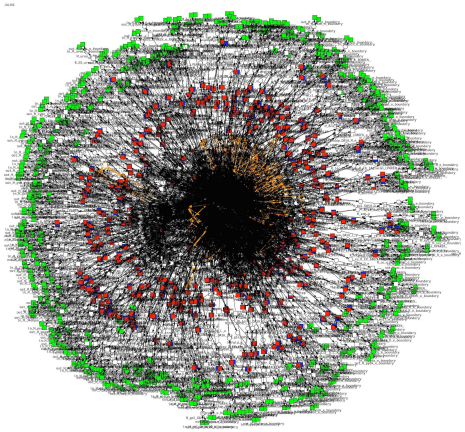


So Big !

We can’t repair this by hand . . .



Techniques & Tools



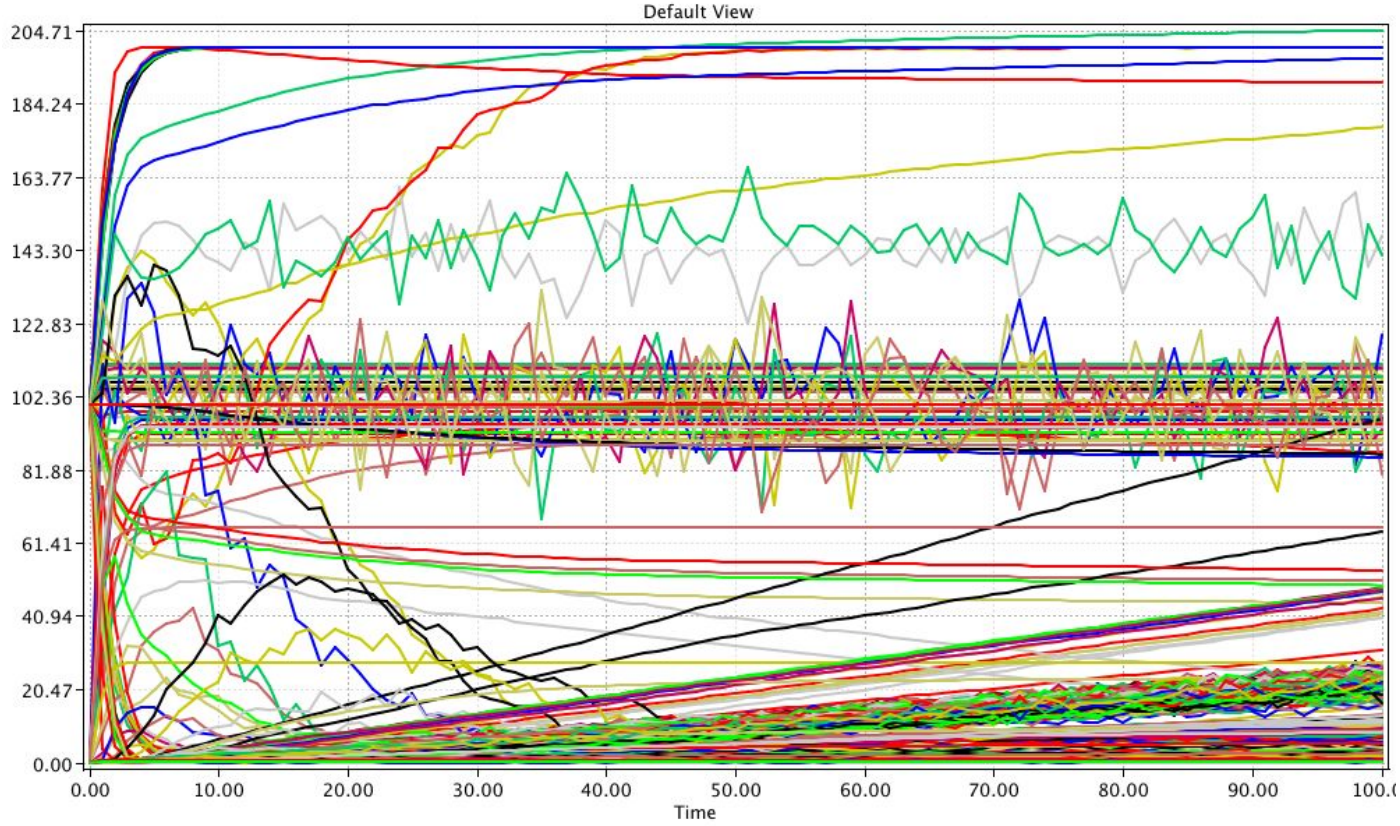
- Visualisation & manual editing - ***Snoopy***
- Structural analysis
  - ***Charlie***
  - ganalysis - ***gprolog*** (170 predicates / 210 lines)
  - ***LoLA*** (SAT checker Minisat)
- Automated graph editing
  - ‘the protocol’ - ***gprolog*** (2k predicates / 2.3k lines), ***LoLA*** & ***Charlie***
- Simulation
  - ***Snoopy*** (delta leaping; stochastic, continuous)
  - ***Marcie*** (delta leaping; stochastic)
- Model checking
  - ***MC2***
  - ***Marcie***

The Workflow

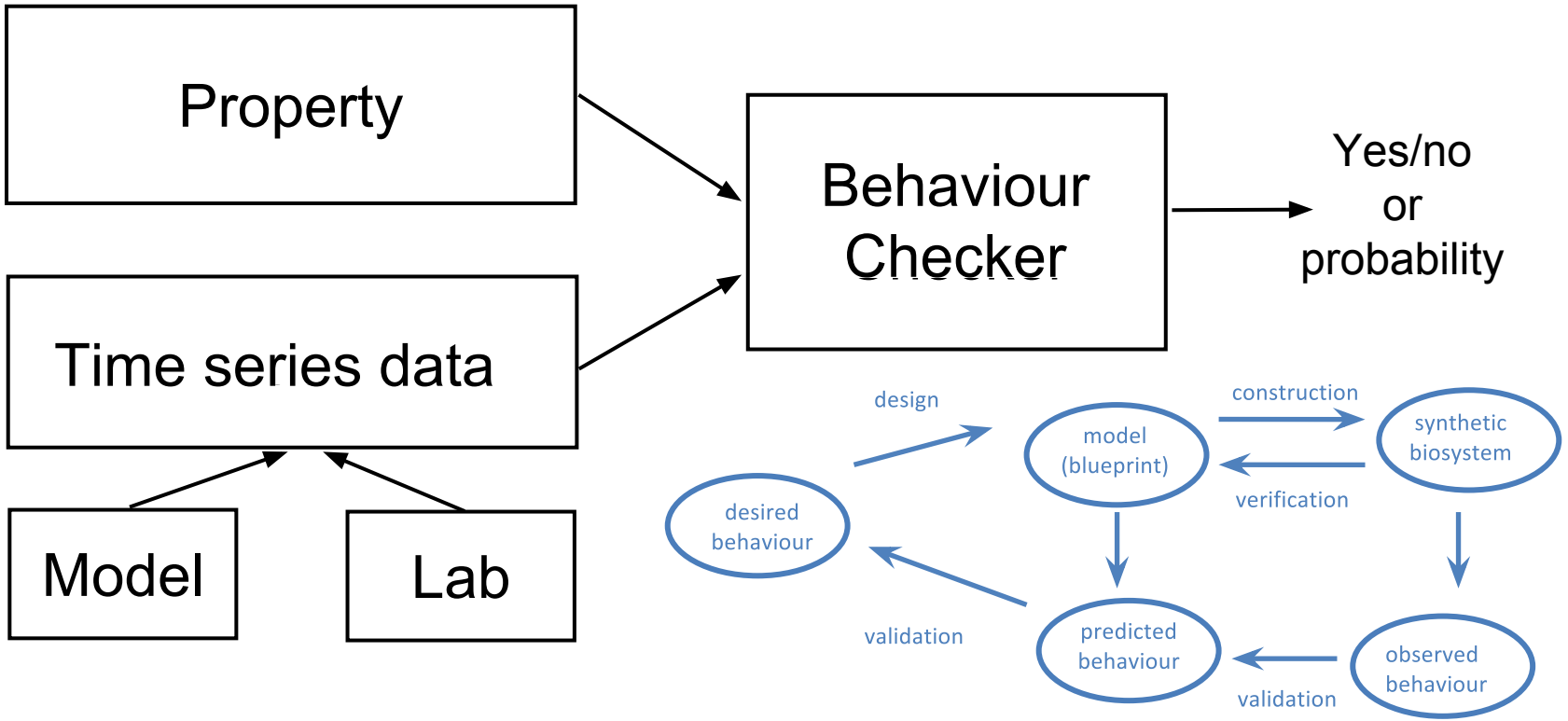
initial model (SBML) → ….. → repaired model

- SBML → Petri net (Snoopy)
  - add boundary reactions (in/out flow) for all boundary conditions
  - reversible reactions → 2*1-way reactions
  - export to graph format (andl)
- Initialise initial model (P-invariants), simulate & analyse
- Automated model repair
- Initialise final model (P-invariants), simulate & analyse
- Compare initial & final models' behaviour

Time Series for all Metabolites



Simulation-based Model Checking



PLTL properties - Metabolites

P>=1 [G (x=0)] % 01_always_steadystate_zero
P>=1 [G (d(x)=0 ^ x>0)] % 02_always_steadystate_above_zero
P>=1 [G (d(x)=0)] % 03_always_steadystate_any_value

P>=1 [F (G (x=0 ^ d(x)=0)) ^ F (d(x) != 0)] % 04_changing_and_finally_steadystate_of_zero
P>=1 [F (G (x>0 ^ d(x)=0)) ^ F (d(x) != 0)] % 05_changing_and_finally_steadystate_above_zero

P>=1 [G (d(x)<0)] % 07a_decreasing
P>=1 [G (d(x)>0)] % 08a_increasing

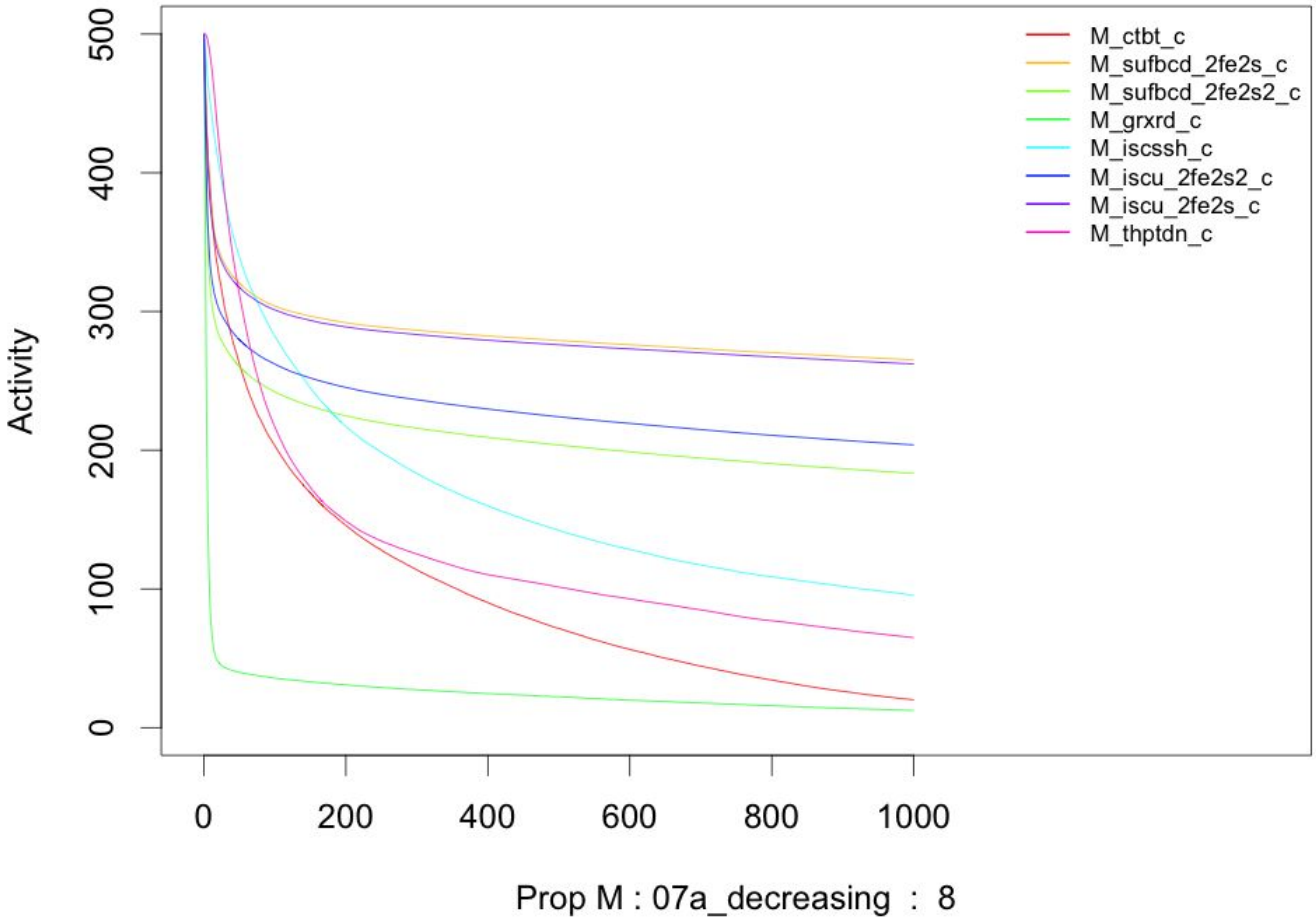
P>=1 [F(d(x)>0) ^ (d(x)>0 U (G d(x)<0))] % 09a_peaks_and_falls
P>=1 [F(d(x)<0) ^ (d(x)<0 U (G d(x)>0))] % 10a_falls_and_rises

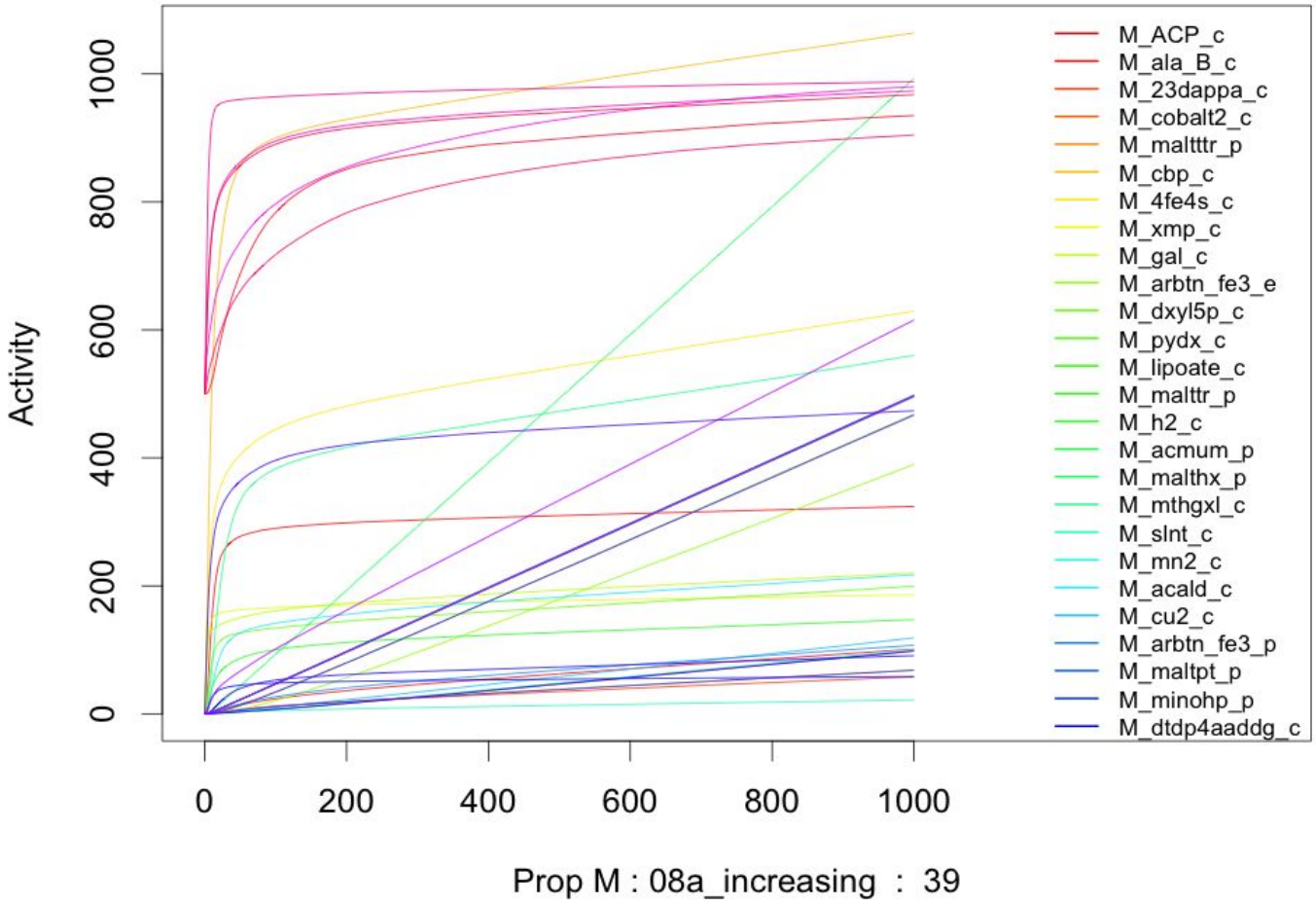
P>=1 [(F (d(x) != 0)) ^ ¬(F(G(x=0 ^ d(x)=0)))] % 13_activity_and_not_finally_steadystate_of_zero

P>=1 [G (x<=0.0001) ^ ¬ G (x=0)] % 14a_always_low_concentrations_0.0001

PLTL properties - Reactions

P>=1 [G (x=0)]	**% 01_never_active**
P>=1 [F (x>0)]	% 02_sometime_active
P>=1 [G (d(x) = 0)]	% 04_always_steadystate_active_any_value
P>=1 [F (G (x>0))]	% 05a_finally_active
P>=1 [F (G (x>0 ^ d(x)=0))]	% 05b_finally_active_steadystate
P>=1 [G (F (x>0))]	% 05c_always_active_again
P>=1 [F (G (x=0))]	% 06_finally_inactive
P>=1 [G (d(x)<0)]	% 07a_always_decreasing_activity
P>=1 [G (d(x)>0)]	% 08a_always_increasing_activity
P>=1 [F(d(x)>0) ^ (d(x)>0 U (G d(x)<0))]	% 09a_activity_peaks_and_falls
P>=1 [F(d(x)<0) ^ (d(x)<0 U (G d(x)>0))]	% 10a_activity_falls_and_rises
P>=1 [G (x<=0.0001) ^ ¬ G (x=0)]	% 14a_rare_events_0.0001


M_ctbt_c
M_sufbcd_2fe2s_c
M_sufbcd_2fe2s2_c
M_grxrd_c
M_iscssh_c
M_iscu_2fe2s2_c
M_iscu_2fe2s_c
M_thptdn_c
Activity
0
100
200
300
400
500
0
200
400
600
800
1000
Prop M : 07a_decreasing : 8

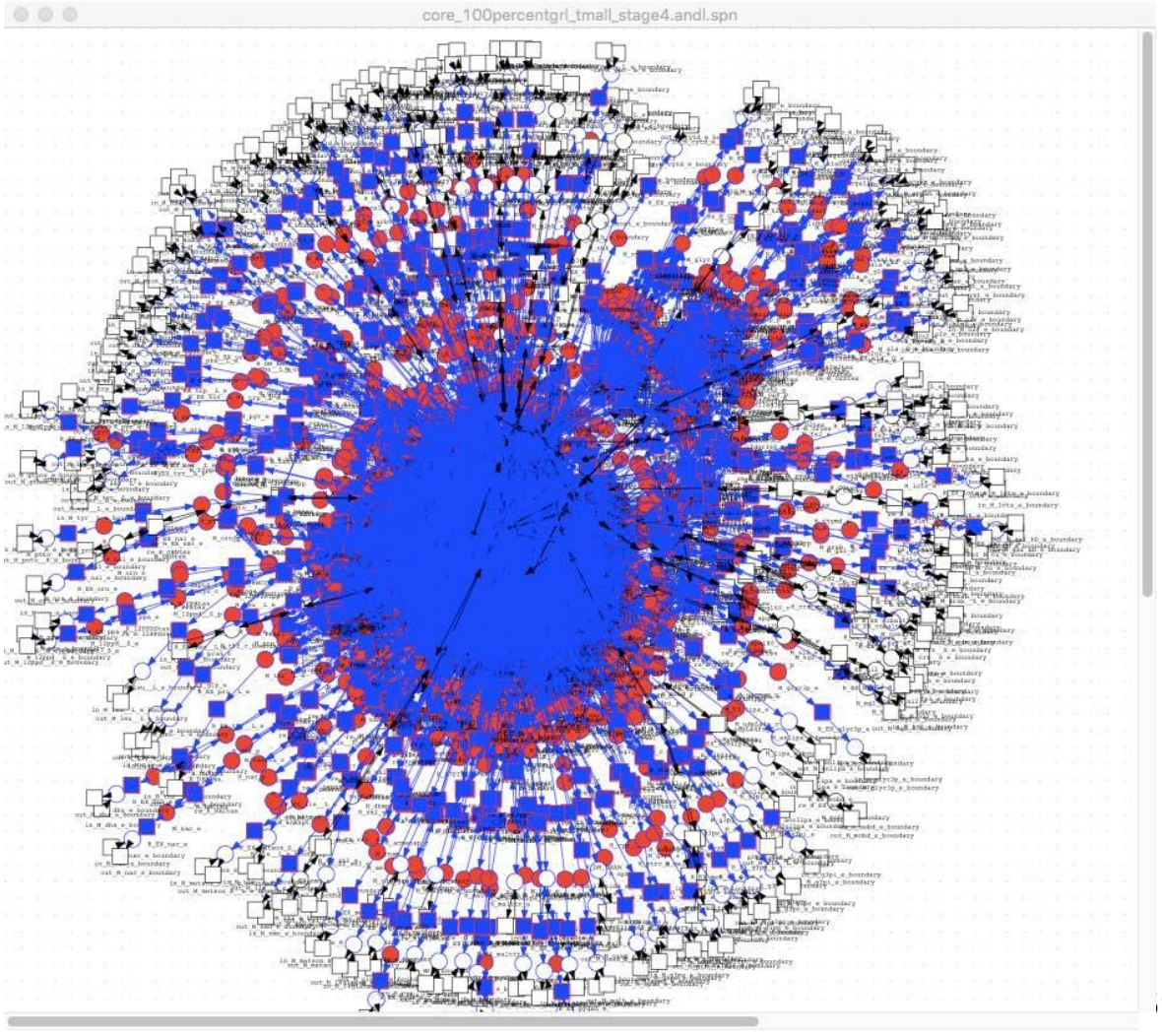

Activity
0
200
400
600
800
1000
Prop M : 08a_increasing : 39
M_ACP_c
M_ala_B_c
M_23dappa_c
M_cobalt2_c
M_maltttr_p
M_cbp_c
M_4fe4s_c
M_xmp_c
M_gal_c
M_arbtn_fe3_e
M_dxyl5p_c
M_pydx_c
M_lipoate_c
M_malttr_p
M_h2_c
M_acmum_p
M_malthx_p
M_mthgxl_c
M_slnt_c
M_mn2_c
M_acald_c
M_cu2_c
M_arbtn_fe3_p
M_maltpt_p
M_minohp_p
M_dtdp4aaddg_c

Dead Networks

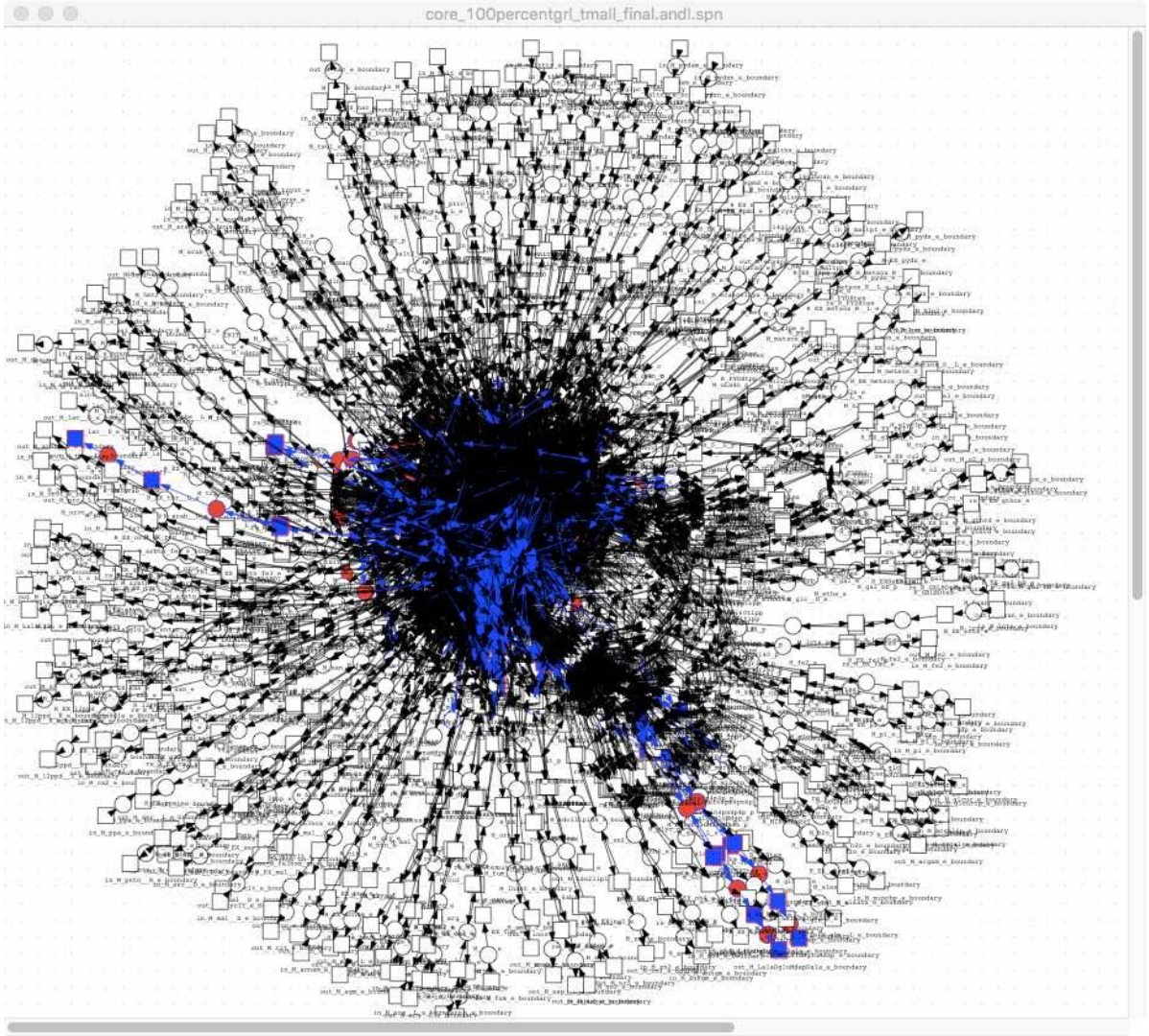
- All dead metabolites
  (M03 - always steady state any value)
  & the reactions for which they are substrates/products

- All dead reactions
  (R01 - never active)
  & their substrates + products

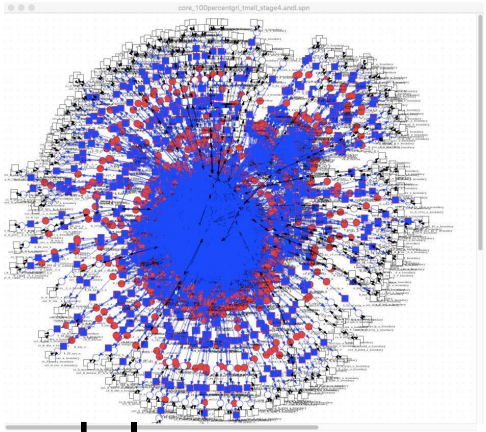
Dead network before repair



Dead network after repair



Conclusions



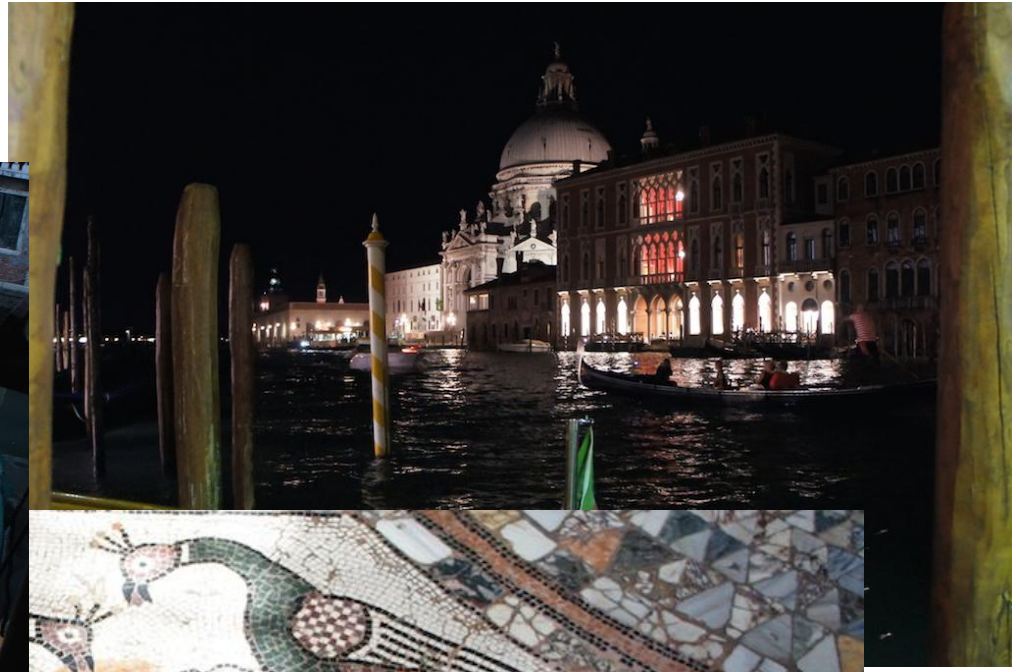
What we achieved so far:

- automated correction protocol
  for bacterial whole genome metabolic models
- set of analytical tools & techniques
- model database

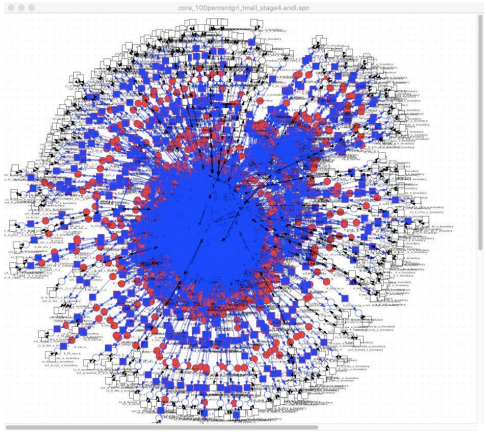
Side-effects:

- tool improvements
- integration within the synthetic biology theme

Carrying on

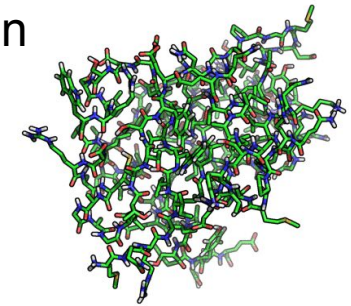


Carrying on



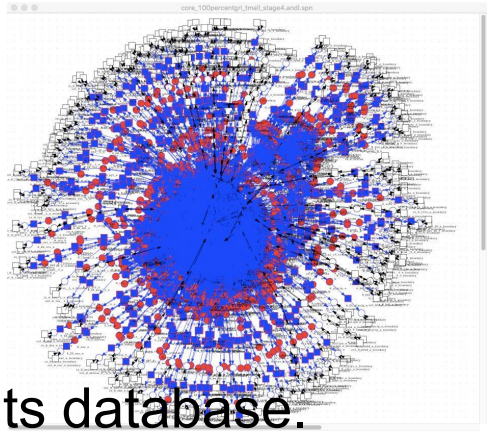
- Improve correction of networks beyond bad siphons (dead nets)

- Gap filling:
  finding missing reactions & metabolites due to
  - genes found but reactions missing in the Monk 55 data set
  - genes/reactions not found due to errors in sequencing etc
  - incomplete knowledge of gene-protein-reaction relation



- Extend model to multiscale by including protein structure (with Alessandro Pandini)

The Future !



- Develop method[s] to **optimise design of bacterial strains** using the constructed models & Brunel's model components database.

- **Select appropriate strain** & **donor alleles/genes** from other strains to optimise
  - target[s] production
  - ease/cost of gene transfer
  - gen[om]e stability

- Identify **genes to modify** to further enhance target achievement

The Team

- David Gilbert
- Monika Heiner

- Bello Suleiman
- Yasoda Jayaweera
- Alessandro Pandini
- Crina Grosan

- Nigel Saunders
- Arshad Khan

Thanks to

CEDPS

- *Supporting MH's visit*
- *Computing power*

BTU Cottbus

- *Christian Rohr*
- *Mostafa Herajy*

Uni Rostock

- *Karsten Wolf (LoLA)*

Questions?

P / T - invariants

A + E <-> A|E -> B + E

- P-invariants:
  - mass conservation
- T-invariants:
  - cyclic behaviour
  - steady state