V21 Minimal Reaction Cut Sets – Dual Description Method

Elementary Flux Modes vs. Extreme Pathways

Minimal Cut Sets

Dual Description Method

Definition of Elementary Flux Modes (EFMs)

A pathway P(**v**) is an **elementary flux mode** if it fulfills conditions C1 – C3.

(C1) **Pseudo steady-state**. **S** · **e** = 0. This ensures that none of the metabolites is consumed or produced in the overall stoichiometry.

(C2) **Feasibility**: rate $e_i \geq 0$ if reaction is irreversible. This demands that only thermodynamically realizable fluxes are contained in **e**.

(C3) **Non-decomposability***:* there is no vector **v** (except the null vector and **e**) fulfilling C1 and C2 and so that P(**v**) is a proper subset of P(**e**).
This is the core characteristics for EFMs and EPs and provides the decomposition of the network into smallest units that are able to hold the network in steady state.

C3 is often called „genetic independence“ because it implies that the enzymes in one EFM or EP are not a subset of the enzymes from another EFM or EP.

Klamt & Stelling Trends Biotech 21, 64 (2003)

Definition of Extreme Pathways (EPs)

The pathway *P(**e**)* is an **extreme pathway**
if it fulfills conditions C1 – C3 AND conditions C4 – C5.

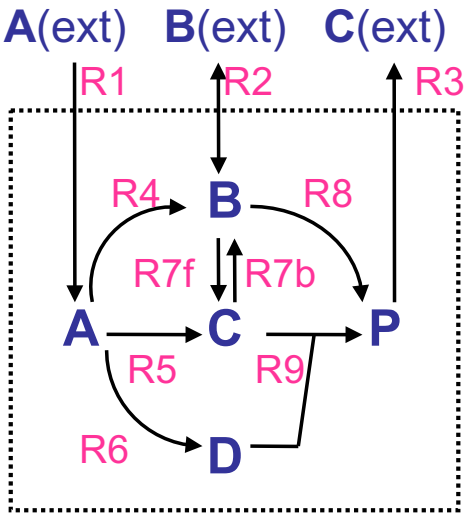
(C4) **Network reconfiguration**: Each reaction must be classified either as exchange flux or as internal reaction.
All **reversible** internal reactions must be **split up** into two separate, irreversible reactions (forward and backward reaction).

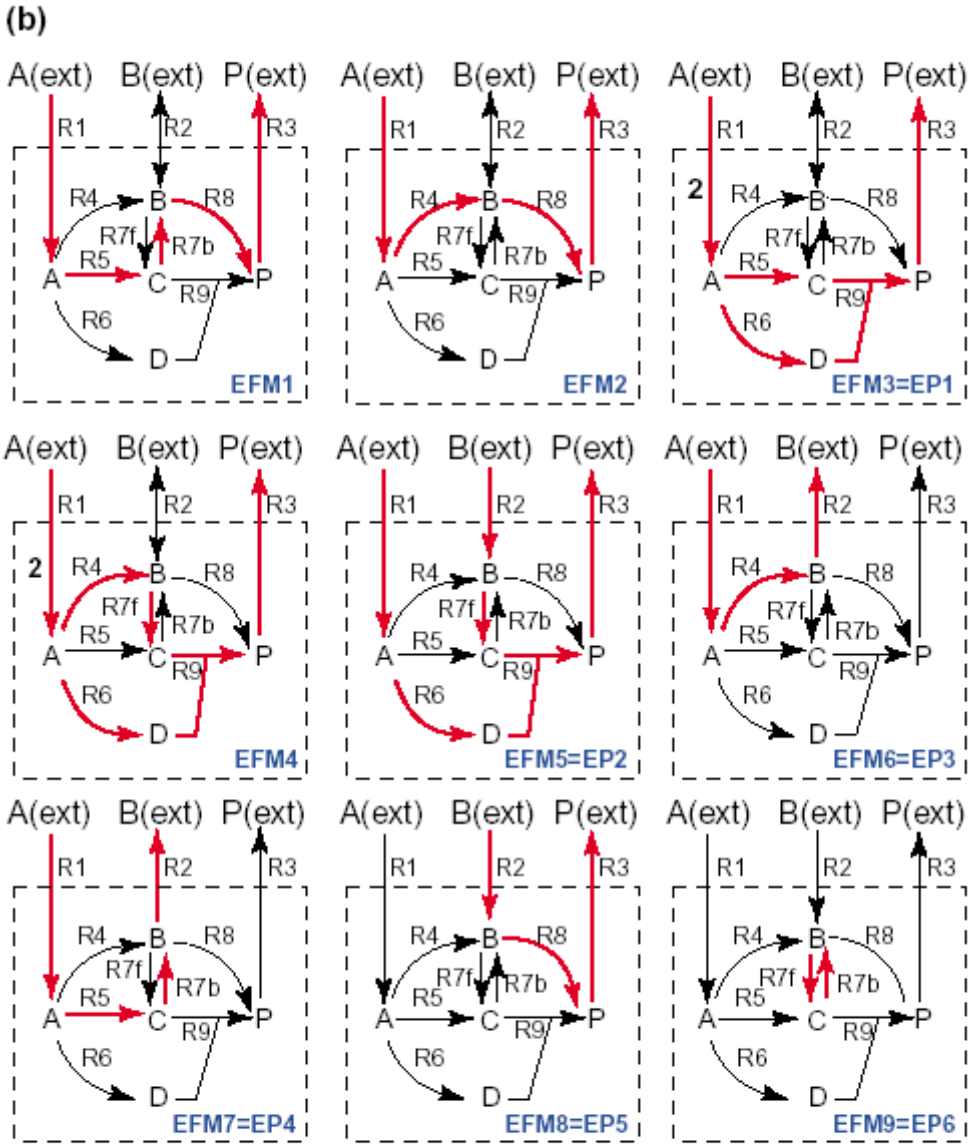
(C5) **Systemic independence**: the set of EPs in a network is the **minimal** set of EFMs that can describe all feasible steady-state flux distributions.

The algorithms for computing EPs and EFMs are quite similar.
We will not cover the algorithmic differences here.

Klamt & Stelling Trends Biotech 21, 64 (2003)

Reconfigured Network: split up R7





3 EFMs are not systemically independent:

EFM1 = EP4 + EP5

EFM2 = EP3 + EP5

EFM4 = EP2 + EP3

Klamt & Stelling Trends Biotech 21, 64 (2003)

Property 1 of EFMs

The only difference in the set of EFMs emerging upon reconfiguration consists in the **two-cycles** that result from splitting up reversible reactions.

However, two-cycles are not considered as meaningful pathways.

Valid for any network: Property 1

Reconfiguring a network by **splitting up reversible reactions**
leads to the same set of meaningful EFMs.

Property 2
If all exchange reactions in a network are irreversible then the sets of meaningful EFMs (both in the original and in the reconfigured network) and EPs coincide.

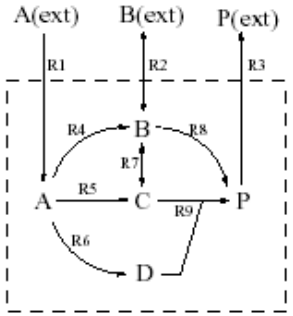
Klamt & Stelling Trends Biotech 21, 64 (2003)

EFMs vs. EPs

What is the consequence **when all exchange fluxes** (and hence all reactions in the network) are made **irreversible**?

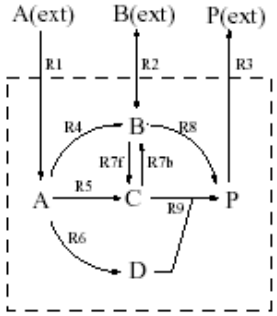
Table 1. Configurations of the example network (upper part N1 and N3; lower part N2 and N4), with corresponding elementary flux modes (EFM) and extreme pathways (EP) (see also Fig. 1)

N1 (R2 and R7 reversible) N3 (as N1 but R2 irreversible)



N1		N3		Reactions								
EFMs		EFMs		R1	R2	R3	R4	R5	R6	R7	R8	R9
EFM1		×		1	0	1	0	1	0	−1	1	0
EFM2		×		1	0	1	1	0	0	0	1	0
EFM3		×		2	0	1	0	1	1	0	0	1
EFM4		×		2	0	1	1	0	1	1	0	1
EFM5		×		1	1	1	0	0	1	1	0	1
EFM6				1	−1	0	1	0	0	0	0	0
EFM7				1	−1	0	0	1	0	−1	0	0
EFM8		×		0	1	1	0	0	0	0	1	0

N2 (R2 reversible, R7 split up) N4 (as N2 but R2 irreversible)



N2		N4		Reactions									
EFMs	EPs	EFMs	EPs	R1	R2	R3	R4	R5	R6	R7f	R8	R9	R7b
EFM1		×	EP1′	1	0	1	0	1	0	0	1	0	1
EFM2		×	EP2′	1	0	1	1	0	0	0	1	0	0
EFM3	EP1	×	EP3′	2	0	1	0	1	1	0	0	1	0
EFM4		×	EP4′	2	0	1	1	0	1	1	0	1	0
EFM5	EP2	×	EP5′	1	1	1	0	0	1	1	0	1	0
EFM6	EP3			1	−1	0	1	0	1	0	0	0	0
EFM7	EP4			1	−1	0	0	1	0	0	0	0	1
EFM8	EP5	×	EP6′	0	1	1	0	0	0	0	1	0	0
EFM9	EP6	×	EP7′	0	0	0	0	0	0	1	0	0	1

Klamt & Stelling Trends Biotech 21, 64 (2003)

Then EFMs and EPs always **co-incide!**

Software: FluxAnalyzer, based on Matlab

Steffen Klamt.

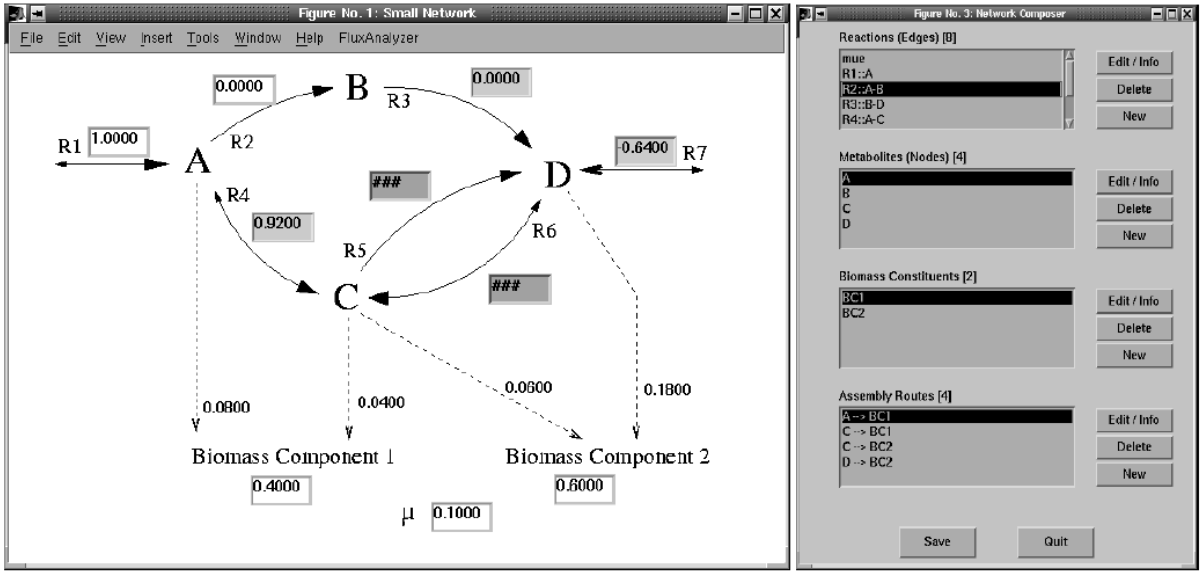


Fig. 1. The network project of 'SMALLNET' constructed by the FluxAnalyzer. Left: interactive flux map displaying a flux scenario (unknown rates are denoted by '###'). Right: network composer.

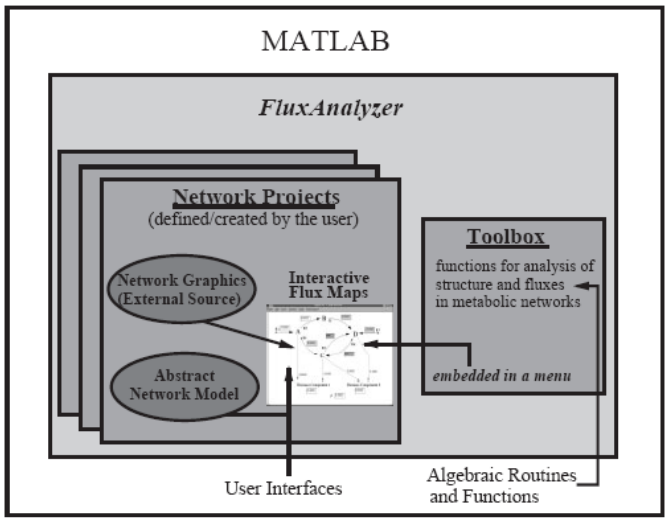


Fig. 2. Structural setup of the FluxAnalyzer.

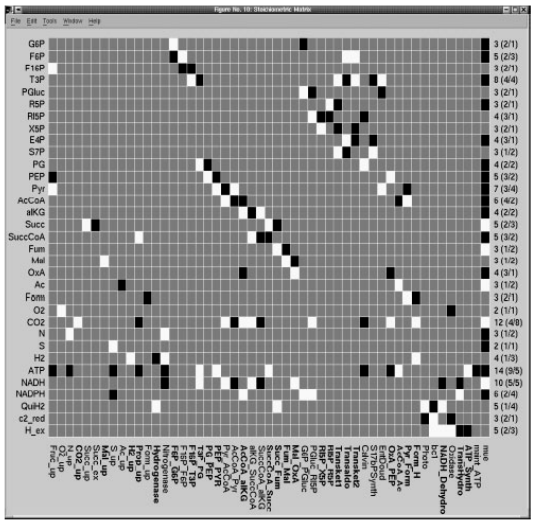


Fig. 3. Concise graphical representation of the stoichiometric matrix (here: catabolic part of the network studied in Klamt *et al.*, 2002)

`FluxAnalyzer` has both EPs and EFMs implemented.

Allows convenient studies of metabolicsystems.

Klamt et al.
Bioinformatics 19, 261 (2003)

Strain optimization based on EFM-analysis

Metabolic Engineering 12 (2010) 112–122

Contents lists available at ScienceDirect

Metabolic Engineering

journal homepage: www.elsevier.com/locate/ymben

Rational design and construction of an efficient *E. coli* for production of diapolycopendioic acid

Pornkamol Unrean, Cong T. Trinh, Friedrich Srienc *

Department of Chemical Engineering and Materials Science, and BioTechnology Institute, University of Minnesota, 240 Gortner Laboratory, 1479 Gortner Ave, St. Paul, MN 55108, USA

Carotenoids (e.g. DPL and DPA) are light-harvesting pigments, UV-protecting compounds, regulators of membrane fluidity, and antioxidants.

They are used as nutrient supplements, pharmaceuticals, and food colorants.

Aim: increase carotenoid synthesis in *E.coli*

Unrean et al. Metabol Eng 12, 112-122 (2010)

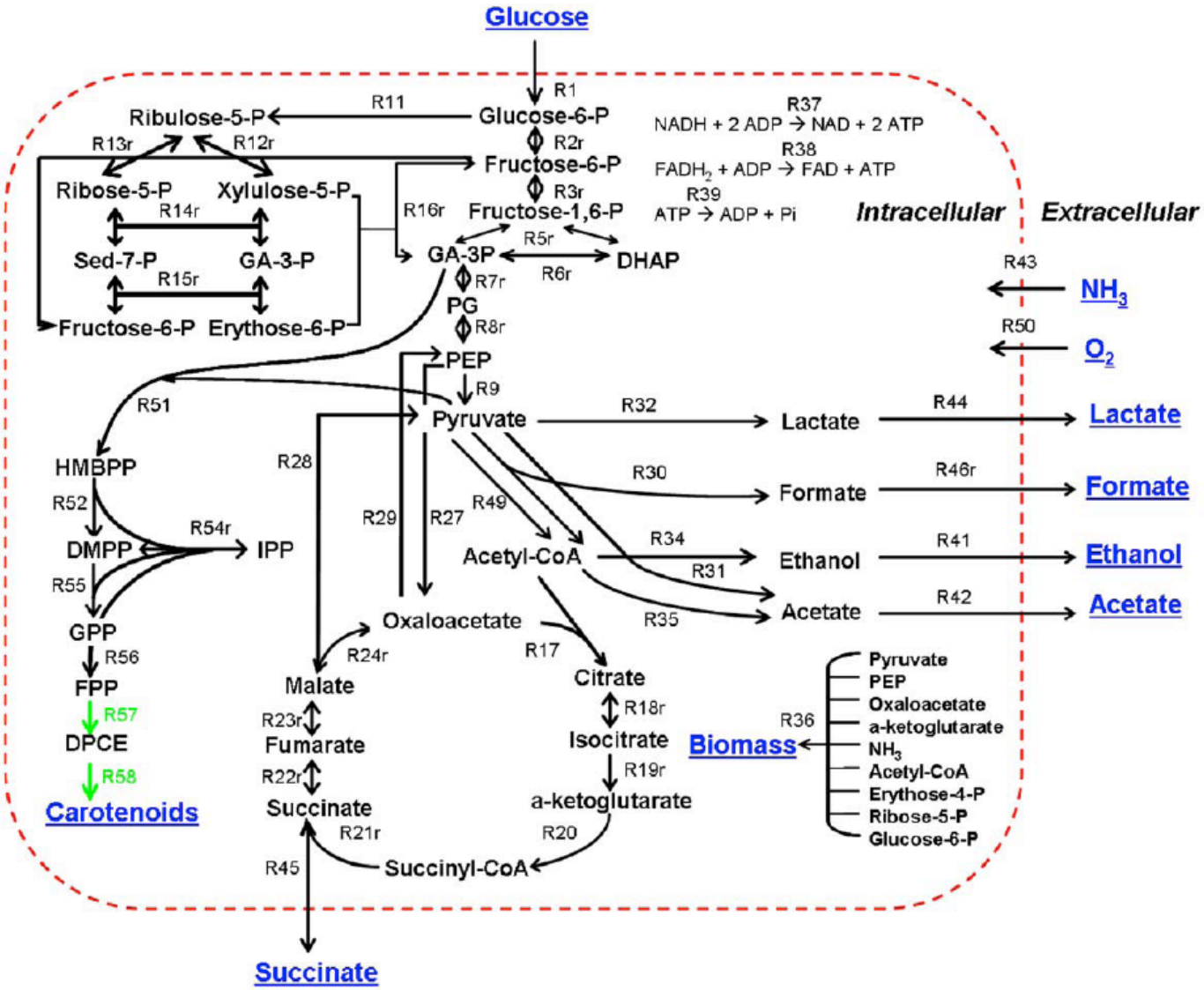
Metabolic network of recombinant *E.coli*

58 metabolic reactions
22 reversible
36 irreversible

57 metabolites

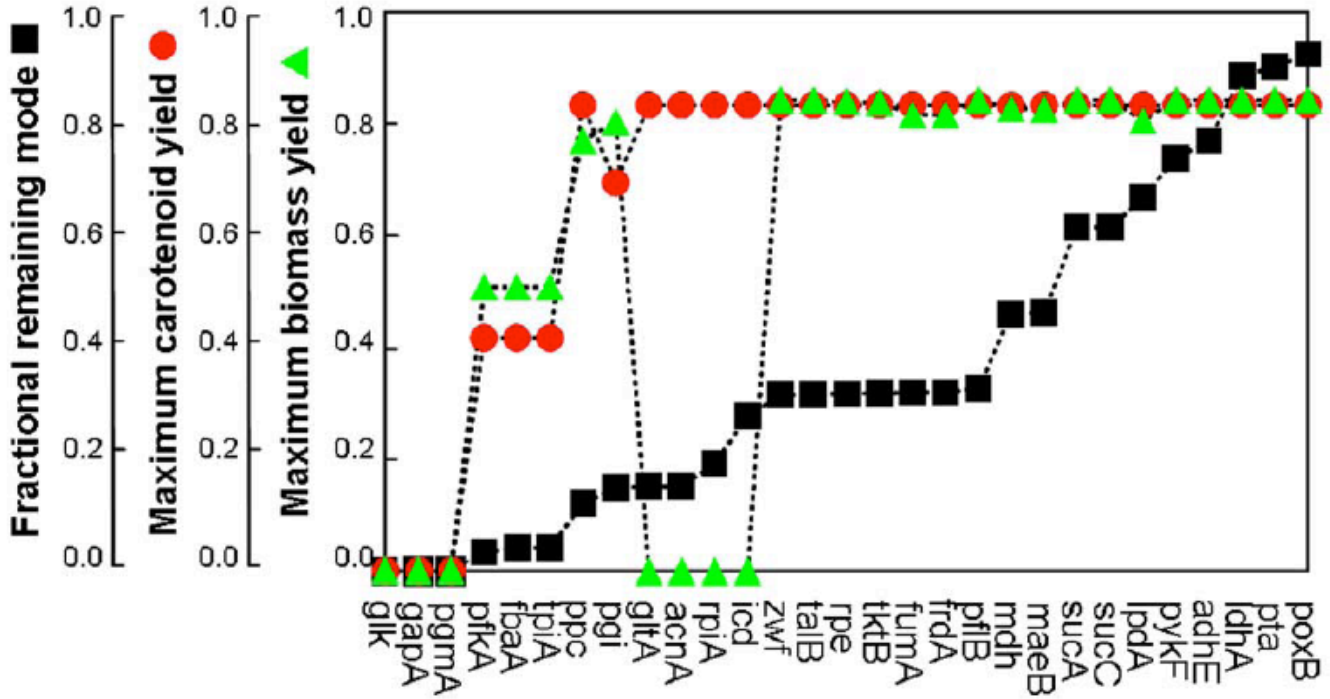
29532 EFMs

In 5923 EFMs, the production of biomass and DPA are coupled.



Unrean et al. Metabol Eng 12, 112-122 (2010)

Effect of single gene deletions



Results of virtual gene knockout calculations (counting number of EFMs and computing their yield from reaction stochiometries).

Select target genes where knockouts still maintain a maximum possible yield of carotenoid production, a reasonable yield of biomass while the largest number of EFMs is eliminated.

Unrean et al. Metabol Eng 12, 112-122 (2010)

Effect of single gene deletions

Strain	Total modes	Aerobic modes	Anaerobic modes	Predicted CRT yield[a]
Wild-type	29,532	24,155	5377	0.0-426
Δ*ldhA*	15,662	13,405	2257	0.0-426
Δ*ldhA*Δ*frdA*	8573	7810	763	0.0-426
Δ*ldhA*Δ*frdA*Δ*poxB*	7541	6861	680	0.0-426
Δ*ldhA*Δ*frdA*Δ*poxB*Δ*pta*	6171	5600	571	0.0-426
Δ*ldhA*Δ*frdA*Δ*poxB*Δ*pta*Δ*adhE*	4099	4099	0	0.0-426
Δ*ldhA*Δ*frdA*Δ*poxB*Δ*pta*Δ*adhE*Δ*pykF*	2573	2573	0	0.0-426
Δ*ldhA*Δ*frdA*Δ*poxB*Δ*pta*Δ*adhE*Δ*pykF*Δ*zwf*	375	375	0	0.0-426
Δ*ldhA*Δ*frdA*Δ*poxB*Δ*pta*Δ*adhE*Δ*pykF*Δ*zwf*Δ*maeB*	5	5	0	0.4-426

[a] Yield is in mg-diapolycopendioic acid/g–glucose.

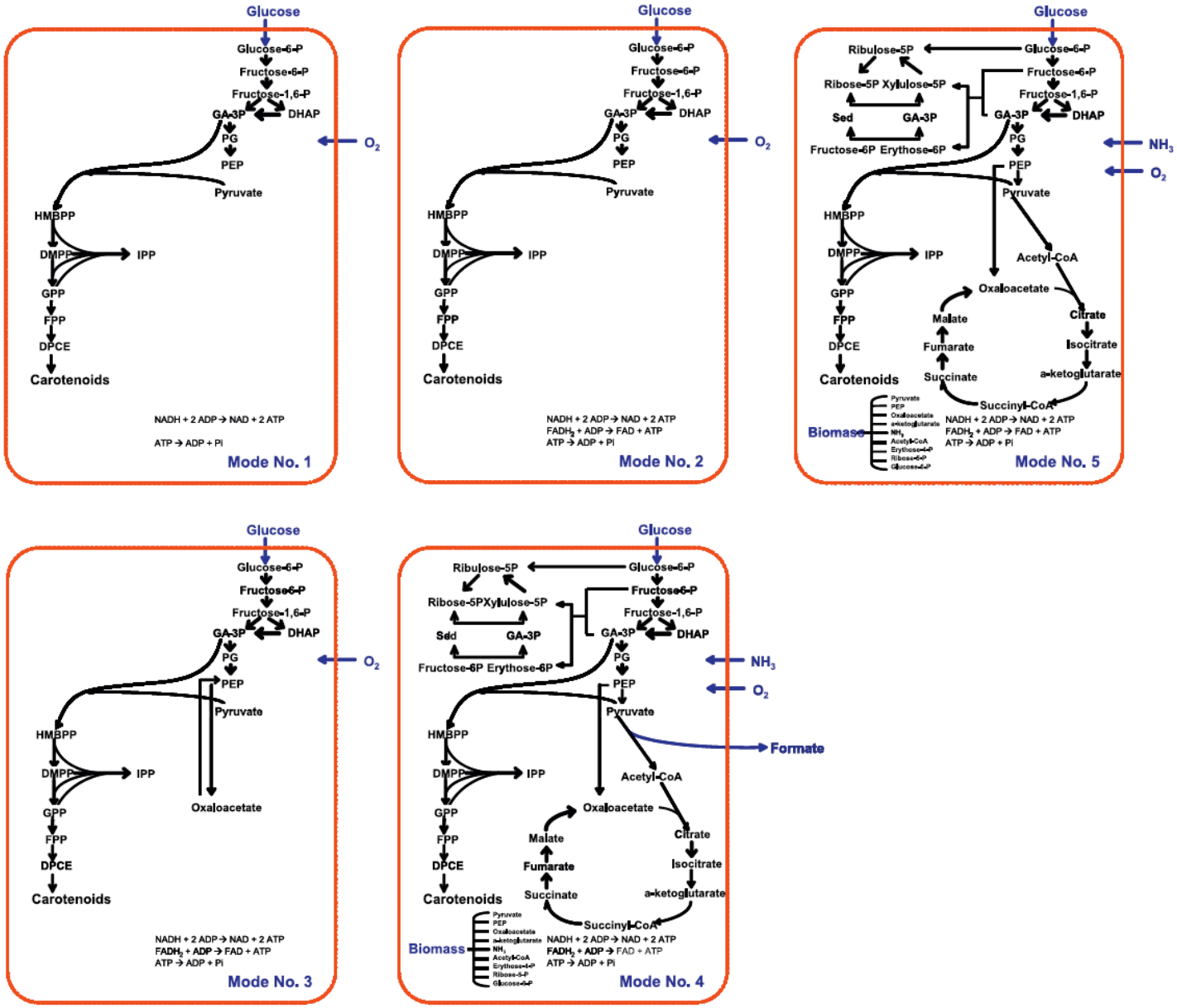
Deleted Reaction	Corresponding gene	Enzyme	Pathway
R9	*pykF*	Pyruvate kinase	Glycolysis
R11	*zwf*	Glucose-6-phosphate-1-dehydrogenase	Pentose phosphate
R22	*frdA*	Fumarate reductase	Fermentation
R28	*maeB*	Malate dehydrogenase	Anapleurotic
R31	*poxB*	Pyruvate oxidase	Fermentation
R32	*ldhA*	Lactate dehydrogenase	Fermentation
R34	*adhE*	Alcohol dehydrogenase	Fermentation
R35	*pta*	Phosphate acetyltransferase	Fermentation

Optimal: 8 gene knockouts lead to predicted over-production of DPL and DPA.

After this deletion, only 5 EFMs remain.

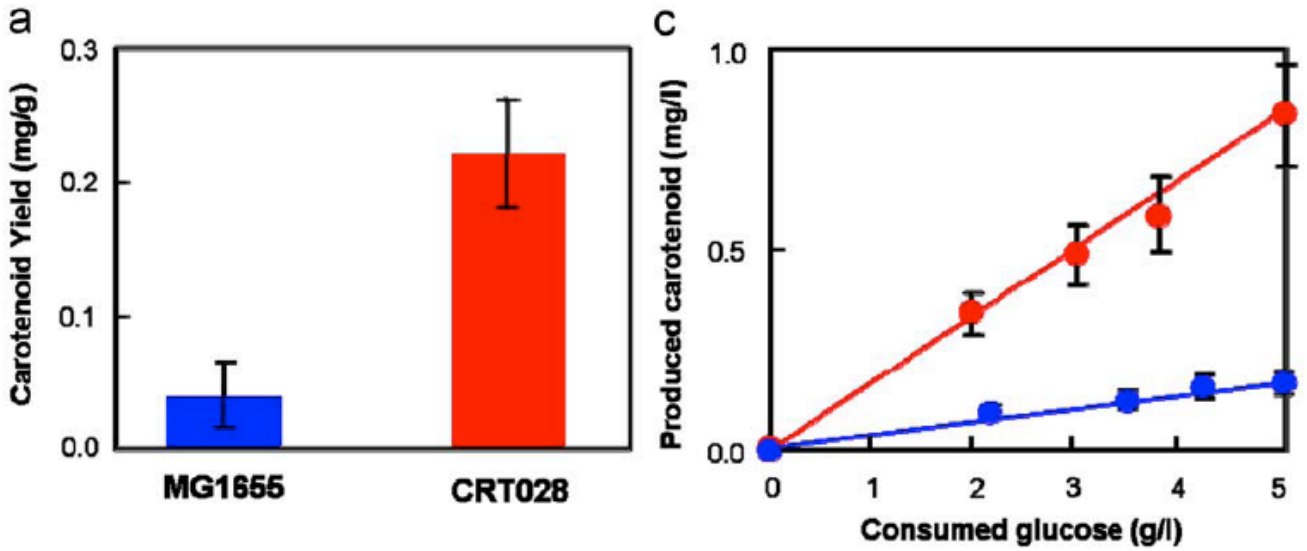
Unrean et al. Metabol Eng 12, 112-122 (2010)

Remaining EFMs



Unrean et al. Metabol Eng 12, 112-122 (2010)

Experimental verification: increased carotenoid yield



Mutant grows slower, but CRT production is increased 4 times.

	MG1655/ pACMNOx	CRT028/ pACMNOx
Growth rate (/h)	0.17 ± 0.02	0.13 ± 0.01
Carotenoid production (mg/l)	0.19 ± 0.02	0.83 ± 0.20
Carotenoid yield (mg carotenoid/g glucose)	0.04 ± 0.00	0.17 ± 0.04
Specific production (mg carotenoid/g cell dry weight-h)	0.01 ± 0.00	0.10 ± 0.02

Unrean et al. Metabol Eng 12, 112-122 (2010)

Complexity of finding and enumerating EFMs

Theorem: Given a stochiometric matrix *S*, an elementary mode can be found in polynomial time.

Theorem: In case all reactions in a metabolic network are reversible, the elementary modes can be enumerated in polynomial time.

The enumeration task becomes dramatically more difficult if the reactions are irreversible.
In this case, the modes of the network form a cone, and the elementary modes are the rays of the cone.

Theorem: Given a flux cone and two coordinates *i* and *j*, deciding if there exists an extreme ray of the cone that contains both r_i and r_j is NP-complete.

Theorem: Given a matrix *S* and a number *k*, deciding whether an elementary mode exist that contains at most *k* reactions is NP-complete.

It is an open question whether all elementary modes of a general network can be enumerated in polynomial time.

Acuna et al. BioSystems 99, 210-214 (2010); BioSystems 95, 51-60 (2009)

Minimal cut sets in biochemical reaction networks

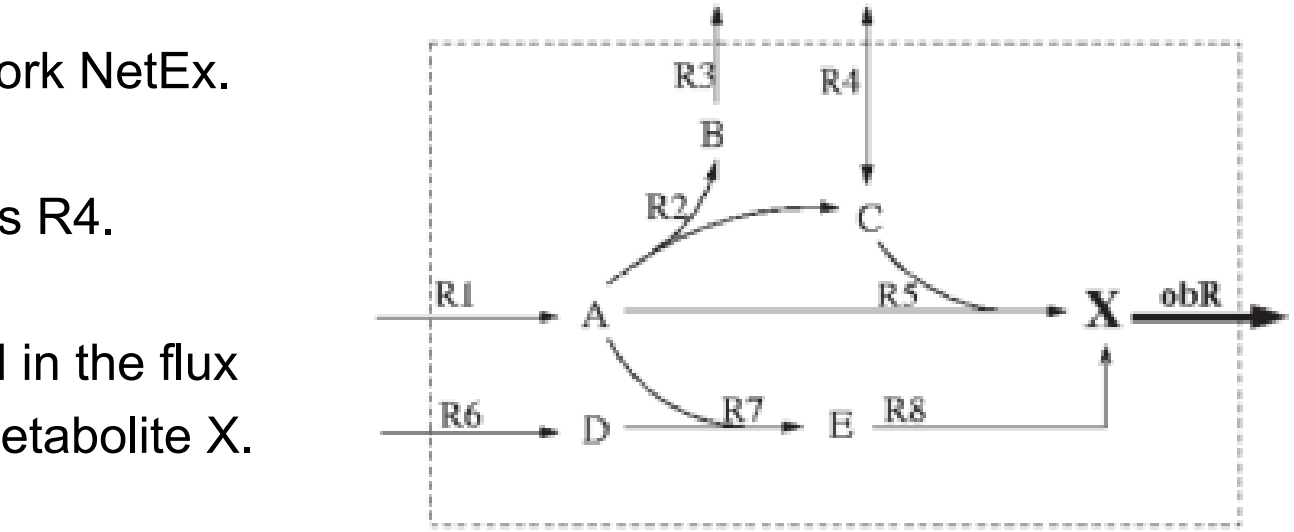
Concept of minimal cut sets (MCSs): smallest „failure modes“ in the network that render the correct functioning of a cellular reaction impossible.

Right: fictitious reaction network NetEx.

The only reversible reaction is R4.

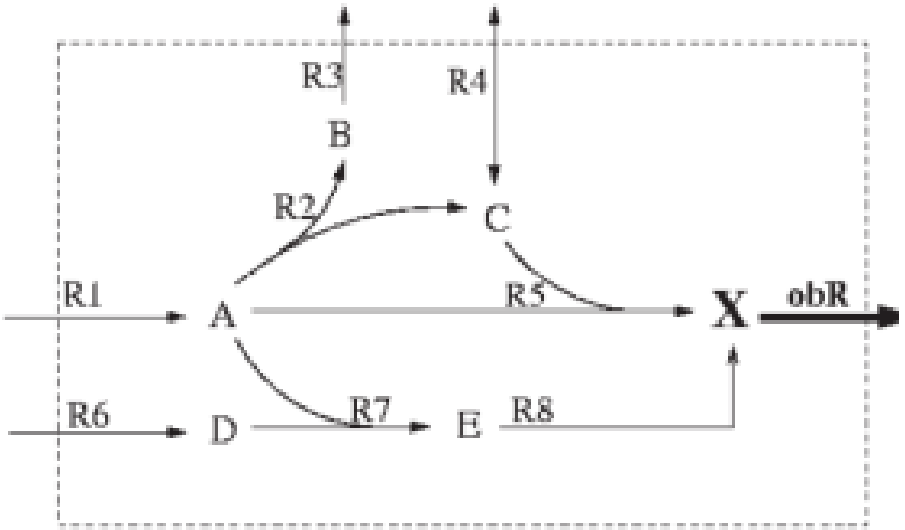
We are particularly interested in the flux `obR` exporting synthesized metabolite X.

→ Characterize **solution space** by computing **elementary flux modes**.



Klamt & Gilles, Bioinformatics 20, 226 (2004)

Elementary flux modes of NetEx



	R1	R2	R3	R4	R5	R6	R7	R8	obR
Elementary modes									
EM1	1	1	1	−1	0	0	0	0	0
EM2	1	0	0	0	0	1	1	1	1
EM3	2	1	1	0	1	0	0	0	1
EM4	1	0	0	1	1	0	0	0	1

One finds 4 elementary flux modes for NetEx.

3 of them (shaded) allow the production of metabolite X.

Klamt & Gilles, Bioinformatics 20, 226 (2004)

Cut set

Now we want to prevent the production of metabolite X.

→ demand that there is no balanced flux distribution possible which involves `obR`.

Definition. A set of reactions is termed a **cut set** (with respect to a defined objective reaction)
if after the removal of these reactions from the network
no feasible balanced flux distribution involves the objective reaction.

Klamt & Gilles, Bioinformatics 20, 226 (2004)

Cut set

A **trivial cut set** is the reaction itself: C0 = {obR}.

Another extreme case is the **removal of all reactions** except obR ..
This is very inefficient if this involves knocking out these genes or developing small molecule inhibitors!

Desirable solutions:

- From an engineering point of view, it might be desirable to cut reactions **at the beginning of a pathway**.

- The production of biomass is usually not coupled to a single gene or enzyme, and can therefore not be directly inactivated.

Klamt & Gilles, Bioinformatics 20, 226 (2004)

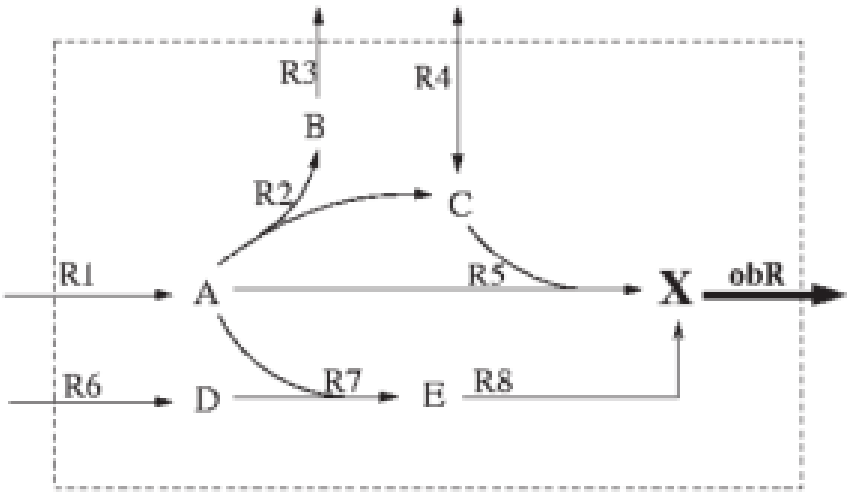
Cut set

C1 = {R5,R8} is a cut set already sufficient for preventing the production of X.

Removing R5 or R8 alone is not sufficient.

Definition. A cut set C (related to a defined objective reaction) is a **minimal cut set** (MCS) if no proper subset of C is a cut set.

→ C1 is a minimal cut set



	R1	R2	R3	R4	R5	R6	R7	R8	obR
Elementary modes									
EM1	1	1	1	−1	0	0	0	0	0
EM2	1	0	0	0	0	1	1	1	1
EM3	2	1	1	0	1	0	0	0	1
EM4	1	0	0	1	1	0	0	0	1
Minimal cut sets (objective reaction: obR)									
MCS0									×
MCS1	×								
MCS2					×	×			
MCS3					×		×		
MCS4					×			×	
MCS5		×		×		×			
MCS6			×	×		×			
MCS7		×		×			×		
MCS8			×	×			×		
MCS9		×		×				×	
MCS10			×	×				×	

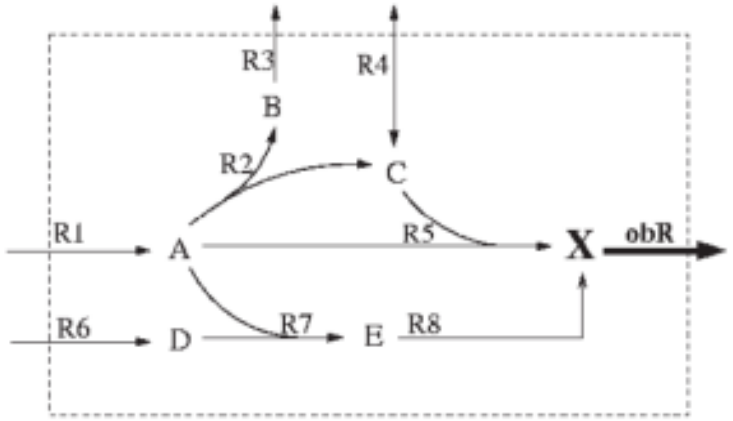
Klamt & Gilles, Bioinformatics 20, 226 (2004)

Remarks

(1) An MCS always guarantees **dysfunction** as long as the assumed network structure is currect. However, additional regulatory circuits or capacity restrictions may allow that even a proper subset of a MCS is a cut set.
The MCS analysis should always be seen from a purely structural point of view.

(2) After removing a complete MCS from the network,
other pathways producing other metabolites may still be active.

(3) MCS4 = {R5,R8} clearly stops production of X.



What about MCS6 = {R3,R4,R6}?

Cannot X be still be produced via R1, R2, and R5?
However, this would lead to accumulation of B
and is therefore **physiologically impossible**.

Klamt & Gilles, Bioinformatics 20, 226 (2004)

Algorithm for computing MCSs

The MCSs for a given network and objective reaction are members of the power set of the set of reaction indices and are uniquely determined.

A systematic computation must ensure that the calculated MCSs are:

(1) cut sets („destroying“ all possible balanced flux distributions involving the objective reaction), and

(2) that the MCSs are really minimal, and

(3) that all MCSs are found.

Klamt & Gilles, Bioinformatics 20, 226 (2004)

Algorithm for computing MCSs

Necessary condition for cut sets: they **interrupt** all possible balanced flux distributions involving the objective reaction,

Use the fact that any feasible steady-state flux distribution **r** in a given network can be represented by a non-negative **linear combination** of the *N* **elementary modes**:

$$\mathbf{r} = \sum_{i=1}^{N} \alpha_i EM_i, \qquad \alpha_i \geq 0$$

To ensure that the rate r_k of the objective reaction is 0 in all **r**,
each EM must contain 0 at the *k*-th place.

$\rightarrow$ If C is a proper cut set the following cut set condition must hold:
For each EM involving the objective reaction (with a non-zero value),
there is at least one reaction in C also involved in this EM.

This guarantees that all EMs, in which the objective reaction participates,
will vanish when the reactions in the cut set are removed from the network.

Klamt & Gilles, Bioinformatics 20, 226 (2004)

Algorithm

ALGORITHM:

(1) Calculate the EMs in the given network

(2) Define the objective reaction obR

(3) Choose all EMs where reaction obR is non-zero and store it in the binary array *em_obR* (*em_obR*[*i*][*j*]==1 means that reaction *j* is involved in EM *i*)

(4) Initialize arrays *mcs* and *precutsets* as follows (each array contains sets of reaction indices): append [*j*] to *mcs* if reaction *j* is essential (*em_obR*[*i*][*j*]=1 for each EM *i*), otherwise to *precutsets*

(5) FOR i=2 TO MAX_CUTSETSIZE

(5.1) *new_precutsets*=[ ];

(5.2) FOR $j = 1$ TO q (q: number of reactions)

(5.2.1) Remove all sets from *precutsets* where reaction *j* participates

(5.2.2) Find all sets of reactions in *precutsets* that do not cover at least one EM in *em_obR* where reaction *j* participates; combine each of these sets with reaction *j* and store the new preliminary cut sets in *temp_precutsets*

(5.2.3) Drop all *temp_precutsets* which are a superset of any of the already determined minimal cut sets stored in *mcs*

(5.2.4) Find all retained *temp_precutsets* which do now cover all EMs and append them to *mcs*; append all others to *new_precutsets*

ENDFOR

(5.3) If isempty(*new_precutsets*)

(5.3.1) Break

ELSE

(5.3.2) *precutsets*=*new_precutsets*

ENDIF

ENDFOR

(6) result: *mcs* contains the MCSs

Klamt & Gilles, Bioinformatics 20, 226 (2004)

According to Acuna (2009) this algorithm is often very inefficient.
More efficient algorithms exist already and are still being developed.

Applications of MCSs

Target identification and repression of cellular functions

A screening of all MCSs allows for the identification of the **best suitable manipulation**.

For practical reasons, the following conditions should be fulfilled:

- usually, a **small number** of **interventions** is **desirable** (small size of MCS)

- **other pathways** in the network should only be **weakly affected**

- some of the cellular functions might be difficult to shut down genetically or by inhibition, e.g. if many isozymes exist for a reaction.

Klamt & Gilles, Bioinformatics 20, 226 (2004)

Applications of MCSs

Network verification and mutant phenotype predictions

We expect that cutting away an MCS from the network is definitely intolerable for the cell with respect to certain cellular reactions/processes.

Such predictions, derived purely from network structure,
are a useful strategy for **verification** of hypothetical or reconstructed networks.

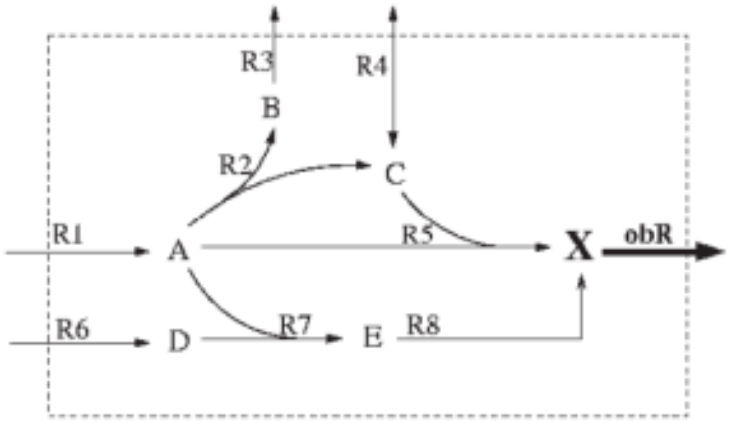
If the **outcome** of **prediction** and experiments differ,
this often indicates an incorrect or incomplete network structure.

Klamt & Gilles, Bioinformatics 20, 226 (2004)

Structural fragility and robustness

If we assume that each reaction in a metabolic network has the same probability to fail, **small MCSs** are most probable to be responsible for a failing objective function.

Define a **fragility coefficient** F_i as the reciprocal of the average size of all MCSs in which reaction *i* participates.



	R1	R2	R3	R4	R5	R6	R7	R8	obR
Elementary modes									
EM1	1	1	1	−1	0	0	0	0	0
EM2	1	0	0	0	0	1	1	1	1
EM3	2	1	1	0	1	0	0	0	1
EM4	1	0	0	1	1	0	0	0	1
Minimal cut sets (objective reaction: obR)									
MCS0									×
MCS1	×								
MCS2					×	×			
MCS3					×		×		
MCS4					×			×	
MCS5		×		×		×			
MCS6			×	×		×			
MCS7		×		×			×		
MCS8			×	×			×		
MCS9		×		×				×	
MCS10			×	×				×	
F_i	1	1/3	1/3	1/3	1/2	3/8	3/8	3/8	1

Besides the essential reaction R1, reaction R5 is most crucial for the objective reaction.

Klamt & Gilles, Bioinformatics 20, 226 (2004)

Example: MCSs in the central metabolism of *E.coli*

objective reaction
„biomass synthesis“

Network: 110 reactions,
89 metabolites,
see Stelling et al. (2002)

Table 2. Overview on computed MCSs in the central metabolism of *E.coli* for growth on four different substrates

	Acetate	Succinate	Glycerol	Glucose
No. of EMs with growth	363	3421	9479	21 592
No. of MCSs (objective reaction: growth)	245	1255	2970	4225
Maximal number of preliminary MCSs (during computation)	3563	69 628	344 196	902 769
Computation time (Intel Pentium, 1 MHZ; 4 GB RAM)	7 s	20 min	5.42 h	29.67 h
F_i values (in parentheses: size of the smallest MCS in which the reaction occurs)				
F16P-bisphosphatase	1 (1)	1 (1)	1 (1)	0.102 (6)
ATP-synthase	1 (1)	0.325 (3)	0.141 (3)	0.149 (3)
SuccCoA-synthetase	0.207 (2)	0.145 (2)	0.125 (2)	0.131 (2)
PEP-carboxylase	0.128 (2)	0.117 (2)	0.120 (2)	0.143 (2)
Malic enzyme	0.5 (2)	0.5 (2)	0.114 (2)	0.123 (2)
R15P-X5P (epimerase)	0.198 (2)	0.135 (2)	0.128 (2)	0.148 (2)
F	0.783	0.718	0.699	0.643

The computation time does not involve the time needed for computing the elementary modes. F_i: fragility coefficient of reaction i; **F**: network (overall) fragility coefficient.

Klamt & Gilles, Bioinformatics 20, 226 (2004)

Conclusion - MCS

A MCS is an irreducible combination of network elements whose simultaneous inactivation leads to a **guaranteed dysfunction** of certain cellular reactions or processes.

Theorem: Determining a reaction cut of minimum cardinality is **NP-hard**.

→ Computing MCSs and EMs becomes challenging in large networks.

MCSs are **inherent** and uniquely determined structural features of metabolic networks similar to EMs.

Analyzing the MCSs gives deeper insights in the **structural fragility** of a given metabolic network and is useful for
identifying target sets for an intended repression of network functions.

Klamt & Gilles, Bioinformatics 20, 226 (2004)
Acuna et al. BioSystems 95, 51-60 (2009)

The Double Description method: Theoretical framework behind EFM and EP / Integration Algorithms

Double Description Method Revisited

Komei Fukuda[1] and Alain Prodon[2]

[1] Institute for Operations Research, ETHZ, CH-8092 Zürich, Switzerland
[2] Department of Mathematics, EPFL, CH-1015 Lausanne, Switzerland

in „Combinatorics and Computer Science Vol. 1120“ edited by Deza, Euler, Manoussakis, Springer, 1996:91

BMC Bioinformatics

BioMed Central

Research article

Open Access

Computation of elementary modes: a unifying framework and the new binary approach

Julien Gagneur[†1] and Steffen Klamt[*†2]

Address: [1]Cellzome AG, Meyerhofstr. 1, 69117 Heidelberg, Germany and [2]Max Planck Institute for Dynamics of Complex Technical Systems, Sandtorstr. 1, D-39106 Magdeburg, Germany

Email: Julien Gagneur - julien.gagneur@cellzome.com; Steffen Klamt* - klamt@mpi-magdeburg.mpg.de

* Corresponding author †Equal contributors

Published: 04 November 2004

BMC Bioinformatics 2004, 5:175 doi:10.1186/1471-2105-5-175

Received: 28 June 2004
Accepted: 04 November 2004

This article is available from: http://www.biomedcentral.com/1471-2105/5/175

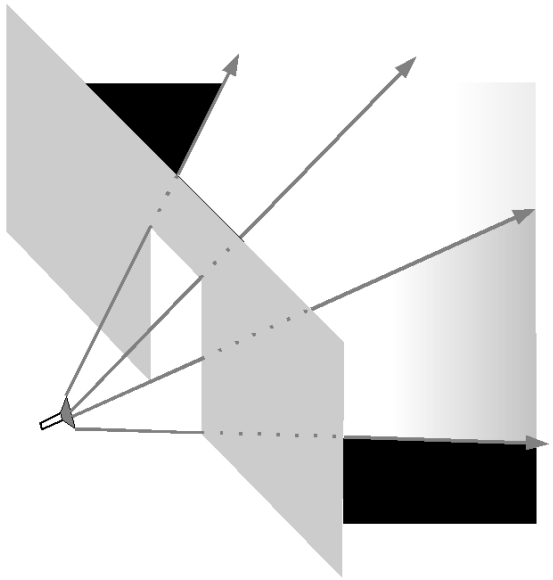
Review: Double Description Method (1953)

The Double Description method is the basis for simple & efficient algorithms for the task of **enumerating extreme rays.**

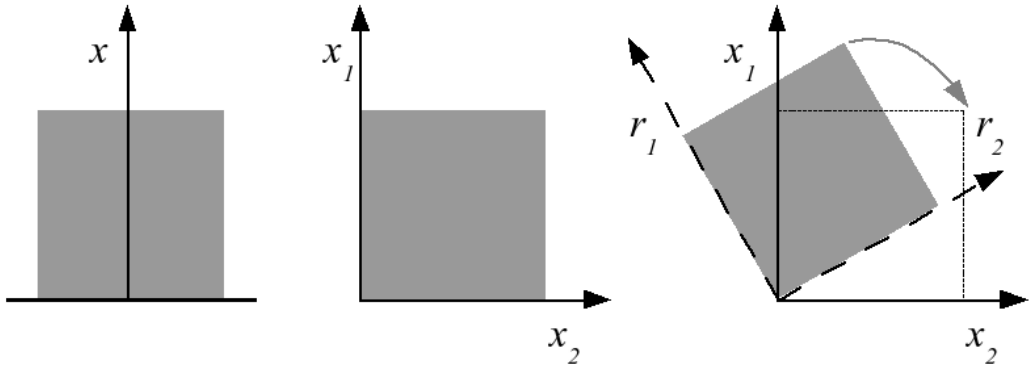
For example, it serves as a framework for popular methods to compute **elementary flux modes** and **extreme pathways.**

Analogy with Computer Graphics problem:

How can one efficiently describe the space in a dark room that is lighted by a torch shining through the open door?



Review: Duality of Matrices



This is the duality

Left: all points above the dividing line (the shaded area) fulfill the condition $x \geq 0$.
Middle: the points in the grey area fulfill the conditions $x_1 \geq 0$ and $x_2 \geq 0$.

But how could we describe the points in the grey area on the right side in a correspondingly simple manner?
Obviously, we could define a new coordinate system (r_1, r_2) as a new set of generating vectors.
But we could also try to transform this area back into the grey area of the middle panel and use the old axes x_1 and x_2.

In 2D, this transformation can be obviously best performed by multiplying all vectors inside the grey area by a two-dimensional **rotation matrix**.

The Double Description Method

A pair (**A**,**R**) of real matrices **A** and **R** is said to be a **double description pair** or simply a ***DD* pair** if the relationship

A x $\geq$ **0** if and only if **x** = **R** λ for some $\lambda \geq$ **0**

holds. The column size of **A** has to be equal to the row size of **R**, say *d*.

For such a pair, the set *P*(**A**) represented by **A** as $P(\mathbf{A}) = \{\mathbf{x} \in \Re^d : \mathbf{Ax} \geq 0\}$

is simultaneously represented by **R** as $\{\mathbf{x} \in \Re^d : \mathbf{x} = \mathbf{R}\boldsymbol{\lambda} \quad \text{for some} \quad \boldsymbol{\lambda} \geq 0\}$

A subset *P* of $\Re^d$ is called **polyhedral cone** if *P* = *P*(**A**) for some matrix **A**, and **A** is called a **representation matrix** of the polyhedral cone *P*(**A**).

Then, we say **R** is a **generating matrix** for *P*.

Each column vector of a generating matrix **R** lies in the cone *P* and every vector in *P* is a nonnegative combination of some columns of **R**.

The Double Description Method

Theorem 1 (Minkowski's Theorem for Polyhedral Cones)
For any $m \times n$ real matrix **A**, there exists some $d \times m$ real matrix **R** such that (**A**,**R**) is a *DD* pair, or in other words, the cone $P(\mathbf{A})$ is generated by **R**.

The theorem states that every polyhedral cone admits a generating matrix.

The nontriviality comes from the fact that the row size of **R** is finite. If we allow an infinite size, there is a trivial generating matrix consisting of all vectors in the cone.

Herrmann Minkowski
1864-1909

Also the converse is true:

Theorem 2 (Weyl's Theorem for Polyhedral Cones)
For any $d \times n$ real matrix **R**, there exists some $m \times d$ real matrix **A** such that (**A**,**R**) is a *DD* pair, or in other words, the set generated by **R** is the cone $P(\mathbf{A})$.

Herrmann Weyl
1885-1955

The Double Description Method

Task: how does one construct a matrix **R** from a given matrix **A**, and the converse?

These two problems are computationally equivalent.
Farkas' Lemma shows that (**A**,**R**) is a *DD* pair if and only if (**R**^{T},**A**^{T}) is a *DD* pair.

A more appropriate formulation of the problem is to require the minimality of **R**:
find a matrix **R** such that no proper submatrix is generating *P*(**A**).

A minimal set of generators is **unique up to positive scaling** when we assume the regularity condition that the cone is **pointed**, i.e. the origin is an extreme point of *P*(**A**).

Geometrically, the columns of a minimal generating matrix are in 1-to-1 correspondence with the **extreme rays** of **P**.

Thus the problem is also known as the **extreme ray enumeration problem**.

No efficient (polynomial) algorithm is known for the general problem.

Double Description Method: primitive form

Suppose that the $m \times d$ matrix **A** is given and let $P(\mathbf{A}) = \{\mathbf{x} : \mathbf{Ax} \geq 0\}$

(This is equivalent to the situation at the beginning of constructing EPs or EFMs where **S** is given.)

The *DD* method is an **incremental** algorithm to construct
a $d \times m$ matrix **R** such that (**A**,**R**) is a *DD* pair.

Let us assume for simplicity that the cone *P*(**A**) is pointed.

Let **K** be a subset of the row indices {1,2,...,*m*} of **A** and
let $\mathbf{A}_\mathbf{K}$ denote the submatrix of **A** consisting of rows indexed by **K**.

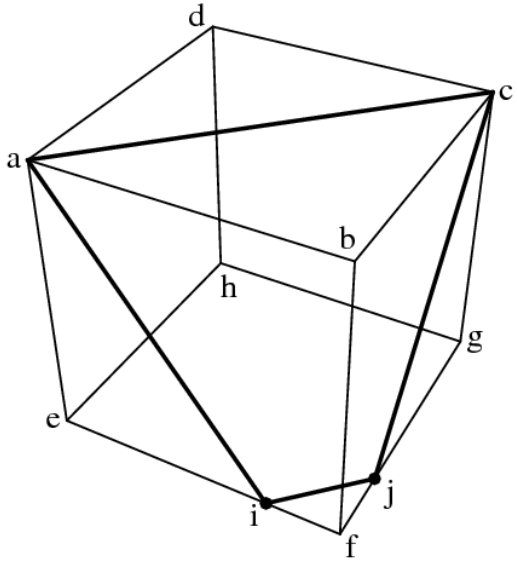
Suppose we already found a generating matrix **R** for $\mathbf{A}_\mathbf{K}$, or equivalently,
($\mathbf{A}_\mathbf{K}$,**R**) is a *DD* pair. If **A** = $\mathbf{A}_\mathbf{K}$, we are done.

Otherwise we select any row index *i* not in **K** and try to construct a *DD* pair
($\mathbf{A}_{\mathbf{K}+\mathbf{i}}$, **R**') using the information of the *DD* pair ($\mathbf{A}_\mathbf{K}$,**R**).

Once this basic procedure is described, we have an algorithm
to construct a generating matrix **R** for *P*(**A**).

Geometric version of iteration step

The procedure can be understood geometrically by looking at the cut-section *C* of the cone $P(\mathbf{A}_K)$ with some appropriate hyperplane *h* in $\Re^d$ which intersects with every extreme ray of $P(\mathbf{A}_K)$ at a single point.



Such a cutsection is illustrated in the Figure.

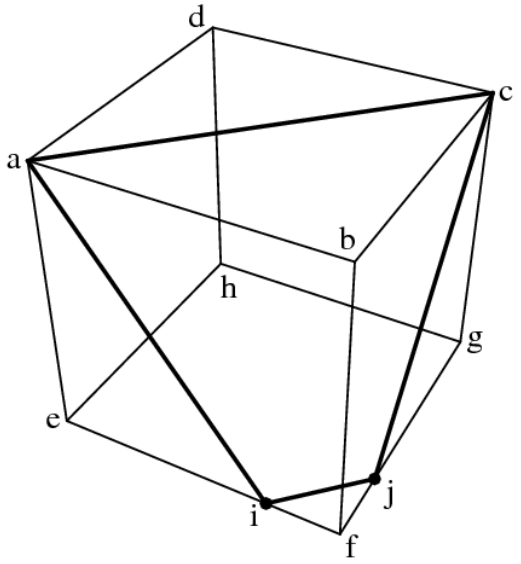
Here, *C* is the cube *abcdefgh.*

Geometric version of iteration step

The newly introduced inequality $\mathbf{A_i} \cdot \mathbf{x} \geq 0$ partitions the space $\Re^d$ into three parts:

$$H_i^+ = \{\mathbf{x} \in \Re^d : \mathbf{A_i} \cdot \mathbf{x} > 0 \}$$
$$H_i^0 = \{\mathbf{x} \in \Re^d : \mathbf{A_i} \cdot \mathbf{x} = 0 \}$$
$$H_i^- = \{\mathbf{x} \in \Re^d : \mathbf{A_i} \cdot \mathbf{x} < 0 \}$$

The intersection of H_i^0 with P and the new extreme points i and j in the cut-section C are shown in bold in the Figure.



Geometric version of iteration step

Let J be the set of column indices of the current generating matrix **R**.

The rays $\mathbf{r_j}$ $(j \in J)$ are then partitioned into three parts accordingly:

$$J^+ = \{j \in J : \mathbf{r_j} \in H_i^+\}$$
$$J^0 = \{j \in J : \mathbf{r_j} \in H_i^0\}$$
$$J^- = \{j \in J : \mathbf{r_j} \in H_i^-\}$$

We will call the rays indexed by J^+, J^0, J^- the **positive, zero, negative** rays with respect to *i,* respectively.

To construct a matrix **R'** from **R**, we generate new $|J^+| \times |J^-|$ rays lying on the *ith* hyperplane H_i^0

- by taking an appropriate positive combination of each positive ray $\mathbf{r_j}$ and each negative ray $\mathbf{r_{j'}}$ and
- by discarding all negative rays.

Geometric version of iteration step

The following lemma ensures that we have a *DD* pair ($\mathbf{A_{K+i}}$,**R**‘), and provides the key procedure for the most primitive version of the *DD* method.

Lemma 3 Let *(*$\mathbf{A_K}$,**R***)* be a *DD* pair and let *i* be a row index of **A** not in **K**.

Then the pair ($\mathbf{A_{K+i}}$,**R**‘*)* is a *DD* pair, where **R**‘ is the $d \times |J'|$ matrix with column vectors $\mathbf{r}_j$ *(*$j \in J'$*)* defined by

$J' = J^+ \cup J^0 \cup (J^+ \times J^-)$, and

$\mathbf{r}_{jj'} = (\mathbf{A}_i \cdot \mathbf{r}_j) \cdot \mathbf{r}_{j'} - (\mathbf{A}_i \cdot \mathbf{r}_{j'}) \cdot \mathbf{r}_j$ for each $(j,j') \in J^+ \times J^-$

Proof omitted.

Finding seed *DD* pair

It is quite simple to find a *DD* pair ($\mathbf{A_K}$,**R**) when |**K**| = 1.
This can serve as the initial *DD* pair.

Another simple (and perhaps the most efficient) way to obtain
an initial *DD* form of *P* is by selecting a maximal submatrix $\mathbf{A_K}$ of **A**
consisting of **linearly independent** rows of **A**.

The vectors $\mathbf{r_j}$'s of matrix **R** are then obtained by solving the system of equations

$$\mathbf{A_K}\,\mathbf{R} = \mathbf{I}$$

where **I** is the identity matrix of size |**K**|.

As we have assumed rank(**A**) = *d,* i.e. $\mathbf{R} = \mathbf{A_K}^{-1}$,
the pair ($\mathbf{A_K}$,**R**) is clearly a *DD* pair,
since $\mathbf{A_K}\cdot\mathbf{x} \geq \mathbf{0} \leftrightarrow \mathbf{x} = \mathbf{A_K}^{-1}\lambda,\ \lambda \geq \mathbf{0}$.

Primitive algorithm for DoubleDescriptionMethod

```
procedure DoubleDescriptionMethod(A);
begin
    Obtain any initial DD pair (A_K, R);
    while K ≠ {1, 2, ..., m} do
    begin
        Select any index i from {1, 2, ..., m} \ K;
        Construct a DD pair (A_{K+i}, R') from (A_K, R);
            /* by using Lemma 3 */
        R := R';    K := K + i;
    end;
    Output R;
end.
```

This algorithm is very primitive.
The straightforward implementation will be quite useless because the size of *J* increases extremely fast.

This is because many vectors $\mathbf{r}_{jj'}$ generated by the algorithm defined in Lemma 3 are unnessary.
We need to avoid generating redundant vectors!

To avoid generating redundant vectors, we will use the zero set or active set Z(**x**) which is the set of inequality indices satisfied by **x** in P(**A**) with equality.

Noting $\mathbf{A}_{i.}$ the ith row of **A**, Z(**x**) = {i : $\mathbf{A}_{i.}\,\mathbf{x}$ = 0}

Towards the standard implementation

Two distinct extreme rays **r** and **r'** of *P* are **adjacent**
if the minimal face of *P* containing both rays contains no other extreme rays.

Proposition 7. Let **r** and **r'** be distinct rays of *P.*

Then the following statements are equivalent
(a) **r** and **r'** are adjacent extreme rays,

(b) **r** and **r'** are extreme rays and the rank of the matrix $\mathbf{A}_{Z(\mathbf{r}) \cap Z(\mathbf{r}')}$ is $d - 2$,

(c) if **r"** is a ray with $Z(\mathbf{r}") \supset Z(\mathbf{r}) \cap Z(\mathbf{r}')$ then either $\mathbf{r}" \simeq \mathbf{r}$ or $\mathbf{r}" \simeq \mathbf{r}'$.

Towards the standard implementation

Lemma 8. Let $(\mathbf{A_K},\mathbf{R})$ be a *DD* pair such that rank($\mathbf{A_K}$) = d and let i be a row index of **A** not in K.

Then the pair ($\mathbf{A_{K+i}}$, $\mathbf{R}'$) is a *DD* pair, where $\mathbf{R}'$ is the $d \times |J'|$ matrix with column vectors $\mathbf{r}_j$ ($j \in J'$) defined by

$J' = J^+ \cup J^0 \cup \text{Adj}$

Adj = $\{(j,j') \in J^+ \times J^- : \mathbf{r}_j$ and $\mathbf{r}_{j'}$ are adjacent in $P(\mathbf{A_K})\}$ and

$\mathbf{r} = (\mathbf{A_i}\,\mathbf{r_j})\,\mathbf{r}_{j'} - (\mathbf{A_i}\mathbf{r_j})\,\mathbf{r_j}$ for each $(j,j') \in$Adj.

Furthermore, if **R** is a minimal generating matrix for $P(\mathbf{A_K})$ then $\mathbf{R}'$ is a minimal generating matrix for $P(\mathbf{A_{K+i}})$.

Algorithm for standard form of double description method

This is now a straightforward variation of the *DD* method which produces a minimal generating set for P:

```
procedure DDMethodStandard(A)
begin
    Obtain any initial DD pair (A_K, R) such that R is minimal;
    while K ≠ {1, 2, ..., m} do
    begin
        Select any index i from {1, 2, ..., m} \ K;
        Construct a DD pair (A_{K+i}, R') from (A_K, R);
            /* by using Lemma 8 */
        R := R';    K := K + i;
    end;
    Output R;
end.
```

To implement `DDMethodStandard`, we must check for each pair of extreme rays **r** and **r'** of $P(\mathbf{A}_K)$ with $\mathbf{A}_i\,\mathbf{r} > 0$ and $\mathbf{A}_i\,\mathbf{r}' < 0$ whether they are adjacent in $P(\mathbf{A}_K)$.

This completes our quick look at the Double Description method.

Summary – Metabolic Networks

Compared to other cellular networks, our understanding of metabolic networks is quite mature. This is due to the almost complete characterization of central metabolism in most organisms and by the ability to perform direct fluxome measurement using e.g. ^{13}C-labelled substrate.

FBA and EP enable us to characterize topological properties of the networks and even make quantitative predictions.

Metabolic network use is highly uneven (power-law distribution) both at the global level and at the level of the individual metabolites.

E. coli responds to changes in growth conditions by reorganizing the rates of selected fluxes predominantly within this high-flux backbone. The use of the other pathways remains unaltered. These reorganizations result in large, discrete changes in the fluxes of the HFB reactions.

Content of final exam (July 27, 2018)

Lecture	Slides relevant for exam
1	17-21
2	1-16, 35-58
3	All
4	20
5	1-20,28-37,47
6	8-33
7	1-10,18-42
8	1-10,14-28
9	18-22,30-37
10	10-23,25-49
11	4-5,9-16,19-25
12	20-23, 31-32

as in V8

as in V16

Lecture	Slides relevant for exam
13	3-9, 34-40
14	9-27
15	5-7, 12-13, 22-23
16	20, 24-25, 27-28, 43-44
17	All (only proofs covered)
18	All (… in lecture)
19	25-41
20	6-10, 13-17, 24-36
21	1-6, 15-26, 30-45 (main idea)
22	1-34 (most likely)
23	45-47 (most likely)
24	1-13 (most likely)
25	-
26	-

Relevant are also the assignments !
(theoretical parts, not the programming parts)