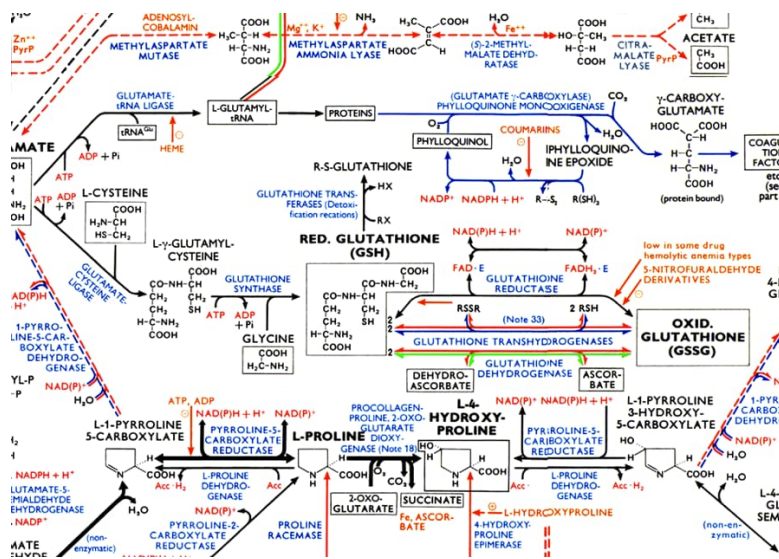


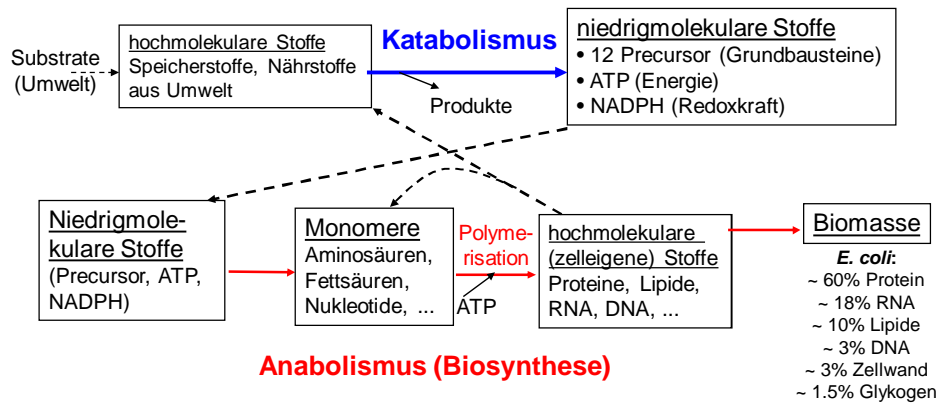
III. Metabolische Netzwerkanalyse

III.1 Einführung: Zellulärer Stoffwechsel und Stoffwechselnetzwerke



Stoffwechsel

- **Stoffwechsel (Metabolismus)** = Gesamtheit aller biochem. Reaktionen in einer Zelle oder einem Organismus, die dem Auf-, Ab- oder Umbau von Nährstoffen bzw. zelleigenen Bestandteilen dienen.
- Stoffwechsel gliedert sich in **Katabolismus** (Abbau + Umbau) und **Anabolismus** (Umbau + Aufbau)



- einige Reaktionen (im Zentralstoffwechsel) haben katabole + anabole Funktionen = **amphibolisch**
- viele zentrale Metabolite (ATP, NAD(P)H) und Stoffwechselwege (z.B. Glykolyse, Zitronensäurezyklus, Pentosephosphatweg) in unterschiedlichen Organismenreichen **konserviert**
- Hauptziel des Stoffwechsels
 - in Mikroorganismen: schnelle Synthese von Biomasse (Reproduktion).
 - in Mehrzellern: spezifisch für Zelltyp (z.B. Muskelzellen, Hautzellen, Leberzellen, Erythrozyten usw.).

Metabolische Netzwerke zeigen Bow-tie-structure (Wdh. aus Kapitel II)

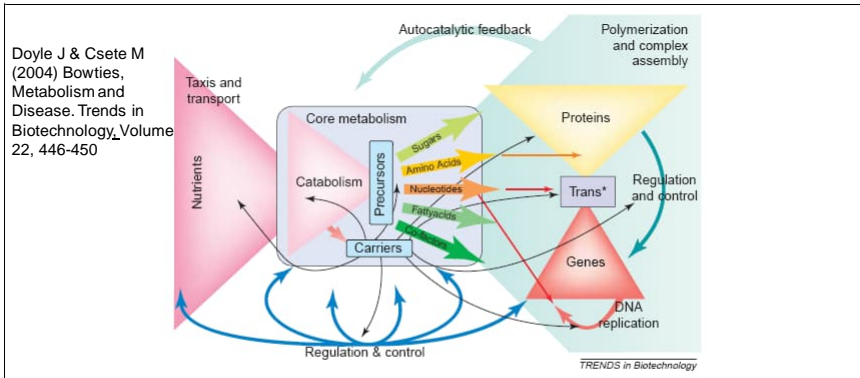
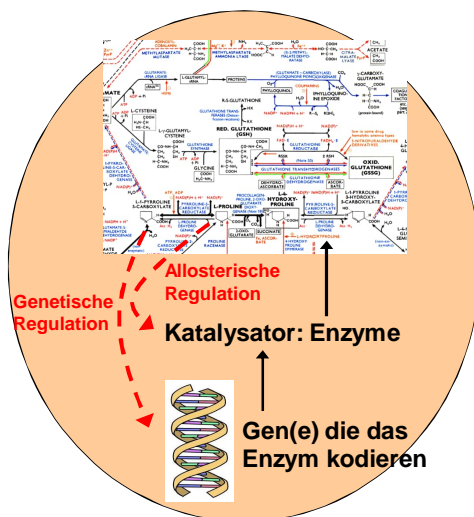


Figure 1. The nested bow-tie architectures of metabolism input a wide range of nutrients and produce a large variety of products and complex macromolecules using a relatively few intermediate common currencies. The common currencies and their enzymes form the knot of the bow tie. The overall bow tie can be decomposed into three principal subsidiary bow ties. One produces the activated carriers, such as ATP, NAD and NADP, that globally supply the cell with energy, reducing power and small moieties. In parallel, catabolism produces a standard group of 12 precursor metabolites, among them glucose 6-phosphate (G6P), fructose 6-phosphate (F6P), phosphoenolpyruvate (PEP), pyruvate (PYR), α -ketoglutarate (AKG) and acetyl-coenzyme A (ACCoA), which are the starting points for the biosynthesis of amino acids, nucleotides, carbohydrates, fatty acids and cofactor building blocks. These building blocks are then used by general-purpose polymerases, particularly in the transcription and translation (trans*) bow tie, to assemble complex macromolecules. This architecture uses selective homogeneity at the knot to facilitate control, organization and management of the enormous heterogeneity in enzyme specificity, action and regulation, and in substrate size, flux and concentration. All modern technologies, from manufacturing to the power grid to the Internet, are organized with bow ties.

- hohe Diversität in Inputs (Substrate) und Outputs (Proteine, Biomasse, ...)
- Zentralstoffwechsel: einheitliche „Protokolle“ und „Währungen“(ATP, NAD(P)H, Precursor) → hohe Flexibilität (plug and play); leicht erweiterbar (evolvierbar); aber: leicht anfällig gegen „Geiselnahme“ dieser Protokolle (z.B. Viren)

Stoffwechsel: molekulare Ebene



Reagierende Spezies: Metabolite.

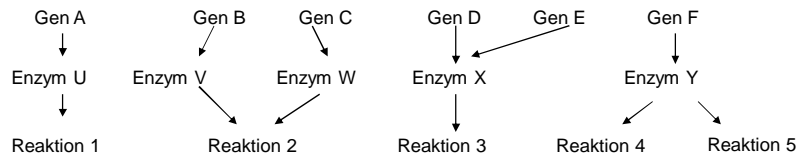
Metabolische Reaktionen:

- Stöchiometrie
- katalysierendes Enzym
- Gen(e), das/die das Enzyme kodieren
- Reversibilität/Thermodynamik
= oft gut bekannt

- allosterische and genetische Regulation
- charakteristische Kinetik

Rekonstruktion von Stoffwechselnetzen

- SN können aus dem sequenzierten Genom rekonstruiert werden (Gen → Enzym → Reaktion)



Enzymw V und W sind Isoenzyme, Enzym X ist Multikomplex und Enzym Y ist multifunktionell

- Für viele Berechnungen benötigt man nur die Reaktionen + Metabolite des Netzwerks. Die Gen → Enzym → Reaktion Abhängigkeiten werden aber für bestimmte Anwendungen oft mitgespeichert.

Stoffwechselnetze in Datenbanken

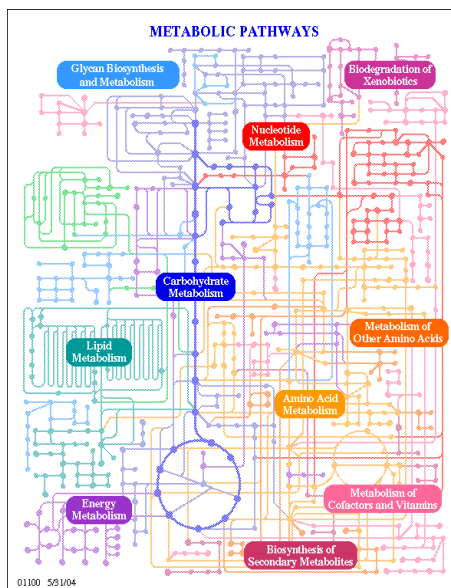
SN sind (z.T. organismenspezifisch) in Datenbanken abgelegt:

- KEGG: www.genome.jp/kegg
8007 Organismen (Stand 2022)
- MetaCyc: www.metacyc.org
- Stöchiometrische Modelle
bigg.ucsd.edu
<http://systemsbiology.ucsd.edu/InSilicoOrganisms/OtherOrganisms>

Weitere relevante Datenbanken:

- Brenda: www.brenda-enzymes.info
(Enzym-Datenbank)
- Equilibrator: Thermodynamische Parameter (z.B. K_{eq}) von Reaktionen:
<https://equilibrator.weizmann.ac.il/>

Reaktionseigenschaften (a)-(d) oft über diese Datenbanken verfügbar; regulatorische und kinetische Eigenschaften (e)-(f) dagegen seltener (einige z.B. in Brenda)



Bsp: KEGG

Genomskalige Stoffwechselnetze

Beispiele für **genomskalige** Netzwerke:

- *E. coli*: Modell iML1515 (2017): 2712 Reaktionen (inkl. Austausch), 1877 Metabolite
- *S. cerevisiae* (*Bäckerhefe*; 2011): 1857 Reaktionen, 1168 Metabolite
- *Homo sapiens* (Recon3D; 2018): 10600 Reaktionen, 5835 Metabolite
→ hohe Komplexität durch Kompartimente und unterschiedliche Zelltypen!

Organism	Strain	Organism Genes	Version	Model Genes	Metabolites	Reactions	Compartments	Reference	Journal	Date	
1	<i>Acinetobacter baumannii</i>	AVE	3760	abvA9ELR91	650	776	891	2 (c,e)	Kim et al.	Mol Biosyst	2010/01/23
2	<i>Acinetobacter baylyi</i>	ADP1	3287	lAbaylyV4	774	701	875	3 (c,e,p)	Duret et al.	BMC Syst Biol	2008/10/09
3	<i>Amycolatopsis balmycitra</i>		8585		983	647	1363		Vongrangnak et al.	Biotechnol Bioeng	2012/01/19
4	<i>Bacillus megaterium</i>	W5H002			1055	993	1112		Zou et al.	J Biotechnol	2013/03/05
5	<i>Bacillus subtilis</i>	168	4114	iBsu103	1103	1138	1437	2 (c,e)	Henry et al.	Genome Biol	2009/06/27
6	<i>Bacillus subtilis</i>		4114	modL_v3	844	988	1020	2 (c,e)	Oh et al.	J Biol Chem	2007/06/19
7	<i>Blattabacterium cussenoti</i>	Bpe		KG238	364	238	296		Gonzalez-Domenach et al.	BMC Microbiol	2012/03/02
8	<i>Blattabacterium cussenoti</i>	Pam		KG230	358	230	289		Gonzalez-Domenach et al.	BMC Microbiol	2012/03/02
9	<i>Bordetella pertussis</i>	10536							Budman et al.	Biotechnol Prog	2012/12/12
10	<i>Buchnera sphidicola</i>	AP5	574	IGT196	196	240	263	2 (c,e)	Thomas et al.	BMC Syst Biol	2009/02/24
11	<i>Burkholderia</i>	0215		WF028	1028	824	859		Fara et al.	BMC Syst Biol	2011/05/26

<http://systemsbiology.ucsd.edu/InSilicoOrganisms/OtherOrganisms>

Metabolische (Stöchiometrische) Netzwerkanalyse: untersucht strukturelle und funktionale Eigenschaften von SN auf der Basis von topologischen Informationen, die meistens relativ leicht zugänglich sind (d.h. kinetische Parameter werden nicht berücksichtigt, sondern nur Reaktionseigenschaften (a)-(c))

Untersuchte Aspekte:

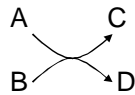
- Konsistenz (nicht verbundene Netzbereiche, dead-ends, ...)
- globaler Aufbau (Konnektivität, Clusterung, bow-tie structure; *siehe Kapitel II*)
- strukturelle Kopplungen (z.B. wenn Enzym 1 aktiv, dann muß auch Enzym 2 aktiv sein)
- funktionale Einheiten (z.B. Pathways)
- Operationsweisen (wie kann aus Substrat S Produkt P synthetisiert werden)
- Performance (z.B. optimale Aubeute)
- Phänotyp-Prädiktionen nach Löschen von Reaktionen / Genen
- Identifikation von Eingriffspunkten (z.B. für Metabolic Engineering und Arzneimittelforschung)

III.2 Hypergraphen und Stöchiometrische Matrix

- Stoffwechselnetze (SN) bestehen aus
Metaboliten = Knoten
biochemischen Reaktionen = (Hyper-) Kanten
- SN sind i.A. keine Graphen sondern **Hypergraphen**

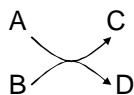
Graph: $A \rightarrow B$ Kanten verbinden immer einen Start- mit einem Endknoten
(nur geeignet für unimolekulare Reaktionen)

Hypergraph: $A + B \rightarrow C + D$ Hyperkante kann beliebig viele Startknoten (Edukte) mit beliebig vielen Endknoten (Produkte) verknüpfen (UND-Verknüpfung)

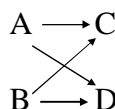


Hypergraphen können in Graphen überführt werden,
z.B. in Substrat-Graphen oder bipartiten Graphen.

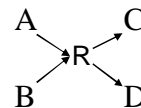
Hypergraph



Substrat-Graph



Bipartiter Graph



- 2 Mengen von Knoten: Metabolite und Reaktionen
- Kanten: entweder von Metabolite (Edukte) zu Reaktionen oder von Reaktionen zu Metabolite (Produkte)

- Überführung Hypergraph \rightarrow Substratgraph eindeutig aber nicht eineindeutig
- Graphen können zur globalen Analyse der Architektur von SN herangezogen werden (siehe Kapitel II) aber weniger für funktionelle Fragestellungen. Beispiel oben: Was passiert, wenn A nicht verfügbar ist? Im Hypergraphen können C und D nicht mehr gebildet werden (Hyperkante zerstört), aber im Substratgraphen existieren weiterhin Kanten $B \rightarrow C$ und $B \rightarrow D$ und im bipartiten Graphen die Pfade $B \rightarrow R \rightarrow C$ und $B \rightarrow R \rightarrow D$.

Hypergraphen

- Hypergraph: verallgemeinerte Graphen;
Menge V von Knoten und Menge E von Hyperkanten: $H=(V,E)$
- Hyperkante (hyperedge): verknüpft beliebig viele Knoten
(Graphen sind spezielle Hypergraphen, wo alle Hyperkanten genau 2 Knoten enthalten (und dann Kanten genannt werden))

Ungerichtete Hypergraphen

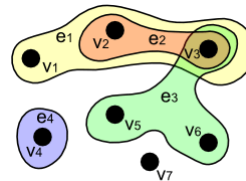
- die *Hyperkanten* sind beliebige nicht-leere Teilmengen der Knoten

Beispiel: für ungerichteten Hypergraphen

$$V = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7\},$$

$$E = \{e_1, e_2, e_3, e_4\} =$$

$$\{\{v_1, v_2, v_3\}, \{v_2, v_3\}, \{v_3, v_5, v_6\}, \{v_4\}\}$$



- ungerichtete Hypergraphen können auch als Mengensysteme verstanden werden (V ist die Grundmenge und E ist eine Familie von Teilmengen aus V)

Gerichtete Hypergraphen

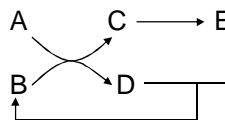
- jede Hyperkante e (hier: *hyperarc*) ist unterteilt in 2 disjunkte Teilmengen, dem Schwanz S (*tail*; „Startknoten“) und dem Kopf K (*head*, „Endknoten“):
 $e=(S,K), S,K \subseteq V$

Beispiel gerichteter Hypergraph:

$$V = \{A, B, C, D, E\},$$

$$E = \{e_1, e_2, e_3\} =$$

$$\{\{A, B\}, \{C, D\}, \{\{C\}, \{E\}\}, \{E, D\}, \{B\}\}$$



Speicherung über Inzidenzmatrix
ähnlich wie im Graphen (-1 für Start und +1 für Endknoten; im ungerichteten Hypergraphen nur +1)

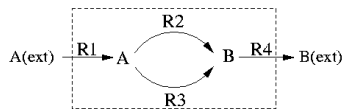
$$I = \begin{matrix} & e_1 & e_2 & e_3 \\ \begin{matrix} A \\ B \\ C \\ D \\ E \end{matrix} & \begin{pmatrix} -1 & 0 & 0 \\ -1 & 0 & 1 \\ 1 & -1 & 0 \\ 1 & 0 & -1 \\ 0 & 1 & -1 \end{pmatrix} \end{matrix}$$

- stöchiometrische (metabolische) Netzwerke sind gerichtete Hypergraphen, wo die Knoten in jeder Hyperkante noch mit (stöchiometrischen) Koeffizienten gewichtet ist, z.B. $2A + 3B \rightarrow 2C$
- man kann diese Koeffizienten (anstelle von -1/+1) bequem in der Inzidenzmatrix eintragen, was dann die stöchiometrische Matrix ergibt

Stöchiometrische Matrix

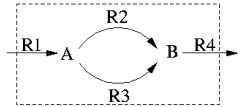
- Struktur eines SN (oder eines beliebigen Reaktionsnetzwerkes) lässt sich mit der **stöchiometrischen Matrix N** erfassen
- **N** ist $m \times q$ Matrix; m : Anzahl Metabolite (Zeilen); q : Anzahl Reaktionen (Spalten)
- Element N_{ij} enthält stöch. Koeffizient des Metaboliten i in Reaktion j

Bsp: SN1



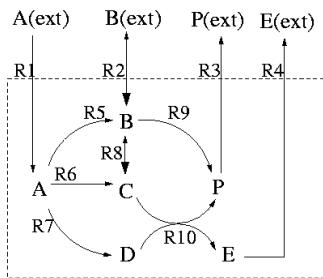
$$\mathbf{N} = \begin{matrix} & \begin{matrix} \mathbf{R1} & \mathbf{R2} & \mathbf{R3} & \mathbf{R4} \end{matrix} \\ \begin{matrix} \rightarrow \mathbf{A} \\ \rightarrow \mathbf{B} \\ \rightarrow \mathbf{Aext} \\ \rightarrow \mathbf{Bext} \end{matrix} \\ \begin{pmatrix} 1 & -1 & -1 & 0 \\ 0 & 1 & 1 & -1 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{matrix}$$

- oft werden **externe Metabolite** (Senken/Quellen in der Umwelt) in **N** nicht explizit berücksichtigt



$$\mathbf{N} = \begin{matrix} & \begin{matrix} \mathbf{R1} & \mathbf{R2} & \mathbf{R3} & \mathbf{R4} \end{matrix} \\ \begin{matrix} \rightarrow \mathbf{A} \\ \rightarrow \mathbf{B} \end{matrix} \\ \begin{pmatrix} 1 & -1 & -1 & 0 \\ 0 & 1 & 1 & -1 \end{pmatrix} \end{matrix}$$

- größeres Beispiel: SN2



$$\mathbf{N} = \begin{matrix} & \begin{matrix} \mathbf{R1} & \mathbf{R2} & \mathbf{R3} & \mathbf{R4} & \mathbf{R5} & \mathbf{R6} & \mathbf{R7} & \mathbf{R8} & \mathbf{R9} & \mathbf{R10} \end{matrix} \\ \begin{matrix} \leftarrow \mathbf{A} \\ \leftarrow \mathbf{B} \\ \leftarrow \mathbf{C} \\ \leftarrow \mathbf{D} \\ \leftarrow \mathbf{E} \\ \leftarrow \mathbf{P} \end{matrix} \\ \begin{pmatrix} 1 & 0 & 0 & 0 & -1 & -1 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & -1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} \end{matrix}$$

$$\begin{aligned}
 \mathit{rev} &= \{\mathbf{R2}, \mathbf{R8}\} \\
 \mathit{irrev} &= \{\mathbf{R1}, \mathbf{R3}, \mathbf{R4}, \mathbf{R5}, \mathbf{R6}, \mathbf{R7}, \mathbf{R9}, \mathbf{R10}\}
 \end{aligned}$$

- Informationen zur Reversibilität der Reaktionen (reversibel / irreversibel) können separate (z.B. in Liste) abgespeichert werden

Stöchiometrische Matrix **N**:

- ist gewöhnlich schwach besetzt (viele Nullen)
- enthält oft "Pseudoreaktionen", wie Aufnahme/Ausscheidung und Wachstum:
→ kumulierte Stöchiometrie aller Precursor+Cofaktoren für 1 gDW)
- fundamental für Strukturanalysen → Methoden aus der linearen Algebra
- ist auch fundamental für dynamische Modellierung von Reaktionsnetzen:

$$\frac{d\mathbf{c}(t)}{dt} = \mathbf{N} \cdot \mathbf{r}(t)$$

$\mathbf{c}(t)$: Metabolitkonzentrationen zur Zeit t [mmol/gDW]

$d\mathbf{c}(t) / dt$: Zeitableitung von $\mathbf{c}(t)$

$\mathbf{r}(t)$: (netto) Reaktionsraten [mmol/(gDW · h)]

$$\mathbf{r}(t) = f(\mathbf{c}(t), \mathbf{P}, t) \longrightarrow \text{Kinetik (benötigt Funktion } f \text{ und Parametersatz } \mathbf{P})$$

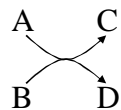
Bsp: Pyruvatkinase
(hier: x statt c)

$$f_{E,PykF} = \frac{x_{PykF} p_{PykF,cat} \frac{x_{PEP}}{p_{PykF,K_{PEP}}} \left(1 + \frac{x_{PEP}}{p_{PykF,K_{PEP}}}\right)^{p_{PykF,n}-1}}{\left(1 + \frac{x_{PEP}}{p_{PykF,K_{PEP}}}\right)^{p_{PykF,n}} + p_{PykF,L} \left(1 + \frac{x_{PEP}}{p_{PykF,K_{PEP}}}\right)^{-p_{PykF,n}}}$$

III.3. Erhaltungsrelationen (ER)

- ER sind *gewichtete Summen von Metabolitkonzentrationen*, die zu jedem Zeitpunkt im System *konstant* sind (unabhängig von kinetischen Parametern)

Beispiel



$$\mathbf{N} = \begin{pmatrix} -1 & 0 \\ -1 & 0 \\ 1 & 0 \\ 1 & 0 \end{pmatrix}$$

Erhaltungsrelationen:

$$\begin{aligned} [A] + [C] &= S1 = \text{CONST.} \\ [B] + [D] &= S2 = \text{CONST.} \\ [A] - [B] &= S3 = \text{CONST.} \\ [A] + 2[C] - [D] &= S4 = \text{CONST.} \\ &\dots \end{aligned}$$

Typische Beispiele in stöch. Modellen biochemischer Netzwerke:

$$\begin{aligned} [\text{NADH}] + [\text{NAD}] &= \text{CONST.} \\ [\text{ATP}] + [\text{ADP}] &= \text{CONST.} \end{aligned}$$

- Herleitung: jede ER ist Vector \mathbf{y} (enthält die Gewichte für jeden Metaboliten)

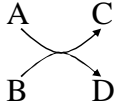
• \mathbf{y} erfüllt: $\mathbf{y}^T \mathbf{c}(t) = \text{CONST}$ (für alle t) $\rightarrow \mathbf{y}^T \mathbf{c}(t)' = \mathbf{y}^T \mathbf{N} \mathbf{r}(t) = 0 \rightarrow \mathbf{y}^T \mathbf{N} = \mathbf{0}^T$

$\mathbf{c}(t)' = \mathbf{N} \cdot \mathbf{r}(t)$ für alle $\mathbf{r}!$

$$\boxed{\mathbf{y}^T \mathbf{N} = \mathbf{0}^T}$$

$$\Leftrightarrow \mathbf{N}^T \mathbf{y} = \mathbf{0}$$

- \rightarrow jede ER \mathbf{y} korrespondiert mit linear abhängigen Zeilen in \mathbf{N}
- \rightarrow alle ER \mathbf{y} liegen im linken Nullraum von \mathbf{N} (dim: $m - \text{rang}(\mathbf{N})$) (= im rechten Nullraum (=Kern) von \mathbf{N}^T)
- \rightarrow suche Basisvektoren für Kern von \mathbf{N}^T (angeordnet in Matrix \mathbf{Y})
MATLAB: $\mathbf{Y} = \text{null}(\mathbf{N}')$
- \rightarrow jede ER \mathbf{y} ist dann Linearkomb. der Spalten in \mathbf{Y} : $\mathbf{y} = \mathbf{Y} \mathbf{b}$



$\mathbf{N}^T \mathbf{y} = (-1 \quad -1 \quad 1 \quad 1) \mathbf{y} = \mathbf{0}$ $\xrightarrow{(4-1)=3 \text{ unabhängige Lösungen } \mathbf{y}_i}$ $\mathbf{Y} = \begin{pmatrix} \mathbf{y}_1 & \mathbf{y}_2 & \mathbf{y}_3 \\ 1 & 1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{matrix} A \\ B \\ C \\ D \end{matrix}$

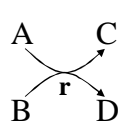
- ER reduzieren das mögliche dynamische Verhalten!

$$\mathbf{Y}^T \mathbf{c}(t) = \text{CONST} \text{ (für alle } t \text{)}$$

- die Konstanten CONST ergeben sich aus Anfangskonzentrationen

$$\text{CONST} = \mathbf{Y}^T \mathbf{c}(0)$$

- man kann diese Information zur Modellreduktion verwenden: modelliere nur $\text{Rang}(\mathbf{N})$ unabhängige Konzentrationen dynamisch; die anderen algebraisch



$$\mathbf{Y} = \begin{pmatrix} \mathbf{y}_1 & \mathbf{y}_2 & \mathbf{y}_3 \\ 1 & 1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{matrix} A \\ B \\ C \\ D \end{matrix}$$

Modelliere nur A explizit

$$\boxed{c_A(t)' = \mathbf{N}_A \mathbf{r}(t) = -\mathbf{r}(\mathbf{c}, \mathbf{p}, t)}$$

\rightarrow Integration ergibt $c_A(t)$

B, C und D ergeben sich algebraisch aus $\mathbf{Y}^T \mathbf{c}(t) = \text{CONST}$

$$\mathbf{N} = \begin{pmatrix} -1 \\ -1 \\ 1 \\ 1 \end{pmatrix}$$

$$\text{CONST}_1 = \mathbf{c}_A(0) - \mathbf{c}_B(0)$$

$$\text{CONST}_2 = \mathbf{c}_A(0) + \mathbf{c}_C(0)$$

$$\text{CONST}_3 = \mathbf{c}_B(0) + \mathbf{c}_D(0)$$

$$\begin{pmatrix} c_B(t) \\ c_C(t) \\ c_D(t) \end{pmatrix} = \begin{pmatrix} c_A(t) - \text{CONST}_1 \\ \text{CONST}_2 - c_A(t) \\ \text{CONST}_3 + \text{CONST}_1 - c_A(t) \end{pmatrix}$$

Weiteres Beispiel für Erhaltungsrelationen:

Netzwerk mit 4 Metaboliten und 2 Reaktionen: $D \rightarrow A$ und $2A + B \rightarrow C$

Zum Zeitpunkt t_0 seien die Konzentration $c_A=c_B=c_C=c_D=2$ [beliebige Einheit].

Zu einem späteren Zeitpunkt messen 2 Experimentatoren unabhängig voneinander die Konzentration noch einmal:

Exp1: $c_A=1$; $c_B=1$; $c_C=3$; $c_D=1$; Exp2: $c_A=3$; $c_B=3$; $c_C=1$; $c_D=1$

Wer von beiden muss falsch liegen?

(1) Stöchiometrische Matrix

$$N = \begin{pmatrix} 1 & -2 \\ 0 & -1 \\ 0 & 1 \\ -1 & 0 \end{pmatrix}$$



(2) Erhaltungsrelationen

$$N^T y = \begin{pmatrix} 1 & 0 & 0 & -1 \\ -2 & -1 & 1 & 0 \end{pmatrix} y = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Gaussches Verfahren

$$N^T y = \begin{pmatrix} 1 & 0 & 0 & -1 \\ 0 & -1 & 1 & -2 \end{pmatrix} y = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Lösen der Gleichung; Freiheitsgrade einsetzen

1. Zeile (Gleichung): Ansatz: $y_4=C_1 \rightarrow y_1=C_1$
2. Zeile (Gleichung): Ansatz: $y_2=C_2 \rightarrow y_3=C_2+2C_1$

(3) Test

t_0 : $\rightarrow y^1: 2+0+2*2+2=8$; $y^2: 0+2+2+0=4$
 Exp1: $\rightarrow y^1: 1+0+2*3+1=8$; $y^2: 0+1+3+0=4$
 Exp2: $\rightarrow y^1: 3+0+2*1+1=6$; $y^2: 0+3+1+0=4$

Experimentator 2 liegt falsch, Da y^1 nicht konsistent mit y^1 von t_0

$$y = \begin{pmatrix} C_1 \\ C_2 \\ C_2+2C_1 \\ C_1 \end{pmatrix} = C_1 \begin{pmatrix} 1 \\ 0 \\ 2 \\ 1 \end{pmatrix} + C_2 \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \end{pmatrix} \rightarrow Y = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 2 & 1 \\ 1 & 0 \end{pmatrix} \begin{matrix} y^1 \\ y^2 \end{matrix}$$

III.4 Stationäre Flussverteilungen (SFV)

- Stoffwechsel operiert oft nahezu im Fließgleichgewicht (quasi-steady state) (Grund: hohe Umsatzraten, besonders im Zentralstoffwechsel)

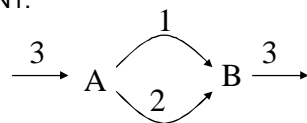
$$\frac{dc(t)}{dt} = N \cdot r(t) \approx 0$$

- fasst man Reaktionsraten als unabhängige Variable auf, dann folgt:

$$N \cdot r = 0 \quad \text{Homogenes lineares Gleichungssystem}$$

- für jeden Metaboliten gilt: Verbrauch und Synthese halten sich die Waage (analog zum 1. Kirchhoffschen Gesetz in Elektrotechnik (Knotenregel))
- erfüllt r obige Gleichung wird es als *stationäre Flussverteilung* (SFV) bezeichnet

Bsp. SN1:

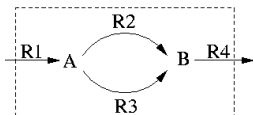


$$N r = \begin{pmatrix} 1 & -1 & -1 & 0 \\ 0 & 1 & 1 & -1 \end{pmatrix} \begin{pmatrix} 3 \\ 1 \\ 2 \\ 3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

III.4.1 Nullraumanalyse

$$\mathbf{N} \cdot \mathbf{r} = \mathbf{0}$$

- jede SFV \mathbf{r} liegt im rechten Nullraum (Kern) von \mathbf{N} : $\ker(\mathbf{N}) = \{\mathbf{r} : \mathbf{N}\mathbf{r} = \mathbf{0}\}$
- $\dim(\ker(\mathbf{N})) = q - \text{Rang}(\mathbf{N})$; $q = \text{Anzahl Spalten (Reaktionen)}$
- suche $q - \text{Rang}(\mathbf{N})$ Basisvektoren für Kern (z.B. über Gaußschen Algorithmus oder MATLAB-Befehl *null*) und ordne sie in Kernmatrix \mathbf{K} an

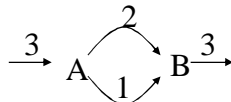


$$\mathbf{N} = \begin{pmatrix} 1 & -1 & -1 & 0 \\ 0 & 1 & 1 & -1 \end{pmatrix} \rightarrow \mathbf{K} = \begin{pmatrix} 1 & 2 \\ 1 & 1 \\ 0 & 1 \\ 1 & 2 \end{pmatrix}$$

$q - \text{rank}(\mathbf{N}) = 4 - 2 = 2$

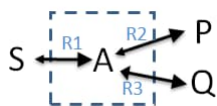


- für jede SFV \mathbf{r} existiert (genau) ein Vektor \mathbf{a} so daß: $\mathbf{r} = \mathbf{K}\mathbf{a}$



Mit Kernmatrix \mathbf{K} von oben ergibt sich: $\mathbf{a} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$

Nullraum: Weiteres Beispiel



$\mathbf{N} = (1 \ -1 \ -1)$
(nur ein interner Metabolit: A)

Steady state: $\mathbf{N}\mathbf{r} = \mathbf{0}$

Lösungsraum der steady-state Flussvektoren \mathbf{r} :

→ Nullraum (Kern) von \mathbf{N}

$$\dim(\ker(\mathbf{N})) = \# \text{reac} - \text{rank}(\mathbf{N})$$

$$2 = 3 - 1$$

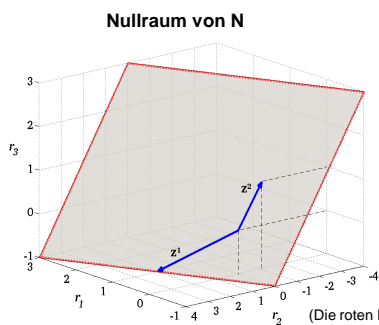
Zwei (nicht eindeutige) Basisvektoren zum Aufspannen des 2-dim. Nullraums (= Lösungsraum der steady-state Flussverteilungen) benötigt.

Eine mögliche Basis:

$$\mathbf{z}^1 = \begin{pmatrix} 1 \\ 2 \\ -1 \end{pmatrix}, \quad \mathbf{z}^2 = \begin{pmatrix} 0 \\ -1 \\ 1 \end{pmatrix}$$

Resultierende Kernmatrix:

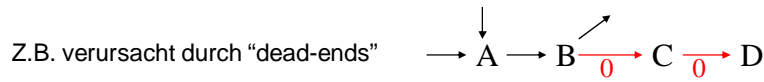
$$\mathbf{K} = \begin{pmatrix} 1 & 0 \\ 2 & -1 \\ -1 & 1 \end{pmatrix} \begin{matrix} \leftarrow r_1 \\ \leftarrow r_2 \\ \leftarrow r_3 \end{matrix}$$



(Die roten Ränder deuten an, dass der (Null)Raum in all seinen Richtungen offen (unbeschränkt) ist).

Analyse der Kernmatrix ermöglicht Identifikation einiger wichtiger struktureller Eigenschaften (mit wenig Aufwand):

(i) Blockierte Reaktionen: Reaktionen deren Rate im steady state immer Null ist



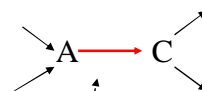
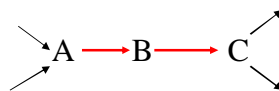
Detektion blockierter Reaktionen: Korrespondierende Zeile einer blockierten Reaktion in \mathbf{K} ist eine Nullzeile (Erinnere: $\mathbf{r} = \mathbf{K}\mathbf{a}$) $\mathbf{K} = \begin{pmatrix} \dots & \dots & \dots \\ 0 & \dots & 0 \\ \dots & \dots & \dots \end{pmatrix} \rightarrow \text{blockiert}$

Anwendungen: strukturelle Inkonsistenzen finden + Modellreduktion

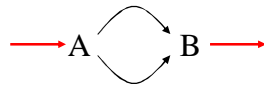
(ii) Gekoppelte Reaktionen: Teilmengen von Reaktionen die immer gemeinsam und immer im gleichen Verhältnis ihrer Reaktionsraten im steady state operieren

Beispiele:

unverzweigte Ketten



... und andere



Identifikation:

Die korrespondierenden Zeilen gekoppelter Reaktionen in der Kernmatrix \mathbf{K} unterscheiden sich nur um skalaren Faktor

$$\mathbf{K} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \\ 1 & 1 \\ 1 & 0 \end{pmatrix}$$

Anwendungen:

- Modellreduktion (Zusammenlegen gekoppelter Reaktionen)
- oft gemeinsam reguliert

III.4.2 Metabolische Flussanalyse

Bestimmung der intrazellulären Flussverteilung im steady-state Experiment

Bilanzgleichung $\mathbf{Nr} = \mathbf{0}$ ist gewöhnlich unterbestimmt
(Nullraum mit unendlich vielen Lösungen)

→ einige Raten müssen gemessen werden
typischerweise Aufnahme/Ausscheidungsdaten im Experiment



→ Partitioniere \mathbf{N} and \mathbf{r} in bekannte (b) und unbekannte (u) Teile:

$$\mathbf{0} = \mathbf{Nr} = \mathbf{N}_b \mathbf{r}_b + \mathbf{N}_u \mathbf{r}_u$$

$$\Leftrightarrow \mathbf{N}_u \mathbf{r}_u = -\mathbf{N}_b \mathbf{r}_b$$

• wenn p Messungen,
dann $q-p=x$ unbekannte Raten
(x = Anzahl Elemente in \mathbf{r}_u)

Flussanalyse: $\mathbf{N}_u \mathbf{r}_u = -\mathbf{N}_b \mathbf{r}_b$ Inhomogenes GLS; mit x Unbekannten in \mathbf{r}_u

Allgemeine Kleinste-Quadrate-Lösung: $\mathbf{r}_u = -\mathbf{N}_u^\# \mathbf{N}_b \mathbf{r}_b + \mathbf{K}_u \mathbf{a}$

$\mathbf{N}_u^\#$: Penrose Pseudo-Inverse (existiert für jede Matrix)

\mathbf{K}_u : Kernmatrix von \mathbf{N}_u - zeigt die Freiheitsgrade im System an

\mathbf{a} = beliebiger Vektor

• Klassifizierung eines Szenarios bzgl. zweier Kriterien (über Rang von \mathbf{N}_u)

Bestimmtheit:

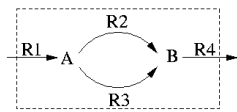
- *bestimmt* (alle Raten eindeutig berechenbar): $\text{Rang}(\mathbf{N}_u) = x$
- *unterbestimmt* (nicht alle Raten berechenbar): $\text{Rang}(\mathbf{N}_u) < x$

Redundanz

- *nicht redundant*: $\text{Rang}(\mathbf{N}_u) = m$ (m : Anzahl Metabolite (Zeilen))
- *redundant* (Inkonsistenzen können auftreten!): $\text{Rang}(\mathbf{N}_u) < m$

→ insgesamt $2 \times 2 = 4$ Fälle; weiteres Vorgehen hängt von Klassifizierung ab

Bsp: SN1



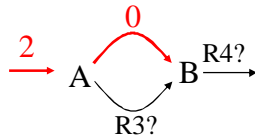
$$\mathbf{N} = \begin{matrix} \text{R1} & \text{R2} & \text{R3} & \text{R4} \\ \begin{pmatrix} 1 & -1 & -1 & 0 \\ 0 & 1 & 1 & -1 \end{pmatrix} \end{matrix} \begin{matrix} \rightarrow \text{A} \\ \rightarrow \text{B} \end{matrix}$$

$$\mathbf{N}_u \mathbf{r}_u = -\mathbf{N}_b \mathbf{r}_b$$

$$\mathbf{r}_u = -\mathbf{N}_u^\# \mathbf{N}_b \mathbf{r}_b + \mathbf{K}_u \mathbf{a}$$

Fall 1: bestimmt, nicht redundant

(in rot: gemessen)



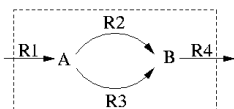
$$\mathbf{r}_b = \begin{pmatrix} \text{R1} \\ \text{R2} \end{pmatrix} = \begin{pmatrix} 2 \\ 0 \end{pmatrix}; \quad x=2; \quad \mathbf{N}_b = \begin{pmatrix} \text{R1} & \text{R2} \\ 1 & -1 \\ 0 & 1 \end{pmatrix}, \quad \mathbf{N}_u = \begin{pmatrix} \text{R3} & \text{R4} \\ -1 & 0 \\ 1 & -1 \end{pmatrix};$$

Rang(\mathbf{N}_u)=x=m=2 → bestimmt, nicht redundant

In bestimmten/nichtredundanten Systemen ist Pseudo-Inverse mit "normaler" Inversen identisch

$$\mathbf{K}_u = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad \mathbf{N}_u^\# = \mathbf{N}_u^{-1} = \begin{pmatrix} -1 & 0 \\ -1 & -1 \end{pmatrix};$$

$$\mathbf{r}_u = \begin{pmatrix} \text{R3} \\ \text{R4} \end{pmatrix} = -\mathbf{N}_u^\# \mathbf{N}_b \mathbf{r}_b + \mathbf{K}_u \mathbf{a} = \begin{pmatrix} 2 \\ 2 \end{pmatrix}$$



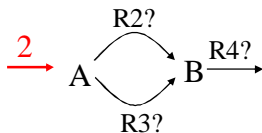
$$\mathbf{N} = \begin{matrix} \text{R1} & \text{R2} & \text{R3} & \text{R4} \\ \begin{pmatrix} 1 & -1 & -1 & 0 \\ 0 & 1 & 1 & -1 \end{pmatrix} \end{matrix} \begin{matrix} \rightarrow \text{A} \\ \rightarrow \text{B} \end{matrix}$$

$$\mathbf{N}_u \mathbf{r}_u = -\mathbf{N}_b \mathbf{r}_b$$

$$\mathbf{r}_u = -\mathbf{N}_u^\# \mathbf{N}_b \mathbf{r}_b + \mathbf{K}_u \mathbf{a}$$

Fall 2: unterbestimmt, nicht redundant

(in rot: gemessen)



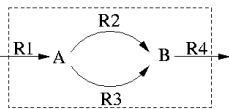
$$\mathbf{r}_b = (\text{R1}) = (2); \quad x=3; \quad \mathbf{N}_b = \begin{pmatrix} \text{R1} \\ 1 \\ 0 \end{pmatrix}, \quad \mathbf{N}_u = \begin{pmatrix} \text{R2} & \text{R3} & \text{R4} \\ -1 & -1 & 0 \\ 1 & 1 & -1 \end{pmatrix};$$

Rang(\mathbf{N}_u)=2=m<x=3 → unterbestimmt, nicht redundant

$$\mathbf{K}_u = \begin{pmatrix} 1 \\ -1 \\ 0 \end{pmatrix}, \quad \mathbf{N}_u^\# = \begin{pmatrix} -0.5 & 0 \\ -0.5 & 0 \\ -1 & -1 \end{pmatrix};$$

Raten sind auch im unterbestimmten System determiniert, falls sie in \mathbf{K}_u eine Nullzeile aufweisen. Hier folgt: R4=2 (und R2 und R3 unbestimmt)

$$\mathbf{r}_u = \begin{pmatrix} \text{R2} \\ \text{R3} \\ \text{R4} \end{pmatrix} = -\mathbf{N}_u^\# \mathbf{N}_b \mathbf{r}_b + \mathbf{K}_u \mathbf{a} = \begin{pmatrix} 1 \\ 1 \\ 2 \end{pmatrix} + \begin{pmatrix} 1 \\ -1 \\ 0 \end{pmatrix} a$$



$$\mathbf{N} = \begin{pmatrix} \text{R1} & \text{R2} & \text{R3} & \text{R4} \\ 1 & -1 & -1 & 0 \\ 0 & 1 & 1 & -1 \end{pmatrix} \begin{matrix} \rightarrow \text{A} \\ \rightarrow \text{B} \end{matrix}$$

$$\mathbf{N}_u \mathbf{r}_u = -\mathbf{N}_b \mathbf{r}_b$$

$$\mathbf{r}_u = -\mathbf{N}_u^\# \mathbf{N}_b \mathbf{r}_b + \mathbf{K}_u \mathbf{a}$$

Fall 3: bestimmt, redundant (= "überbestimmt")

(in rot: gemessen)



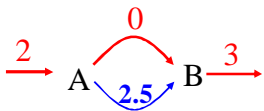
$$\mathbf{r}_b = \begin{pmatrix} \text{R1} \\ \text{R2} \\ \text{R4} \end{pmatrix} = \begin{pmatrix} 2 \\ 0 \\ 3 \end{pmatrix}; \quad x=1; \quad \mathbf{N}_b = \begin{pmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \end{pmatrix}; \quad \mathbf{N}_u = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$$

$\text{Rang}(\mathbf{N}_u)=1=x < m=2 \rightarrow$ bestimmt, redundant

$$\mathbf{K}_u = (0), \quad \mathbf{N}_u^\# = (-0.5 \quad 0.5);$$

$$\mathbf{r}_u = (\text{R3}) = -\mathbf{N}_u^\# \mathbf{N}_b \mathbf{r}_b + \mathbf{K}_u \mathbf{a} = -(-0.5 \quad 0.5) \begin{pmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \end{pmatrix} \begin{pmatrix} 2 \\ 0 \\ 3 \end{pmatrix} + 0 \cdot \mathbf{a} = 2.5$$

Lösung



In inkonsistenten redundanten Szenarien kann die von der Pseudo-Inversen gelieferte Lösung die steady-state Bedingung nicht erfüllen; allerdings hat diese Lösung den **kleinsten (quadratischen) Fehler** bzgl. der steady-state-Abweichung (hier: 0.5^2 (für A) + 0.5^2 (für B) = 0.5). Generell sollten inkonsistente Szenarien über (minimale) Korrekturen der gemessenen Raten **konsistent gemacht** werden, bevor man (mit den korrigierten Werten) die unbekanntes Raten bestimmt. Ein solches Verfahren wird im Folgenden skizziert.

Schätzverfahren für (inkonsistente) gemessene Raten r_b in redundanten Systemen

Redundante Systeme (mit gemessenen Raten r_b) sind gewöhnlich inkonsistent. Bevor man eine Schätzung für die unbekanntes Raten r_u berechnet, sollte man daher eine Korrektur der („falsch“) gemessenen Raten $r_{b,\text{korrekt}}$ bestimmen, so dass sich daraus ein konsistentes System ergibt (d.h. dass es ein r_u gibt mit $\mathbf{N}_u r_u = -\mathbf{N}_b r_{b,\text{korrekt}}$).

Beim Korrekturverfahren der (gewichteten) Kleinsten Quadrate kann man die „Genauigkeit“ der Messungen (die Messvarianz σ_i^2) mit einfließen lassen, diese werden auf der Diagonalen der Varianzmatrix \mathbf{F} angegeben. Kennt man keine Messvarianzen nimmt man für \mathbf{F} die Einheitsmatrix.

Zunächst bestimmt man die Redundanzmatrix \mathbf{R} , die sich durch Einsetzen der Lösung r_u in der Ausgangsgleichung ergibt und den Fehler quantifiziert (in konsistenten Systemen ist \mathbf{R} eine Nullmatrix!):

$$\mathbf{N}_u r_u + \mathbf{N}_b r_b = (-\mathbf{N}_u \mathbf{N}_u^\# \mathbf{N}_b + \mathbf{N}_b) r_b = \mathbf{R} r_b \rightarrow \mathbf{R} = \mathbf{N}_b - \mathbf{N}_u \mathbf{N}_u^\# \mathbf{N}_b$$

Für das weitere Vorgehen benötigt man die reduzierte Redundanzmatrix \mathbf{R}_r , die aus (beliebigen) $\text{rang}(\mathbf{R})$ vielen unabhängigen Zeilen von \mathbf{R} besteht ($\text{rang}(\mathbf{R})$ ist Redundanzgrad). Im besprochenen Fall 3 ist z.B.

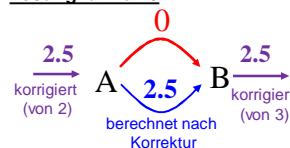
$$\mathbf{R} = \begin{pmatrix} 0.5 & 0 & -0.5 \\ 0.5 & 0 & -0.5 \end{pmatrix} \text{ und somit kann man in diesem Fall wählen } \mathbf{R}_r = (0.5 \quad 0 \quad -0.5).$$

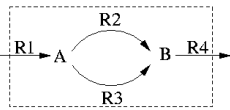
Möchte man den Vektor $r_{b,\text{mess}}$ nun so korrigieren ($r_{b,\text{korrekt}} = r_{b,\text{mess}} - \delta$), dass der gewichtete Fehler $\delta^T \mathbf{F}^{-1} \delta$ minimal wird, so lautet die Lösung (siehe Stephanopoulos et al.):

$$\mathbf{r}_{b,\text{korrekt}} = \mathbf{r}_{b,\text{mess}} - \delta = (\mathbf{I} - \mathbf{F} \mathbf{R}_r^T (\mathbf{R}_r \mathbf{F} \mathbf{R}_r^T)^{-1} \mathbf{R}_r) \mathbf{r}_{b,\text{mess}} \quad (\mathbf{I} = \text{Einheitsmatrix})$$

Im besprochenen Fall 3 mit der oben gegebenen reduzierten Redundanzmatrix \mathbf{R}_r und bei einer beliebigen Diagonalmatrix \mathbf{F} , wo die Einträge (Messvarianzen) auf der Diagonalen identisch sind, liefert diese Gleichung gerade $r_{b,\text{korrekt}} = (2.5, 0, 2.5)^T$. Das heißt, R1 wurde von 2 auf 2.5 und R4 von 3 auf 2.5 korrigiert (R2 bleibt unverändert). Danach kann man dann mit der klassischen Pseudo-Inversen-Lösung fortfahren und es ergibt sich als Resultat für die unbekanntes Rate $R3=2.5$.

Lösung für Fall 3





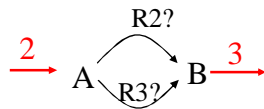
$$\begin{array}{cccc}
 \text{R1} & \text{R2} & \text{R3} & \text{R4} \\
 \mathbf{N} = \begin{pmatrix} 1 & -1 & -1 & 0 \\ 0 & 1 & 1 & -1 \end{pmatrix} & \rightarrow \text{A} & & \\
 & & & \rightarrow \text{B}
 \end{array}$$

$$\mathbf{N}_u \mathbf{r}_u = -\mathbf{N}_b \mathbf{r}_b$$

$$\mathbf{r}_u = -\mathbf{N}_u^\# \mathbf{N}_b \mathbf{r}_b + \mathbf{K}_u \mathbf{a}$$

Fall 4: unterbestimmt, redundant

(in rot: gemessen)



R1 und R4 inkonsistent!

$$\mathbf{r}_b = \begin{pmatrix} \text{R1} \\ \text{R4} \end{pmatrix} = \begin{pmatrix} 2 \\ 3 \end{pmatrix}; \quad x = 2; \quad \mathbf{N}_b = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}; \quad \mathbf{N}_u = \begin{pmatrix} -1 & -1 \\ 1 & 1 \end{pmatrix},$$

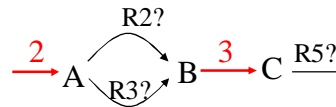
$\text{Rang}(\mathbf{N}_u) = 1 < x = m = 2 \rightarrow$ unterbestimmt, redundant

$$\mathbf{K}_u = \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \quad \mathbf{N}_u^\# = \begin{pmatrix} -0.25 & 0.25 \\ -0.25 & 0.25 \end{pmatrix}; \quad \rightarrow \text{Keine Rate berechenbar}$$

Inkonsistentes redundantes Szenario \rightarrow Schätzverfahren (wie Fall 3) liefert: $\hat{\mathbf{R}}_1 = \hat{\mathbf{R}}_3 = 2.5$

$$\mathbf{r}_u = \begin{pmatrix} \text{R2} \\ \text{R3} \end{pmatrix} = -\mathbf{N}_u^\# \mathbf{N}_b \hat{\mathbf{r}}_b + \mathbf{K}_u \mathbf{a} = -\begin{pmatrix} -0.25 & 0.25 \\ -0.25 & 0.25 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} 2.5 \\ 2.5 \end{pmatrix} + \begin{pmatrix} 1 \\ -1 \end{pmatrix} \mathbf{a} = \begin{pmatrix} 1.25 \\ 1.25 \end{pmatrix} + \begin{pmatrix} 1 \\ -1 \end{pmatrix} \mathbf{a}$$

Auch ein unterbestimmtes und redundantes System kann berechenbare Raten haben, z.B. R5 im Szenario rechts



Metabolische Flussanalyse (MFA):

- ermittelt physiologische Schnapsschüsse ("metabolischer Phänotyp" in einem Experiment)
- nützlich auch zum Testen "was passiert wenn"
- welche Sätze vorgegebener Raten determinieren andere?
- Sensitivitätsanalyse:

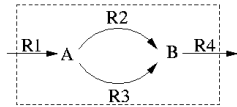
$$\mathbf{r}_u = -\mathbf{N}_u^\# \mathbf{N}_b \mathbf{r}_b + \mathbf{K}_u \mathbf{a} \quad \longrightarrow \quad \frac{d\mathbf{r}_u}{d\mathbf{r}_b} = -\mathbf{N}_u^\# \mathbf{N}_b$$

(für alle bestimmten $r_{u,i}$)

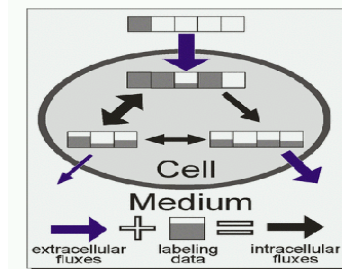
Limitierungen der MFA:

- Messen der externen Raten reicht zur Bestimmung der internen Flüsse oft nicht aus – Probleme durch Zyklen und parallele Wege:

Bsp: R1 und R4 reichen zur Bestimmung von R2 und/oder R3 nicht aus, da letztere einen parallelen Weg beschreiben
 → mind. eine Reaktionsrate eines parallelen Weges bzw. Zyklus' muss gemessen werden!



→ Alternative: Markierungsexperimente mit **¹³C-Isotopen** liefern weitere (nichtlineare) Nebenbedingungen zum weiteren Auflösen der Flüsse



- große (genomskalige) Netze mit (zu) vielen Freiheitsgraden:
 (z.B. *E. coli* iAF1260 Netzwerk: 2382 reactions, 1668 metabolites → 714 Freiheitsgrade!)
- MFA (auch mit ¹³C-Markierung) nur für kleinere Modelle sinnvoll (typischerweise Zentralstoffwechsel)

III.4.3 Optimale Flussverteilungen (Flux Balance Analysis, FBA)

- optimale Flussverteilungen wichtig z.B. für **Metabolic Engineering** (z.B. wie kann max. Produktausbeute erzielt werden?)
- einige Bakterien wachsen "optimal" (d.h. mit max. Biomassessynthese)

Allgemeines Optimierungsproblem für Flussverteilungen:

Nebenbedingungen (*Constraints*)

(NB1) $\mathbf{N}r = \mathbf{0}$ (Fließgleichgewicht)

(NB2) $\mathbf{A}r = \mathbf{b}$ (optional: inhomogene NB - z.B. Messungen)

(NB3) $\alpha_i \leq r_i \leq \beta_i$ Flussgrenzen für jede Reaktion
 $\alpha_i=0$ wenn Reaktion *i* irreversibel

Zielfunktion (was wird optimiert):

$$\mathbf{c}^T \mathbf{r} = c_1 r_1 + c_2 r_2 + \dots + c_q r_q = \max! \text{ (oder min!)}$$

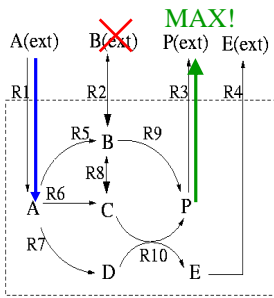
typische Zielfunktionen:

"natürliche Zielfunktion": Maximiere Wachstum (Rate μ)! Maximiere ATP Produktion!

Metabolic Engineering: Maximiere Produktausstoß!

- lineares Optimierungsproblem ("Linear Programming Problem")
- verschiedene Algorithmen, z.B. Simplex-Algorithmus

Beispiel: **Optimale Flussverteilung zur Synthese von P_{ext} aus 1 mol A_{ext}** (ohne B_{ext})??



Nebenbedingungen

(NB1) $\mathbf{Nr} = \mathbf{0}$ (N siehe Folie vorne)

(NB2)

$$\mathbf{Ar} = \mathbf{b} : \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \mathbf{r} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

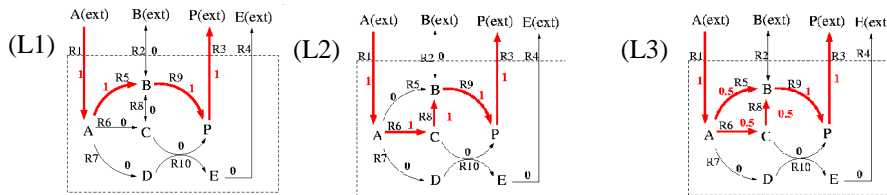
(NB3) untere Schranken: $\alpha = (0, -\text{Inf}, 0, 0, 0, 0, 0, -\text{Inf}, 0, 0)$

obere Schranken: $\beta = (\text{Inf}, \text{Inf}, \dots, \text{Inf}, \text{Inf})$

Zielfunktion:

$$\mathbf{c}^T \mathbf{r} = (0, 0, 1, 0, 0, 0, 0, 0, 0, 0) \quad \mathbf{r} = r_{R3} = \max!$$

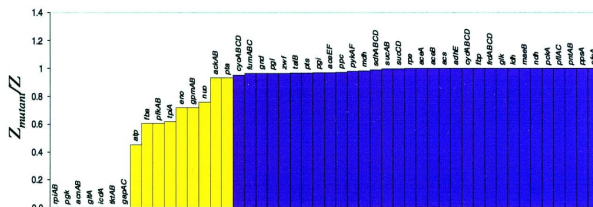
Optimale Lösungen (wo $r_3 = 1$ ist)



Hier existieren unendlich viele (!) optimale Lösungen, die man aus konvexen Kombinationen der ersten und zweiten gewinnen kann (z.B. (L3) = 0.5*(L1) + 0.5*(L2)). Die Eindeutigkeit von optimalen Lösungen sollte immer überprüft werden!

Anwendungen für optimale Flussverteilungen

- Bestimmung von Flussverteilungen mit maximaler Produkt/Biomassesyntesrate
- Erkennen anderer extremer physiologischer Leistungen
- ist eine bestimmte Funktion (z.B. Synthese eines Metaboliten) in einem Netz überhaupt realisierbar?
 - optimiere nach jeweiliger Rate und überprüfe, ob diese größer Null ist
 - Anwendung: kann Mutante (Reaktion x geblockt, d.h. Rate ist 0) noch wachsen? Vorgehen: setze Rate von x auf 0 und maximiere μ



Gene deletions in *E. coli* MG1655 central intermediary metabolism; maximal biomass yields on glucose for all possible single gene deletions in the central metabolic pathways. The optimal value of the mutant objective function (Z_{mutant}) compared with the "wild-type" objective function (Z), where Z is defined in Eq. 3. The ratio of optimal growth yields (Z_{mutant}/Z). The results were generated in a simulated aerobic environment with glucose as the carbon source (Quelle: Edwards JS, Palsson BO: *The Escherichia coli* MG1655 in silico metabolic genotype: its definition, characteristics, and capabilities. *Proc Natl Acad Sci USA* 2000, 97:5528-33)

- Lineare Optimierung auch in großen (genom-skaligen) Netzen algorithmisch relativ problemlos

**Flussvariabilitätsanalyse (flux variability analysis):
„Flussanalyse“ mittels linearer Optimierung**

Substrat

$r_{\text{Substrate_Aufnahme}} = 9.45$

01100 7/5/02

Messungen

$r_{O_2} = 6.31$

$r_{CO_2} = 8.31$

$\mu_{\text{Wachstum}} = 0.71$

Klassische MFA:
oft unterbestimmt.

→ Bestimme die möglichen Bereiche aller Flüsse!

Bestimme Wertebereich (min/max Werte) aller unbekanntener Flüsse i :

$$r_{i,\min} = \min_r r_i$$

$$r_{i,\max} = \max_r r_i$$

s.t. (C1), (C2), (C3), (C4)

C1) Steady-state: $Nr = 0$

C2) Reversibilität: $r_{\text{Irrev}} \geq 0$

C3) Flussgrenzen: $l_i \leq r_i \leq u_i$

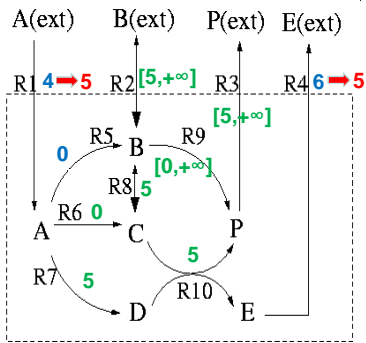
C4) Gemessene Flüsse: $r_j = m_j$

**Flussvariabilitätsanalyse (flux variability analysis):
Flussanalyse mittels linearer Optimierung**

- wenn $r_i^{\min} = r_i^{\max}$, dann ist r_i eindeutig bestimmt
- auf diese Art und Weise lässt sich generell auch eine Metabolische Flussanalyse durchführen (ohne explizite Berechnung via Pseudo-Inverse etc.)
 - Vorteile: (i) die Irreversibilitäten und Min/Max-Raten werden berücksichtigt
 - (ii) ist die Rate nicht eindeutig, so ist zumindest die mögliche Spanne bekannt
- Nachteile: (i) $2 \cdot x$ Optimierungen notwendig (x = Anzahl unbekannter Raten)
- (ii) inkonsistente (redundante) Systeme nicht (direkt) behandelbar
 - aber: über Einbau von Schlupfvariablen für die Korrektur möglich

Auflösung von Inkonsistenzen bei der FVA

Wie vorgehen, wenn der LP- Solver bei der FVA eine Inkonsistenz (Fehler: infeasibility) meldet?



Korrekturmethode: finde minimale Änderungen δ_i in den gemessenen Flüssen, die das Szenario konsistent machen

Methods:

- **Quadratic Program (QP):** $\min \sum w_i \delta_i^2$
- **Linear Program:** $\min \sum w_i (\delta_i^+ + \delta_i^-)$

Weights:

- **Uniform:** $w_i = 1$
- **Relative:** $w_i = 1/|r_{i,measured}|$
- **Measurement variance:** $w_i = 1/\sigma_i^2$

Blau: gemessene Flüsse → Inkonsistenz bei FVA detektiert!

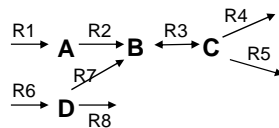
Rot: korrigierte gemessene Flüsse, die das Szenario konsistent machen (QP-Lösung mit $w_i=1$).

Grün: mittels FVA berechnete Flüsse bzw. Flussintervalle im konsistent gemachten System.

Klamt S, von Kamp A (2022) Analyzing and Resolving Infeasibility in Flux Balance Analysis of Metabolic Networks. *Metabolites* 12: 585.

Flussvariabilitätsanalyse (FVA) vs. MFA

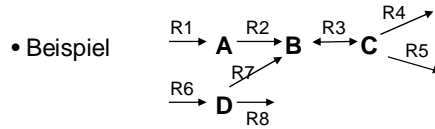
• Beispiel



Szenario		R1	R2	R3	R4	R5	R6	R7	R8
R1=1	Flussanalyse (MFA klassisch)								
	Flussvariabilitätsanalyse (FVA)								
R1=1; R6=0;	Flussanalyse (MFA klassisch)								
	Flussvariabilitätsanalyse (FVA)								
R1=1; R6=0; R2=2; R8=0	Flussanalyse (MFA klassisch)								
	Flussvariabilitätsanalyse (FVA)								

* = geschätzt

Flussvariabilitätsanalyse (FVA) vs. MFA



Szenario		R1	R2	R3	R4	R5	R6	R7	R8
R1=1	Flussanalyse (MFA klassisch)	1	1	$[-\infty, \infty]$	$[-\infty, \infty]$	$[-\infty, \infty]$	$[-\infty, \infty]$	$[-\infty, \infty]$	$[-\infty, \infty]$
	Flussvariabilitätsanalyse (FVA)	1	1	$[1, \infty]$	$[0, \infty]$	$[0, \infty]$	$[0, \infty]$	$[0, \infty]$	$[0, \infty]$
R1=1; R6=0;	Flussanalyse (MFA klassisch)	1	1	$[-\infty, \infty]$	$[-\infty, \infty]$	$[-\infty, \infty]$	0	$[-\infty, \infty]$	$[-\infty, \infty]$
	Flussvariabilitätsanalyse (FVA)	1	1	1	$[0, 1]$	$[0, 1]$	0	0	0
R1=1; R6=0; R2=2; R8=0	Flussanalyse (MFA klassisch)	1.5*	1.5*	1.5*	$[-\infty, \infty]$	$[-\infty, \infty]$	0	0	0
	Flussvariabilitätsanalyse (FVA)	(1.5*)	(1.5*)	(1.5*)	$[0, 1.5]$	$[0, 1.5]$	0	0	0

* Nach Anpassung

Maximierung der Rate vs. Maximierung der Ausbeute

Bisher haben wir die Maximierung einer **Reaktionsrate** r_P betrachtet:

$$\max r_P !$$

(z.B.: maximale Wachstumsrate oder Produktsyntheserate). Eine Lösung kann über Lineare Optimierung gefunden werden.

In manchen Anwendungen (z.B. Biotechnologie) ist man aber an der Maximierung einer **Produktausbeute (yield)** $Y^{P/S} = r_P / r_S$ interessiert, wo r_P die Produktsyntheserate und r_S die Substratverbrauchsrate (Substrataufnahmerate) darstellen:

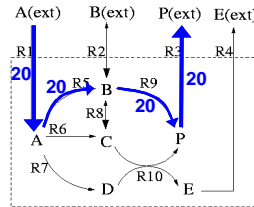
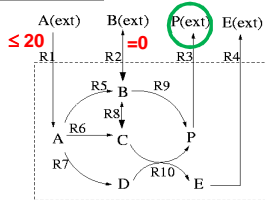
$$\max r_P / r_S !$$

Ausbeutemaximierung ist somit die Optimierung eines Verhältnisses von Reaktionsraten. Derartige Optimierungen gehören zum **linear-fractional programming** und sind etwas komplizierter als die lineare Optimierung einer Rate.

Beispiel: Maximierung der Rate vs. Maximierung der Ausbeute

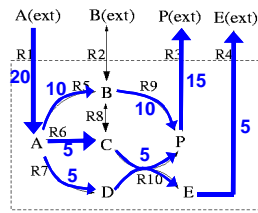
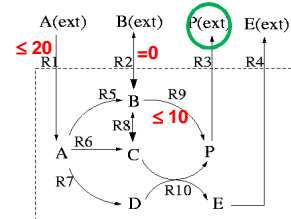
Ratenoptimale Lösungen sind oft auch optimal in der Ausbeute. Aber nicht immer!

Szenario 1:

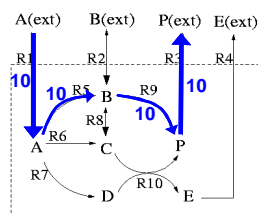


(Eine) optimale Lösung für maximale Syntheserate von P(ext). Diese Lösung ist auch für die Ausbeute optimal: $P_{ext}/A_{ext} = 1!$

Szenario 2:



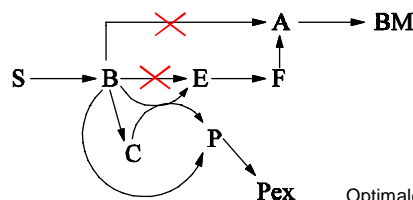
Lösung für optimale (maximale) Rate für Synthese von P(ext). Aber Ausbeute ist $15/20=0.75$ und somit nicht ausbeutenoptimal.



(Eine) Lösung für optimale Ausbeute ($10/10=1$) für Synthese von P(ext) aus A(ext). Ist aber nicht ratenoptimal für P(ext)-Synthese!

Stammoptimierung (Metabolic Engineering) mittels optimaler Flussverteilungen (OptKnock / RobustKnock - Methode)

- viele Mikroorganismen optimieren ihre Wachstumsrate
- Adaptive Evolution: nach Gen-knockouts oder Änderung der Umweltbedingungen stellen sich die Bakterien um (zum Teil durch Selektion) und wachsen nach einer Weile optimal bzgl. der neuen Situation
- Idee: Suche nach Mutationen, so daß Biomassessynthese strikt an die Synthese eines gewünschten Produktes gekoppelt wird (Produktsynthese wird dann implizit zusammen mit Biomassessynthese optimiert)

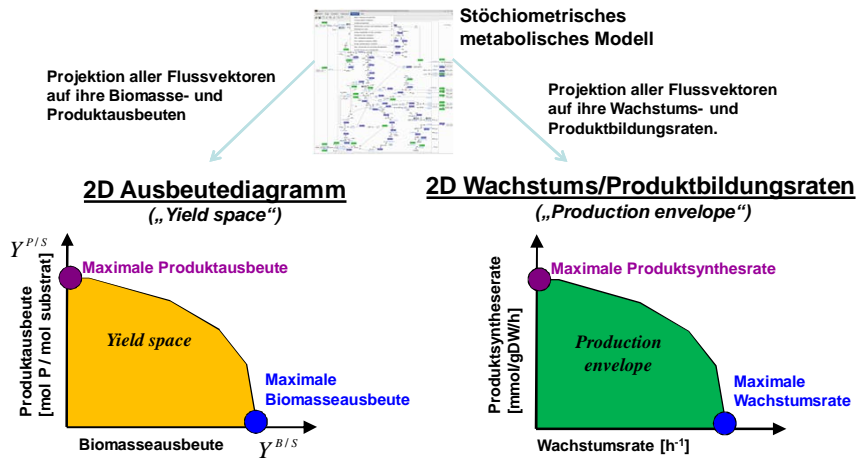


- S: Substrat
- Pex: Produkt von Interesse
- BM: Biomasse
- A: Aminosäure (essentiell für Synthese von BM)

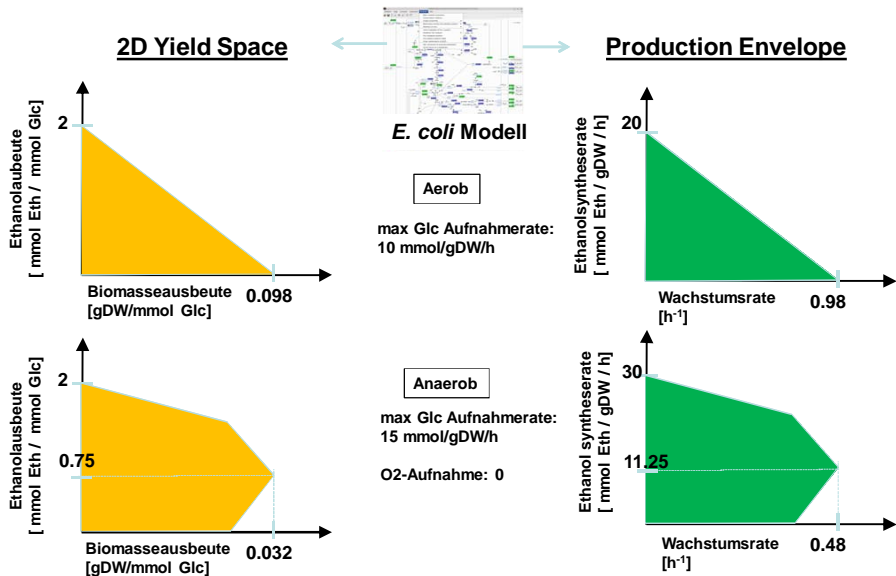
Optimale knock-outs: B→A und B→E
BM kann dann nur mit Ausscheidung von Pex produziert werden

Wachstum vs. Produktsynthese: 2D Trade-off-Diagramme von Biomasse/Produktsynthese

→ Produktsynthese konkurriert immer mit (und schmälert) Biomassesynthese.
 → Dieser „Trade-off“ lässt sich in 2-dimensionalen Diagrammen der möglichen Biomasse/Produktausbeuten bzw. -raten visualisieren, die wichtige Werkzeuge für die Stammoptimierung sind.

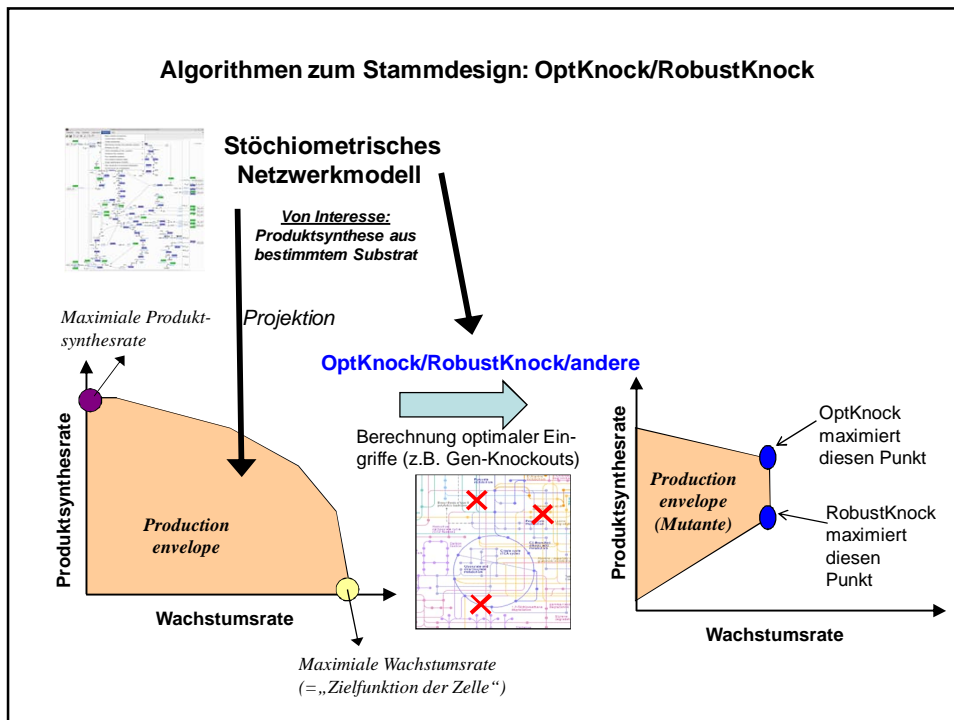


2D Trade-off-Diagramme von Biomasse/Produktsynthese Beispiel: Ethanolsynthese in *E. coli* aerob + anaerob



Yield Space und Production Envelope können auch unterschiedliche Gestalt haben!

Algorithmen zum Stammdesign: OptKnock/RobustKnock



Algorithmen zum Stammdesign: OptKnock/RobustKnock

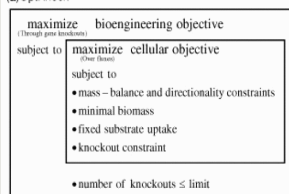
- OptKnock und RobustKnock führen zu sogenannten Bilevel- und Trilevel-Optimierungsproblemen (siehe rechts):

- Level 1: (innen): optimiere wachstum!
- Level 2: (außen): maximiere durch zu wählende knock-outs den dabei erzielbaren maximalen (OptKnock) bzw. minimalen (RobustKnock) Ausstoß an Produkt

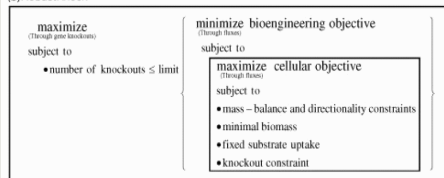
- Lösung mittels Mixed Integer Linear Programming (MILP):
 - kontinuierliche Größen: Flüsse
 - Integer-Größen: knockouts (an/aus)
- Kopplung von Produkt- und Biomasse-synthese kann durch adaptive Evolution weiter verbessert werden

- erfolgreich in der Praxis eingesetzt (z.B. Lactat-Production mit *E. coli*)

(a) OptKnock



(b) RobustKnock



Alternativer Algorithmus zur Bestimmung von Interventionsstrategien für die Kopplung von Biomasse- und Produktsynthese: **Minimal Cut Sets (Kap. III.4.5)**

Einbau weiterer Nebenbedingungen in das Optimierungsproblem Beispiel: "Enzyme allocation constraints"

Motivation:

Die Summe aller Flüsse in der Zelle kann nicht beliebig groß werden, da sie durch Enzyme katalysiert werden müssen, für die nur begrenzt Platz ist (Anteil Gesamtproteom an Zellmasse liegt bei ca. 50%).

Ableitung einer Nebenbedingung für die Enzymallokation (siehe Bekiaris and Klamt, 2020):

Die Rate r_i einer Reaktion i hängt von vorhandener Menge (Konzentration) des Enzyms E_i [mmol/gDW], seiner katalytischen Konstante (Wechselzahl) $k_{cat,i}$ [1/s] und einem Sättigungs/Inhibierungsterm der Substrate ab ($f_i(S)$), der charakteristisch für die jeweilige Kinetik ist: $r_i = E_i \cdot k_{cat,i} \cdot f_i(S)$ ($E_i \cdot k_{cat,i} = v_{max}$)

Es gilt $f_i(S) \leq 1$ für alle Kinetiken und somit kann man schreiben: $r_i \leq E_i \cdot k_{cat,i}$

Für einen gegebenen Fluss r_i beträgt die minimale Enzymkonz. E_i somit $E_i \geq r_i / k_{cat,i}$

Umrechnung von Konz. E_i [mmol/gDW] in rel. Masse m_i [g / gDW] mittels Molekulargewicht MW_i [g/mol] des Enzyms ergibt: $E_i \cdot MW_i \geq r_i / k_{cat,i} \rightarrow m_i \geq r_i \cdot MW_i / k_{cat,i}$

Mit einer bekannten Abschätzung für die maximale Menge aller metabolisch relevanten Enzyme P (z.B. ~ 0.2 g/gDW) ergibt sich: $\sum r_i \cdot \frac{MW_i}{k_{cat,i}} \leq P$

Dies kann man umschreiben zu:

$$-\sum r_i \cdot \frac{MW_i}{k_{cat,i}} + r_{Pool} = 0; \quad r_{Pool} \leq P$$

(Achtung: reversible Reaktionen müssen hier vorher in 2 irreversible aufgesplittet werden und $k_{cat,i}$ ist in [1/h] umzurechnen!)

Integration
in stöch. Matrix
(neue Pseudoreaktion R_{Pool}
und neuer Pseudometabolit M_{Pool})

$$\mathbf{N}_{new} = \begin{bmatrix} R_1 & \dots & R_r & R_{Pool} \\ M_1 & s_{1,1} & \dots & s_{1,r} & 0 \\ \vdots & \vdots & \ddots & \vdots & 0 \\ M_m & s_{m,1} & \dots & s_{m,r} & 0 \\ M_{Pool} & \frac{MW_1}{k_{cat,1}} & \dots & \frac{MW_r}{k_{cat,r}} & 1 \end{bmatrix}$$

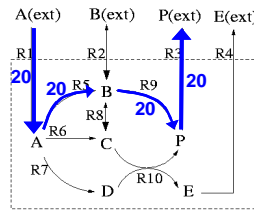
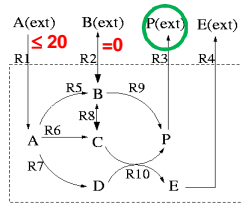
$r_{Pool} \leq P$

Mit "steady state" Annahme $\mathbf{N}_{new} \mathbf{r}_{new} = \mathbf{0}$ und flux bound $r_{Pool} \leq P$ sind die Enzymallokationsbedingungen integriert.

Beispiel: Effekte von Enzymallokationsproblemen

Szenario 1 (max. Rate für Substrat-Aufnahme):

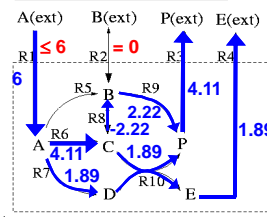
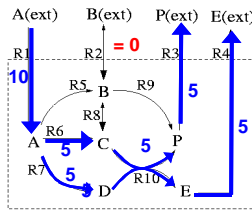
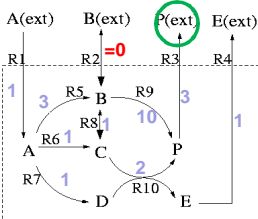
Optimierung für R3!



Eine mögliche optimale Lösung für maximale Syntheserate von P(ext). Die Ausbeute-optimale Lösung wird verwendet.

Szenario 2: Proteinkosten

Szenario 3: Proteinkosten + max. Substrataufnahme



„1“ = rel. Enzymkosten ($= \frac{MW_i}{k_{cat,i}}$)

Optimale Lösung für max. Syntheserate für P. Der Ausbeute-optimale Weg wird nicht genutzt aufgrund der hohen Enzymkosten.

Aufnahme von Substrat A beschränkt. Nun ist die optimale Lösung eine Kombination der Wege mit optimaler bzw. suboptimaler Ausbeute.

Angenommene Gesamtzymkapazität: 50

Konstellationen wie in Szenario 3 erklären respirofermentatives Wachstum (Kombination von Atmung und Gärstoffwechsel; Bekiaris and Klamt, 2020)

Lösungsräume für lineare Gleichungs- und Ungleichungssysteme (im Kontext der Metabolischen Netzwerkanalyse)

Lineare Gleichungen / Ungleichungen (= *linear constraints*) in der Metabolischen Netzwerkanalyse:

- | | | |
|------------------------------|-------------------------------|--|
| (1) Stationarität: | $\mathbf{Nr} = \mathbf{0}$ | (praktisch immer!) |
| (2) Reversibilität: | $r_j \geq 0$ | wenn Reaktion j irreversibel |
| (3) max/min Flussraten: | $l_j \leq r_j \leq u_j$ | (z.B., max. Substrataufnahmerate) |
| (4) Messungen: | $r_j = b$ | (z.B., gemessene Produktbildungsrate) |
| (5) andere Nebenbedingungen: | $\mathbf{Ar} \leq \mathbf{b}$ | (z.B. Proteinallokationsconstraints
oder „Summe von 2 Raten ist kleiner 1“) |

[Hinweis: (2) und (4) könnten in (3) integriert werden; (1)-(4) komplett in (5)]

(Stöchiometrische) Modellierungsmethoden, die lineare Bedingungen des Typs (1)-(5) verwenden (dabei (1) obligatorisch) werden auch **constraint-basiert** („*constraint-based modeling of metabolic networks*“) genannt.

Wir schauen uns im Folgenden an, wie sich der Lösungsraum durch Zugabe der einzelnen constraints (1)-(4) verändert und welche Methoden darauf jeweils einsetzbar sind ...

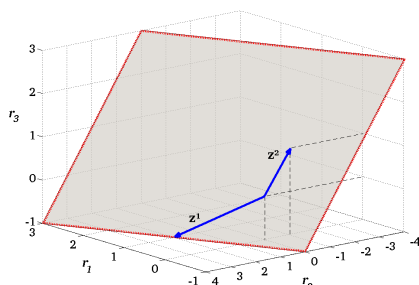
Lösungsräume für lineare Gleichungs- und Ungleichungssysteme (im Kontext der Metabolischen Netzwerkanalyse)

Constraints

C1) Steady-state: $\mathbf{Nr} = \mathbf{0}$

→ **Nullraumanalyse**

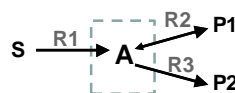
- Gekoppelte Reaktionen
- Geblockte Reaktionen
- ...



Die roten Ränder deuten unbeschränkte Richtungen an.

Lösungsraum der Flussvektoren r:

Nullraum von N



$$\mathbf{N} = (1 \ -1 \ -1)$$

$$\dim(\ker(\mathbf{N})) = \#\text{reac} - \text{rank}(\mathbf{N})$$

(Nicht eindeutige) Basisvektoren des Nullraums von N

$$\mathbf{z}_1 = \begin{pmatrix} 1 \\ 2 \\ -1 \end{pmatrix}, \quad \mathbf{z}_2 = \begin{pmatrix} 0 \\ -1 \\ 1 \end{pmatrix}$$

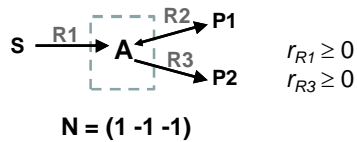
Lösungsräume für lineare Gleichungs- und Ungleichungssysteme (im Kontext der Metabolischen Netzwerkanalyse)

Constraints

- C1) Steady-state:** $Nr = 0$
C2) Irreversibilität: $r_{Irrev} \geq 0$

Lösungsraum der Flussvektoren r:

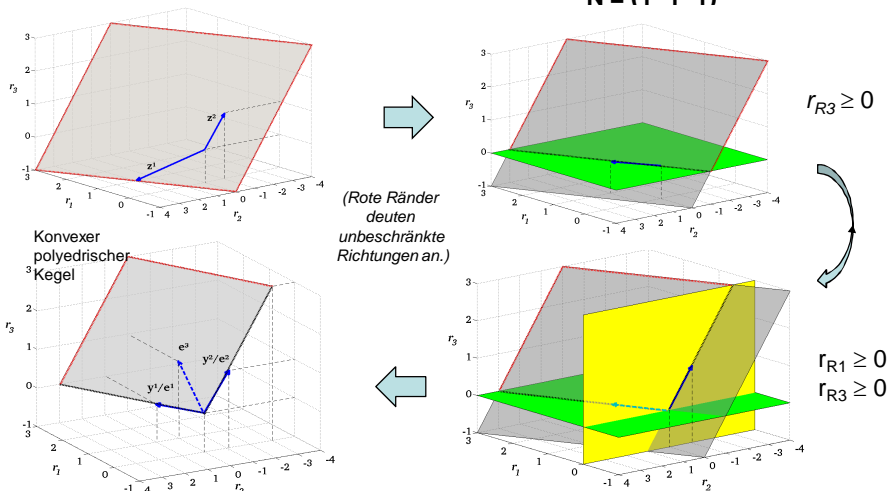
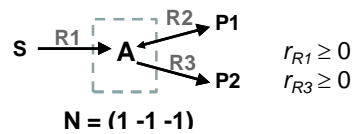
Konvexer polyedrischer Kegel



Vom Nullraum zum polyedrischen Kegel

Constraints

- C1) Steady-state:** $Nr = 0$
C2) Irreversibilität: $r_{Irrev} \geq 0$



Lösungsräume für lineare Gleichungs- und Ungleichungssysteme (im Kontext der Metabolischen Netzwerkanalyse)

Constraints

- C1) Steady-state:** $Nr = 0$
C2) Irreversibilität: $r_{Irrev} \geq 0$

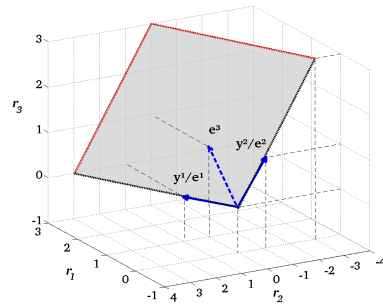
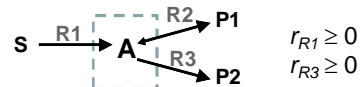
→ Analyse des Kegels mittels
Elementarmoden (Kap. III.4.4)

$$e_1 = y_1 = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}, e_2 = y_2 = \begin{pmatrix} 0 \\ -1 \\ 1 \end{pmatrix}, e_3 = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}$$

Kanten (Extremstrahlen): $y_1 (= e_1), y_2 (= e_2)$
 Elementarmoden: e_1, e_2, e_3

Lösungsraum der Flussvektoren r:

Konvexer polyedrischer Kegel
 (= Flusskegel)



Kegel K: wenn $x \in K$, dann auch βx ($\beta \geq 0$)

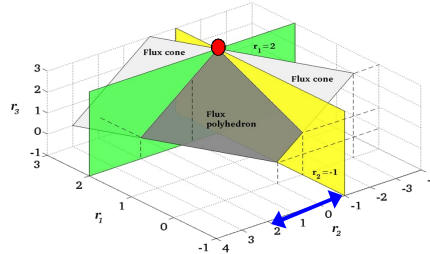
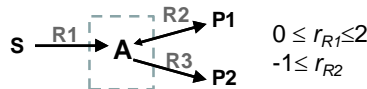
Lösungsräume für lineare Gleichungs- und Ungleichungssysteme (im Kontext der Metabolischen Netzwerkanalyse)

Constraints

- C1) Steady-state:** $Nr = 0$
C2) Irreversibilität: $r_{Irrev} \geq 0$
C3) max./min. Raten: $l_j \leq r_j \leq u_j$

Lösungsraum der Flussvektoren r:

Beschränkter/unbeschränkter Flusspolyeder



→ Flussoptimierung (FBA)

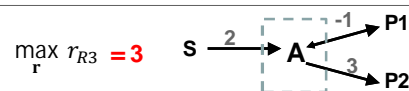
Analyse optimaler Flussverteilungen.

$$\max_r w^T r \quad \text{s.t. (C1),(C2),(C3)}$$

- Lineare Optimierungsprobleme.
 → Beispiele: maximiere Biomasse- oder Produktsynthese.

→ Flussvariabilitätsanalyse

Bestimme/analysiere mögliche Flussbereiche. $r_{i,\min} = \min_r r_i$
 $r_{i,\max} = \max_r r_i$
 s.t. (C1), (C2), (C3)



Möglicher Bereich für R2: $r_2 \in [-1, 2]$

Lösungsräume für lineare Gleichungs- und Ungleichungssysteme (im Kontext der Metabolischen Netzwerkanalyse)

Constraints

- C1) Steady-state:** $\mathbf{N}\mathbf{r} = \mathbf{0}$
- C2) Irreversibilität:** $r_{Irrev} \geq 0$
- C3) max./min. Raten:** $l_i \leq r_i \leq u_i$
- C4) Gemessene Flüsse:** $r_j = m_j$

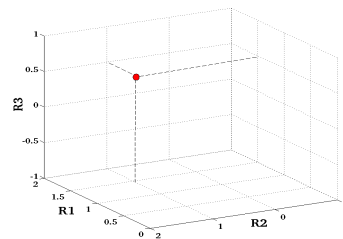
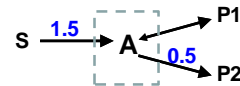
→ Metabolische Flussanalyse

Berechne interne Flussverteilungen aus experimentellen Daten

Lösungsraum der Flussvektoren \mathbf{r} :

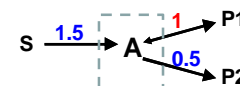
Polyeder (im (determinierten) Spezialfall verbleibt einzelner Punkt!)

Gemessen:
 $r_1=1.5$
 $r_3=0.5$



Gemessen:
 $r_1=1.5; r_3=0.5$

Berechnet: $r_2=1$



III.4.4 Elementarmoden und Pathwayanalyse

Ziel: Beschreibe minimale funktionale Einheiten eines SN im Fließgleichgewicht

Notation *support*: $P(\mathbf{r}) = \{i: r_i \neq 0\}$ $\mathbf{P} \begin{pmatrix} 1.4 \\ 0 \\ 5 \end{pmatrix} = \{1,3\}$

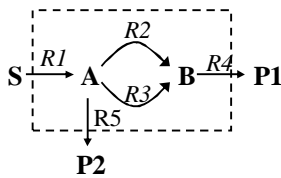
Elementarmoden: Ein Reaktionsratenvektor \mathbf{e} ist ein Elementarmodus (EM) wenn er die folgenden 3 Eigenschaften erfüllt

- C1) *Fließgleichgewicht*: $\mathbf{N}\mathbf{e} = \mathbf{0}$
- C2) *Thermodynamik*: $e_j \geq 0$ (falls Reaktion j irreversibel)
- C3) *Unzerlegbarkeit*: Es gibt keinen nichtrivialen Ratenvektor \mathbf{r} , der C1 und C2 erfüllt und für den gilt: $P(\mathbf{r}) \subset P(\mathbf{e})$ (EMen sind support-minimal)

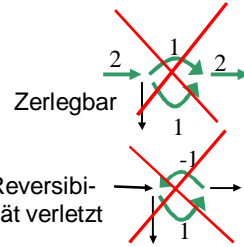
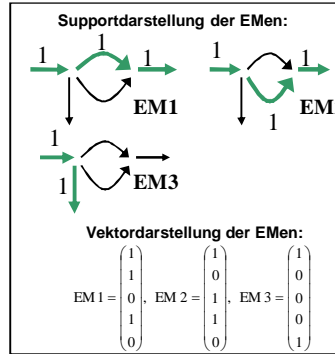
Wenn eine SFV \mathbf{e} ein EM ist, so stellt $k\mathbf{e}$ (k : positiv und skalar) den gleichen EM dar. Jeder EM \mathbf{e} ist also über seine Nichtnullkomponenten (support $P(\mathbf{e})$) charakterisiert und damit ist $P(\mathbf{e})=P(k\mathbf{e})$.

Beispiel und Eigenschaften von Elementarmoden

Beispiel: SN3



3 Elementarmoden



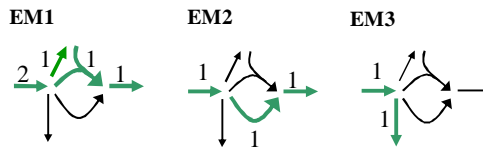
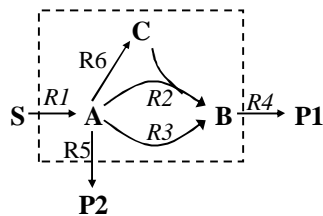
- EMen entsprechen metabolischen Routen (die externe Substrate und Produkte verbinden) oder internen Zyklen ohne Nettoumsatz

- EM1 und EM2 zeigen Synthesewege für P1
- EM3 ist (einziger) Syntheseweg für P2
- andere Synthesewege gibt es nicht, allerdings kann P1 z.B. auch über Linearkombination von EM1 und EM2 produziert werden

- SN3 ist ein Graph (nur unimolekulare Reaktionen) wodurch die EMen relativ einfache Struktur haben und Wegen oder Zyklen in Graphen entsprechen

- Elementarmoden werden komplizierter wenn bimolekulare Reaktionen involviert sind (also ein echter Hypergraph vorliegt) wie in SN4:

Beispiel: SN4



Beachte, dass in EM1 der Koeffizient für Reaktion R1 gerade 2 ist und R6 und R2 gleichzeitig laufen müssen.

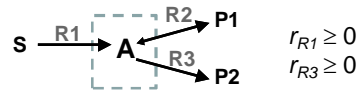
Erinnere: Flusskegel und geometrische Deutung von Elementarmoden

Constraints

- C1) Steady-state:** $Nr = 0$
C2) Irreversibilität: $r_{Irrev} \geq 0$

Lösungsraum der Flussvektoren r:

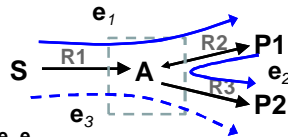
Konvexer polyedrischer Kegel (= Flusskegel)



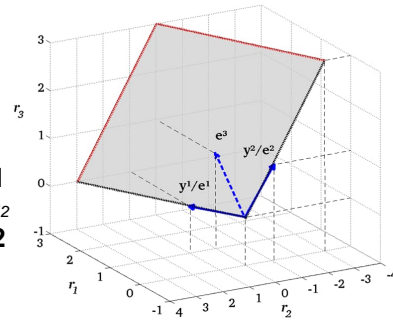
→ Analyse des Kegels mittels **Elementarmoden**

→ Elementarmoden umfassen alle **Kantenvektoren (= Extremstrahlen)** des Flusskegels, sowie einige weitere spezielle Flussverteilungen

$$e_1 = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}, e_2 = \begin{pmatrix} 0 \\ -1 \\ 1 \end{pmatrix}, e_3 = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}$$



Kanten (Extremstrahlen): e_1, e_2
 Elementarmoden: e_1, e_2, e_3



Geometrische Deutung und Berechnung von Elementarmoden

Flusskegel: $F = \{r: Nr=0; r_{Irrev} \geq 0\}$

- Alle Kanten (Extremstrahlen) des Flusskegels sind EMen (es kann aber weitere EMen im Innern geben).
- Alle realisierbaren SFV $r \in F$ *konisch* (mit nichtnegativen Linearkombinationen) aus den EMen generieren:

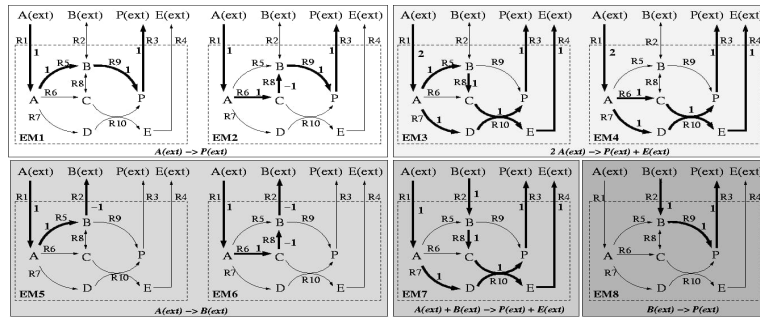
$$r = \sum_j \alpha_j e^j ; (\alpha_j \geq 0)$$

- EMen können über Varianten der *double description method* berechnet werden (bekannt aus kombinatorischer/algorithmischer Geometrie zur Berechnung der Extremalstrahlen polyedrischer Kegel, die aus Gleichungs/Ungleichungssystem $Ax \geq 0$ entstehen)
- Anzahl an EMen wächst in großen Netzen exponentiell; daher können meist nur mittelgroße Netze (ca 120 Reaktionen; <1 Mill. EMen) untersucht werden

Anwendungen von Elementarmoden

- EMen enthalten alle minimalen Synthesewege eines Produktes inkl. aller optimalen und sub-optimalen Synthesewege

8 Elementarmoden in SN2

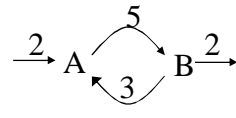
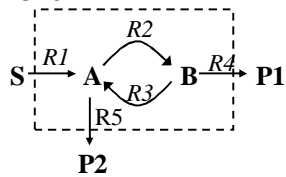


Beispiel SN2: die (2) optimalen Routen um aus A Produkt P zu produzieren werden identifiziert; alle optimalen SFV sind Linearkombinationen davon (vgl. mit Optimierung!)

Anwendungen von Elementarmoden

- EM ohne Beteiligung externer Metabolite ist interner Zyklus, dessen Nettofluss (sofern alle beteiligten Metabolite im Modell berücksichtigt sind) aus thermodynamischen Gründen Null sein muss

Bsp: SN3'



Kann real nicht auftreten, da Zyklus-EM Nettofluss 3 hat.

- **blockierte** Reaktionen sind dadurch erkennbar, dass sie in keinem EM involviert sind (Rate ist überall Null)
- **gekoppelte** Reaktionen treten immer gemeinsam in den EMen auf; es kann auch hierarchische Kopplungen geben: in SN3 ist R1 allen anderen Reaktionen übergeordnet (R2-R5 operieren nur wenn R1 aktiv ist – das gilt nicht umgekehrt)

Anwendungen von Elementarmoden

- desto mehr EMen für eine Funktion zur Verfügung stehen desto flexibler (robuster) ist das Netzwerk (z.B. 2 EMen für P1-Synthese, nur einer für P2 in SN3)
 - je häufiger eine Reaktion in EMen auftaucht, desto wichtiger ist sie für Funktion des Netzwerkes; **essentielle** Reaktionen treten in allen EMen auf (z.B. R1)
 - entfernt man Reaktion *i* aus dem Netz ("Mutation") verbleiben all diejenigen EMen funktional und erhalten, die *i* nicht involvieren
z.B. in SN3:
 - (1) EM2 und EM3 bleiben aktiv wenn R2 inaktiviert wird
 - (2) EM3 bleibt aktiv wenn R4 entfernt wird
 - (3) kein EM verbleibt, wenn R1 entfernt wird
- verbleibende Funktionalität in Mutanten läßt sich leicht mit EMen überprüfen
- Bestimmung von gezielten Eingriffen: **Minimal Cut Sets** (folgt ...)

Beispiel Zentralstoffwechsel *E.coli* : Elementarmodenanalyse

110 Reaktionen
89 Metabolite

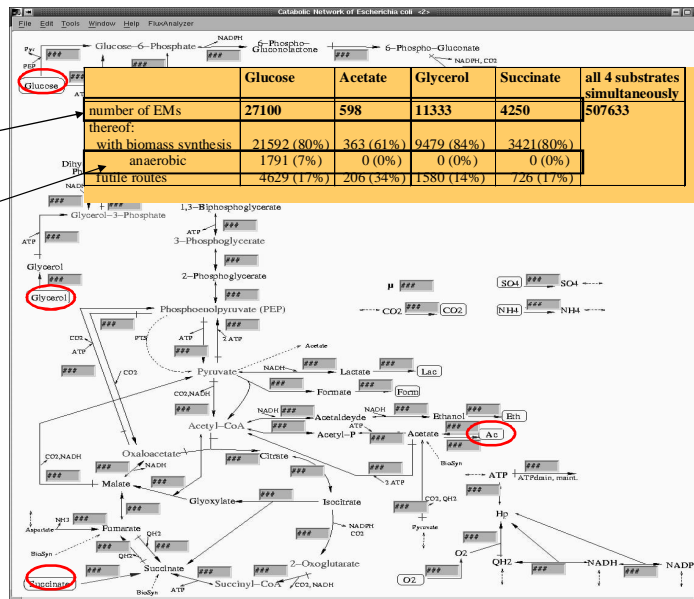
(Flexibilität für versch. Substrate)

(Fähigkeit für anaerobes Wachstum)

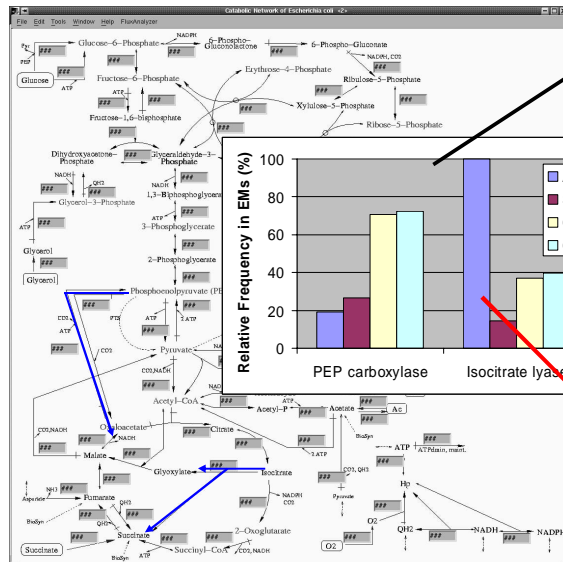
Substrate:

Glucose
Succinat
Glycerol
Acetat

Stelling, Klamt, et al.; Nature, 2002



EM-Analyse in *E.coli*: Wichtigkeit einzelner Reaktionen



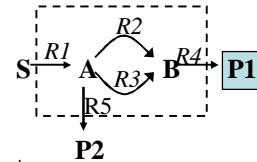
Korreliert mit gemessenen Genexpressionsdaten

Essentiell (auch im Experiment!)

→ Phänotyppräditionen von Mutanten

III.4.5 Minimal cut sets

Aufgabe: Unterdrücke Synthese von P1 in Netz SN3!



Cut Set = Menge von Eingriffen (knock-outs), die eine vorgegebene Zielfunktionalität unterbinden

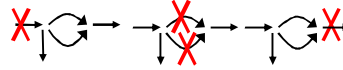
hier: c ist cut set falls für jedes r mit $Nr = 0$; $r_i \geq 0 \forall i \in Irrev$; $r_c = 0 \forall c \in c$ folgt, dass $r_{R4} = 0$ ist

Minimal Cut Set (MCSs) = support-minimales cut set (keine Teilmenge eines MCS ist cut set)

3 Beispiele für cut sets:

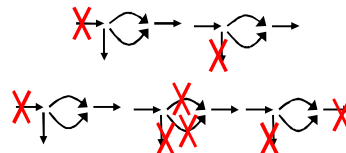


Insgesamt 3 minimal cut sets:



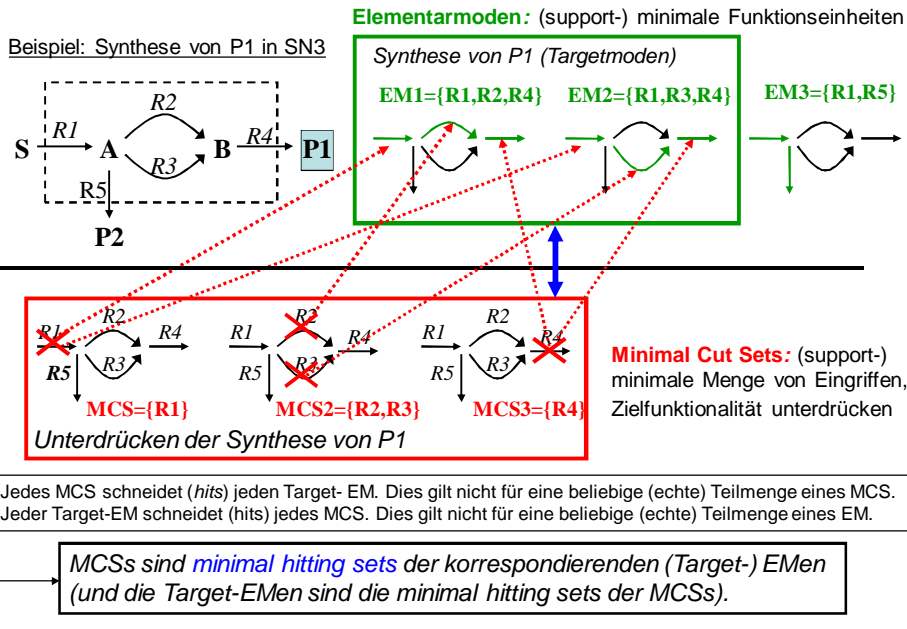
Weitere Beispiele für MCS in SN3:

- MCSs für Synthese von P2
- MCSs zum Unterdrücken aller Flussverteilungen



Minimal Cut Sets können aus EMen berechnet werden ...

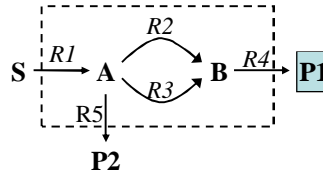
Dualität: Minimal cut sets und Elementarmoden



Ungerichtete Hypergraphen und Minimal Hitting Sets

- **Erinnere:** ungerichteten Hypergraphen $H=(V,E)$:
 V = Knoten (Grundmenge an Objekten);
 E = Hyperkanten = Teilmenge der Knoten
-
- ein **hitting set** c ist Teilmenge von V , die jede Hyperkante in E schneidet:
 $c \subset V$ und $\forall e \in E: c \cap e \neq \emptyset$
 - c ist **minimal hitting set (MHS)** wenn c hitting set ist, jedoch keine Teilmenge von c
 - endlicher Hypergraph hat eindeutige und endliche Menge an MHSs \rightarrow diese spannen den **transversalen hypergraphen** von H : $H_{MHS}=tr(H)$
 - **hier:** die MCSs sind die MHSs des Hypergraphen H_{EM} der aus V ={alle Reaktionen} und E ={alle Targetmoden} besteht (beachte: die Reaktionen sind die Knoten !!)
 - umgekehrt sind die EMen die MHSs des Hypergraphen H_{MCS} mit V ={alle Reaktionen} und E ={alle MCSs} $\rightarrow H_{EM}$ und H_{MCS} sind **dual** zueinander
 - vielfältige Anwendungen für MHSs auch in anderen Bereichen

Elementarmoden, Minimal Cut Sets und Minimal Hitting Sets

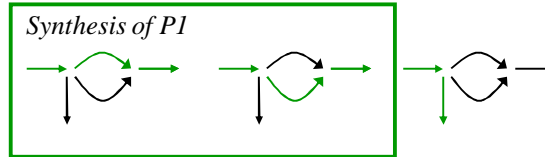


- die EMen spannen einen Hypergraphen H_{EM} auf mit
 - V = Menge der Reaktionen
 - E = Menge der Targetmoden

$$H_{EM} = (V, E_{EM})$$

$$V = \{R1, R2, R3, R4\}$$

$$E_{EM} = \{\{R1, R2, R4\}, \{R1, R3, R4\}\}$$

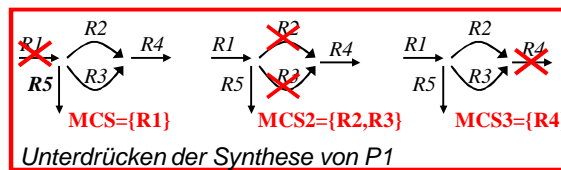


- die MCSs sind die MHSs von H_{EM} und spannen den transversalen Hypergraphen H_{MHS} auf:

$$H_{MHS} = tr(H_{EM}) = (V, E_{MHS})$$

$$V = \{R1, R2, R3, R4\}$$

$$E_{MHS} = \{\{R1\}, \{R2, R3\}, \{R4\}\}$$



Unterdrücken der Synthese von P1

Es gilt: $H_{EM} = tr(H_{MHS}) = tr(tr(H_{EM}))$

Minimal Hitting Sets

- Berechnung von Minimal Hitting Sets: z.B. *Berge-Algorithmus*

```

% gegeben: Hypergraph H=(V,E)
% (in unserem Fall ist V=Menge der Reaktionen, E=Menge der EMen)

% berechnet: Menge M der MHSs für alle Hyperkanten e in E

initial MHS collection M = { }
for each set e in E
  for each set m in M
    if m and e are disjoint
      remove m from M
      for each element k in e
        add (m union k) to M %Kombinationsschritt
      end
    end
  end
end
test minimality of all new m in M % Minimalität testen
end
    
```

Bsp: Berechnung der MCS (als MHS der EMs) zur Blockierung der P-Synthese in SN3

Schritt 1: EM1
EM1={R1,R2,R4}

MHS1={R1}
MHS2={R2}
MHS3={R4}

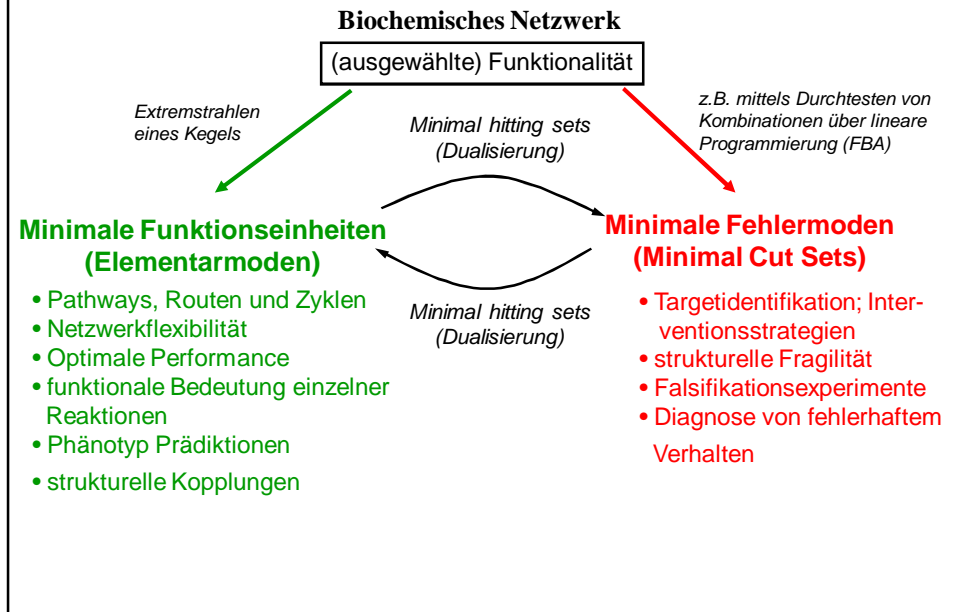
Schritt 2: EM2
einarbeiten
EM2={R1,R3,R4}

HS1={R1}
~~HS2={R2,R4}~~
HS3={R2,R3}
~~HS4={R2,R4}~~
HS5={R4}

HS2 und HS4 sind nicht minimal und werden deshalb verworfen. HS1, HS3 und HS5 sind die gesuchten MHSs (MCSs)

- aufwändig, insbesondere wenn Hypergraph viele Hyperkanten hat (erinnere: hohe Anzahl an EMen!);
Möglichkeit: beschränke Suche auf MHS mit vorgegebener Schranke für Kardinalität

Minimal cut sets und Elementarmoden



Beispiel: MCSs im *E. coli* Zentralstoffwechsel

Deletionsziel: "Blockiere Wachstum"

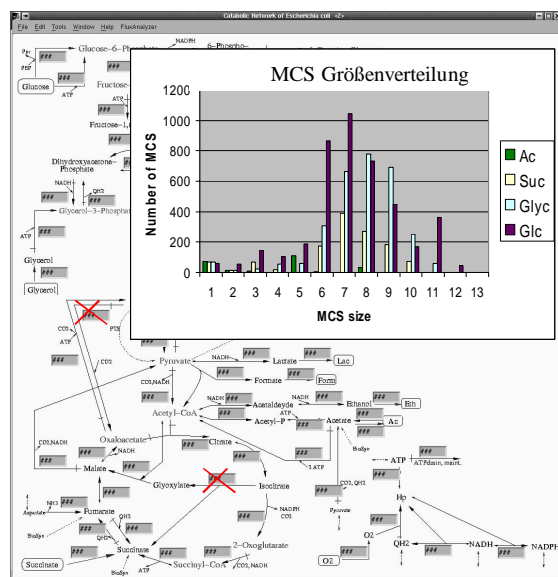
4 Substrate

	#EMs	#EMs with μ	#MCS
Ac	598	363	245
Succ	4250	3421	1255
Glyc	11333	9479	2970
Gluc	27100	21592	4225

MCSs (+ EMs) hängen stark vom Substrat (Umgebung) ab.

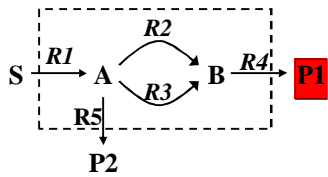
Beispiel:
{PEP-Carboxylase, Isocitrat-Lyase}

- MCS für Glukose und Glycerol
- cut set aber nicht minimales für Wachstum auf Acetat
- kein cut set für Succinat

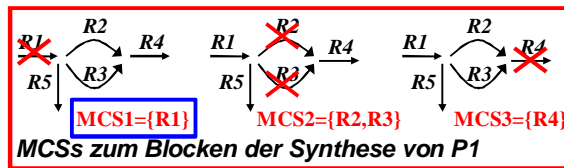


III.4.6 Minimal Cut Sets mit Nebenbedingungen (constrained MCSs)

- MCSs blockieren die „Targetmoden“, können aber auch unerwünschte Nebeneffekte erzeugen



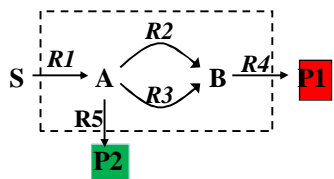
Blockiere Synthese von P1!



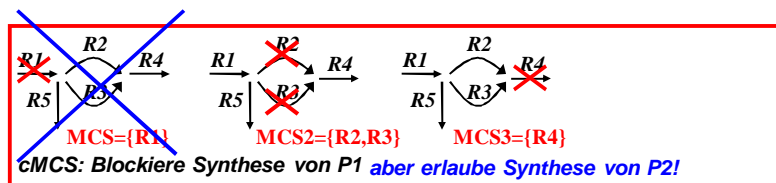
Bei MCS1: Synthese von P2 nicht mehr möglich!

Minimal Cut Set (MCS): (*support-*) minimale Menge von Interventionen (cuts) die unerwünschte Flussverteilungen (Funktionen) blocken

Constrained MCSs: Minimal Cut Sets mit Nebenbedingungen



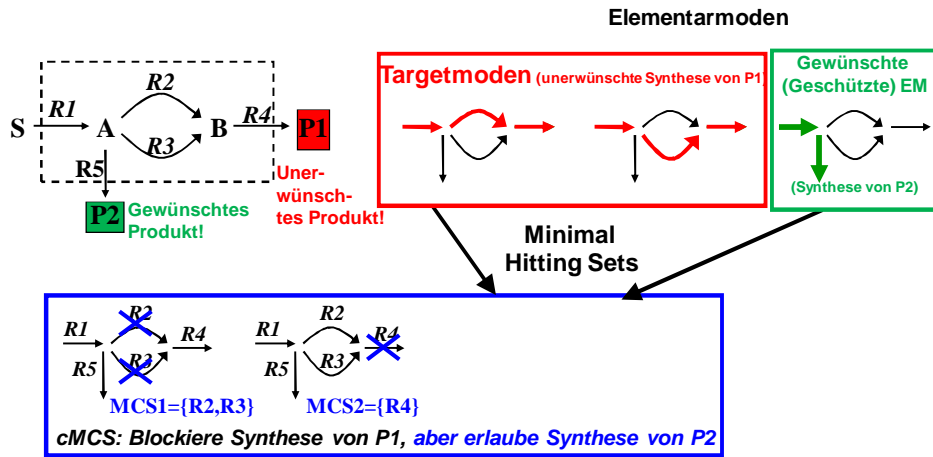
Blockiere Synthese von P1!
Erlaube Synthese von P2!



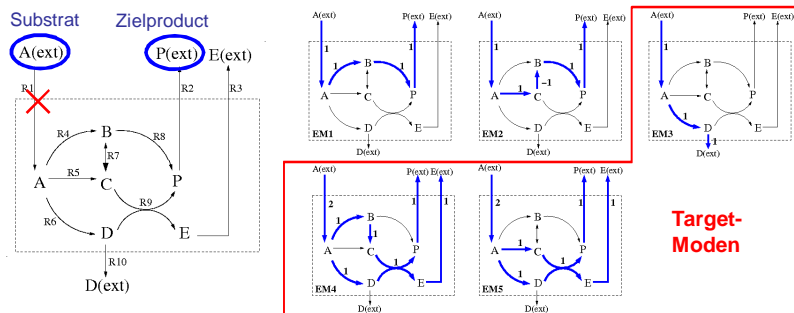
Constrained Minimal Cut Sets: (*support-*) minimale Menge von Interventionen (cuts), die unerwünschte Flussverteilungen (Funktionen) blocken **und gewünschte erhalten**

Unerwünschte und erwünschte Flussverteilungen werden nun über **zwei** Mengen von Elementarmoden spezifiziert.

Constrained MCSs: Minimal Cut Sets mit Nebenbedingungen



**Beispiel: Interventionsziel sei
alle nicht-optimalen Produktionswege für P zu blocken**



→ MCS1={R6}, MCS2={R1}, MCS3={R2, R10}, MCS4={R9,R10}, MCS5={R3,R10}, MCS6={R4,R5,R10}, MCS7={R5,R7,R10}

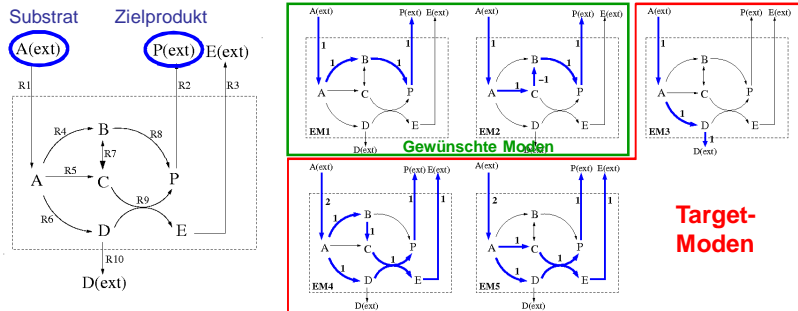
→ einige MCSs produzieren ungewollte Nebeneffekte und blocken nicht nur die Targetmoden sondern auch gewünschte EMen (z.B. ist nach KO von R1 Synthese von P gar nicht möglich)

→ Betrachte MCSs mit Nebenbedingungen (constrained MCSs; cMCSs): definiere Menge **T** der Target-Moden und Menge **D** der gewünschten (desired) Moden und spezifiziere Mindestanzahl **n** an EMen in **D** die nicht durch die cMCSs eliminiert werden dürfen

**Beispiel: Interventionsziel sei
alle nicht-optimalen Produktionswege für P zu blocken**

Original: Blocke alle nicht-optimalen Produktionswege für P!

Neu: Blocke alle nicht-optimalen Produktionswege für P und verschone mindestens einen ($n=1$) oder beide ($n=2$) optimalen Produktionswege!



Target EMen: $T=\{EM3,EM4,EM5\}$ nicht-optimale EMen für Synthese von P)

Gewünschte EMen: $D=\{EM1,EM2\}$ (alle optimalen EMen für Synthese von P)

Fall 1) $n=1$ (mindestens ein EM von D darf nicht getroffen werden):

→ $MCS1=\{R6\}$, $MCS2=\{R9,R10\}$, $MCS3=\{R3,R10\}$, $MCS4=\{R5,R7,R10\}$

Fall 2) $n=2$ (beide EMen in D dürfen nicht getroffen werden):

→ $MCS1=\{R6\}$, $MCS2=\{R9,R10\}$, $MCS3=\{R3,R10\}$

Verallgemeinerung: MCSs mit Nebenbedingungen

- **Algorithmus** für minimal hitting set Berechnung kann leicht angepasst werden (checke nach jeder Iteration, ob das vorläufige MCSs die Nebenbedingung bzgl. der gewünschten EMen erfüllt)
 - die Menge der **zulässigen MCSs** wird kleiner
 - Berechnung der cMCSs daher oft schneller
- **große Vielfalt an komplexen Interventionsstrategien** kann über die cMCSs formuliert und berechnet werden
- viele andere Methoden (z.B. OptKnock oder RobustKnock, siehe III.4.3) können als **Spezialfälle der cMCS** formuliert werden
- **alle minimalen knock-out Strategien** können berechnet werden (... wenn das Netzwerk nicht zu groß ist)

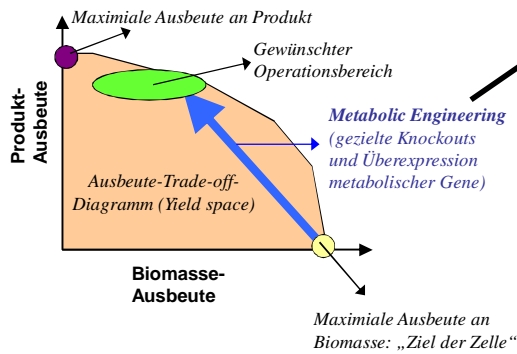
III.4.7 Anwendung von MCS: Metabolic Engineering: (Re-)Design zellulärer Fabriken



Stöchiometrisches Netzwerkmodell

Von Interesse: Produktsynthese aus bestimmtem Substrat

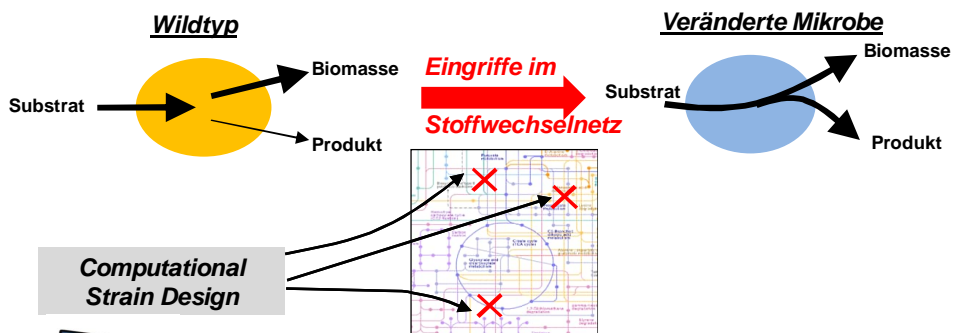
Modell-gestützte Suche nach genetischen Interventionen, die ein gegebenes Interventionsziel erfüllen.



Designstrategie?

Kopplung von Wachstum und Produktsynthese als eine Metabolic Engineering Strategie

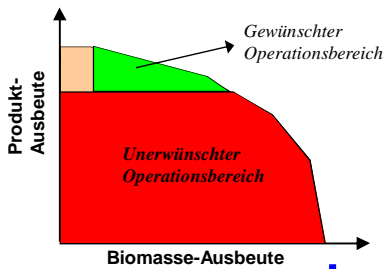
Stöchiometrische Kopplung von Wachstum und Produktsynthese
→ **gewünschtes Produkt wird (essenzielles) Nebenprodukt des Wachstums**



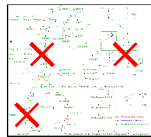
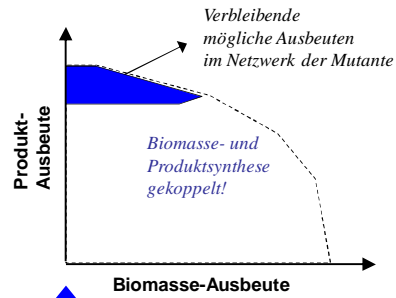
- **Bilevel/Trilevel-Optimierungsprobleme:** OptKnock, RobustKnock (siehe vorne)
- **Minimal Cut Sets ...**

Stammdesign mit Constrained Minimal Cut Sets für die wachstumsgekoppelte Produktsynthese

(1) Berechne Ausbeute-Diagramm und spezifiziere **gewünschtes** und **ungewünschtes** Verhalten



(3) Beispiel nach Implementation eines Minimal Cut Sets



(2) Berechnung aller minimalen Knockout-Kombinationen (Constrained Minimal Cut Sets), die **unerwünschte** Stoffflussverteilungen eliminieren und (einige) **gewünschte** beibehalten

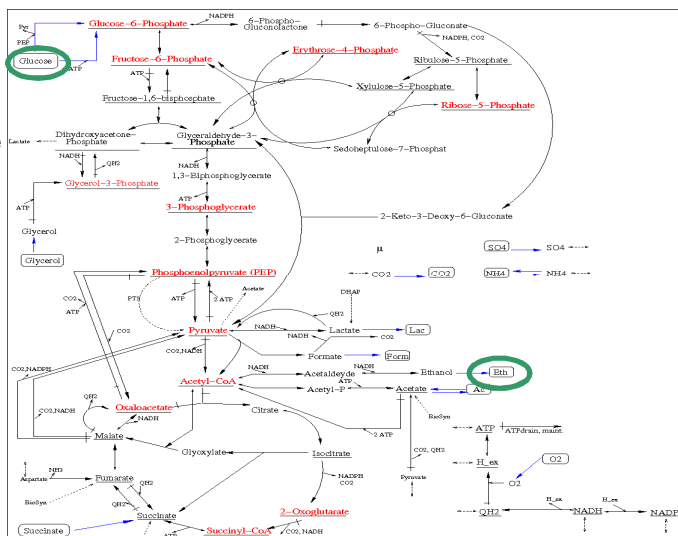
Beispiel: Kopplung von Biomasse- und Ethanolsynthese in *E. coli*

Ziel: suche nach Interventionsstrategien die zwangsläufig zur (anaeroben) Synthese von Ethanol aus Glukose führen und dabei der Zelle noch Biomassesynthese ermöglichen!

Vorgaben:

Ethanol-Ausbeute:
> 1.4 mol/mol Glukose

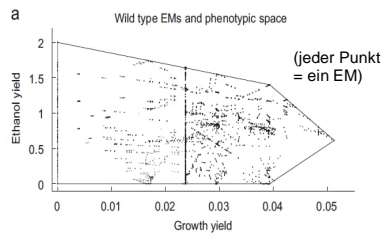
Biomasse-Ausbeute:
> 20 g/mol Glukose



Beispiel: Kopplung von Biomasse- und Ethanolsynthese in *E.coli*

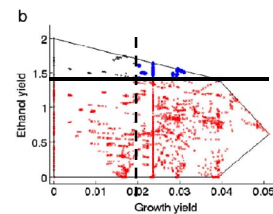
Ziel: suche nach Interventionsstrategien die zwangsläufig zur (anaeroben) Synthese von Ethanol führen und dabei der Zelle noch eine Mindestmenge an Biomassesynthese erlauben!

1) Berechne EMen und analysiere Ausbeute-Diagramm



2) Selektiere Target- und gewünschte EMen

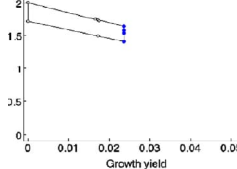
Rote Punkte = Target EMen
Blaue Punkte = gewünschte EMen
Schwarze Punkte = „egal“



3) Berechne cMCSs und wähle den geeignetsten aus

- hier: n=1 (mind. ein gewünschter EM muss „überleben“)
- insgesamt 1988 cMCS
- mindestens 5 knockouts notwendig

Beispiel für verbleibendes Ausbeutediagramm in einer berechneten Fünffach-Mutante



- hohe Produktausbeute garantiert
- vernünftige Biomasseausbeute weiterhin möglich

Beispiel: Kopplung von Biomasse- und Ethanolsynthese in *E.coli*

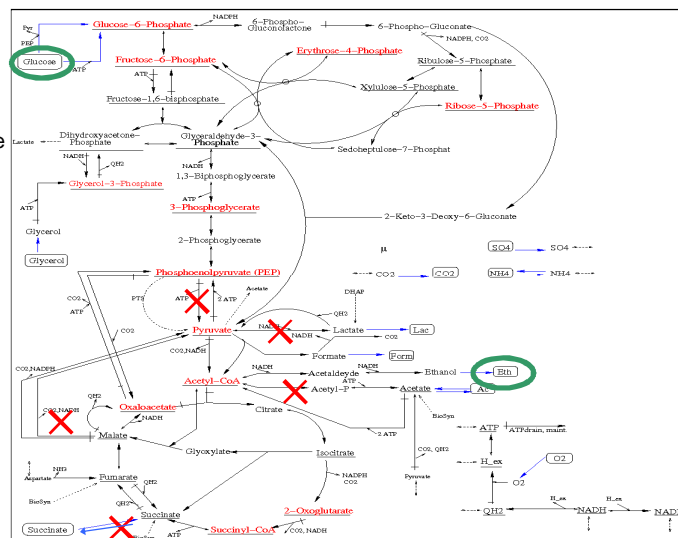
Ziel: suche nach Interventionsstrategien die zwangsläufig zur (anaeroben) Synthese von Ethanol führen und dabei der Zelle noch eine Mindestmenge an Biomassesynthese erlauben!

Vorgaben:

Ethanol-Ausbeute:
> 1.4 mol/mol Glukose

Biomasse-Ausbeute:
> 20 g/mol Glukose

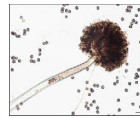
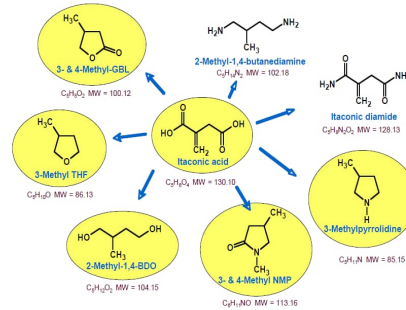
- mind. 5 knockouts
- ca. 2000 Minimal Cut Sets lösen das Problem



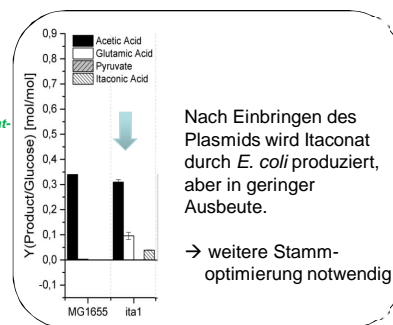
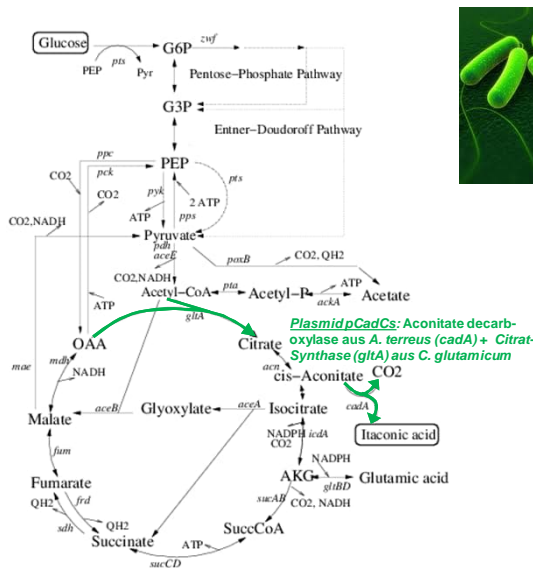
Beispiel: Stammoptimierung von *E. coli* zur Itaconsäuresynthese

Itaconsäure

- Plattform-Chemikalie; insbes. für Polymersynthese
- Bisher Synthese durch filamentöse Organismen (*Aspergillus terreus*)
→ Nachteile: Wachstum langsam; sauerstoffsensitiv, ...
- Prozess mit *E. coli* als unkompliziertes Lieblings-„Arbeitspferd“ wünschenswert.
- Derzeit nur relativ geringe Ausbeuten bei der Synthese von Itaconsäure aus Glukose mittels *E. coli*.

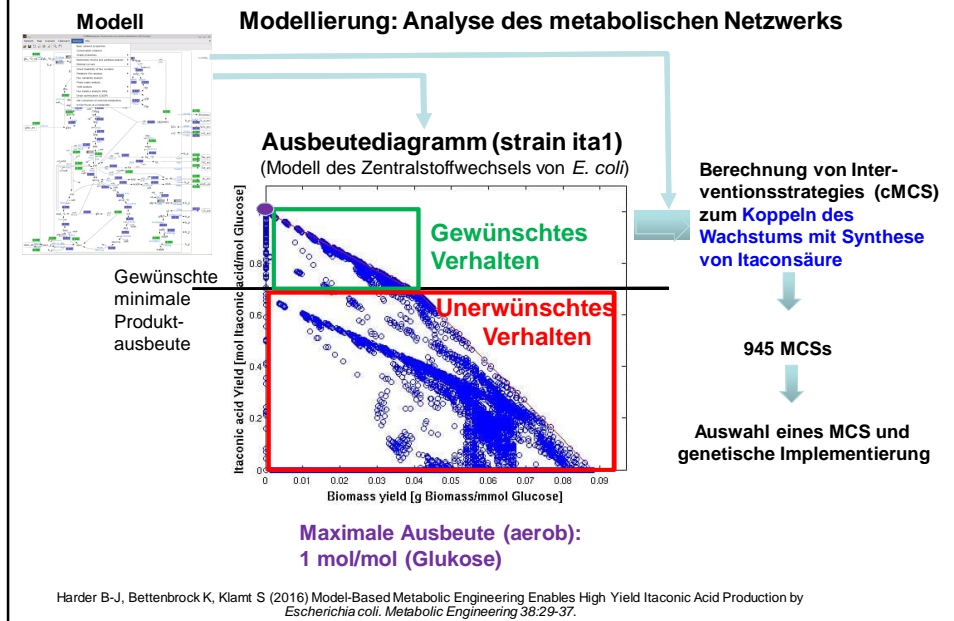


Beispiel: Stammoptimierung von *E. coli* zur Itaconsäuresynthese

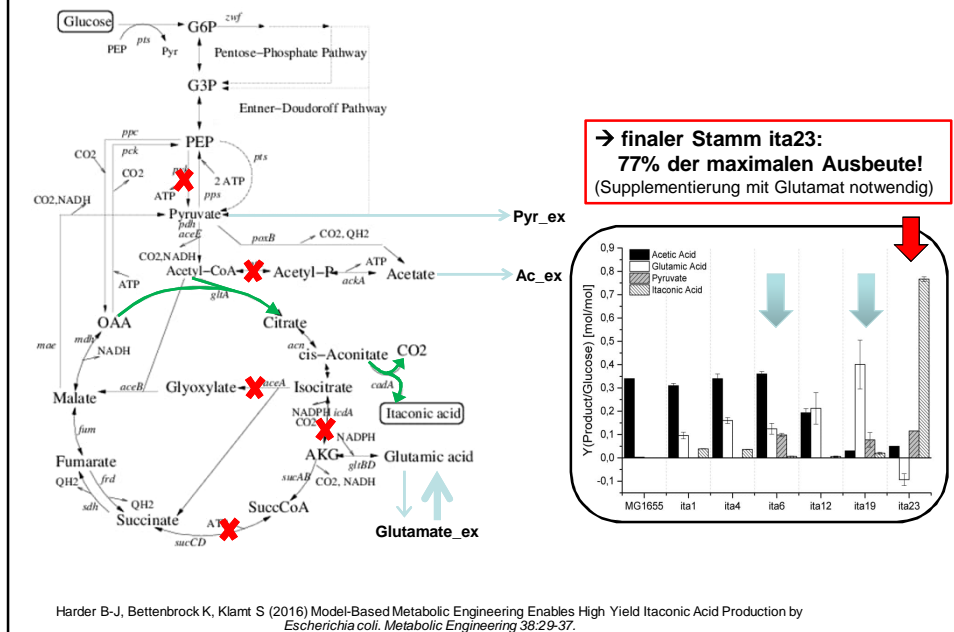


Harder B-J, Bettenbrock K, Klamt S (2016) Model-Based Metabolic Engineering Enables High Yield Itaconic Acid Production by *Escherichia coli*. *Metabolic Engineering* 38:29-37.

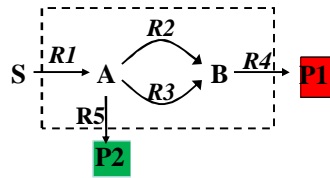
Beispiel: Stammoptimierung von *E. coli* zur Itaconsäuresynthese



Beispiel: Stammoptimierung von *E. coli* zur Itaconsäuresynthese



III.4.7 Alternative Berechnung von (Constrained) Minimal Cut Sets



Blockiere Synthese von P1!
Erlaube Synthese von P2!

Berechnung von constrained MCS: spezifiziere **gewünschte (geschützte)** und **unerwünschte** Verhaltensweisen (Flussverteilungen)

in kleinen Netzen

Berechne EMen und spezifiziere dann **gewünschte EM** und **Targetmoden**

Berechnung der MCS
 via Minimal Hitting Sets

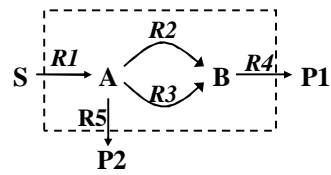
in kleinen und großen Netzen

Spezifiziere **gewünschtes/unerwünschtes** Verhalten via (Un)gleichungen

Beispiel: **Unerwünscht: $R4 \geq 1$**
Gewünscht: $R5 \geq 1$

Berechnung der MCSs via
 Dualitätsalgorithmus

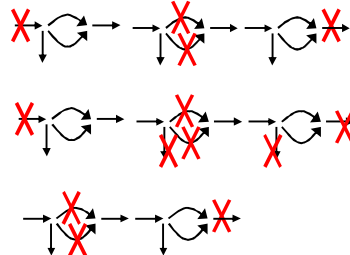
Berechnung von Minimal Cut Sets via Ungleichungen und Dualität



Beispiele

- 1) Blockiere Synthese von P1:
 Unerwünscht: $R4 \geq 1$
 Gewünscht: (Nichts)
- 2) Blockiere Synthese von P1 und P2:
 Unerwünscht: $R4 + R5 \geq 1$
 Gewünscht: (Nichts)
- 3) Blockiere Synthese von P1 +
 ermögliche Synthese von P2:
 Unerwünscht: $R4 \geq 1$
 Gewünscht: $R5 \geq 1$

Lösungen



Berechnung von Minimal Cut Sets via Ungleichungen and Dualität

Die spezifizierten Ungleichungen für **gewünschte (D)** und **unerwünschte (T)** Flussverteilungen kann man nutzen, um gemischt-ganzzahlige lineare Optimierungsprobleme (Mixed-Integer Linear Programming / MILP) zu formulieren, mit denen MCSs berechnet werden können.

Der Vorteil dieses Ansatzes ist, dass die EMen nicht in einem ersten Schritt bestimmt werden müssen, was die Berechnung der kleinsten MCSs (mit den wenigsten Eingriffen) auch in genomskaligen Netzwerken ermöglicht.

$$\text{minimize } \sum_i z_i$$

$$\text{s. t.}$$

$$\begin{pmatrix} \mathbf{N}^T & \mathbf{I} & \mathbf{T}^T & 0 \\ 0 & 0 & \mathbf{t}^T & 0 \\ 0 & 0 & 0 & \mathbf{N} \\ 0 & 0 & 0 & \mathbf{D} \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ \mathbf{v} \\ \mathbf{w} \\ \mathbf{r} \end{pmatrix} = \begin{pmatrix} 0 \\ -c \\ 0 \\ \mathbf{d} \end{pmatrix}$$

$$\forall i \in \text{Irrev}: z_i = 0 \rightarrow v_i \leq 0$$

$$\forall i \in \text{Rev}: z_i = 0 \rightarrow v_i = 0$$

$$(1 - z_i) \cdot lb_i \leq r_i \leq (1 - z_i) \cdot ub_i$$

$$u \in \mathbb{R}^m, t \in \mathbb{R}^t, w \in \mathbb{R}_{\geq 0}^t, v, r \in \mathbb{R}^n, c > 0, z_i \in \{0,1\}$$

von Kamp A, Klamt S (2014) Enumeration of smallest intervention strategies in genome-scale metabolic networks. PLoS Computational Biology 10(1):e1003378.

Berechnung von MCS für wachstumsgekoppelte Produktsynthesen via Ungleichungen und Dualität

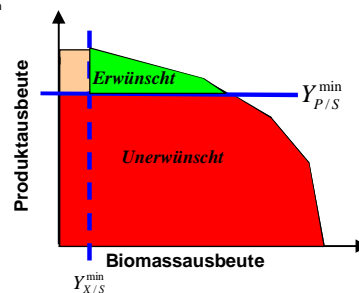
Blockiere Flussvektoren mit geringer Produktausbeute unter $Y_{P/S,\min}$ und erhalte (einige) Flussvektoren mit minimaler Biomasseausbeute $Y_{X/S,\min}$ (und dann hoher Produktausbeute):

$$\text{Unerwünscht: } Y_{P/S} = r_p / r_s \leq Y_{P/S,\min} \rightarrow r_s \cdot Y_{P/S,\min} - r_p \geq 0$$

$$\text{Erwünscht } Y_{X/S} = \mu / r_s \geq Y_{X/S,\min} \rightarrow r_s \cdot Y_{X/S,\min} - \mu \leq 0$$

(Alternativ könnte man auch eine minimale Wachstumsrate $\mu \geq \mu_{\min}$ fordern)

Für beide Ungleichungen (unerwünscht/gewünscht) müssen wir noch sicherstellen, dass die Substrataufnahme größer Null ist (sonst müssten wir auch den Nullvektor blockieren, was nicht möglich ist). Dazu kann man z.B. $r_s \geq 0.01$ bei beiden Ungleichungen hinzufügen. Das ist nicht nötig wenn der ATP Erhaltungsbedarf (ATPM) einen Wert größer Null hat, da dann eine Substrataufnahme bereits garantiert ist. Dies ist in den meisten Modellen so gegeben.



von Kamp A, Klamt S (2014) Enumeration of smallest intervention strategies in genome-scale metabolic networks. PLoS Computational Biology 10(1):e1003378.

Berechnung von MCS für wachstumsgekoppelte Produktsynthesen via Ungleichungen und Dualität

Beispiel: Suche nach MCS die wachstumsgekoppelte Synthese von Ethanol mit $Y_{\text{Eth/Glc,min}} = 1.4$ erzwingen mit möglicher Wachstumsrate von mindestens 0.05 h^{-1}

Unerwünscht:
 $1.4 * r_{\text{Glc_up}} - r_{\text{Eth_ex}} \geq 0$
 $r_{\text{Glc_up}} \geq 0.01$

Erwünscht:
 $\mu \geq 0.05 \text{ h}^{-1}$

Spezifikation im CellNetAnalyzer

module	target	value	direction
1.4 Glc_up - Eth_ex	T	0	>=
1 Glc_up	T	0.0100	>=

module	target	value	direction
1 mue	D	0.0500	>=

Gene KOs:
 Max. Solutions: 10
 Max. Size: 7
 Time Limit [sec]: inf

Solver: CPLEX
 intlinprog
 CLPK
 gurobi

MCS search: any MCS (fast)
 by cardinality
 smallest MCS first

Consider constraints given in text boxes.
 Advanced: Define knockout/addition costs for genes/reactions

Compute MCS Save Load Cancel

Referenzen

- Klamt S, Hädicke O, von Kamp A (2014) Stoichiometric and Constraint-Based Analysis of Biochemical Reaction Networks. Large-Scale Networks in Engineering and Life Sciences. Edited by Benner P, Findeisen R, Flockerzi D, Reichl U and Sundmacher K, Springer, pp. 263-316 (online verfügbar)
- Price ND, Reed JL and Palsson BØ. 2004. Genome-scale Models of Microbial Cells: Evaluating the consequences of constraints. Nature Reviews Microbiology, 2:886-897.
- Durot M, Bouruignon PY, Schachter V. 2009. Genome-scale models of bacterial metabolisms: reconstruction and applications. FEMS Microbiol Rev 33:164-190.
- Trinh CT, Wlaschin A, Sreenc F. 2009. Elementary mode analysis: a useful metabolic pathway analysis tool for characterizing cellular metabolism. Appl Microbiol Biotechnol. 81:813-826.
- Klamt S. 2006. Generalized concept of minimal cut sets in biochemical networks. Biosystems, 83:233-247.
- Gagneur J and Klamt S. 2004. Computation of elementary modes: a unifying framework and the new binary approach. BMC Bioinformatics, 5:175. <freely available at BMC Bioinformatics>
- Stelling J, Klamt S, Bettenbrock K, Schuster S and Gilles ED. 2002. Metabolic network structure determines key aspects of functionality and regulation. Nature, 420: 190-193.
- Klamt S, Schuster S and Gilles ED. 2002. Calculability analysis in underdetermined metabolic networks illustrated by a model of the central metabolism in purple nonsulfur bacteria. Biotechnology & Bioengineering 77 (7): 734-751
- Stephanopoulos G, Aristidou, A Nielsen J. 1998. Metabolic Engineering. Academic Press, San Diego.

Referenzen (2)

- Burgard AP, et al. 2003. Optknock: a bilevel programming framework for identifying gene knockout strategies for microbial strain optimization. *Biotechnol. Bioeng* 84:647–657.
- Tepper N and Shlomi T. 2010. Predicting metabolic engineering knockout strategies for chemical production: accounting for competing pathways. *Bioinformatics* 26(4):536-543.
- Feist AM, Zielinski DC, Orth JD, Schellenberger J, Herrgard MJ and Palsson BO. 2010. Model-driven evaluation of the production potential for growth-coupled products of *Escherichia coli*. *Metabolic Engineering*
- Thiele I and Palsson BO. 2010. A protocol for generating a high-quality genome-scale metabolic reconstruction, *Nature Protocols*, 5: 93-121.
- Hädicke and Klamt (2011): Computing complex metabolic intervention strategies using constrained minimal cut sets. *Metabolic Engineering*, 13: 204-213.
- Harder B-J, Bettenbrock K, Klamt S (2016) Model-Based Metabolic Engineering Enables High Yield Itaconic Acid Production by *Escherichia coli*. *Metabolic Engineering* 38:29-37.
- Bekiaris PS, Klamt S (2020) Automatic construction of metabolic models with enzyme constraints. *BMC Bioinformatics* 21:19.
- Klamt S, von Kamp A (2022) Analyzing and Resolving Infeasibility in Flux Balance Analysis of Metabolic Networks. *Metabolites* 12: 585.

Übungen

- 1) Laden Sie eine genomskaliges E.coli-Stoffwechselnetz in Cytoscape:
 - Cytoscape: /data/bio/werkzeuge/Cytoscape-v2.5
 - Datei: Ec_iJR904.xml (laden unter Load/Import/Network(multiple file ...))Wählen Sie unter Layout/Cytoscape Layouts/Spring Embedded aus.
Wie zeigt sich hier die Eigenschaft skalenfreier Netzwerke?
- 2) Untersuchen Sie die Konnektivität eines genomskaligen E.coli-Netzes
 - a) Kopieren Sie die Datei /data/mpi_meeting/pclabor/zellnet/coli.mat in Ihr pclaborXX home Verzeichnis
 - b) Laden Sie die Datei in MATLAB mit *load coli*
 - geladene Variable *colinet* ist eine Struktur und enthält als Felder u.a. *colinet.stoichMat*
 - c) Schauen Sie sich mit dem Befehl *spy* die Struktur der Matrix an
 - d) bestimmen sie für jeden Metaboliten (Zeile in der stöch. Matrix) die (totale) Konnektivität
 - e) suchen Sie nach den 20 größten Konnektivitäten (MATLAB Befehl *sort*) und schauen Sie in der Namensliste *colinet.specID* (in den Zeilen stehen die Namen) nach, um welche Metabolite es sich dabei handelt
 - f) stellen sie die Konnektivitäten im Histogramm dar (Befehl *hist*; 10 Klassen)
 - g) wie kann man aus den Daten den Exponenten der Potenzgesetzverteilung der Konnektivitäten bestimmen?

Übungen

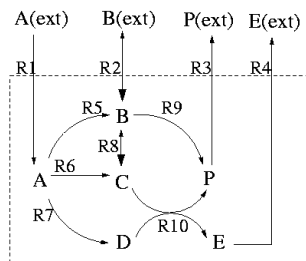
3) Ein Reaktionsnetzwerk bestehe aus den 2 Reaktionen



- Wieviele lin. unabh. Erhaltungsrelationen hat das System?
- Versuchen Sie welche zu erraten (ohne zu rechnen).
- Rechnen Sie eine Basis manuell (mit Papier und Bleistift) aus!
- Benutzen Sie dann MATLAB um eine Basis für die Erhaltungsrelationen zu bestimmen.

Übungen

Netzbeispiel SN2:



$$N = \begin{bmatrix} 1 & 0 & 0 & 0 & -1 & -1 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & -1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \begin{matrix} \leftarrow A \\ \leftarrow B \\ \leftarrow C \\ \leftarrow D \\ \leftarrow E \\ \leftarrow P \end{matrix}$$

$\begin{matrix} \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow \\ R1 & R2 & R3 & R4 & R5 & R6 & R7 & R8 & R9 & R10 \end{matrix}$

$$\begin{aligned} rev &= \{R2, R8\} \\ irrev &= \{R1, R3, R4, R5, R6, R7, R9, R10\} \end{aligned}$$

4) Bestimmen Sie durch "bloßes Anschauen" Redundanz und Bestimmtheit der folgenden Flusszenarien und "erraten" Sie für alle nichtredundanten Szenarien die eindeutig berechenbaren Raten im steady state:

(Hinweis: es existieren keine Erhaltungsrelationen in SN2)

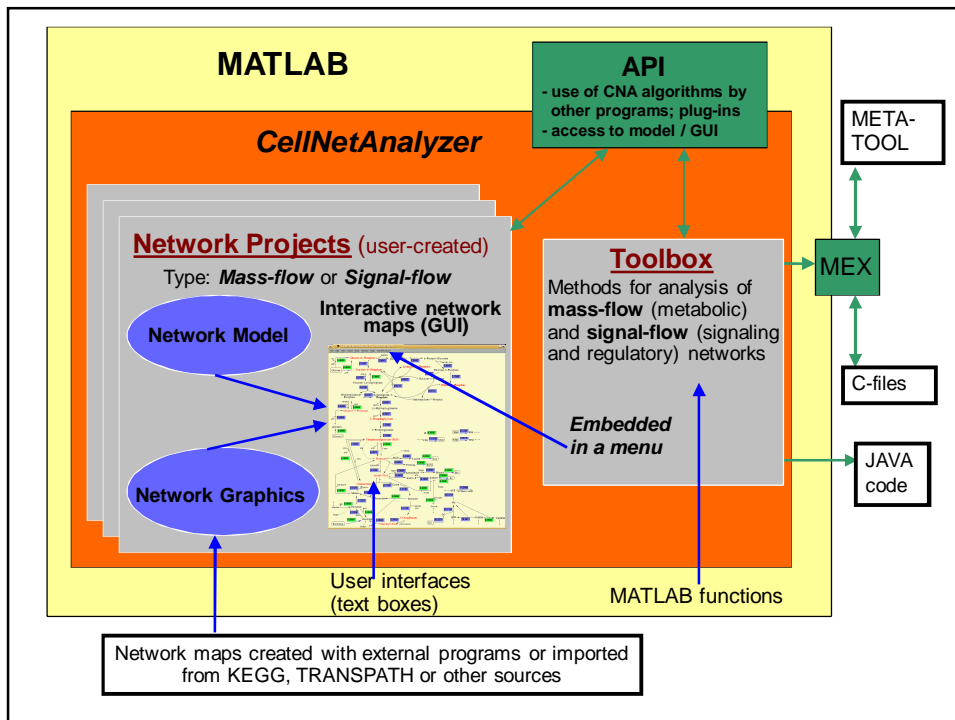
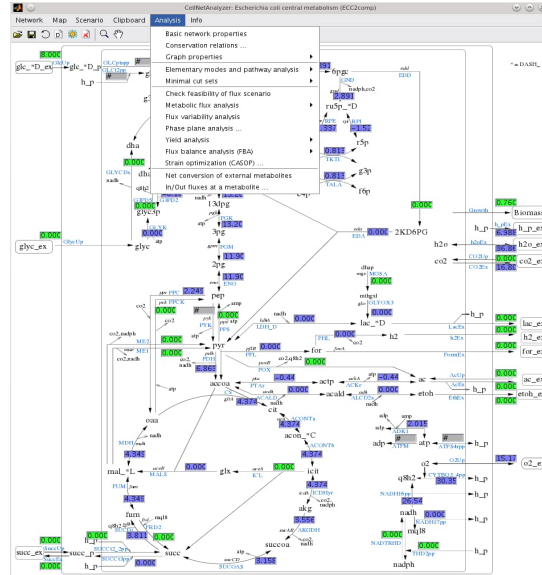
- $R1=3$
- $R1=3, R4=2$
- $R1=3, R4=2, R3=2$
- $R1=3, R4=2, R3=2, R7=2$
- $R1=3, R4=2, R3=2, R5=0$

Bestimmen Sie außerdem für jedes Szenario den gültigen Wertebereich für $R5$, der sich jeweils nach der Flussvariabilitätsmethode ergeben würde.

Übungen: Metabolische Netzwerkanalyse mit CellNetAnalyzer

www.mpi-magdeburg.mpg.de/projects/cna/cna.html (freier Download)

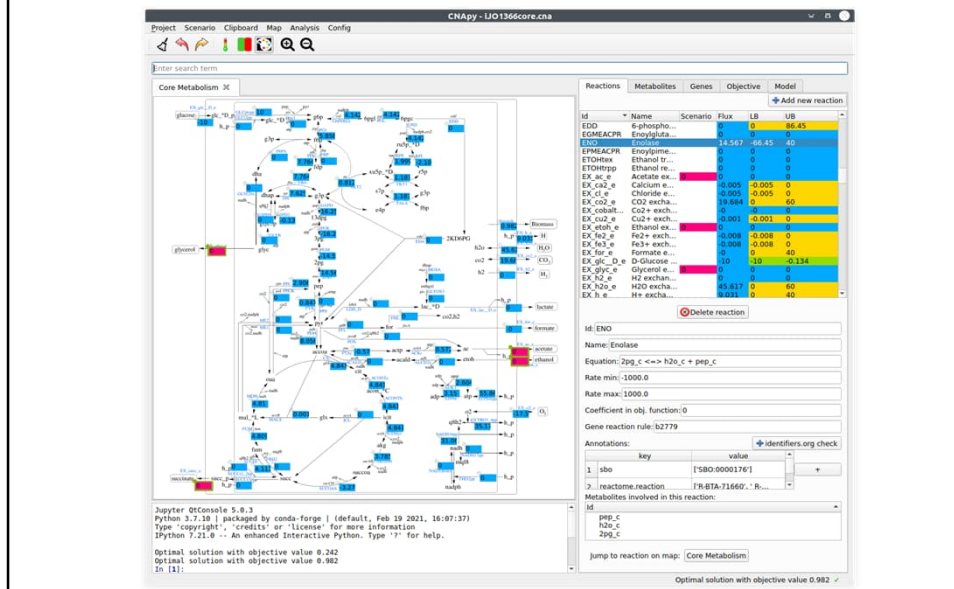
- Graphische Benutzeroberfläche für MATLAB
- metabolische + Signaltransduktionsnetze
- *Netzwerk-Projekte*: Modell + interaktive Netzarten
- *Toolbox* (Menü) für metabolische Netzwerkanalyse
 - Erhaltungsrelationen
 - Flussanalyse
 - Flussoptimierung
 - Elementarmodenanalyse
 - Minimal Cut Sets
 - ...



Neues Tool: CNapy (CellNetAnalyzer in Python)

<https://github.com/cnapy-org/CNapy> (freier Download)

Ähnliche Handhabung wie in CNA aber modernere Benutzeroberfläche (selber Ausprobieren!).



Übungen

- 5) Laden Sie CellNetAnalyzer (CNA) in MATLAB. Gehen Sie dazu ins Verzeichnis `/data/mpl_meeting/plclabor/zellnet/cna` und starten Sie in MATLAB `startcna`. Machen Sie sich mit den Funktionalitäten vertraut.

[Sie können sich CNA für Ihren privaten Rechner runterladen über:

www.mpi-magdeburg.mpg.de/projects/cna/cna.html

- 6) Starten sie im CNA das Netzwerk „Stoich. Network Example 2“. Überprüfen Sie Ihre Ergebnisse aus Übung 4.
- 7) Starten Sie das Netzwerk „EColiCore2compressed (Small maps)“ (vorletztes Modell im Project Manager) und berechnen Sie die Flussverteilung, die sich aus dem voreingestellten Szenario bei Optimierung (mit „Flux Balance Analysis“) im Modell ergibt. Interpretieren Sie die Flussverteilung physiologisch!

Betrachten Sie dann Wachstum auf **Glycerol** (die Substrate Glukose, Succinat, und Acetat sollen nicht zur Verfügung stehen). Berechnen/beantworten Sie dann

- die **maximale Wachstumsrate** (Reaktion „growth“)
- die **maximal Ethanolsyntheserate** („R_EthEx“)
- Kann ***E. coli anaerob*** (ohne Sauerstoffaufnahme) auf Glycerol wachsen?

Übungen

7) (fortgesetzt)

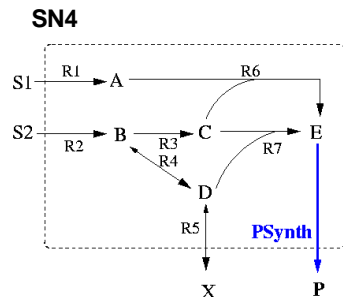
- d) Für welche der 5 Substrate (Glukose, Succinat, Acetat, Glycerol) ist die Reaktion **der Phosphoglyceratkinase (Reaktion R_PGK)** essenziell für Wachstum?
- e) Wieviel **mmol ATP kann *E. coli* aerob und wieviel anaerob** aus einem mmol Glukose machen?
- f) Limitieren Sie die maximale **Sauerstoffaufnahme rate auf 5 mmol/gDW/h** und betrachten Sie Wachstum auf Glukose (max. Aufnahme rate von 10). Was ist die maximal erreichbare Wachstumsrate und welche Biomasseausbeute wird da erreicht? Maximieren Sie dann für dasselbe Szenario die Biomasseausbeute! Was sind wesentliche Unterschiede zwischen den beiden resultierenden optimalen Flusserteilungen?
- g) Betrachten Sie Wachstum auf Glukose und nehmen Sie an, dass die **Wachstumsrate mindestens 0.1 h^{-1}** ist. Wie viele **Reaktionen im Modell sind essenziell?** (Hinweis: verwenden Sie **FVA**).

Übungen

- 8) Schreiben Sie eine Matlab Funktion *mfa*, die eine stöch. Matrix **N** und eine $q \times 2$ Matrix **B** übergeben bekommt. In **B** sollen in der ersten Spalte die Indizes der bekannten Reaktionsraten stehen und in der zweiten Spalte dann jeweils die eigentlichen Raten. *mfa* soll das Flusszenario klassifizieren (bzgl. Redundanz und Bestimmtheit) und im Falle von nichtredundanten Systemen die eindeutig berechenbaren Raten ausgeben.
- Tipp: *null(A)* berechnet Nullraum von A; *pinv(A)* die Pseudoinverse von A
rank(A) berechnet Rang von A

Übungen

9) Betrachten Sie Netzwerk SN4 (S1, S2, X, P sind externe Metabolite):



- Ermitteln Sie die (6) Elementarmoden (EMen) des Netzes (nur durch Anschauen und Überlegen, ohne zu rechnen)
- Gibt es im Netzwerk (essentielle) Reaktionen, die in allen stationären Flussverteilungen verwendet werden? Begründen Sie anhand der EMen!
- Gibt es essentielle Reaktionen für die Synthese von P? Begründen Sie anhand der EMen!

10)

- Berechnen Sie die Minimal Cut Sets (MCSs) zur Unterdrückung der Synthese von P in SN4 (ausgehend von den EMen - minimal hitting set Algorithmus)! Skizzieren Sie die Zwischenresultate bei der Berechnung der MCSs!
- Wie sieht die Menge T der Target-EMen und wie die Menge D der gewünschten EMen aus wenn die Synthese von X weiterhin möglich sein soll? Wie sehen die resultierenden MCSs aus, die diese Nebenbedingung erfüllen?

Berechnung von Minimal Cut Sets via Elementarmoden im CellNetAnalyzer

Panel zum Handeln der EMen

Wenn die Berechnung der MCS gestartet wird, werden die aktuell selektierten EMen als die (unerwünschten) Targetmoden betrachtet.

Gewünschte (geschützte) EMen müssen vorher selektiert und auf das Clipboard kopiert werden ...

Starte Berechnung der MCS

... wo aus sie dann bei der Berechnung der MCS als Input verwendet werden können.

Berechnung von Minimal Cut Sets via Ungleichungen und Dualität im CellNetAnalyzer

Die Dualitäts-basierte Berechnung von MCS kann bequem über eine Dialogbox gestartet werden, in der alle Ungleichungen und Parameter spezifiziert werden.

Beispiel: Suche nach MCS die wachstumsgekoppelte Synthese von Ethanol mit $Y_{\text{Eth/Glc, min}} = 1.4$ erzwingen mit möglicher Wachstumsrate von mindestens 0.05 h^{-1}

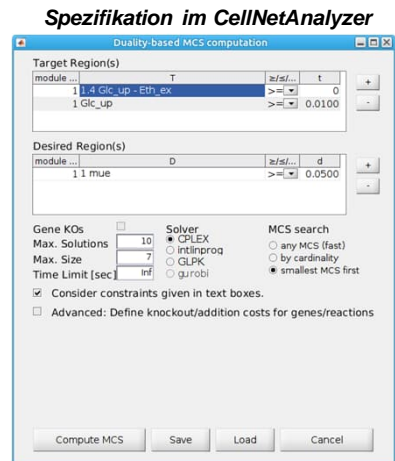
Unerwünscht:

$$1.4 * r_{\text{Glc_up}} - r_{\text{Eth_ex}} \geq 0$$

$$r_{\text{Glc_up}} \geq 0.01$$

Erwünscht:

$$\mu \geq 0.05 \text{ h}^{-1}$$



Übungen

- 11) Konstruieren Sie mit CellNetAnalyzer ein neues Netzwerk-Projekt, dass SN4 abbildet und überprüfen Sie Ihre Resultate von Übung 9!

(Hinweis: Sie können für die Netzwerk-Grafik einen Screenshot von der vorherigen Seite machen. „Schneiden“ Sie dann die Netzgrafik aus und speichern Sie sie in einer Grafikdatei (z.B. Endung PCX ab).

Berechnen Sie bei der Überprüfung der MCS diese sowohl via EMen als auch via Ungleichungen!

Übungen

- 12) Laden Sie im CNA wieder das „EColiCore2compressed (Small maps)“ Netzwerk. In diesem Modell gibt es 4 versch. Substrate (Glucose, Succinat, Glycerol, Acetat). Berechnen Sie für Wachstum auf **Succinat** und danach auf **Acetate** (separat! also jeweils alle anderen Substrataufnahmen schließen!) die EMen. Beantworten Sie damit jeweils:
- wieviele Moden gibt es?
 - was ist die maximale Biomasseausbeute für das Substrat?
 - gibt es mehrere optimale Lösungen für die Biomassesynthese?
 - kann E.coli mit dem Substrat anaerob wachsen?
 - kann eine Isocitratlyase-Mutante (Reaktion $R_{ICL}=0$) auf dem Substrat wachsen? Und eine Fructose-Bisphosphatase-Mutante (Reaktion $R_{FBP}=0$)
 - Wie ist die maximale Ausbeute an Ethanol für das Substrat?
 - Überprüfen Sie die Ergebnisse für b), d), e) und f) für die beiden Substrate nun mittels linearer Optimierung („Flux Balance Analysis (FBA)“). Bestimmen Sie die Antworten b), d), e) und f) auch für Glukose (nur FBA, keine EModen).

Übungen

- 13) Berechnen Sie (constrained) Minimal Cut Sets (via EMen + via Ungleichungen), mit denen im Netzwerk „EColiCore2compressed (Small maps)“ eine Kopplung von Wachstum und Succinatsynthese (Substrat: Glukose) erzwungen werden kann. Es soll eine molare Mindestausbeute von 1.2 Succinat pro Glukose erreicht werden und eine minimale Biomasseausbeute von 0.005 gDW/(mmol Glukose) soll möglich sein.

Achtung: laden Sie für alle Berechnungen zuerst das Szenario „**glucose standard.val**“!

- Verschaffen Sie sich zunächst einen Überblick über die möglichen Biomasse- und Succinatausbeuten auf Glukose: berechnen Sie das 2-dimensionale Yield-Space-Diagramm (x-Achse: Biomasseausbeute (growth/R_GlcUp), y-Achse: Succinatausbeute (R_{SuccEx}/R_{GlcUp})).

Wie groß ist die maximale Succinatausbeute ohne Wachstum und wie groß wenn wir eine Biomasseausbeute von mind. 0.005 gDW/(mmol Glukose) fordern?
- Berechnen Sie für obige Erfordernisse (Succinatausbeute ≥ 1.2 mmol/(mmol Glukose); Biomasseausbeute ≥ 0.005 gDW/(mmol Glukose) die MCS (max. Größe 5) via Ungleichungen! Wie kann man die geforderten Ausbeute als Ungleichung spezifizieren?

Wie viele MCS bis max. Größe 5 gibt es und wie viele cuts braucht man mindestens?
- Berechnung Sie analog die MCS bis Größe 5 via EMen (Achtung bei den Targetmoden bitte selektieren nach max. Succinatausbeute ≤ 1.199 , bei gewünschte Moden ≥ 1.2). Wieviele EMen, Targetmoden, gewünschte Moden und MCS ergeben sich?