

V. Einführung in Methoden der Netzwerkrekonstruktion

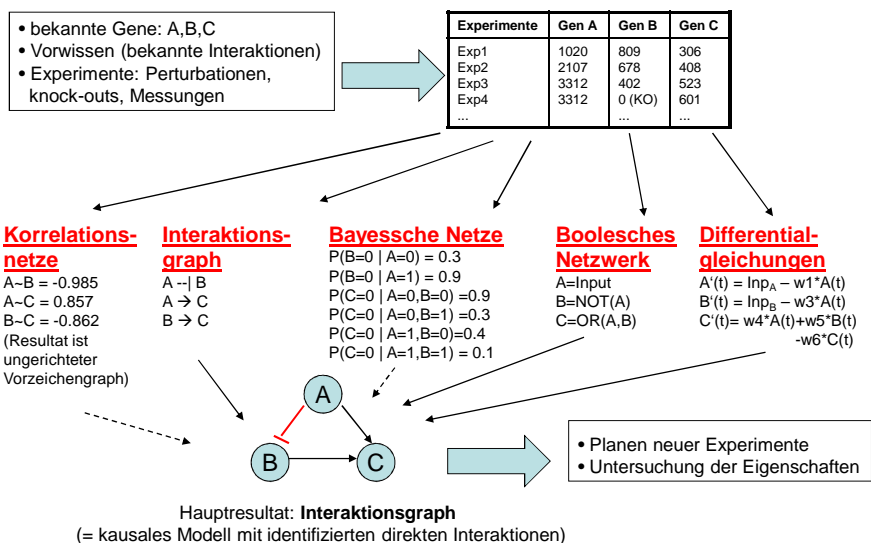
V.1 Einleitung: Netzwerkrekonstruktion

- bisher: Netzwerkstruktur gegeben → berechne Eigenschaften und Prädiktionen
- jetzt: Daten und Knoten gegeben → identifiziere die Interaktionen zwischen den Knoten durch gezielte Perturbationsexperimente
- Ursprung: genregulatorische Netze (viele Daten durch Microarrays)
- Synonyme:
 - Netzwerkrekonstruktion (network reconstruction)
 - Netzwerkidentification (network identification)
 - umgekehrtes Konstruieren (reverse engineering)
 - Netzwerkinferenz (network inference)
- Zwei Kernprobleme der Rekonstruktion:
 - (1) Verrauschte Daten
 - (2) Ungenügende Menge an Daten (eindeutige Identifikation der Struktur nicht möglich)

Es gibt **viele unterschiedliche Techniken** zur Netzwerkrekonstruktion. Welche Technik im Einzelfall angewendet, hängt von den Randbedingungen ab: (unterstrichene Varianten deuten jene an, auf die wir hier fokussieren)

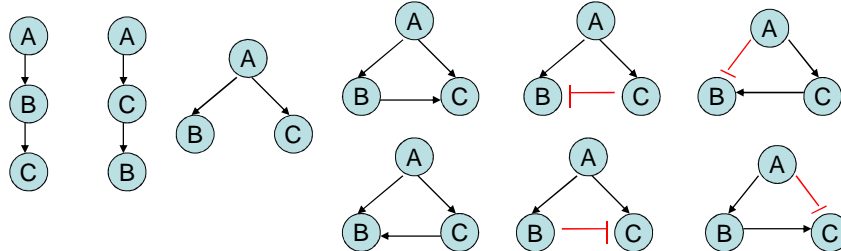
- biologisches Phänomen (Signalflüsse, metabolisches Netz, ...)
- ob transiente Effekte oder nur stationäre Zustände betrachtet werden
- zur Verfügung stehende Daten (statisch/Zeitreihen, quantitativ/qualitativ)
- ob Rauschen in Daten berücksichtigt werden soll oder nicht
- angewendete Perturbationen/Störungen in Experimenten
 - * extern (z.B. Substratkonzentration) oder/und intern (Gendeletionen)
- Berücksichtigung von vorhandenem Vorwissen
- ob paarweise (A wirkt auf C) oder kombinatorische Verknüpfungen (A UND B aktivieren C) rekonstruiert werden sollen
- was für ein Modell soll rekonstruiert werden
 - * Graph (Interaktionsgraph, Assoziationsgraph, Bayessches Netz)
 - * Boolesches Modell
 - * stöchiometrisches Netzwerk
 - * Differentialgleichungsmodell (linear/nichtlinear)

Die meisten Methoden (außer bei stöch. Netzen) rekonstruieren einen Interaktionsgraphen + evtl. weitere, vom gewählten Modelltyp abhängige Informationen (z.B. kinet. Parameter).



Ein Kernproblem der Netzwerkrekonstruktion ist die Unterscheidung von direkten (Kanten) und indirekten (Pfade) Effekten. Die Identifizierbarkeit direkter Effekte hängt von der realen Netzstruktur und den zur Verfügung stehenden Daten ab.

Bsp: Relevante Knoten: 3 (A, B, C). Experimenteller Befund: Aktivitätserhöhung in A (Perturbation) erhöht Aktivität von B und C. Welche Interaktionsgraphen können dies erklären? Hier ist eine unvollständige (!) Auswahl möglicher Lösungen.



Nur zusätzliche Experimente (z.B. Knock-out / Aktivitätssteigerung in B und C) sowie u.U. transiente Betrachtungen (z.B. Aktivität von B erst niedriger, dann höher) ermöglicht eine (teilweise) Diskriminierung der Varianten.

Netzwerkrekonstruktion ist ein hochgradig kombinatorisches Problem:

- $2 \cdot n \cdot (n-1)$ mögliche +/- Kanten in Graphen mit n Knoten
- daraus können $3^n \cdot (n-1)!$ Interaktionsgraphen erstellt werden (also im Beispiel mit 3 Knoten: 729!)
- Experimente und Daten reichen für eindeutige Diskriminierung oft nicht aus.

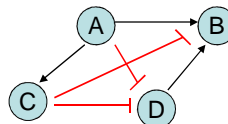
Oft wird die Komplexität des Problems durch Annahmen (z.B. nur $k \ll n$ viele Kanten pro Knoten erlaubt, Fokus auf Minimalvarianten, ...) stark reduziert.

V.2 Graphenbasierte Rekonstruktion: Transitive Reduktion

- gegeben: n Knoten die in n Perturbationsexperimenten (z.B. Gen knock-out oder Überexpression) einzeln angeregt werden; es wird gemessen, welche Knoten sich in ihrer Aktivität bzw. Konzentration verändern (+/- Effekte)
→ *Annahme*: das gesuchte Netz ist azyklisch!
- gesucht: **minimaler** gerichteter Interaktionsgraph (mit Vorzeichen), der die in den Perturbationsexperimenten gemessenen Beeinflussungen erklären kann

Beispiel: 4 Knoten → 4 einzelne Perturbationsexperimente

Perturbation	beeinflusste Größen (gemessene (signifikante) Perturbationseffekte)
A ↑	B ↑, C ↑, D ↓
B ↑	-
C ↑	B ↓, D ↓
D ↑	B ↑



Vollständiger Perturbationsgraph
 G^P (Triviale Lösung des
Rekonstruktionsproblems)

Aus den gemessenen Perturbationseffekten, die direkte und indirekte Beeinflussungen widerspiegeln, kann man bereits einen (Perturbations-)Graphen erstellen, in dem jede Beeinflussung durch eine Kante mit entsprechendem Vorzeichen dargestellt wird (siehe Graph oben). Dies wäre die triviale Lösung des Rekonstruktionsproblems. Es ist jedoch davon auszugehen dass die meisten gemessenen Effekte indirekt sind. Man versucht daher den minimalen Graphen (minimal bzgl. der Anzahl der Kanten) zu konstruieren, der die gemessenen Beeinflussungen erklären kann.

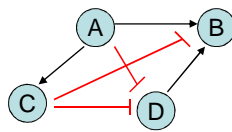
Äquivalenz bzgl. Beeinflussbarkeit: Zwei Graphen G und G' sind (beeinflussungs-) äquivalent falls gilt: *Existiert in G ein (Beeinflussungs-)Weg von einem Knoten A zu einem anderen Knoten B mit Vorzeichen s , so existiert auch in G' ein Weg von A nach B mit Vorzeichen s (und umgekehrt).*

Die **Transitive Reduktion** zu einem gegebenen Perturbationsgraphen ist der beeinflussungsäquivalente Graph G^T mit minimaler Anzahl von Kanten.

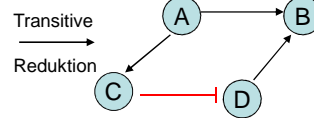
Beispiel (Fortsetzung):

Perturbation	beeinflusste Größen (gemessene (signifikante) Perturbationseffekte)
A ↑	B ↑, C ↑, D ↓
B ↑	-
C ↑	B ↓, D ↓
D ↑	B ↑

Triviale Lösung
→ Perturbationsgraph G^P



Transitive Reduktion: G^T
→ minimaler Graph, der die
Perturbationseffekte erklären kann



→ In azyklischen Graphen ist das Ergebnis der Transitiven Reduktion eindeutig (es existiert genau ein Graph G^T mit der minimalen Anzahl Kanten)

→ Algorithmus: Entferne Kanten, für die es alternativen Weg gibt ...

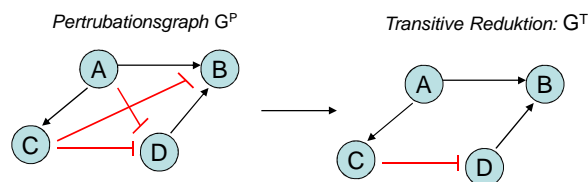
Algorithmus zur Berechnung der Transitiven Reduktion von **azyklischen** Graphen:

Teste für jede Kante $A \rightarrow B$ (mit Vorzeichen s) ob es einen alternativen Weg von A nach B gibt mit folgenden Eigenschaften:

- (i) der Weg läuft nicht über die Kante $A \rightarrow B$
- (ii) das Vorzeichen des Weges ist s .

Gibt es einen solchen Weg, dann ist $A \rightarrow B$ ein „shortcut“ und kann entfernt werden.

Bsp (Fortsetzung):

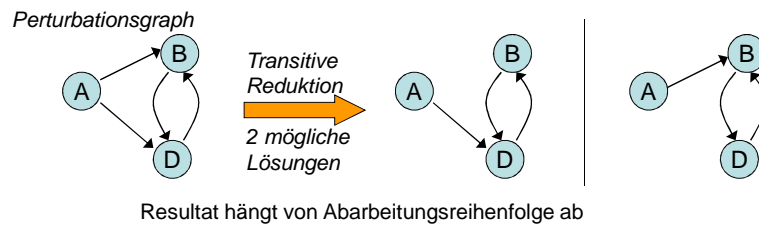


- Negative Kante von A nach D kann entfernt werden wegen negativem Weg von A über C nach D
- Negative Kante von C nach B kann entfernt werden wegen negativem Weg von C über D nach B

Algorithmus zur Berechnung der Transitiven Reduktion in zyklischen Graphen:

Im Falle von **zyklischen Graphen** ist die Transitive Reduktion nicht eindeutig (Resultat abhängig vom der Reihenfolge der Kantenentfernungen) und der Rechenaufwand deutlich komplizierter.

Bsp. für nicht eindeutige transitive Reduktion in einem zyklischen Graphen:



Hilfreich ist es dann, Kanten mit (Konfidenz-)Gewichten zu betrachten (z.B. über Korrelationsgewichte) und damit eine sinnvolle Abarbeitungsreihenfolge festzulegen.

Transitive Reduktion

Vorteile:

- einfach
- bietet Minimallösung an → die reale Lösung muss den gefundenen Graphen als Teilgraphen enthalten
- Verarbeitung von Mehrfach-Perturbationen (z.B. Doppelmutante) möglich
→ liefert besseren (vollständigeren) Perturbationsgraphen

Nachteile/Probleme

- nur azyklische Netze einfach; zyklische Netze problematischer
- an allen Knoten muss perturbiert/gemessen werden ($n \cdot (n-1)$ Messungen)
- Binarisierung von Daten notwendig (nicht trivial)
(verrauschte Daten → verrauschter Perturbationsgraph → schlechte Ergebnisse bei Transitiver Reduktion)

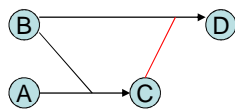
V.3 Beispiel für Rekonstruktion eines Booleschen Modells

(nach Ideker et al., 2000)

- gegeben: n Knoten und m Perturbationsexperimente, bei denen bestimmte Knoten (auch in Kombinationen) an/ausgeschaltet werden (knock-out oder knock-in) und dann die resultierenden Aktivierungen der Knoten im steady state gemessen werden
 → *Binarisierung* der Daten auf 0/1-Werte erforderlich
 → *Annahme*: das Netz (d.h. der unterliegende Interaktionsgraph) ist azyklisch!

- gesucht: **minimales Boolesches Netz** (d.h. die Booleschen Funktionen in jedem Knoten), welches die gemessenen Daten am besten wiedergeben kann

- Beispiel:



Boolesche Funktionen (diese sind im realen Fall unbekannt und gerade zu identifizieren):
 $A=1$ (Input) , $C=A \cdot B$
 $B=1$ (Input) , $D=B \cdot \neg C$

Verfügbare Perturbationsexperimente (P0...P4) und gemessene (binarisierte) Zustände im steady-state

Perturbationsexp.	A	B	C	D
P0 (unperturbiert)	1	1	1	0
P1 (knock-out A)	<u>0</u>	1	0	1
P2 (knock-out B)	1	<u>0</u>	0	0
P3 (knock-out C)	1	1	<u>0</u>	1
P4 (knock-in D)	1	1	1	<u>1</u>

Vorgehensweise: 3 Schritte (durchzuführen für jeden Knoten i)

(1) Identifiziere potenzielle Vorgängerknoten

Betrachte alle Paare (x,y) von Perturbationsexperimenten, in denen Knoten i verschiedene Aktivierungen (0/1) hat (Ausnahme: Experimente wo i selbst direkt perturbiert wurde). Für jedes dieser Paare (x,y) finde die Menge S_{xy} aller Knoten, die ebenfalls verschiedene Zustände in diesen Experimenten hat.

→ Mindestens einer dieser Knoten in S_{xy} muss die verschiedenen Zustände von i in Experimenten x und y verursacht haben und muss daher in der Booleschen Funktion von i auftauchen

Perturbationspaare + assoziierte Knotenänderungen im Beispiel:

A und B: keine induzierten Veränderungen (da Inputs)
 C: $(0,1) \rightarrow S_{0,1}=\{A,D\}$, $(0,2) \rightarrow S_{0,2}=\{B\}$, $(1,4) \rightarrow S_{1,4}=\{A\}$, $(2,4) \rightarrow S_{2,4}=\{B,D\}$
 D: $(0,1) \rightarrow S_{0,1}=\{A,C\}$, $(0,3) \rightarrow S_{0,3}=\{C\}$, $(1,2) \rightarrow S_{1,2}=\{A,B\}$, $(2,3) \rightarrow S_{2,3}=\{B\}$

(2) Identifiziere minimale Menge an Vorgängerknoten

Finde (für jeden Knoten) die kleinste Menge S_{min} , so daß $S_{xy} \cap S_{min} \neq \emptyset$ für alle xy .

→ dies ist *minimum hitting set* (d.h. minimal hitting set mit kleinster Kardinalität)

→ kann z.B. über hitting set Algorithmus bestimmt werden

→ falls Lösung S_{min} nicht eindeutig → betrachte alle Lösungen separat

Beispiel:

$C: \cup S_{xy} = \{\{A,D\}, \{B\}, \{A\}, \{B,D\}\} \rightarrow S_{min} = \{A,B\}$ (eindeutig)

$D: \cup S_{xy} = \{\{A,C\}, \{C\}, \{A,B\}, \{B\}\} \rightarrow S_{min} = \{B,C\}$ (eindeutig)

(3) Bestimme aus den Vorgängerknoten mit Hilfe der Experimente die Wahrheitstabelle (d.h. die Booleschen Funktionen)

Fehlende Einträge (Experimente) in der Wahrheitstabelle markiert man mit *

Knoten C: $S_{min} = (A,B)$
(=Vorgängerknoten)

Knoten D: $S_{min} = (B,C)$
(=Vorgängerknoten)

A	B	C	Abgeleitet aus Exp.
0	0	*	-
0	1	0	P1
1	0	0	P2
1	1	1	P0,P4

B	C	D	Abgeleitet aus Exp.
0	0	0	P2
0	1	*	-
1	0	1	P1, P3
1	1	0	P0

Die fehlenden Werte in der Tabelle können mit gezielten Experimenten ermittelt werden.

Zusammenfassung: Rekonstruktion Boolescher Netze

Vorteile: - einfach, schnell, pos/neg. Effekte werden erfasst
- liefert Vorschläge für weitere Experimente

Nachteile: - nur azyklische Netze
- viele Experimente notwendig (mind. $2^{n-1}-1$ bei n Knoten)
- Binarisierung von Daten notwendig (nicht trivial; insbes. bei verrauschten Daten)

V.4 Beispiel für Rekonstruktion eines dynamischen Modells

(hier: lineares ODE Modell, nach Yeung et al, 2002)

Ansatz: das System mit n Knoten lässt sich über n lineare Differentialgleichungen und n^2 Parametern (Gewichte) beschreiben:

$$\dot{x}_i(t) = \sum_{j=1}^n w_{ij} x_j(t) + p_i$$

$$\dot{\mathbf{x}} = \mathbf{W}\mathbf{x} + \mathbf{p}$$

x_i : Aktivitäts/Konzentrationslevel Knoten i

w_{ij} : Gewicht (wie stark und mit welchem Vorzeichen) wirkt j auf i

p_i : Perturbation in i (externer Stimulus) + basaler Input (konstitutive Aktivierung)

Der Ansatz ist auch im Falle nichtlinearer Systeme annäherungsweise berechtigt wenn man annimmt dass man kleine Auslenkungen (Perturbationen) aus einem steady state x_0 heraus vornimmt; die Gleichung entspricht dann einer Linearisierung mit \mathbf{W} als Jacobimatrix in x_0 . Sind die n basalen Inputs nicht bekannt, können diese ebenfalls im Ansatz mitermittelt werden.

→ Bestimme über verschiedene Perturbationsexperimente mit Messungen im transienten Verlauf die n^2 vielen Parameter mittels folgendem Ansatz:

$$\dot{x}_i = \sum_{j=1}^n w_{ij} x_j + p_i \Leftrightarrow \dot{x}_i - p_i = \sum_{j=1}^n w_{ij} x_j$$

→ p_i , x_j und x_i' (angenähert durch $x_i'(t_1) = (x_i(t_1) - x_i(t_0))/(t_1 - t_0)$) sind gemessen; die w_{ij} sind zu bestimmen

→ mit m Perturbationsexperimenten (wo jeweils alle p_i , x_j und x_i' gemessen werden) folgt lineares Gleichungssystem mit $m \times n$ Gleichungen und $n \times n$ Unbekannten:

Für jede Perturbation k erhält man für jede Spezies i folgende Gleichung:

$$(x_1^k, x_2^k, \dots, x_n^k) \begin{pmatrix} w_{i1} \\ w_{i2} \\ \vdots \\ w_{in} \end{pmatrix} = x_i^k - p_i^k$$

Das vollständige Gleichungssystem kann geschrieben werden als:

$$\begin{pmatrix} x_1^1 & x_2^1 & \dots & x_n^1 \\ x_1^2 & x_2^2 & \dots & x_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ x_1^m & x_2^m & \dots & x_n^m \end{pmatrix} \cdot \begin{pmatrix} w_{11} & w_{21} & \dots & w_{n1} \\ w_{12} & w_{22} & \dots & w_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ w_{1n} & w_{2n} & \dots & w_{nn} \end{pmatrix} = \begin{pmatrix} x_1^1 & x_2^1 & \dots & x_n^1 \\ x_1^2 & x_2^2 & \dots & x_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ x_1^m & x_2^m & \dots & x_n^m \end{pmatrix} - \begin{pmatrix} p_1^1 & p_2^1 & \dots & p_n^1 \\ p_1^2 & p_2^2 & \dots & p_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ p_1^m & p_2^m & \dots & p_n^m \end{pmatrix}$$

Oder kurz: $(\mathbf{X}^T)_{m \times n} \cdot (\mathbf{W}^T)_{n \times n} = (\mathbf{X}^T)_{m \times n} - (\mathbf{P}^T)_{m \times n}$

- Die w_{ij} sind eindeutig bestimmt wenn mindestens n Perturbationsexperimente vorliegen und die resultierende Messmatrix \mathbf{X} vollen Rang n hat.
- aufgrund fehlender Daten/Experimente oft $m < n$
→ schränke Suchraum ein, z.B. erlaube nur $r < n^2$ viele $w_{ij} \neq 0$
- oft misst man nur im steady state, die Gleichung vereinfacht sich dann zu:

$$(\mathbf{X}_{steady_state}^T)_{m \times n} \cdot (\mathbf{W}^T)_{n \times n} = -(\mathbf{P}^T)_{m \times n}$$

Beispiel: Netzwerk mit 3 Knoten (A, B, C). Es wurden 3 Perturbationsexperimente durchgeführt. Folgende steady-state Konzentrationen wurden gemessen.

Experiment	p_A	p_B	p_C	$X_{S,A}$	$X_{S,B}$	$X_{S,C}$
1 (Wildtyp)	1	1	1	2	1	1
2 (Perturbation in B)	1	<u>4</u>	1	3.5	2.5	2.5
3 (Knock-out in A)	0	1	1	<u>0</u>	0.33	1.66
4 (Knock-out in C)	1	1	0	0.5	0.5	<u>0</u>

Es resultieren folgende Gleichungssysteme (sortiert nach Knoten, in denen die Kanten zeigen). Beachte, dass bei knock-out Experimenten die Gleichung für die entfernte Spezies NICHT verwendet werden kann:

Kanten/Gewichte in Knoten A

$$\begin{matrix} \text{Exp1} \\ \text{Exp2} \\ \text{Exp4} \end{matrix} \begin{pmatrix} 2 & 1 & 1 \\ 3.5 & 2.5 & 2.5 \\ 0.5 & 0.5 & 0 \end{pmatrix} \begin{pmatrix} w_{AA} \\ w_{AB} \\ w_{AC} \end{pmatrix} = \begin{pmatrix} -1 \\ -1 \\ -1 \end{pmatrix}$$

(Inverse)

$$(w_{AA} \ w_{AB} \ w_{AC})^T \approx (-1 \ -1 \ 2)^T$$

Kanten/Gewichte in Knoten B

$$\begin{matrix} \text{Exp1} \\ \text{Exp2} \\ \text{Exp3} \\ \text{Exp4} \end{matrix} \begin{pmatrix} 2 & 1 & 1 \\ 3.5 & 2.5 & 2.5 \\ 0 & 0.33 & 1.66 \\ 0.5 & 0.5 & 0 \end{pmatrix} \begin{pmatrix} w_{BA} \\ w_{BB} \\ w_{BC} \end{pmatrix} = \begin{pmatrix} -1 \\ -4 \\ -1 \\ 0 \end{pmatrix}$$

(Kleinste-Quadrate-Schätzung mit Pseudo-Inversen)

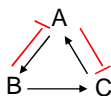
$$(w_{AA} \ w_{AB} \ w_{AC})^T \approx (1 \ -3 \ 0)^T$$

Kanten/Gewichte in Knoten C

$$\begin{matrix} \text{Exp1} \\ \text{Exp2} \\ \text{Exp3} \end{matrix} \begin{pmatrix} 2 & 1 & 1 \\ 3.5 & 2.5 & 2.5 \\ 0 & 0.33 & 1.66 \end{pmatrix} \begin{pmatrix} w_{CA} \\ w_{CB} \\ w_{CC} \end{pmatrix} = \begin{pmatrix} -1 \\ -1 \\ -1 \end{pmatrix}$$

(Inverse)

$$(w_{CA} \ w_{CB} \ w_{CC})^T \approx (-1 \ 2 \ -1)^T$$



$$\dot{\mathbf{x}}(t) = \begin{pmatrix} -1 & -1 & 2 \\ 1 & -3 & 0 \\ -1 & 2 & -1 \end{pmatrix} \mathbf{x}(t) + \mathbf{p}$$

(Genau dieses System wurde zur Erzeugung der synthetischen Daten (in der Tabelle) verwendet)

Rekonstruktion mit linearem ODE-System:

Vorteile:

- dynamisches Modell wird rekonstruiert (Zeitreihen können einfließen); pos./neg. Effekte in der Rekonstruktion enthalten (mit Gewichtung)
- Zyklen erlaubt
- einfacher Algorithmus
- erweiterbar mit weiteren Nebenbedingungen (z.B. max. Anzahl Nichtnullgewichte)

Nachteile/Probleme

- viele Daten erforderlich
- stark nichtlineare System nur bedingt behandelbar

V.5 Zusammenfassung: Netzwerkrekonstruktion

- wichtiges Problem der Systembiologie; viele Verfahren
- eingesetzt insbes. für (gen)regulatorische Netzwerke
- oft limitiert aufgrund von Datenmenge + Datenqualität (Rauschen)
- Performance verschiedener Verfahren wird oft mittels synthetischer Daten und Netzwerke getestet
 - der Vergleich der Methoden ist nicht-trivial (z.B. was ist als schwerwiegender zu betrachten: falsch positive Kanten oder falsch negative Kanten?)

Referenzen

- * Hecker M, Lambeck S, Toepfer S, van Someren E, Guthke R. 2009. Gene regulatory network inference: Data integration in dynamic models – A review. *Biosystems*, 96:86-103.
- van Someren EP, Wessels LFA, Backer E and Reinders MJT. 2002. Genetic network modeling. *Pharmacogenomics*, 3:507-525.
- Gardner TS and Faith JJ. Reverse-engineering transcription control networks. 2005. *Physics of Life Reviews*, 2:65-88.
- Wagner A. How to reconstruct a large genetic network from n perturbations in fewer than n^2 easy steps. 2001. *Bioinformatics*, 17: 1183-1197.
- Ideker T., Thorsson V and Karp R. Discovery of Regulatory Interactions Through Perturbation: Inference and Experimental Design. 2000. *Proceedings of the Pacific Symposium on Biocomputing (2000)*, 5:302-313.
- Yeung MKS, Tegner J and Collins JJ. 2002. Reverse engineering gene networks using singular value decomposition and robust regression. *Proc. Natl. Acad. Sci.*, 99: 6163-6168.
- Klamt S, Flassig RJ and Sundmacher K. 2010. TRANSWESD: inferring cellular networks with transitive reduction. *Bioinformatics*, 26:2160-2168

* besonders = empfohlen

Übungen

- 1) In einem Netzwerk mit 4 Knoten A, B, C, D (Annahme: keine Zyklen) ergaben sich bei Perturbationsexperimenten folgende steady-state Messungen: (Hinweis: synthetische (also idealisierte) Daten aus einem ODE Modell!)

Experiment	p_A	p_B	p_C	p_D	$X_{S,A}$	$X_{S,B}$	$X_{S,C}$	$X_{S,D}$
1 (Wildtyp)	1	1	0	0	0.2	0.2	0.222	0.897
2 (knock-out in A)	0	1	0	0	<u>0</u>	0.2	0	0.667
3 (Knock-out in B)	1	0	0	0	0.2	<u>0</u>	0.667	0.870
4 (Überexpression in C)	1	1	<u>0.5</u>	0	0.2	0.2	<u>0.722</u>	0.96
5 (Überexpression in D)	1	1	0	<u>0.1</u>	0.2	0.2	0.222	<u>0.997</u>
6 (Knock-out in A und B)	0	0	0	0	<u>0</u>	<u>0</u>	0	0
7 (Knock-out in B und C)	1	0	0	0	0.2	<u>0</u>	<u>0</u>	0

- Rekonstruieren Sie den Interaktionsgraphen mit der Methode der Transitiven Reduktion. Hinweis: Vorher müssen Sie über einen Schwellenwert festlegen, ab wann eine Spezies ihren Wert signifikant verändert hat. Wir nehmen an, dass 5% Änderung signifikant ist.
- Rekonstruieren Sie das Netzwerk als (minimales) Boolesches Modell. Hinweis: Sie müssen die Daten vorher geeignet binarisieren. Nehmen Sie 50% des jeweils maximalen Aktivierungslevels für jede Spezies als Schranke.
- Rekonstruieren Sie das Netzwerk als lineares ODE Modell. Nach Rekonstruktion der W-Matrix sollten Sie kleine Einträge entfernen, um so nur signifikante Interaktionen darzustellen.

Übungen

2) Schreiben Sie je eine MATLAB-Funktion zur Netzwerkrekonstruktion mit

- a) Transitiver Reduktion
- b) Booleschen Netzen
- c) linearen ODEs