Berechnung optimaler Bayesscher Netzwerke

Ivo Große
IPK Gatersleben
Bioinformatikzentrum Gatersleben-Halle

Markov Modelle

- **Markov 0**
  - Nukleotide statistisch unabhängig
  - position weight matrix (Staden 1984)

- **Markov 1**
  - statistische Abhängigkeit zwischen nächsten Nachbarn
  - weight array matrix (Zhang et al. 1993)

- **Markov 2**
  - statistische Abhängigkeit zwischen nächsten und übernächsten Nachbarn

- **Markov d**
  - statistische Abhängigkeit zwischen d nächsten Nachbarn

Markov Modelle → Problem

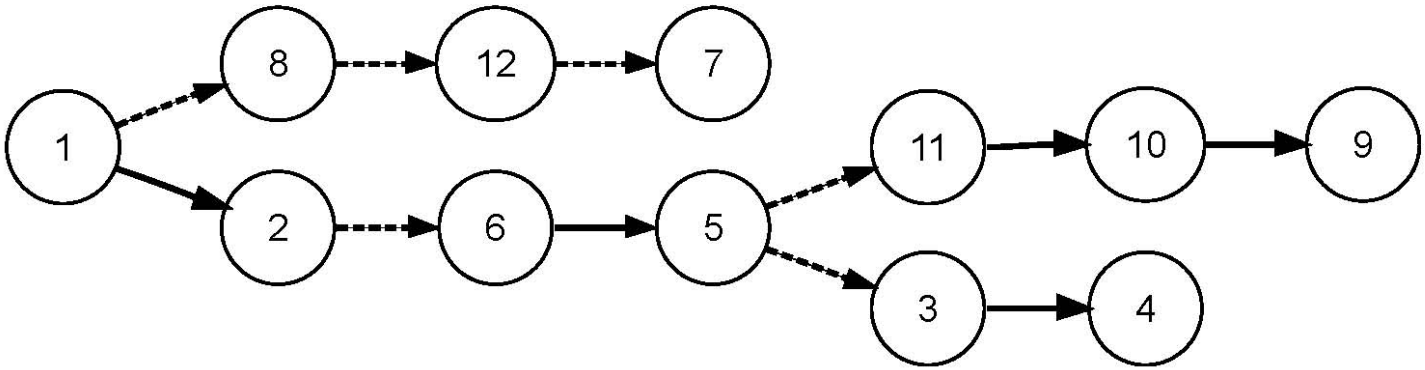
Problem: statistische Abhängigkeiten nur zwischen nächsten Nachbarn

Idee: Bayes Netze

- Theorie (Pearl 1988, Buntine 1991, Heckerman 1995)

Definition:

- Gerichteter azyklischer Graph (DAG)
- An jedem Knoten: Markov Modell



Bayes Netze → Struktur-Lernen

d=1: einfach	→	maximaler Spannbaum (MST) (Kruskal, Prim, …)
d>1: NP-schwer	→	Heuristiken (Friedman et al. 1999, Brejova et al. 2003)

MEME

Ziel: Suche nach unbekannten Bindestellen in upstream Regionen ko-exprimierter Gene

gegeben Motif, gesucht Positionen, Lösung einfach

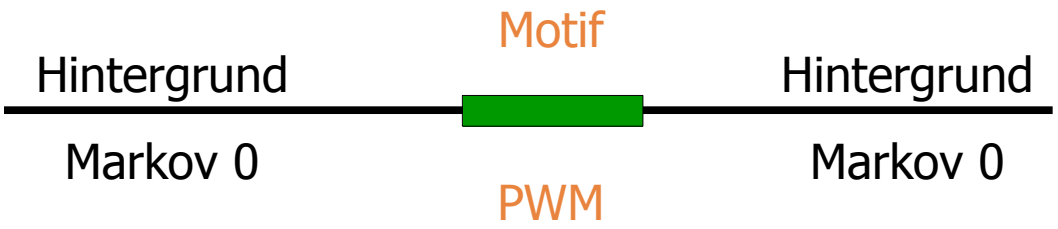
gegeben Positionen, gesucht Motif, Lösung einfach

gegeben	weder Motif noch Positionen
gesucht	Motif und Positionen
Lösung	???

Expectation Maximization (EM) → **MEME**
(Bailey & Elkan 1995)

MEME → Problem

MEME basiert auf Markov 0



Ziel: MEME mit Markov, Bayes Baum, Bayes Netz, … → generische Implementierung!

Beispiel:

→ Markov Modelle (für Hintergrund: Thijs et al. 2001, Liu et al. 2001)

→ Sigma-70 (Hershberg & Margalit 2005)

Prozentsatz der korrekt gefundenen Bindungsstellen

	hMM(0)	hMM(1)	hMM(2)	hMM(3)	hMM(4)
iMM(0)	0.3	0.0	**53.9**	**55.0**	51.2
iMM(1)	0.6	0.0	46.5	45.3	46.2

MEME → Markov Modelle, Bayes Bäume, Bayes Netze

Strukturlernen in Mischmodellen → Probleme!!!

Markov	→ maximum likelihood	→ EM	
Bayes Netz	→ maximum likelihood	→ EM	→ d=1 einfach (MST) → d>1 NP-schwer

Bayes Netze → Strukturlernen

Es gibt $O(n! \, 2^{n(n-1)/2})$ DAGs.

Für jeden DAG können wir die Likelihood berechnen.

2 Optimierungsprobleme:

- suche den DAG, der die Likelihood maximiert.
- suche den DAG mit maximal k Eltern pro Knoten, der die Likelihood maximiert.

Beide Probleme sind NP-schwer (Chickering et al. 1994)

Heuristiken:

sparse candidate Algorithmus (Friedman et al. 1999)
greedy (Brejova et al. 2003)

Idee: ???

→ Analog zum Handelsreisendenproblem (TSP)

→ Dynamisches Programmieren

Bayes Netze → Strukturlernen → DP

Literatur:

60er → ???

70er, 80er, 90er → ???

2004:

- Koivisto & Sood. Exact Bayesian Structure Discovery in Bayesian Networks
- Ott et al. Finding Optimal Models for Small Gene Networks

2005:

- Singh & Moore. Finding Optimal Bayesian Networks by Dynamic Programming

Bayes Netze → Strukturlernen → DP

Naive Implementierung:

Laufzeit: $O(n\,2^n)$

Speicher: $O(n\,2^n)$

Mehr als 40 Knoten unmöglich (Ott et al. 2004; Hansen et al. 2004)

20 Gene:

Laufzeit: 50 Stunden, Sun Fire 15K mit 96 CPUs (Ott et al. 2004)

38 Gene:

Speicher: 600 GB (Hansen et al. 2004)

Problem: exponentielles Wachstum der Laufzeit und des Speicherbedarfs!!!

Geht es irgendwie schneller / weniger naiv?

Bayes Netze → Strukturlernen → DP

Gegeben:

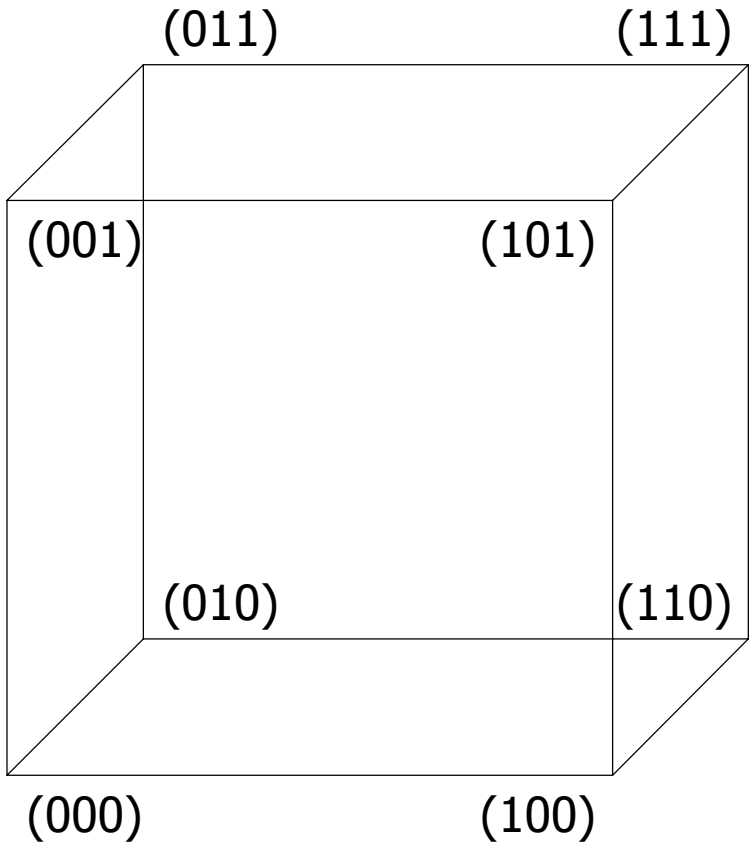
→ Hyperwürfel mit Kantengewichten

Gesucht:

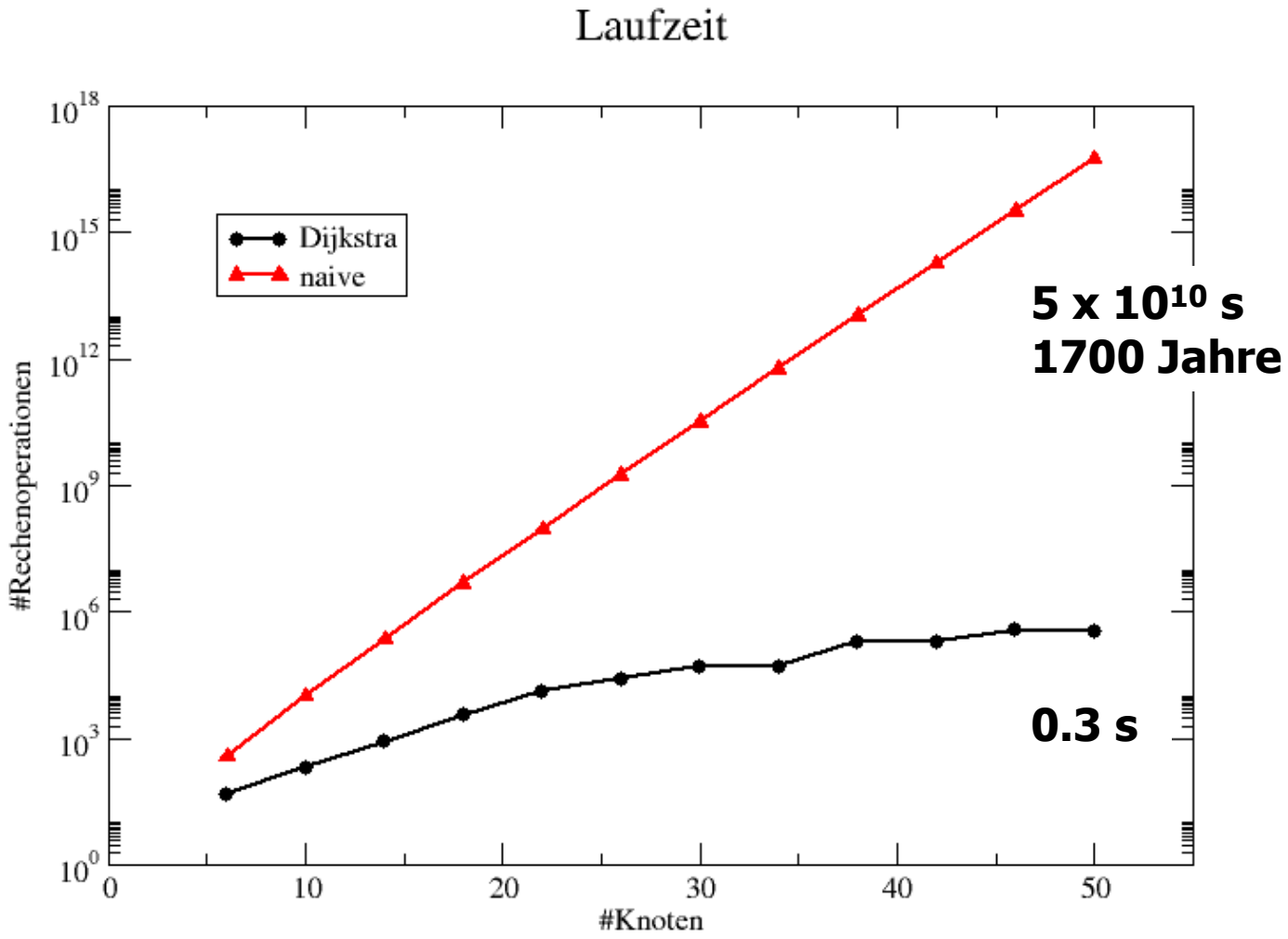
→ kürzester Pfad von (00…0) nach (11…1)

Idee: ???

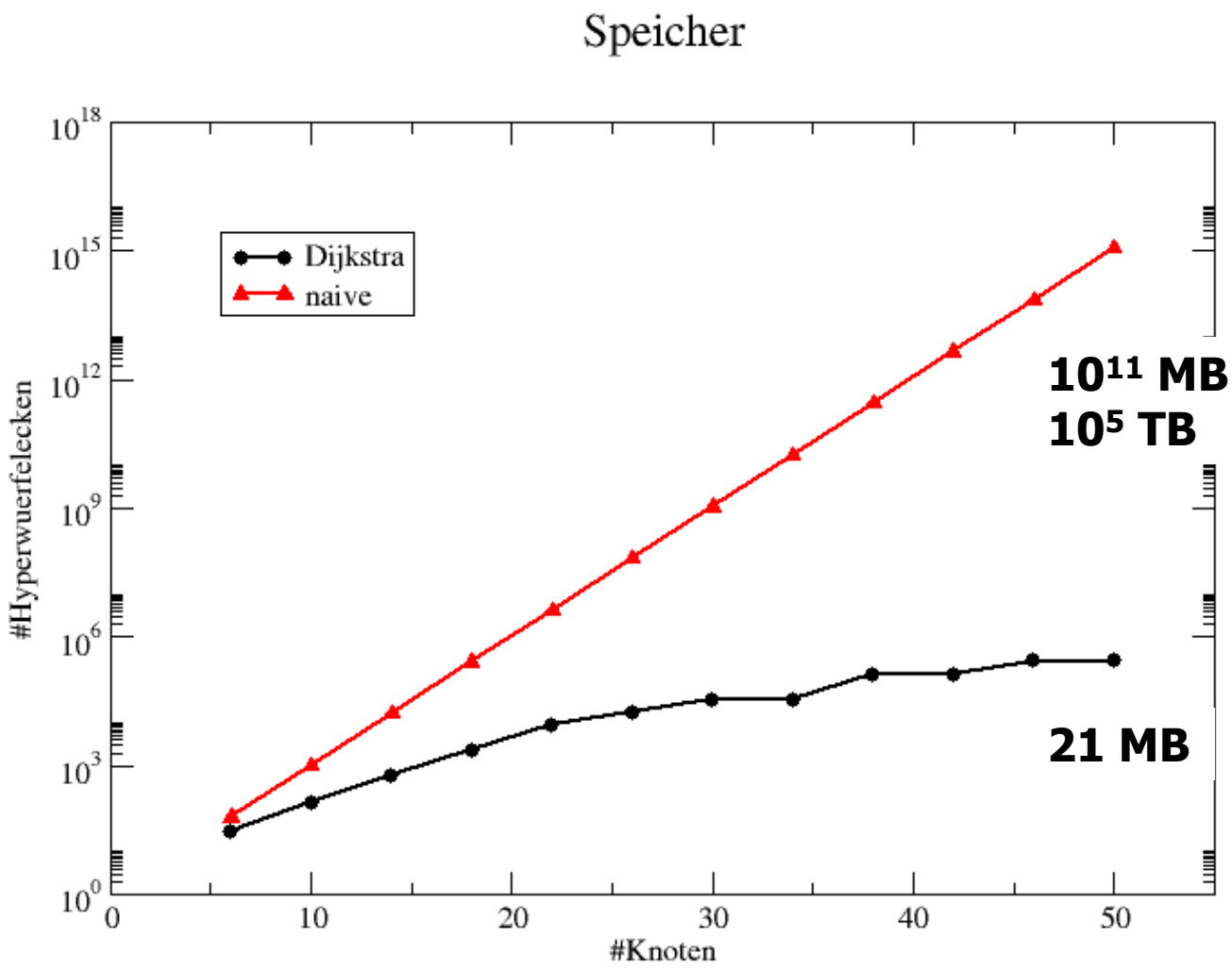
→ Dijkstra!



Dijkstra-DP


Laufzeit
#Rechenoperationen
#Knoten
Dijkstra
naive
10^{18}
10^{15}
10^{12}
10^{9}
10^{6}
10^{3}
10^{0}
0
10
20
30
40
50
5 x 10^{10} s
1700 Jahre
0.3 s

Dijkstra-DP


Speicher
#Hyperwuerfelecken
#Knoten
Dijkstra
naive
10^{18}
10^{15}
10^{12}
10^{9}
10^{6}
10^{3}
10^{0}
0
10
20
30
40
50
10^{11} MB
10^{5} TB
21 MB

Dijkstra-DP

Typischerweise: Netze mit ca. 50 Knoten kein Problem

Optimum klar ←→ Algorithmus schnell

Optimum heiß umkämpft ←→ Algorithmus langsam

Algorithmus:
Finde optimalen DAG mit maximal k und minimal m Eltern pro Knoten

Fallbeispiel → Expressionsdaten

Adenokarzinom:

→ 41 Patienten (MLU)

Apoptose Stoffwechselweg aus KEGG:

→ 14 Gene

BN(sparse candidate alg.) → SCA

BN(dynamic programming) → DP

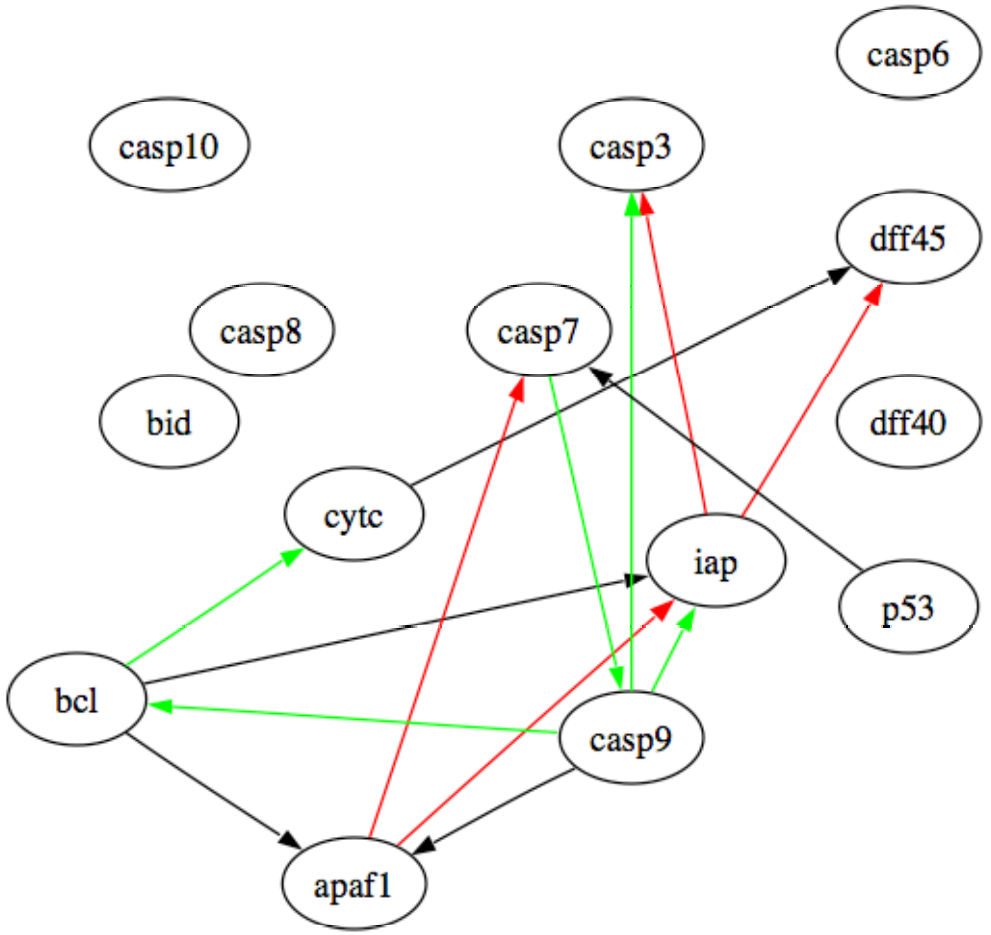
Score(SCA) = -300.9 = 99.4%

Score(BN) = -300.2

schwarz = SCA & DP

rot = nur SCA

grün = nur DP



Fallbeispiel → Metabolitdaten

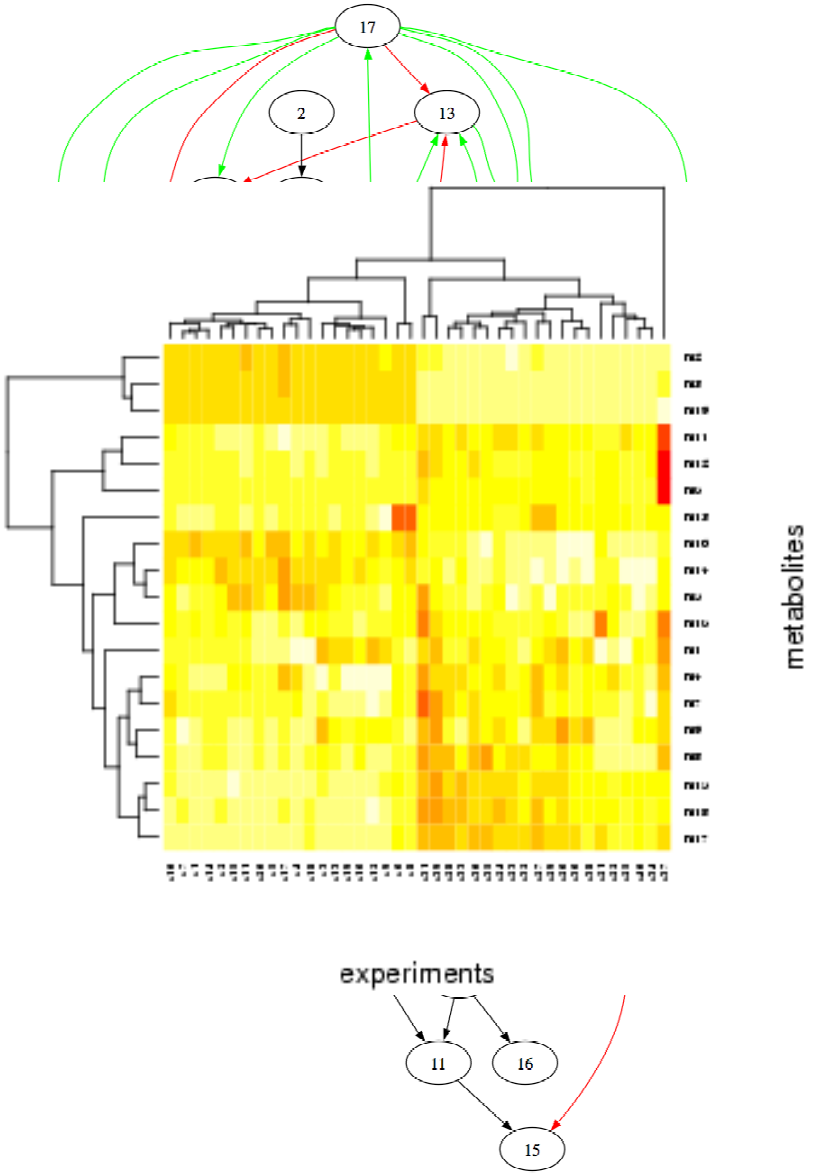
Arabidopsis Samen (IPB)
→ 20 x Wildtyp
→ 20 x KO Mutante

LC Massenspektrometrie (IPB)
→ 19 Metabolite

BN(sparse candidate alg.) → SCA
BN(dynamic programming) → DP

Score(SCA) = 237.7 = 96.8%
Score(BN) = -232.3

schwarz = SCA & DP
rot = nur SCA
grün = nur DP



Zusammenfassung

MEME + Markov / BT / BN → exaktes Strukturlernen

BNs → NP-schwer → DP

Naive-DP → Dijkstra-DP

Potentielle Anwendungen: heuristisch → exakt

Danksagung

Experimentatoren @ IPK, IPB, MLU, …

Studenten:
- Berlin: Valentin Ziegler
- Halle: Andre Gohr, Michaela Mohr, Yvonne Pöschl

Jerusalem: Yoseph Barash, Nir Friedman, Tommy Kaplan, Hanah Margalit
Berlin: Stefan Hougardy
Halle: Holger Blaar, Gerold Jäger, Steffen Neumann, Stefan Posch, Jop Sibeyn
Magdeburg: Stephan Mertens

BMBF