Incorporating graph priors in Bayesian networks

Sushmita Roy
sroy@biostat.wisc.edu

Computational Network Biology
Biostatistics & Medical Informatics 826
https://compnetbiocourse.discovery.wisc.edu

Oct 2nd, Oct 4th 2018

Plan for this section

- Overview of network inference (Sep 18th)
- Directed probabilistic graphical models Bayesian networks (Sep 18th, Sep 20th)
- Gaussian graphical models (Sep 25th)
- Dependency networks (Sep 27th)
- Integrating prior information for network inference (Oct 2nd, 4th)

Integrating priors into graph structure learning

- We will look at two approaches to integrate other types of data to better learn regulatory networks
- Bayesian network structure prior distributions (Oct 2nd)
- Dependency network parameter prior distributions (Oct 4th)

Plan for today

- Overview of integrative network inference
- Defining priors on graph structure
- Learning Bayesian networks with priors using Markov Chain Monte Carlo
- Applications of Bayesian networks with priors
  - Inferring the yeast cell cycle network
  - Inferring cancer signaling

Why prior-based structure learning?

- Priors enable us to provide additional information to constrain the structure of the graph
- Learning genome-scale networks is computationally challenging
  - The space of possible graphs is huge
  - There is not sufficient amount of training examples to learn these networks reliably
  - Multiple equivalent models can be learned
- One type of data (expression) might not inform us of all the regulatory edges

Types of integrative inference frameworks

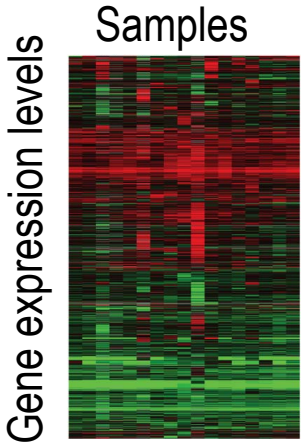
- Supervised learning
  - Require examples of interaction and non-interactions
  - Train a classifier based on edge-specific features
- Unsupervised learning
  - Edge aggregation
  - Model-based learning
    - Auxiliary datasets serve to provide priors on the graph structure

Unsupervised network inference

- Do not assume the presence of a gold standard set of edges
- Have been applied primarily for regulatory networks with a few exceptions
- Some approaches for integrative inference in regulatory networks
  - Inferelator (Greenfield et al., Bioinformatics 2013)
  - Lirnet (Lee et al., Plos computational biology 2009)
  - Physical Module Networks (Novershtern et al., Bioinformatics 2011)
  - iRafNet (Petralia et al., 2015)
  - MERLIN+P (Fotuhi-Siahpirani & Roy, 2016)

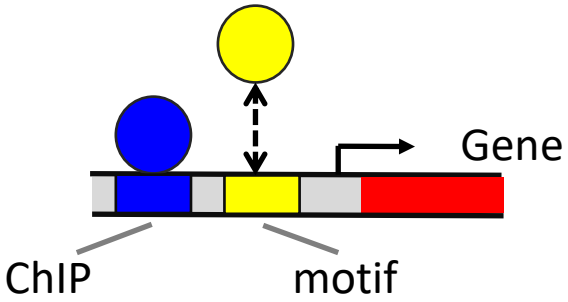
Types of data for reconstructing transcriptional networks

- Expression data
  - Genome-wide mRNA levels from multiple microarray or RNA-seq experiments
  - Gene expression can come from time courses as well as single time point



- Complementary datasets
  - ChIP-chip and ChIP-seq
  - Sequence specific motifs

These can be used as priors



Classes of methods for incorporating priors

- Parameter prior based approaches
  - Inferelator (Greenfield et al., Bioinformatics 2013)
  - Lirnet (Lee et al., Plos computational biology 2009)
- Structure prior based approaches
  - Dynamic Bayesian network (Hill et al., Bioinformatics, 2012, Werhli et al., 2007)
  - Physical module networks (Novershtern et al., Bioinformatics 2011)
  - MERLIN-P (Siahpirani et al.,2016)

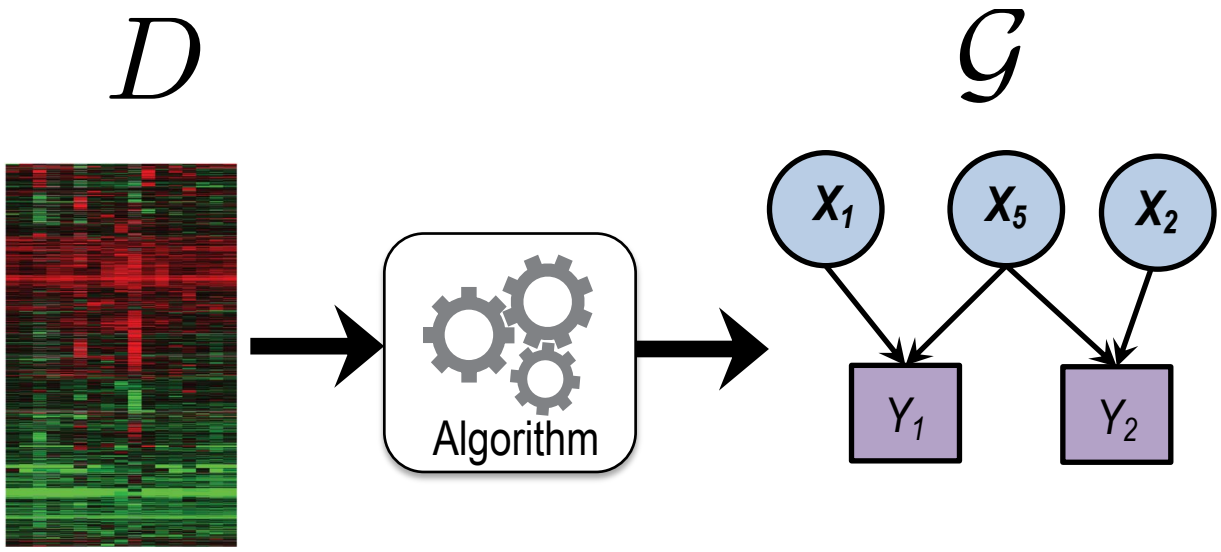
Prior-based approaches for network inference

- Given
  - Gene expression data and
  - Complementary data that supports the presences of an edge
    - Presence of a sequence motif on a gene promoter
    - ChIP-chip/seq binding of factor X on gene Y's promoter
- Do
  - Predict which regulators drive the expression of a target gene, while incorporating complementary evidences as much possible
- How?
  - Place a prior on the graph where the prior is obtained from complementary data

Plan for today

- Overview of integrative network inference
- Defining priors on graph structure
- Learning Bayesian networks with priors using Markov Chain Monte Carlo
- Applications of Bayesian networks with priors
  - Inferring the yeast cell cycle network
  - Inferring cancer signaling

Bayesian formulation of network inference



$$P(\mathcal{G}|D) \propto P(D|\mathcal{G})P(\mathcal{G})$$

Posterior distribution

Data likelihood

Model prior

Optimize posterior distribution of graph given data

A few computational concepts

- Energy of a graph and the Gibbs distribution
- Dynamic Bayesian networks
- Markov Chain Monte Carlo
- Hyper parameters

Energy function of a network G

- A function that measures agreement between a given graph G and prior knowledge
- Allows one to incorporate both positive and negative prior knowledge

Energy function on a graph

- A graph G is represented by a binary adjacency matrix
  - $G_{ij} = 0$ if there is no edge from node i to node j
  - $G_{ij} = 1$ if there is an edge from i to j
  - $G_{ji} = 1$ if there is an edge from j to i
- Encode a "prior" network as follows:
  - $B_{ij} = 0.5$ if we don't have any prior
  - $0 < B_{ij} < 0.5$ if we know that there is no edge
  - $B_{ij} > 0.5$ if we know there is an edge
- Energy of G is

$$E(G) = \sum_{ij=1} |B_{ij} - G_{ij}|$$

Energy function of a graph

- Energy E of a network G is defined as

$$E(G) = \sum_{ij=1} |B_{ij} - G_{ij}|$$

- This is 0 when there is perfect agreement between prior knowledge B and G
- Higher the energy of G the greater the mismatch

Using the energy to define a prior distribution of a graph

- A prior distribution for a graph G can be defined using $E(G)$

$$P(G|\beta) = \frac{1}{Z(\beta)}\exp(-\beta E(G))$$

- This is also called a Gibbs distribution
- β is the hyperparameter: parameter of the prior distribution
- Z is the partition function

$$Z(\beta) = \sum_{G} \exp(-\beta E(G))$$

- In general the partition function is hard to compute

Incorporating multiple sources of prior networks

- Suppose we have two sources of prior networks
- We can represent them as two prior networks B^1 and B^2
- And define the energy of G with respect to both of these

$$E_1(G) = \sum_{i,j=1} |B^1_{i,j} - G_{ij}|$$

$$E_2(G) = \sum_{i,j=1} |B^2_{i,j} - G_{ij}|$$

Prior distribution incorporating multiple prior networks

- The prior takes the form of another Gibbs distribution

$$P(G|\beta_1, \beta_2) = \frac{1}{Z}\exp(-(\beta_1 E_1(G) + \beta_2 E_2(G)))$$

- This can be extended to more prior networks in the same way
- The partition functions are in general hard to compute
- However, for a particular class of BNs, these partition functions can be computed easily

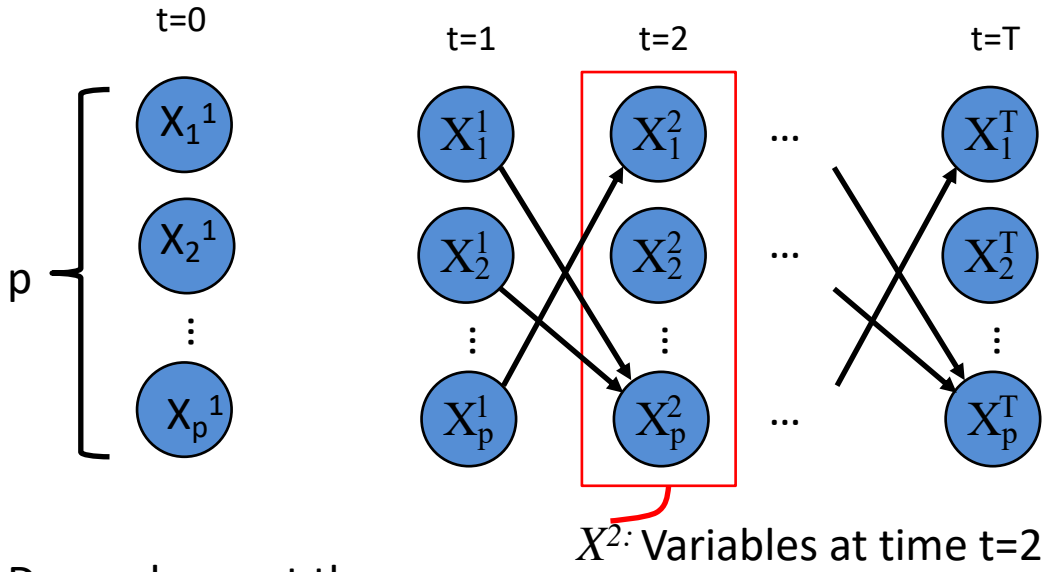
Dynamic Bayesian networks

- Bayesian networks that we have seen so far do not allow for cyclic dependencies
- If we have time series data, we can overcome this limitation using a Dynamic Bayesian network

Dynamic Bayesian Nets (DBNs)

- A DBN is a Bayesian Network for dynamic processes
- Suppose we have a time course with T time points
- Let $X^t = \{X_1^t \cdots, X_p^T\}$ denote the set of p random variables at time t
- Let $\mathbf{X} = \{X^1 \cdots, X^T\}$
- A DBN over these variables defines the joint distribution of $\mathrm{P}(\mathbf{X})$, where
- A DBN, like a BN, has a directed acyclic graph G and parameters Θ
- G typically specifies the dependencies between time points
  - In addition we need to specify dependence (if any) at $t=0$

A DBN for *p* variables and *T* time points



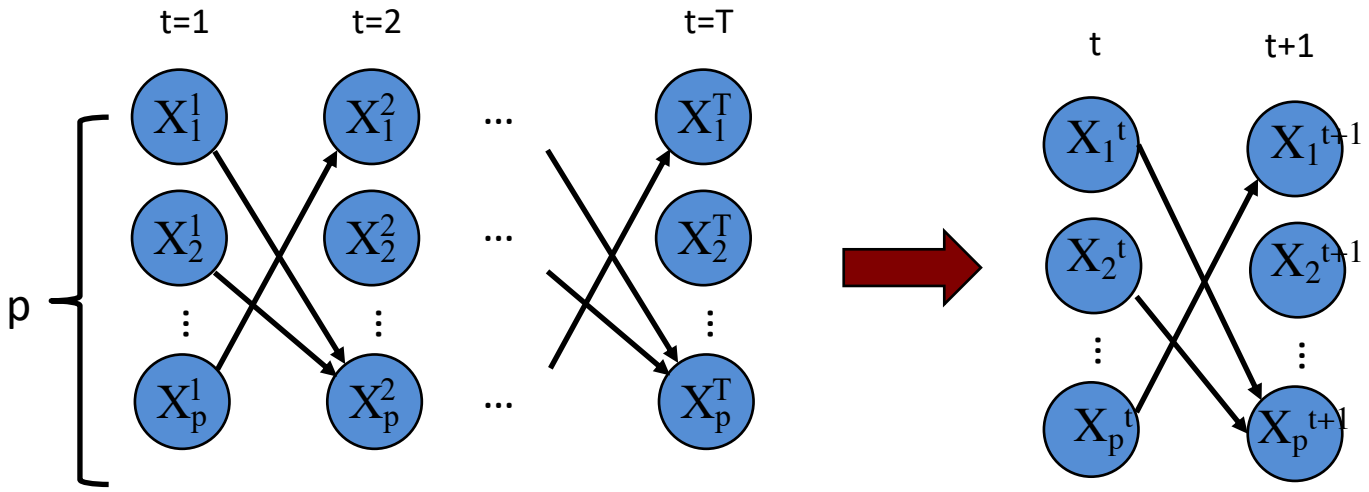
Dependency at the first time point

Stationary assumption in a Bayesian network

The stationarity assumption states that the dependency structure and parameters do not change with t

$$P(X^{t+1}|X^t) = P(X^t|X^{t-1})$$

Due to this assumption, we only need to specify dependencies between two sets of variables (and possibly for the first time point)



Dynamic Bayesian networks

Joint Probability Distribution can be factored into a product of conditional distributions :

$$P(\mathbf{X}|G,\Theta) = P(\mathbf{X}^1)\prod_{i=1}^{p}\prod_{t=2}^{T} P(X_i^t|X_{\pi_G(i)}^{t-1},\theta_i)$$

Graph encoding dependency structure

Parents of X_i^t defined by the graph

The partition function for a prior over DBN

- In the DBN, if
  - we allow parents only from the previous time point
  - we allow each node to have at most m parents
- The prior distribution decomposes over individual nodes and their possible parent sets

$$E(G) = \sum_{i=1}^{N} \mathcal{E}(n, \pi_G(n))$$

$$\mathcal{E}(n, \pi_G(i)) = \sum_{j \in \pi_G(n)} (1 - B_{jn}) + \sum_{j \notin \pi_G(n)} B_{jn}$$

The partition function for a DBN prior

- The partition function is computed easily by summing over all variables and their potential parent sets

$$Z(\beta) = \sum_{G} \exp(-\beta E(G))$$

$$= \sum_{\pi_G(1)} \cdots \sum_{\pi_G(N)} \exp(-\beta(\mathcal{E}(N, \pi_G(1)) + \cdots + \mathcal{E}(1, \pi_G(N))))$$

Each summation represents a sum over possible configurations for the parent set.
If we restrict the number of parents to *m,* this is polynomial in N

Plan for today

- Overview of integrative network inference
- Defining priors on graph structure
- Learning Bayesian networks with priors using Markov Chain Monte Carlo
- Applications of Bayesian networks with priors
  - Inferring the yeast cell cycle network
  - Inferring cancer signaling

Markov Chain Monte Carlo (MCMC) sampling

- We have looked at a greedy hill climbing algorithm to learn the structure of the graph
- MCMC provides an alternative (non-greedy) way of finding the graph structure
- The idea is to estimate the distribution, $P(G|D)$, and draw "samples" of G from this distribution
- MCMC is a general strategy to sample from a complex distribution
  - If we can sample from the distribution, we can also estimate specific properties of the distribution

MCMC for learning a graph structure

- Recall the Bayesian framework to learning Bayesian networks
- We wish to estimate $P(G|D)$ and draw multiple G's from this distribution
  - But this distribution is difficult to estimate directly
- We will devise a Markov Chain such that its stationary distribution will be equal to $P(G|D)$
- We will then use the Markov Chain to also draw potential G's

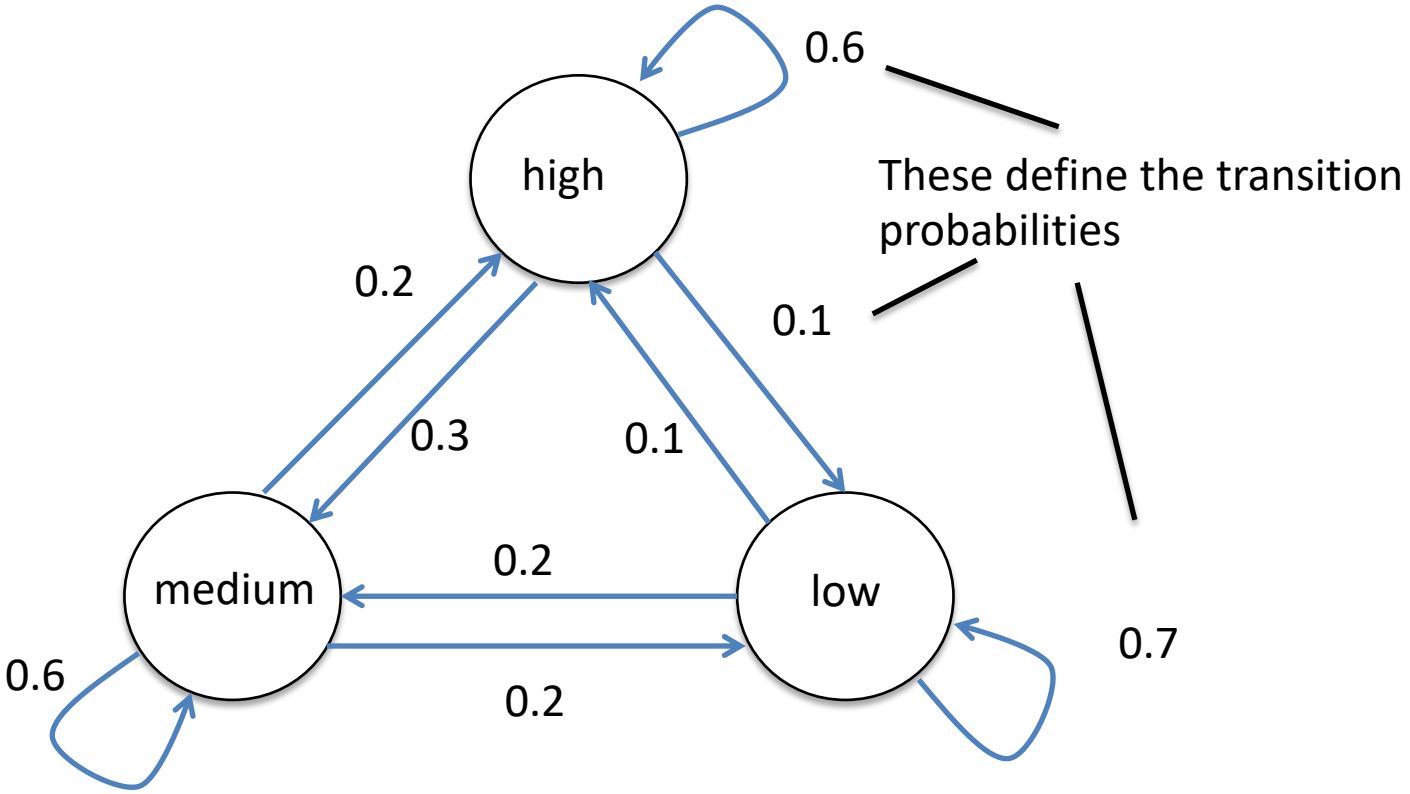
Markov chain

- A Markov chain is a probabilistic model for sequential observations where there is a dependency between the current and the previous state
- It is defined by a graph of possible states and a transition probability matrix defining transitions between each pair of state
- The states correspond to the possible assignments a variable can state
- One can think of a Markov chain as doing a random walk on a graph with nodes corresponding to each state

A very simple Markov chain

- Suppose we have a time series measurement of a gene's expression level
- Let the gene's expression be discretized and so the gene can take three values: HIGH, MEDIUM, LOW
- Let X_t denote the expression state of the gene at time t
- The temporal nature of this data suggests X_{t+1} depends on X_t
- We can model the time series of gene expression states using a Markov chain

A very simple Markov chain



$P(X_{t+1}=high \mid X_t=low)=0.1$

We will use the $T(X_i/X_j)$ to denote the transition probabilities

Markov Chain and Stationary distributions

- The stationary distribution is a fundamental property of a Markov chain
- Stationary distribution of a Markov Chain specifies the probability of being in a state independent of the starting position
- A Markov chain has a stationary distribution if it is:
  - Irreducible: non-zero probability to all states
  - Aperiodic: has self transition probability
- Not all Markov Chains have a stationary distribution

Stationary distribution of a Markov chain

- Given a Markov chain with transition probabilities $T(X_i/X_k)$
- We define the probability distribution over states at the next time step as X_i as:

$$P_{n+1}(X_i) = \sum_k T(X_i|X_k) P_n(X_k)$$

- When n tends to infinity, $P_n(X_i)$ converges to the stationary distribution $P_\infty(X_i)$

$$P_\infty(X_i) = \sum_k T(X_i|X_k) P_\infty(X_k)$$

Markov Chains for Bayesian network structure learning

- We need to devise a Markov chain over the space of possible graphs such that the stationary distribution of this Markov chain is the posterior distribution of the graph, $P(G|D)$

$$P_\infty(G_i) = P(G_i|D)$$

- Let G_i denote a graph at step i and let G_k denote the graph at previous step k
- We need to define the transition probability of going from G_k *to* G_i

How do we make sure we will draw from the right distribution?

- That is, when the Markov chain has converged to its stationary distribution, how do we make sure that the stationary distribution is the right posterior distribution?
- If the transition probabilities satisfy, a condition called “detailed balance”, we can get the right distribution

$$\frac{T(M_k|M_i)}{T(M_i|M_k)} = \frac{P(G_k|D)}{P(G_i|D)}$$

Markov Chains for Bayesian network structure learning

- In practice, for us to set up a Markov chain for Bayesian network search, we need to propose a new structure, and accept it with some probability
- Let $Q(G_i|G_k)$ denote the proposal probability
  - This is dependent upon the local graph moves we allow
- Let $A(G_i|G_k)$ denote the acceptance probability:
  - This is designed in a way to make the jump to G_i proportional to how well G_i describes the data
- The transition probability is $T(G_i|G_k)=Q(G_i|G_k)A(G_i|G_k)$
- We will keep running the propose and accept steps of our chain until convergence

Acceptance probability

- The acceptance probability is defined as

$$A(G_i|G_k) = \min\left[\frac{P(D|G_i)P(G_i)Q(G_k|G_i)}{P(D|G_k)P(G_k)Q(G_i|G_k)}, 1\right]$$

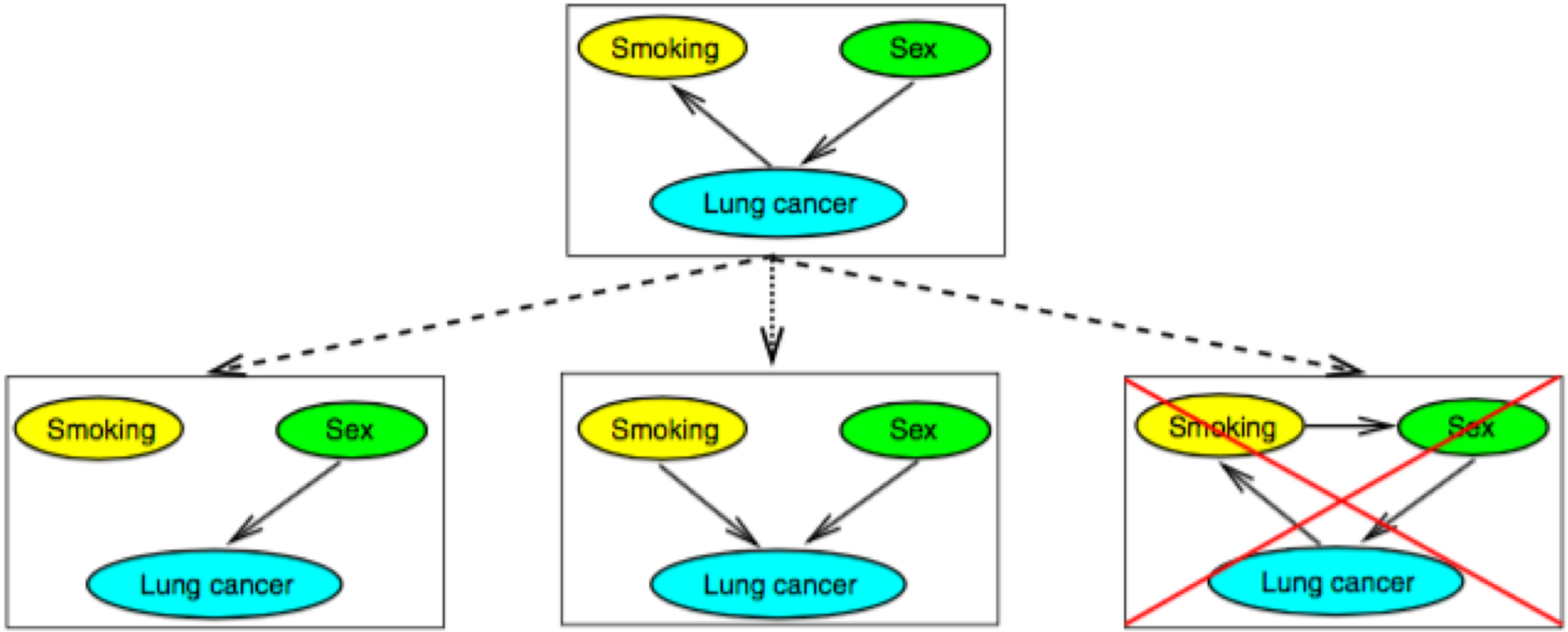
- If the proposal distribution is symmetric, the above simplifies to (this is not the case for DAGs)

$$A(G_i|G_k) = \min\left[\frac{P(D|G_i)P(G_i)}{P(D|G_k)P(G_k)}, 1\right]$$

Metropolis Hastings algorithm

- Start from an initial structure G_0
- Iterate from n=1.. N
  - Propose a new structure G_n from G_{n-1} using $Q(G_n/G_{n-1})$
  - Accept G_n with probability $A(G_n|G_{n-1})$
- Discard an initial "burn in" period to make sure the Markov Chain has reached a stationary distribution
- Using the new samples, estimate different features of the graph, or aggregate different graphs

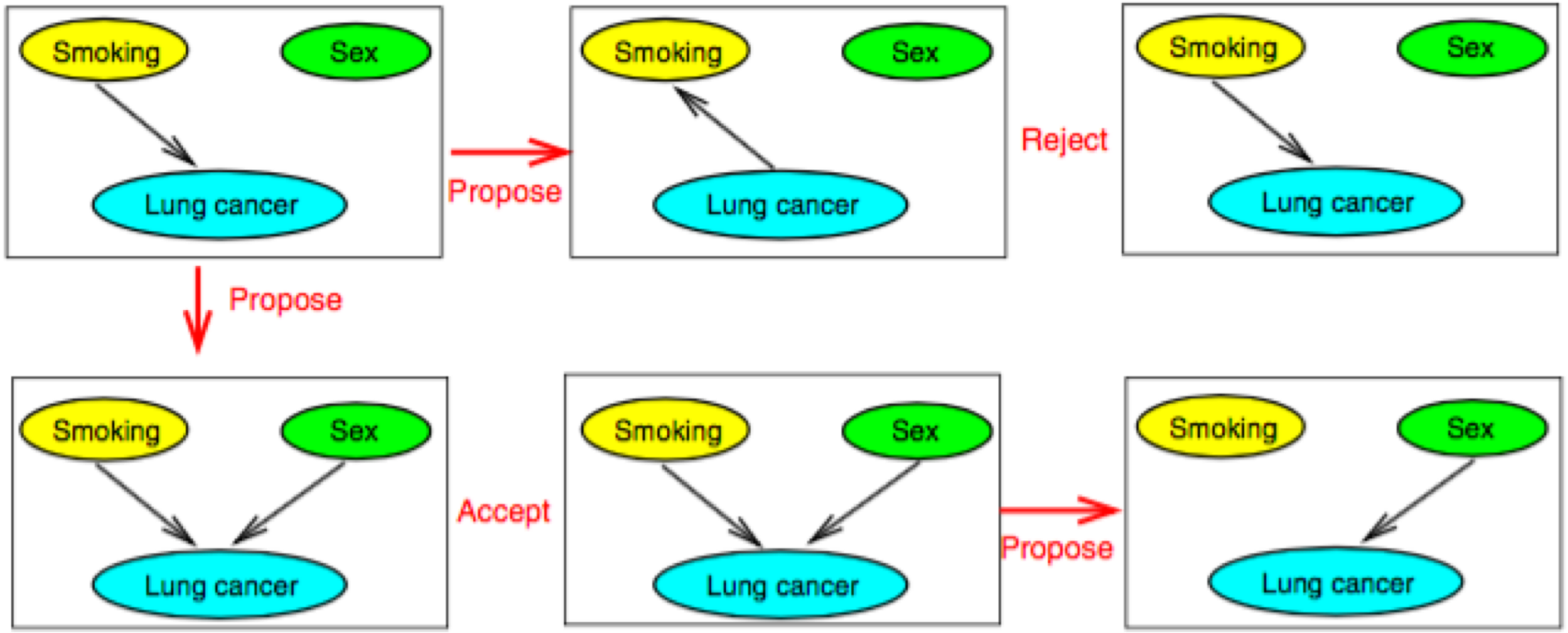
Elementary proposal moves for DAGs



The proposal distribution is defined by the moves on the graph. The above example shows a scenario where we have two valid configurations, and a third invalid configuration.

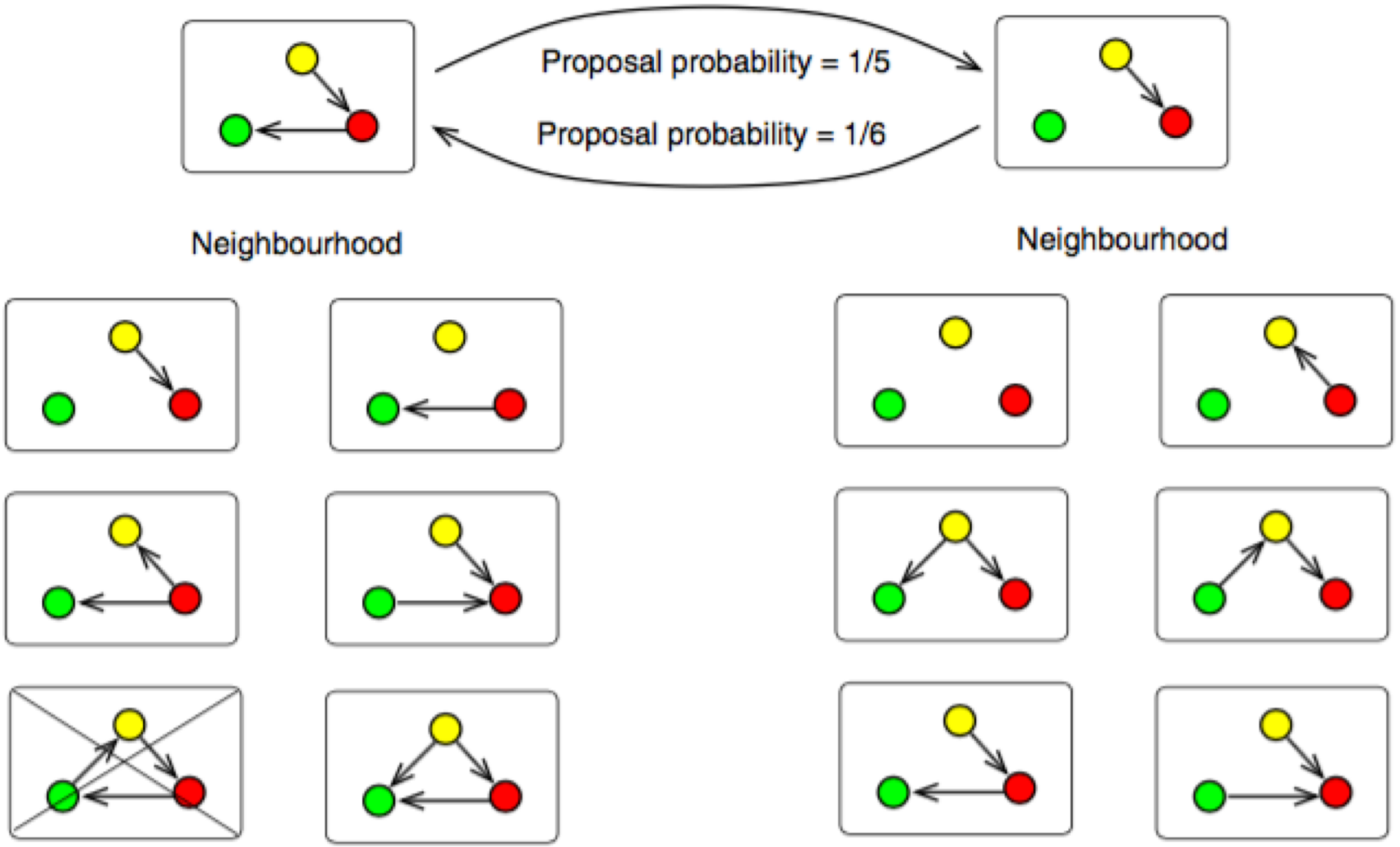
Husmeier, 2005

MCMC example



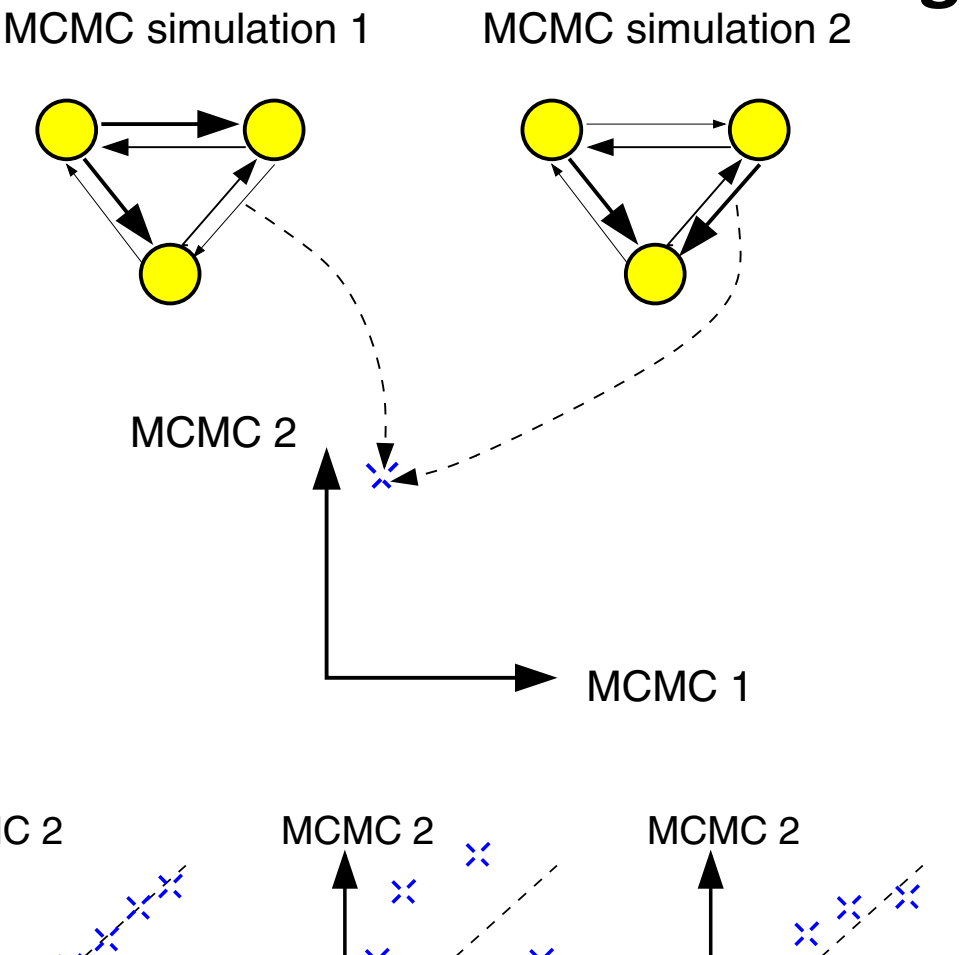
Husmeier 2005

Defining a proposal distribution from elementary moves



Notice that the neighborhood of the two DAGs are not of the same size

A heuristic to check for MCMC convergence



MCMC for learning a graph prior and structure

- Recall that our prior distribution over graphs has a parameter β
- Ideally, we would like to search over the space of priors and structures, that is sample from

$$P(\beta, G|D)$$

- The proposal distribution and the acceptance probabilities need to be updated

MCMC over graph structure and parameters

- We need two proposal distributions, one for the graph structure and one for the hyper parameter
- Proposing new graph structures

$$Q(G_{new}|G_{old})$$

- Proposing new a hyper parameter

$$R(\beta_{new}|\beta_{old})$$

- Accordingly, we need to define new acceptance probabilities

Acceptance probabilities

- Acceptance for the graph

$$A(G_n|G_{n-1}) = \min\left\{\frac{P(D|G_n)P(G_n|\beta_{n-1})Q(G_{n-1}|G_n)}{P(D|G_{n-1})P(G_{n-1}|\beta_{n-1})Q(G_n|G_{n-1})}, 1\right\}$$

- Acceptance for the hyperparameter

$$A(\beta_n|\beta_{n-1}) = \min\left\{\frac{P(G_{n-1}|\beta_n)P(\beta_n)R(\beta_{n-1}|\beta_n)}{P(G_{n-1}|\beta_{n-1})P(\beta_{n-1})R(\beta_n|\beta_{n-1})}, 1\right\}$$

MCMC over graph structure and hyperparameter

- This would proceed in a similar way as before
- We start with an initial configuration $\{G_0, \beta_0\}$
- Repeat for n=1.. N steps
  - Given current value of the hyperparameter β_{n-1} propose G_n from G_{n-1} and accept with $A(G_n|G_{n-1})$
  - Given current G_n propose a new parameter and accept with probability $A(\beta_n|\beta_{n-1})$

Plan for today

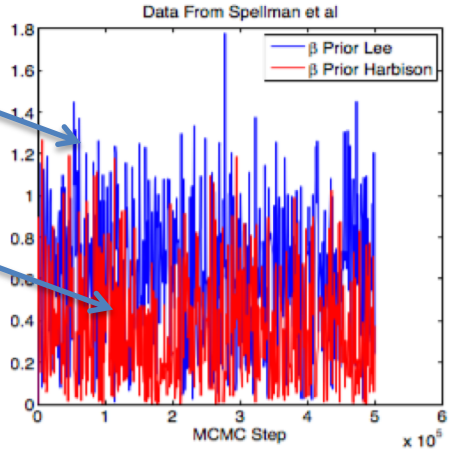
- Overview of integrative network inference
- Defining priors on graph structure
- Learning Bayesian networks with priors using Markov Chain Monte Carlo
- Applications of Bayesian networks with priors
  - Inferring the yeast cell cycle network
  - Inferring cancer signaling

Performance on real data

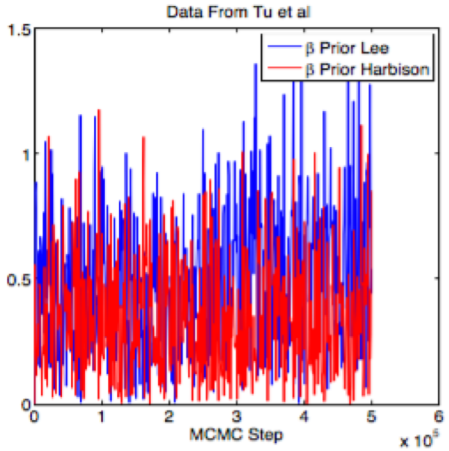
- Two settings
  - Yeast cell cycle time series expression data
    - Two time course datasets were available
    - Two prior networks
  - RAF signaling pathway
    - One non-time course data
    - One prior network
- Questions asked
  - Can different prior networks be distinguished
  - Does prior improve the network inference
  - Are the hyperparameters estimated accurately

Inferred hyperparameters for the yeast cell cycle

Red and blue show the trajectory of the hyperparameter values during the MCMC



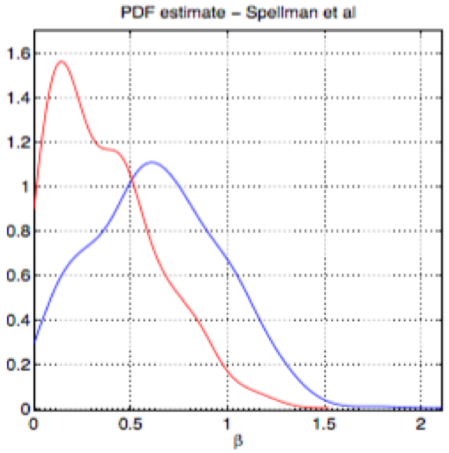
(a)



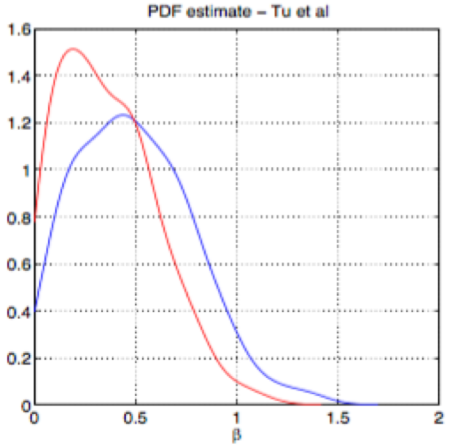
(b)

The two prior networks are very similar

Posterior probability of the hyper parameters: close to 0.



(c)

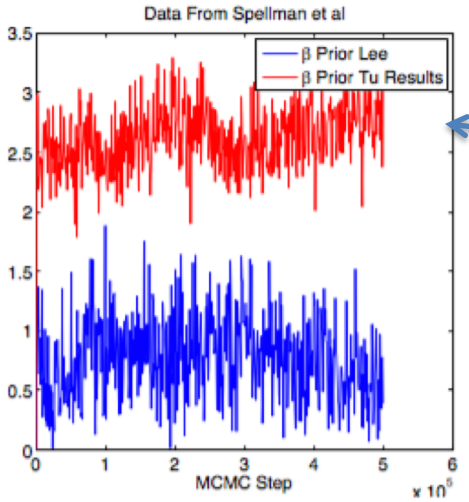


(d)

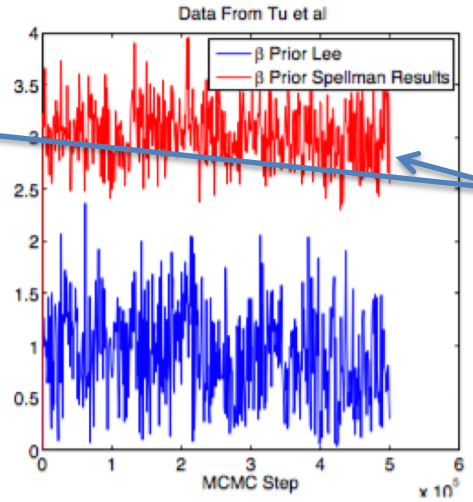
Using a slightly different prior

- Use one of the expression datasets to learn a graph
- Use this graph as one prior and combine with one of the other two binding network priors

Prior hyperparameters can be distinguished

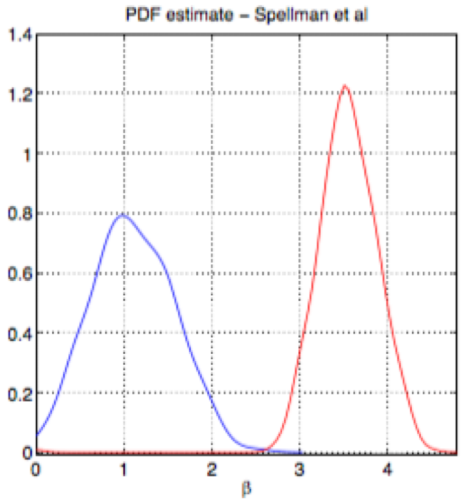


(a)

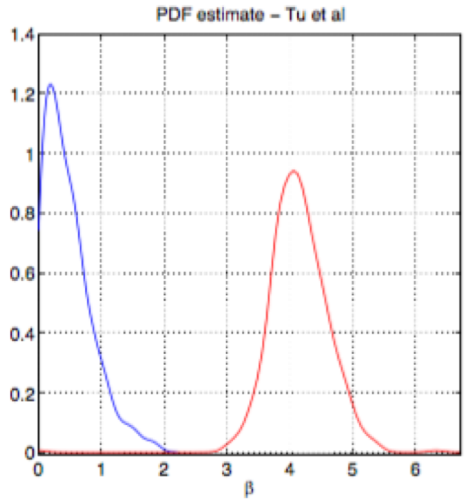


(b)

Prior that is consistent with the data



(c)



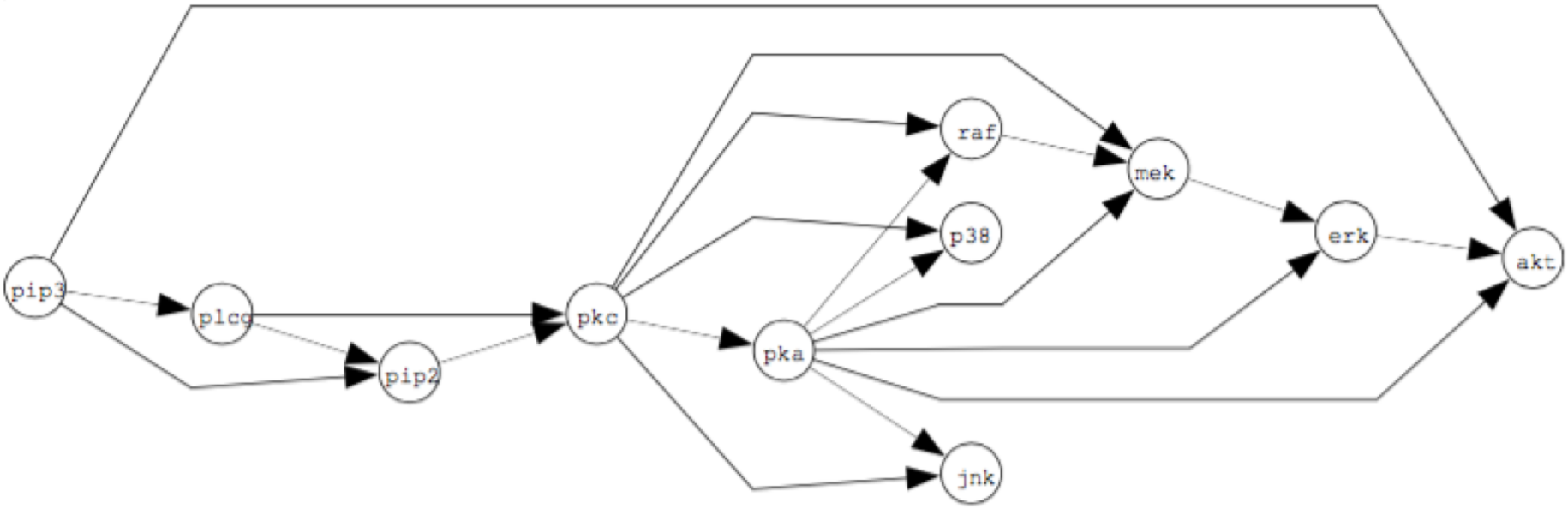
(d)

Conclusions from the Yeast cell cycle study

- None of the TF binding priors appear consistent with the data

- More consistency is observed if a prior network is obtained from an expression dataset

Assessing on a well-studied gold standard network: Raf signaling pathway

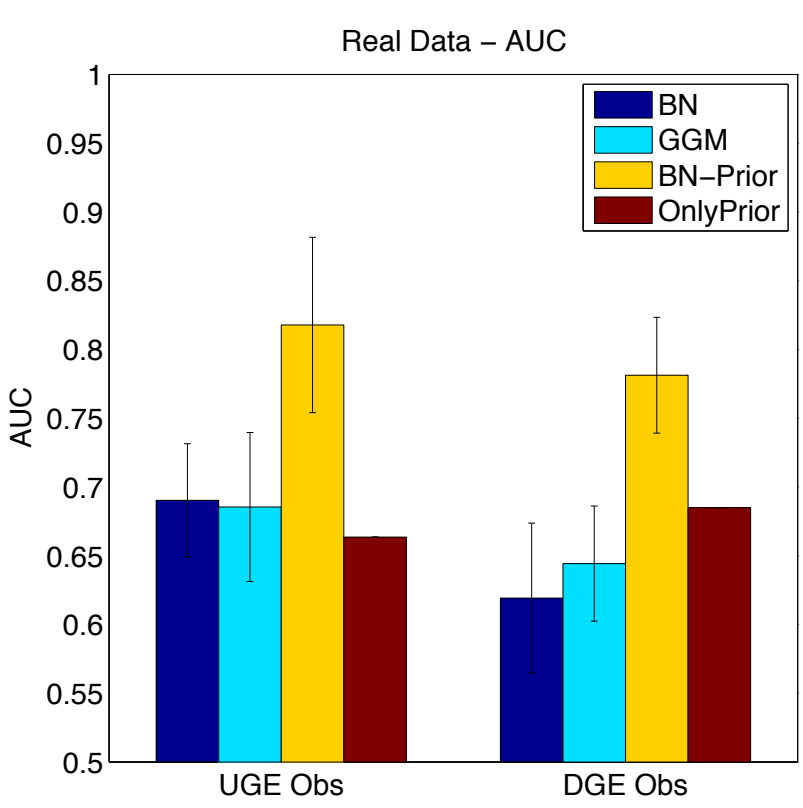


11 phospho proteins in all.

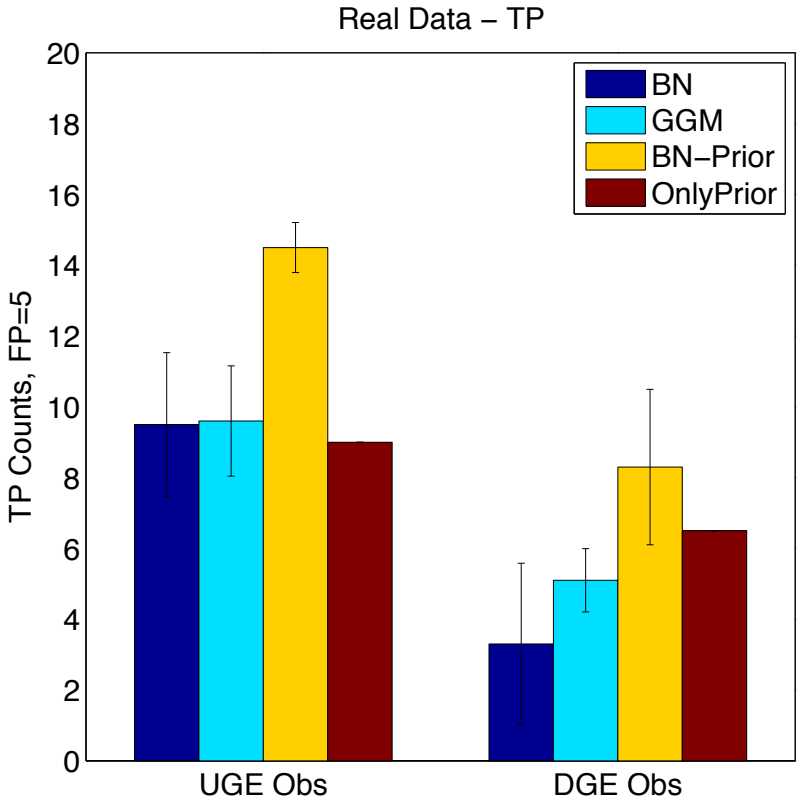
Results on RAF signaling

- The data are not time course
- However the partition function computation is a “tight” upper bound and can be used
- Compare against
  - Prior alone
  - Data alone

Prior helps!

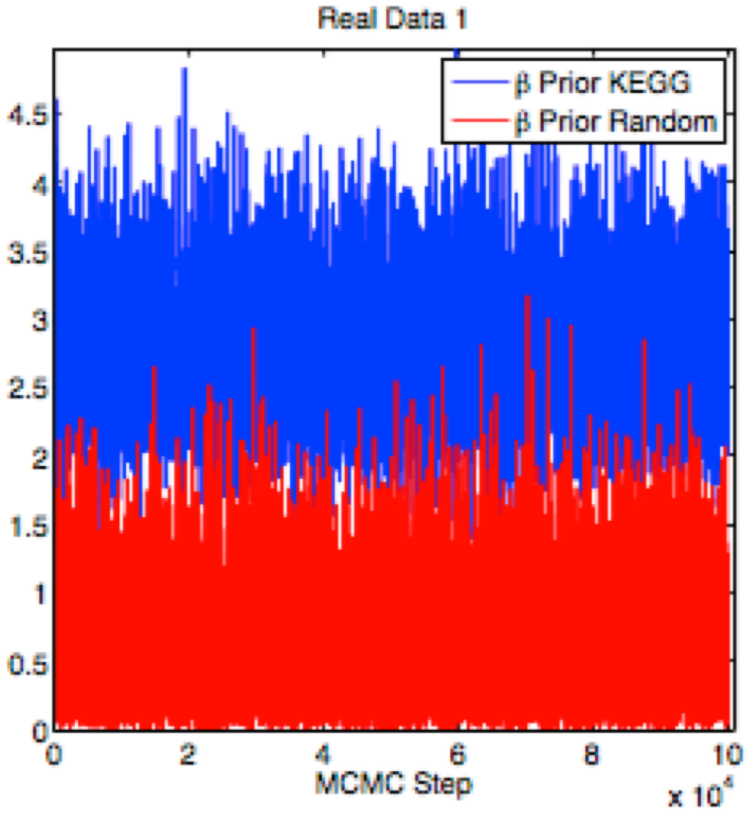


(a)

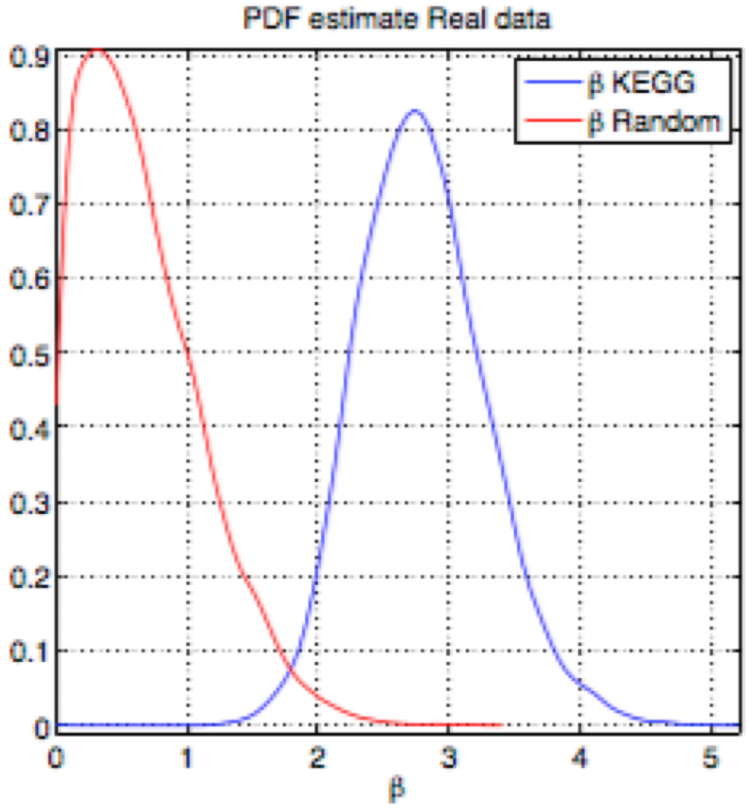


(b)

Method can discriminate between true and random prior



(a)



(b)

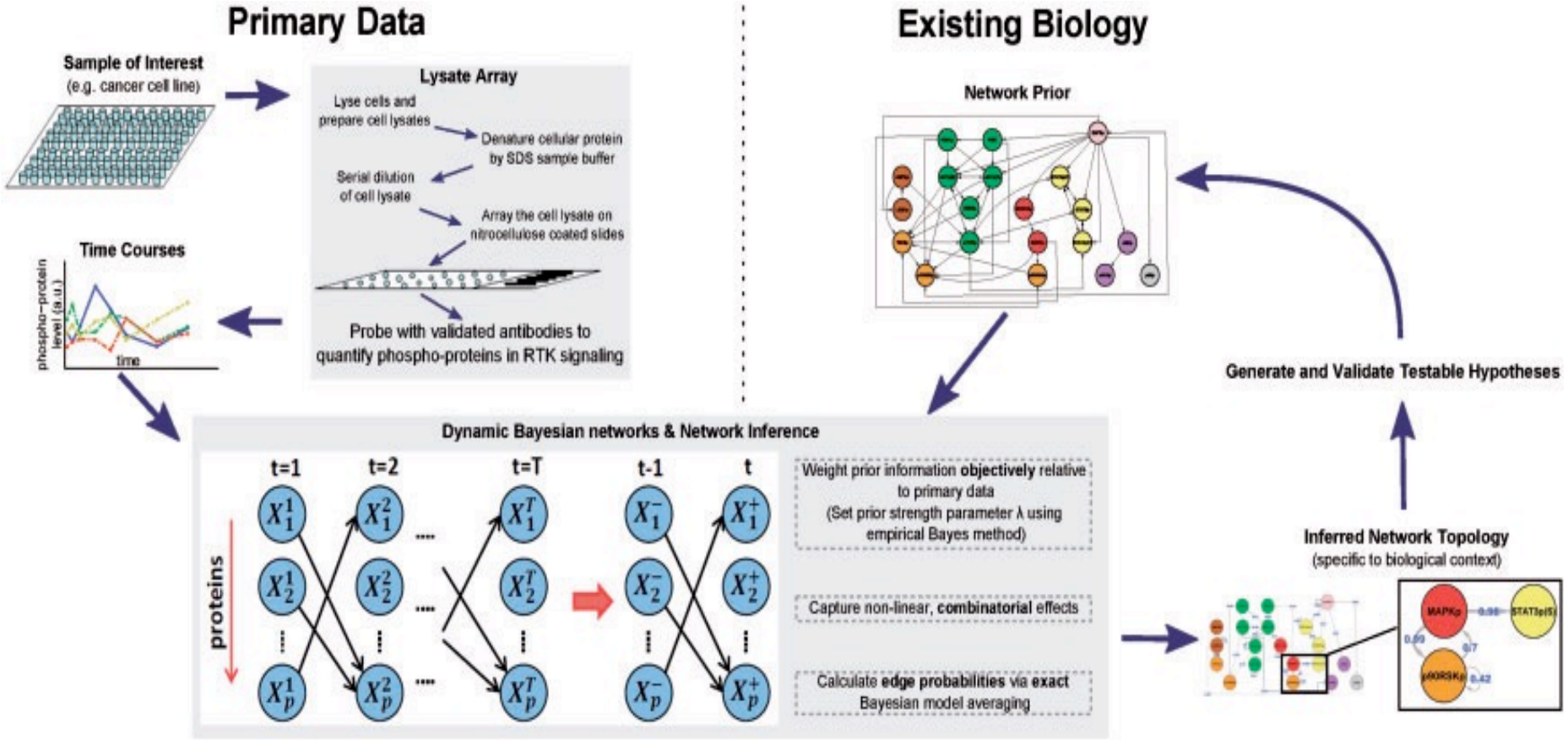
Plan for today

- Overview of integrative network inference
- Defining priors on graph structure
- Learning Bayesian networks with priors using Markov Chain Monte Carlo
- Applications of Bayesian networks with priors
  - Inferring the yeast cell cycle network
  - Inferring cancer signaling

Bayesian Inference of Signaling Network Topology in a Cancer Cell Line (Hill et al 2012)

- Protein signaling networks are important for many cellular diseases
  - The networks can differ between normal and disease cell types
- But the structure of the network remains incomplete
- Temporal activity of interesting proteins can be measured over time, that can be used infer the network structure
- Build on prior knowledge of signaling networks to learn a better, predictive network
- BNs are limiting because they do not model time

Applying DBNs to infer signaling network topology



Hill et al., Bioinformatics 2012

Application of DBNs to signaling networks

- Dataset description
  - Phospho-protein levels of 20 proteins
  - Eight time points
  - Four growth conditions
- Use known signaling network as a graph prior
- Estimate CPDs as conditional regularized Gaussians
- Assume a first-order Markov model
  - X^t depends on on X^{t-1}

Integrating prior signaling network into the DBN

- A Bayesian approach to graph learning

$$P(G|\mathbf{D}) \propto P(\mathbf{D}|G)P(G)$$

Data likelihood Graph prior

- Graph prior is encoded as (Following Mukherjee & Speed 2008)

$$P(G) \propto \exp(\lambda f(G))$$

Prior strength Graph features

- Where $f(G)=-|\mathrm{E}(\mathrm{G})\backslash\mathrm{E}^*|$ is defined as the number of edges in the graph G, $E(G)$, that are not in the prior set E^*
- This prior does not promote new edges, but penalizes edges that are not in the prior

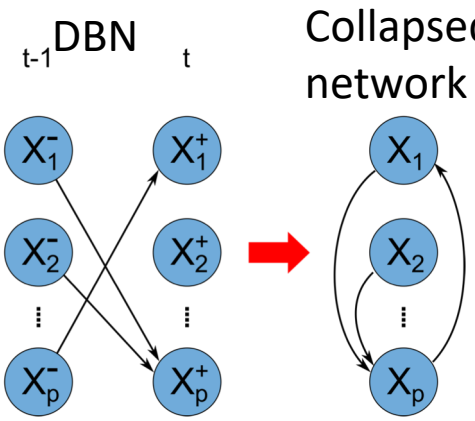
Calculating posterior probabilities of edges

- For each edge *e*, we need to calculate

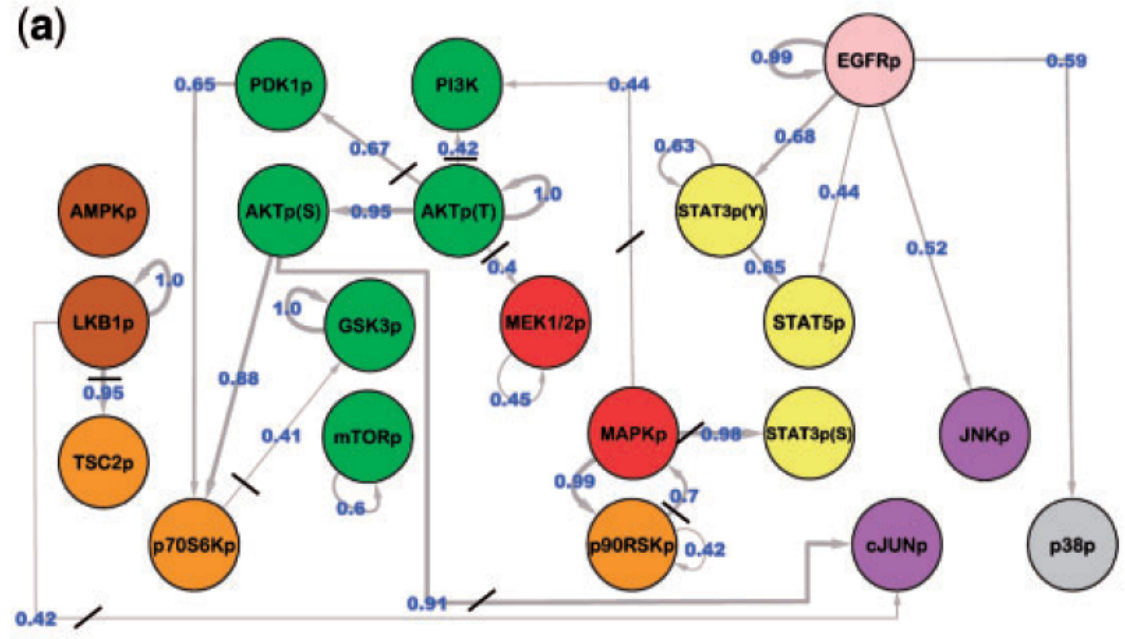
$$P(e|\mathbf{D}) = \Sigma_{G \in \mathcal{G}} I_{e \in G} P(G|\mathbf{D})$$

- Although this is intractable in general, this work makes some assumptions
  - Allow edges only forward in time
    - The learning problem decomposes to smaller per-variable problems that can be solved by variable selection
  - Assume $P(G)$ factorizes over individual parent sets
  - To compute the posterior probability, the sum goes over all possible parent sets
    - Assume a node can have no more than $\mathrm{d}_{\max}$ parents

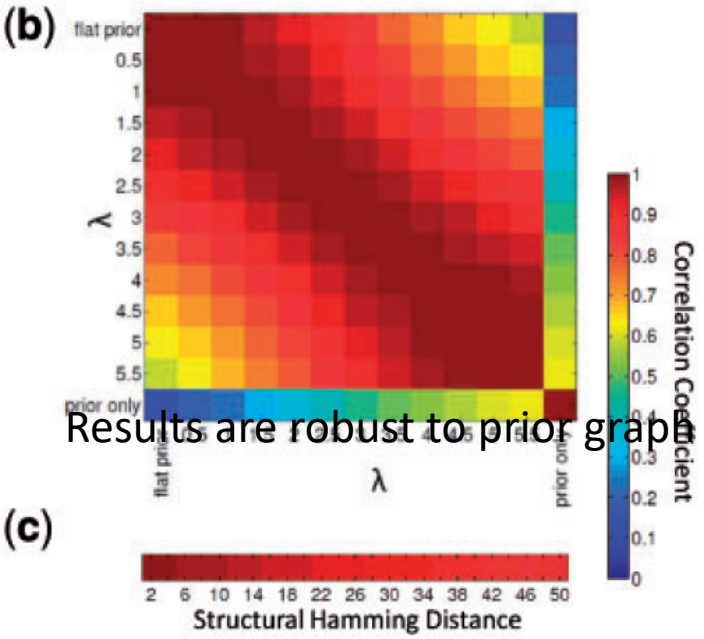
Inferred signaling network using a DBN



Inferred signaling network



Results are not sensitive to prior values



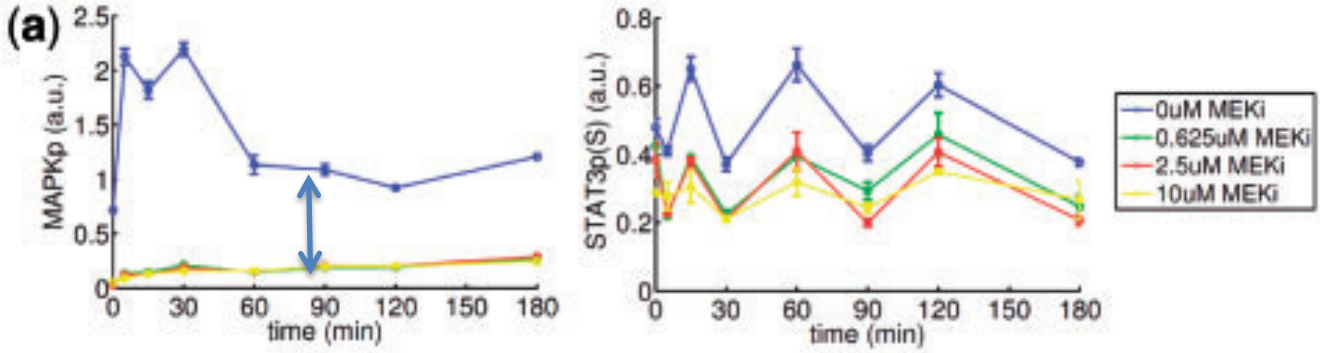
Results are robust to prior graph

Using the DBN to make predictions

- Although many edges were expected, several edges were unexpected
- Select novel edges based on posterior probability and test them based on inhibitors
- For example, if an edge was observed from X to Y, inhibition of X should affect the value of Y if X is a causal regulator of Y
- Example edge tested
  - MAPKp to STAT3p(S727) with high probability (0.98)
    - Apply MEKi, which is an inhibitor of MAPK, and measure MAPKp and STAT3p post inhibition
  - AKTp to p70S6Kp, AKTp to MEKp and AKTp to cJUNp

Experimental validation of links

Add MAPK inhibitor and measure MAPK and STAT3



MAPK is significantly inhibited (P-value $5X10^{-4}$)
STAT3 is also inhibited (P-value $3.3X10^{-4}$)

Their success is measured by the difference in the levels of the targets as a function of the levels of the inhibitors

Summary

- Prior knowledge can be incorporated as a energy functions on a graph and used to define a prior distribution
  - Extensible to multiple priors
- Markov Chain Monte Carlo (MCMC) sampling approach enables us to search over the graph and hyperparameter space
- MCMC can distinguish between good and bad (inconsistent priors)
- Adding prior helped network structure learning for a small gold-standard network
- Adding priors was also helpful in simulations for the cancer signaling network

References

- Introduction to Learning Bayesian Networks from Data. Dirk Husmeier 2005
- Reconstructing Gene Regulatory Networks with Bayesian Networks by Combining Expression Data with Multiple Sources of Prior Knowledge. Adriano V. Werhli and Dirk Husmeier 2007
- Bayesian inference of signaling network topology in a cancer cell line. S. M. Hill, Y. Lu, J. Molina, L. M. Heiser, P. T. Spellman, T. P. Speed, J. W. Gray, G. B. Mills, and S. Mukherjee, *Bioinformatics (Oxford, England)*, vol. 28, no. 21, pp. 2804-2810, Nov. 2012.Hill et al., Bio