Causal Graphical Models

Elias Chaibub Neto and Brian S Yandell

SISG 2012

July 12, 2012

Correlation and Causation

The ideal . . . is the study of the direct influence of one condition on another . . . [when] all other possible causes of variation are eliminated . . . The degree of correlation between two variables . . . [includes] all connecting paths of influence . . .. [Path coefficients combine] knowledge of . . . correlation among the variables in a system with . . . causal relations.

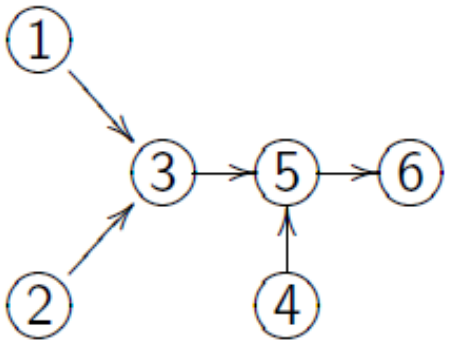
Sewall Wright (1921)

Basic concepts

Directed graphical models

A graphical model is a multivariate probabilistic model whose conditional independence relations are represented by a graph.

We will focus on directed acyclic graph (DAG) models (aka Bayes nets),



Assuming the Markov property, the joint distribution factors according to the conditional independence relations:

$$P(1,2,3,4,5,6) = P(6 \mid 5)\, P(5 \mid 3,4)\, P(4)\, P(3 \mid 1,2)\, P(2)\, P(1)$$

$$6 \perp\!\!\!\perp \{1,2,3,4\} \mid 5, \quad 5 \perp\!\!\!\perp \{1,2,3\} \mid 4, \quad \text{and so on}$$

i.e., each node is independent of its non-descendants given its parents.

Standard Bayesian networks and causality

Even though the direct edges in a Bayes net are often interpreted as causal relations, in reality they only represent conditional dependencies.

Different phenotype networks, for instance,

$$Y_1 \to Y_2 \to Y_3 \ , \quad Y_1 \leftarrow Y_2 \to Y_3 \ , \quad Y_1 \leftarrow Y_2 \leftarrow Y_3 \ ,$$

can represent the same set of conditional independence relations ($Y_1 \perp\!\!\!\perp Y_3 \mid Y_2$, in this example). When that is the case, we say the nets are *Markov equivalent*.

In general (although it is not always true), Markov equivalent networks will have equivalent likelihood functions, so that model selection criteria cannot distinguish between them. The best we can do is to learn *equivalent classes* of *likelihood equivalent* phenotype networks from the data.

Genetics as a mean to reduce the size of equivalence classes

The incorporation of genetic information can help distinguish between likelihood equivalent nets two distinct ways:

1. By creating priors for the network structures, using the results of causality tests (Zhu et al. 2007).

2. By augmenting the phenotype network with QTL nodes, creating new sets of conditional independence relations (Chaibub Neto et al. 2008, 2010).

Genetic priors

Consider the networks

$$G_Y^1 : \; Y_1 \rightarrow Y_2 \rightarrow Y_3 \; , \quad G_Y^2 : \; Y_1 \leftarrow Y_2 \leftarrow Y_3 \; .$$

These Markov equivalent networks have the same likelihood, i.e.,

$$P(Y \mid G_Y^1) = P(Y \mid G_Y^2) \; .$$

If the phenotypes are associated with QTLs, we can use the results of the causality tests to compute prior probabilities for the network structures. If

$$\frac{P(G_Y^1)}{P(G_Y^2)} \neq 1 \; , \quad \text{then} \quad \frac{P(G_Y^1 \mid Y)}{P(G_Y^2 \mid Y)} = \frac{P(G_Y^1)}{P(G_Y^2)} \neq 1 \; ,$$

and we can use the posterior probability ratio to distinguish between the networks.

Augmenting the phenotype network with QTL nodes

By augmenting the phenotype network with a QTL node,

$$G^1 : \; Q \to Y_1 \to Y_2 \to Y_3 \; , \quad G^2 : \; Q \to Y_1 \leftarrow Y_2 \leftarrow Y_3 \; ,$$

we have that G^1 and G^2 have distinct sets of conditional independence relations:

$$Y_2 \perp\!\!\!\perp Q \mid Y_1 \; , \quad \text{on} \quad G^1$$
$$Y_2 \not\perp\!\!\!\perp Q \mid Y_1 \; , \quad \text{on} \quad G^2$$

Hence, G^1 and G^2 are no longer likelihood equivalent.

In the inferential approaches we address here we adopt this augmentation approach.

d-separation

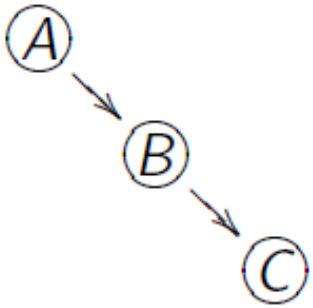
Graphical criterion to read out conditional independence relations from a DAG.

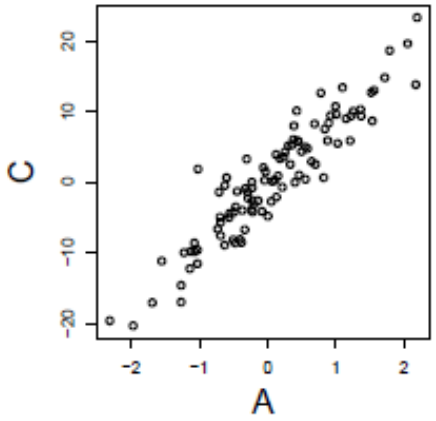
Definition (d-separation): A path p is said to be d-separated (or blocked) by a set of nodes Z if and only if

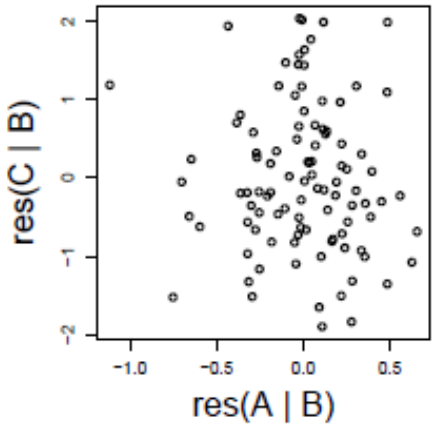
1. p contains a chain $i \to m \to j$ or a fork $i \leftarrow m \to j$ such that the middle node m is in Z, or
2. p contains an inverted fork (or collider) $i \to m \leftarrow j$ such that the middle node m is not in Z and such that no descendant of m is in Z.

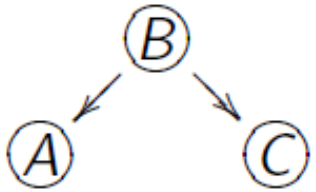
A set Z is said to d-separate X from Y if and only if Z blocks every path from a node in X to a node in Y. X and Y are d-connected if they are not d-separated (Pearl, 1988, 2000).

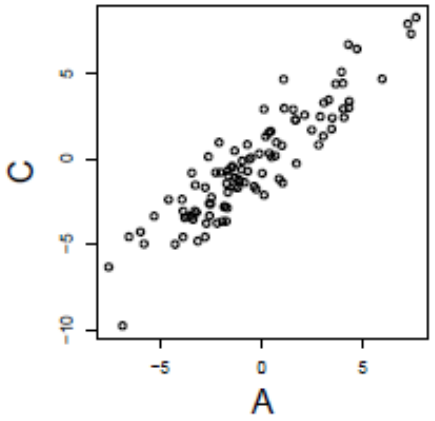
d-separation

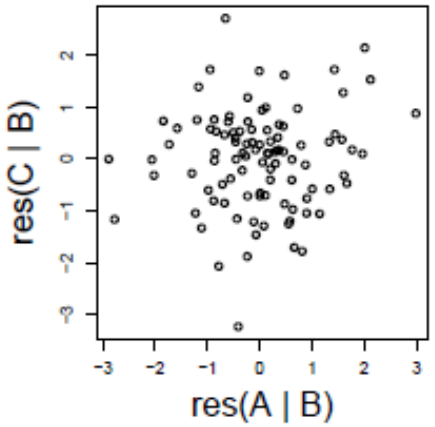


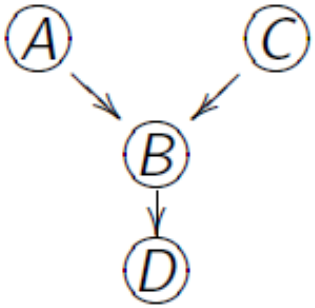


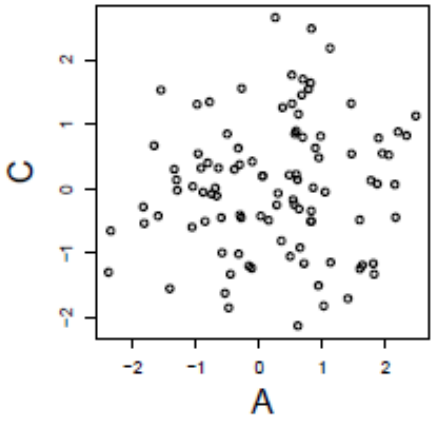


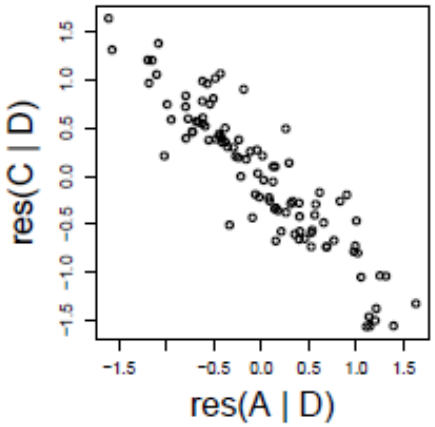










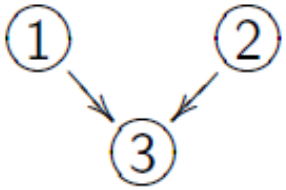


Simple graphical criterium to detect Markov equivalence

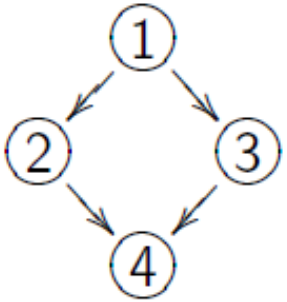
Detecting Markov equivalence: Two DAGs are Markov equivalent if and only if they have the same skeletons and the same set of v-structures. (Verma and Pearl 1990).

The **skeleton** of a causal graph is the undirected graph obtained by replacing its arrows by undirected edges.

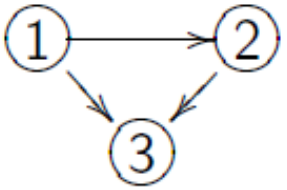
A **v-structure** is composed by two converging arrows whose tails are not connected by an arrow.



v-structure



v-structure



not a v-structure

Simple graphical criterium to detect Markov equivalence

DAG structures	skeletons	v-structures
$Y_1 \to Y_2 \to Y_3$	$Y_1 - Y_2 - Y_3$	$\emptyset$
$Y_1 \to Y_2 \leftarrow Y_3$	$Y_1 - Y_2 - Y_3$	$Y_1 \to Y_2 \leftarrow Y_3$
$Y_1 \leftarrow Y_2 \to Y_3$	$Y_1 - Y_2 - Y_3$	$\emptyset$

Extended DAG structures	skeletons	v-structures
$Q_1 \to Y_1 \to Y_2 \to Y_3$	$Q - Y_1 - Y_2 - Y_3$	$\emptyset$
$Q_1 \to Y_1 \leftarrow Y_2 \to Y_3$	$Q - Y_1 - Y_2 - Y_3$	$Q \to Y_1 \leftarrow Y_2$

Faithfulness assumption

Given a graph and a probability distribution associated with it, all the conditional independence relations spanned by a probability distribution must match the d-separation relations predicted from the graph structure (Spirtes et al. 2000).

Unfaithfulness example:

$$Y_1 = \epsilon_1 \,, \quad Y_2 = \beta_{21}\, Y_1 + \epsilon_2 \,, \quad Y_3 = \beta_{31}\, Y_1 + \beta_{32}\, Y_2 + \epsilon_3$$

$$\epsilon_k \sim N(0, \sigma_k^2) \,, \quad Cov(Y_1\,, Y_3) = (\beta_{31} + \beta_{32}\,\beta_{21})\,\sigma_1^2$$

$$\text{If } \beta_{31} = -\beta_{32}\beta_{21} \text{ then } Cov(Y_1\,, Y_3) = 0.$$

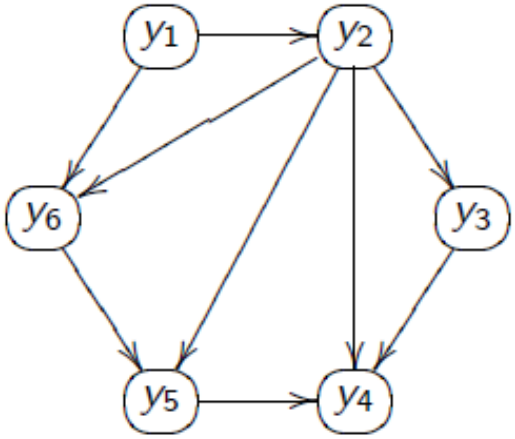
Although the data is generated from *a*, its probability distribution is faithful to *b*.

The PC skeleton algorithm

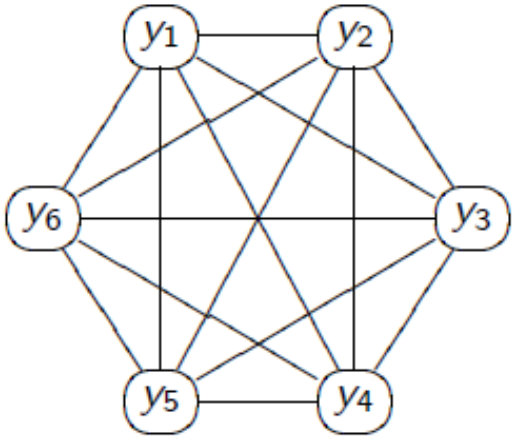
Infers the skeleton of the causal model (Spirtes et al. 1993).

PC skeleton algorithm

Suppose the true network describing the causal relationships between six transcripts is



The PC-algorithm starts with the complete undirected graph



and progressively eliminates edges based on conditional independence tests.

PC skeleton algorithm

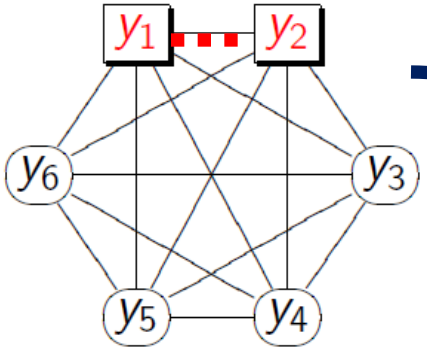
The algorithm performs several rounds of conditional independence tests of increasing order.

It starts with all zero order tests, then performs all first order, second order, and so on.

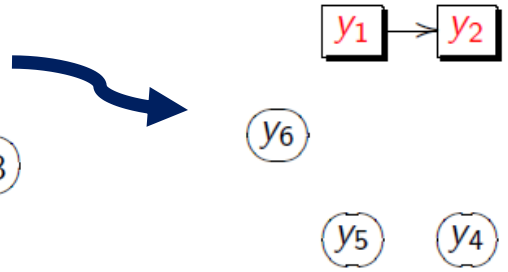
- Remark: in the Gaussian case zero partial correlation implies conditional independence, thus

$$i \perp\!\!\!\perp j \mid k \quad \Leftrightarrow \quad cor(i, j \mid k) = 0 \quad \Rightarrow \quad \text{drop } (i, j) \text{ edge}$$

PC algorithm - zero order

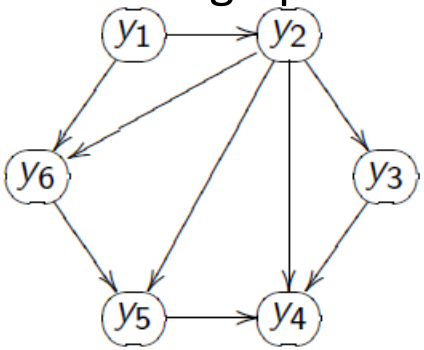


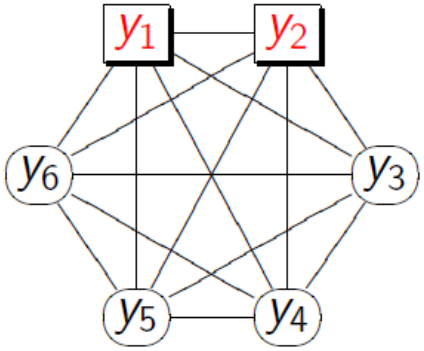
initial network



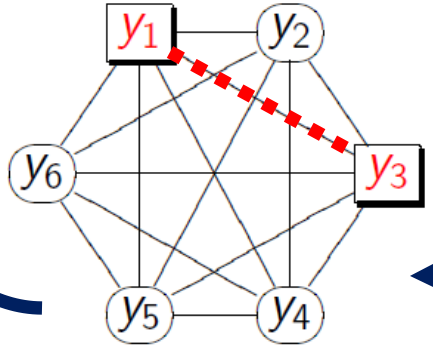
direct effect of y_1 on y_2

true graph

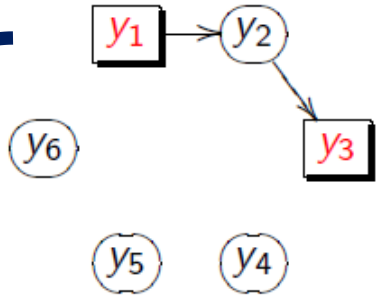




keep edge



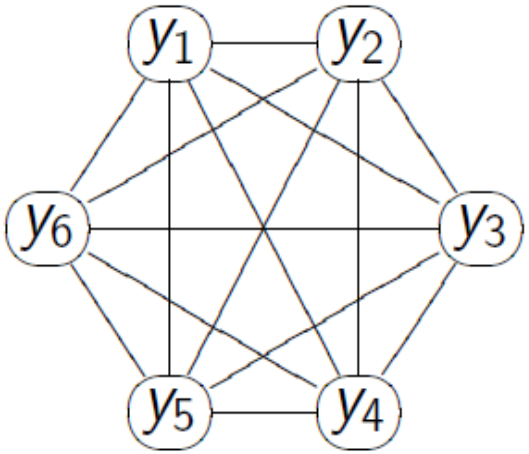
move to next edge



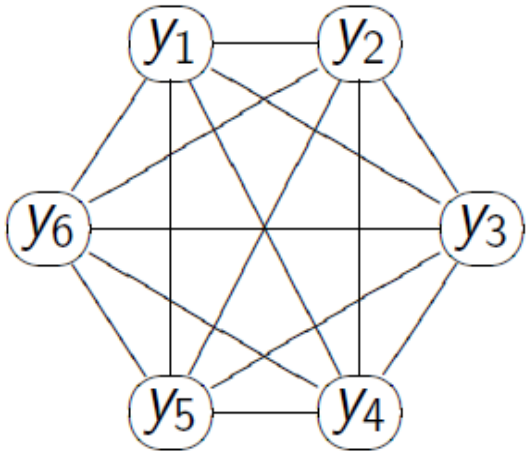
indirect effect of y_1 on y_3

and
so on

PC algorithm - zero order



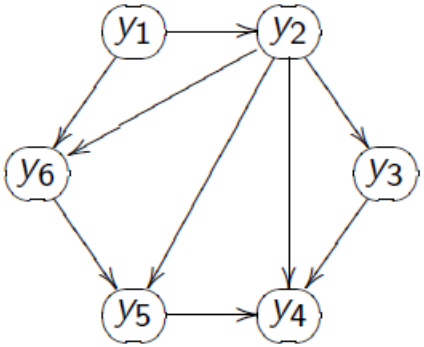
After all zero order conditional independence tests

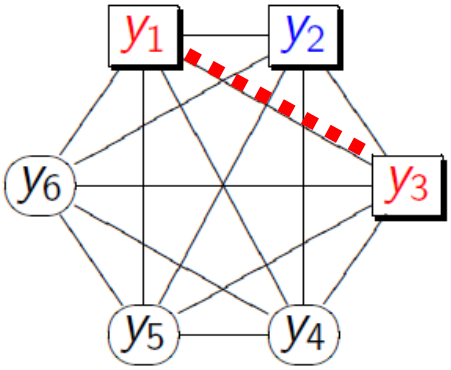


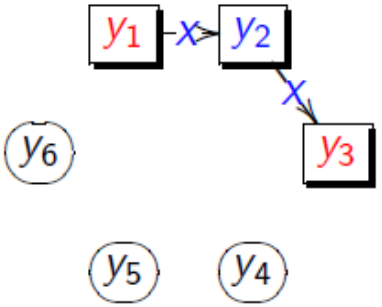
The algorithm then moves to first order conditional independence tests.

PC algorithm - first order

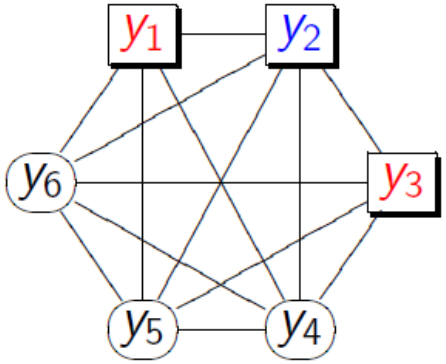
true graph







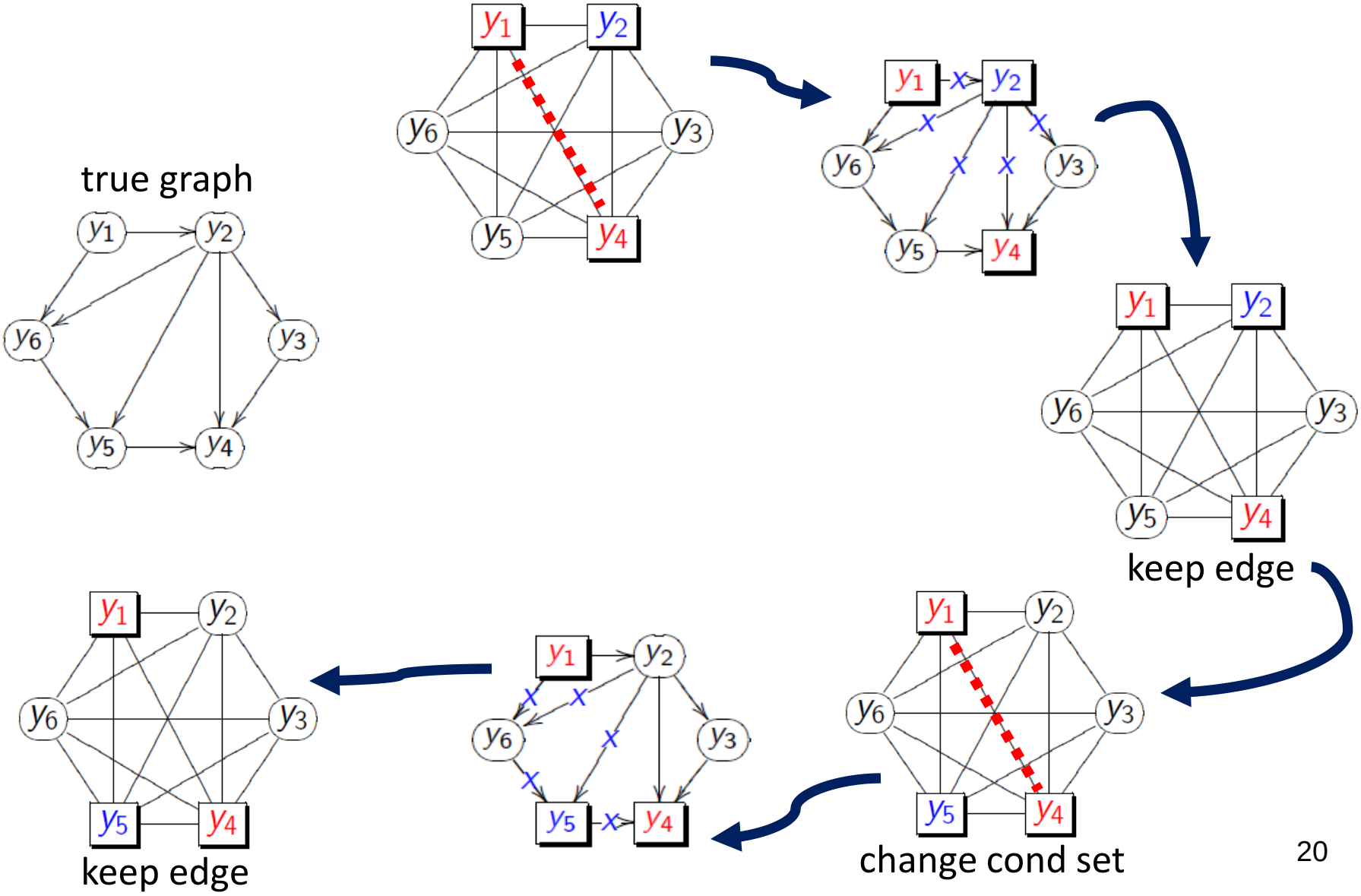
y_2 d-separates y_1 from y_3



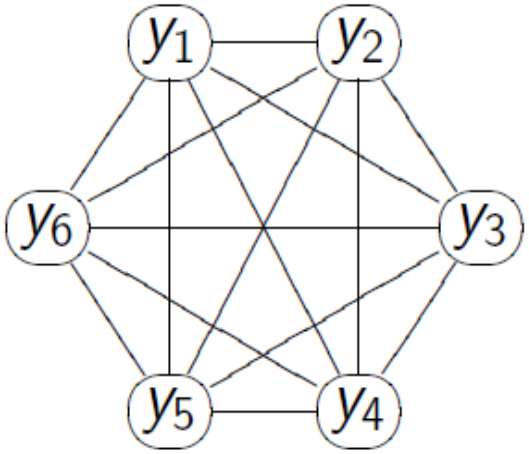
drop edge

Move to next edge

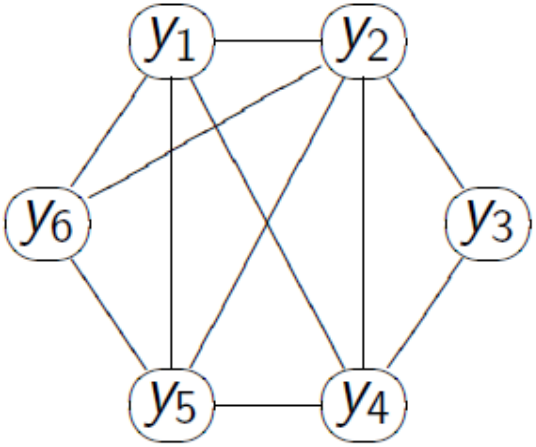
PC algorithm - first order



PC algorithm - first order



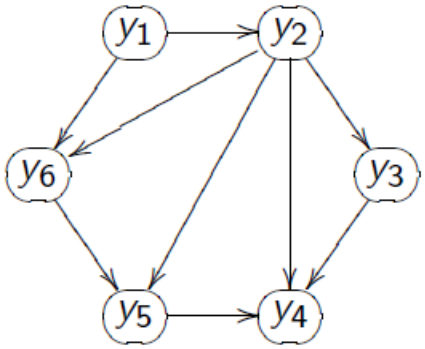
After all first order conditional independence tests.

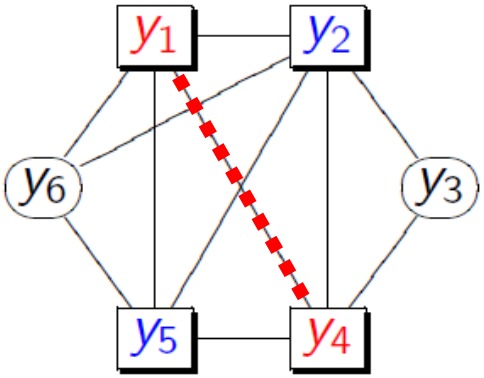


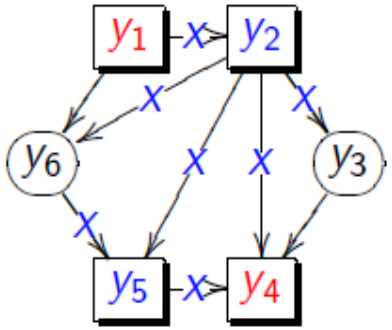
The algorithm then moves to second order conditional independence tests.

PC algorithm - second order

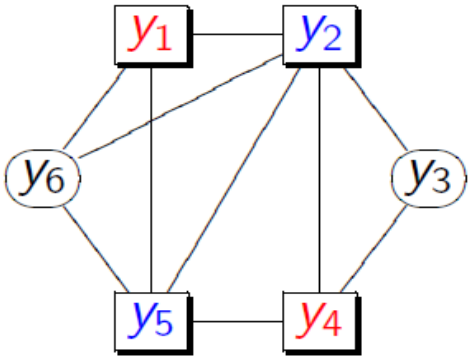
true graph







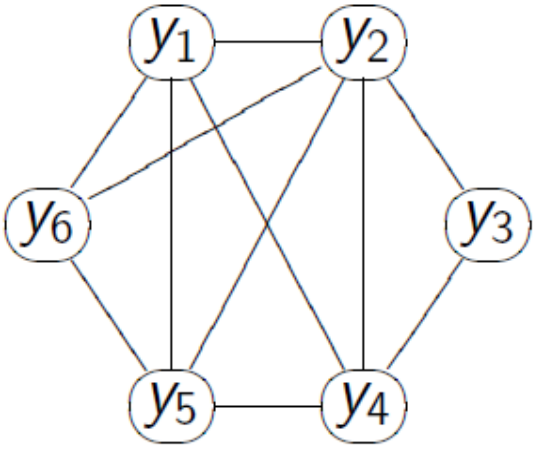
(y_2, y_5) d-separate y_1 from y_4



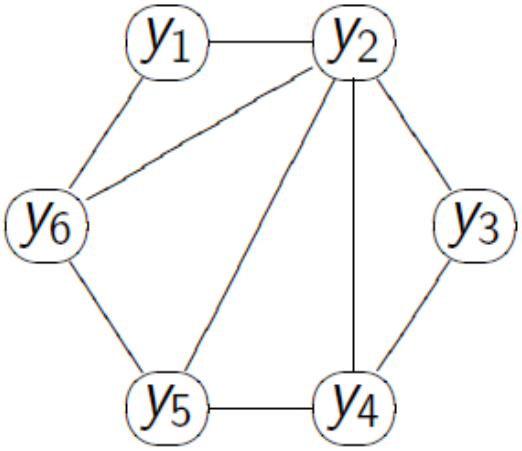
drop edge

move to
next edge

PC algorithm - second order



After all second order conditional independence tests



Then the algorithm moves to third order, fourth order …

Edge orientarion with the QDG algorithm

Edge orientation

We perform model selection using a direction LOD score

$$LOD = \log_{10} \left\{ \frac{\prod_{i=1}^{n} f(y_{1i} \mid \mathbf{q}_{1i}) f(y_{2i} \mid y_{1i}, \mathbf{q}_{2i})}{\prod_{i=1}^{n} f(y_{2i} \mid \mathbf{q}_{2i}) f(y_{1i} \mid y_{2i}, \mathbf{q}_{1i})} \right\}$$

where $f()$ represents the predictive density, that is, the sampling model with parameters replaced by the corresponding maximum likelihood estimates.

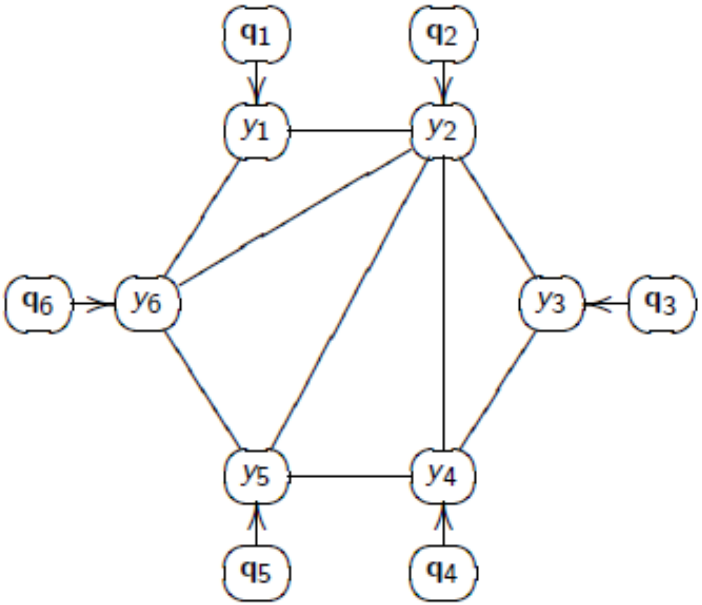
QDG algorithm

The QTL-driven Dependency Graph algorithm is composed of 7 steps:

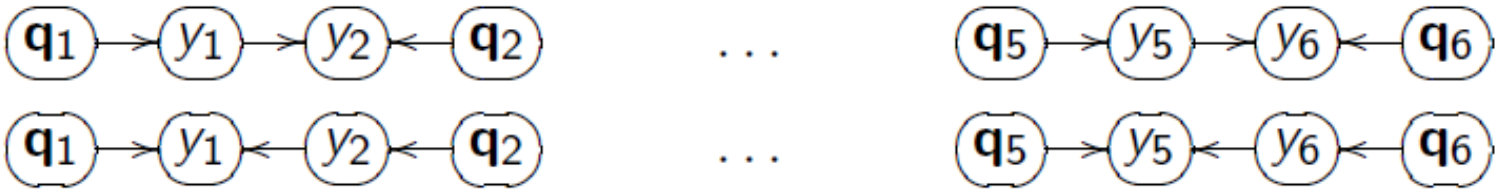
1. Get the causal skeleton (with the PC skeleton algorithm).
2. Use QTLs to orient the edges in the skeleton.
3. Choose a random ordering of edges, and
4. Recompute orientations incorporating causal phenotypes in the models (update the causal model according to changes in directions).
5. Repeat 4 iteratively until no more edges change direction (the resulting graph is one solution).
6. Repeat steps 3, 4, and 5 many times and store all different solutions.
7. Score all solutions and select the graph with best score.

QDG algorithm - step 2

Now suppose that for each transcript we have a set of e-QTLs

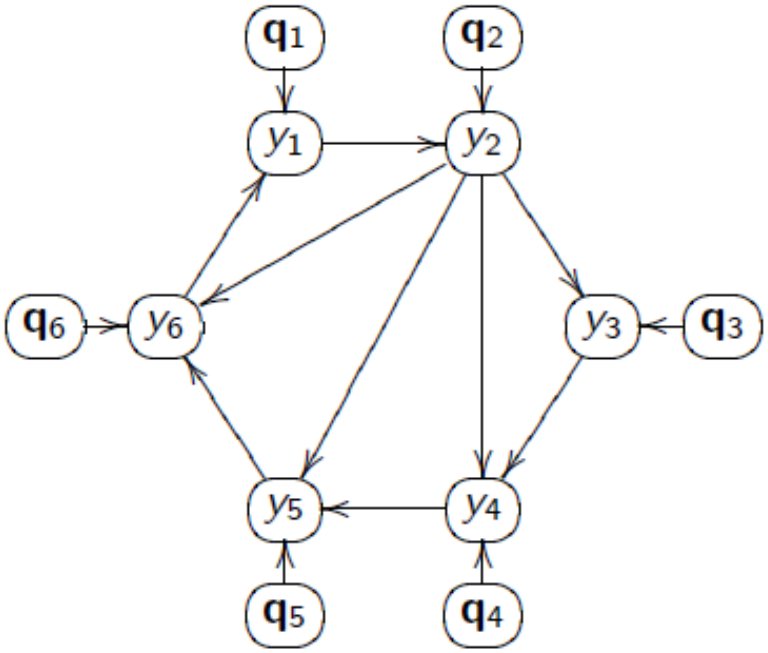


Given the QTLs we can distinguish causal direction:

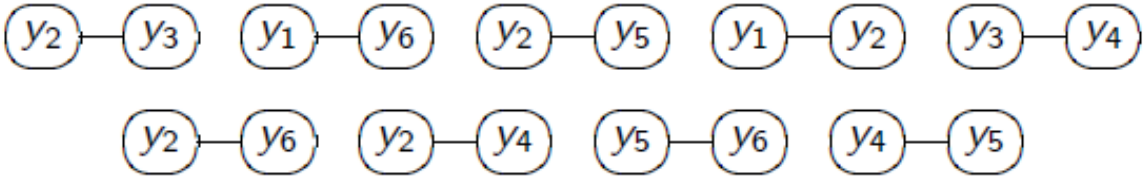


QDG algorithm - steps 2 and 3

First estimate of the causal model, DG_0, (using only QTLs to infer causal direction)



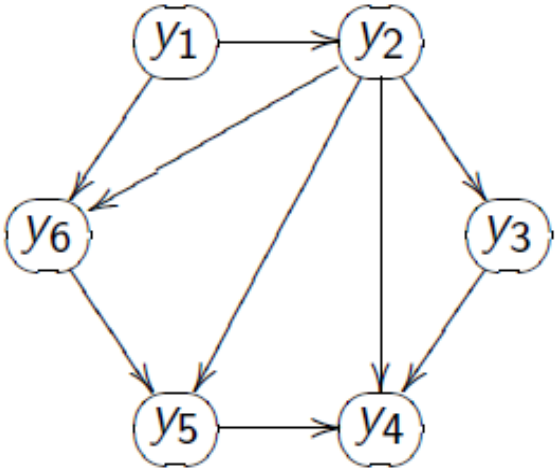
In step 3 we randomly choose an ordering of all edges in DG_0. Say,

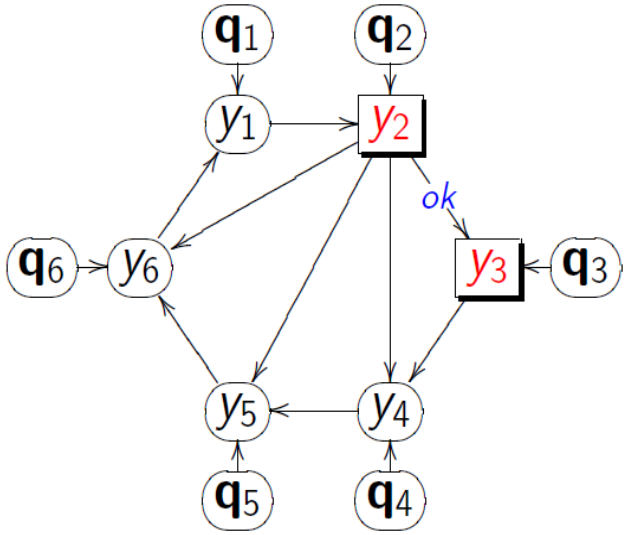


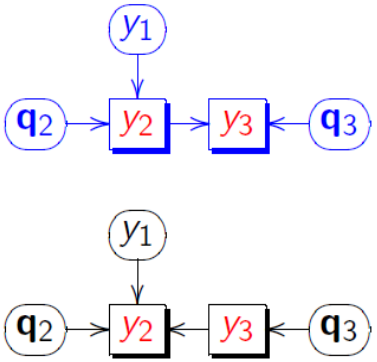
In step 4 we recompute the directions including other transcripts as covariates in the models (following the above ordering).

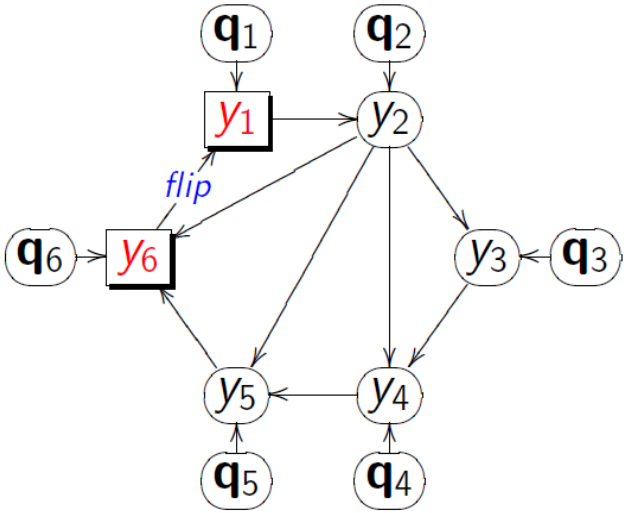
QDG algorithm - step 4

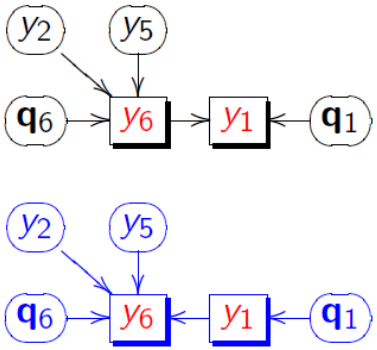
true graph











QDG algorithm - steps 5, 6, and 7

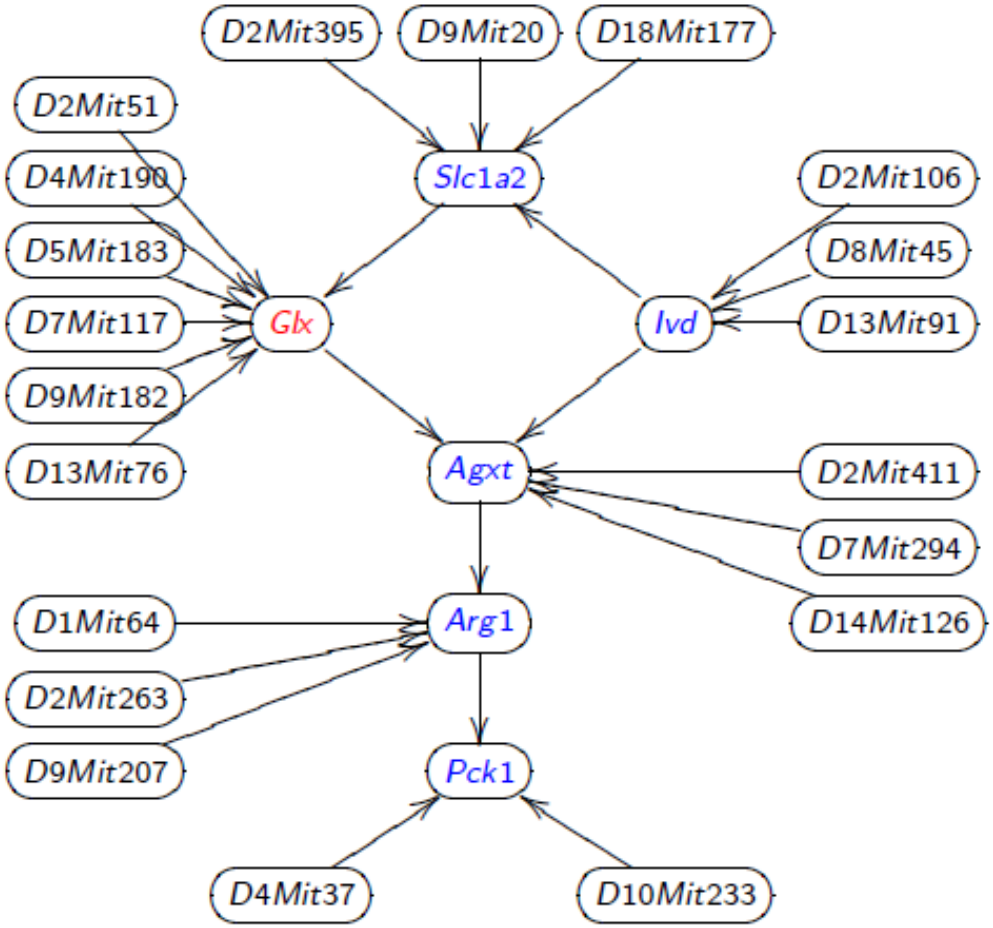
Step 5: repeat 4 iteratively until no more edges change direction (the resulting graph is one solution).

Step 6: repeat the process starting from different random orderings several times, and store all different solutions.

Step 7: score all solutions and select the graph with best score.

Real data example

Network of metabolites and transcripts involved in liver metabolism.



Four out of six predictions were validated experimentally (Ferrara et al. 2008).

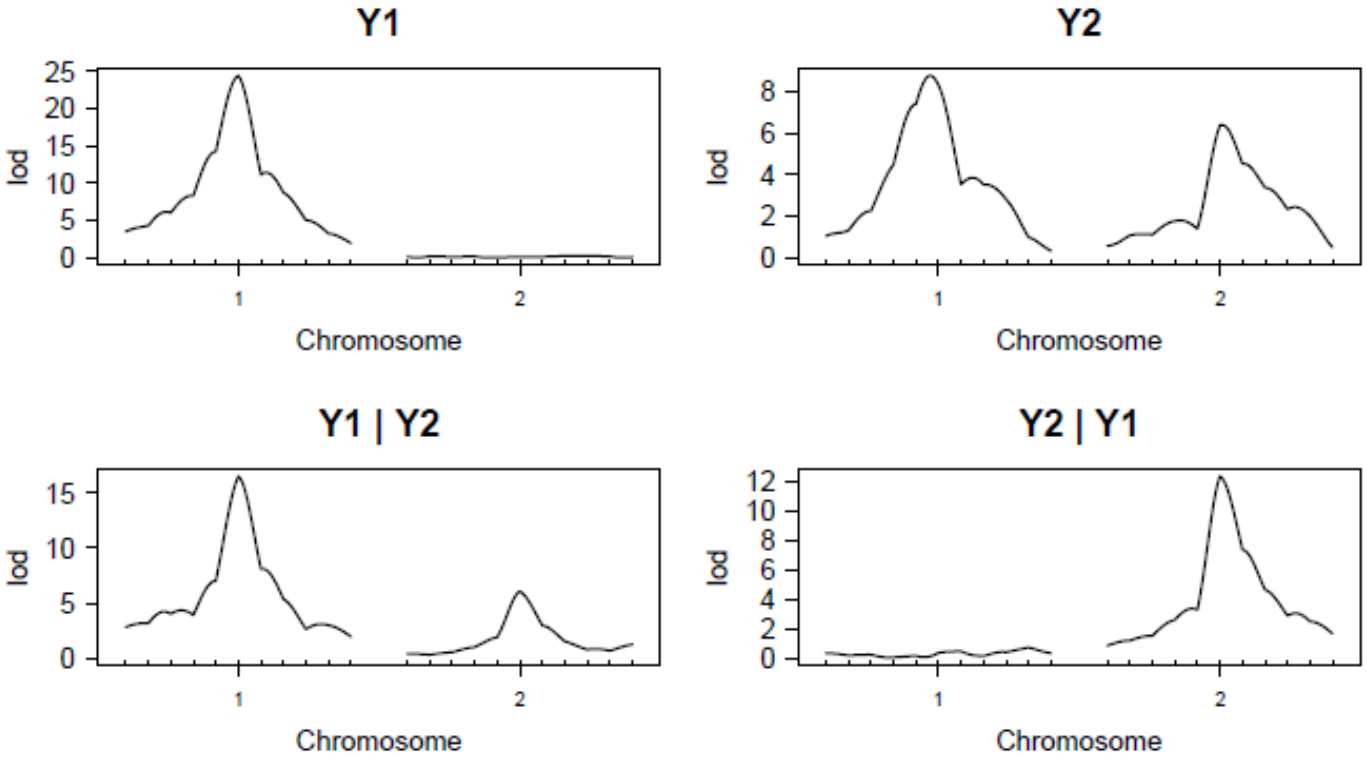
QTLnet algorithm

QTLnet algorithm

- Perform joint inference of the causal phenotype network and the associated genetic architecture.
- The genetic architecture is inferred conditional on the phenotype network.
- Because the phenotype network structure is itself unknown, the algorithm iterates between updating the network structure and genetic architecture using a Markov chain Monte Carlo (MCMC) approach.
- QTLnet corresponds to a mixed Bayesian network with continuous and discrete nodes representing phenotypes and QTLs, respectively.

QTL mapping conditional on the pheno net structure

We simulated data from the model $Q_1 \rightarrow Y_1 \rightarrow Y_2 \leftarrow Q_2$ with Q_1 located on chr 1, and Q_2 on chr 2.



- Y_2 maps indirectly to Q_1 (top right), but Y_1 d-separates Y_2 and Q_1 (bottom right).
- Y_1 is marginally independent from Q_2 (top left), but conditional on Y_2 became associated (bottom left).

QTLnet algorithm - MCMC steps

1. Propose a new phenotype network, $\mathcal{M}_{new}$, by adding, deleting or reversing (with parent orphaning) an edge.

2. Recompute the genetic architecture (only for the phenotypes y_t whose parent set, $pa(y_t)$, has changed).

3. Compute the marginal likelihood $p(\mathbf{y} \mid \mathbf{q}, \mathcal{M}_{new})$.

4. Accept or reject the new phenotype network and QTLs according to the Metropolis-Hastings acceptance probability:

$$\alpha = min\left\{1, \frac{p(\mathbf{y} \mid \mathbf{q}, \mathcal{M}_{new})\, p(\mathcal{M}_{new})}{p(\mathbf{y} \mid \mathbf{q}, \mathcal{M}_{old})\, p(\mathcal{M}_{old})} \frac{q(\mathcal{M}_{old} \mid \mathcal{M}_{new})}{q(\mathcal{M}_{new} \mid \mathcal{M}_{old})}\right\} .$$

QTLnet algorithm

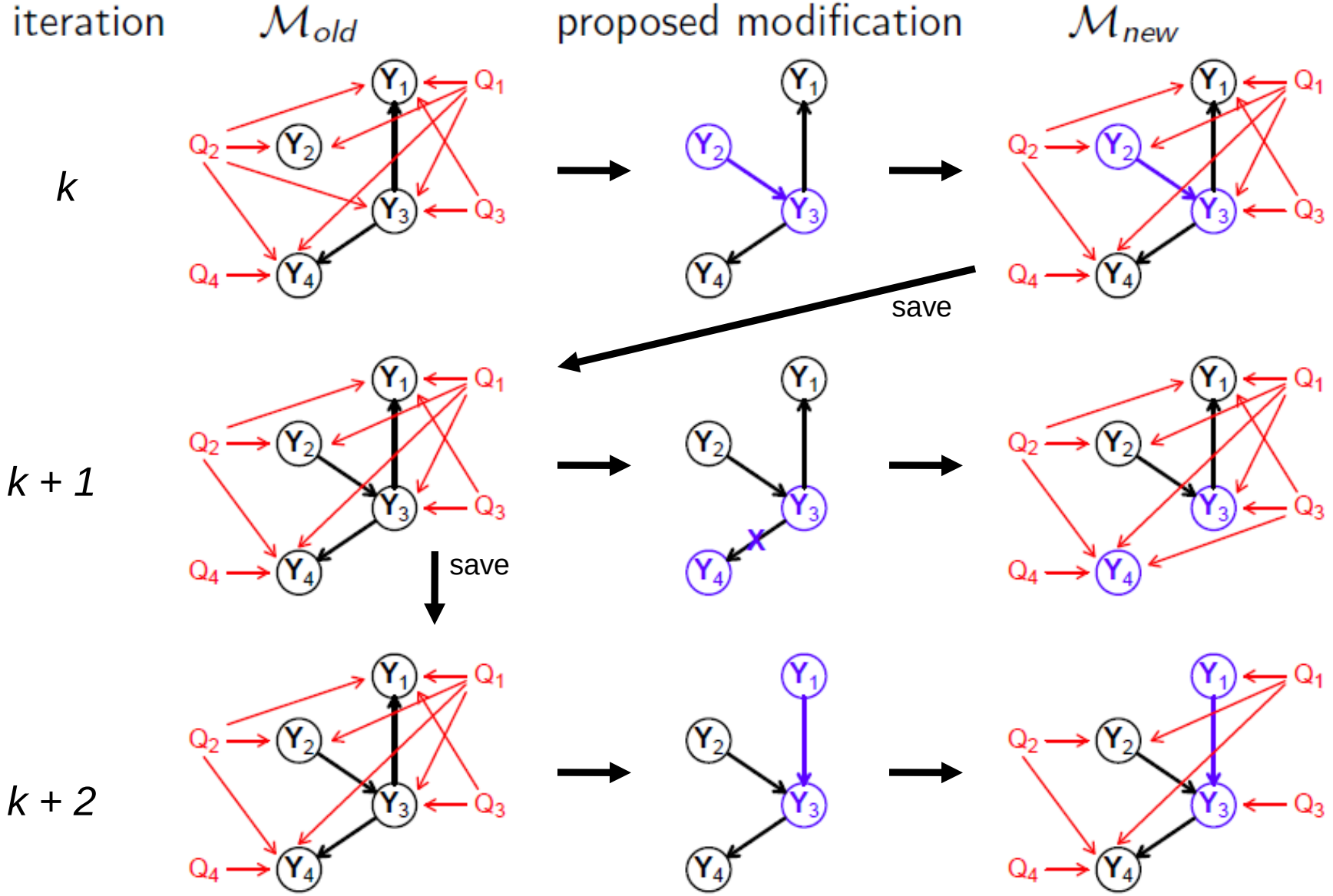
We approximate the Bayes factor comparing old and new models by

$$\frac{p(\mathbf{y} \mid \mathbf{q}, \mathcal{M}_{new})}{p(\mathbf{y} \mid \mathbf{q}, \mathcal{M}_{old})} \approx \exp\left\{-\frac{1}{2}(BIC_{\mathcal{M}_{new}} - BIC_{\mathcal{M}_{old}})\right\},$$

and adopt $p(\mathcal{M}_{new})/p(\mathcal{M}_{old}) = 1$. The proposal distribution ratio is computed as

$$\frac{q(\mathcal{M}_{old} \mid \mathcal{M}_{new})}{q(\mathcal{M}_{new} \mid \mathcal{M}_{old})} = \frac{\#\text{ of DAGs that can be reached from } \mathcal{M}_{old}}{\#\text{ of DAGs that can be reached from } \mathcal{M}_{new}}.$$

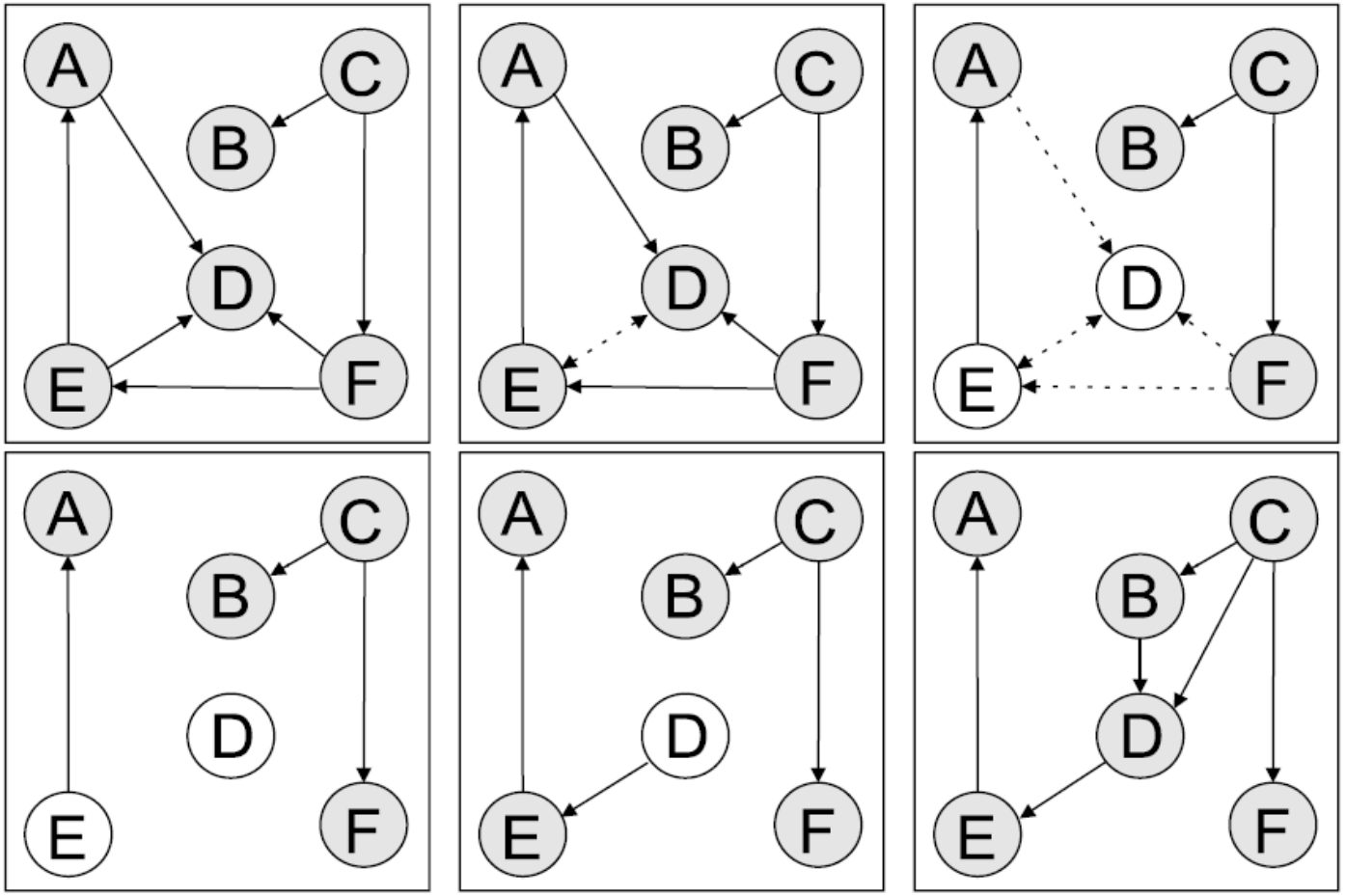
QTLnet algorithm



Neighborhood edge reversal

select edge
drop edge
identify parents

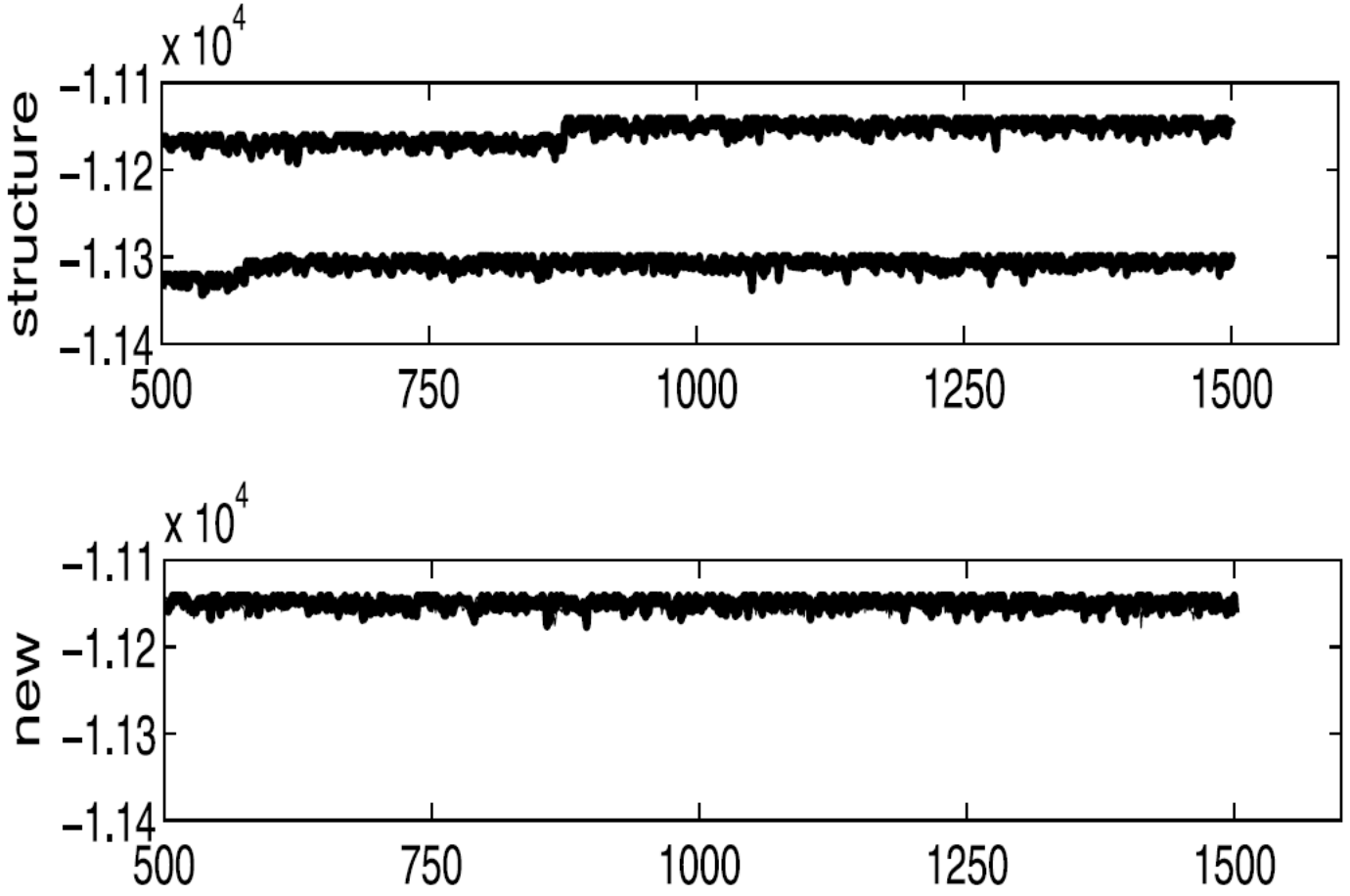
orphan nodes
reverse edge
find new parents



from Grzegorczyk and Husmier (2008)

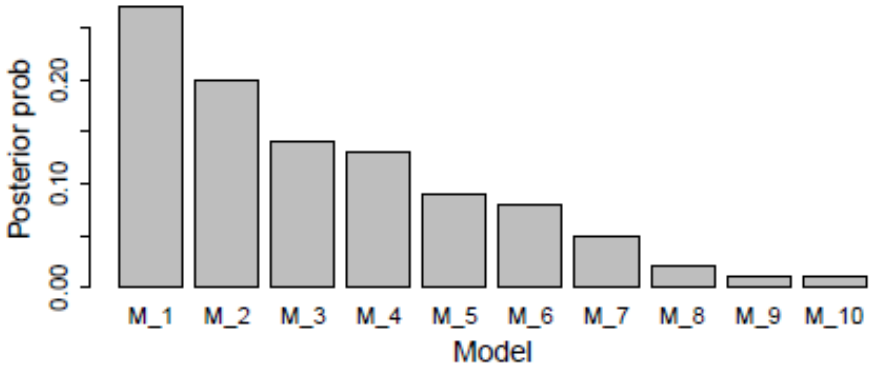
Neighborhood edge reversal

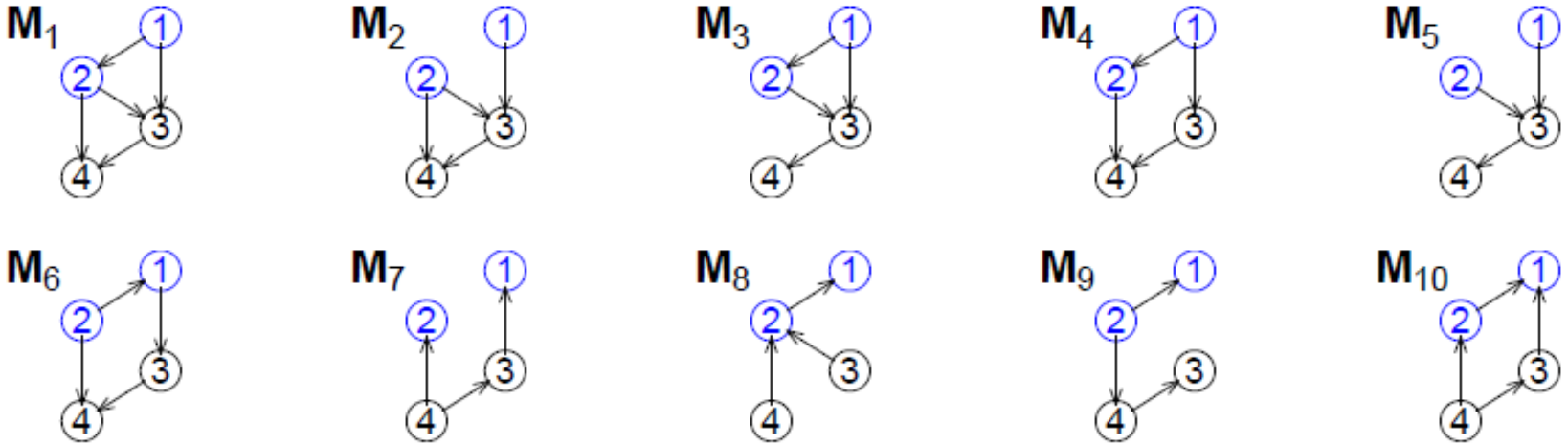
Trace plots of the logarithmic scores of the DAGs after the burn-in phase.



from Grzegorczyk and Husmier (2008)

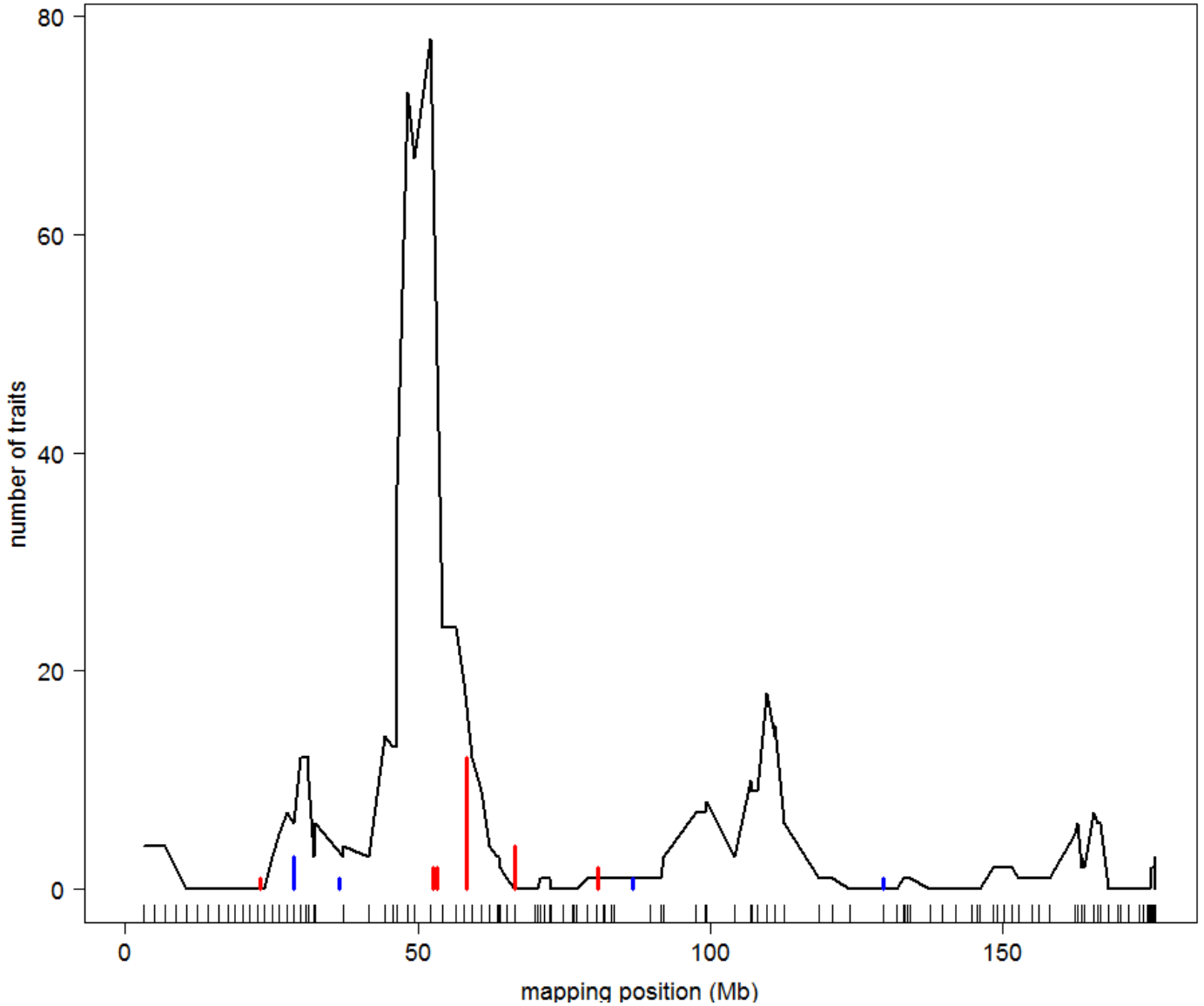
Bayesian model averaging



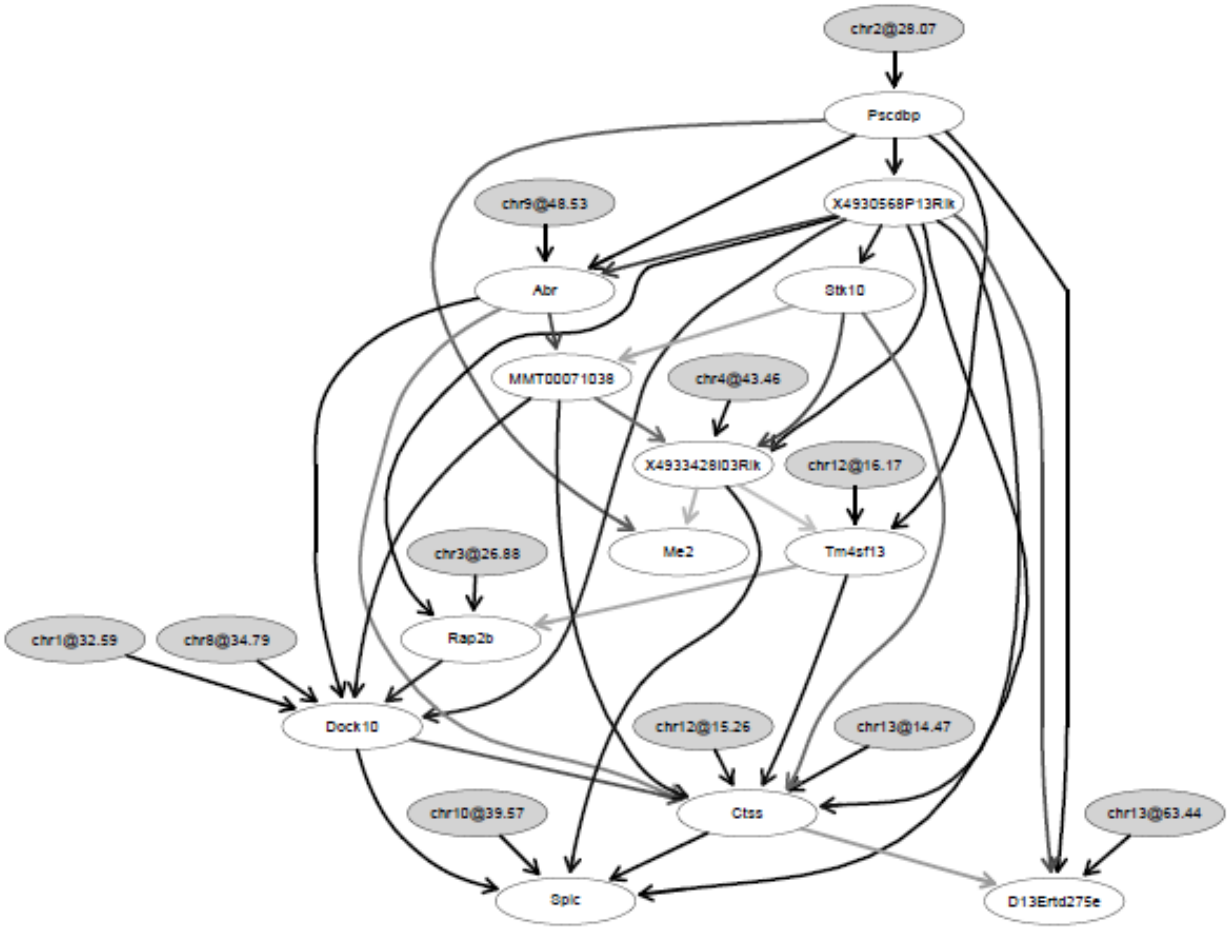


$$Pr(Y_1 \rightarrow Y_2) = Pr(M_1) + Pr(M_3) + Pr(M_4) = 0.54$$
$$Pr(Y_1 \ldots Y_2) = Pr(M_2) + Pr(M_5) + Pr(M_7) = 0.34$$
$$Pr(Y_1 \leftarrow Y_2) = Pr(M_6) + Pr(M_8) + Pr(M_9) + Pr(M_{10}) = 0.12$$

BxH ApoE-/- chr 2: causal architecture



BxH ApoE-/- chr 2: causal network for transcription factor Pscdbp



Scaling up to larger networks

- Reduce complexity of graphs
  - restrict number of causal edges into each node

BIC computations by maximum number of parents

#	3	4	5	6	all
10	1,300	2,560	3,820	4,660	5,120
20	23,200	100,720	333,280	875,920	10.5M
30	122,700	835,230	4.40M	18.6M	16.1B
40	396,800	3.69M	26.7M	157M	22.0T
50	982,500	11.6M	107M	806M	28.1Q

(limit complexity by allowing only 3-4 parents)

- make task parallel: run on many machines
  - pre-compute BIC scores
  - run multiple parallel Markov chains

Parallel phases for larger projects

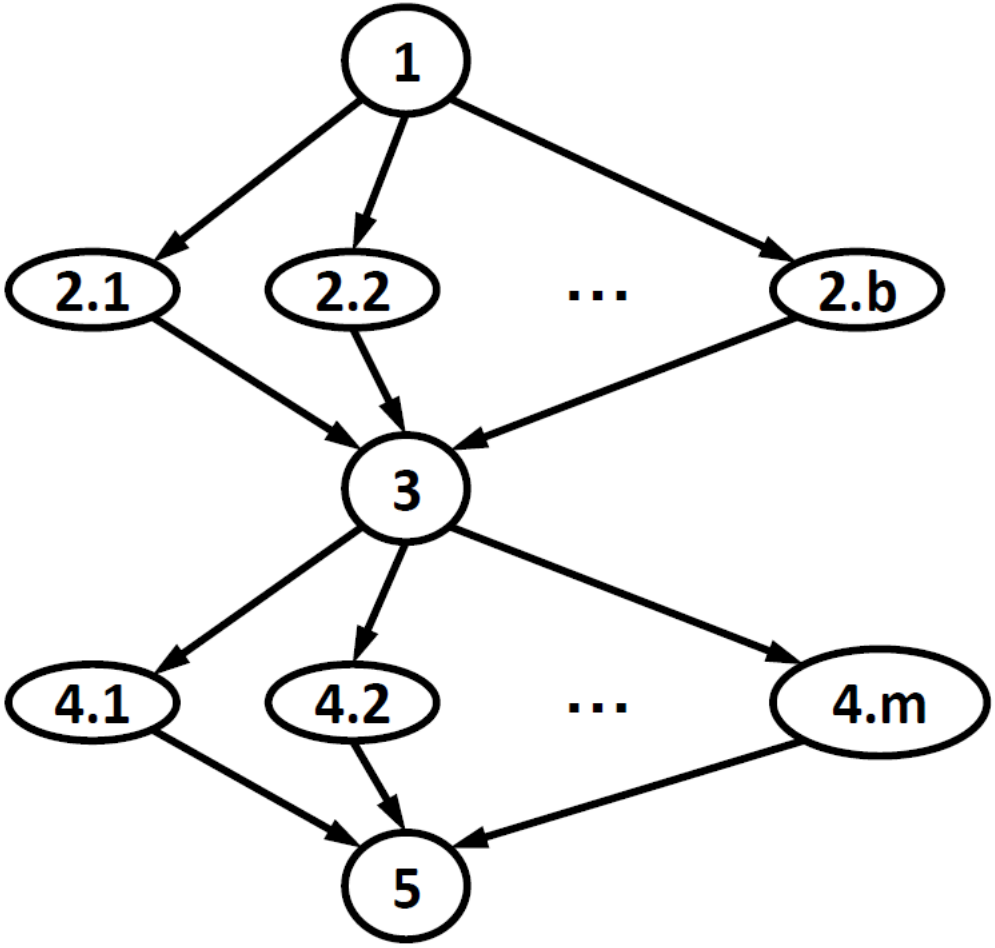
Phase 1: identify parents

Phase 2: compute BICs

Phase 3: store BICs

Phase 4: run Markov chains

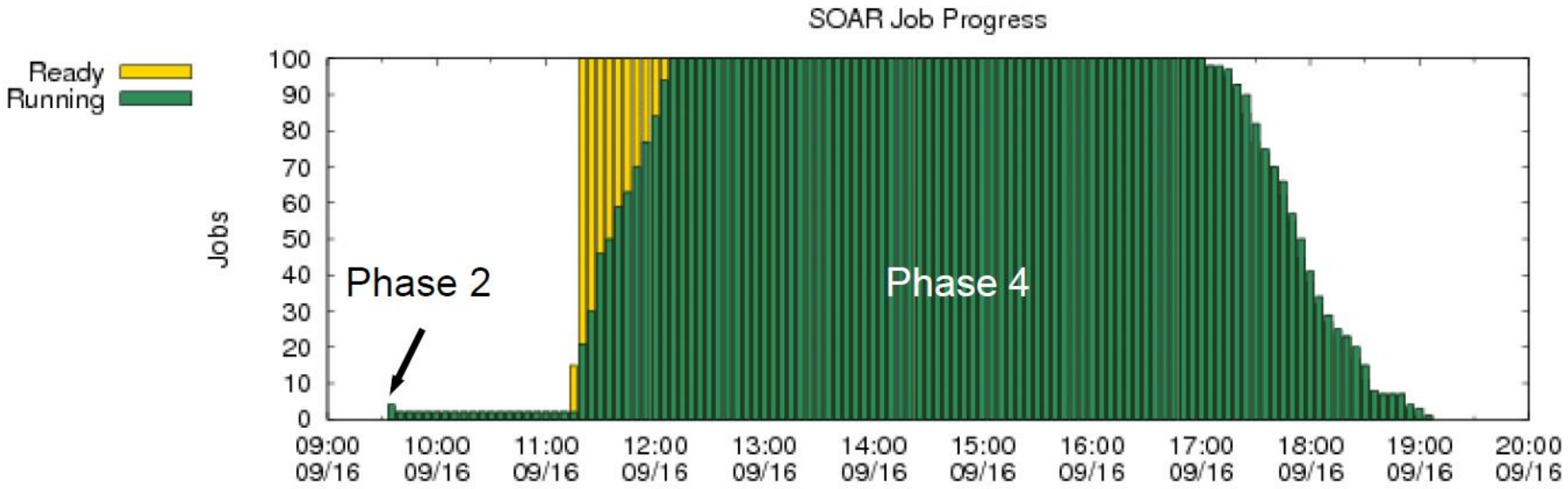
Phase 5: combine results



Parallel implementation

R/qtlnet available at CRAN

- Condor cluster: chtc.cs.wisc.edu
  - System Of Automated Runs (SOAR)
    - ~2000 cores in pool shared by many scientists
    - automated run of new jobs placed in project



Final remarks

Potential issues

- Steady state (static) measures may not reflect dynamic processes (Przytycha and Kim 2010).
- Population-based estimates (from a sample of individuals) may not reflect processes within an individual.

References

1. Chaibub Neto et al. (2008) *Genetics* **179**: 1089-1100.
2. Chaibub Neto et al. (2010) *Annals of Applied Statistics* **4**: 320-339.
3. Ferrara et al. (2008) *Plos Genetics* **4**: e1000034.
4. Grzegorczyk and Husmier (2008) *Machine Learning* **71**: 265-305.
5. Pearl (1988) *Probabilistic reasoning in intelligent systems* Morgan Kauffman.
6. Pearl (2000) *Causality: models, reasoning and inference* Cambridge U Press
7. Przytycha and Kim (2010) *BMC Biology* **8**: 48.
8. Spirtes et al. (2000) *Causation, prediction and search* MIT Press.
9. Wright (1921) *Journal of Agricultural Research* **20**: 557-585.
10. Verma and Pearl (1990) In *Readings in uncertain reasoning* Morgan Kauffmann
11. Zhu et al. (2007) *Plos Computational Biology* **3**: e69.