Dependency networks

Sushmita Roy

sroy@biostat.wisc.edu

Computational Network Biology

Biostatistics & Medical Informatics 826

https://compnetbiocourse.discovery.wisc.edu

Sep 27th 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)

Goals for today

- Dependency networks
- GENIE3
- Evaluation of expression-based network inference methods

Recall the different types of probabilistic graphs

- In each graph type we can assert different conditional independencies
- Correlation networks
- Markov networks
  - Gaussian Graphical models
- Dependency networks
- Bayesian networks

Dependency network

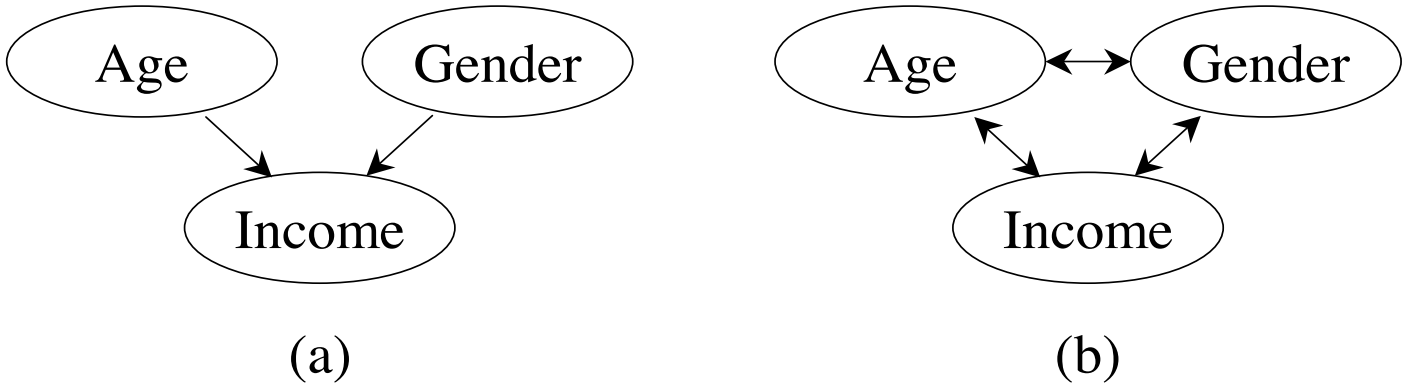
- A type of probabilistic graphical model
- Approximate Markov networks
  - Are much easier to learn from data
- As in Bayesian networks has
  - A graph structure
  - Parameters capturing dependencies between a variable and its parents
- Unlike Bayesian network
  - Can have cyclic dependencies
  - Computing a joint probability is harder
    - It is approximated with a “pseudo” likelihood.

Dependency Networks for Inference, Collaborative Filtering and Data visualization
Heckerman, Chickering, Meek, Rounthwaite, Kadie 2000

Original motivation of dependency networks

- Introduced by Heckerman, Chickering, Meek, et al 2000
- Often times Bayesian networks can get confusing
  - Bayesian networks learned represent correlation or predictive relationships
  - But the directionality of the edges are mistakenly interpreted as causal connections
- (Consistent) Dependency networks were introduced to distinguish between these cases

Dependency network vs Bayesian network



Bayesian network

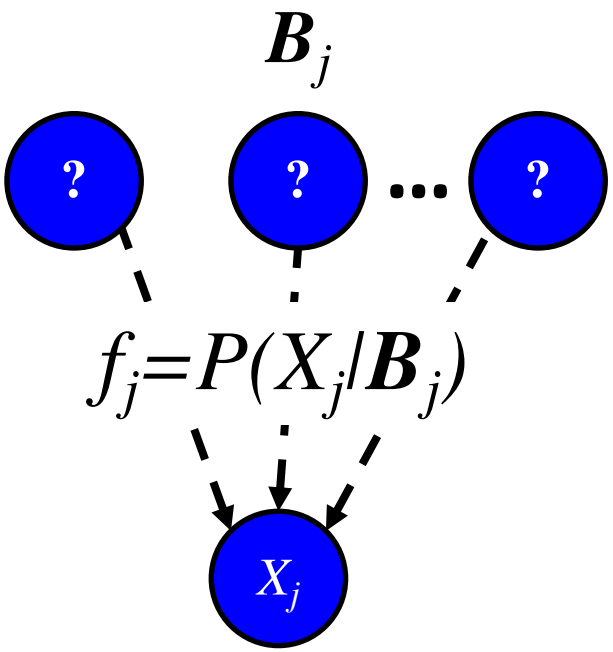
Dependency network

Often times, the Bayesian network on the left is read as if “Age” determines “Income”. However, all this model is capturing is that “Age” is predictive of “Income”.

Dependency Networks for Inference, Collaborative Filtering and Data visualization
Heckerman, Chickering, Meek, Rounthwaite, Kadie 2000

Learning dependency networks

- Entails estimating the Markov blanket of each random variable



- Let $\boldsymbol{B}_j$ denote the Markov Blanket of a variable X_j.
- $\boldsymbol{X}_{-j}$ denotes all variables other than X_j
- Given $\boldsymbol{B}_j$, X_j is independent of all other variables, $\boldsymbol{X}_{-j}$

$$P(X_j|\mathbf{X}_{-j}) = P(X_j|\mathbf{B}_j)$$

- $\boldsymbol{B}_j$ can be estimated by finding the set of variables that best predict X_j
- This requires us to specify the form of $P(X_j|\boldsymbol{B}_j)$

Different representations of $f_j = P(X_j | B_j)$

- If X_j is continuous
  - f_j can be a linear function
  - f_j can be a regression tree
  - f_j can be an ensemble of trees
    - E.g. random forests
- If X_j is discrete
  - f_j can be a conditional probability table
  - f_j can be a conditional probability tree

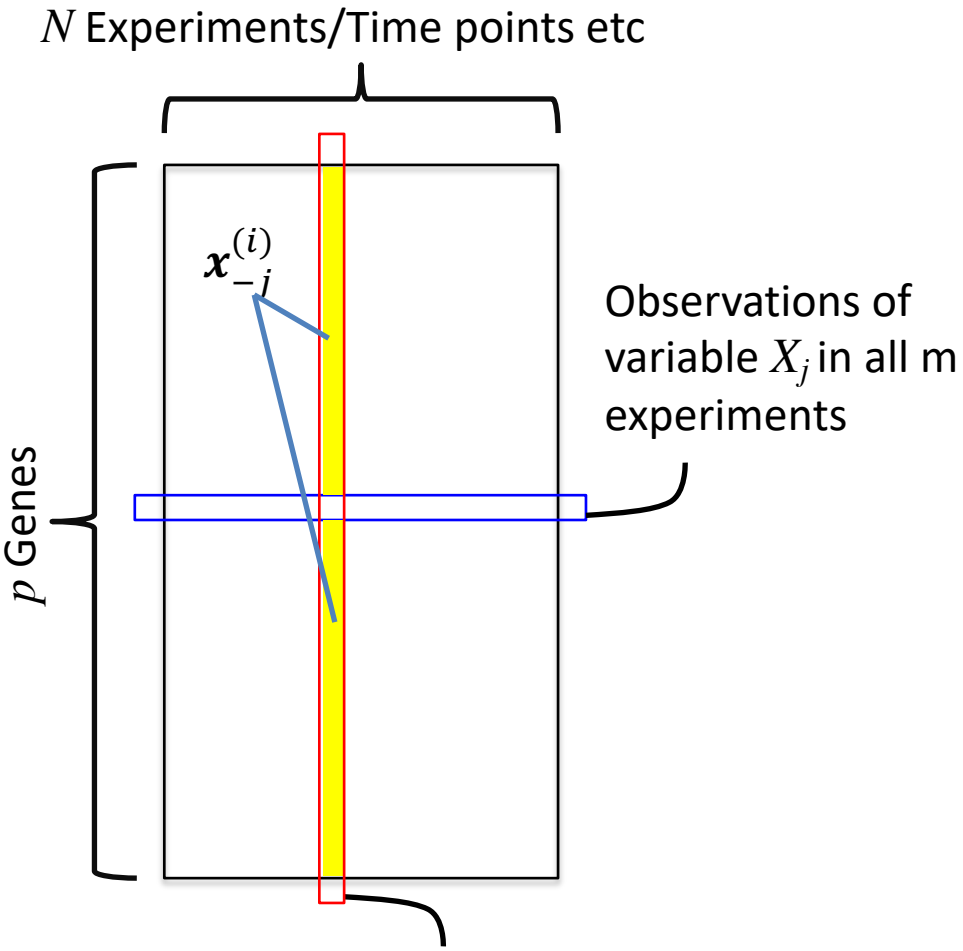
Popular dependency networks implementations

- Learned by solving a set of linear regression problems
    - TIGRESS (Haury et al, 2010)
        - Uses a constraint to learn a “sparse” Markov blanket
        - Uses “stability selection” to estimate confidence of edges
- Learned by solving a set of non-linear regression problems
    - Non-linearity captured by Regression Tree (Heckerman et al, 2000)
    - GENIE3: Non-linearity captured by Random forest (Huynh-Thu et al, 2010)
    - Inferelator (Bonneau et al, Genome Biology 2005)
        - Can handle time course and single time point data
        - Non-linear regression is done using a logistic transform
        - Handles linear and non-linear regression

Goals for today

- Dependency networks
- GENIE3
- Evaluation of expression-based network inference methods

Expression data matrix



GENIE3: GEne Network Inference with Ensemble of trees

- Solves a set of regression problems
  - One per random variable
  - Minimizes the prediction error per variable X_j

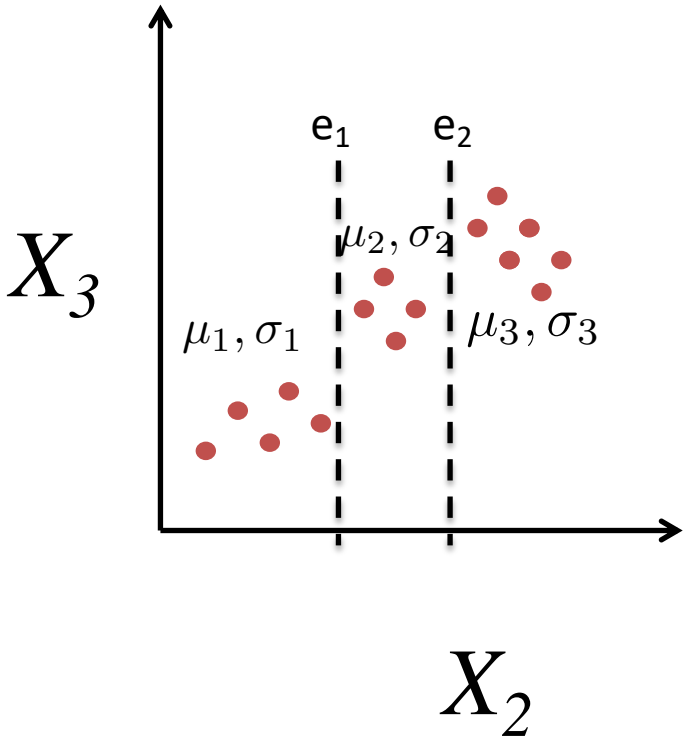
$$\sum_{i=1}^{N} \left(x_j^i - f_j\left(\mathbf{x}_{-j}^i\right) \right)^2$$

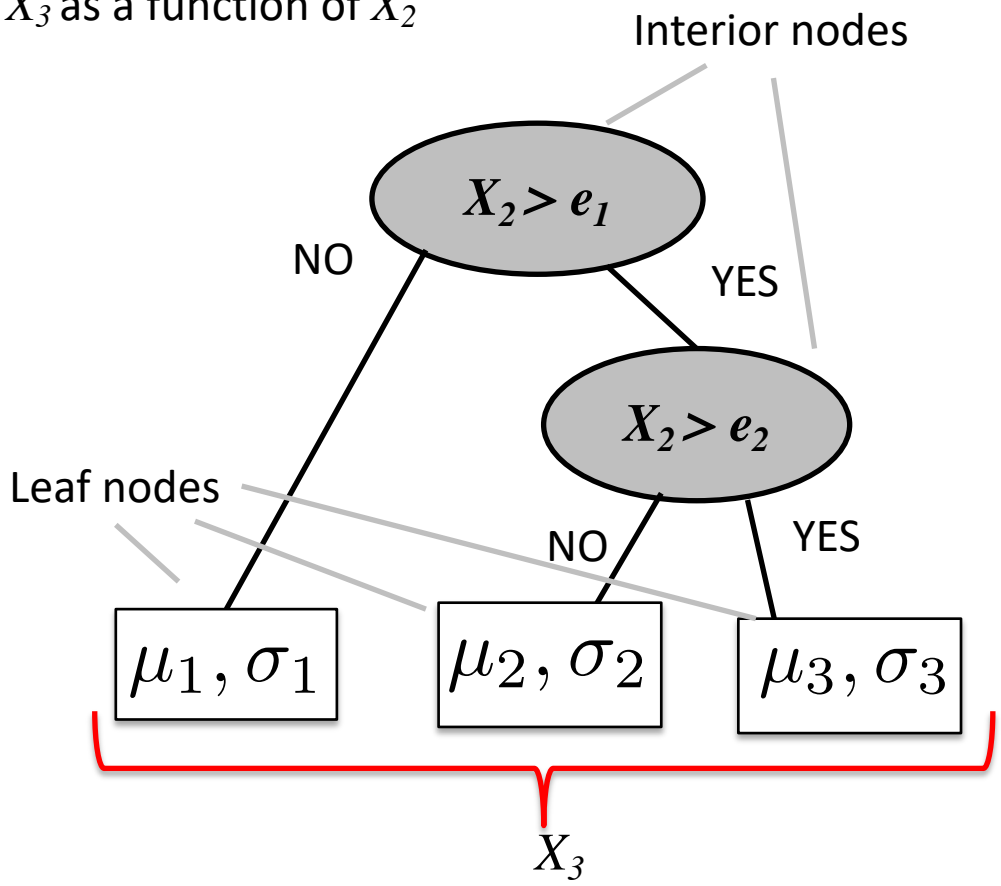
- Uses an Ensemble of regression trees to represent f_j
  - Models non-linear dependencies
- Outputs a directed cyclic graph with a confidence of each edge
  - Directionality means "good predictor"
- Focus on generating a ranking over edges rather than a graph structure and parameters
  - Rank is determined by confidence

Inferring Regulatory Networks from Expression Data Using Tree-Based Methods Van Anh Huynh-Thu, Alexandre Irrthum, Louis Wehenkel, Pierre Geurts, Plos One 2010

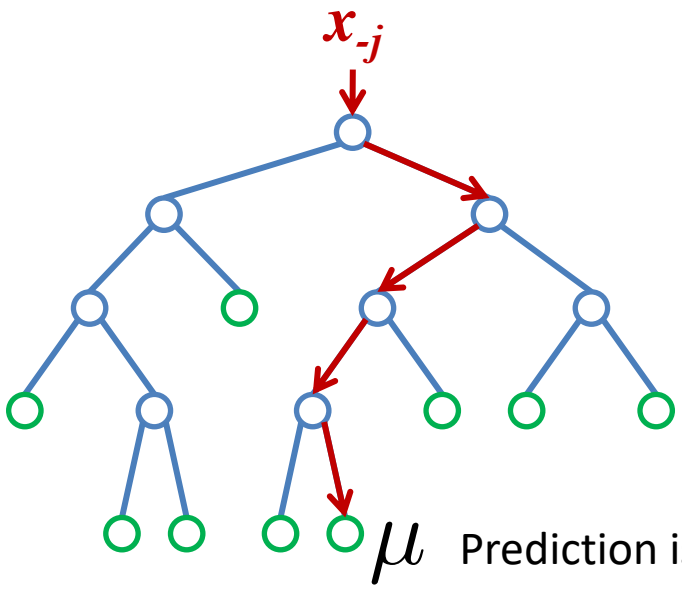
Recall our very simple regression tree for two variables

The tree specifies X_3 as a function of X_2





Prediction of X_j using a single tree

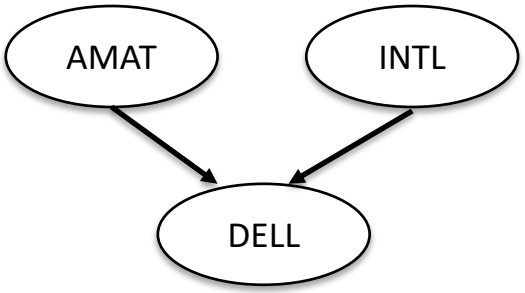


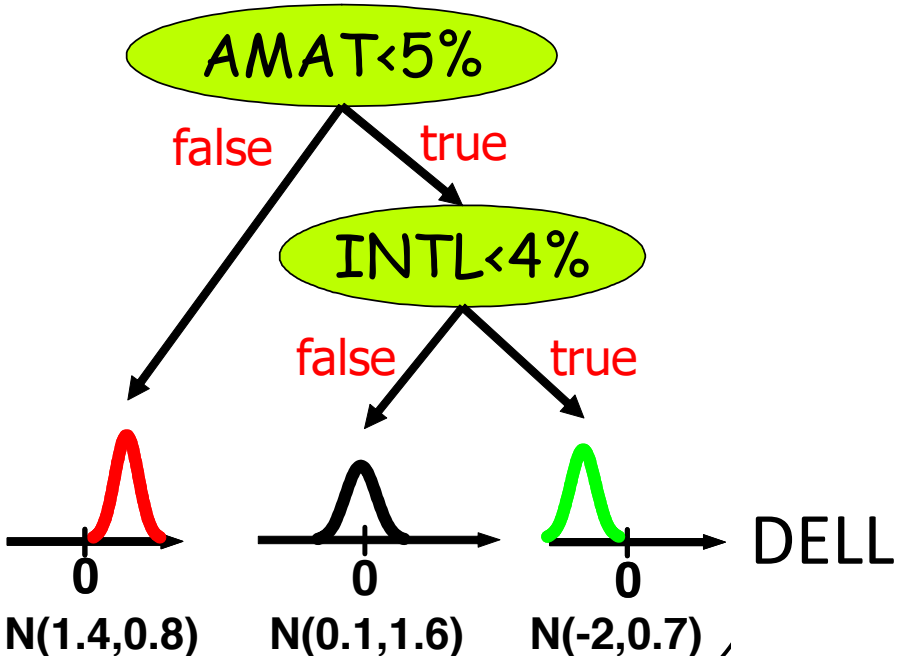
leaf nodes
split nodes

Prediction is the mean at each leaf node

Taken from ICCV09 tutorial by Kim, Shotton and Stenger: http://www.iis.ee.ic.ac.uk/~tkkim/iccv09_tutorial

Prediction example with a regression tree





Suppose we observe AMAT=10% and INTL=7%

What is DELL's predicted value? 1.4

Quantifying a split on the tree

- Let $\boldsymbol{X}_{-j}$ denote the set of candidate variables we can split on
- A split is defined by a tuple, (X_i, s), s is the test value of $X_i, X_i \in \boldsymbol{X}_{-j}$
- The best split of a leaf node is found by enumerating over all possible splits defined by the predictor variables and split values s:

$$min_{i,s}\left(\sum_{k \in S_{left}} (x_j^k - \mu^{S_{left}})^2 + \sum_{l \in S_{right}} (x_j^l - \mu^{S_{right}})^2\right)$$

The set of samples in the left node

The set of samples in the right node

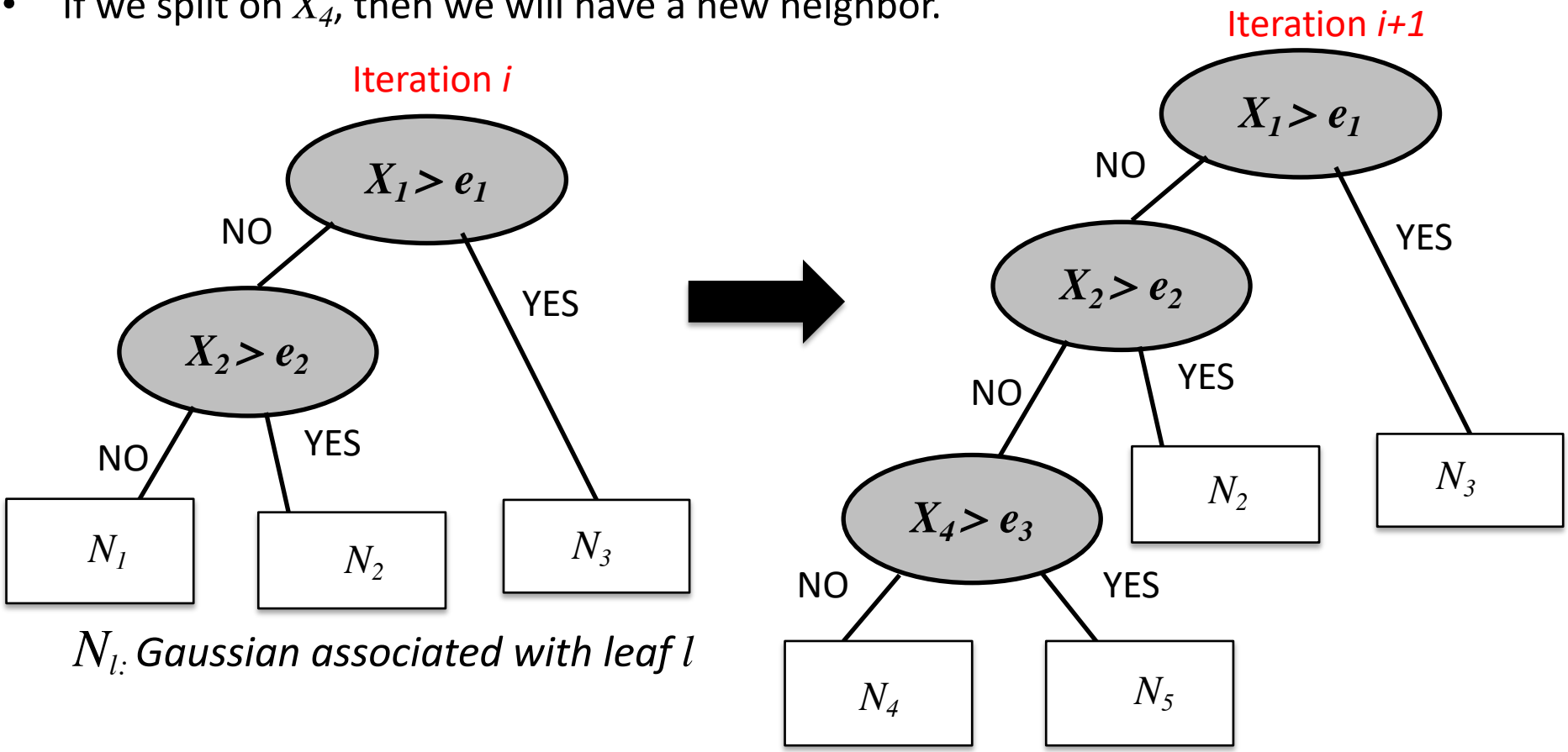
S_{left} and S_{right} are sets of samples obtained by testing X_i for a particular split s

Algorithm for learning a regression tree

- **Input:** dataset D, variable X_j, candidate predictors $\boldsymbol{X}_{-j}$ of X_j
- **Output**: Tree T
- Initialize T to a leaf node, μ, σ estimated from all samples of X_j. Assign all samples to leaf node
- While not converged
  - For every leaf node l in T
    - Find the best split, (X_i, s) at l
    - If the split improves prediction power or convergence criteria are not met
      - add two leaf nodes, l_{left} and l_{right} to l
      - Assign sample $\boldsymbol{x}^{(m)}$ to l_{left} if $x_i^m < s$, and to l_{right} otherwise
      - Update parameters associated with l_{left} and l_{right}

One iteration of regression tree learning

- Let $\mathbf{X}=\{X_1,X_2,X_3,X_4\}$
- Assume we are searching for the neighbors of X_3 and it already has two neighbors X_1 and X_2
- X_1, X_2, X_4 will all be considered as candidate splits using the examples at each current leaf node
- If we split on X_4, then we will have a new neighbor.



N_l: *Gaussian associated with leaf* l

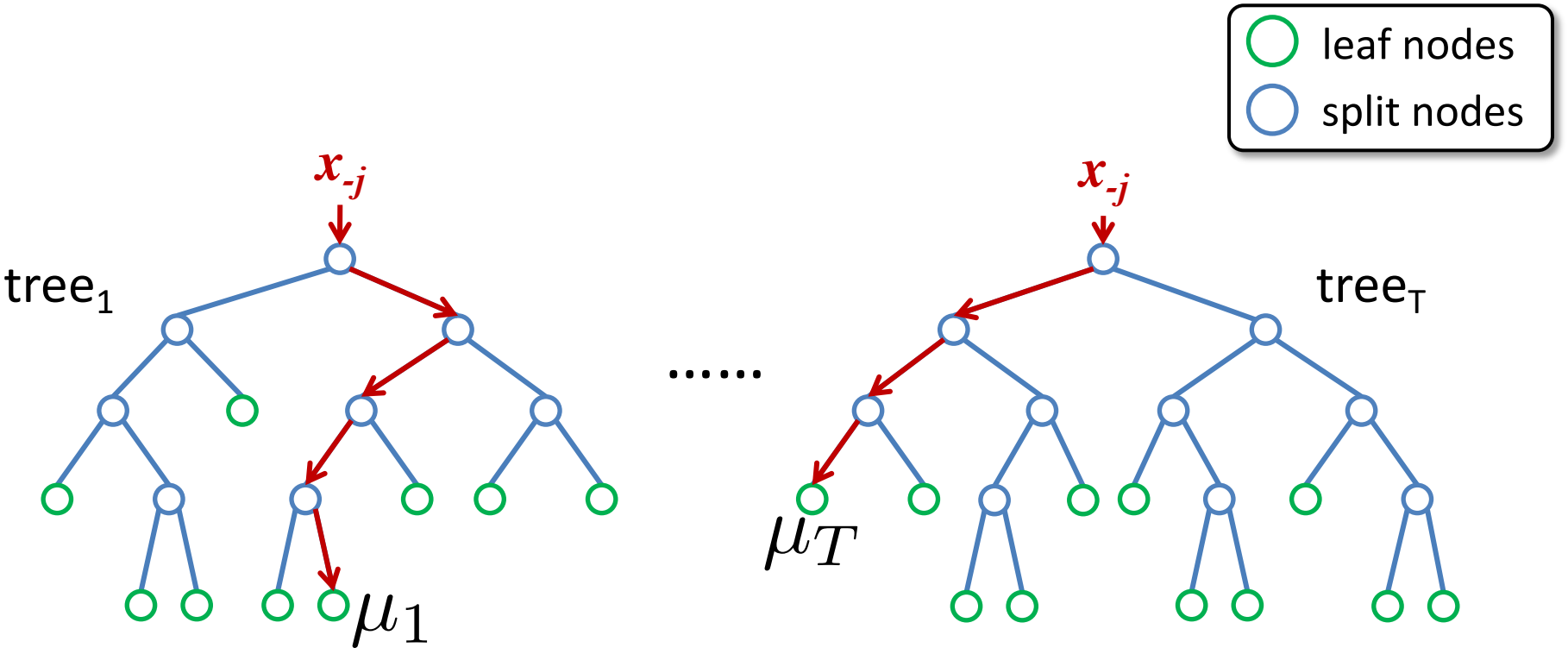
Convergence criteria

- Minimum number of examples at a leaf node
- Depth of a tree
- Error tolerance

An Ensemble of trees

- A single tree is prone to “overfitting”
- Instead of learning a single tree, ensemble models make use of a collection of trees

Prediction using an Ensemble of Trees

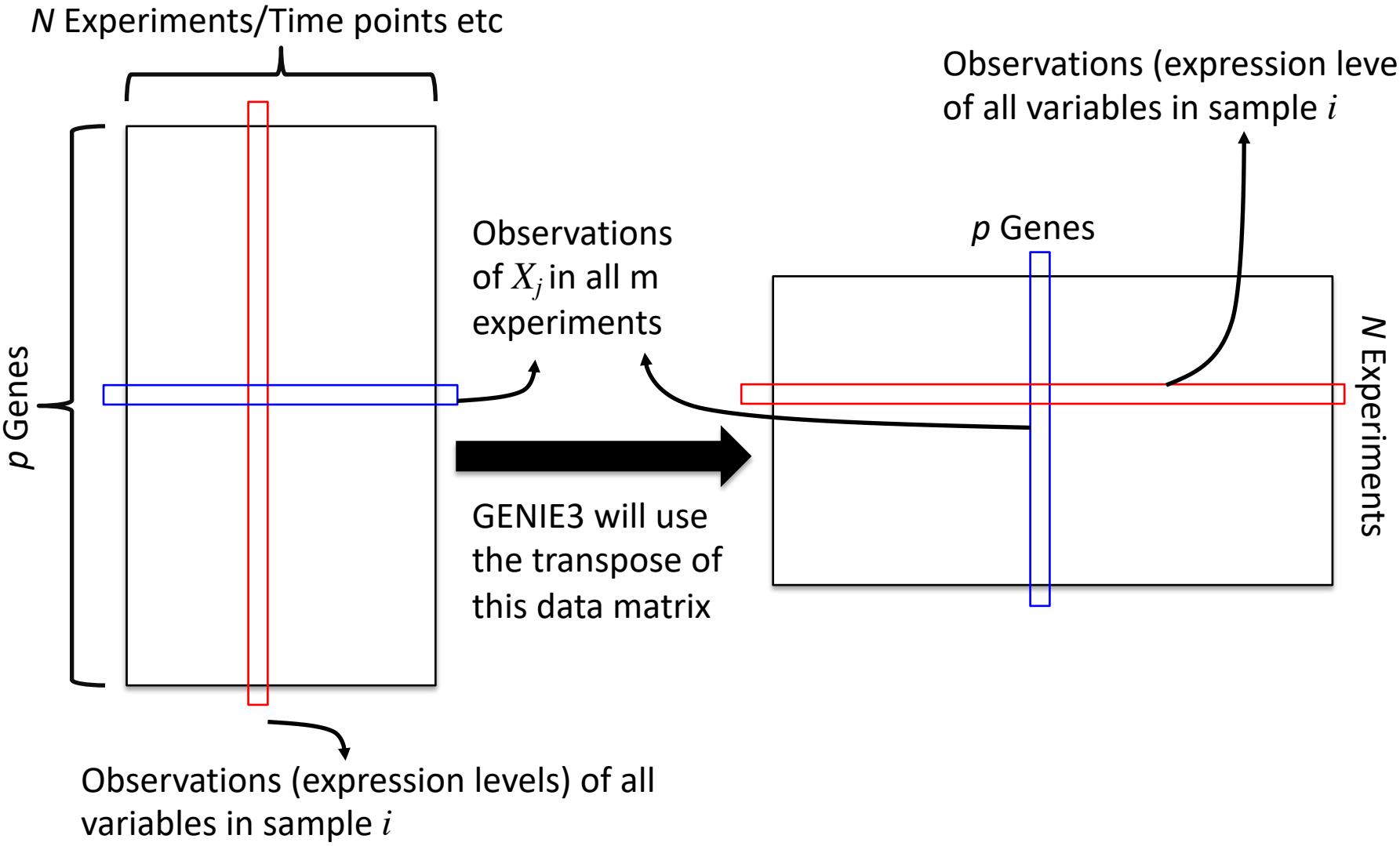


– Prediction is

$$\widehat{x_j} = \frac{1}{T}\sum_{t=1}^{T}\mu_t$$

Taken from ICCV09 tutorial by Kim, Shotton and Stenger: http://www.iis.ee.ic.ac.uk/~tkkim/iccv09_tutorial

Expression data matrix



GENIE3 algorithm sketch

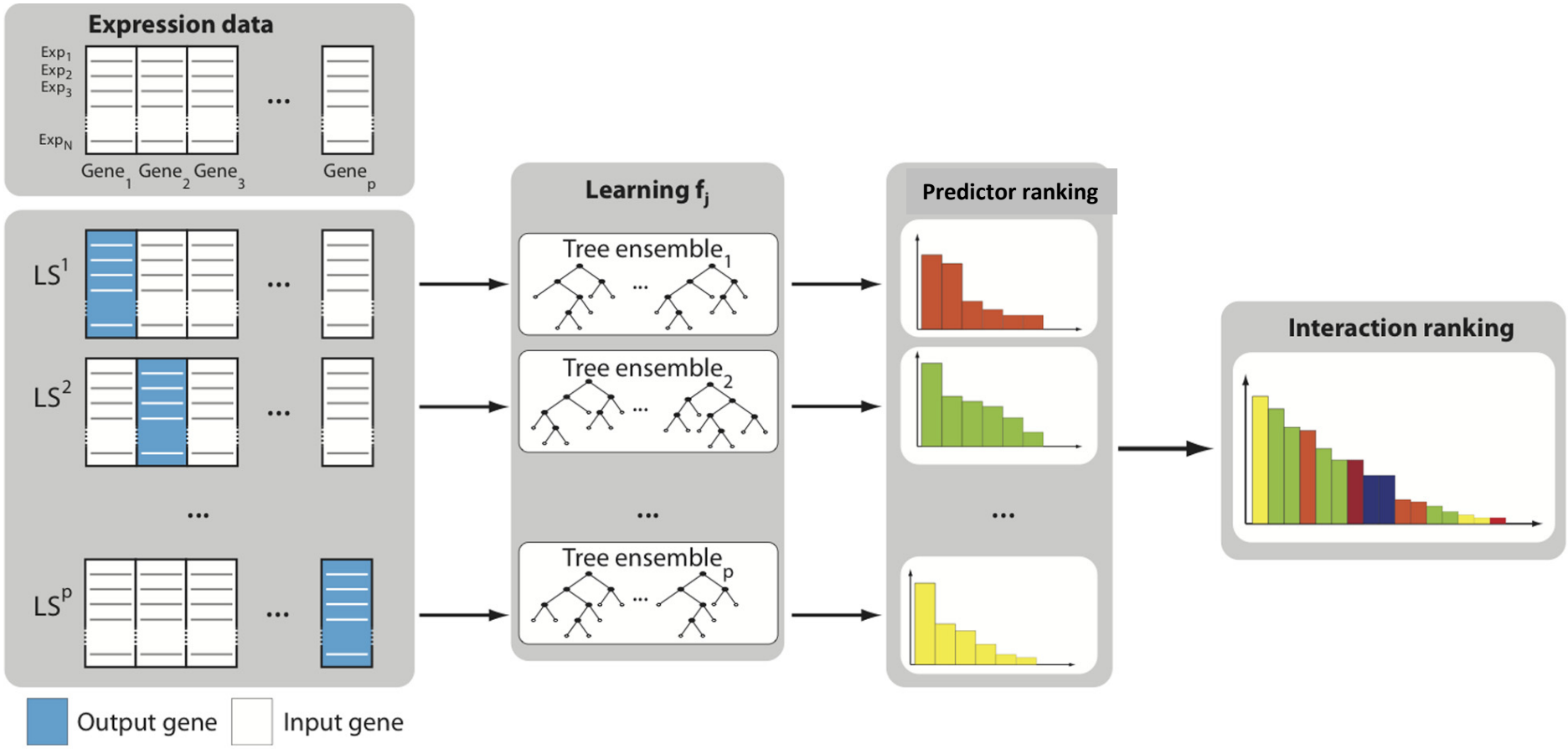


Figure from Huynh-Thu et al.

GENIE3 algorithm sketch

- For each X_j, generate learning samples of input/output pairs
  - $LS_j = \{(\boldsymbol{x}_{-j}^k, x_j^k),\ k = 1..N\}$
  - On each LS_j learn f_j to predict the value of X_j
  - f_j is an ensemble of regression trees
  - Estimate w_{ij} for all genes $i \neq j$
    - w_{ij} quantifies the confidence of the edge between X_i and X_j
    - Associated with the decrease in variance of X_j when X_i is included in f_j
- Generate a global ranking of edges based on each w_{ij}

Learning f_j in GENIE3

- Uses two types of Ensembles to represent the f_j:
  - Random forest or Extra Trees
- Learning the Random forest
  - Generate M=1000 bootstrap samples
  - At each node to be split, search for best split among K randomly selected variables
  - K was set to p-1 or $(p\text{-}1)^{1/2,}$ where p is the number of regulators/parents
- Learning the Extra-Trees
  - Learn 1000 trees
  - Each tree is built from the original learning sample
  - At each node, the best split is determined among K random splits, each split determined by randomly selecting one input (without replacement) and a threshold

Computing the importance weight of a predictor

- Importance is computed at each interior node
- Remember each predictor can show up multiple times as interior nodes
- For an interior node, importance is given by the reduction in variance when splitting on that node

$$I(\mathcal{N}) = \#SVar(S) - \#S_tVar(S_t) - \#S_fVar(S_f)$$

Interior node

Set of data samples that reach this node

$\#S$: Size of the set S

Var(S): variance of the output variable x_j in set S

S_t: subset of S when a test at $\mathcal{N}$ is true

S_f: subset of S when a test at $\mathcal{N}$ is false

$$Var(S) = \frac{1}{\#S}\sum_{i=1}^{\#S}(\mu_j^S - x_j^i)^2$$

Computing the importance weight of a predictor

- For a single tree the overall importance is then sum over all points in the tree where this node is used to split
- For an ensemble the importance is averaged over all trees
- To avoid bias towards highly variable genes, normalize the expression genes to all have unit variance

Goals for today

- Dependency networks
- GENIE3
- Evaluation of expression-based network inference methods

Evaluating the network

- Assessing confidence
- Area under the precision recall curve
- Do modules or target sets of genes participate in coherent function?
- Can the network predict expression in a new condition?

Assessing confidence in the learned network

- Typically the number of training samples is not sufficient to reliably determine the "right" network
- One can however estimate the confidence of specific features of the network
  - Graph features $f(G)$
- Examples of $f(G)$
  - An edge between two random variables
  - Order relations: Is X, Y's ancestor?

How to assess confidence in graph features?

- What we want is $P(f(G)|D)$, which is

$$\Sigma_G f(G) P(G|D)$$

- But it is not feasible to compute this sum

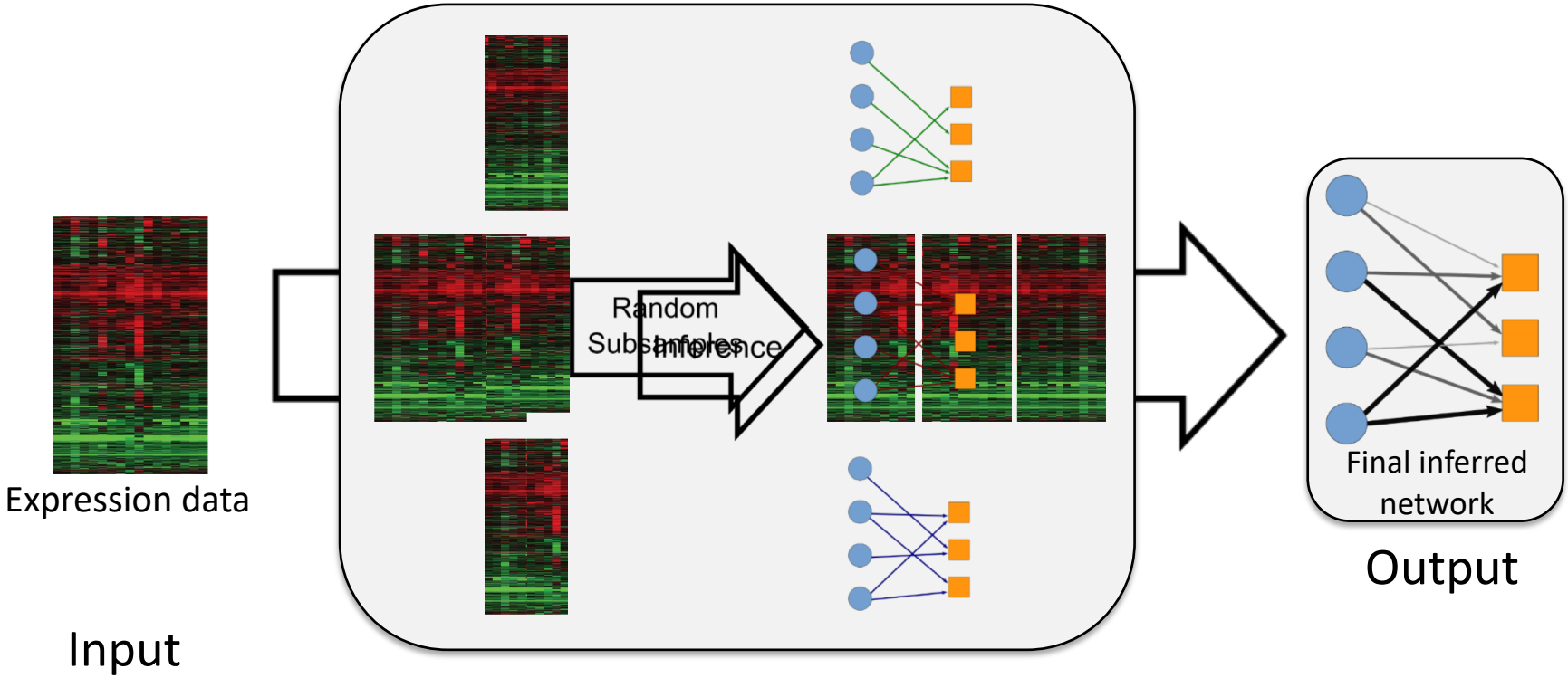
- Instead we will use a “bootstrap” procedure

Bootstrap to assess graph feature confidence

- For $i=1$ to m
  - Construct dataset $\mathbf{D}_i$ by sampling with replacement N samples from dataset $\mathbf{D}$, where N is the size of the original $\mathbf{D}$
  - Learn a graphical model $\{G_i, \Theta_i\}$
- For each feature of interest f, calculate confidence

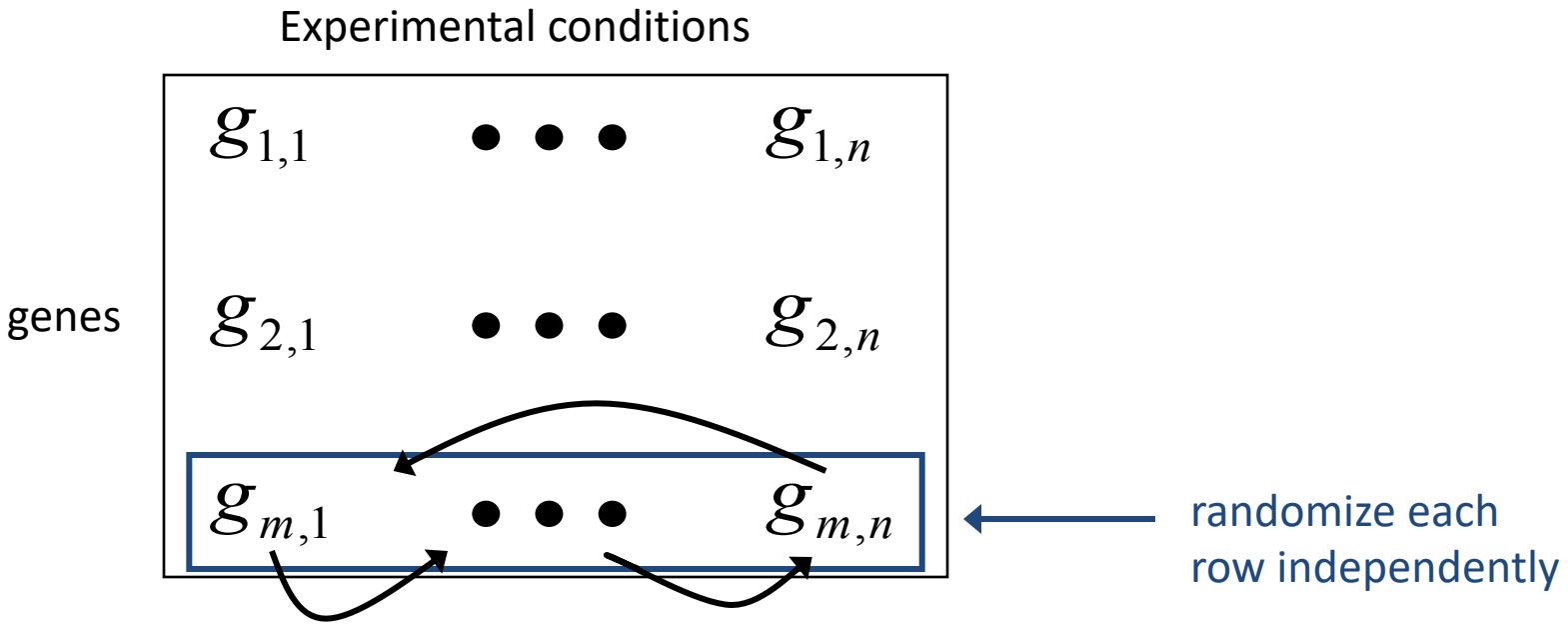
$$\mathrm{Conf}(f) = \frac{1}{m} \sum_{i=1}^{m} f(G_i)$$

Bootstrap/stability selection



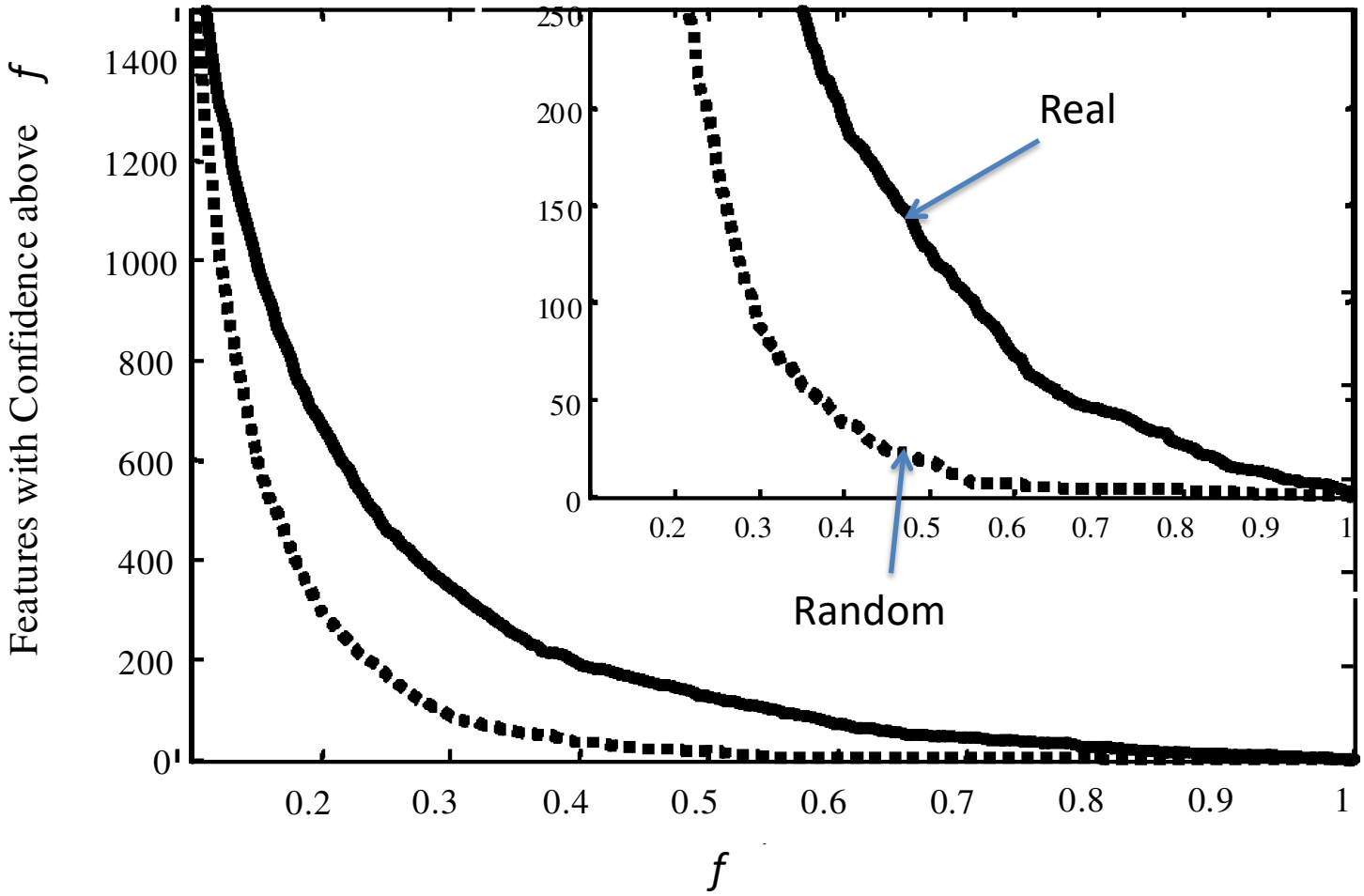
Does the bootstrap confidence represent real relationships?

- Compare the confidence distribution to that obtained from randomized data
- Shuffle the columns of each row (gene) separately
- Repeat the bootstrap procedure



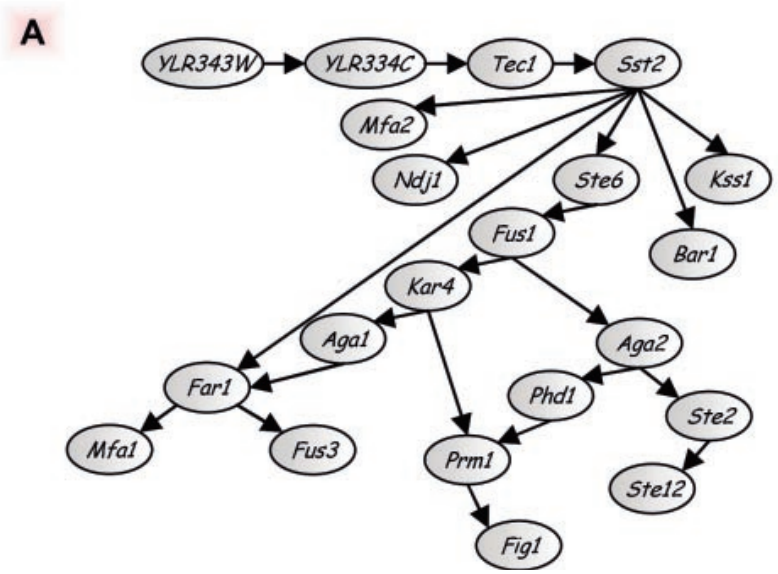
Slide credit Prof. Mark Craven

Bootstrap-based confidence differs between real and actual data

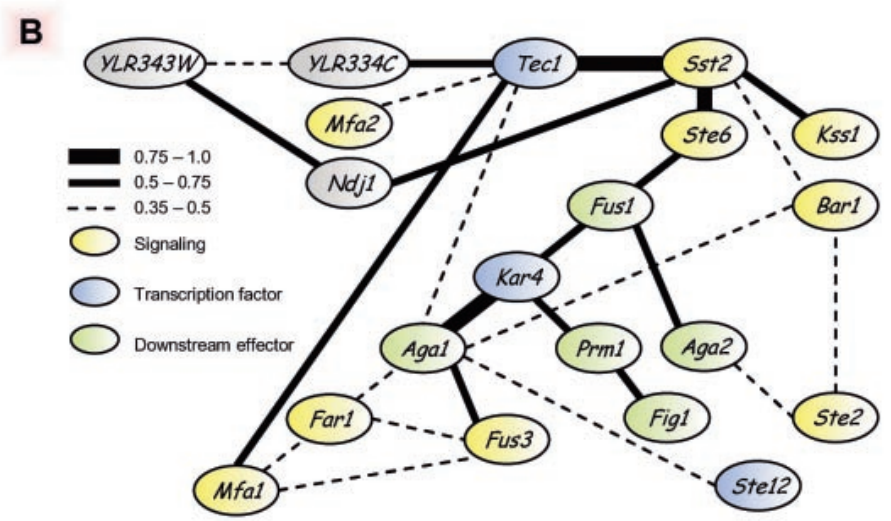


Friedman et al 2000

Example of a high confidence sub-network



One learned Bayesian network



Bootstrapped confidence Bayesian network: highlights a subnetwork associated with yeast mating pathway. Colors indicate genes with known functions.

Nir Friedman, Science 2004

Area under the precision recall curve (AUPR)

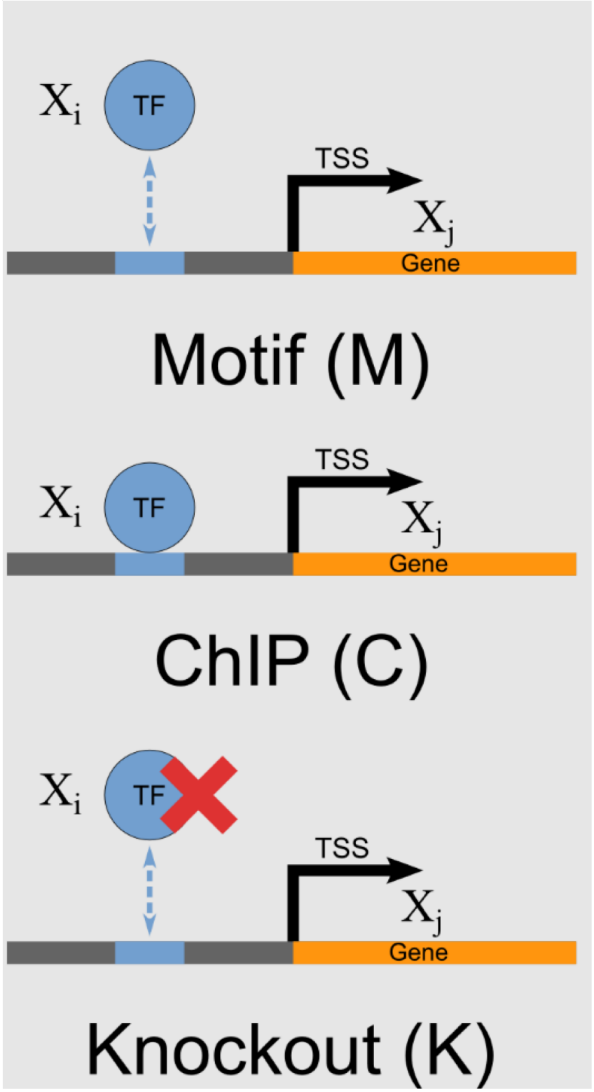
- Assume we know what the “right” network is
- One can use Precision-Recall curves to evaluate the predicted network
- Area under the PR curve (AUPR) curve quantifies performance

$$\text{Precision} = \frac{\#\text{ of correct edges}}{\#\text{ of predicted edges}}$$

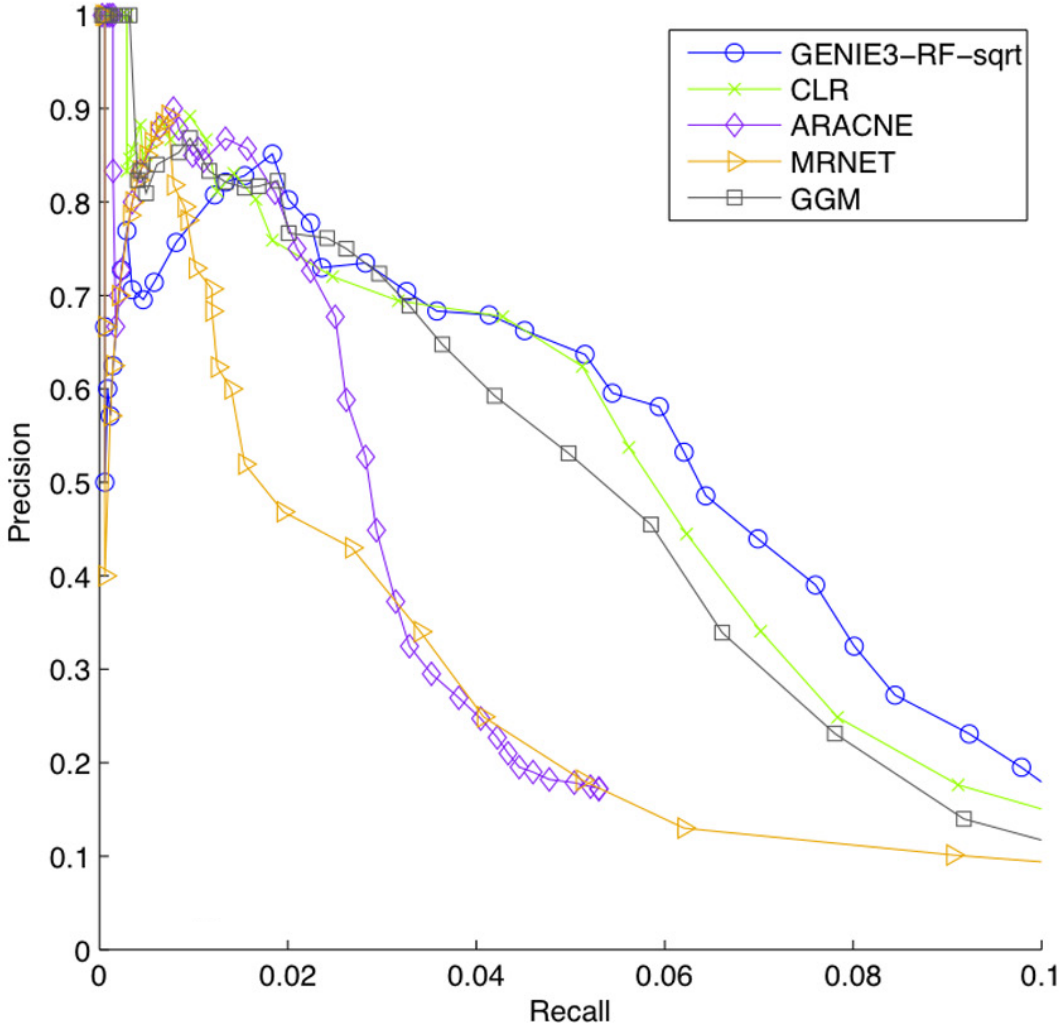
$$\text{Recall} = \frac{\#\text{ of correct edges}}{\#\text{ of true edges}}$$

Experimental datasets to assess network structure for gene regulatory networks

- Sequence specific motifs
- ChIP-chip and ChIP-seq
- Factor knockout followed by whole-transcriptome profiling

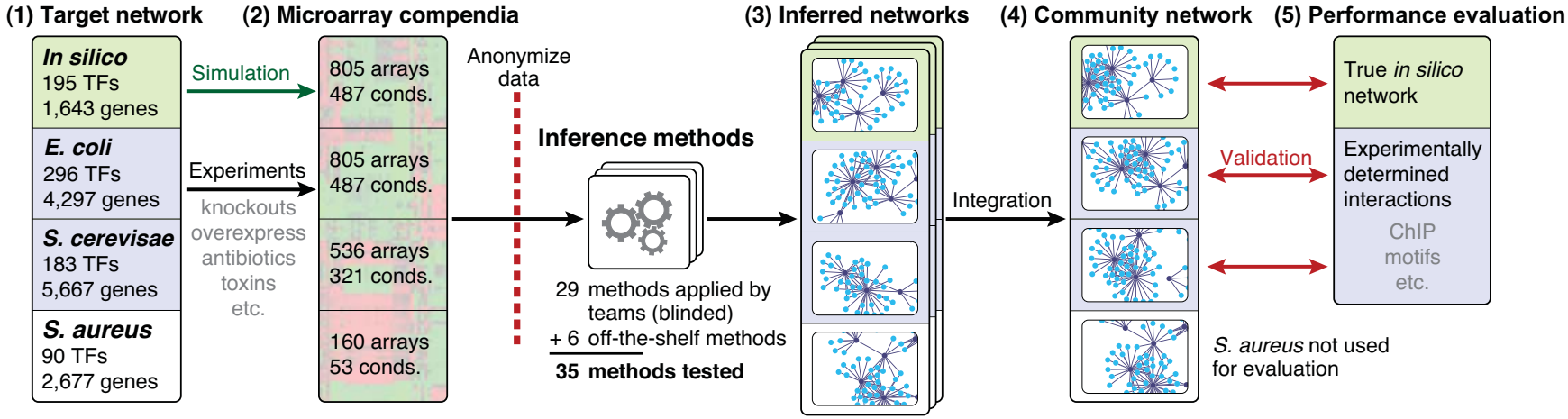


AUPR based performance comparison



DREAM: Dialogue for reverse engineeting assessments and methods

Community effort to assess regulatory network inference

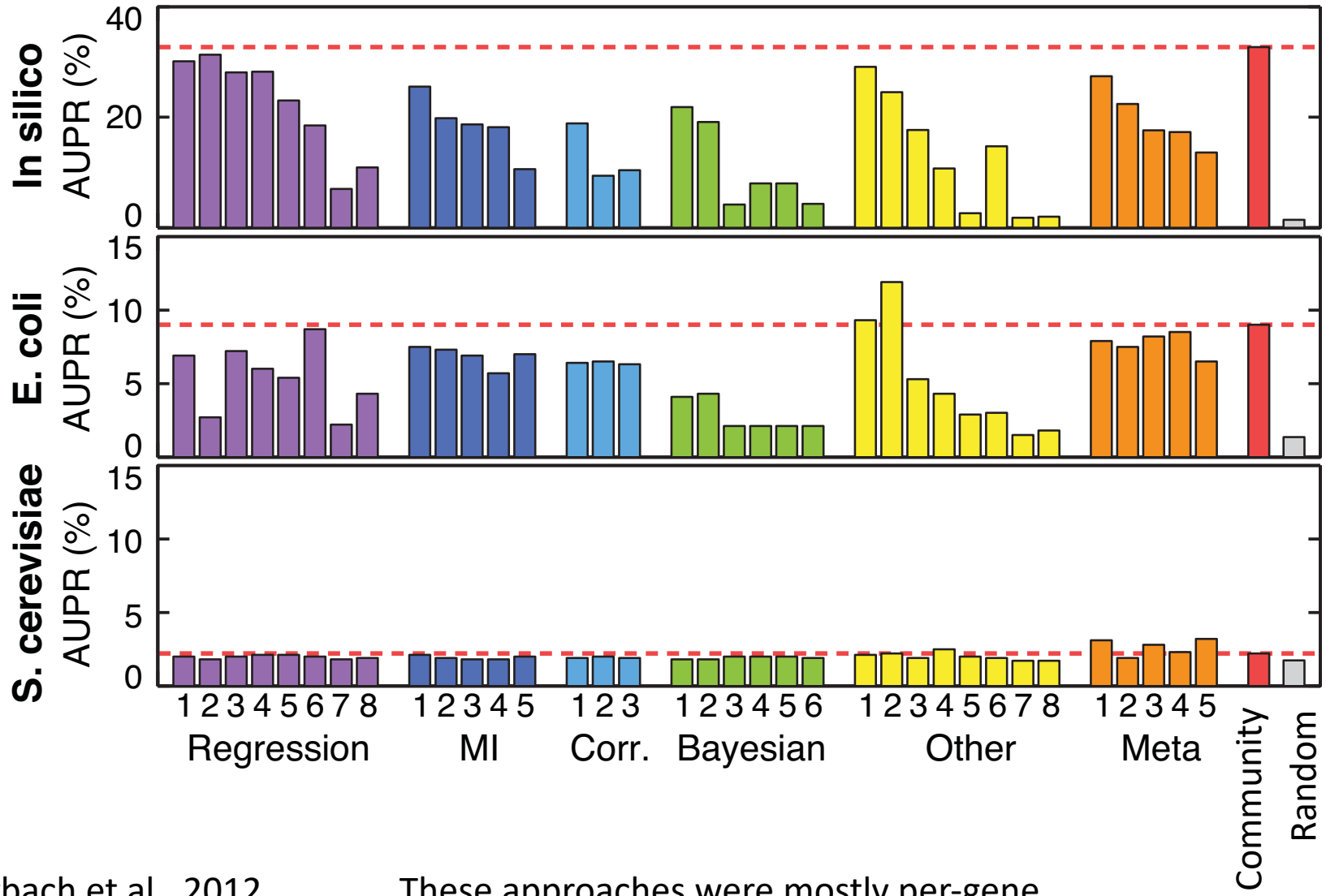


DREAM 5 challenge

Previous challenges: 2006, 2007, 2008, 2009, 2010

Marbach *et al.* 2012, Nature Methods

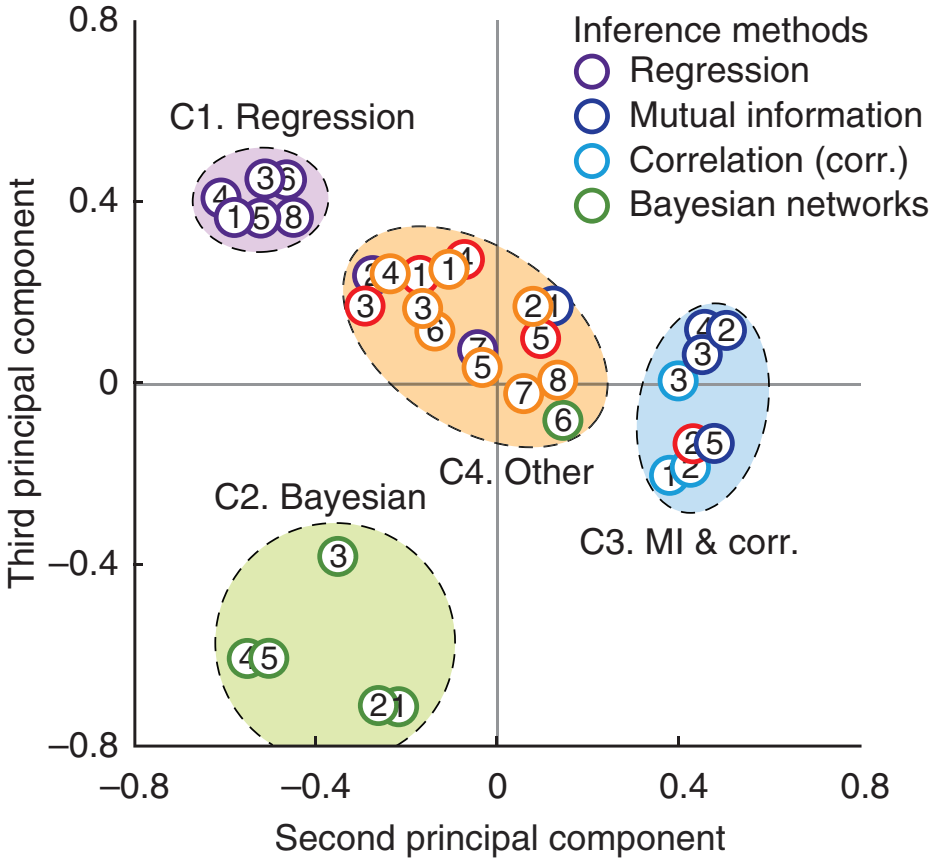
Where do different methods rank?



Marbach et al., 2012

These approaches were mostly per-gene

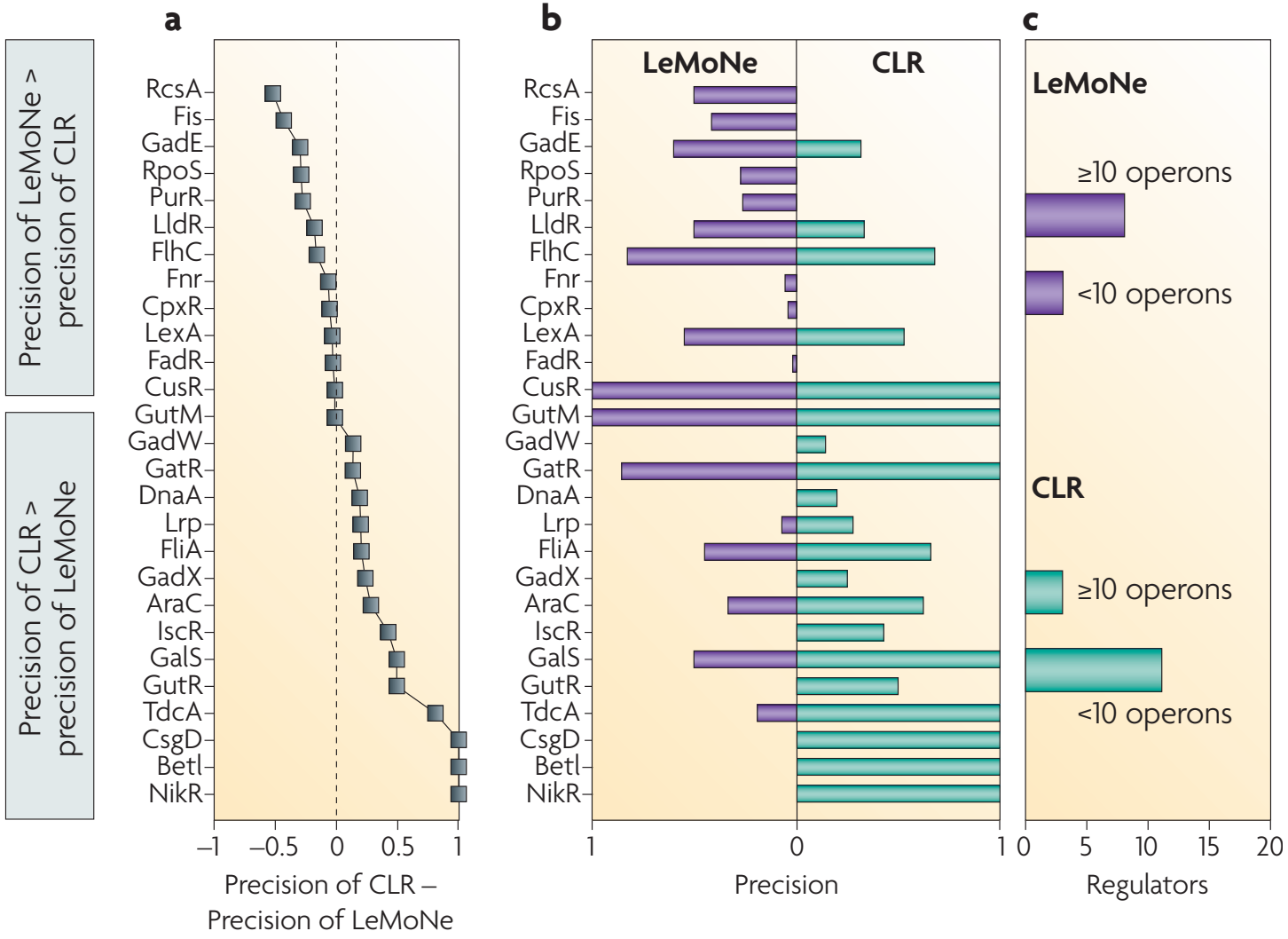
Methods tend to cluster together



These approaches were mostly per-gene

Marbach et al., 2012

Comparing per-module (LeMoNe) and per-gene (CLR) methods



Marchal & De Smet, Nature Reviews Microbiology, 2010

Some comments about expression-based network inference methods

- We have seen multiple types of algorithms to learn these networks
  - Per-gene methods (learn regulators for individual genes)
    - Sparse candidate, GENIE3, ARACNE, CLR
  - Per-module methods
    - Module networks: learn regulators for sets of genes/modules
    - Other implementations of module networks exist
      - LIRNET: Learning a Prior on Regulatory Potential from eQTL Data (Su In Lee et al, Plos genetics 2009, http://www.plosgenetics.org/article/info%3Adoi%2F10.1371%2Fjournal.pgen.1000358)
      - LeMoNe: Learning Module Networks (Michoel et al 2007, http://www.biomedcentral.com/1471-2105/8/S2/S5)
  - Methods that combine per-gene and per-module (MERLIN)
- Methods differ in
  - how they quantify dependence between genes
  - Higher-order or pairwise
  - Focus on structure or structure & parameters
- Expression alone is not enough to infer the structure of the network
- Integrative approaches that combine expression with other types of data are likely more successful (next lectures)

References

- Markowetz, Florian and Rainer Spang. "Inferring cellular networks-a review.." *BMC bioinformatics* 8 Suppl 6 (2007): S5+.
- N. Friedman, M. Linial, I. Nachman, and D. Pe'er, "Using bayesian networks to analyze expression data," *Journal of Computational Biology*, vol. 7, no. 3-4, pp. 601-620, Aug. 2000. [Online]. Available: http://dx.doi.org/10.1089/106652700750050961
- Dependency Networks for Inference, Collaborative Filtering and Data visualization Heckerman, Chickering, Meek, Rounthwaite, Kadie 2000
- Inferring Regulatory Networks from Expression Data Using Tree-Based Methods Van Anh Huynh-Thu, Alexandre Irrthum, Louis Wehenkel, Pierre Geurts, Plos One 2010
- D. Marbach et al., "Wisdom of crowds for robust gene network inference," *Nature Methods*, vol. 9, no. 8, pp. 796-804, Jul. 2012. [Online]. Available: http://dx.doi.org/10.1038/nmeth.
- R. De Smet and K. Marchal, "Advantages and limitations of current network inference methods." *Nature reviews. Microbiology*, vol. 8, no. 10, pp. 717-729, Oct. 2010. [Online]. Available: http://dx.doi.org/10.1038/nrmicro2419