Computational methods for inferring gene regulatory networks

Sushmita Roy
sroy@biostat.wisc.edu
STAT/BMI 877: Statistical Methods for Molecular Biology
https://www.biostat.wisc.edu/~kendzior/STAT877/

Feb 21st 2017

Some of the material covered in this lecture is adapted from BMI 576
Why networks?

“A system is an entity that **maintains its function** through the **interaction of its parts**”

— Kohl & Noble
To understand cells as systems: measure, model, predict, refine

Uwe Sauer, Matthias Heinemann, Nicola Zamboni, Science 2007
Plan for this segment

• Introduction to molecular networks and expression-based network inference (Today, Feb 21st)
• Module-based network inference methods (Thursday, Feb 23rd)
• Integrative network inference (Thursday, Feb 23rd)
Goals for today

• Background
  – Introductory graph theory
  – Different types of cellular networks
• Expression-based regulatory network inference
  – Per-gene vs Per-module methods
• Probabilistic graphical models to represent regulatory networks
  – Bayesian networks (Sparse candidates)
  – Dependency networks (GENIE3)
A network

• Describes connectivity patterns between the parts of a system
  – Vertex/Nodes: parts, components
  – Edges/Interactions/links: connections

• Edges can have signs, directions, and/or weight

• A network is represented as a graph
  – Node and vertex are used interchangeably
  – Edge, link, and interaction are used interchangeably
Defining a graph

- A graph is defined by two sets: $V$ and $E$
- $V$: set of vertices
- $E$: set of edges
- Depending upon the graph, we may include additional attributes about the vertices and edges
A few graph-theoretic concepts

- Directed graph
- Undirected graph
- Weighted graph
- Degree of a node
- Subnetworks/subgraphs
  - Path
  - Cycle
  - Cliques
  - Connected component
Representing a graph

- Adjacency matrix
- Adjacency list
Adjacency matrix

Matrix-based representation

\[
\begin{array}{cccccccccc}
A & B & C & D & E & F & G & H \\
\hline
A & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \\
B & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\
C & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
D & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
E & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
F & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\
G & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\
H & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
\end{array}
\]

\[V=\{A, B, C, D, E, F, G, H\}\]
\[E=\text{set of vertex pairs } (u, v) \text{ connected with an edge}\]

Adapted from Barabasi & Oltvai, Nature Reviews Genetics 2004
Adjacency list

List based representation; space efficient

A → B → C → D → E → H
B → A → C → G
C → A → B
D → A
E → A → F
F → G → E
G → B → F → H
H → A → G
Directed graphs

Edges have directionality on them

Adjacency matrix is no longer symmetric
Weighted graphs

Edges have weights on them; we can have directed and undirected weighted graphs. The example shown is that of a directed weighted graph.
Node degree

- Undirected network
  - Degree, $k$: Number of neighbors of a node

- Directed network
  - In degree, $k_{in}$: Number of incoming edges
  - Out degree, $k_{out}$: Number of outgoing edges

In degree of B is 1
Out degree of A is 2
A subgraph of a graph is defined by the subset of nodes and edges. We will use subgraph and subnetwork interchangeably.

Path: a set of connected edges from one node to another

Path length: The total number of edges in a path

Shortest path: The path between two vertices with the shortest path length

Cycle: a path which starts and ends at the same node
**Subnetworks/Subgraphs**

**Connected component:** A subgraph spanning a vertex subset where every vertex can be “reached” from another vertex.

**Clique:** A set of nodes that have edges between all pairs of vertices.

Two connected components

5 node clique
Different types of networks

- **Physical networks**
  - *Transcriptional regulatory networks*: interactions between regulatory proteins (transcription factors) and genes
  - *Protein-protein*: interactions among proteins
  - *Signaling networks*: protein-protein and protein-small molecule interactions to relay signals from outside the cell to the nucleus

- **Functional networks**
  - *Metabolic*: reactions through which enzymes convert substrates to products
  - *Genetic*: interactions among genes which when perturbed together produce a significant phenotype than when individually perturbed
Transcriptional regulatory networks

- Directed, signed, weighted graph
- Nodes: TFs and Target genes
- Edges: A regulates B’s expression level

Regulatory network of *E. coli*. 153 TFs (green & light red), 1319 targets

Vargas and Santillan, 2008
Reactions associated with Galactose metabolism
Protein-protein interaction networks

- Un/weighted graph
- Nodes: Proteins
- Edges: Protein X physically interacts with protein Y

Barabasi et al. 2003
Goals for today

• Background
  – Introductory graph theory
  – Different types of cellular networks

• Expression-based regulatory network inference
  – Per-gene vs Per-module methods

• Probabilistic graphical models to represent regulatory networks
  – Bayesian networks (Sparse candidates)
  – Dependency networks (GENIE3)
Computational inference of transcriptional regulatory networks

- We will focus on transcriptional regulatory networks
- These networks control what genes get activated when
- Precise gene activation or inactivation is crucial for many biological processes
- Microarrays and RNA-seq allows us to systematically measure gene activity levels
Why do we need to computationally infer transcriptional networks?

• Why infer transcriptional networks?
  – To understand how gene expression is controlled
  – To understand how cells process extra-cellular cues
  – Many diseases are associated with changes in transcriptional networks

• Why do so computationally?
  – Experimental detection of networks is hard, expensive
  – A first step towards having an *in silico* model of a cell
  – A model can be used to make predictions that can be tested and refine the model
Methods to infer regulatory networks

- **Experimental**
  - ChIP-chip and ChIP-seq
  - Factor/regulator knockout followed by genome-wide transcriptome measurements
  - DNase I/ATAC-seq + motif finding

- **Computational**
  - Supervised network inference
  - Unsupervised network inference
    - Expression-based network inference
Types of data for reconstructing transcriptional networks

- **Node-specific datasets**
  - Transcriptomes: Genome-wide mRNA levels
  - Can potentially recover genome-wide regulatory networks

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

---

Phenotypic Variation in Yeast


Gene expression levels

Samples

Gene expression levels

Gene

ChIP

motif

 Samples

Gene

ChIP

motif

 doi:10.1371/journal.pgen.1000223.g003
A simple example of transcriptional regulation

Assume HSP12’s expression is dependent upon Hot1 and Sko1 binding to HSP12’s promoter.
What do we want a model for a regulatory network to capture?

HSP12’s expression is dependent upon Hot1 and Sko1 binding to HSP12’s promoter.

Structure
Who are the regulators?

Function
How do they determine expression levels?

Booleans
Linear
Diff. Eqns
Probabilistic

\[
X_3 = \psi(X_1, X_2)
\]
Mathematical representations of regulatory networks

Models differ in the function that maps regulator input levels to target levels

Boolean Networks

Differential equations

Probabilistic graphical models

\[
\begin{align*}
\frac{dX_3(t)}{dt} &= \kappa \ g(X_1(t), X_2(t)) \\
&- rX_3(t)
\end{align*}
\]

\[
P(X_3 \mid X_1, X_2) = N(X_1a + X_2b, \sigma)
\]
Regulatory network inference from expression

Genes

Expression level of gene $i$ in experiment $j$

Experiments

Expression-based network inference

X1

Structure

X2

X3 = f(X1, X2)

Function

Regulatory network inference from expression
Expression-based regulatory network inference

• Given
  – A set of measured mRNA levels across multiple biological samples

• Do
  – Infer the regulators of genes
  – Infer how regulators specify the expression of a gene

• Algorithms for network reconstruction vary based on their meaning of interaction
Two classes of expression-based methods

• Per-gene/direct methods (Today)

• Module based methods (Thursday)
Per-gene methods

- Key idea: find the regulators that “best explain” expression level of a gene

Probabilistic graphical methods
- Bayesian network
  - Sparse Candidates
- Dependency networks
  - GENIE3, TIGRESS

Information theoretic methods
- Context Likelihood of relatedness
- ARACNE
Module-based methods

- Find regulators for an entire module
  - Assume genes in the same module have the same regulators
- Module Networks (Segal et al. 2005)
- Stochastic LeMoNe (Joshi et al. 2008)
A non-exhaustive list of expression-based network inference method

<table>
<thead>
<tr>
<th>Method Name</th>
<th>Per-module</th>
<th>Per-gene</th>
<th>Parameters</th>
<th>Model type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sparse candidate</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>Bayesian network</td>
</tr>
<tr>
<td>CLR</td>
<td>✓</td>
<td></td>
<td></td>
<td>Information theoretic</td>
</tr>
<tr>
<td>ARACNE</td>
<td>✓</td>
<td></td>
<td></td>
<td>Information theoretic</td>
</tr>
<tr>
<td>TIGRESS</td>
<td>✓</td>
<td></td>
<td></td>
<td>Dependency network</td>
</tr>
<tr>
<td>Inferelator</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>Dependency network</td>
</tr>
<tr>
<td>GENIE3</td>
<td>✓</td>
<td></td>
<td></td>
<td>Dependency network</td>
</tr>
<tr>
<td>ModuleNetworks</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>Bayesian network</td>
</tr>
<tr>
<td>LemonTree</td>
<td>✓</td>
<td></td>
<td></td>
<td>Dependency network</td>
</tr>
<tr>
<td>WGCNA</td>
<td>✓</td>
<td></td>
<td></td>
<td>Correlation</td>
</tr>
</tbody>
</table>
Goals for today

• Background
  – Introductory graph theory
  – Different types of cellular networks
• Expression-based regulatory network inference
  – Per-gene vs Per-module methods
• Probabilistic graphical models to represent regulatory networks
  – Bayesian networks (Sparse candidates)
  – Dependency networks (GENIE3)
Probabilistic graphical models (PGMs)

- A marriage between probability and graph theory
- Nodes on the graph represent random variables
- Graph structure specifies statistical dependency structure
- Graph parameters specify the nature of the dependency
- PGMs can be directed or undirected
- Examples of PGMs: Bayesian networks, Dependency networks, Markov networks, Factor graphs
Different types of probabilistic graphs

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

- An undirected graph
- Edges represent high correlation
  - Need to determine what “high” is
- Edge weights denote different values of correlation
- Cannot discriminate between direct and indirect correlations

Random variables represent gene expression levels

- Msb2
- Sho1
- Ste20

A measure of statistical correlation (e.g. Pearson’s correlation)

An undirected weighted graph.
Popular examples of correlational networks

- Weighted Gene Co-expression Network Analysis (WGCNA)
  - Zhang and Horvath 2005

- Relevance Networks
  - Butte & Kohane, 2000 Pacific symposium of biocomputing
Limitations of correlational networks

- Correlational networks cannot distinguish between direct and indirect dependencies.
- This makes them less interpretable than other PGMs.

For any co-expression network, there are several possible regulatory networks that can explain these correlations.

What we would like is to be able to discriminate between direct and indirect dependencies.

Here we need to review conditional independencies.
Conditional independencies in PGMs

• The different classes of models we will see are based on a general notion of specifying statistical independence
• Suppose we have two genes $X$ and $Y$. We add an edge between $X$ and $Y$ if $X$ and $Y$ are not independent given a third set $Z$.
• Depending upon $Z$ we will have a family of different PGMs
Conditional independence and PGMs

• Correlational networks
  – $Z$ is the empty set

• Markov networks
  – $X$ and $Y$ are not independent given all other variables
  – Gaussian Graphical models are a special case

• Dependency networks
  – Approximate Markov networks
  – May not be associated with a valid joint distribution

• First-order conditional independence models
  – Explain the correlation between two variables by a third variable

• Bayesian networks
  – Generalize first-order conditional independence models
Bayesian networks (BN)

- A special type of probabilistic graphical model
- Has two parts:
  - A graph which is directed and acyclic
  - A set of conditional distributions
- Directed Acyclic Graph (DAG)
  - The nodes denote random variables $X_1 \ldots X_N$
  - The edges
    - encode statistical dependencies between the random variables
    - establish parent child relationships
- Each node $X_i$ has a conditional probability distribution (CPD) representing $P(X_i | Pa(X_i))$; $Pa$: Parents
- Provides a tractable way to represent large joint distributions
Key questions in Bayesian networks

• What do the CPDs look like?
• What independence assertions can be made in Bayesian networks?
• How can we learn them from expression data?
Notation

- $X_i$: $i^{th}$ random variable
- If there are few random variables, we will just use upper case letters. E.g. $A, B, C$.
- $X = \{X_1, .., X_p\}$: set of $p$ random variables
- $x_i^k$: An assignment of $X_i$ in the $k^{th}$ sample
- $x_{-i}^k$: Set of assignments to all variables other than $X_i$ in the $k^{th}$ sample
- $D$: A dataset constituting the collection of all joint assignments
An example Bayesian network

Adapted from Kevin Murphy: Intro to Graphical models and Bayes networks: http://www.cs.ubc.ca/~murphyk/Bayes/bnintro.html
Expression data matrix

$p$ Genes

$N$ Experiments/Time points etc

Observations (expression levels) of all variables in sample $k$

Observations of variable $X_i$ in all $N$ experiments

$x_{-j}^k$
Bayesian networks compactly represent joint distributions

\[ P(X_1, \cdots, X_p) = \prod_{i=1}^{p} P(X_i | Pa(X_i)) \]
Example Bayesian network of 5 variables

\[ P(X) = P(X_1, X_2, X_3, X_4, X_5) \]

No independence assertions

Independence assertions

\[ P(X) = P(X_1)P(X_2)P(X_4)P(X_3 | X_1, X_2)P(X_5 | X_3, X_4) \]

Assume \( X_i \) is binary

Needs \( 2^5 \) measurements

Needs \( 2^3 \) measurements

No independence assertions

Needs \( 2^5 \) measurements
CPD in Bayesian networks

• The CPD $P(X_i|Pa(X_i))$ specifies a distribution over values of $X_i$ for each combination of values of $Pa(X_i)$
• CPD $P(X_i|Pa(X_i))$ can be parameterized in different ways
  • $X_i$ are discrete random variables
    – Conditional probability table or tree
  • $X_i$ are continuous random variables
    – CPD can be conditional Gaussians or regression trees
Representing CPDs as tables

- Consider four binary variables $X_1, X_2, X_3, X_4$

\[
P(X_4 \mid X_1, X_2, X_3)
\]

as a table

<table>
<thead>
<tr>
<th>$X_1$</th>
<th>$X_2$</th>
<th>$X_3$</th>
<th>$X_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>t</td>
<td>t</td>
<td>t</td>
<td>0.9</td>
</tr>
<tr>
<td>t</td>
<td>t</td>
<td>f</td>
<td>0.9</td>
</tr>
<tr>
<td>t</td>
<td>f</td>
<td>t</td>
<td>0.9</td>
</tr>
<tr>
<td>t</td>
<td>f</td>
<td>f</td>
<td>0.9</td>
</tr>
<tr>
<td>f</td>
<td>t</td>
<td>t</td>
<td>0.8</td>
</tr>
<tr>
<td>f</td>
<td>t</td>
<td>f</td>
<td>0.5</td>
</tr>
<tr>
<td>f</td>
<td>f</td>
<td>t</td>
<td>0.5</td>
</tr>
<tr>
<td>f</td>
<td>f</td>
<td>f</td>
<td>0.5</td>
</tr>
</tbody>
</table>

Pa($X_4$): $X_1, X_2, X_3$
Estimating CPD table from data

- Assume we observe the following assignments for $X_1, X_2, X_3, X_4$

<table>
<thead>
<tr>
<th>$X_1$</th>
<th>$X_2$</th>
<th>$X_3$</th>
<th>$X_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
</tbody>
</table>

$N=7$

For each joint assignment to $X_1, X_2, X_3$, estimate the probabilities for each value of $X_4$

For example, consider $X_1=T$, $X_2=F$, $X_3=T$

$P(X_4=T|X_1=T, X_2=F, X_3=T)=2/4$

$P(X_4=F|X_1=T, X_2=F, X_3=T)=2/4$
Bayesian network representation of a regulatory network

Inside the cell

**Random variables**

- **Hot1:** $X_1$
- **Sko1:** $X_2$
- **Hsp12:** $X_3$

**Bayesian network**

**REGULATORS (PARENTS)**

- $P(X_1)$
- $P(X_2)$

**TARGET (CHILD)**

- $P(X_3|X_1,X_2)$

**Bayesian network**

- $P(X_3)$

**Inside the cell**

**Random variables**

- $X_1$
- $X_2$
- $X_3$
Learning problems in Bayesian networks

• Parameter learning on known graph structure
  – Given a set of joint assignments of the random variables, estimate the parameters of the model

• Structure learning
  – Given a set of joint assignments of the random variables, estimate the structure and parameters of the model
  – Structure learning subsumes parameter learning
Structure learning using score-based search

\( \text{Score}(B) \) describes how well \( B \) describes the data

\( \text{Score}(B_1) \), \( \text{Score}(B_2) \), \( \text{Score}(B_3) \), \( \text{Score}(B_m) \)

Exhaustive search is not computationally tractable
Scores for Bayesian networks

• Maximum likelihood

\[ \text{Score}_{ML} = \max_{\theta} P(D|G, \theta) \]

• Regularized maximum likelihood

\[ \text{Score}_{BIC} = \max_{\theta} P(D|G, \theta) - \frac{d}{2} \log(N) \]

\[ d : \text{no. of parameters} \]

\[ N : \text{no. of samples} \]

• Bayesian score

\[ \text{Score}_{Bayes} = P(G|D) \propto P(D|G)P(G) \]
Decomposability of scores

• The score of a Bayesian network $\mathcal{B}$ decomposes over individual variables

\[
\text{Score}(\mathcal{B}) = \sum_i S(X_i)
\]

\[
S(X_i) = \prod_{d=1}^{N} P(X_i = x_i^d | Pa(X_i) = \text{pa}_{X_i}^d)
\]

$\text{pa}_{X_i}^d$ Joint assignment to $Pa(X_i)$ in the $d^{th}$ sample

• Enables efficient computation of the score change to local changes
Graph structure learning from expression is a computationally difficult problem

• Given 2 TFs and 3 nodes how many possible networks can there be?

There can be a total of $2^6$ possible networks.
Greedy Hill climbing to search Bayesian network space

- Input: Data $D$, An initial Bayesian network, $B_0=\{G_0, \Theta_0\}$
- Output: $B_{\text{best}}$
- Loop for $r=1, 2..$ until convergence:
  - $\{B_r^1, \ldots, B_r^m\} = \text{Neighbors}(B_r)$ by making local changes to $B_r$
  - $B_{r+1} = \text{arg max}_j (\text{Score}(B_r^j))$
- Termination:
  - $B_{\text{best}} = B_r$
Local changes to $B_i$

Current network

![Diagram of network with nodes A, B, C, D and edges between them.]

- **Add an edge**: $B_r^1$
- **Delete an edge**: $B_r^2$
- **Check for cycles**
Challenges with applying Bayesian network to genome-scale data

- Number of variables, \( p \) is in thousands
- Number of samples, \( N \) is in hundreds
Bayesian network-based methods to handle genome-scale networks

• Sparse candidate algorithm
  – Friedman, Nachman, Pe’er. 1999
  – Friedman, Linial, Nachman, Pe’er. 2000.

• Module networks
  – Segal, Pe’er, Regev, Koller, Friedman. 2005
The Sparse candidate Structure learning in Bayesian networks

• A fast Bayesian network learning algorithm
• Key idea: Identify $k$ “promising” candidate parents for each $X_i$
  – $k<<p$, $p$: number of random variables
  – Candidates define a “skeleton graph” $H$
• Restrict graph structure to select parents from $H$
• Early choices in $H$ might exclude other good parents
  – Resolve using an iterative algorithm
Sparse candidate algorithm

- **Input:**
  - A data set $D$
  - An initial Bayes net $B_0$
  - A parameter $k$: max number of parents per variable

- **Output:**
  - Final $B_r$

- **Loop for $r=1,2,..$ until convergence**
  - **Restrict**
    - Based on $D$ and $B_{r-1}$ select candidate parents $C^r_i$ for $X_i$
    - This defines a skeleton directed network $H_r$
  - **Maximize**
    - Find network $B_r$ that maximizes the score $\text{Score}(B_r)$ among networks satisfying
    $$Pa^r(X_i) \subseteq C^r_i$$

- **Termination:** Return $B_r$
Selecting candidate parents in the Restrict Step

- A good parent for $X_i$ is one with strong statistical dependence with $X_i$
  - Mutual information provides a good measure of statistical dependence $I(X_i; X_j)$
  - Mutual information should be used only as a first approximation
    - Candidate parents need to be iteratively refined to avoid missing important dependences
- A good parent for $X_i$ has the highest score improvement when added to $Pa(X_i)$
Sparse candidate learns good networks faster than hill-climbing

Greedy hill climbing takes much longer to reach a high scoring bayesian network

Dataset 1
100 variables

Dataset 2
200 variables
Some comments about choosing candidates

• How to select $k$ in the sparse candidate algorithm?

• Should $k$ be the same for all $X_i$?

• Regularized regression approaches can be used to estimate the structure of an undirected graph
  – Schmidt, Niculescu-Mizil, Murphy 2007
  – Estimate an undirected dependency network
  – Learn a Bayesian network constrained on the dependency network structure
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

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

- 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 $D_i$ by sampling with replacement $N$ samples from dataset $D$, where $N$ is the size of the original $D$
  – Learn a network $B_i=\{G_i, \Theta_i\}$

• For each feature of interest $f$, calculate confidence

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

Experimental conditions

\[
\begin{array}{ccc}
  g_{1,1} & \cdots & g_{1,n} \\
  g_{2,1} & \cdots & g_{2,n} \\
  g_{m,1} & \cdots & g_{m,n} \\
\end{array}
\]
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
Goals for today

• Background
  – Introductory graph theory
  – Different types of cellular networks

• Expression-based regulatory network inference
  – Per-gene vs Per-module methods

• Probabilistic graphical models to represent regulatory networks
  – Sparse Candidates Bayesian networks
  – GENIE3: Regression-based methods
Limitations with Bayesian networks

• Cannot model cyclic dependencies
• In practice have not been shown to be better than dependency networks
  – However, most of the evaluation has been done on structure not function
• Directionality is often not associated with causality
  – Too many hidden variables in biological systems
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

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: estimate the best predictors of a variable

- $f_j$ can be of different types.
- Learning requires estimation of each of the $f_j$ functions.
- In all cases learning requires us to minimize an error of predicting $X_j$ from its neighborhood.
Different representations of $f_j$

• 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
GENIE3: GEne Network Inference with Ensemble of trees

- Solves a set of regression problems
  - One per random variable
- 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
- Focus on generating a ranking over edges rather than a graph structure and parameters

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

• A rooted binary tree $T$
• Each node in the tree is either an interior node or a leaf node
• Interior nodes are labeled with a binary test $X_i < u$, $u$ is a real number observed in the data
• Leaf nodes are associated with univariate distributions of the child
A very simple regression tree for two variables

The tree specifies $X_3$ as a function of $X_2$.
Learning a regression tree

- Assume we are searching for the neighbors of a variable $X_3$ and it already has two parents $X_1$ and $X_2$
- $X_4$ will be considered as a new neighbor using “split” operations of existing leaf nodes

$N_l$: Gaussian associated with leaf $l$
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
A Random forest: An Ensemble of Trees

- Prediction is

\[ \hat{x}_j = \frac{1}{T} \sum_{t=1}^{T} \mu_t \]

Taken from ICCV09 tutorial by Kim, Shotton and Stenger: http://www.iis.ee.ic.ac.uk/~tkkim/iccv09_tutorial
GENIE3 algorithm sketch

• For each $X_j$, generate learning samples of input/output pairs
  
  – $LS_j = \{(x_{-j}^k, x_j^k), \, k=1..N\}$
  
  – 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 $w_{ij}$
GENIE3 algorithm sketch

Figure from Huynh-Thu et al.
Evaluation of network inference methods

- 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

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

\[
\text{Recall} = \frac{\text{# of correct edges}}{\text{# of true edges}}
\]
DREAM: Dialogue for reverse engineering assessments and methods

Community effort to assess regulatory network inference

(1) Target network

- **In silico**
  - 195 TFs
  - 1,643 genes

- **E. coli**
  - 296 TFs
  - 4,297 genes

- **S. cerevisiae**
  - 183 TFs
  - 5,667 genes

- **S. aureus**
  - 90 TFs
  - 2,677 genes

(2) Microarray compendia

- Simulation
- Experiments
  - knockouts
  - overexpress
  - antibiotics
  - toxins etc.

- 805 arrays 487conds.
- 805 arrays 487conds.
- 536 arrays 321conds.
- 160 arrays 53conds.

(3) Inferred networks

- Anonymize data
- Inference methods
  - 29 methods applied by teams (blinded)
  - 6 off-the-shelf methods
  - 35 methods tested

(4) Community network

- Integration
- Validation

- Experimentally determined interactions
  - ChIP motifs
  - etc.

(5) Performance evaluation

- True in silico network

DREAM 5 challenge


Marbach et al. 2012, Nature Methods
Where do different methods rank?

These approaches were mostly per-gene
Methods tend to cluster together

These approaches were mostly per-gene

Marbach et al., 2012
Take away points

• Network inference from expression provides a promising approach to identify cellular networks

• Graphical models are one representation of networks that have a probabilistic and graphical component
  – Network inference naturally translates to learning problems in these models

• Different types of algorithms to learn these networks
  – Per-gene methods
    • Sparse Candidate, GENIE3: learn regulators for individual genes
  – Per-module methods
    • Module networks: learn regulators for sets of genes/modules

• Successful application of Bayesian networks to expression data requires additional considerations
  – Reduce potential parents
  – Bootstrap based confidence estimation

• Network reconstruction from expression alone is challenging
  – New methods aim to combine other types of measurements (next lecture)