CS-E5885 Modeling biological networks
Boolean networks and relevance networks as models of biological networks

Harri Lähdesmäki

Department of Computer Science
Aalto University

February 12, 2021

Outline

- Boolean networks
- Relevance networks
- Introduction to information theoretics concepts
- Aracne algorithm

- This lecture is based on a collection of articles listed at the end of the slides

Boolean networks

- An (over-)simplified representation of a (biological) network system
- A generalization of binary cellular automata
  - A directed graph where each node i is associated with binary state value x_i and parent nodes $\mathrm{pa}(x_i)$, $i =, 1, \ldots, n$
  - A deterministic update rule, i.e., Boolean function, $f_i(\cdot) : \mathbb{B}^{|\mathrm{pa}(x_i)|} \rightarrow \mathbb{B}$ for each node x_i
  - Typically update rules $f_1, \ldots, f_n$ operate synchronously over time $t = 0, 1, 2, \ldots$

$$x_i(t) = f_i(\mathrm{pa}(x_i)(t))$$

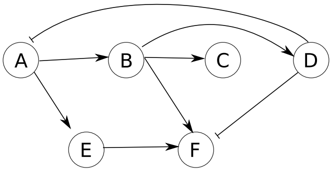
Boolean networks

- An (over-)simplified representation of a (biological) network system
- A generalization of binary cellular automata
  - A directed graph where each node i is associated with binary state value x_i and parent nodes $\mathrm{pa}(x_i)$, $i =, 1, \ldots, n$
  - A deterministic update rule, i.e., Boolean function, $f_i(\cdot) : \mathbb{B}^{|\mathrm{pa}(x_i)|} \to \mathbb{B}$ for each node x_i
  - Typically update rules $f_1, \ldots, f_n$ operate synchronously over time $t = 0, 1, 2, \ldots$

$$x_i(t) = f_i(\mathrm{pa}(x_i)(t))$$

- Boolean networks can be considered as a special case of dynamic Bayesian networks without stochasticity
- Parent variables used to predict $x_i(t)$ are the values at time point $t - 1$

Boolean networks (2)



- The rule table:

$$
\begin{aligned}
g_A(t+1) &:= NOT(g_D(t)) \\
g_E(t+1) &:= g_A(t) \\
g_B(t+1) &:= g_A(t) \\
g_C(t+1) &:= g_B(t) \\
g_F(t+1) &:= AND(g_E(t), g_B(t), \\
&\quad NOT(g_D(t))) \\
g_D(t+1) &:= g_B(t)
\end{aligned}
$$

- The state vectors

$$
\begin{aligned}
[g_i(0)]_i &= [1,0,0,0,0,0] \\
[g_i(1)]_i &= [1,1,0,0,1,0] \\
[g_i(2)]_i &= [1,1,1,1,1,1] \\
[g_i(3)]_i &= [0,1,1,1,1,0] \\
[g_i(4)]_i &= [0,0,1,1,0,0] \\
[g_i(5)]_i &= [0,0,0,0,0,0]
\end{aligned}
$$

Figure: An example of a Boolean network

Boolean networks (2)

- Primary applications include coarse-scale modeling of gene regulatory networks and signaling pathways
  - Qualitative
- Best constructed from databases of known interactions or networks

Boolean networks (2)

- Primary applications include coarse-scale modeling of gene regulatory networks and signaling pathways
  - Qualitative
- Best constructed from databases of known interactions or networks
- A Boolean network model can be easily used to study effects and propagation of
  - External stimuli
  - Gene/protein knock-down

  at qualitative level
- Can handle genome/cell-wide networks
- Lack quantitative details and can thus be misleading (or at least results need to be assessed with care)

Boolean networks (2)

- Primary applications include coarse-scale modeling of gene regulatory networks and signaling pathways
  - Qualitative
- Best constructed from databases of known interactions or networks
- A Boolean network model can be easily used to study effects and propagation of
  - External stimuli
  - Gene/protein knock-down

  at qualitative level
- Can handle genome/cell-wide networks
- Lack quantitative details and can thus be misleading (or at least results need to be assessed with care)
- Some concepts related to Boolean networks
  - Attractors, basins of attraction, criticality, sensitivity, reachability, etc.

Boolean networks (3)

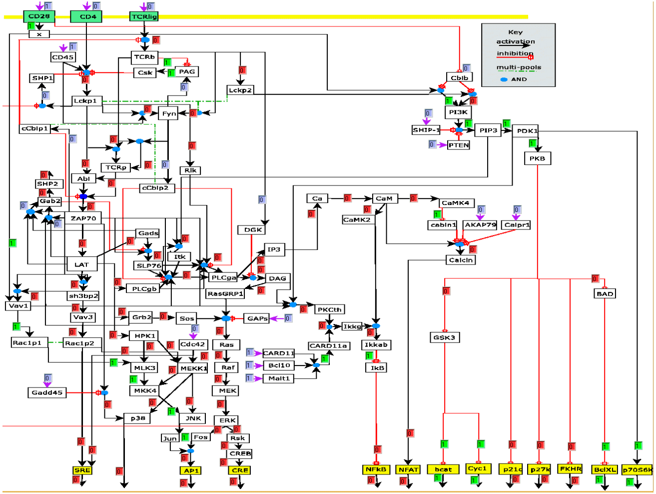
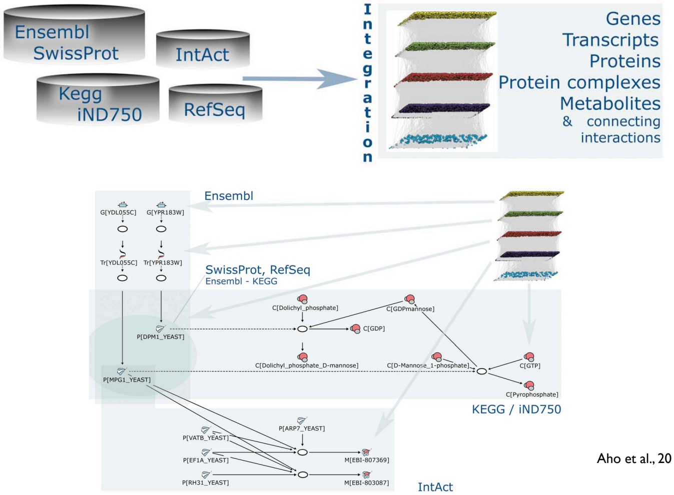


Figure: A logical model of T cell activation from (Saez-Rodriguez et al., 2007)

Boolean networks (4)



Aho et al., 2010

Figure: A comprehensive, logical model of yeast molecular network (Aho et al., 2010)

Relevance networks

- A "quick and dirty" statistical approach to find similarly behaving molecules (genes, proteins, etc.)
- Assume no prior information about the interactions in the network

Relevance networks

- A "quick and dirty" statistical approach to find similarly behaving molecules (genes, proteins, etc.)
- Assume no prior information about the interactions in the network
- Measure similarity by correlation or mutual information, i.e. the similarity between molecules' abundance as random variables
- Relevance networks:
  - Measure similarity of entities using correlation or mutual information
  - Build a similarity matrix
  - Propose interactions which have similarity value over a given threshold

Covariance

- Expectation (the average value) of a discrete-valued or real-valued random variable X

$$\mathbb{E}[X] = \sum_i p(x_i)x_i \quad \text{or} \quad \int p(x)xdx$$

- Co-variance as the measure of strength of dependency between two real-valued random variables X and Y

$$\mathrm{cov}(X, Y) = \mathbb{E}[(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])]$$

Covariance

- Expectation (the average value) of a discrete-valued or real-valued random variable X

$$\mathbb{E}[X] = \sum_i p(x_i)x_i \quad \text{or} \quad \int p(x)xdx$$

- Co-variance as the measure of strength of dependency between two real-valued random variables X and Y

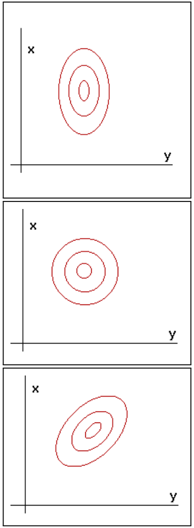
$$\mathrm{cov}(X, Y) = \mathbb{E}[(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])]$$

- Assume sample data of both X and Y: $x = (x_1, \dots, x_n)$ and $y = (y_1, \dots, y_n)$
- Sample mean and sample covariance

$$m_x = \frac{1}{n}\sum_{i=1}^{n} x_i \quad \text{and} \quad s_{xy} = \frac{1}{n}\sum_{i=1}^{n}(x_i - m_x)(y_i - m_y)$$

Covariance

- Given a data set, covariance tells us about dependencies
- Example on the right:
  - Top: no co-variance between x and y, x has higher variance than y, diagonal co-variance matrix with inequal entries
  - Middle: no co-variance between x and y, equal variance for x and y, diagonal co-variance matrix with equal entries
  - Bottom: x and y co-vary, co-variance matrix will have non-zero off-diagonal entries



Correlation matrix and correlation network

- Correlation is computed from covariance by normalizing by the standard deviations σ_x and σ_y

$$\mathrm{corr}(x, y) = \frac{\mathrm{cov}(x, y)}{\sigma_x \sigma_y} = \frac{\mathrm{cov}(x, y)}{\sqrt{\mathrm{cov}(x, x)\mathrm{cov}(y, y)}}$$

Correlation matrix and correlation network

- Correlation is computed from covariance by normalizing by the standard deviations σ_x and σ_y

$$\mathrm{corr}(x, y) = \frac{\mathrm{cov}(x, y)}{\sigma_x \sigma_y} = \frac{\mathrm{cov}(x, y)}{\sqrt{\mathrm{cov}(x, x)\mathrm{cov}(y, y)}}$$

- For p variables $\mathbf{x} = (x_1, \ldots, x_p)$, empirical correlation matrix R (size p-by-p) collects all pairwise empirical correlations

$$r_{x_i x_j} = \frac{s_{x_i x_j}}{\sqrt{s_{x_i x_i} s_{x_j x_j}}}$$

Correlation matrix and correlation network

- Correlation is computed from covariance by normalizing by the standard deviations σ_x and σ_y

$$\mathrm{corr}(x, y) = \frac{\mathrm{cov}(x, y)}{\sigma_x \sigma_y} = \frac{\mathrm{cov}(x, y)}{\sqrt{\mathrm{cov}(x, x)\mathrm{cov}(y, y)}}$$

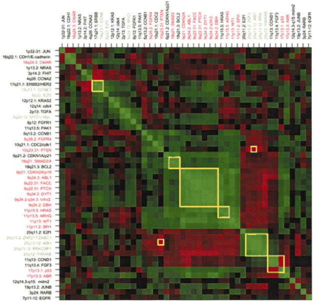
- For p variables $\mathbf{x} = (x_1, \ldots, x_p)$, empirical correlation matrix R (size p-by-p) collects all pairwise empirical correlations

$$r_{x_i x_j} = \frac{s_{x_i x_j}}{\sqrt{s_{x_i x_i} s_{x_j x_j}}}$$

- Correlation network for the data X (n measurements, p variables) is obtained from R by defining a threshold $0 \leq \tau \leq 1$ and drawing an edge between vertex x_i and x_j if $|r_{x_i x_j}| \geq \tau$

Correlation matrix and correlation network

- Different thresholds give different networks
- A large threshold gives high precision (predictions are correct), but low recall (most interaction are not found)
- A smaller threshold has high recall (most interactions are revealed), but low precision (many errors)



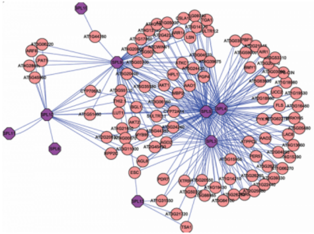
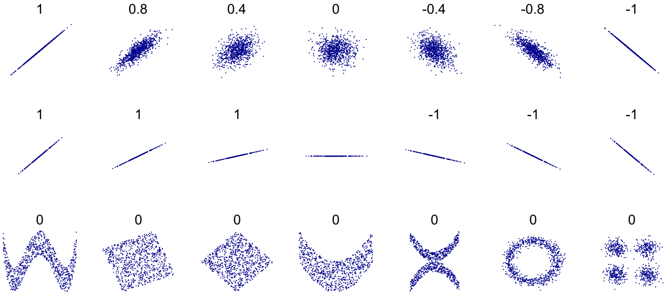


Figure: From `https://en.wikipedia.org/wiki/Correlation_and_dependence`

Weakness of correlation

- Correlation measures linear dependency



Weakness of correlation thresholding

- Obtaining the edges of the graph by thresholding the correlation (or covariance) matrix is simple
- However, the method is sensitive in detecting spurious correlations that are due to other (controlling) variables
- For example:
  - Protein interactions $z - x$ and $z - y$ may be reflected as a correlation between $x - y$
  - However, there may not be any physical interaction between them x and y
- Correlation is an inherently pairwise concept: adding variables to the data does not have effect on correlation between existing vertices

Mutual information

- An alternative to correlation is mutual information (MI), which also measures the statistical dependency between genes
  - Measures how much the uncertainty in the variable A is reduced by knowing the variable B
  - If A determines B completely (i.e. deterministic relationship), then MI is maximal
  - If A is not related to B at all, then MI is zero

Mutual information

- An alternative to correlation is mutual information (MI), which also measures the statistical dependency between genes
  - Measures how much the uncertainty in the variable A is reduced by knowing the variable B
  - If A determines B completely (i.e. deterministic relationship), then MI is maximal
  - If A is not related to B at all, then MI is zero
- Defined for random variables X and Y with either continuous or discrete values, with proper probability distributions $p(X = x)$ and $p(Y = y)$
- We will assume discrete-valued random variables for now

Information and entropy

- Information content (in bits) of a data item (or a message) $X = x$ with probability distribution $p(X = x)$ is $I(X = x) = -\log p(X = x)$, i.e., more unlikely an event is, more information it contains
  - I.e. a deterministic event has no information, unlikely event has high information
  - Information thus measures uncertainty or "surprisal" of an event

Information and entropy

- Information content (in bits) of a data item (or a message) $X = x$ with probability distribution $p(X = x)$ is $I(X = x) = -\log p(X = x)$, i.e., more unlikely an event is, more information it contains
  - I.e. a deterministic event has no information, unlikely event has high information
  - Information thus measures uncertainty or "surprisal" of an event
- Entropy $H(X)$ is the expected information (i.e. expected uncertainty)

$$H(X) = \mathbb{E}[I(X)] = -\sum_{i=1}^{n} p(x_i) \log p(x_i)$$

Information and entropy

- Information content (in bits) of a data item (or a message) $X = x$ with probability distribution $p(X = x)$ is $I(X = x) = -\log p(X = x)$, i.e., more unlikely an event is, more information it contains
  - I.e. a deterministic event has no information, unlikely event has high information
  - Information thus measures uncertainty or "surprisal" of an event
- Entropy $H(X)$ is the expected information (i.e. expected uncertainty)

$$H(X) = \mathbb{E}[I(X)] = -\sum_{i=1}^{n} p(x_i) \log p(x_i)$$

- Entropy is thus the "average" uncertainty or suprisal we are going to see in a random variable
  - Entropy is highest with uniform distributions: i.e. no idea what values we are going to get
  - Entropy is lowest with highly peaked distributions: we already know very well what values we are going to get

Entropy example

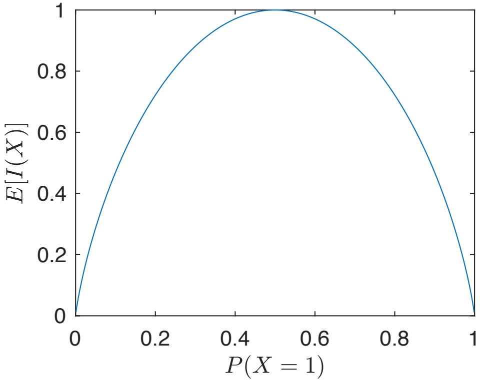
- A coin flip is a random variable X with two outcomes $\{\text{tails, heads}\}$
- A fair coin has probability distribution $p(X = \text{heads}) = 0.5$ and $p(X = \text{tails}) = 0.5$
- The entropy is thus:

$$\begin{aligned}\mathbb{E}[I(\text{``coin''})] &= -p(\text{heads}) \log p(\text{heads}) - p(\text{tails}) \log p(\text{tails}) \\ &= -0.5 \cdot \log(0.5) - 0.5 \cdot \log(0.5) \\ &= 1 \;\; (\text{with binary log})\end{aligned}$$

- An unfair coin with $p(X = \text{heads}) = 0.9$ has entropy $\mathbb{E}[I(\text{``biased coin''})] = -0.9 \cdot \log(0.9) - 0.1 \cdot \log(0.1) \approx 0.4$

Entropy example

- Entropies for biased coins



Relative entropy

- The relative entropy is a measure between two distributions $p(X)$ and $q(X)$
- Better known as the Kullback-Leibler distance between two probability distributions

$$
\begin{aligned}
D_{\mathrm{KL}}(p||q) &= \sum_{i=1}^{n} p(x_i) \log \frac{p(x_i)}{q(x_i)} \\
&= \mathbb{E}_p \left[\log \frac{p(X)}{q(X)} \right]
\end{aligned}
$$

Relative entropy

- The relative entropy is a measure between two distributions $p(X)$ and $q(X)$
- Better known as the Kullback-Leibler distance between two probability distributions

$$
\begin{aligned}
D_{\mathrm{KL}}(p||q) &= \sum_{i=1}^{n} p(x_i) \log \frac{p(x_i)}{q(x_i)} \\
&= \mathbb{E}_p \left[\log \frac{p(X)}{q(X)} \right]
\end{aligned}
$$

- Relative entropy is non-negative and is zero iff $p = q$ for all x_i
- But relative entropy is not a distance measure because
  - It is not symmetric: $D_{\mathrm{KL}}(p||q) \neq D_{\mathrm{KL}}(q||p)$
  - It does not satisfy the triangle inequality

Mutual information

- Mutual information (MI) is a measure of the amount of information that one random variable Y contains about another random variable X
- Given both the joint distribution $p(x, y)$ and the marginal distributions $p(x) = \sum_y p(x, y)$ and $p(y) = \sum_x p(x, y)$, the mutual information $I(X|Y)$ is the relative entropy between the joint distribution $p(x, y)$ and the product distribution $p(x)p(y)$

$$
\begin{aligned}
I(X|Y) &= D_{\mathrm{KL}}(p(X, Y)||p(X)p(Y)) \\
&= \sum_{x,y} p(x, y) \log \frac{p(x, y)}{p(x)p(y)}
\end{aligned}
$$

Mutual information

- Mutual information (MI) is a measure of the amount of information that one random variable Y contains about another random variable X
- Given both the joint distribution $p(x, y)$ and the marginal distributions $p(x) = \sum_y p(x, y)$ and $p(y) = \sum_x p(x, y)$, the mutual information $I(X|Y)$ is the relative entropy between the joint distribution $p(x, y)$ and the product distribution $p(x)p(y)$

$$
\begin{aligned}
I(X|Y) &= D_{\mathrm{KL}}(p(X, Y)||p(X)p(Y)) \\
&= \sum_{x,y} p(x, y) \log \frac{p(x, y)}{p(x)p(y)}
\end{aligned}
$$

- $I(X|Y)$ measures the similarity between $p(X, Y)$ and $p(X)p(Y)$
- An illustration:

Data processing inequality

- Consider three random variables that satisfy a Markov chain (directed graphical model)

$$X \to Y \to Z,$$

  i.e., $p(x, y, z) = p(x)p(y|x)p(z|y)$

- Now we have

$$\begin{aligned} p(x, z|y) &= \frac{p(x, y, z)}{p(y)} = \frac{p(x)p(y|x)p(z|y)}{p(y)} \\ &= \frac{p(x)p(y, x)p(z|y)}{p(x)p(y)} = \frac{p(x|y)p(y)p(z|y)}{p(y)} \\ &= p(x|y)p(z|y) \end{aligned}$$

- The above Markov chain is thus equivalent to a conditional independency

$$X \to Y \to Z \;\text{ iff }\; X, Z \perp Y$$

Data processing inequality

- Consider three random variables that satisfy a Markov chain (directed graphical model)

$$X \to Y \to Z,$$

- Data processing inequality theorem says that if $X \to Y \to Z$ then

$$I(X|Y) \geq I(X|Z) \ \text{ and } \ I(Y|Z) \geq I(X|Z)$$

- Thus $I(X|Z) \leq \min\left(I(X|Y), I(Y|Z)\right)$

Aracne algorithm

- Aracne algorithm (Margolin et al., 2006) uses the MI to find statistically dependent pairs of variables/molecules/genes while removing redundant statistical correlations

Aracne algorithm

- Aracne algorithm (Margolin et al., 2006) uses the MI to find statistically dependent pairs of variables/molecules/genes while removing redundant statistical correlations
- Aracne initialises the network G by adding an edge between variables x_i and x_j if $I(X_i|X_j) \geq I_0$, where I_0 is a threshold
- Aracne then examines all triplets of variables x_i, x_j and x_k for which all three MI values exceed I_0 and removes the edge with the smallest MI
- All possible triplets are analyzed regardless of whether some variables have been considered already in other triplets
  - Does not depend on the order the variable triplets are processed

Aracne algorithm

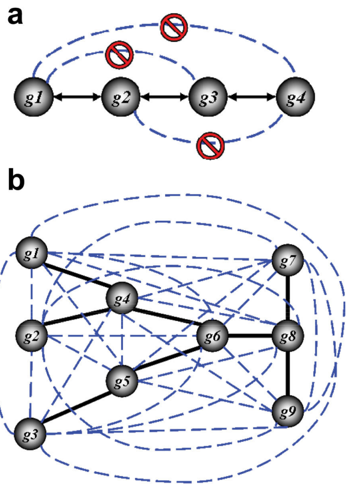


Figure: Illustration of the data processing inequality from (Margolis et al., 2006)

Aracne algorithm

- We have derived the information theoretic measures assuming discrete-valued random variables
- Real-world data is typically continuous
- The above information theoretic measures can be generalized to continuous-valued variables by replacing the sums with integrals
- Integrals can be approximated by numerical integration

Aracne algorithm

- We have derived the information theoretic measures assuming discrete-valued random variables
- Real-world data is typically continuous
- The above information theoretic measures can be generalized to continuous-valued variables by replacing the sums with integrals
- Integrals can be approximated by numerical integration
- Observed data may come from an unknown probability density
- In Aracne algorithm unknown densities are estimated using the Gaussian kernel density estimation

Kernel density estimator

- Assume observed data $\mathcal{D} = (\mathbf{x}_1, \ldots, \mathbf{x}_N)$, where each $\mathbf{x}_i = (x_{i1}, \ldots, x_{id})^T \in \mathbb{R}^d$
- The Gaussian kernel density estimate is defined as

$$p(\mathbf{x}|\mathcal{D}) = \frac{1}{N} \sum_{i=1}^{N} \mathcal{N}(\mathbf{x}|\mathbf{x}_i, \sigma^2 I),$$

  where I is the d-by-d identity matrix
- The only parameter that can be tuned is the so-called bandwidth σ^2
- Aracne uses this non-parametric density estimator for each dimension k and pair of dimensions k and l
  - Notice that to process three variables X_i, X_j and X_k for removal of edges, MI needs to be evaluated only for pairs of variables, i.e., only 2-D numerical integrals are needed

Kernel density illustration

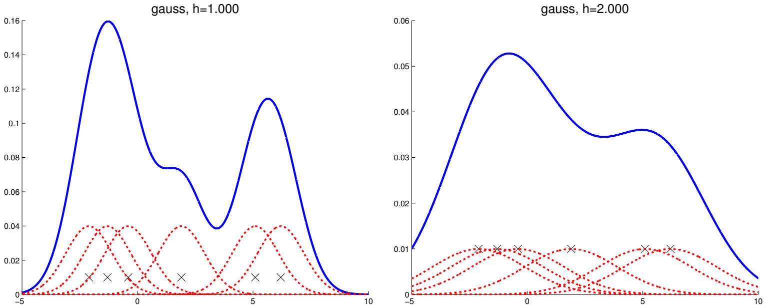


Figure: Illustration of Gaussian kernel density estimation from (Murphy, 2012)

Human B cell network: Aracne algorithm

- Data: 336 genome-wide expression profiles for perturbations of B cell phenotypes
- Focus on subnetwork around MYC gene
- Independent validation: MYC ChIP assay that measures binding of MYC protein on gene promoters
  - Provides direct experimental that MYC regulates a target gene

Human B cell network: Aracne algorithm

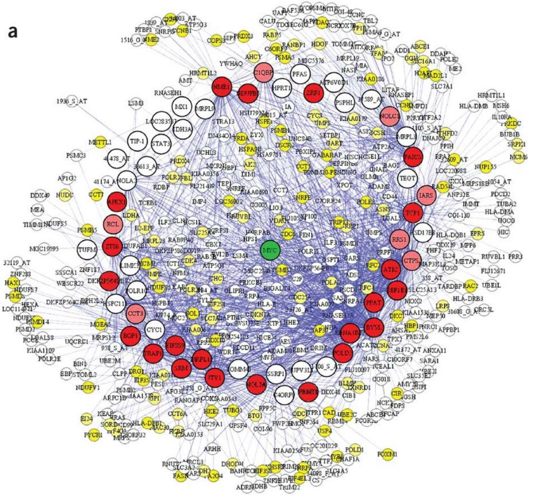


Figure: MYC subnetwork inferred by Aracne from B cell expression data (Basso et al., 2005)

Human B cell network: Aracne algorithm

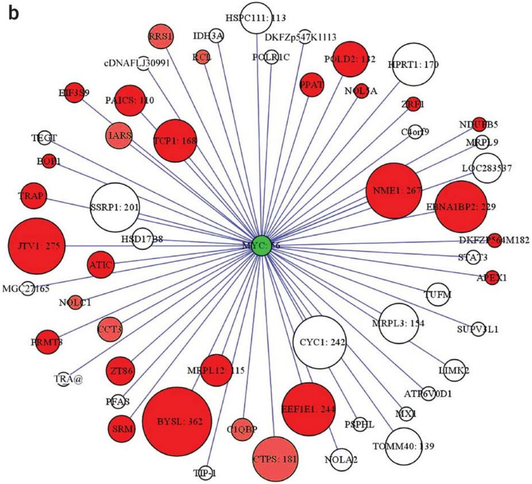


Figure: MYC subnetwork inferred by Aracne from B cell expression data (Basso et al., 2005)

References

- Aho, T., et al, Reconstruction and validation of RefRec: a global model for the yeast molecular interaction network, *PLoS ONE*, 5(5):e10662, 2010.
- Basso K, et al., Reverse engineering of regulatory networks in human B cells. *Nat Genet.* 2005, 37(4):382-90.
- Margolin AA et al., ARACNE: An Algorithm for the Reconstruction of Gene Regulatory Networks in a Mammalian Cellular Context, *BMC Bioinformatics*, 7(Suppl 1):S7, 2006.
- Murphy K (2012) Machine learning: a probabilistic perspective, MIT Press.
- Saez-Rodriguez, J., et al. (2007) A logical model provides insights into T cell receptor signaling. *PLoS Computational Biology*, 3(8): e163.