Biophysics
A Computational Approach
Concepts, Models, Methods and Algorithms
Lecture 6: Static and Temporal Networks

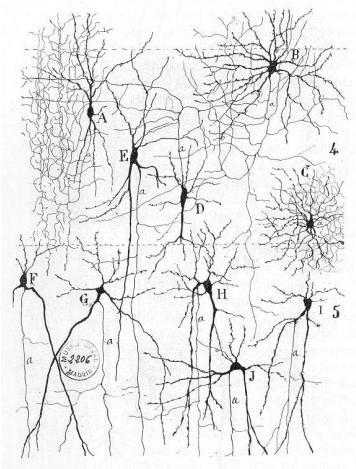
Dieter W. Heermann

Heidelberg University

December 20, 2016

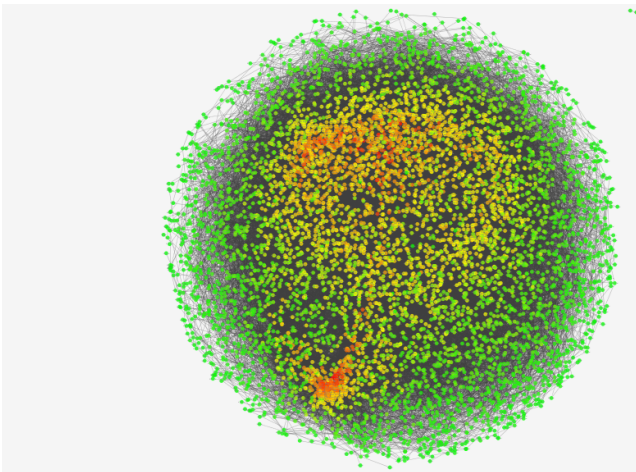
Table of Contents

Introduction I



Neural network

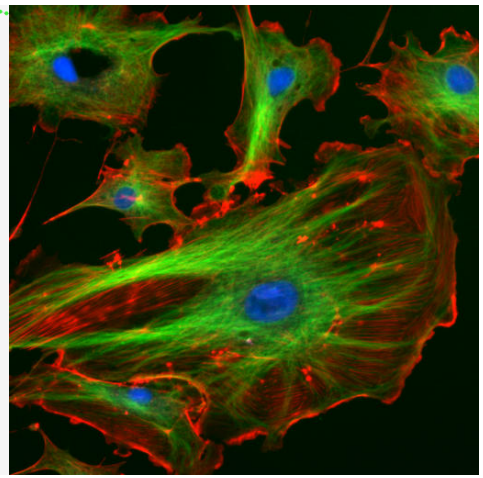
Source:https://en.wikipedia.org/wiki/Biological_neural_network



Gene Co-Expression network

Source: Wikipedia

https://en.wikipedia.org/wiki/Gene_co-expression_network



Cytoskeleton

Source: Wikipedia

https://de.wikipedia.org/wiki/Cytoskelett

Introduction II

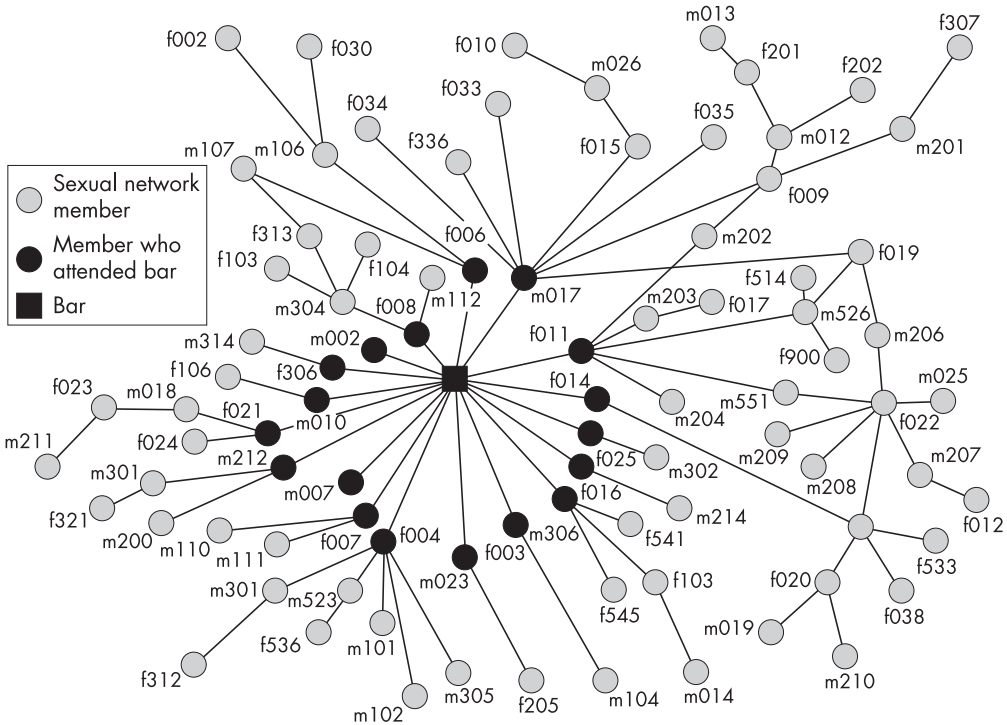
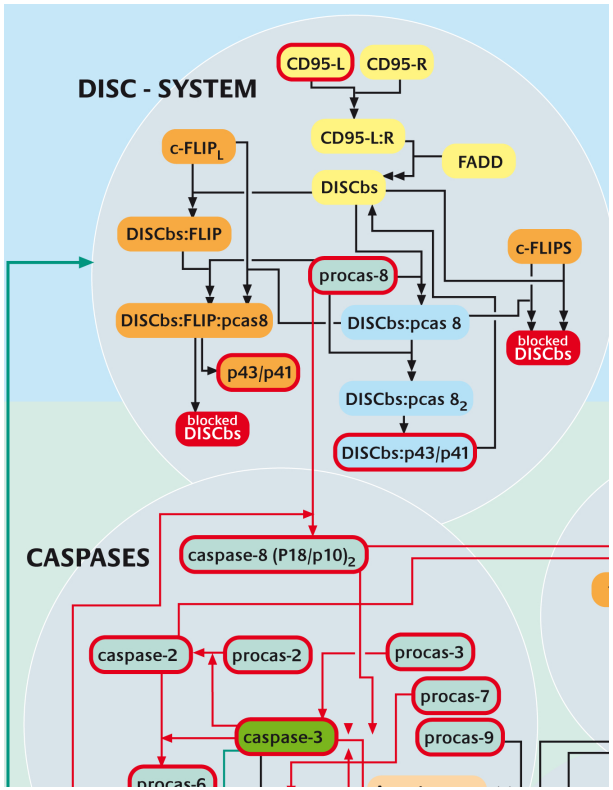


Figure 2 Network members (n = 89) viewed by their connection through a bar associated with gonorrhoea acquisition. A prefix to the unique identifier of "m" designates a male and "f" indicates a female sexual partner. Bar patrons possessed significantly higher information centrality measures compared to non-patrons (table 3).

Taken from: P De, A E Singh, T Wong, et al. doi: 10.1136/sti.2003.007187 2004 80: 280-285 Sex Transm Infect

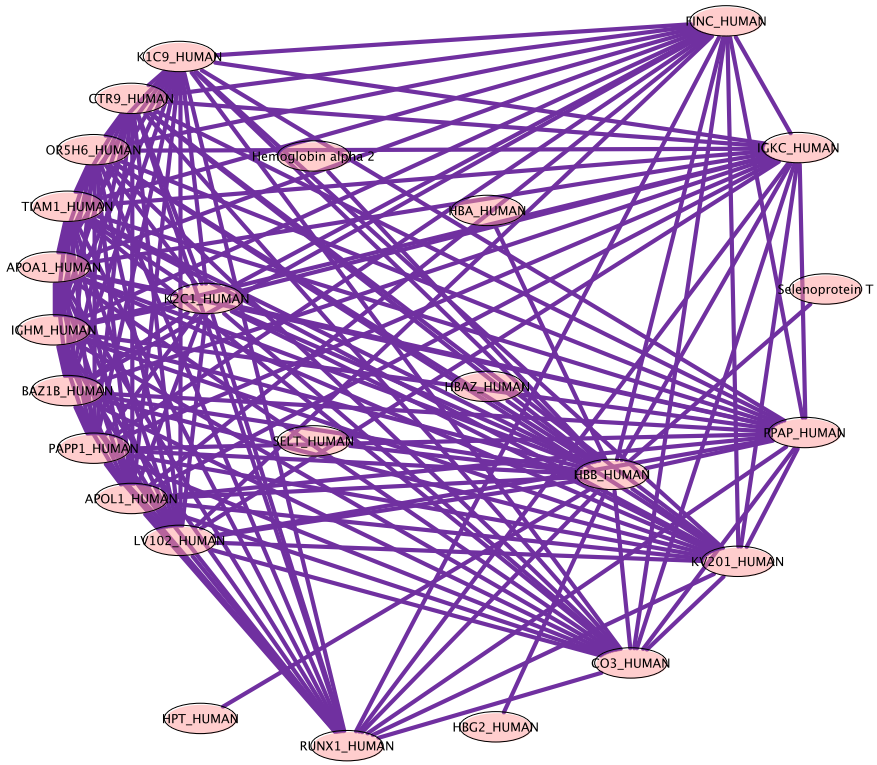
Sexual network analysis of a gonorrhoea outbreak

Apotosis

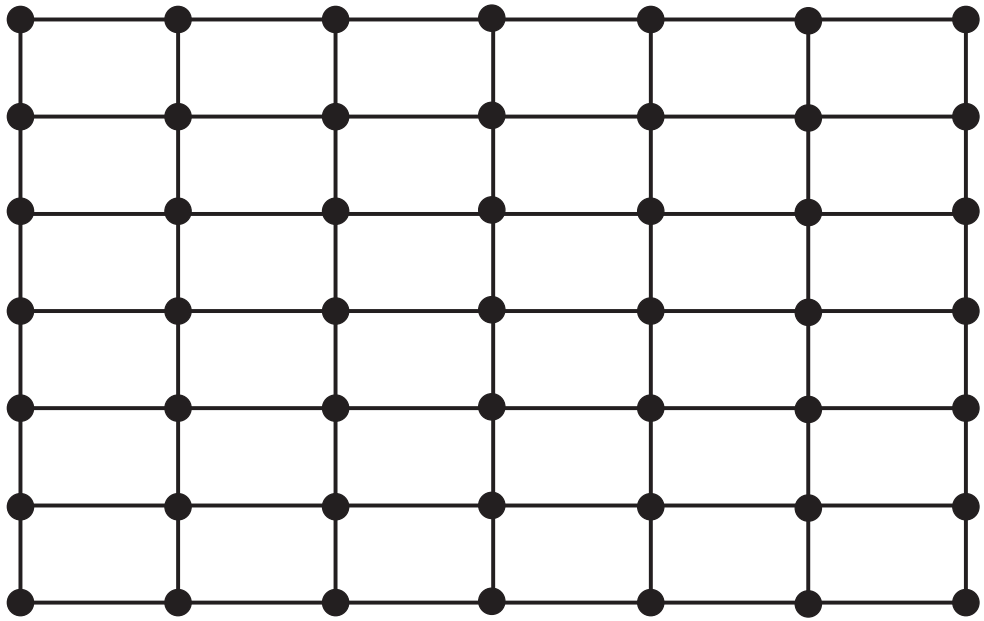


Taken from Bentele M, Lavrik I, Ulrich M, Stosser S, Heermann DW, Kalthoff H, Krammer PH, Eils R (2004). Mathematical modeling reveals threshold mechanism in CD95-induced apoptosis. J Cell Biol 166, 839-851

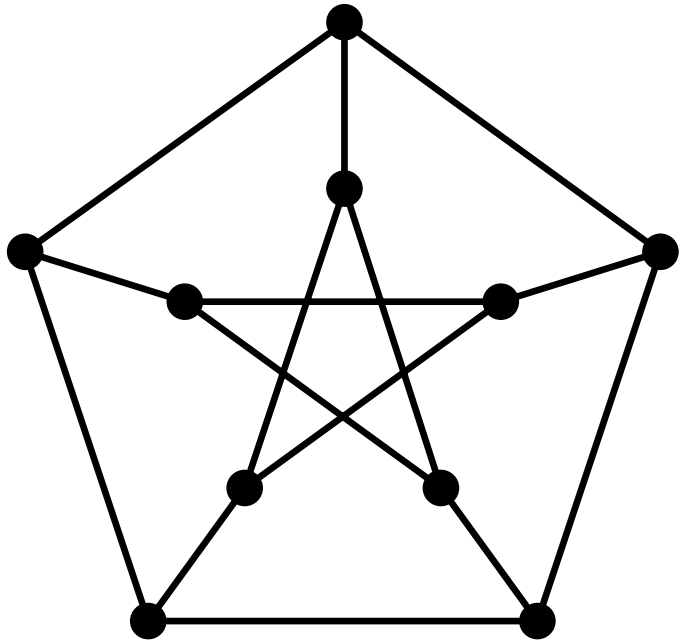
Gene Interaction Network



Graph Models: Lattices I

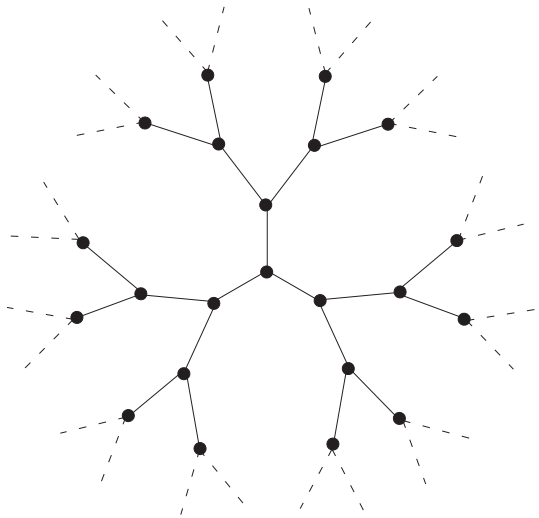


Graph Models: Lattices II

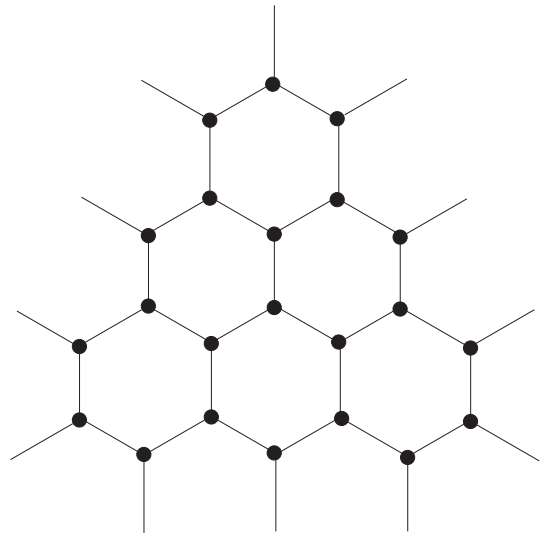


The Petersen is graph a where no pair of vertices are separated by more than two steps and the shortest circuit has length five.

Graph Models: Lattices III

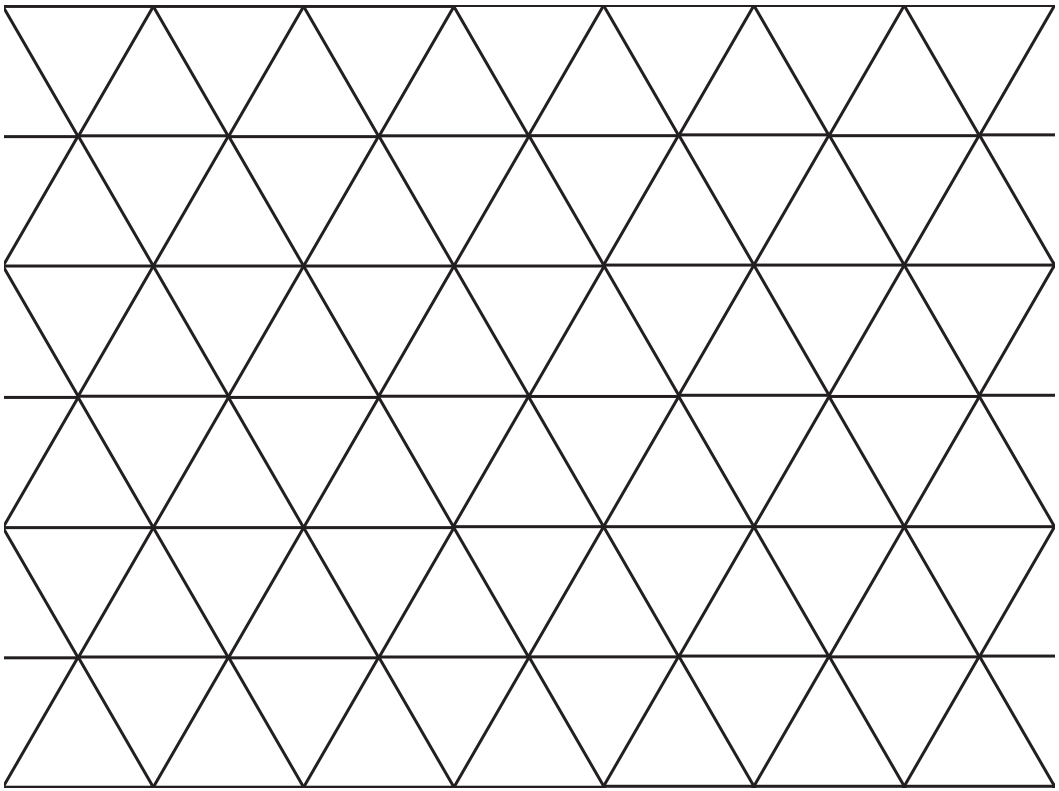


Cayley tree with degree 3



Hexagonal graph with degree 3

Graph Models: Lattices IV



Graph Models I

A **graph** consists of a nonempty set of **points** or **vertices**, and a set of **edges** that link together the vertices. A graph may be of two many forms: **directed** or **undirected**.
In a directed graph the direction of any given edge is defined.
An edge in a graph that joins two vertices is said to be **incident** to both vertices. The **degree** of a vertex is determined by the number of distinct edges that are incident to the vertex.
Two edges in a graph are **adjacent** if they connect to the same vertex.
A **loop** is an edge that links a vertex to itself.
A **path** through a graph is a traversal of consecutive vertices along a sequence of edges. A **cycle** is a path in which the initial vertex of the path is also the terminal vertex of the path.

- A graph $G_1 = (V_1, E_1)$ is a **subgraph** of $G = (V, E)$ if $V_1 \subset V$ and $E \subset E_1$.
- In general each edge may be associated with a direction and weight $w_i \in \mathbf{R}$.
- In a **directed graph** we attach a direction to each edge e_s^d. $e_s^d = (v_i, v_j)$ means that the edge e_s starts at node v_i and ends at node v_j.
- In an undirected graph the order in which nodes are written does not matter

$$e_s^n = (v_i, v_j) = (v_j, v_i) \tag{1}$$

- We allow for $v_i = v_j$. Such an edge is said to be a **one-edged loop** attached to v_i.
- It is possible to allow for more than one edge between nodes v_i and v_j.

- If a graph contains neither multiple edges between pairs of nodes nor loops, then the graph is called **simple**.
- For a simple graph the number of edges is at most

$$M^{\max} = \frac{N(N-1)}{2} \tag{2}$$

  In this case the graph is called **fully connected**.
- The degree d_i is number of edges incident to a node v_i. In a directed graph we distingiush between the **in-degree** d_i^{IN} and the **out-degree** d_i^{OUT}, ie., the number of nodes ending on or starting from node v_i.
- We define the **neighbourhood** $\Gamma(v_i)$ of node v_i through

$$\Gamma(v_i) := \{v_j | v_j \in V \wedge (v_i, v_j) \in E\} \tag{3}$$

  clearly the degree (in-degree) is also the size of the neighbourhood

$$d_i = |\Gamma(v_i)| \tag{4}$$

Graph Models III

- In all graphs we have

$$\sum_i d_i = 2M, \quad M = |E| \tag{5}$$

  For directed graphs $\sum_i d_i = M$
- The **average degree** $\bar{d}$ of a graph is defined as

$$\bar{d} := \frac{1}{N} \sum_{i=1}^{N} d_i \tag{6}$$

- For directed graphs we have

$$\frac{1}{N} \sum_{i=1}^{N} d_i^{\mathrm{IN}} = \frac{1}{N} \sum_{i=1}^{N} d_i^{\mathrm{OUT}} \tag{7}$$

- The **degree distribution** is defined by

$$P(k) := \frac{1}{N} \sum_{i=1}^{N} \delta_{d_i,k} \quad \text{for} k = 0, 1, 2, \ldots \tag{8}$$

  The degree distribution summarizes information about the local environment of a graph
- $P^{\mathrm{IN}(k)}$ can be different from $P^{\mathrm{OUT}(k)}$

Labeled (Coloured) Graphs I

Biological graphs are generally labeled with information. To each node we have an associated vector of properties v_i
(Potts model)

Paths, Components and Trees I

- A **path** from node v_i to v_j is a sequence of edges which can be traversed to reach v_j starting from v_i.
- In a directed graph paths cannot go against the direction of an edge.
- Node v_j is **connect** to node v_i if there is a path from node v_i to node v_j. In an undirected graph, if there is a path from node v_i to node v_j, then there is also a path from v_j to v_i.
- The **adjacency matrix** corresponding to the graph G is an NxN matrix A whose entries a_{ij} are 0 if node i is not connect to j by an edge and 1 otherwise.
- For a weighted graph we generalize this such that the entries with an edge have a real number assigned.
- If there is a path starting from and ending on node $v_i \in V$, then this is called a **loop**.
- A set of k nodes $C = \{v_1, ..., v_k\}$ where each node in C can be reached from other nodes in C but not from any node outside of C is called a **connected component of size** k.
- For a simple graph the number of connected components K is given by

$$K \geq N - M \tag{9}$$

- If there is more than one path between a pair of nodes $v_i, v_j \in V$, then the graph contains **closed paths** or loops.

Paths, Components and Trees II

- In an undirected simple graph, if there is precisely one path between each pair of nodes $v_i, v_j \in V$, then there cannot be any loop and the graph is called a **tree**.
- If a graph consist of several components, each of which is a tree, the graph os called a **forrest**.
- A **spanning tree** T of a connected graph with nodes $V_T = V_G$ and edges $E_T \subseteq E_G$ is a graph such that (V_T, E_T) is a tree.

Distances and Diameter I

- If two nodes are connected by a sequence of nodes and edges, then the **distance** l_{ij} between them is defined as the number of edges that have to be traversed to reach node v_j from v_i

$$l_{ij} := \min\{x_{ij} | x_{ij} \text{ is the length of a path from } v_i \text{ to } v_j \text{ along } e_s \in E\} \tag{10}$$

- If there is no path, then we set $l_{ij} = \infty$
- In general $l_{ij} \neq l_{ji}$
- The **diameter** of a graph is defined as the maximum distance between two nodes in the graph

$$D = \max\{l_{ij} | v_i, v_j \in V\} \tag{11}$$

- For a disconnected graph we have $D = \infty$. This can be made finite by defining D to be the diameter of the largest connected component.
- The **clustering coefficient** measures the probability that two nodes v_j and v_k, which are both neighbours of $v - i$ $((v_i, v_j), (v_i, v_k \in E)$ are themselves connect by an edge $(v_j, v_k) \in E$

$$c_i := \frac{2\eta_i}{d_i(d_i - 1)} \quad \text{for} d_i \geq 2 \tag{12}$$

where η_i is the number of edges among the nodes connected to v_i.

Distances and Diameter II

- The **average path length** is defined as

$$\bar{l} := \frac{2}{N(N-1)} \sum_{i=1}^{N} \sum_{j=1}^{N} l_{ij} \tag{13}$$

  $l_{ii} = 0$ by definition.

- The **distribution of graph distances** is defined as

$$\lambda(l) := \frac{2}{N(N-1)} \sum_{i=1}^{N} \sum_{j=1}^{N} \delta_{l_{ij},l} \quad l = 1, 2, ... \tag{14}$$

Random Graphs: Erdös-Renyi-Model I

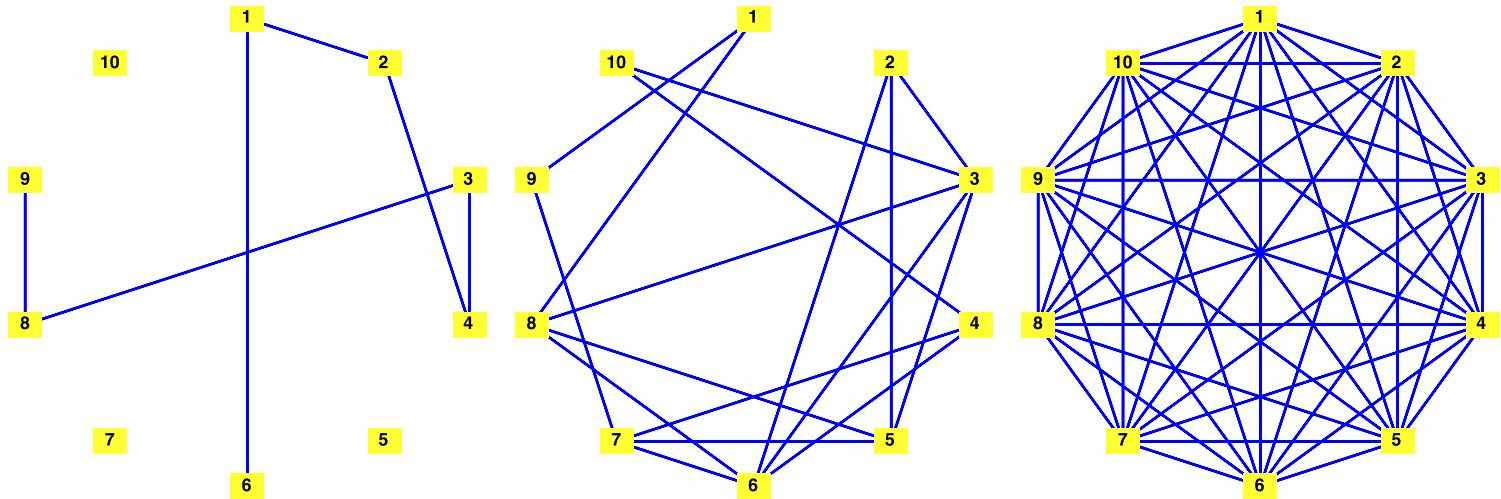


Figure: Example: Erdös-Renyi Model, A $p = 0.1$, B $p = 0.3$, C $p = 1$

- Graphical display of dependence structure between multiple interacting quantities (expression levels of different genes).
- Probabilistic semantics: Fits the stochastic nature of both the biological processes and noisy experiments. Capable of handling noise and estimating the confidence in the different features of the network.
- Due to lack of data: Extract features that are pronounced in the data rather than a single model that explains the data.

Random Graphs: Erdös-Renyi-Model II

- Random variable X_i = measured expression level of gene i represented by nodes.
- Edges = regulatory interactions between genes.

- Define the functional form of the conditional distributions (e.g. multinomial for discrete variables, linear Gaussian for continuous variables).
  - Find the best network structure S
  - Given a network structure, find the best set of parameters for the conditional distributions (the most probable structure/parameter vector given the data)

Random Graphs: Erdös-Renyi-Model III

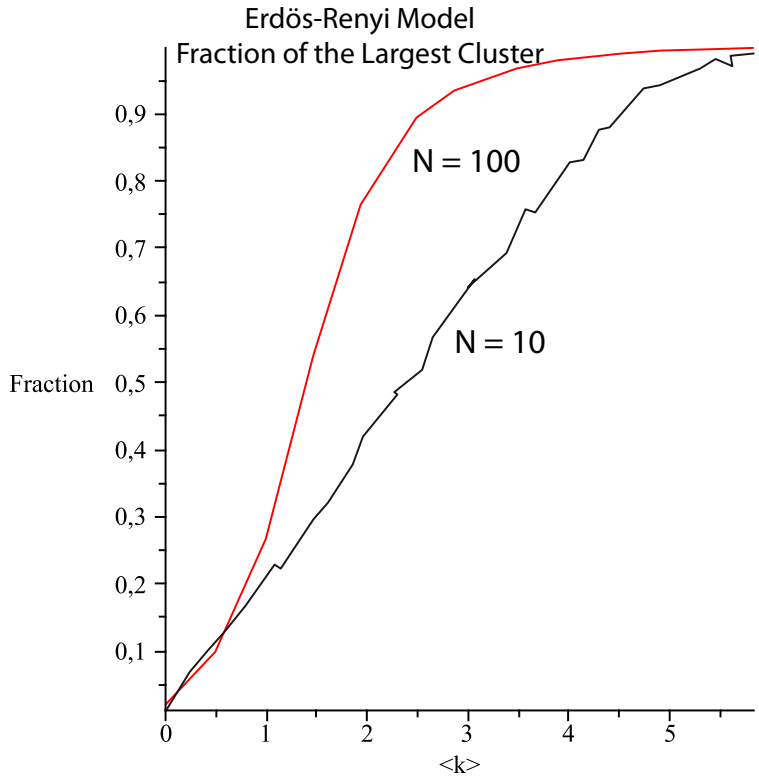


Figure: Example: Erdös-Renyi Model, A $p = 0.1$, B $p = 0.3$, C $p = 1$

Erdös-Rényi Model

Let G be a graph with N nodes. Any two nodes have an edge with probability p. Each of the $(N-1)N/2$ edges appears independently with probability p.

Random Graphs: Erdös-Renyi-Model IV

- The total number of edges is a random variable with expectation value

$$< n >= p[N(N-1)/2]$$

- If G_0 is a graph with nodes $V_1, V_2, ..., V_N$ and n edges the probability of realizing this graph is

$$P(G_0) = p^n(1-p)^{N(N-1)/2-n}$$

A graph $G_1 = (V_1, E_1)$ is a **subgraph** of $G = (V, E)$ if $V_1 \subseteq V$ and $E_1 \subseteq E$
Examples

- **cycles** are closed loops of k edges such that every two consecutive edges and only those have a common node
- **complete subgraphs** (of order k) contains k nodes and all of the possible $k(k-1)/2$ edges

Question: Is there a critical probability that marks the appearance of arbitraray subgraphs consisting of k nodes l edges?
In a random graph with connection probability p the degree k_i of node i follows a binomial distribution with parameters $N-1$ and p

$$P(k_i = k) = C^k_{N-1} p^k (1-p)^{N-1-k} \tag{15}$$

There are C^k_{N-1} equivalent ways of selecting the k end points.

Random Graphs: Erdös-Renyi-Model V

- For $i \neq j$ two different nodes, $P(k_i = k)$ and $P(k_j = k)$ are close to being to independent random variables

Let X_k be the number of nodes with degree k. Our goal is to determine the probability that X_k takes on a given value

$$P(X_k = r) \tag{16}$$

$$< X_k >= NP(k_i = k) = \lambda_k = NC_{N-1}^k p^k (1-p)^{N-1-k} \tag{17}$$

Assume $P(X_k = r)$ approaches a Poisson distribution

$$P(X_k = r) = e^{\lambda_k} \frac{\lambda_k^r}{r!} \tag{18}$$

Thus the degree distribution of a random graph is a binomial distribution

$$P(k) = C_{N-1}^k p^k (1-p)^{N-1-k} \tag{19}$$

For large N this can be replaced by a Poisson distribution

$$P(k) = e - pN \frac{(pN)^k}{k!} = e^{<k>} \frac{< k >^k}{k!} \tag{20}$$

Let $G = (V, E)$ be a directed acyclic graph. We assume that the vertices $i \in V$ $(1 \leq i \leq n)$ represent for example genes and correspond to random variables x_i. For each y_i we define a conditioned probability

$$P(x_i|\text{parent}(x_i)) = P(x_i|P_a(x_i)) \tag{21}$$

and the joint probability distribution

$$P(x_1, ..., x_n) = \prod_{i=1}^{n} P(x_i|P_a(x_i)) \tag{22}$$

We define a parameter θ to be the set of conditioned probabilities
From the graph shown in figure 3 representing the Bayesian network we can read off the joint probability distribution

$$P(x_1, ..., x_n) = P(x_1)P(x_2)P(x_3|x_1, x_2)P(x_4|x_2)P(x_5|x_3, x_4) \tag{23}$$

and θ is the set of local conditional probabilities.

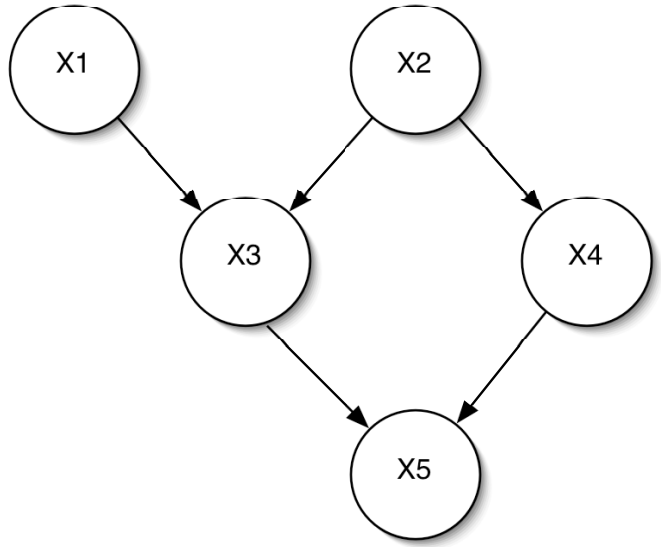


Figure: A simple Bayesian network

Let x be distributed as $x \sim f_\theta(x)$ with parameter θ. Our aim is to estimate the parameter θ, given that there are iid observations $\{x_i\}$ of the random variable x.
Example:
Let $\{x_i\}$ be a sequence of results of rolling a dice such that $x_i = 1$ denotes head and $x_i = 0$ tail. The dice is unfair so that we have for the probability that the random variable takes on the result $x_i = 1$ is $P(x_i = 1) \neq 1/2$. Define the parameter $\theta = P(x_i = 1)$. We need an estimate for θ. Of course, we know that we are dealing with a binomial distribution (see section **??**) and

$$P(x = 1) = p, \quad P(x = 0) = q, \quad q = 1 - p \tag{24}$$

How many successful realizations k do we have after we have tried n-times? This is given by the binomial distribution (recall that the reverse question gives the negative binomial distribution, see appendix).
Without any prior information we can use the maximum likelihood method to estimate the parameter θ. Let $x_1, ..., x_n$ denote the observations, with $H + T = n$ where H is the number of trails we have obtained head, and T the number of trails we have obtained tail. Then

$$\begin{aligned} \hat{\theta} &= \text{argmax} f_\theta(y_i) \quad i = 1, \ldots n && (25) \\ &= \text{argmax} \theta^H (1-\theta)^T && (26) \\ &= \frac{H}{H+T} && (27) \end{aligned}$$

which corresponds to the empirical frequency of the sample for the event H. Even if we have prior information in the sense that we have a previous set of experiments where the number of heads was 5 and the number of tails was 10, we would obtain

$$\hat{\theta} = \frac{5+H}{5+H+10+T} \tag{28}$$

which is irrelevant if the number of total experiments $n = T + H$ goes to infinity. But now suppose we are not sure that we measured 5 heads and 10 tails and can only tell that presumably the probability with which we obtained head was $1/4$. For this uncertainty we assume a prior distribution for *theta* $P(\theta)$

$$P(\theta) = \beta(5, 10) = \frac{\Gamma(5)\Gamma(10)}{\Gamma(5+10)} \tag{29}$$

where we have assumed that the prior is distributed as a *beta distribution*

$$\beta(p,q) \sim \frac{\Gamma(5)\Gamma(10)}{\Gamma(5+10)} = \int_0^1 x^{p-1}(1-x)^{q-1}dx \tag{30}$$

Assume that we make new observations. We use the Bayesian law to compute the posterior distribution

$$P(\theta|x) = \frac{P(x|\theta)P(\theta)}{P(x)} \tag{31}$$

$$= \frac{P(x|\theta)P(\theta)}{\int P(x|\theta)P(\theta)d\theta} \tag{32}$$

where $P(x|\theta)$ is the likelihood function. In the new experiment we find 50 heads and 50 tails so that we obtain for the likelihood function

$$P(x|\theta) = \theta^{50}(1-\theta)^{50} \tag{33}$$

from which we obtain the posterior distribution

$$P(\theta|x) = \frac{\theta^{59}(1-\theta)^{69}}{\int \theta^{59}(1-\theta)^{69}d\theta} \tag{34}$$

Note that the posterior function is the same as the prior function. In such a case we call the distribution conjugate. Thus the beta distribution is the conjugate to the binomial distribution.
The result of a Bayesian analysis is a posterior distribution and not a single value!
This distribution can be used to make predictions.

Graphic representation of a joint distribution over a set of random variables A, B, C, D, E.

$$P(A, B, C, D, E) = P(A) * P(B) * P(C|A) * P(D|A, B) * P(E|D)$$

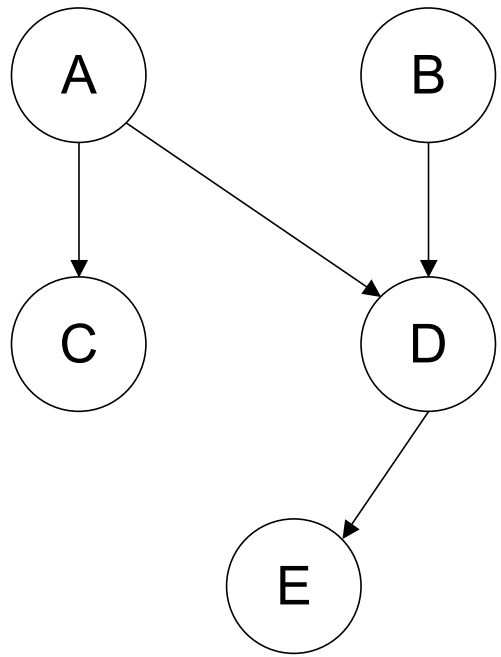


Figure: Nodes represent gene expression while edges encode the interactions (cf. inhibition, activation)

Bayesian Networks for Gene Expression Data II

- Given a set of random variables $X = (X_1, ..., X_n)$, a Bayesian network is defined as a pair $BN = (S, \theta)$, where
  - S is a directed acyclic graph (DAG), which is a graphical representation of the conditional independencies between variables in X
  - θ is the set of parameters for the conditional probability distributions of these variables.
  - In a Bayesian network, the probability of a state $x = (x_1, x_2, ..., x_n)$ is factored as

$$P(x) = P(x_1|pa(x_1))P(x_2|pa(x_2))...P(x_n|pa(x_n)),$$

    where pa(x) denotes the parents of node x in the graph S

- A Bayesian network should be a DAG (Direct Acyclic Graph).
- Random variable X_i = measured expression level of gene i. Arcs = regulatory interactions between genes.
- However, there are lots regulatory networks having directed cycles.
- Solve this by expanding into the time direction

Bayesian Networks for Gene Expression Data III

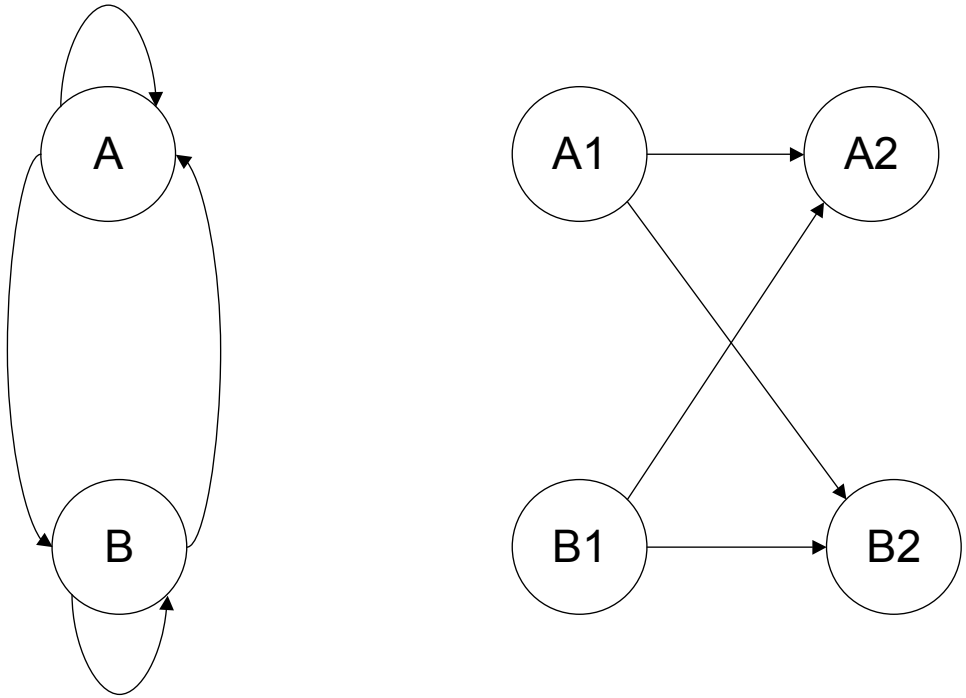


Figure: Use DBN (Dynamic Bayesian Networks: BN with constraints on parents and children nodes) for sequential gene expression data

Bayesian Networks for Gene Expression Data IV

We are looking for a Bayesian network that is most probable given the data D (gene expression)

$$BN^* = \text{argmax}_{\mathbf{BN}}\{P(BN|D)\}$$

where

$$P(BN|D) = \frac{P(D|BN)P(BN)}{P(D)}$$

There are many networks. An exhaustive search and scoring approach for the different models will not work in practice (the number of networks increases super-exponentially, $O(2^{n^{\mathbf{2}}})$ for dynamic Bayesian networks)
Idea: Sample the networks such that we eventually have sampled the most probable networks
Monte Carlo

- Recall detailed balance condition for Monte Carlo

$$P(BN_{old}|D)P(BN_{old} \rightarrow BN_{new}|D) = P(BN_{new}|D)P(BN_{new} \rightarrow BN_{old}|D)$$

- Let us look at

$$P(BN|D) = \frac{P(D|BN)P(BN)}{P(D)}$$

- Assume $P(BN)$ is uniformly distributed *(We could incorporate knowledge)*
- Choose next BN with probability $P(BN_{\mathbf{new}})$

- Accept the new BN with the following Metropolis-Hastings accept/rejection criterion:

$$
\begin{aligned}
P &= \min\left\{1, \frac{P(BN_{\mathbf{new}}|D)P(BN_{new} \rightarrow BN_{old}|D)}{P(BN_{\mathbf{old}}|D)P(BN_{old} \rightarrow BN_{new}|D)}\right\} \\
&= \min\left\{1, \frac{P(D|BN_{\mathbf{new}})P(BN_{\mathbf{new}})P(D)}{P(D|BN_{\mathbf{old}})P(BN_{\mathbf{old}})P(D)}\right\} \\
&= \min\left\{1, \frac{P(D|BN_{\mathbf{new}})P(BN_{\mathbf{new}})}{P(D|BN_{\mathbf{old}})P(BN_{\mathbf{old}})}\right\} \\
&= \min\left\{1, \frac{P(D|BN_{\mathbf{new}})}{P(D|BN_{\mathbf{old}})}\right\}
\end{aligned}
$$

Discrete model

- Even though the amount of mRNA or protein levels, for example, can vary in a scale that is most conveniently modeled as continuous, we can still model the system by assuming that it operates with functionally discrete states
  - **activated** / **not activated** (2 states)
  - **under expressed** / **normal** / **over expressed** (3 states)
- Discretization of data values can be used to compromise between the
  - averaging out of noise
  - accuracy of the model
  - complexity/accuracy of the model/parameter learning

Bayesian Networks for Gene Expression Data VI

- Qualitative models can be learned even when the quality of the data is not sufficient for more accurate model classes
- Let N_{ijk} be the number of times we observe variable/node i in state k given parent node configuration j
- Summarize the number of total number of observations for variable i with parent node configuration j,

$$N_{ij} = \sum_{k=\mathbf{1}}^{r_i} N_{ijk}$$

- Since our states are discrete we use a multinomial distribution
- the ML estimate of multinomial probabilities is obtained by the normalized counts

$$\hat{\theta}_{ijk} = \frac{N_{ijk}}{N_i j}$$

- A convenient prior distribution to choose for the parameters θ is given by the Dirichlet distribution

$$(\theta_{ij\mathbf{1}}, \ldots, \theta_{ijr_i}) \sim \text{Dirichlet}(\alpha_{ij\mathbf{1}}, \ldots, \alpha_{ijr_i})$$

- The Dirichlet distribution has PDF

$$f(\theta_{ij\mathbf{1}}, \ldots \theta_{ijr_i}; \alpha_{ij\mathbf{1}}, \ldots \alpha_{ijr_i}) = \frac{1}{B(\alpha_{ij})} \prod_{i=\mathbf{1}}^{r_i} \theta_{ijr_i}^{\alpha_{ijr_i} - \mathbf{1}}$$

  with $\theta_{ijr_i} \geq 0, \sum_i \theta_{ijr_i} = 1$ and hyperparameters $\alpha_{ijr_i} \geq 0, \alpha_{ij} = \sum_k \alpha_{ijr_i}$

- The normalization constant, the Beta function, can be expressed using the gamma function

$$B(\alpha_{ij}) = \frac{\prod_{k=\mathbf{1}}^{r_i} \Gamma(\alpha_{ijr_i})}{\Gamma(\alpha_{ij})}$$

- The convenience arises from the fact that the distribution is conjugate to the multinomial distribution, i.e., if $P(\theta)$ is Dirichlet and $P(X|\theta)$ is multinomial, then $P(\theta|X)$ is Dirichlet as well
- The multinomial distribution is given (for $N_{ij} = \sum_k N_{ijk}$) by

$$f(N_{ij\mathbf{1}}, \ldots, N_{ijr_i} | N_{ij}, \theta_{ij\mathbf{1}}, \ldots, \theta_{ijr_i}) = \frac{N_{ij}!}{N_{ij\mathbf{1}}! \cdots N_{ijr_i}!} \theta_{ij\mathbf{1}}^{N_{ij\mathbf{1}}} \cdots \theta_{ijr_i}^{N_{ijr_i}}$$

  and is the distribution of observations in r_i classes if N_{ij} observations are selected as outcomes of independent selection from the classes with probabilities $\theta_{ijk}, k = 1, \ldots r_i$

Structural Properties

- In order to get reliable results we can focus on features that can be inferred

- for example, we can define a feature, an indicator variable f with value 1 if and only if the structure of the model contains a path between nodes A and B
- Looking at a set of models S with a good fit we can approximate the posterior probability of feature f by

$$P(f|D) = \sum_S f(S)P(S|D)$$

- With gene regulatory networks, one can look for only the most significant edges based on the scoring
- A Markov chain is defined over Bayesian nets so that it approaches a steady-state distribution as it is being run, and the probabilities of the states (networks) correspond to their posterior probability
- Individual nets are created as states in the chain and after (assumed) convergence, samples S_i are taken
- Posterior probability of an edge can then be approximated with

$$P(f(S)|D) \approx \frac{1}{n} \sum_{i=\mathbf{1}}^{n} f(S_i)$$

Bayesian Networks for Gene Expression Data IX

- To work out the Monte Carlo Method to generate networks we first have to compute $P(D|S)$

$$\begin{aligned} P(D|S) &= \int_{\theta} P(D|\theta, S)P(\theta|S)d\theta \\ &= \dots \\ &= \prod_{i=\mathbf{1}}^{n}\prod_{j=\mathbf{1}}^{q_i}\frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij}+N_{ij})}\prod_{k=\mathbf{1}}^{r_i}\frac{\Gamma(\alpha_{ijk}+N_{ijk})}{\Gamma(\alpha_{ijk})} \end{aligned}$$

- Monte Carlo moves: ADD, REMOVE, REVERSE edge in network

Algorithm 1 Gene Expression Network

1. TO BE DONE

Weight Matrix Model I

A **weight matrix model** considers the interactions between all combinations of genes. A weight matrix [1] consists of $n \times n$ weight values, each of which indicates the influence of one specific gene on another. A positive value for $w_{i,j}$ models gene j stimulating the expression of gene i. A negative value models repression, while a value of zero indicates that gene j does not influence the transcription of gene i.
The expression state of a transcriptional regulatory network containing n genes (with discrete states) is represented by a vector $\mathbf{u}(t)$. The net regulatory effect, of gene j on gene i is the expression level of j, i.e., $u_j(t)$, times its regulatory influence on i which is determined by the weight $w_{i,j}$. The total regulatory input to i, i.e. $r_i(t)$, is derived by summing over all the genes in the system

$$r(i) = \sum_j w_{ij} u_j(t) \tag{35}$$

The response of each gene to the regulatory input in this model is calculated with a dose-response function

$$\frac{x_i(t+1) = 1}{1 + e^{-(\alpha_i r_i(t) + \beta_i)}} \tag{36}$$

where α_i and β_i are two gene specific constants that define the shape of the dose-response curve for gene i. This assumes that each gene has a static dose-dependent response to activating and repressing regulatory influences.
The constants can be incorporated into the weight matrix, replacing W with Z

Weight Matrix Model II

$$z_{ij} = \alpha_i w_{ij} \tag{37}$$

We further define a column weight $z_{i0} = \beta_i$ and $u_0(t) = 1$. Thus the vector of net regulations ($\mathbf{r}(t)$) now is $\mathbf{s}(t)$

$$s_i(t) = \sum_j u_j(t) z_{ij} \tag{38}$$

and

$$\frac{x_i(t+1) = 1}{1 + e^{-s_i(t)}} \tag{39}$$

yielding

$$\frac{u_i(t+1) = m_i x(t) = m_i}{1 + e^{-\sum_j z_{ij} u_j(t)}} \tag{40}$$

here m_i the maximal expression level for gene i used to get the real expression output for i, i.e. $u_i(t+1)$.

Random Boolean Networks I

A Boolean network is one of the simplest model for the behaviour of genomic network. Such network consists of n nodes (e.g. representing genes) which can either be repressed or expressed (the node has state 0 or 1, respectively). The dynamics of the network is determined by a list of n (Boolean) functions which each receive input from k specified nodes. Every node has its own specific function, which can determine its next state from the current states of all the input nodes.
A *Boolean network* [2, 3] $G(V, F)$ is thus defined by a set of nodes $V = \{x_1, ..., x_N\}$ and a list of Boolean functions $F = (f_1, ..., f_k)$ corresponding to the edges. In these models, gene expression is quantized to just two levels: ON and OFF (1 or 0). The state of a node (gene) is completely determined by the values of other nodes at time t by means of underlying logical Boolean functions. The model is represented in the form of directed graphs. Each x_i represents the state (expression) of gene i, where $x_i = 1$ represents the fact that gene i is expressed and $x_i = 0$ means it is not expressed. The time evolution of the network is described by

$$x(t+1) = f_i(x_{j_1}(t), ..., x_{j_{k_i}}(t)) \tag{41}$$

where k_i denotes the connectivity of the node i. For the average connectivity we have

$$\langle K \rangle >= \frac{1}{N} \sum_{i=1}^{N} k_i \quad . \tag{42}$$

The list of Boolean functions F represents the rules of regulatory interactions between genes. The tuple (G, F) is one realization. In the *quenched model* one realizations is chosen at kept fixed for all times. In the *annealed model* a new realization is chosen at random at very time step.
Any given gene transforms its inputs (regulatory factors that bind to it) into an output, which is the state or expression of the gene itself. All genes are assumed to update synchronously or asynchronously in accordance with the functions assigned to them and this process is then repeated. Despite the simplicity of the model, they can capture a number of essential features of real genomic networks.

Random Boolean Networks III

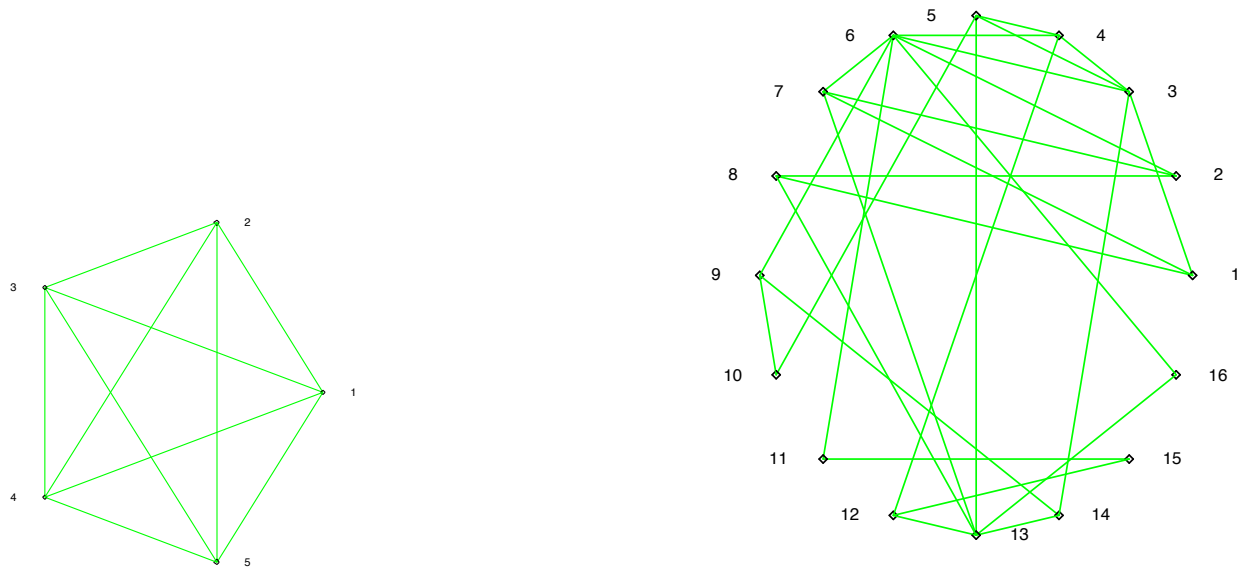


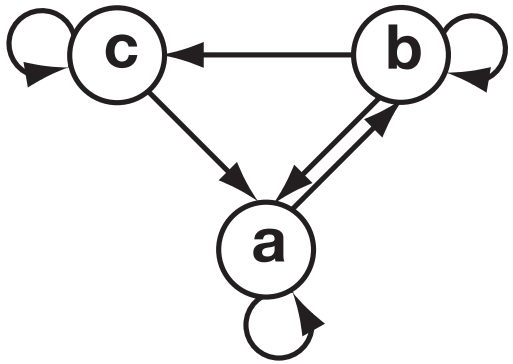
Figure: A random network with 5 nodes and 10 edges and one with 16 nodes and 28 edges

Random Boolean Networks IV

Asynchronous random boolean networks (ARBN) incorporate all the cases in which at each time point a single node is selected in order to be updated. The node to be updated can be chosen at random or according to a deterministic rule based:

- clock scheme [4, 5],
- cyclic scheme [6],
- random independent scheme [7], and
- random order scheme [7]

Consider the example shown in figure 7. The assignment of values to nodes made in the truth table fully describes the state of the model at any given time. The change of model state over time is fully defined by the functions in F. If we assign initial values to the nodes, all further states are determined. Consider the time evolution of the states as a trajectory. Since the number of possible states is finite, all trajectories eventually end up in single *steady states*, or a *cycle of steady states*. In this context we define an *attractor* of a trajectory as a single steady state, or a cycle at the end of the trajectory. The *basin of attraction* for a specific attractor is a set of all trajectories leading to it. States in gene networks are often characterized by stability in the sense that small changes in value of a few nodes do not change the attractor.



t			t+1		
a	b	c	a	b	c
0	0	0	0	1	1
0	0	1	1	0	1
0	1	0	1	1	1
0	1	1	0	1	1
1	0	0	1	1	1
1	0	1	1	1	1
1	1	0	1	0	1
1	1	1	1	1	0

Figure: A sample boolean network

At each time point t the system can be in one of the 2^N possible states. We have 2^{2^k} possible logic functions per node which yield

$$\text{possible} \quad \text{nets} = N_F = \left(\frac{2^{2^k} N!}{(N-k)!} \right)^N \tag{43}$$

The geometry of the network strongly influences the behaviour. We can distinguish between

- irregular networks

- regular networks like lattices where for example we have $|x_{j_1}(t), ..., x_{j_{k_i}(t)}| = 2d$ neighbours on a simple hypercubic lattice

From the statistical mechanics point of view we have to specify an ensemble of the coupling functions, i.e. each must be assigned a weight. For this we can consider the following options

- equal weight ensemble
- magnetization biased
- forcing functions
- additive functions

In the equal weight ensemble we assign every coupling function the same weight $1/N_F$.
In the magnetization bias situation the probability of occurrence of a coupling function is p if the result is 0 and $1-p$ if the result is 1.
In the forcing function ensemble the value of the function is determined if one of the arguments $m \in \{1, ..., k\}$ takes on a predetermined value. For example $x_m = 0$. The value of the function is not determined if $x_m = 1$.
And for the additive function ensemble we have

$$x_i(t+1) = \Theta(f_i(t)) \tag{44}$$

and

$$f(t) = h + \sum_{j=1}^{N} c_{ij} x_j(t) \tag{45}$$

where h is a bias.
Now let us look at the response of the system if we change conditions. Let

$$\Sigma_0 = \{x_1(0), ..., x_N(0)\} \tag{46}$$

denote the starting state and

$$\hat{\Sigma_0} = \{\hat{x_1}(0), ..., \hat{x_N}(0)\} \tag{47}$$

another starting state which differs from Σ_0 only in a few nodes. To describe the difference we introduce the *Hamming distance*

$$D(t) = \sum_{i=1}^{N} (x_i(t) - \hat{x}(t))^2 \tag{48}$$

to describe the difference in the evolution of the network, given two initial conditions. Clearly, if the network is such that any local discrepancy remains localized, then we do expect the Hamming distance to remain finite in the thermodynamic limit. If, on the other hand the discrepancy (or damage) can be propagated to almost every vertex, then we expect the Hamming distance to diverge.
Assume that we are dealing with the uniform distribution for the coupling functions. On average, a change of a single vertex will affect the argument of k functions. Hence $kD(0)$ functions are affected. Each of these is affected with a probability $1/2$ so that $D(1) = kD(0)$ and in general

$$D(t) = \left(\frac{k}{2}\right)^t D(0) \tag{49}$$

We can thus distinguish three phases

- chaotic phase for $k_c > 2$
- frozen for $k_c < 2$

- critical for $k_c = 2$

In the last case the state of the system will be dominated by fluctuations.
Let us now look at the case where

$$f_i = \begin{cases} 0 & \text{with probability } p \\ 1 & \text{with probability } 1-p \end{cases} \tag{50}$$

For a given p and a given connectivity k we will have critical values

$$k_c(p) \quad \text{and} \quad p_c(k) \tag{51}$$

Let

$$a(t) \equiv 1 - D(t)/N \tag{52}$$

be the probability that two vertices have the same value in Σ_t and $\hat{\Sigma}$. The probability that the arguments of the functions f_i have the same value is given by

$$\rho_k = [a(t)]^k \tag{53}$$

The overlap is the same in the next time step if the arguments of the coupling functions are identical. This occurs with probability ρ_k. The overlap is also the same if the arguments of the coupling functions are also different, which occurs with

probability $1-\rho_k$ but the values are nevertheless the same, which occurs with probability $2p(1-p)$. Altogether we have

$$a(t+1) = 1-(1-\rho_k)2p(1-p) \quad (54)$$

$$= 1-\frac{1-[a(t)]^k}{k_c} \quad (55)$$

where we have used

$$k_c = \frac{1}{2p(1-p)} \quad (56)$$

$$p_c = \frac{1}{2} \pm \sqrt{\frac{1}{4}-\frac{1}{k}} \quad (57)$$

Clearly the recursion relation 55 has a trivial fixpoint given by

$$a^* = 1 \quad (58)$$

What kind of fixpoint is this? To answer this question we perturb the fixpoint slightly with δa_t. Then coming from below we have

$$1 - \delta a_{t+1} = 1 - \frac{1 - [1 - \delta a_t]^k}{k_c} \tag{59}$$

yielding

$$\delta a_{t+1} \approx \frac{k}{k_c} \delta a_t \tag{60}$$

The trivial fixpoint is unstable if $k/k_c > 1$. The resulting phase diagram can be seen in figure 8.

Random Boolean Networks XII

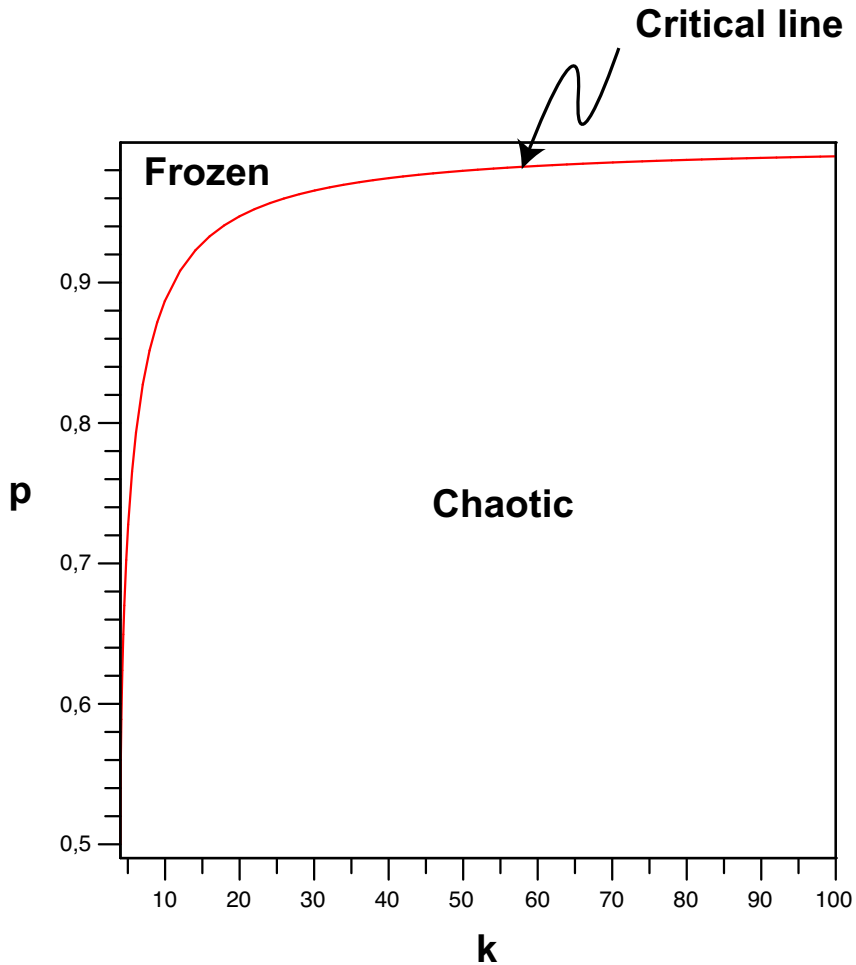


Figure: The phase diagram is only shown for $0 \leq p \leq 0.5$.

Scale-Free Networks I

So far we could attach a definite scale to the network. We shall now investigate networks that are scale free. For this we use the following connectivity distribution

$$P(k) = \frac{1}{\zeta(\gamma)} k^{-\gamma} \quad , \quad \gamma > 1 \quad . \tag{61}$$

with

$$\zeta(\gamma) = \sum_{k=1}^{\infty} k^{-\gamma} \tag{62}$$

being the Riemann Zeta-function.
The condition that $P(k)$ is normed requires that $\gamma > 1$. For the first moment we obtain

$$\langle k \rangle = \sum_{k=1}^{\infty} kP(k) = \begin{cases} \infty & \text{if } 1 < \gamma \leq 2 \\ \zeta(\gamma - 1)/\zeta(\gamma) & \text{if } \gamma > 2 \end{cases} \tag{63}$$

As done in the previous section we considered the case of the annealed model. Every element receives k inputs with probability $P(k)$

$$a(t+1) = G(a(t)) \tag{64}$$

and recall that $a(t) = 1 - D(t)/N$. The average probability that $k = 1, 2, ...$ controlling elements of the coupling function f are identical is

$$\mu(a) = \sum_{k=1}^{\infty} a^k P(k) \quad . \tag{65}$$

From this we can calculate the recursion function G

$$G(a) = 1 - 2p(1-p)[1 - \sum_{k=1}^{\infty} a^k P(k)] \tag{66}$$

For the fixpoint a^* of eq 64 we can again analyse the stability by considering $a^* + \delta a^*$. a^* is unstable iff

$$\begin{aligned} 1 &= \lim_{a \uparrow 1} \frac{dG(a)}{da} \\ &= 2p(1-p) \sum_{k=1}^{\infty} kP(k) \\ &= 2p(1-p)\langle k \rangle \quad . \end{aligned} \tag{67}$$

The phase transition is along this line because the fixpoint is stable for $\lim_{a\uparrow 1} \frac{dG(a)}{da} < 1$ and unstable for $\lim_{a\uparrow 1} \frac{dG(a)}{da} > 1$.
We shall now analyse the trajectory through phase space in terms of *limit cycles* and *attractors*. For this we switch to the quenched RBN. In this model the coupling functions are independent of time. For every initial condition we can follow Σ_t which surely will reach a state that was visited before. In this case it will cycle. Thus the phase space $\Omega = 2^N$ can be partitioned into cycles and attractors. An attractor can be defined as set of points from the phase space $A_t \equiv \{\Sigma_t\} \subset \Omega$ which will be mapped onto itself $A_{t+1} = A_t = A_0$. A *basin of attraction* of an attractor A_0 is a subset of Ω with

$$\text{ex } T < \infty : \Sigma_T \in A_0 \tag{68}$$

Let $k = N$. Assume that we start with Σ_0 and follow the trajectory. Let q_t denote the probability that the random walk is still not closed after t steps and let p_t denote the probability that the walk terminates (closes) after exactly t steps. If the trajectory is open after t steps then $t + 1$ different points were visited. Hence there are $t + 1$ possibilities to terminate the walk in the next step and this happens with the probability $\rho_t = (t + 1)/|\Omega|)$. Thus

$$p_{t+1} = \frac{t+1}{|\Omega|} q_t \quad . \tag{69}$$

The probability that the walk after $t + 1$ steps is open is

$$q_{t+1} = q_t(1-\rho_t) = q_t\left(1-\frac{t+1}{|\Omega|}\right) \tag{70}$$

with $q_0 = 1$. Due to the fact that the phase space is growing exponentially we can write

$$\begin{aligned} q_t &= \prod_{i=1}^{t}\left(1-\frac{i}{|\Omega|}\right) \approx \prod_{i=1}^{t} e^{-i/|\Omega|} && (71)\\ &= e^{-\sum_i i/|\Omega|} = e^{-t(t+1)/2|\Omega|} && (72) \end{aligned}$$

We can now calculate the average cycle length. Let $P(L)$ denote the probability that a given starting point is in the basin of attraction with cycle length L. The closing event happens with equal probability so that

$$P(L) = \sum_{t=L}^{|\Omega|} \frac{p_t}{t} \tag{73}$$

and for the expectation value

$$\langle L\rangle \approx |\Omega|^{1/2} \quad . \tag{74}$$

Scale-Free Networks V

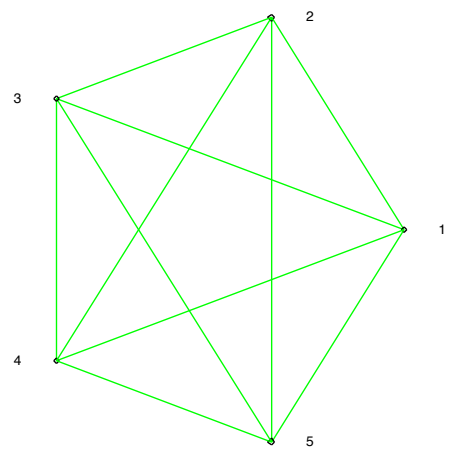


Figure: A random network with 5 nodes and 10 edges

Barabasi-Albert-Model I

Scale free graphs or networks can also arise from growth models. For this we need preferential attachment

Algorithm 2 Barabasi-Albert-Model

1. Start with a disconnected set of m_0 nodes
2. New nodes enter the network ate any time step.
3. For any new node m' new edges are formed.
4. The m' new edges connect the nodes with the old nodes. The latter ones are extracted with a probability

$$P(k_i) = \frac{k_i}{\sum_j k_j}$$

Assume that at every time step only one vertex enters. Then we have

$$n = m_0 + t \tag{75}$$

$$m' = \frac{1}{2}\sum_{j=1}^{n} k_j = mt \tag{76}$$

Barabasi-Albert-Model II

Assume that verteices enter the graph at a constant rate. Assume too that the degree is a continuous variable. The variation of the degree with time is given by

$$\frac{\partial k_i}{\partial t} = A\Pi(k_i) = A\frac{k_i}{\sum_{j=1}^{m_0+t-1} k_j} = A\frac{k_i}{2mt} \tag{77}$$

The constanr A is the change in connectivity in one time step, hence $A = m$. Initially at t_0, the initial degree is $k(t_0) = m$. It follows that

$$\frac{\partial k_i}{\partial t} = \frac{k_i}{2t} \Rightarrow k_i(t) = m\left(\frac{t}{t_0}\right)^{1/2} \tag{78}$$

The probability $P(k_i < k)$ that a vertex has a degree k is

$$P(k_i < k) = P\left(t_i > \frac{m^2 t}{k^2}\right) \tag{79}$$

Since vertices enter at a constant rate, their distribution is uniform in time

$$P(t) = \text{const} \tag{80}$$

Thus

$$\int_0^{m_0+1} P(t)dt = 1 \Rightarrow P(t) = \frac{1}{m_0 + t} \tag{81}$$

and hence

$$P\left(t_i > \frac{m^2 t}{k^2}\right) = 1 - P\left(t_i \leq \frac{m^2 t}{k^2}\right) = 1 - \frac{m^2 t}{k^2(m_0 + t)} \tag{82}$$

It follows that

$$P(k) = \frac{\partial P(k_i > k}{\partial k} = \frac{2m^2 t}{m_0 + t}\frac{1}{k^3} \tag{83}$$

The distribution floows a power law

$$P(k) \sim k^{-\gamma} \quad , \quad \gamma = 3 \tag{84}$$

Hence the degree-distribution is scale-free.

Cytoskeletal Networks I

abc

Neural Networks I

The general goal is to create **artificial neural networks** (graphs) (**ANN**) that imitate to some extend the capabilities of the human brain:

- learning
- generalization
- adaptivity
- fault tolerance
- ...

We want this for example for

- pattern classification
- function approximation
- ...

Pioneering work was done by McCulloch and Pitts with the Perceptron [8, 9]. This was extended by Minsky and Papert [10].
McCulloch and Pitts proposed a binary threshold model as a computational model for an artificial neuron. Let $x_1, ..., x_n$ be the input values and $y = 0, 1$ be the output. The perceptron is defined by

$$y = \begin{cases} 0, & \sum_i x_i w_i \leq b \\ 1, & \sum_i x_i w_i > b \end{cases} . \tag{85}$$

where $w_1, ..., w_n$ are the synaptical weights that Rosenblat [9] introduced (see Figure 11). This can be reformulated as

$$y = \Theta(\sum_{j=1}^{n} w_j x_j - b) \tag{86}$$

This generates an output of 1 if the sum is above a certain threshold.
Sometimes we include b in the sum and set $w_0 = -b$ and x_0 to a constant input $x_0 = 1$.

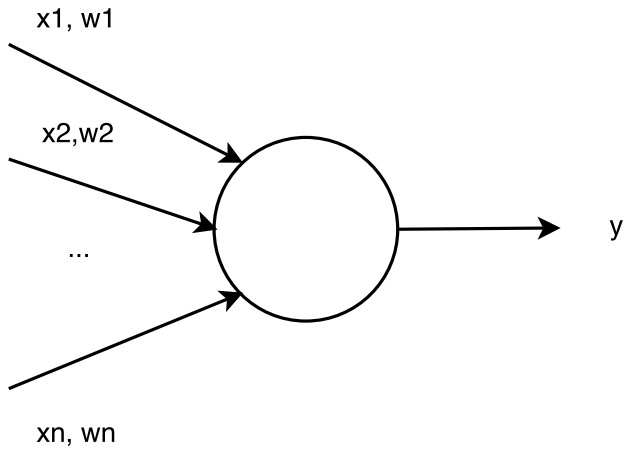


Figure: Perceptron

In this setting

- positive weights correspond to **excitatory synapses**
- negative weights correspond to **inhibitory synapses**

Clearly one can also use other activation function like

- piecewise linear
- sigmoid neuron
- gaussian

Most often used is the sigmoid function (here the **logistic function**)

$$g(x) = \frac{1}{1 + e^{-\beta x}} \tag{87}$$

The above constructed node is then the basic unit in a network (graph) of nodes. Thus the ANN's are weighted directed graphs where

$$\text{neuron} \quad \cong \quad \text{node} \tag{88}$$

$$\text{connection between neuron} \quad \cong \quad \text{directed edge with weights} \tag{89}$$

Example: XOR

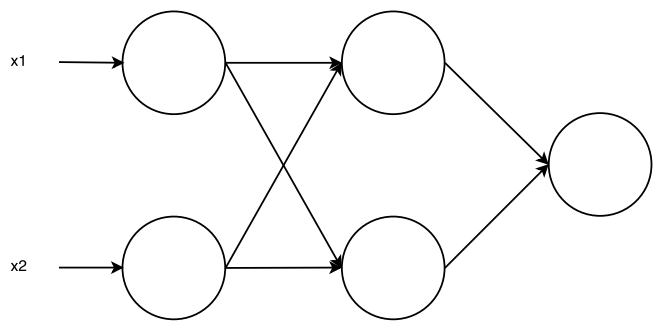


Figure: Perceptron XOR

Neural Networks V

Connectionist models for gene regulation in the form of recurrent Hopfield [11] networks have been proposed by Mjolsness and others [12–14] to describe regulatory networks as directed graphs or matrices of interactions without restrictions on connectivity. These continuous time networks model interphase expression of a cell based on interaction weights that are free to take positive and negative real values.

Neural Networks VI

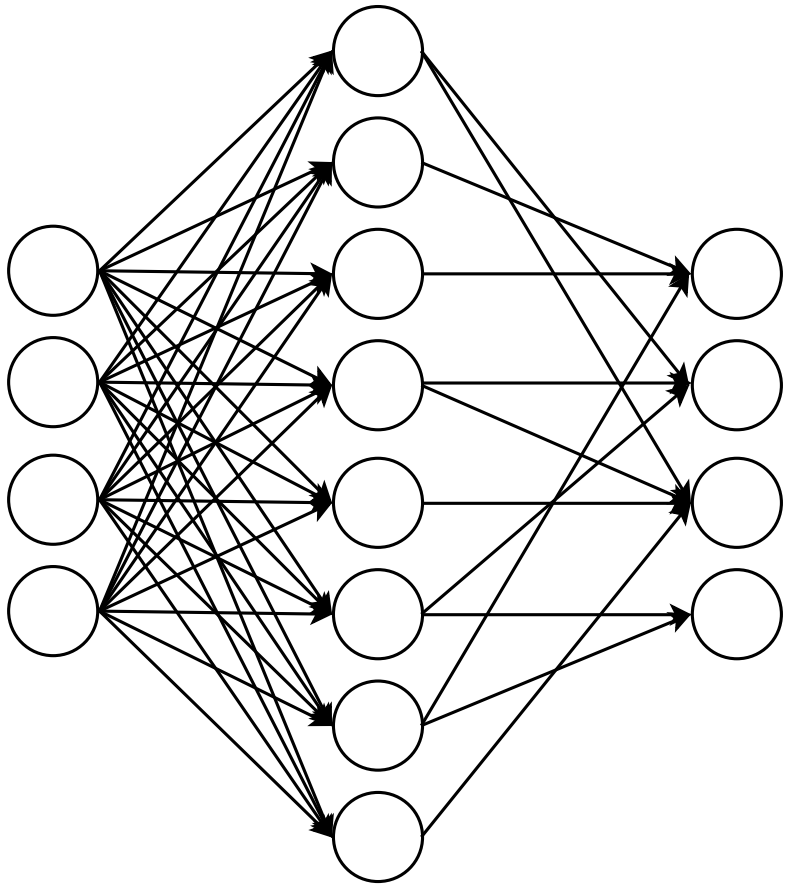


Figure: Neural network

Backpropagation Algorithm I

backpropagation [15]

Gels I

Many systems exhibit a network structure (see figure **??**). The mechanical properties of such networks are often characteristic of a solid even though they being disordered and may even be mostly liquid like. The structure of such a network is described by the following structure parameters. First there is an elastically active network chain between two crosslinks. Then there are dangling chains which are attached to the network by a single point. The gel fraction includes all the material attached to the network.
Let us look at the classical model of gelation proposed by Stockmayer and Flory [16–18] for the sol gel transition. The model assumes monomers having f valences which can bind (simple) to other monomers to form a network. Not all valences are bound. We assume that each valence has probability of $p(T, c_i, ...)$ of being saturated. Here p is considered to be function of temperature T, concentration c etc.
Take the generating function

$$F_0 = \sum_n \omega_n^{(0)}(p)\theta^n \tag{90}$$

where $\omega_n(p)$ is the probability that a randomly picked monomer (A) is part of a cluster of $n+1$ monomers. θ is a factor for every monomer in the cluster except for the considered monomer.
Now

$$F_0 = [1 - p + p\theta F_1(p, \theta]^f = [(1-p)u(p, \theta)]^f \tag{91}$$

where F_1 (with $\omega_m^{(1)}$ being the function corresponding to the situation that monomer B, attached to monomer A at the ν-th valence of A, is attached to m other monomers. To make progress on the computation we assume that the f bonds of A are un-correlated and that the network has no loops (Cayley-tree, see figure 14) (thus a mean-field approximation). Hence

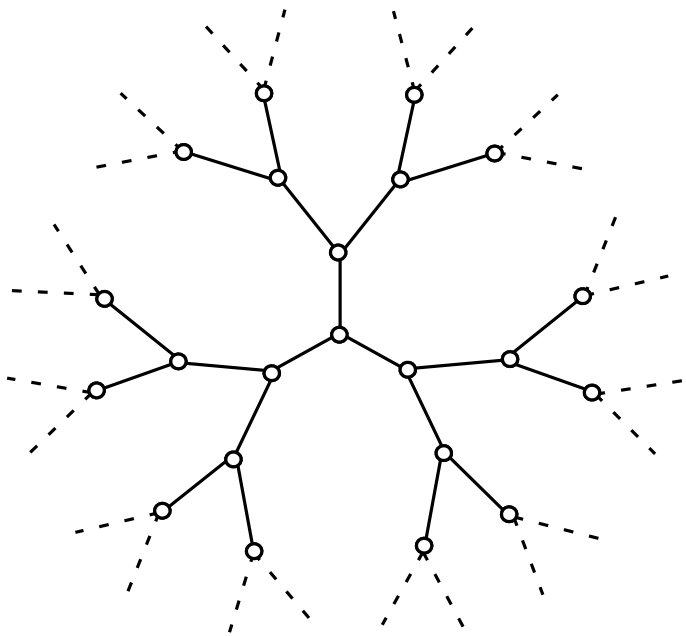


Figure: The Cayley tree is a tree with all nodes having the same connectivity, here $q = 3$ and without loops

$$F_1(p,\theta) = [1 - p + p\theta F_1(p,\theta)]^{f-1} = [(1-p)u(p,\theta)]^{f-1} \tag{92}$$

Solving this we find F_0 and find all $\omega_n^{(0)}$

$$\frac{1}{p\theta(1-p)^{f-2}}\left(\frac{p\theta}{1-p}F_1\right) = \left(1 + \frac{p\theta}{1-p}\right)^{f-1} \tag{93}$$

or

$$u - 1 = xu^{f-1} \tag{94}$$

Thus the function $u = 1 + F_1 p\theta/(1-p)$ is a function of $x = p\theta(1-p)^{f-2}$ alone. One can calculate the Taylor-expansion of u

$$u = \sum_{n=0}^{\infty} \frac{[(f-1)n]!x^n}{[(f-2)n+1]!n!} \tag{95}$$

from which we find

Gels IV

$$F_1 = (1-p)^{f-1} \sum_{n=1}^{\infty} \frac{[(f-1)n]!}{[(f-2)n+1]!n!} x^{n-1} \tag{96}$$

$$F_0 = (1-p)^{f} f \sum_{n=1}^{\infty} \frac{[(f-1)n]!}{[(f-2)n+2]![n-1]!} x^{n-1} \tag{97}$$

$$\tag{98}$$

To understand F_1 we go back to equation 92 and write

$$F_1 = w^{f-1} \tag{99}$$

$$\frac{w-1}{p\theta} = w^{f-1} - 1/\theta \tag{100}$$

Since w is a monotonically increasing function of θ we find that only one intersection is possible. Assume $\theta = 1$, then there are two cases. Case one

$$f - 1 < 1/p \qquad \text{oder} \qquad p < \frac{1}{f-1} \tag{101}$$

This mean that at monomer B the average number of bonds is

$$p(f-1) < 1 \tag{102}$$

The chain forming dies out

$$\sum \omega_n(p) = 1 \tag{103}$$

and we only find finite connectivity. The system is in the sol phase.
If

$$f - 1 > 1/p \qquad \text{or} \qquad p(f-1) > 1 \tag{104}$$

the from generation to generation the number of bonds increase. The *gel point* in this theory is thus

$$p_c = \frac{1}{f-1} \tag{105}$$

We shall now investigate the behaviour of the system in the neighbourhood of the gel point. To do so, we will expand various property function around

$$\Delta = p - p_c = p - \frac{1}{f-1} \qquad . \tag{106}$$

The gel fraction is given by

$$G = 1 - F_0 = 2f\frac{f-1}{f-2}\Delta \sim \Delta^{\beta} \quad , \quad \beta = 1 \tag{107}$$

Next we look at the *polymerization index* N_ω

$$N_\omega = \frac{\sum_n n\omega_n(p)}{\sum_n \omega_n(p)} = \frac{\partial}{\partial\theta} \ln F_0(\theta = 1) \tag{108}$$

We have

$$F_0' = pf(1 + F_1')F_1 \tag{109}$$

$$F_1' = p(f-1)(1 + F_1')F_1^{(f-2)/(f-1)} \tag{110}$$

and for $\theta = 1$, $p < p_c$, i.e.,$F_0 = F_1 = 1$

$$N_\omega = \frac{pf}{1 - p(f-1)} \approx \frac{f}{|\Delta|} \sim |\Delta|^{-\gamma} \tag{111}$$

The next observable we investigate is the cluster size distribution. With the help of the Stirling formula we find from equation 98

$$\omega_n \sim e^{-nq(\Delta} n^{-3/2} \sim n^{3/2} e^{-n\Delta^{1/\sigma}} \tag{112}$$

with

$$q(\Delta) = -(f-2)\ln\left(1-\frac{f-1}{f-2}\Delta\right) - \ln[1+(f-1)\Delta] \quad (113)$$

$$\approx \frac{1}{2}\frac{(f-1)^3}{f-2}\Delta^2 \quad (114)$$

All in all we find the typical signature for a second-order phase transition with critical exponents $\beta, \gamma, \sigma, \tau$ given by

$$G \sim \Delta^\beta \quad , \quad \beta = 1 \quad (115)$$

$$N \sim |\Delta|^{-\gamma} \quad , \quad \gamma = 1 \quad (116)$$

$$\omega_n \sim n^{-\tau} e^{-n|\Delta|^{1/\sigma}} \quad , \quad \tau = 3/2, \sigma = 1/2 \quad (117)$$

Exercise 1: **Erdös-Renyi-Model with fitness**

Algorithm 3 Barabasi-Albert-Model

---

1. Start with a disconnected set of m vertices, each characterized by a constant ability (fitness) η_i to attract new edges. Assume a distribution $\rho(\eta)$
2. As in the Barababsi-Albert-model there is growth since vertices enter the system
3. There is preferential attachment

$$\Pi(k_i, \eta_i) = \frac{k_i \eta_i}{\sum_{j=1}^{n} k_j \eta_j}$$

Exercise 2: **Linear Neurons**
Work out the backpropagation algorithm for a linear function $\sigma(z) = z$.

Exercise 3: Show that the network remains invariant under multiplication of the weights and biases with a constant factor $c > 0$.

Bibliography I

[1] M. Widom I. Al-Lehyani. D. pages 112–123, 1997. C. Weaver, C.T. Workman, G.D. Stormo: *Modeling regulatory networks with weight matrices*, Pacific Symp. Biocomp. 99 4:1999. Networks.

[2] The large scale structure S. A. Kauffman and 167-190 dynamics of genetic control circuits: an ensemble approach. Journal of Theoretical Biology, 44. 1974. 1974.

[3] The origins of order: Self-organization S. A. Kauffman and New York selection in evolution. Oxford University Press. 1993. 1993.

[4] The origins of order: Self-organization S. A. Kauffman and New York selection in evolution. Oxford University Press. S. 1993. Low, D. Lapsley, Optimization flow control, i: Basic algorithm and convergence, IEEE/ACM Transactions on Networking, 1999.

[5] R. Thomas, editor. *Kinetic Logic: A Boolean approach to the analysis of complex regulatory systems*. Lecture Notes in Biomathematics, Springer Verlag, 1979.

[6] Kinetic Logic: A Boolean approach to the analysis of complex regulatory systems Lecture Notes in Biomathematics Springer Verlag 29 R. Thomas, ed. Y. 1979. Kanada, The effects of randomness in asynchronous 1D cellular automata, Proceedings of ALIFE IV, 1994.

Bibliography II

[7] Kinetic Logic: A Boolean approach to the analysis of complex regulatory systems Lecture Notes in Biomathematics Springer Verlag 29 R. Thomas, ed. I. pages 65–75, 1979. Harvey, T. Bossomaier, Time out of joint: attactors in asynchronous random Boolean networks, Proceedings of the Fourth European Conference on Artificial Life (ECAL97), MIT Press, 1997, p.

[8] W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5:115–133, 1943.

[9] Frank Rosenblat. *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*. Spartan Books, 1962.

[10] Marvin Minsky and Seymour A. Papert. *Perceptrons: An introduction to Computational Geometry*. MIT Press, 1969.

[11] Proc. Nat. Acad. Sci. 79 2554-2558 J. J. Hopfield. 1982. 1982.

[12] J. Reinitz E. Mjolsness, D. H. Sharp. *J. Theor. Biol.*, 152:429–453, 1991.

[13] Mechanisms of Development 49 133-158 J. Reinitz, D. H. Sharp. 1995. 1995.

[14] M. Kaufman R. Thomas, D. Thieffry. *Bul. Math. Biol.*, 57(2):247–276, 1995.

[15] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 10 1986.

[16] M. Eden Pmc. 4th Berkpley Symg. Math Stat. Pmb. 4 233. P. 1961. J.Flory, in Principles of Polymer Chemistry, Cornell Univ. Press, Ithaca 1953.

[17] M.Gordon and 3933 G. R.Scantlebury, J.Polym.Sci. C16. 1968. 1968.

[18] D. R.Miller and 1063 C. W.Macosko, Macromolecules 13. 1980.

Index I

Index II

Index III

Index IV

Index V