Quick recap :

Graph-based learning

Two situations:

① $\{\vec{y}_i\}_{i=1}^{N}$ A dataset as a cloud of $\left\{\begin{matrix}\text{vectors}\\ \text{points}\end{matrix}\right\}$ in $\mathbb{R}^P$

$\vec{y}_i \in \mathbb{R}^P$

$\downarrow$

$D_{N\times N}$ or $S_{N\times N}$

summarises pair-wise interactions between samples.

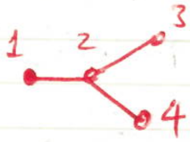
② Our dataset is directly given by pairwise relationships between samples or observations (entries)

Both in $\left\{\begin{matrix} ① & \text{Data} \longrightarrow \underline{D_{N\times N}} \text{ or } S_{N\times N} \\ \text{or} & \\ ② & \text{we only have } D_{N\times N} \end{matrix}\right\}$

we can represent the dataset as a graph where the nodes (or vertices) are the samples and the links (or edges) are the pair-wise relationships:

Definition: Graph: $G = \{V, E\}$ $V = \{i\}_{i=1}^{N}$

e.g.



N=4

$V = \{1,2,3,4\}$
$E = \{(1,2),(2,3),(2,4)\}$

$E = \{(i,j)\}_{i\neq j}$

Graph can be

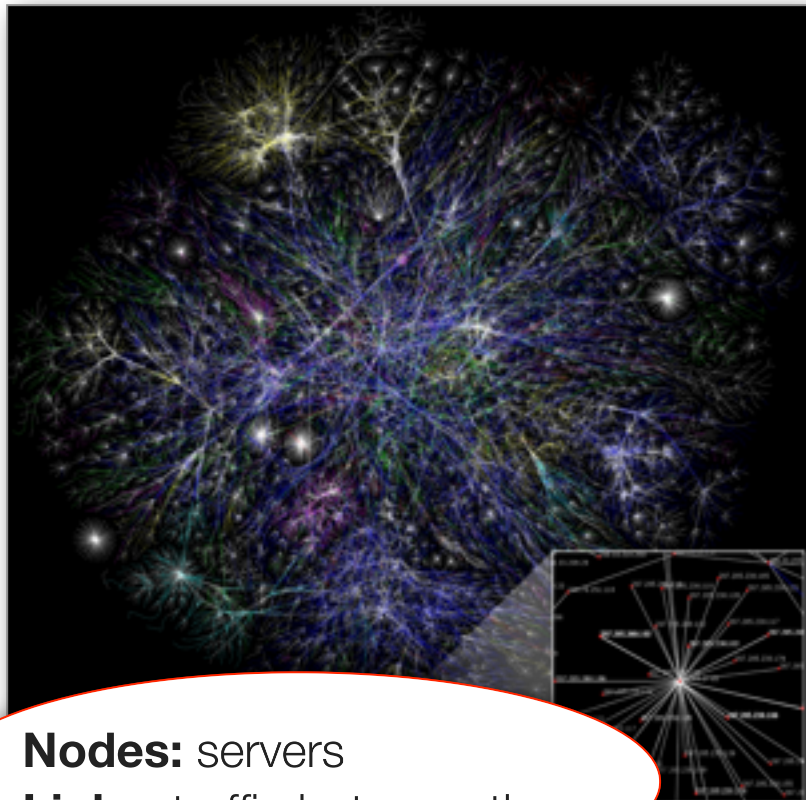
$\left[\begin{matrix}\text{undirected / directed}\\ \text{unweighted / weighted}\end{matrix}\right]$

Why graphs?

- In our Machine Learning lectures we looked at problems where we have $i=1,2,...N$ observations.
- But we assumed (implicitly or explicitly) that these N samples were independent from each other - i.e. that they do not interact with one another.
- Sometimes there are many interactions, and in fact, such interactions are a **crucial feature of the data**.
- Many interactions in the real world:
    - social media (follow, re-tweet), friendships, family
    - proteins in cells bind to one another
    - transport (shared commute, same to/from flights)
    - etc, etc...

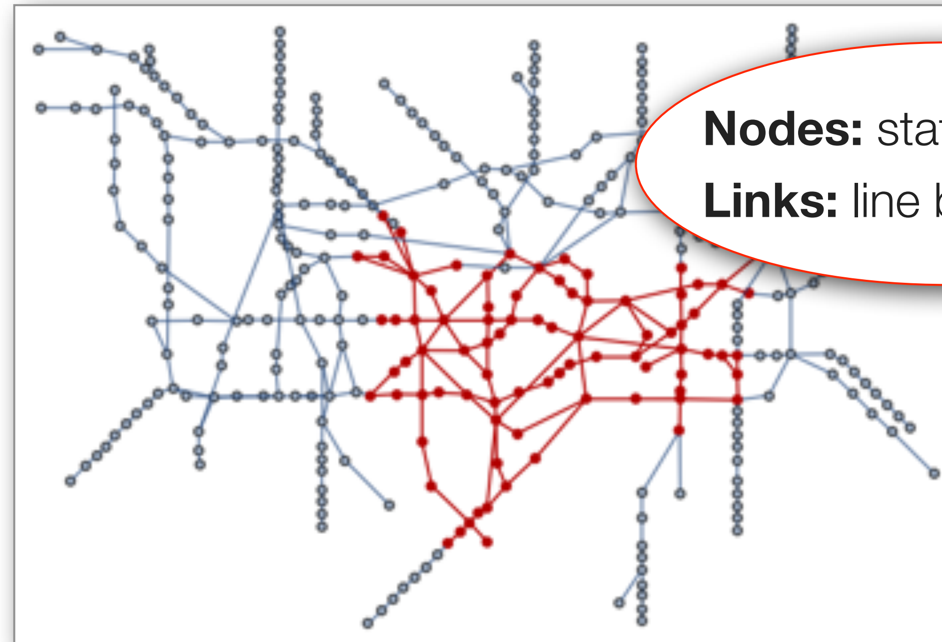
A few networks

The internet



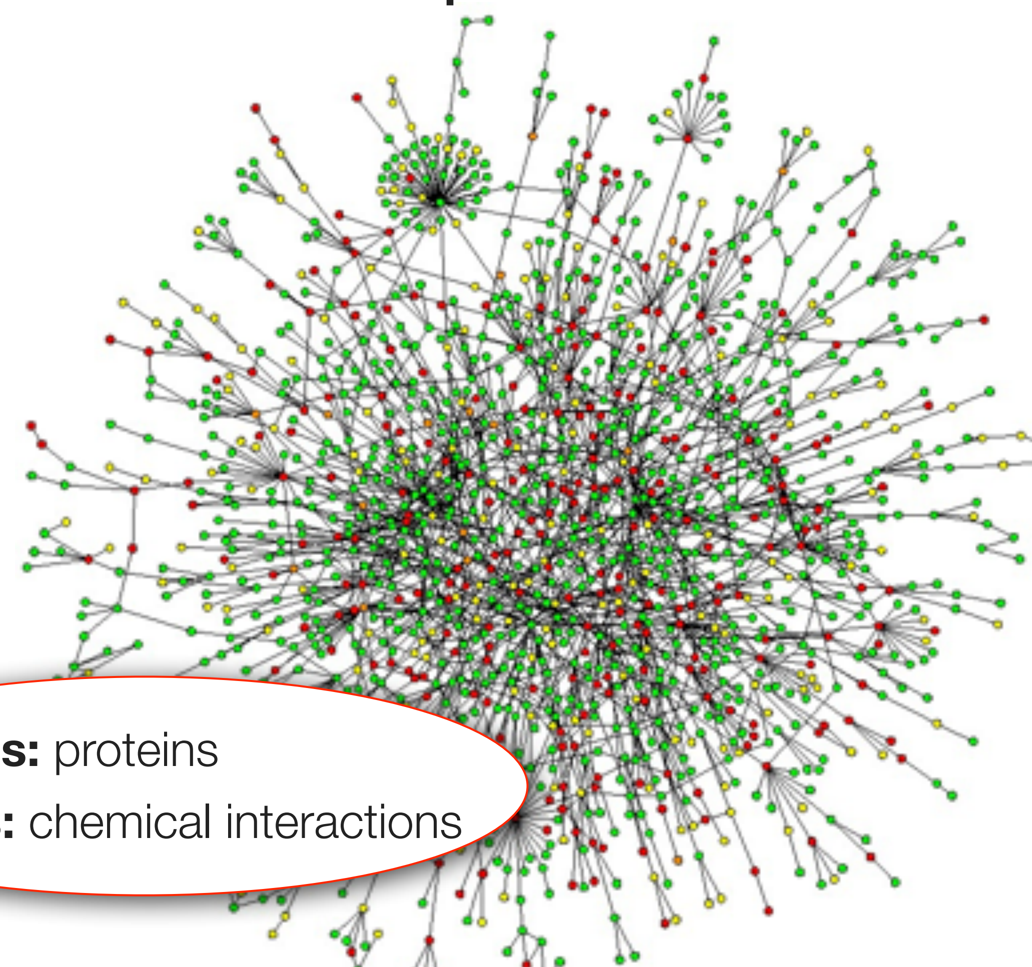
Nodes: servers
Links: traffic between them

London Underground



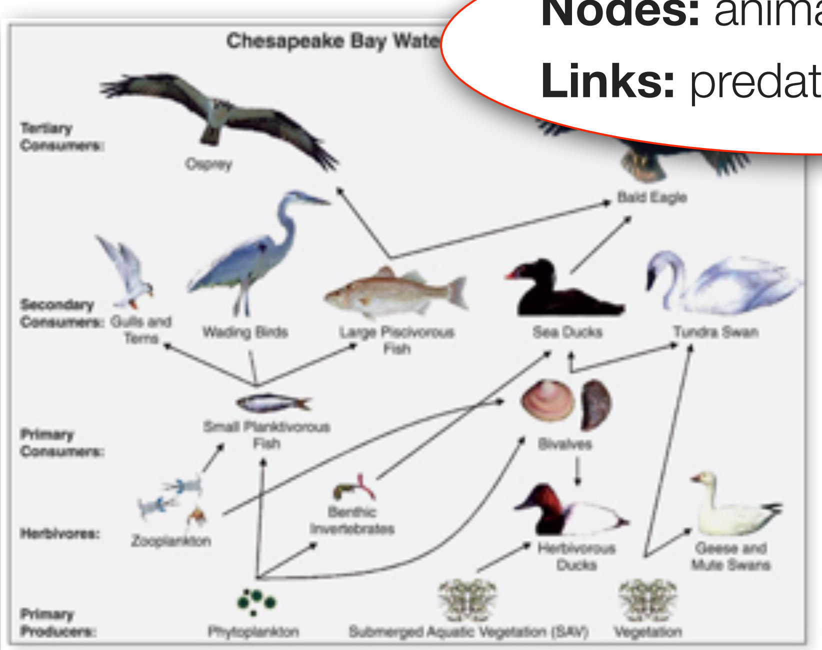
Nodes: stations
Links: line between them

***S cerevisiae* protein interactions**

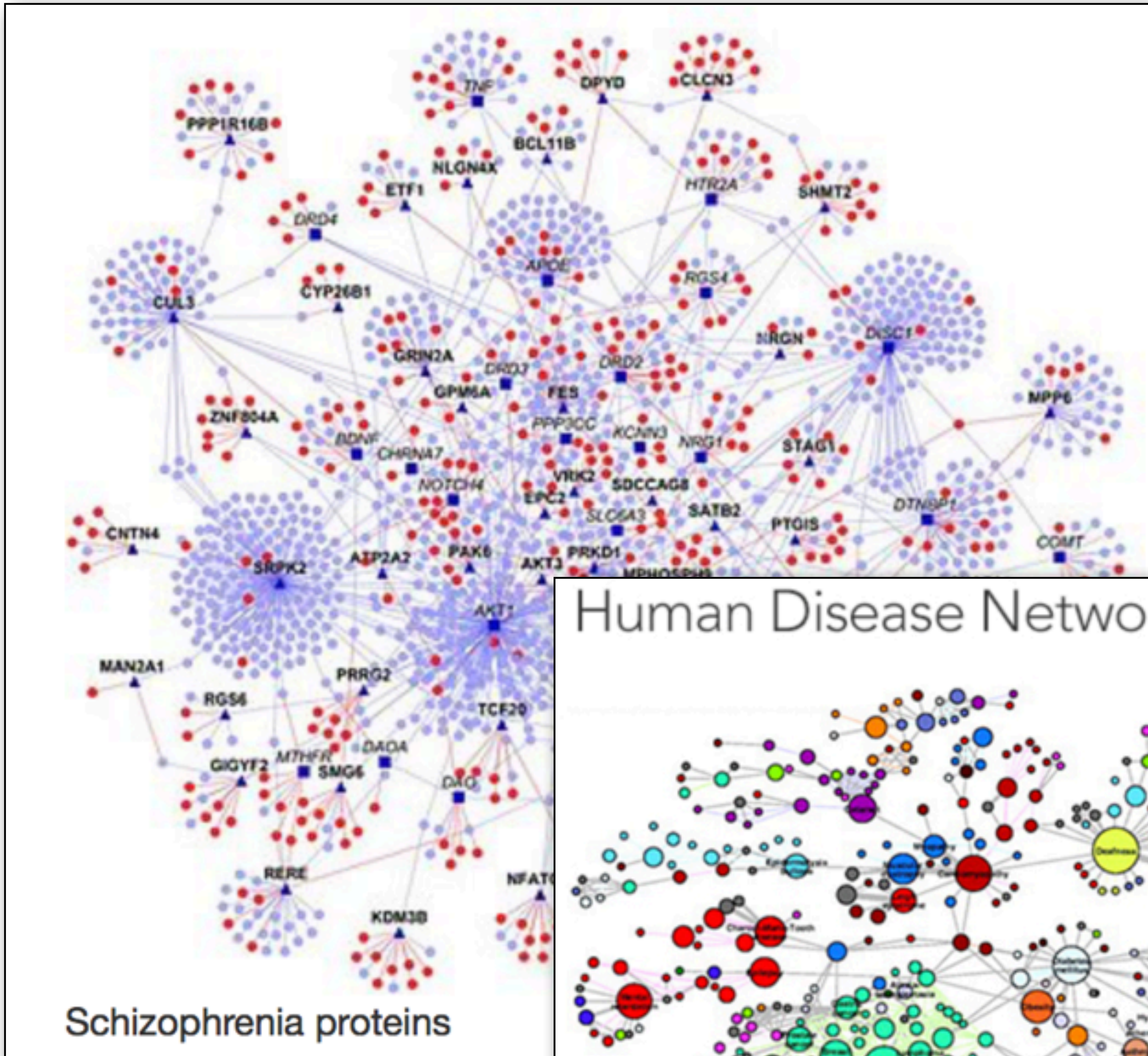


Nodes: proteins
Links: chemical interactions

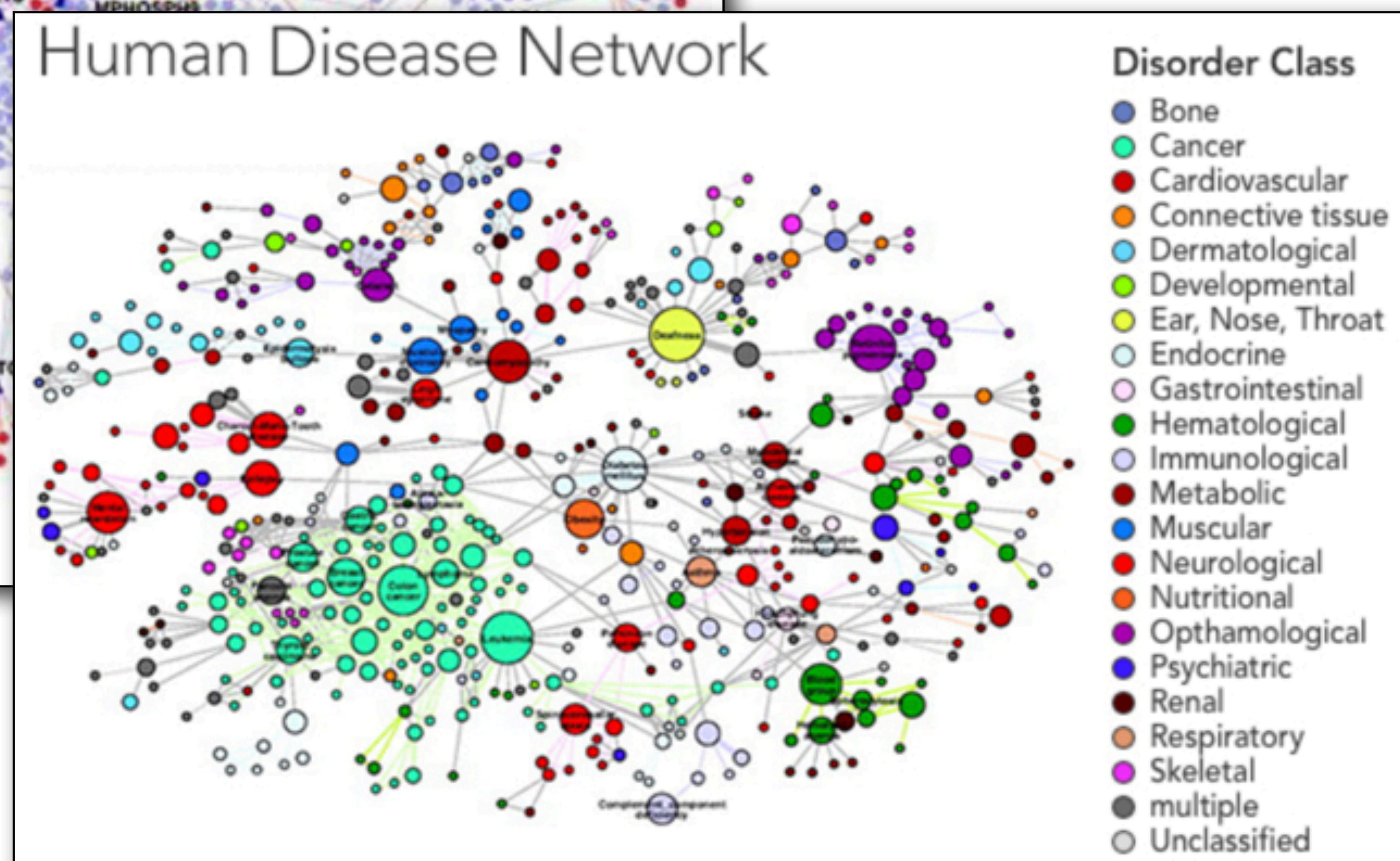
Food webs



Nodes: animals
Links: predator-prey relation



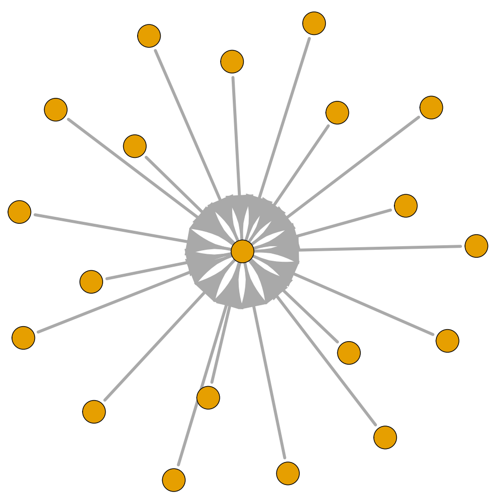
Schizophrenia proteins



Network science

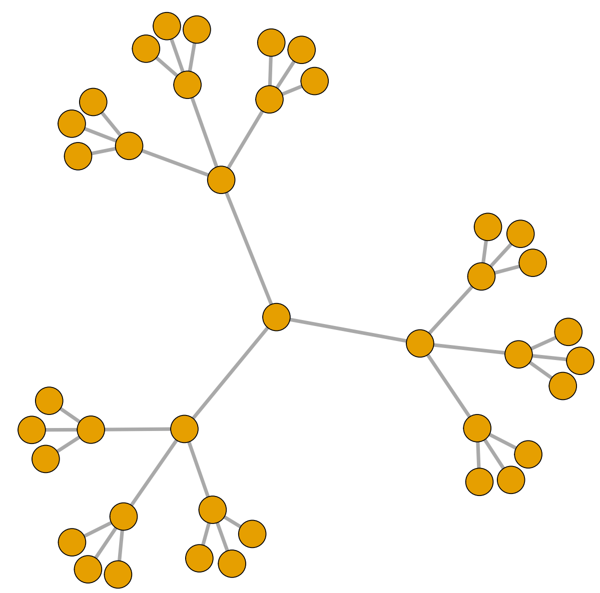
- Network science provides methods to study such complex interactions.
- Especially useful for ***large*** networks, i.e. those with many components (players) and complex connectivity.

Small networks are intuitive



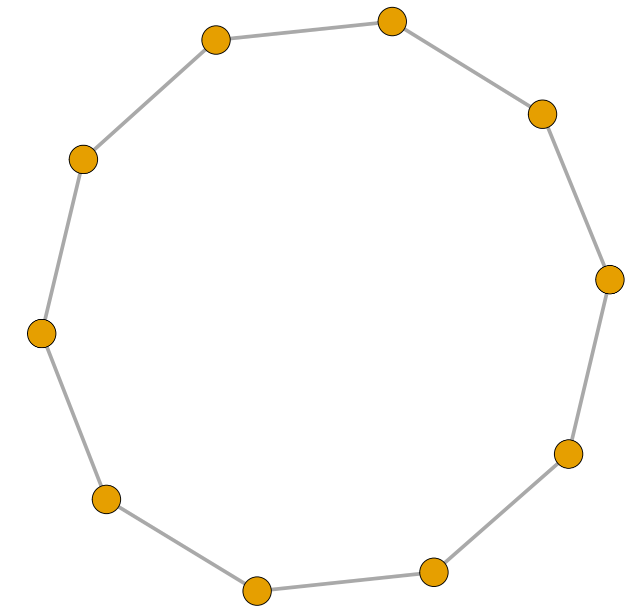
star

one hub



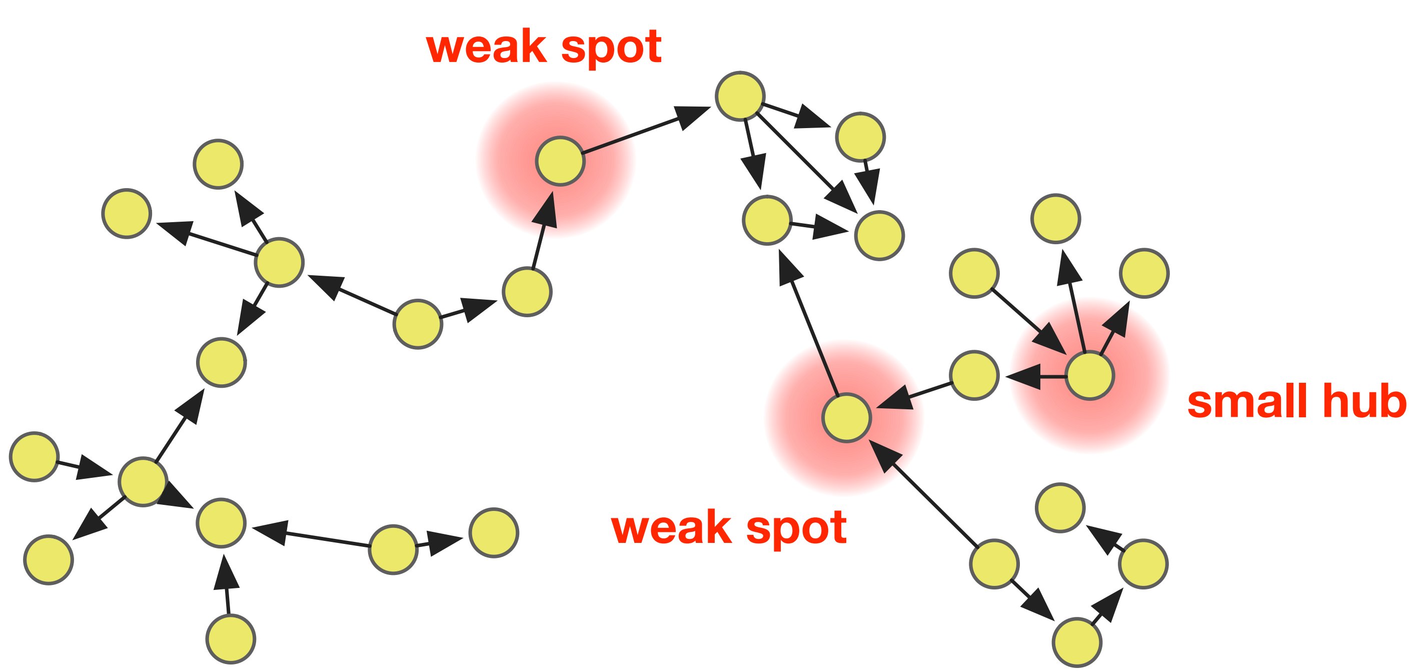
tree

several small hubs



ring

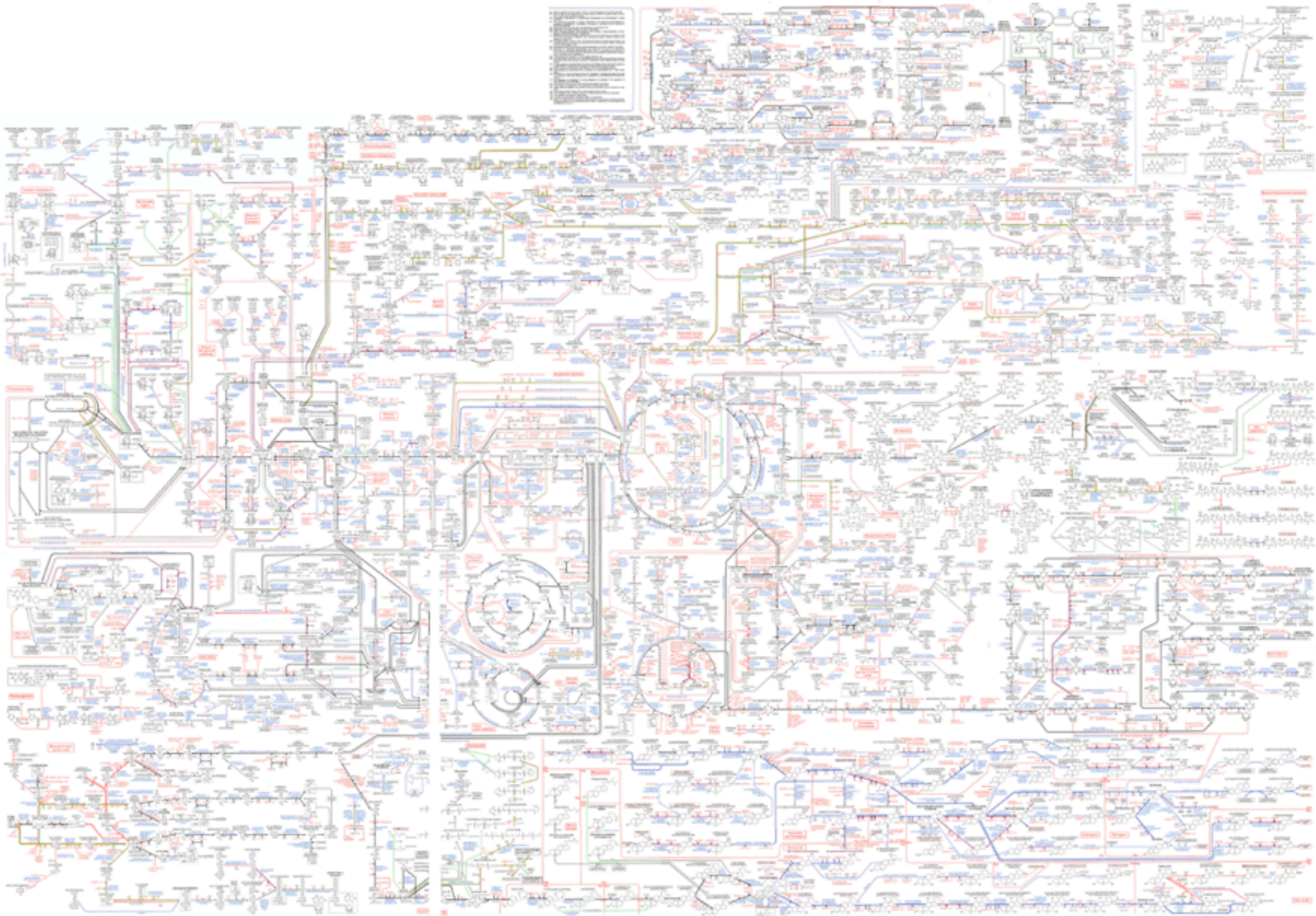
only two links per node



But large ones are not (at all)

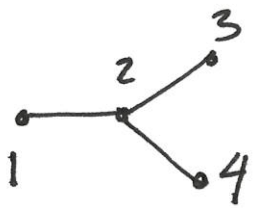
wikipedia

But large ones are not (at all)



Human metabolism

Example:



Undirected

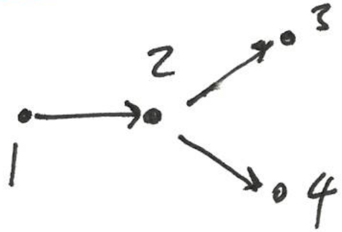
$N=4$

$$A_{N\times N} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \qquad A\vec{1} = \begin{bmatrix} 1 \\ 3 \\ 1 \\ 1 \end{bmatrix} = \vec{d}$$

degree of node 1

degree vector

Directed



$$A = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \qquad A\vec{1} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \vec{d}_{in}$$

in-degree

$$A \neq A^T$$

$$A^T\vec{1} = \begin{bmatrix} 1 \\ 2 \\ 0 \\ 0 \end{bmatrix} = \vec{d}_{out}$$

out-degree

---

(1) Graph construction from data.

We do not have a graph
but just $\{y_i\}_{i=1}^{N}$

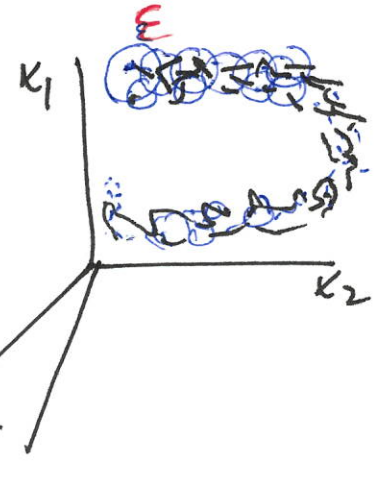
$\longrightarrow$ $D_{N\times N}$, $S_{N\times N}$

Adjacency of weighted complete graphs.

- Thresholding $S_{N\times N}$ is not the best strategy (usually precision matrix is important)
- Geometric graphs:

(i) ε-ball

(ii) kNN



$D_{N\times N}$

Parameters

The "other" matrix (besides A):

Laplacian matrix

Extension of the Laplacian operator:

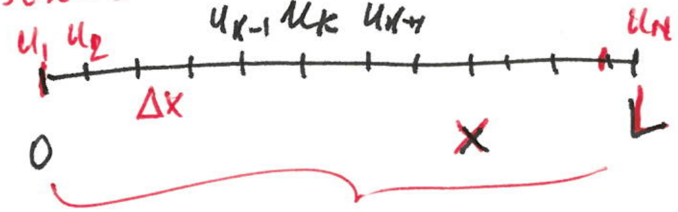
⊙ Recall the PDE:

$$u = u(x,t), \quad \frac{\partial u}{\partial t} = \underbrace{\nabla^2}_{\text{Laplacian}} u$$

{ Heat equation or Diffusion equation

In this case: $\nabla^2 u = \frac{d}{dx}\left(\frac{d}{dx}u\right)$ $\quad x \in [0, L], \; t \in \mathbb{R}^+$

Discretisation:



$$\vec{u} = \begin{pmatrix} u_1 \\ \vdots \\ u_N \end{pmatrix}_{N \times 1}$$

$$\nabla^2 u = \frac{d}{dx}\left(\frac{d}{dx}u\right) \simeq \Delta(\Delta u) = \Delta(u_{k+1} - u_k)$$

$$= (u_{k+2} - u_{k+1}) - (u_{k+1} - u_k)$$

$$= \underline{u_{k+2} - 2u_{k+1} + u_k}$$

Can be rewritten in terms of a matrix $\longrightarrow$

Equation becomes:

$$\frac{d\vec{u}}{dt} = \underbrace{\begin{pmatrix} -1 & 1 & & & & & \\ 1 & -2 & 1 & & & 0 & \\ & \ddots & \ddots & \ddots & & & \\ & & 1 & -2 & 1 & & \\ 0 \cdots & & & 1 & -2 & 1 & 0 \cdots 0 \\ & & & & 1 & -2 & 1 \\ & 0 & & & & \ddots & \ddots \\ & & & & & 1 & -1 \end{pmatrix}_{N \times N}}_{\boxed{\text{Laplacian matrix}} \,\equiv\, -L} \begin{pmatrix} u_1 \\ \vdots \\ u_k \\ u_{k+1} \\ u_{k+2} \\ \vdots \\ u_N \end{pmatrix}$$

Graph:

$$-L = A - D$$

where $D = \mathrm{diag}(\vec{d}) = \mathrm{diag}(A\vec{1})$

Definition: $\boxed{\text{Combinatorial graph Laplacian. } \quad L = D - A}$

$$\boxed{\frac{d\vec{u}}{dt} = -L\vec{u}}$$

Heat equation on the graph.

Solution: $\vec{u}(t) = e^{-Lt}\,\vec{u}(0)$

Note that: ① $L\vec{1} = (D-A)\vec{1} = \vec{d}-\vec{d} = 0.$

i.e., $\vec{1}_{N\times 1}$ is an eigenvector of L with eigenvalue zero.

② So $\vec{1}$ is a stationary point of the heat equation.

...