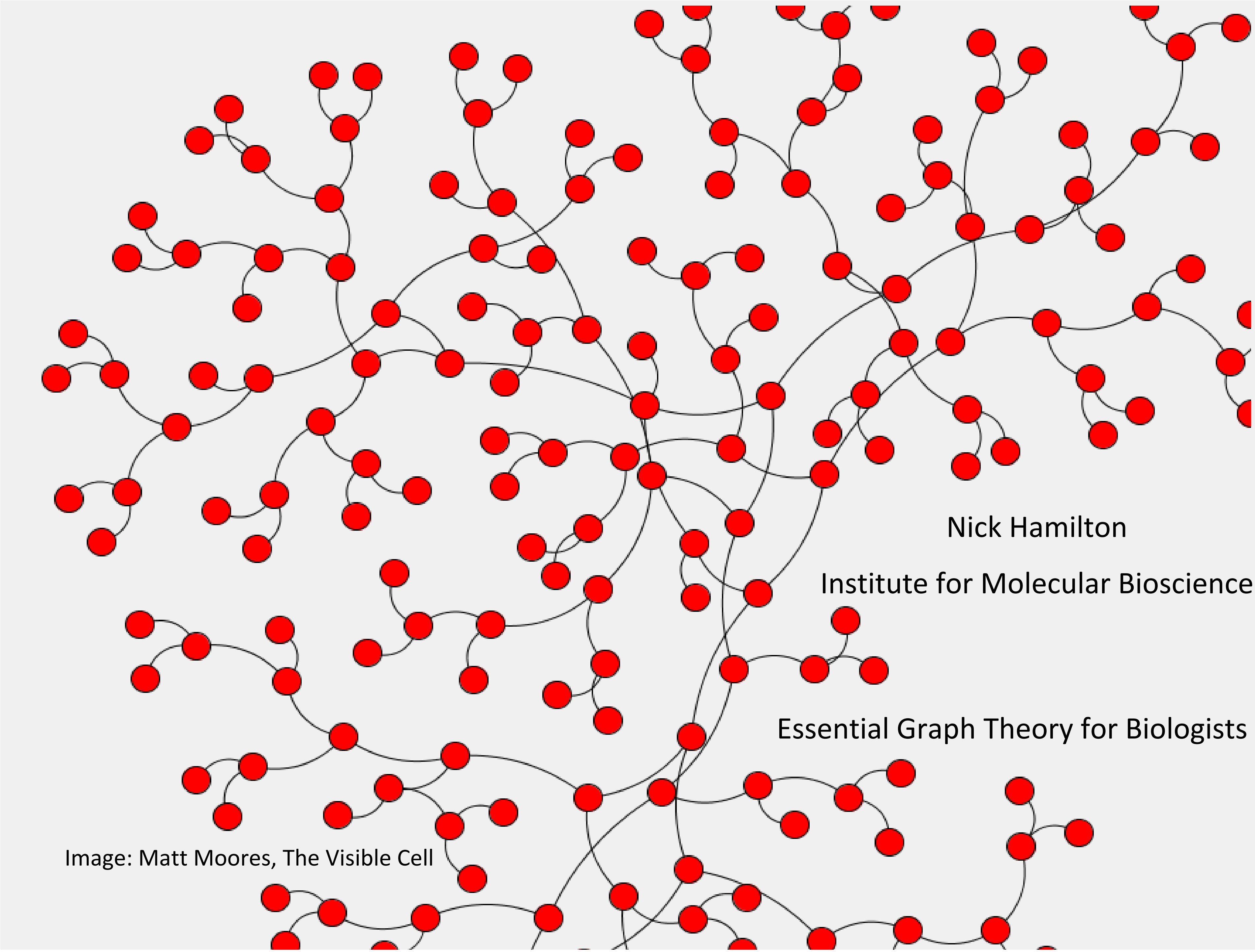


Nick Hamilton

Institute for Molecular Bioscience

Essential Graph Theory for Biologists

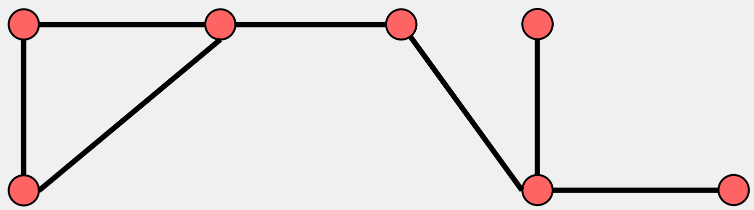
Image: Matt Moores, The Visible Cell

Outline

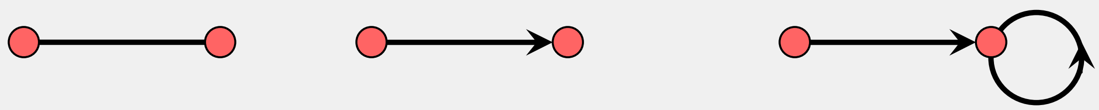
- Core definitions
- Which are the most important bits?
- What happens when I break it? Robustness
- What are the functional modules?
- Are there functional modules?
- Getting around in a graph
- Graph algorithms
- Trees & hierarchical structure
- Small world and scale free graphs
- Software

Core Definitions

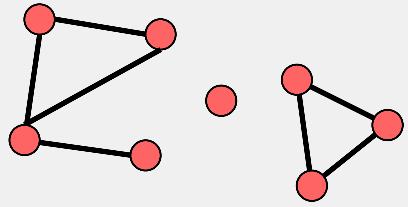
A graph is a collection of ***nodes*** or ***vertices*** and a set of **edges** that connect pairs of nodes.



Edges may be ***undirected*** or ***directed*** or have ***loops***



A graph might have multiple disconnected ***components***

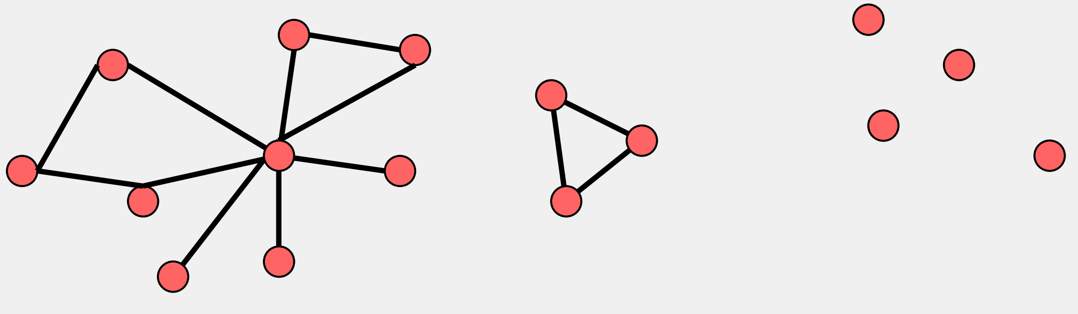


3 components

A simple example

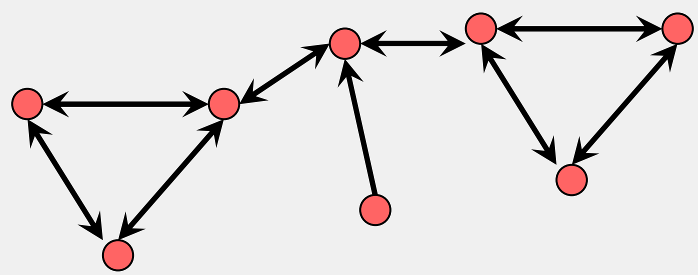
Nodes: people in this room

Edges: "are friends"



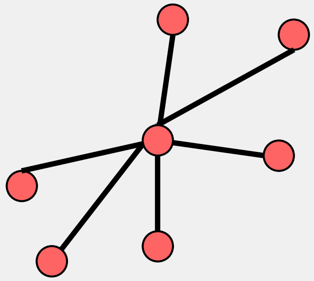
Nodes: people in this room

Edges: "likes"



Which graph bit is the most important?

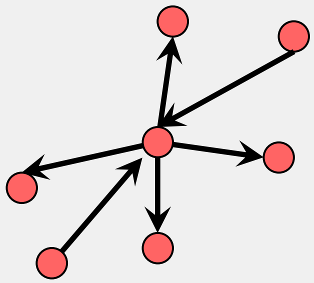
For an undirected graph, the ***degree*** of a node is the number of edges connected to a node



Degree 6

Degree 0

If the graph is directed, define ***in-degree*** and ***out-degree*** defined similarly



In-degree 2
Out-degree 4

Which graph bit is the most important?

A ***hub node*** is a node of “high” degree, relatively

The inevitable example, the **p53 protein interaction network**

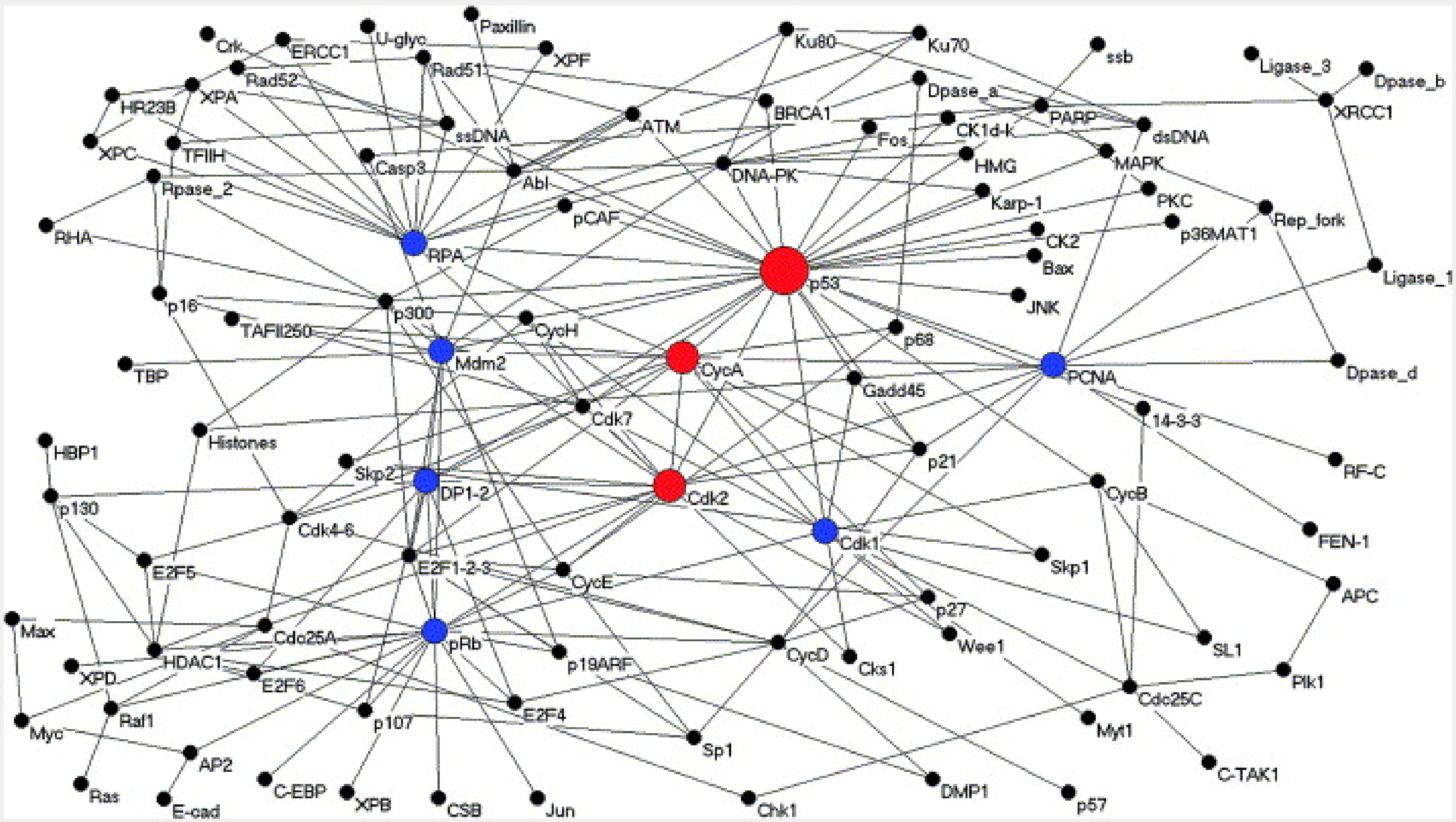
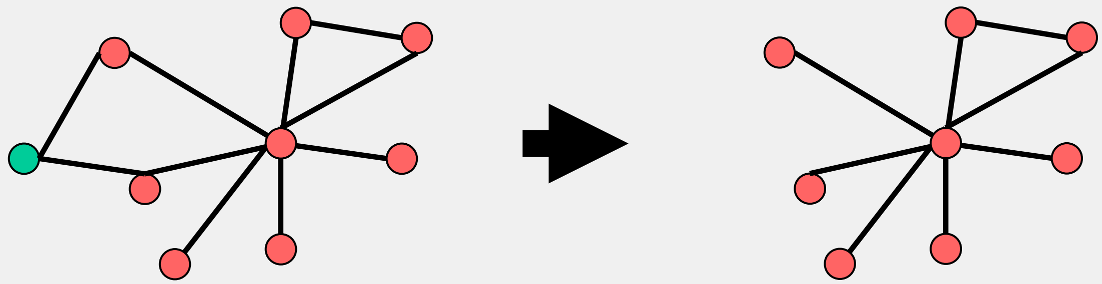


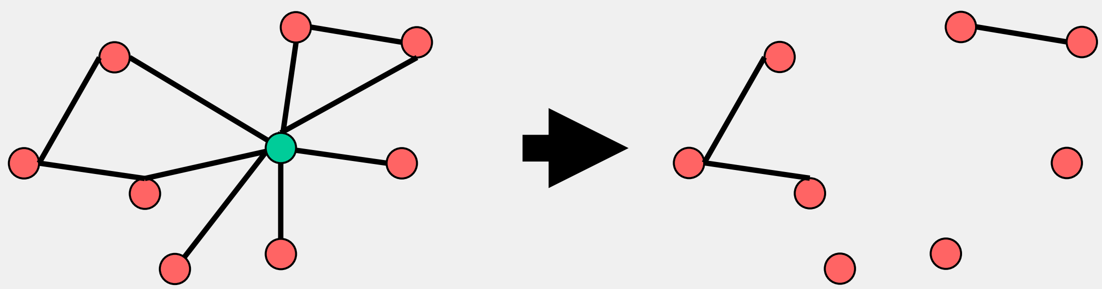
Image: Dartnell et al, FEBS Letters 579, 2005

Importance: What happens if I break it?

Node Deletion. Take the graph and delete a node and all its edges.



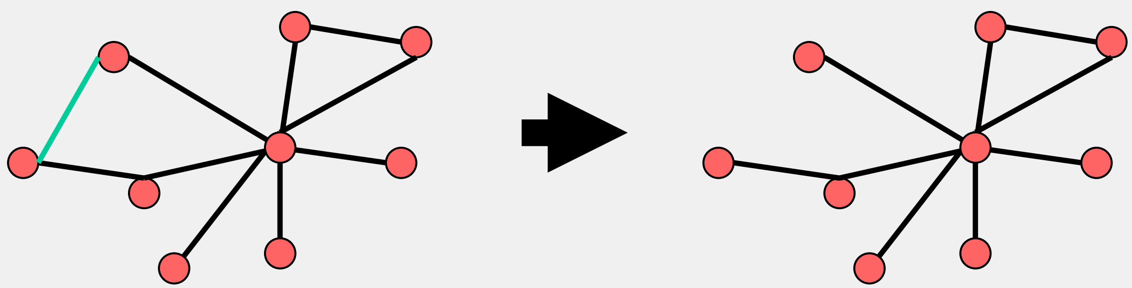
Node separation set: a subset of nodes whose deletion causes the number of components in the graph to increase



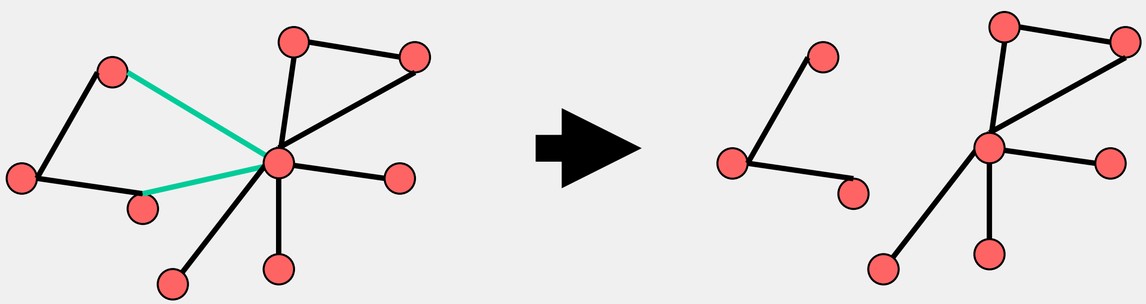
Mutations reducing p53 activity are present in over 50% of human tumours! (Haupt et al. 2003)

Importance: What happens if I break it?

Edge Deletion. Delete an edge (but not the nodes it joins)



Cut set: as for node separation set, but deleting edges

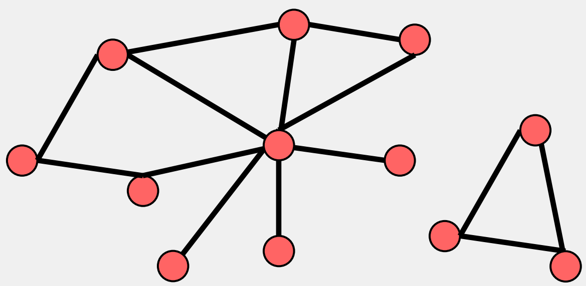


Network Robustness: how hard is it to break the network?

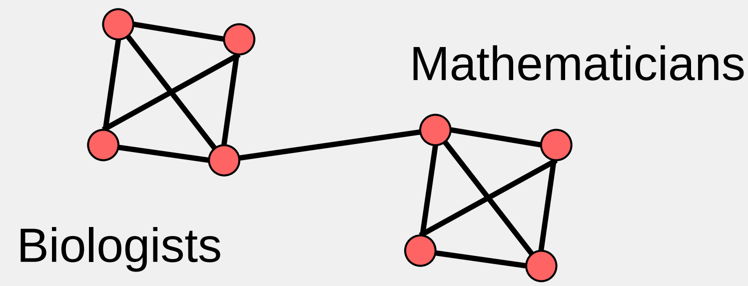
Delete a random node or edge: it is still connected?

What at the (functional) modules?

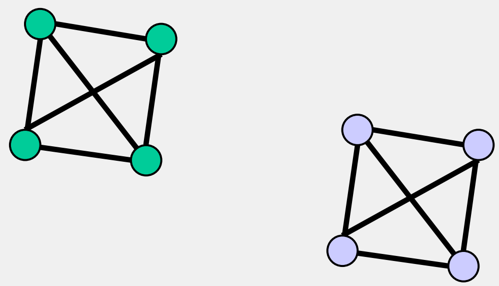
Components



But what about:



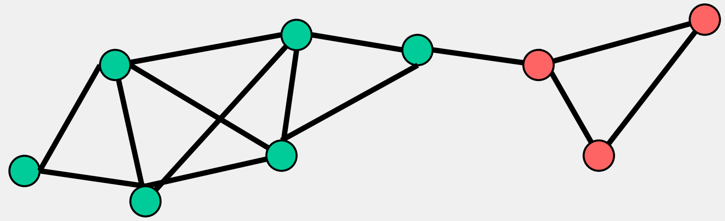
Clique. A subset of nodes, each pair joined by an edge



A maximal clique is contain in no larger clique

What at the (functional) modules?

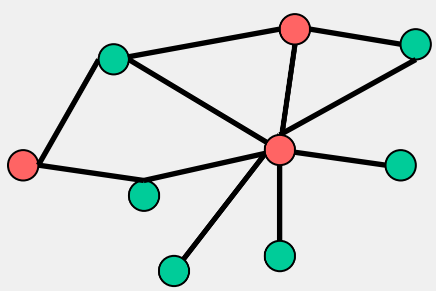
e-Near Clique. A subset of nodes such that a fraction of e pairs of nodes have an edge between them



10/15 – near clique

3-clique

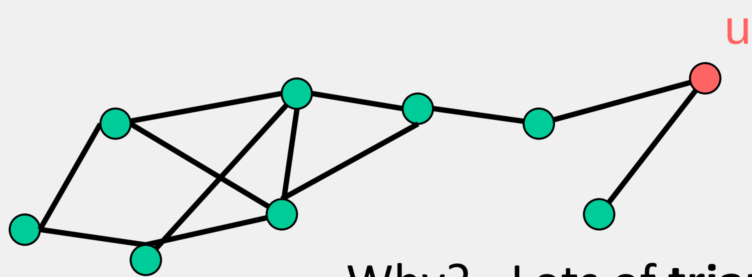
Co-Clique. A subset of nodes, no two joined by an edge



Green nodes are a co-clique

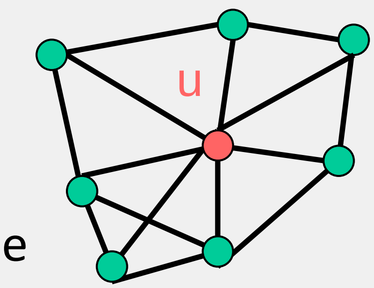
Are there modules? - Clustering Coefficient

How do we tell if a node u is in a cluster?



C_u = 0

C_u = 8/21



Why? - Lots of **triangles** on the node

- i.e. mutual connection

For a **node u** of **degree k**, where there are **e edges** between neighbours of u, define the ***cluster coefficient C_u*** as:

$$C_u = e / [k(k-1)/2]$$

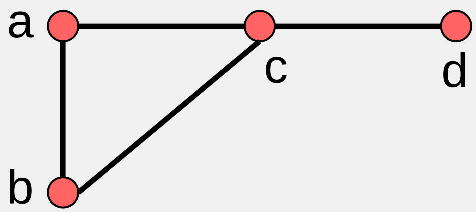
triangles on u (e)

Maximum possible # triangles on u (k(k-1)/2)

For a graph, then define the ***average cluster coefficient***

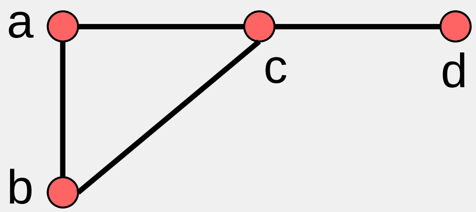
Getting around in a Graph

Path. A “walk” through the graph with no repeated edges



a-c-d

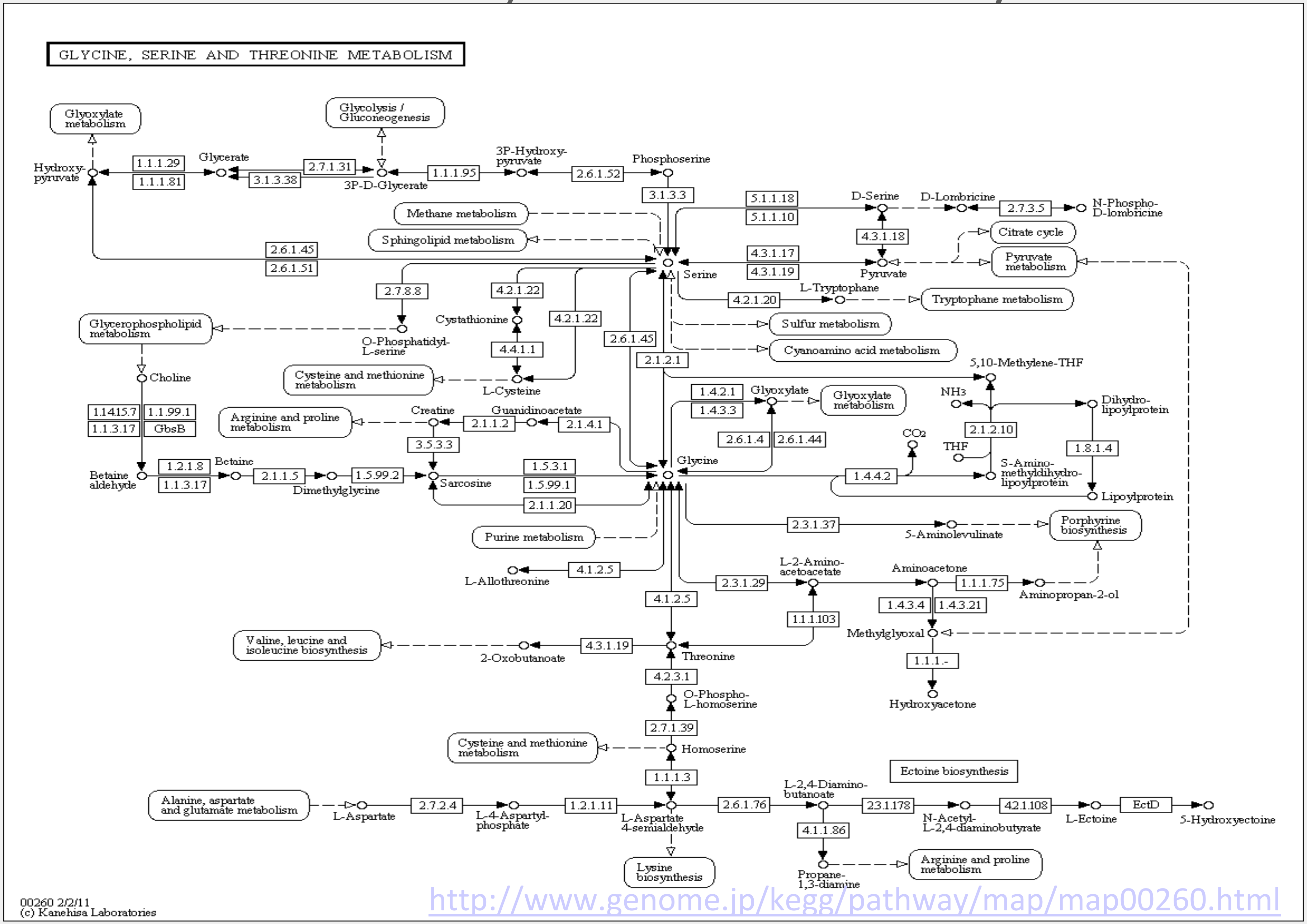
Cycle. A path that begins and ends at the same node



a-b-c-a

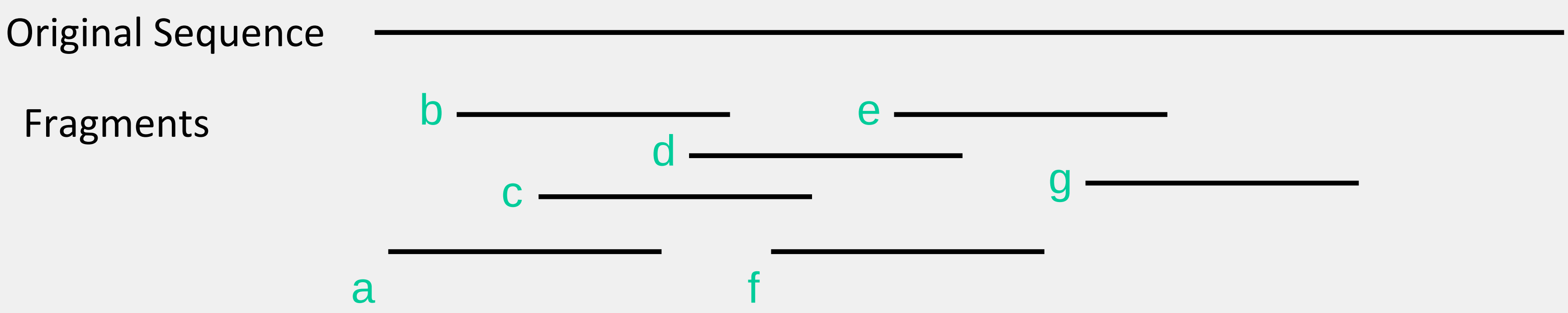
Connected. There is a path between any two nodes

For instance, Metabolic Pathways



http://www.genome.jp/kegg/pathway/map/map00260.html

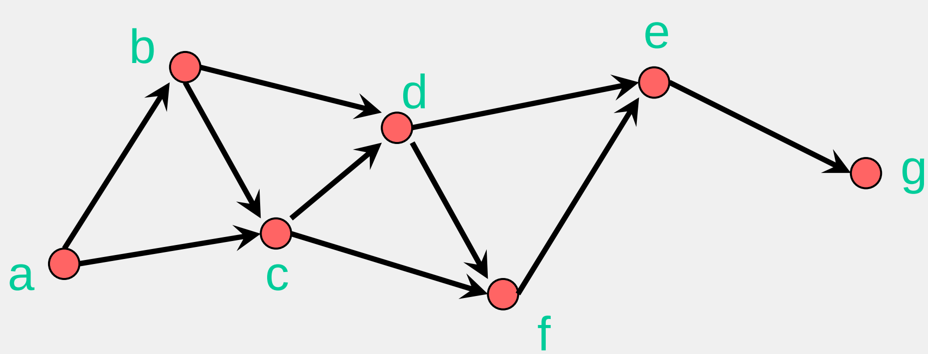
Path Example: Shotgun sequence reconstruction



Construct ***overlap graph***

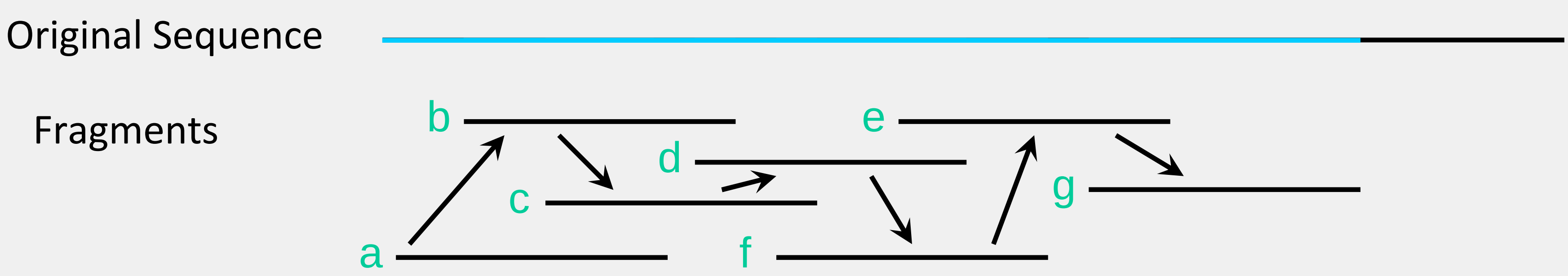
nodes: sequence fragments

edges: the **tail** of one fragments overlaps the **head** of another

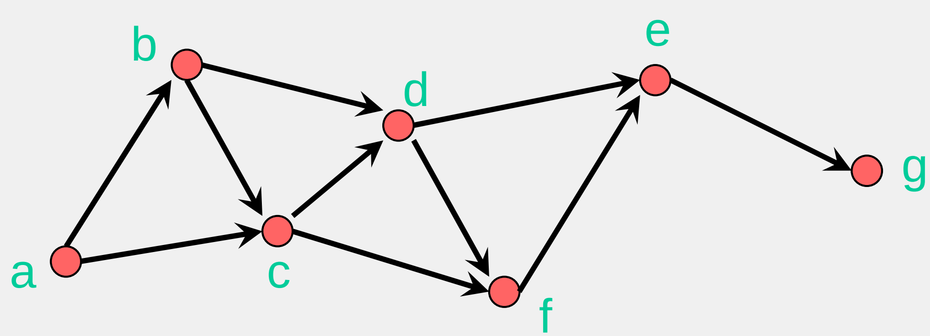


Warning: the above ignore all the awful details: sequencing errors, repeats, …

Hamiltonian Paths



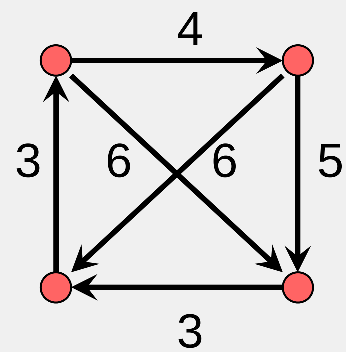
Hamiltonian Path: Visits every node exactly once

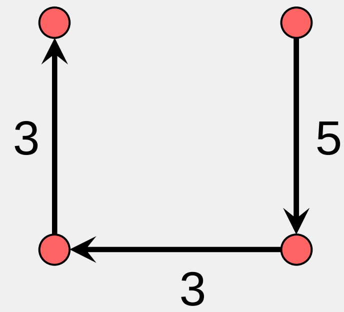


Edge Weights

But there might be **multiple Hamiltonian paths**

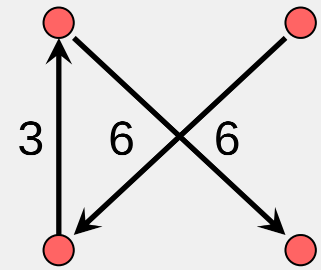
Which is "best"?





Total 11

or



Total 15

?

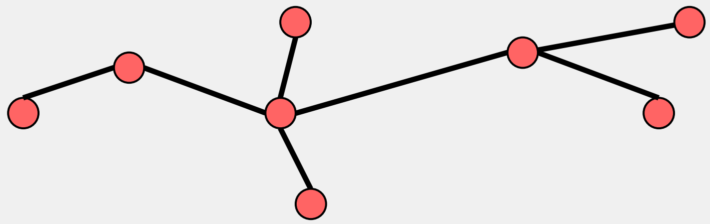
Use ***edge weights*** : amount of overlap between fragments

More overlap means a **shorter combined sequence**: better

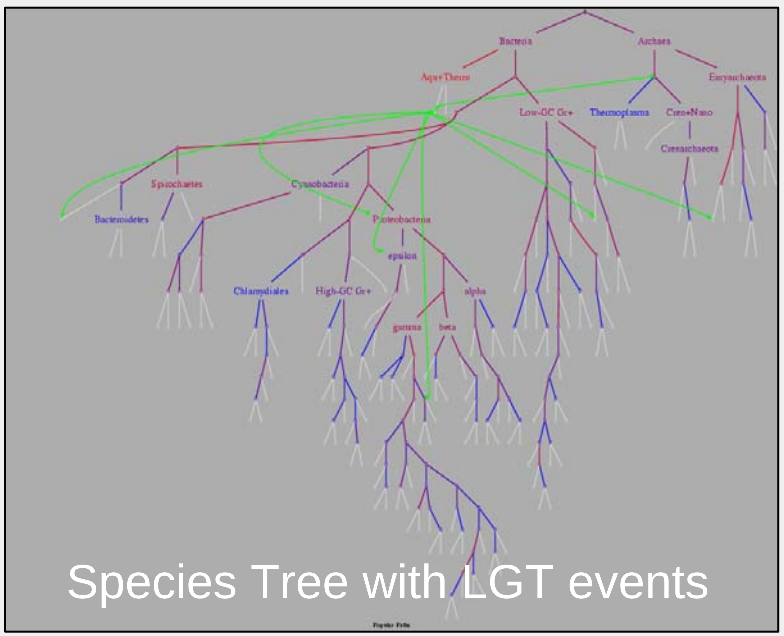
In fact this is just the "famous" **travelling salesman problem**

Trees and Hierarchical Structure

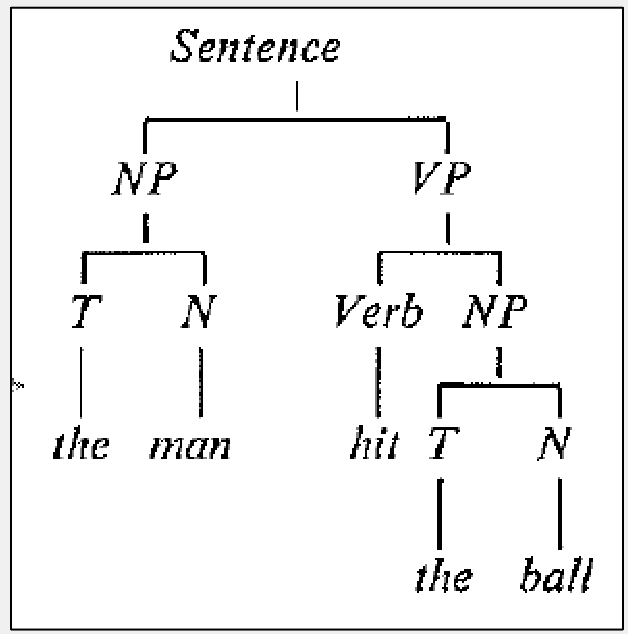
A ***tree*** is an undirected connected acyclic graph



A ***directed tree*** is a directed graph that would be tree if the directions were ignored



Species Tree with LGT events



Noam Chomsky, Syntactic Structures

Small World Networks

Stanley Milgram in 1967 "showed" social networks have "**six degrees of separation**" and other shocking experiments

Variations: Six degrees of Kevin Bacon, Erdös Number, Six degrees of Eric Clapton

Defining characteristics of small world networks

- Most nodes are not directly connected to each other
- Can get from between most pair of nodes in few steps

[For N nodes, average pair distance proportional to Log(N)]

Watts & Strogatz (Nature, 1998): constructed networks with small average shortest path & high clustering coefficient

Properties and Examples of Small World Networks

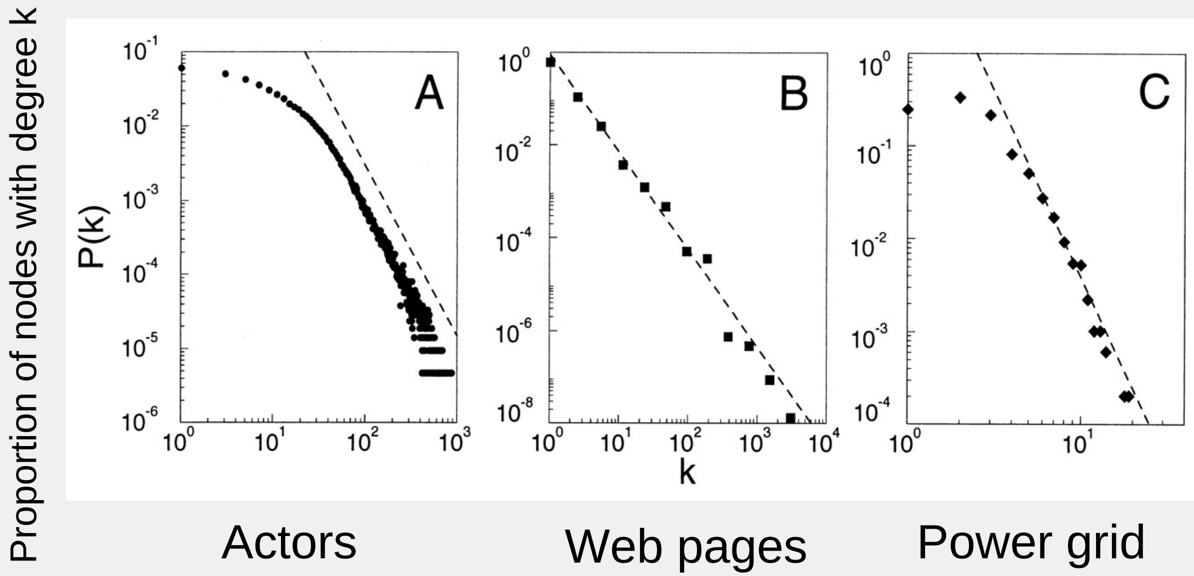
- Lots of **hubs**
- Often have **cliques** and near cliques
- Said to be **robust** to perturbation (though hubs are vulnerable)

For example

- Transcriptional networks
- Metabolic networks
- Protein interaction networks
- Neural connections
- You name it, it is a small world!

Scale Free Networks

- Barabasi & Albert (Science, 1999)
- Have **power law** distribution of degrees: $P(k) \sim k^{-\alpha}$



- Can be constructed by **preferential attachment**
- They are "**ultra-small worlds**": Log(Log(N)) steps (Cohen & Havlin, 2003)

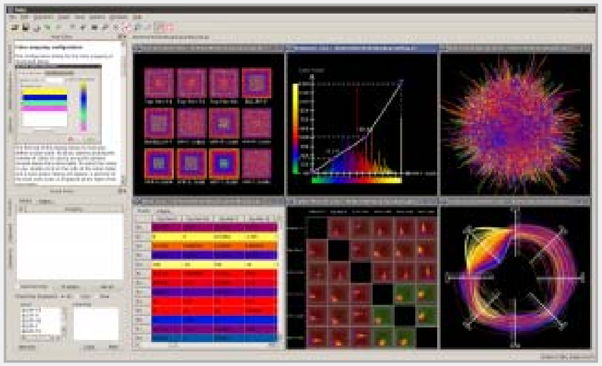
Software for Graph Exploration & Visualisation

See:

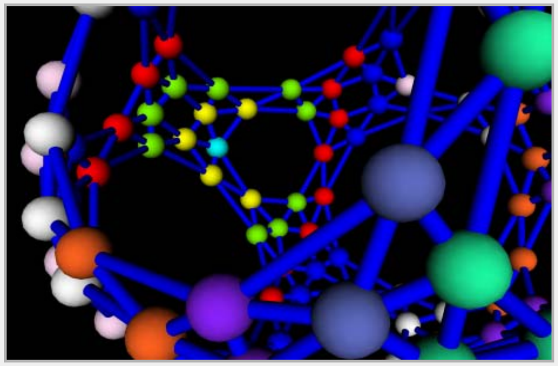
http://www.google.com/Top/Science/Math/Combinatorics/Software/Graph_Drawing/

For a selection of tools

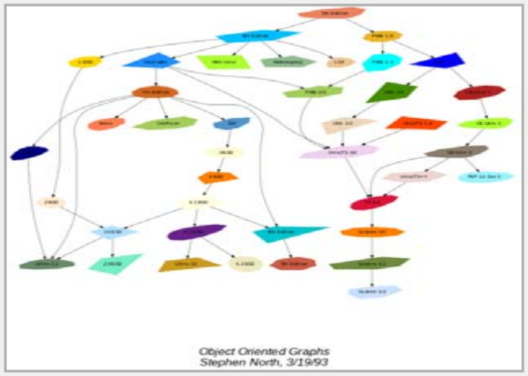
Tulip: 2D and 3D interactive visualisation of graphs



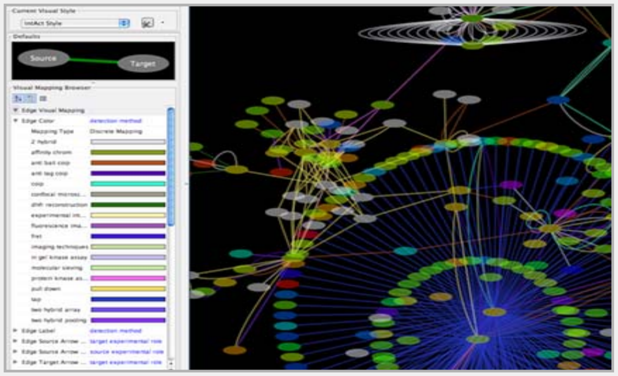
Pajek: graph algorithms and visualisation



GraphViz: sophisticated graph layout



Cytoscape: viz. interaction networks/pathways



Matlab (MatlabBGL): Graph algorithms & metrics

```
>> [d ft dt pred] = dfs(A)
>> [d dt pred] = bfs(A);

>> [d pred] = dijkstra_sp(A,u);
>> [d pred] = bellman_ford_sp(A,u);
>> [d pred] = dag_sp(A,u);
>> D = johnson_all_sp(A);
>> D = floyd_warshall_all_sp(A);

>> T = kruskal_mst(A);
>> T = prim_mst(A);

>> cc = components(A)
>> [a C] = biconnected_components(A);

>> [flow cut R F] = max_flow(A,u,v);

>> print_func = @(str) @(u) fprintf('called %s(%s)\n', str, char(labels(u)));
>> breadth_first_search(A,1,struct('examine_vertex',print_func('examine_vertex')));

>> [d pred rank] = astar_search(A,u,heuristic_func);
```

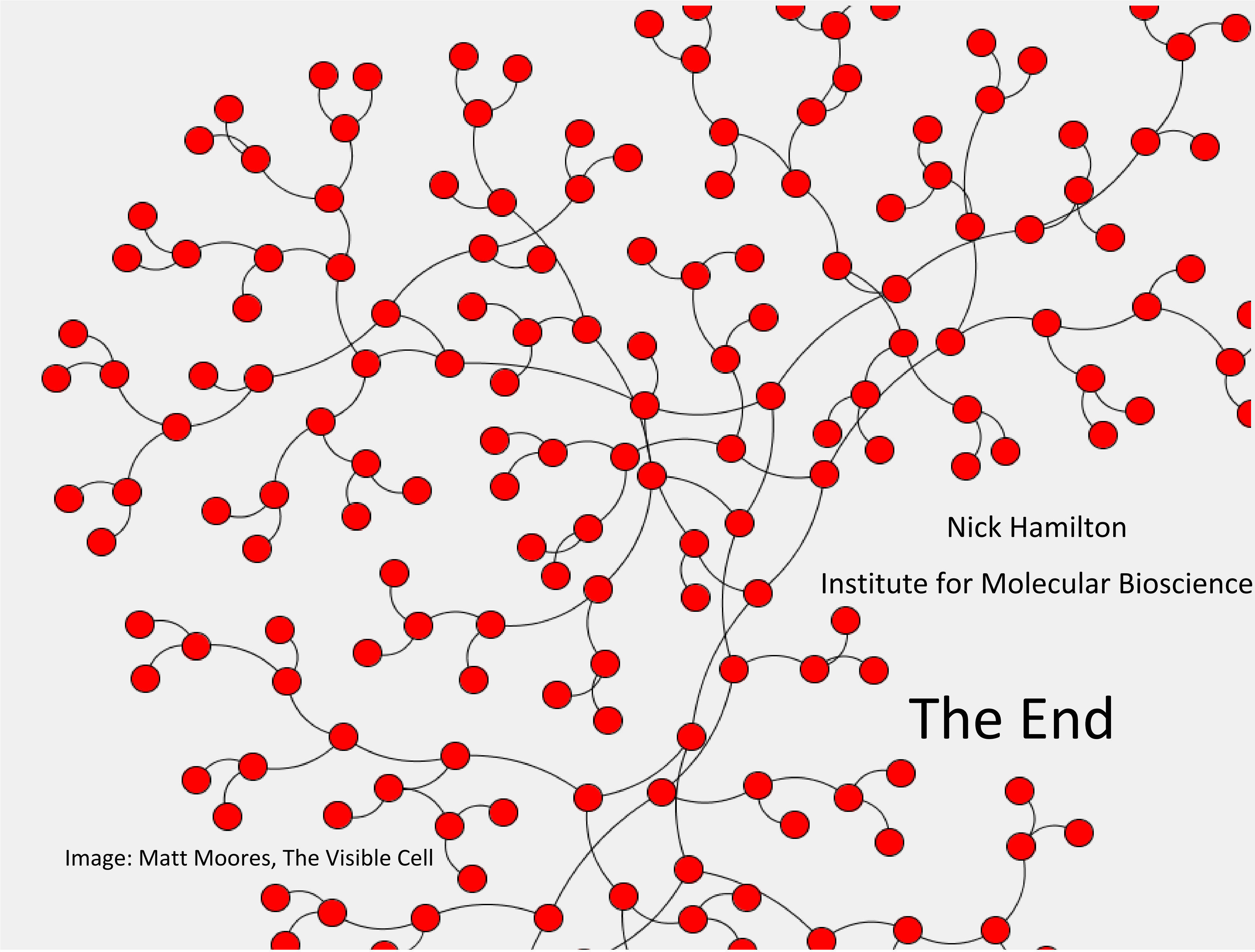
images nicked from the respective websites

Further Reading

- Mark Buchanan, *Small World: Uncovering Nature's Hidden Networks*
- Albert & Barabasi, Emergence of scaling in random networks, *Science* **286**(286):509-512 , 1999
- Watts, & Stogatz, Collective dynamics of small world networks, *Nature* **393**:440-444, 1998

Summary

- ***Node Degree***: Which are the most important bits?
- ***Node & Edge Cuts***: What happens when I break it? Robustness
- ***Cliques & Clusters***: What are the functional modules?
- ***Cluster Coefficient***: Are there functional modules?
- ***Paths & Edge Weights***: Getting around in a graph
- ***Graph algorithms***: Are usually hard
- ***Trees***: Are ubiquitous
- ***Small world and scale free graphs***: Are popular
- ***Software***: There is some



Nick Hamilton

Institute for Molecular Bioscience

The End

Image: Matt Moores, The Visible Cell