Introduction to graph theory and molecular networks

Sushmita Roy

sroy@biostat.wisc.edu

Computational Network Biology

Biostatistics & Medical Informatics 826

https://compnetbiocourse.discovery.wisc.edu

Sep 11th 2018

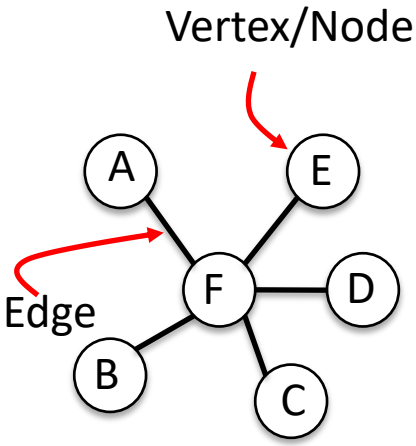
Some of these materials are from Introduction to Bioinformatics, BMI/CS 576.

Goals for today

- Introductory Graph theory
- Molecules of life
- Different types of molecular networks

A network

- Describes connectivity patterns between the parts of a system
  - Vertex/Nodes: parts, components
  - Edges/Interactions/links: relationships
- Edges can have signs, directions, and/or weight
- A network is represented as a graph
  - Node and vertex are used interchangeably
  - Edge, link, and interaction are used interchangeably



Notation

- u, v, v_i, s, t: A vertex of a graph
- G: A graph defined by a tuple (V,E)
- V: set of vertices, {v_1,.., v_N} where N is the number of nodes
- E: set of edges, each edge is defined by a pair (v_i, v_j) representing a link between these two vertices
- For an edge (v_i, v_j), v_j is said to be adjacent to v_i

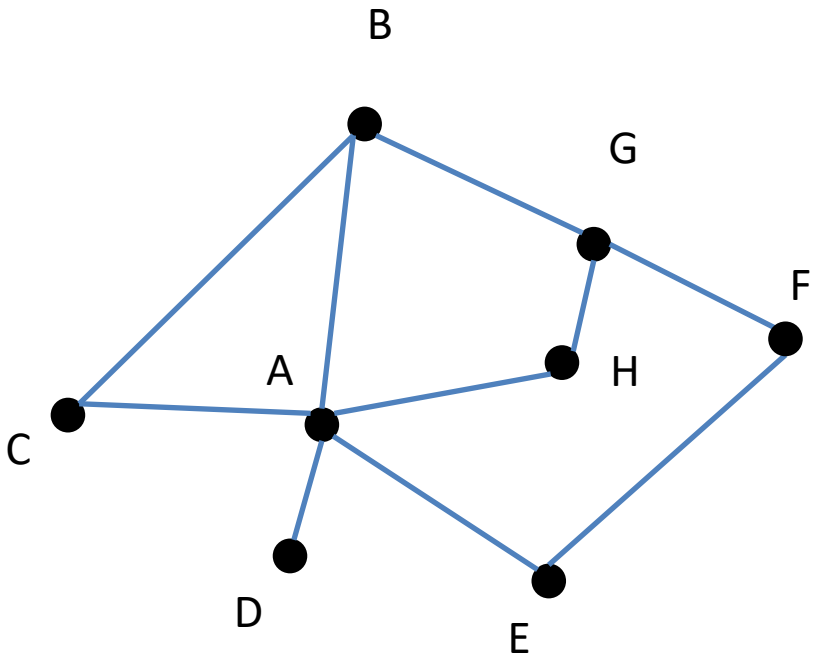
A few graph-theoretic concepts

- Representing a graph
  - Directed/Undirected/Weighted graph
- Connectivity on a graph
- Subnetworks/subgraphs
- Traversal on a graph

Representing a graph

- Adjacency matrix
- Adjacency list

Adjacency matrix



Matrix-based representation; dense

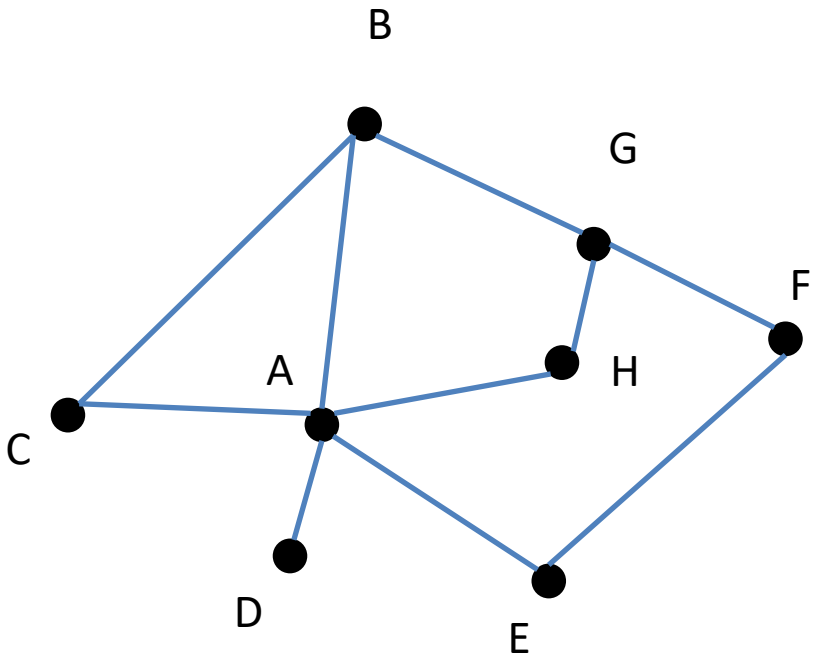
	A	B	C	D	E	F	G	H
A	0	1	1	1	1	0	0	1
B	1	0	1	0	0	0	1	0
C	1	1	0	0	0	0	0	0
D	1	0	0	0	0	0	0	0
E	1	0	0	0	0	1	0	0
F	0	0	0	0	1	0	1	0
G	0	1	0	0	0	1	0	1
H	1	0	0	0	0	0	1	0

V={A, B, C, D, E, F, G, H}

E= {(A,B),(A,C),(A,D),(A,E),(A,H),(B,C),(B,G),(D,A),(E,A),(E,F),(F,G),(G,H)}

Adapted from Barabasi & Oltvai, Nature Reviews Genetics 2004

Adjacency list



List based representation; sparse, space efficient

A → B → C → D → E → H

B → A → C → G

C → A → B

D → A

E → A → F

F → G → E

G → B → F → H

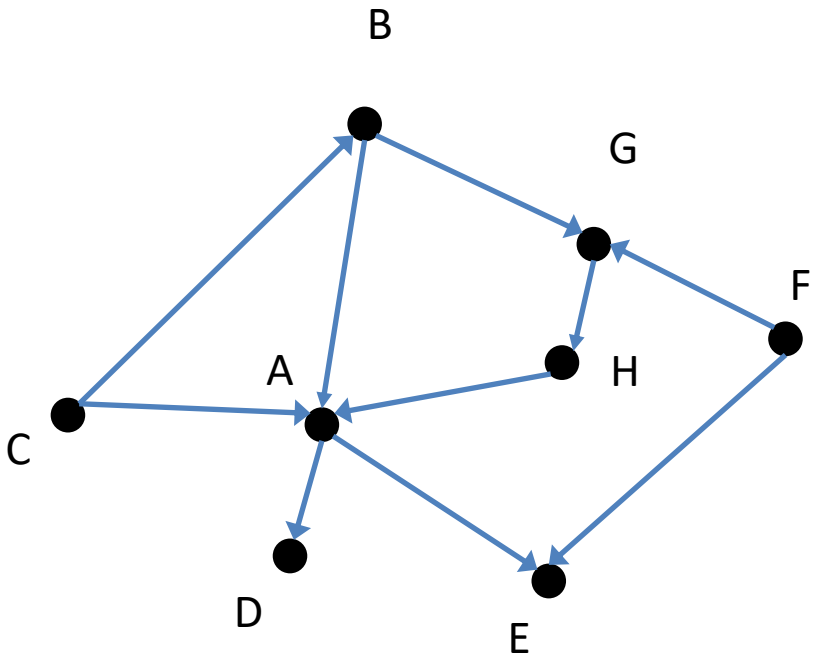
H → A → G

Adjacency list versus matrix

- Adjacency list
- Space efficient
- Asking if there is an edge can be slow
- Preferred when the graph is sparse

- Adjacency matrix
- Very fast to ask if there is an edge
- Storage is $N^{2,}$ where N is the number of nodes in the graph
- Preferred when the graph is dense

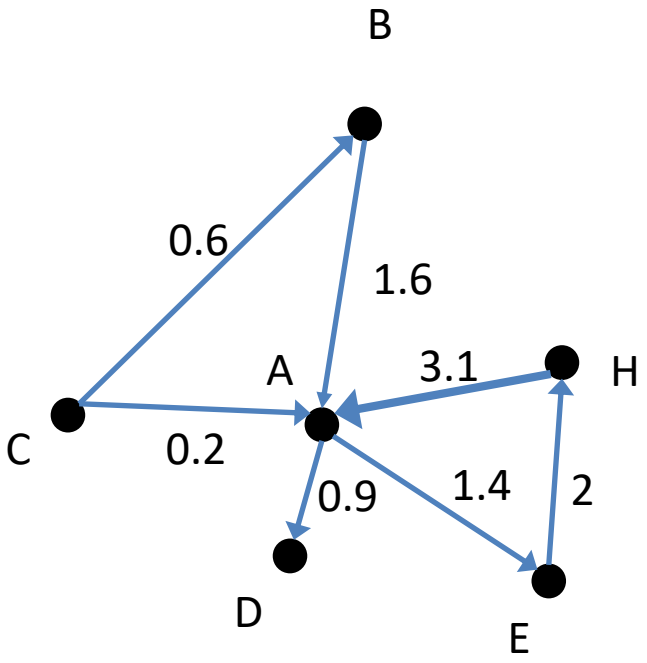
Directed graphs



	A	B	C	D	E	F	G	H
A	0	0	0	1	1	0	0	0
B	1	0	0	0	0	0	1	0
C	1	1	0	0	0	0	0	0
D	0	0	0	0	0	0	0	0
E	0	0	0	0	0	0	0	0
F	0	0	0	0	1	0	1	0
G	0	0	0	0	0	0	0	1
H	1	0	0	0	0	0	0	0

- Edges have directionality on them
- Adjacency matrix is no longer symmetric

Weighted graphs

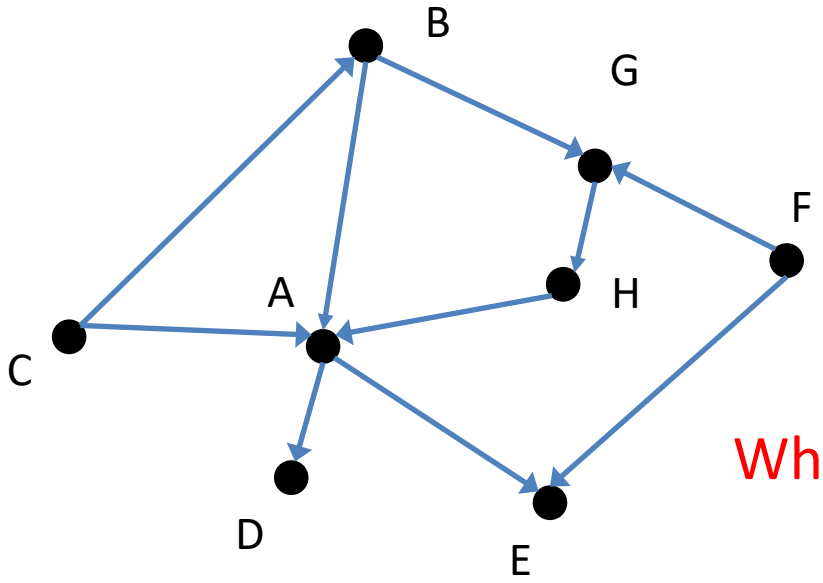


	A	B	C	D	E	H
A	0	0	0	0.9	1.4	0
B	1.6	0	0	0	0	0
C	0.2	0.6	0	0	0	0
D	0	0	0	0	0	0
E	0	0	0	0	0	2
H	3.1	0	0	0	0	0

- Edges have weights on them.
- We can have directed and undirected weighted graphs.
- The example shown is that of a directed weighted graph

Node degree

- Undirected network
  - Degree, k: Number of neighbors of a node
- Directed network
  - In degree, k_{in}: Number of incoming edges
  - Out degree, k_{out}: Number of outgoing edges



- In degree of B is 1
- Out degree of A is 2

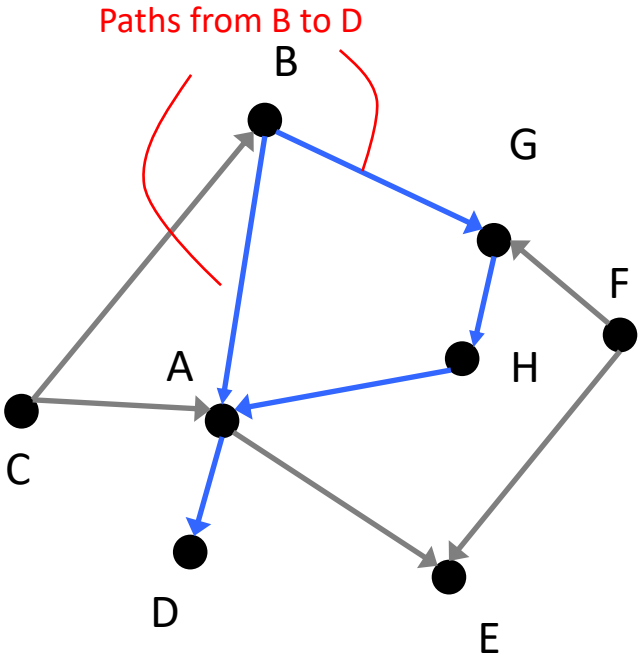
What is the out degree of E?

Paths and cycles

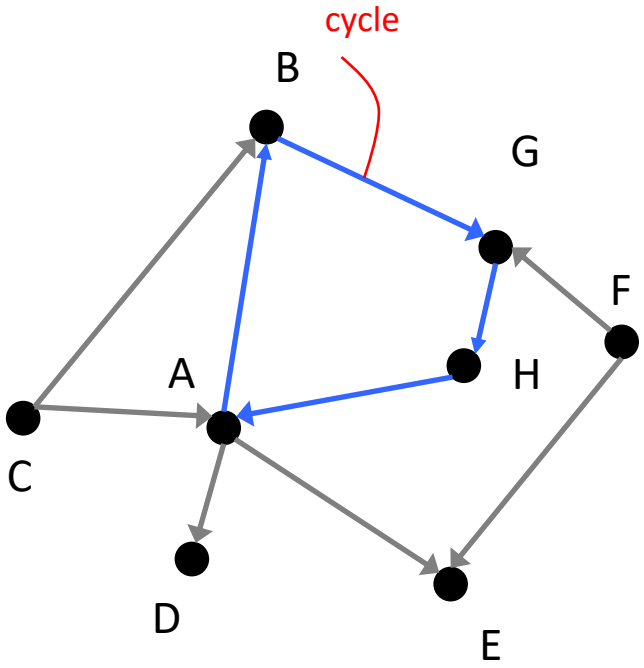
- **Path**:
  - a path from vertex s to t in G is a sequence of vertices $(v_0,\ldots,v_k)$ such that $s=v_0$ and $t=v_k$ and (v_i,v_{i+1}) are edges in E.
  - A path is **simple** if there are no repetitions of a vertex.
- **Reachable**: A vertex t is **reachable** from vertex s if there is a path from s to t
- **Path length**: The total number of edges in a path
- **Shortest path**: The path between two vertices with the shortest path length
- **Cycle**: A path where v_0 and v_k are the same

Paths and cycles

Paths



A cycle



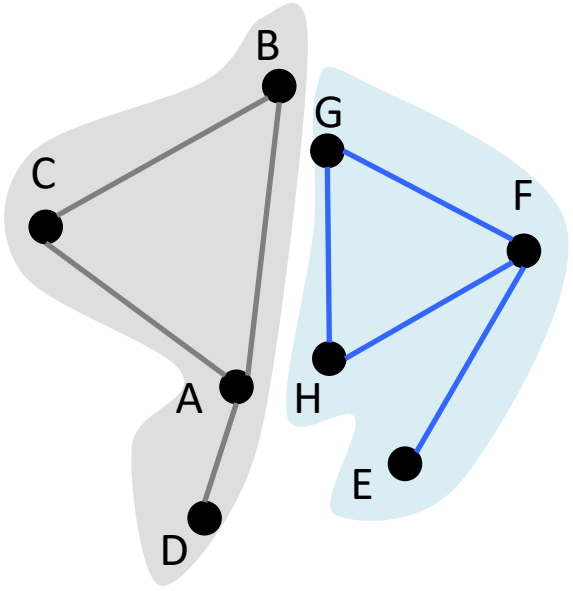
- There are two paths from B to D
- Which is the shortest path?

Connected components

- **Connected components**: The set of vertices that are reachable from one node to another
- **Strongly connected components**: The set of vertices that are reachable from one vertex to another in a directed graph.
- **Connected graph**: An undirected graph is connected if every pair of vertices is connected by a path
- **Strongly connected graph**: A directed graph where all vertices are reachable from each other

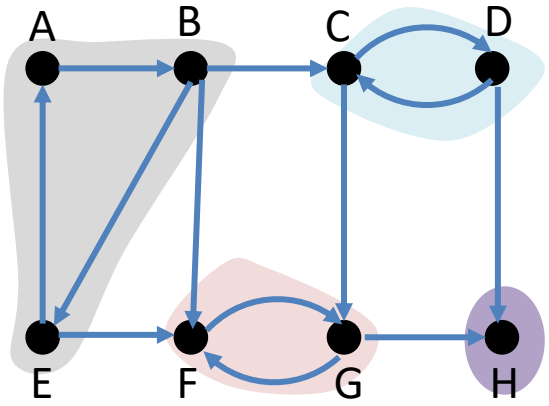
Connected components

Connected components in an undirected graph



Two connected components

Connected components in a directed graph



Four strongly connected components

Special types of graphs

- **Complete graph**: an undirected graph where all vertices are neighbors of each other
- **Bipartite graph**: a graph $G=(V,E)$ whose vertex set is divided into to sets, V_1 and V_2 such that for every edge (u,v) in E, u is in V_1 and v is in V_2
- **Directed acyclic graph**: A directed graph that has no cycles
- **Tree**: A graph where every pair of vertices are connected by a unique simple path

Subgraph

- A graph $G'=(V,'E')$ is a subgraph of a graph $G=(V,E)$ if $V' \subseteq V$ (V' is a subset of V) and $E' \subseteq E$.
- Given a subset $V' \subseteq V$, $G'=(V',E')$ is a subgraph **induced** by V' if $E'=\{(u,v) \in E; u, v \in V'\}$.
- We will use subgraph and subnetwork interchangeably
- Subgraphs we just saw:
  - Cycle, path, connected component

Common graph traversal algorithms

- Breadth-first search
- Depth-first search

Breadth-first search (BFS)

- Given a graph $G=(V,E)$ and a source vertex s, BFS explores G to
  - find every vertex that is reachable from s
  - Computes the shortest path length from s to all reachable vertices
- BFS explores all vertices at a particular distance before the next
  - So it uniformly accesses all vertices across the “breadth” of the frontier

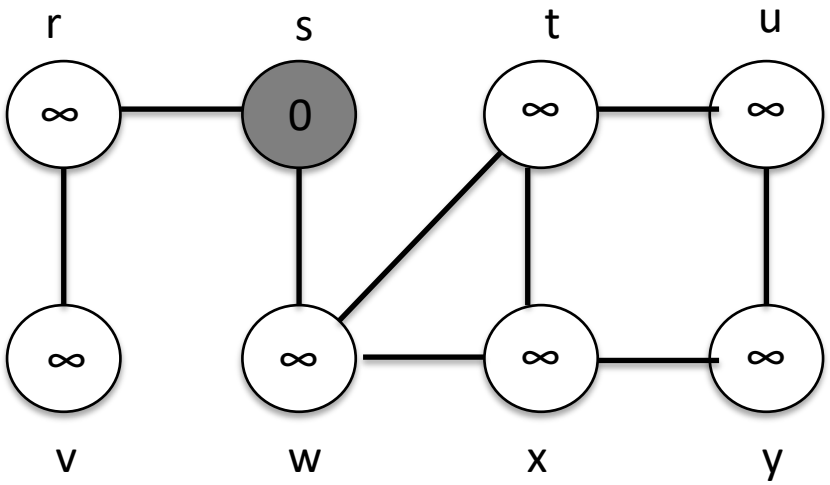
Breadth first search algorithm sketch

- We will need two types of data structures
  - Three arrays: color *color*, distance *d*, predecessor π
  - A queue, queue *Q,* used for doing the traversal of nodes in a first in first out order
- Color: A node is white, black or gray
  - White: as yet undiscovered
  - Black: all neighbors have been discovered
  - Gray: some neighbors may not have been discovered
- Distance keeps track of the shortest path length
- Predecessor is used to produce the path

Breadth first search algorithm

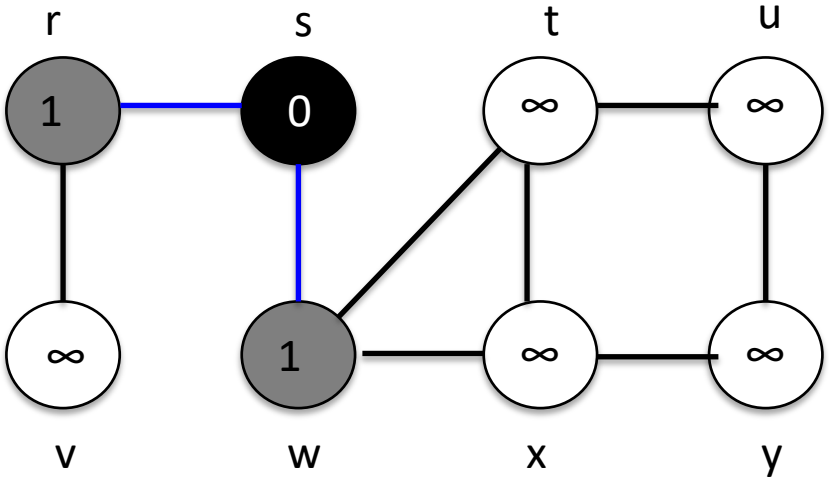
```
1:  procedure BFS(G,s)
2:      for each vertex u ∈ V(G) \ {s} do
3:          color[u]=WHITE
4:          π[u]=NIL
5:          d[u] = ∞
6:      end for
7:      color[s]=GRAY
8:      d[s] = 0
9:      π[s]=NIL
10:     Q = ∅
11:     Push(Q, s)
13:     while Q ≠ ∅ do
14:         u=Pop(Q)
15:         for each v ∈ Adj[u] do
16:             if color[v]==WHITE then
17:                 color[v]=GRAY
18:                 d[v] = d[u] + 1
19:                 π[v] = u
20:                 Push(Q, v)
21:             end if
22:         end for
23:         color[u]=BLACK
24:     end while
25: end procedure
```

Breadth first search example



Q={s}

Before while loop

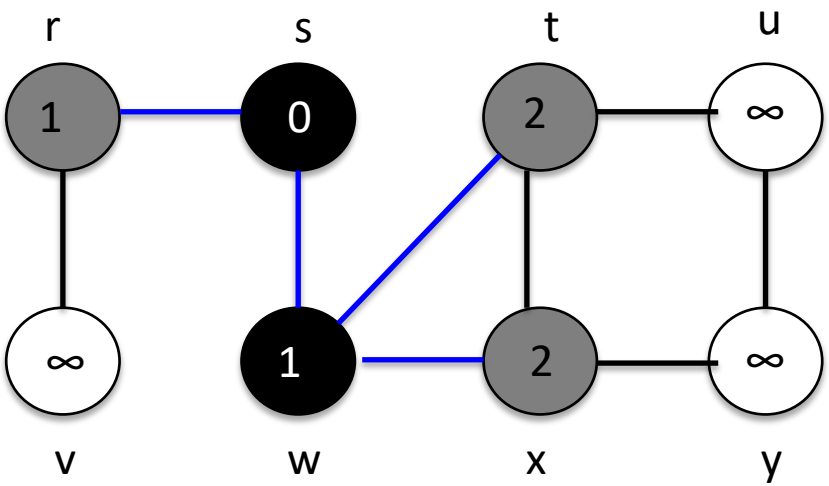


Q={w,r}

Iteration 1

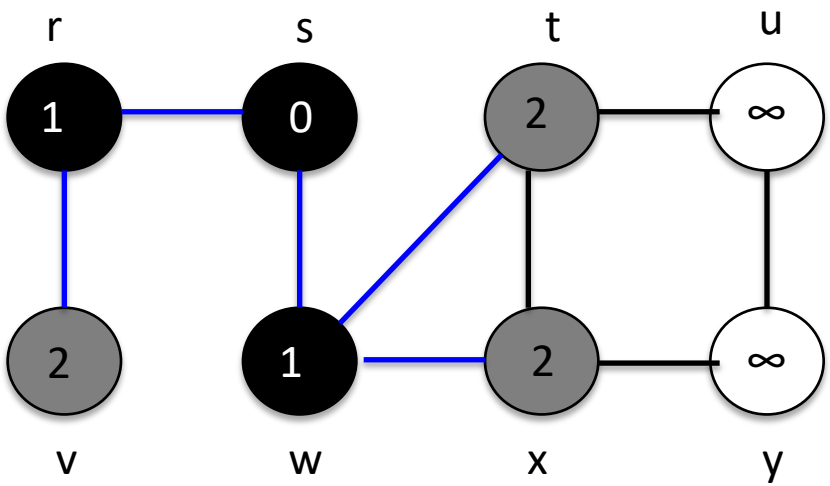
Adapted from *Introduction to Algorithms*, 2nd Edition, Cormen, Leiserson, Rivest, Stein

Breadth first search continued



Q={r,t,x}

Iteration 2



Q={t,x,v}

Iteration 3

Depth first search

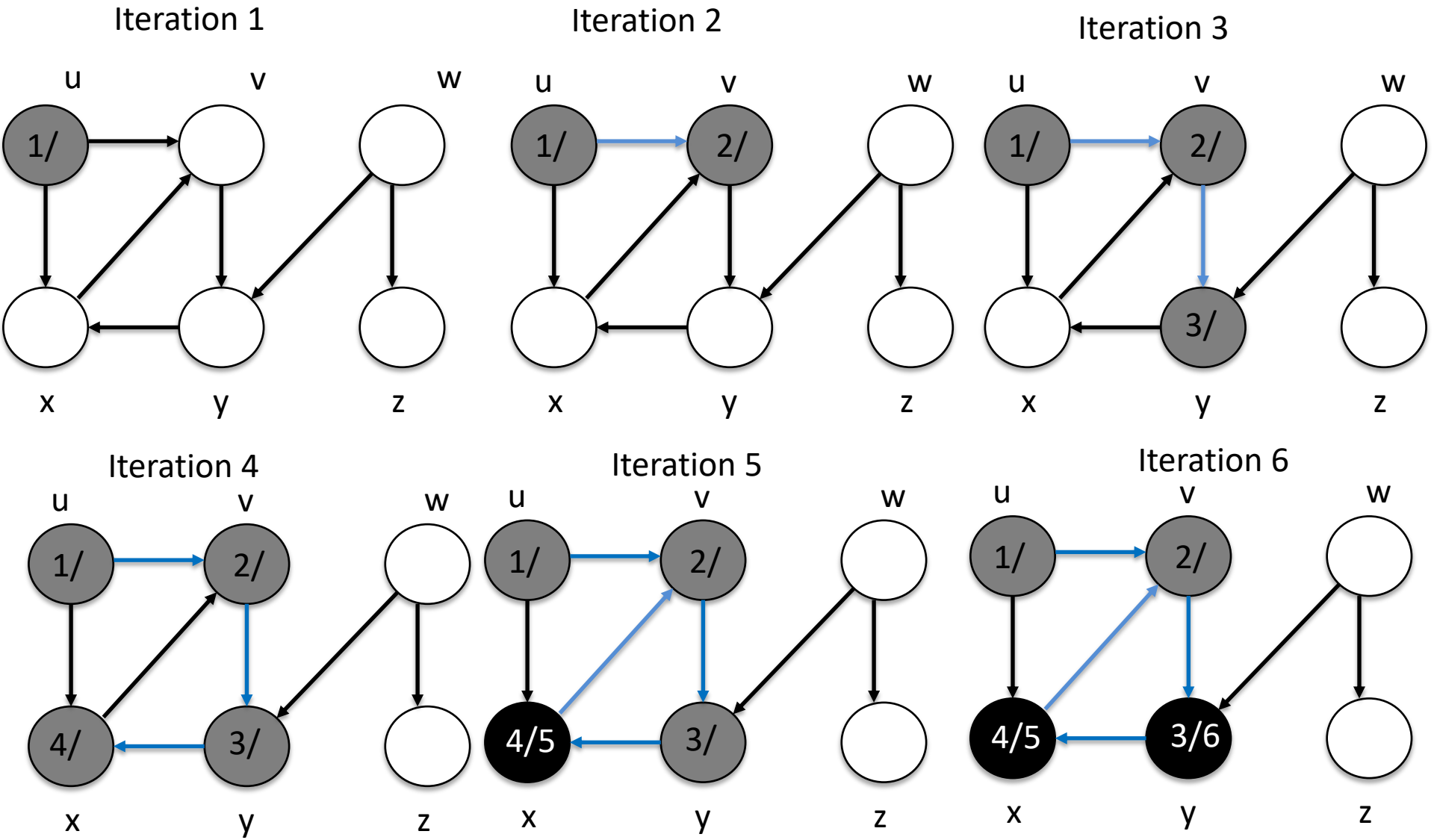
- Searches deeper in the graph whenever possible
- Edges are explored from the most recently discovered vertex with unexplored edges leaving it
- DFS is used for ”topological sort” and to find “strongly connected components”
- Like BFS needs color, predecessor
- Additionally stores start (d) and end time (f) of a node’s discovery

Depth first search algorithm

```
1:  procedure DFS(G)
2:      for each vertex u ∈ V(G) do
3:          color[u]=WHITE
4:          π[u]=NIL
5:      end for
6:      time=0
7:      for each vertex u ∈ V(G) do
8:          DFS-VIST(u)
9:      end for
10: end procedure
```

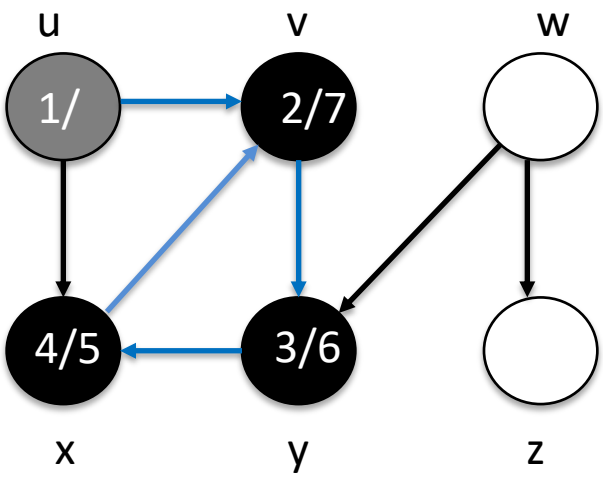
```
1:  procedure DFS-VISIT(u)
2:      color[u]=GRAY
3:      time=time+1
4:      d[u]=time
5:      for each vertex v in Adj(u) do
6:          if color[v]=WHITE then
7:              π[v] = u
8:              DFS-VISIT(v)
9:          end if
10:     end for
11:     color[u]=BLACK
12:     time=time+1
13:     f[u]=time
14: end procedure
```

Depth first search example

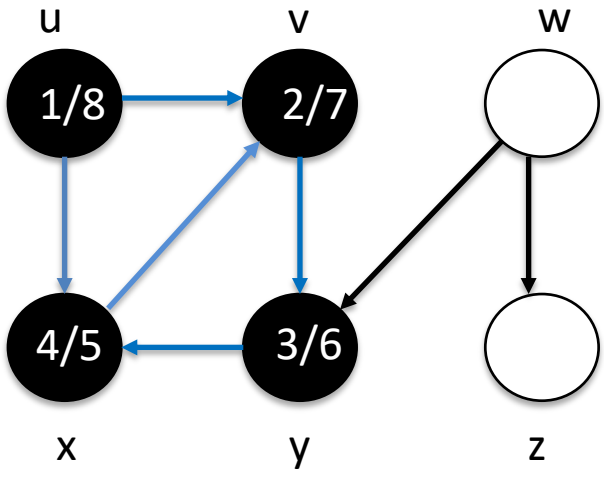


Depth first search example

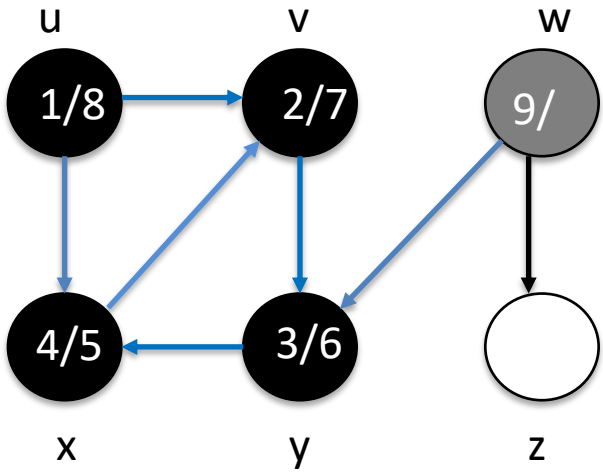
Iteration 7



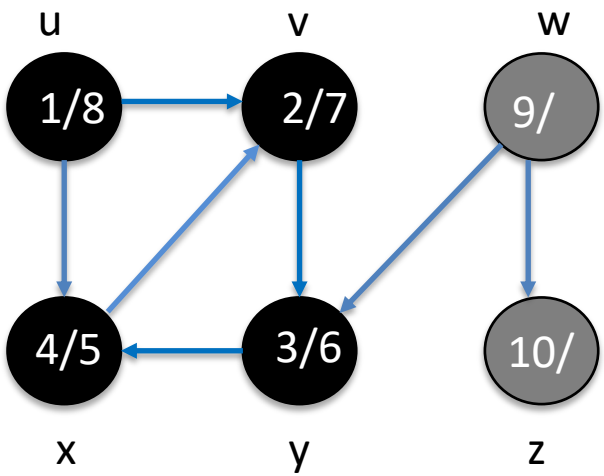
Iteration 8



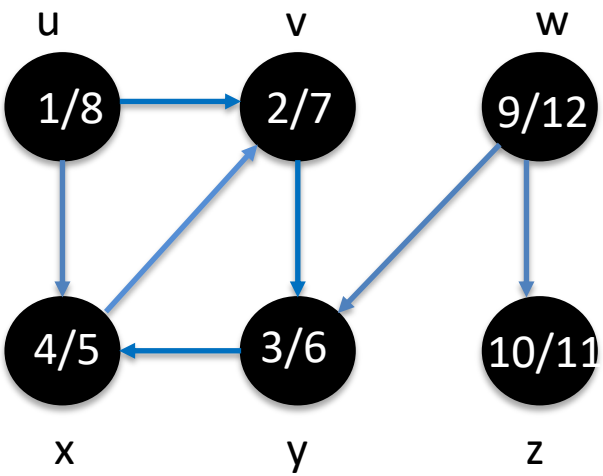
Iteration 9



Iteration 10



Final iteration



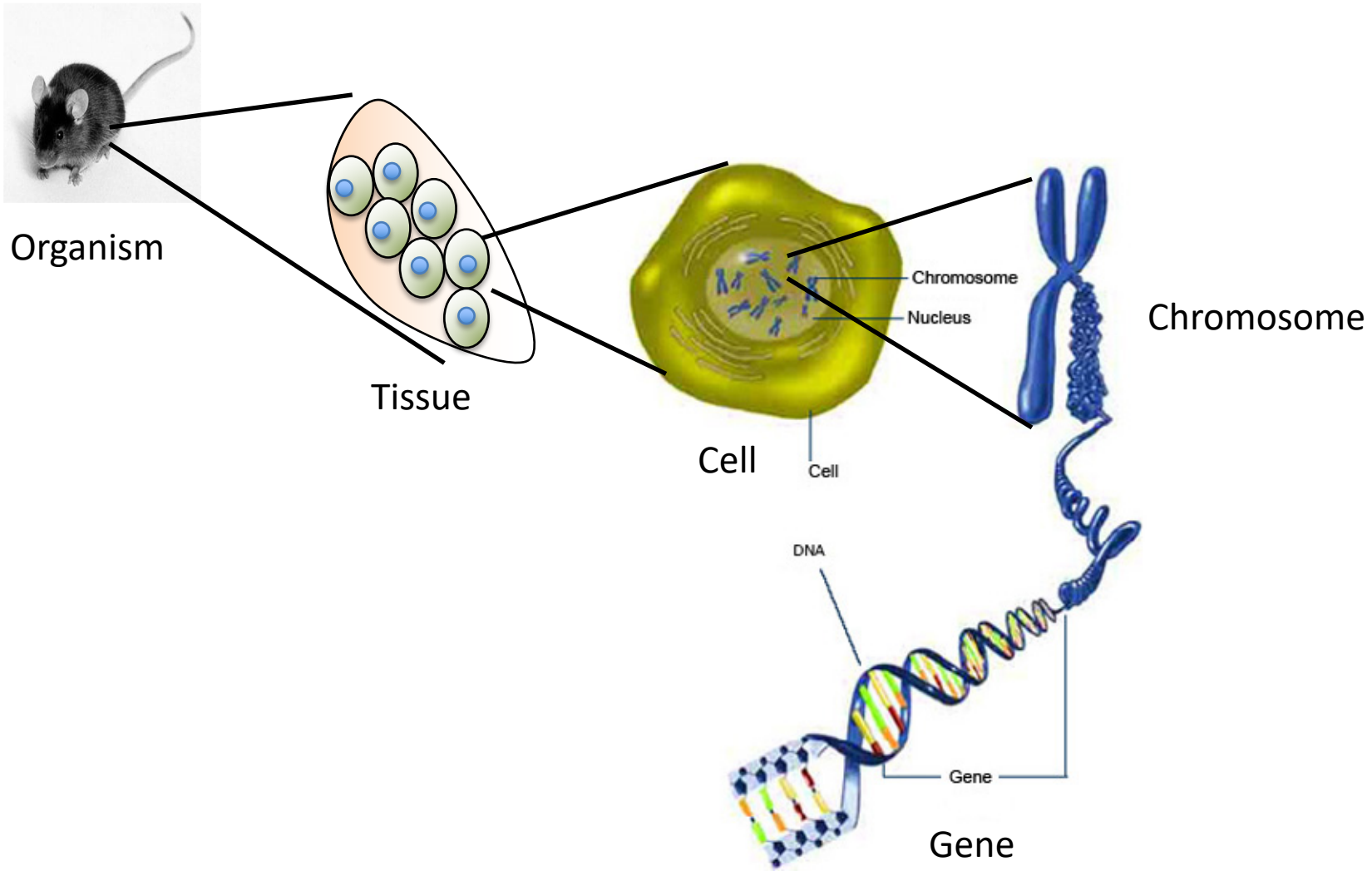
Take away points

- Adjacency lists and matrices are used to represent and analyze graphs
- Definitions of paths, cycles, connected components
- Breadth-first search
- Depth-first search

Goals for today

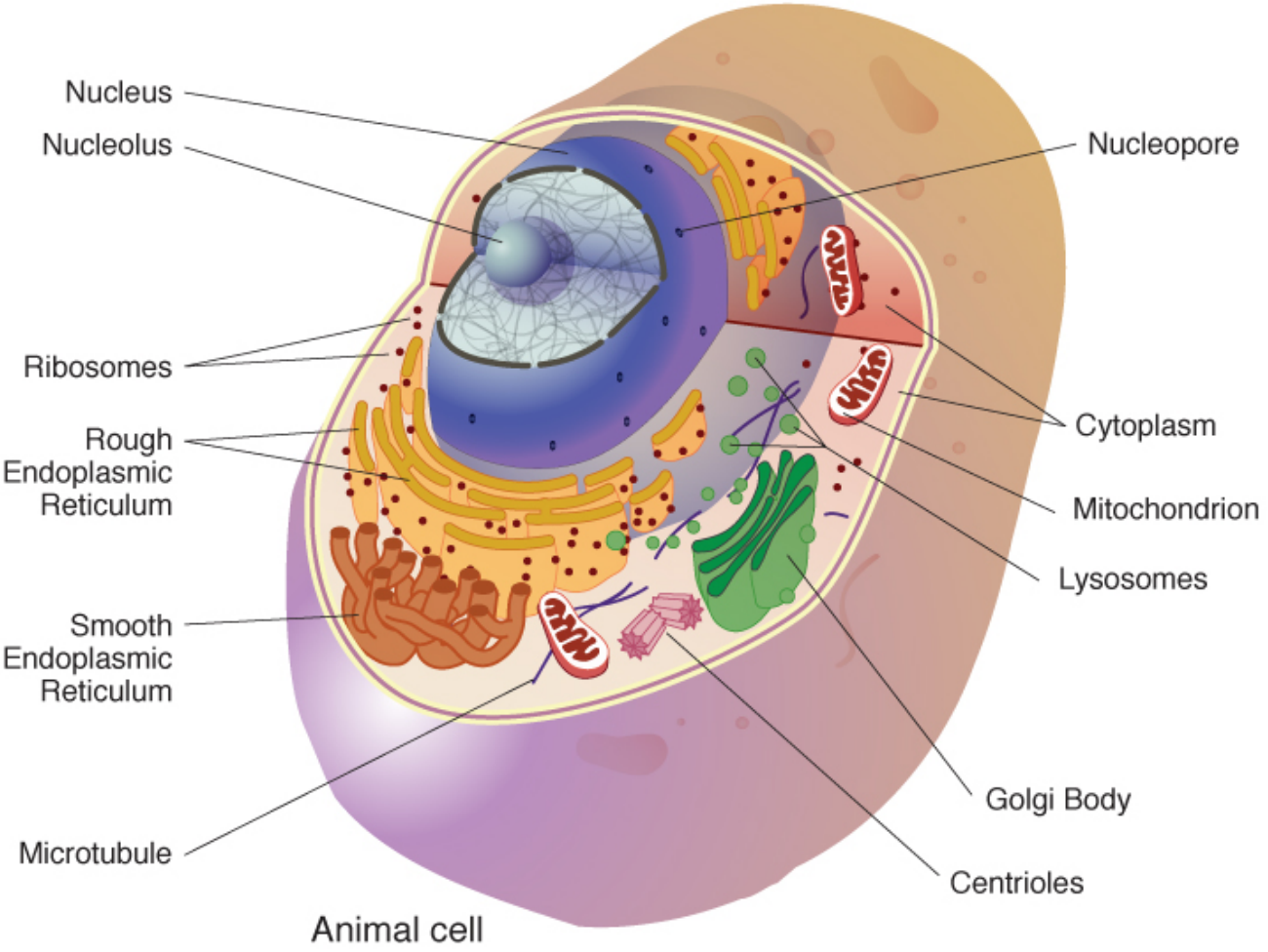
- Introductory Graph theory
- **Molecules of life**
- Different types of molecular networks

Organization of biological information



http://publications.nigms.nih.gov/thenewgenetics/chapter1.html

An animal cell



http://www.genome.gov/Glossary/index.cfm?id=25

Molecules of life

- Deoxyribonucleic acid (DNA)
- Ribonucleic acid (RNA)
  - Messenger RNA (mRNA)
    - Makes proteins
  - Non-coding RNA (ncRNA)
- Proteins
- Metabolites
- While DNA is mostly static, RNA, proteins, metabolites change between cell types, tissues, environmental conditions

Deoxyribonucleic acid (DNA)

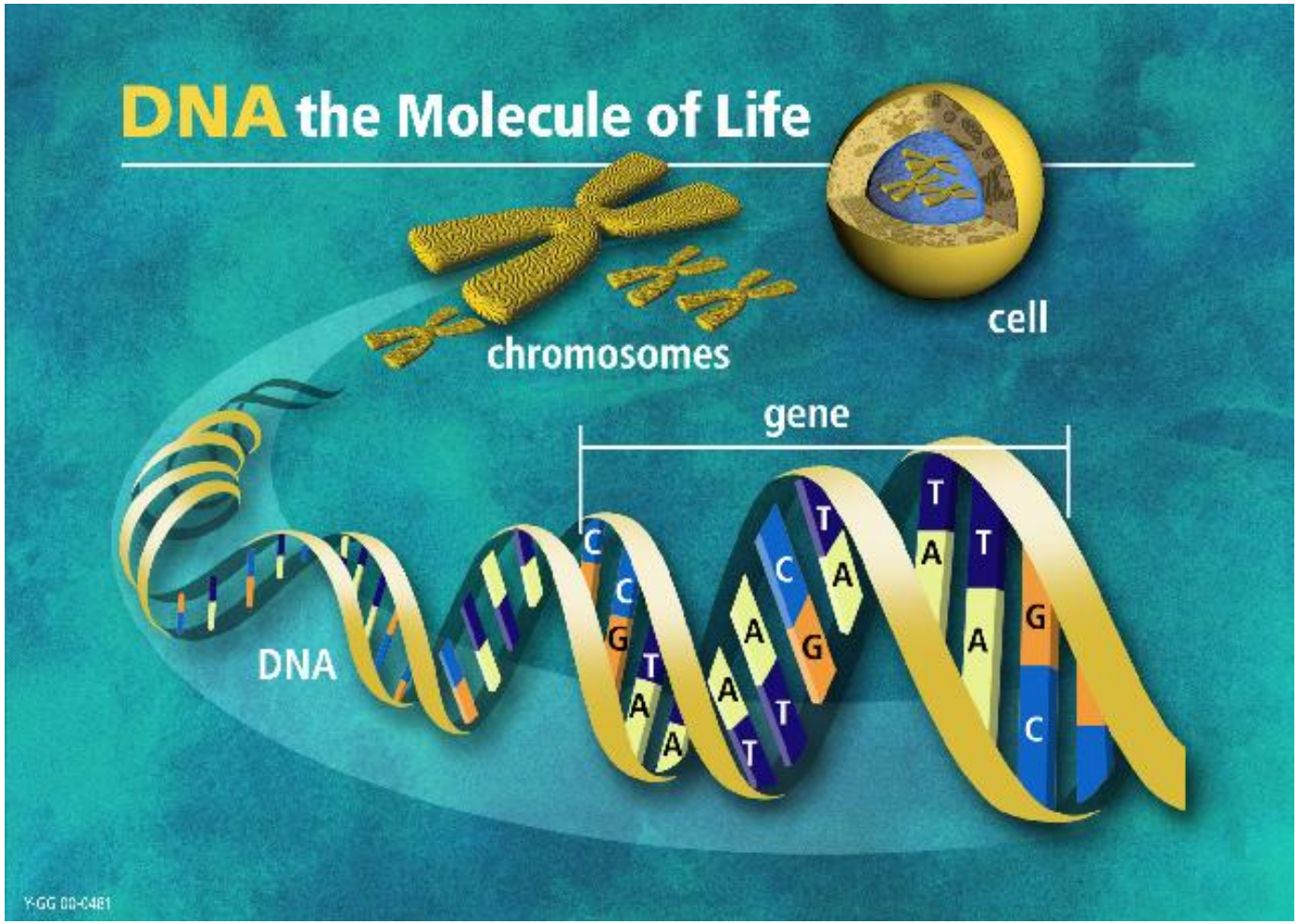
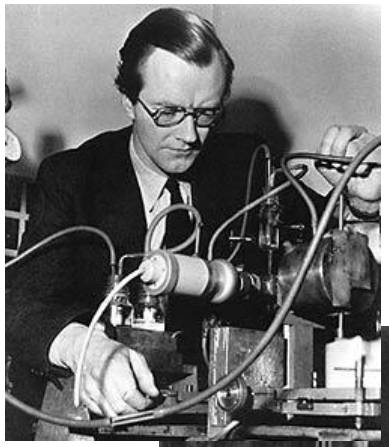


image from the DOE Human Genome Program
http://www.ornl.gov/hgmis

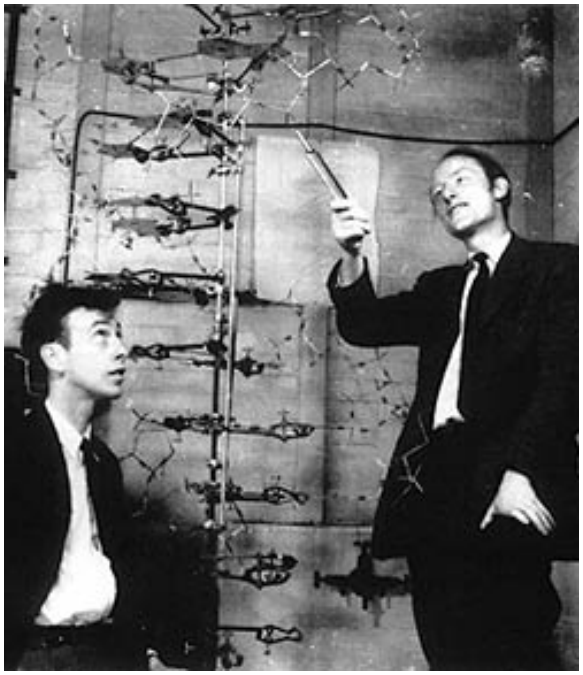
DNA is a double helical molecule

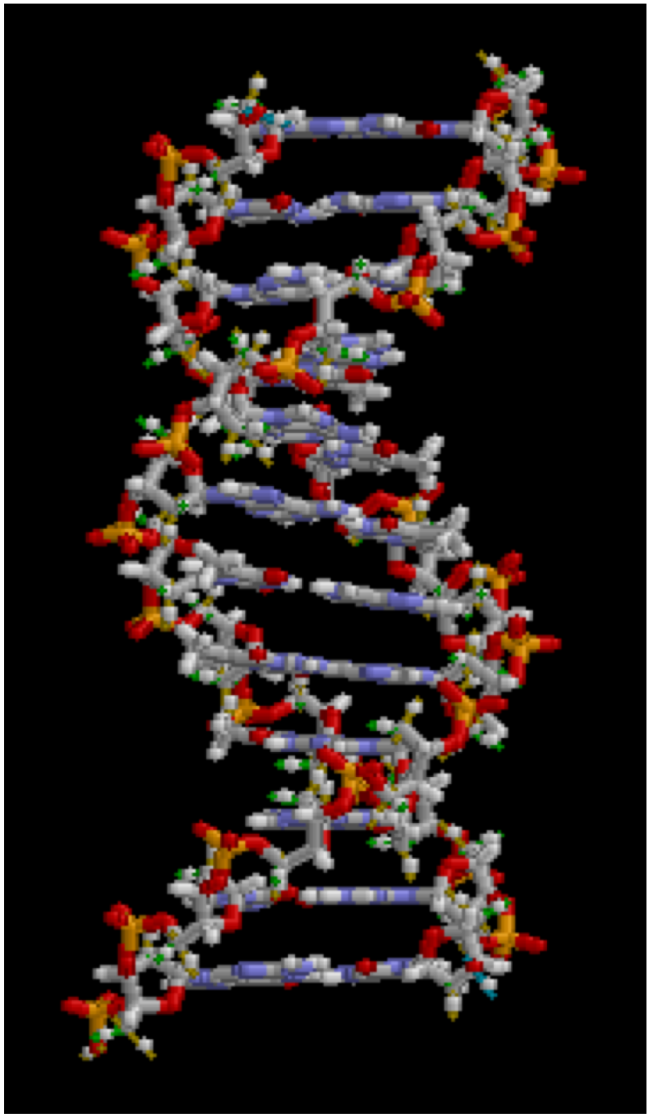


Maurice Wilkins

Rosalind Frankin

Watson and Crick



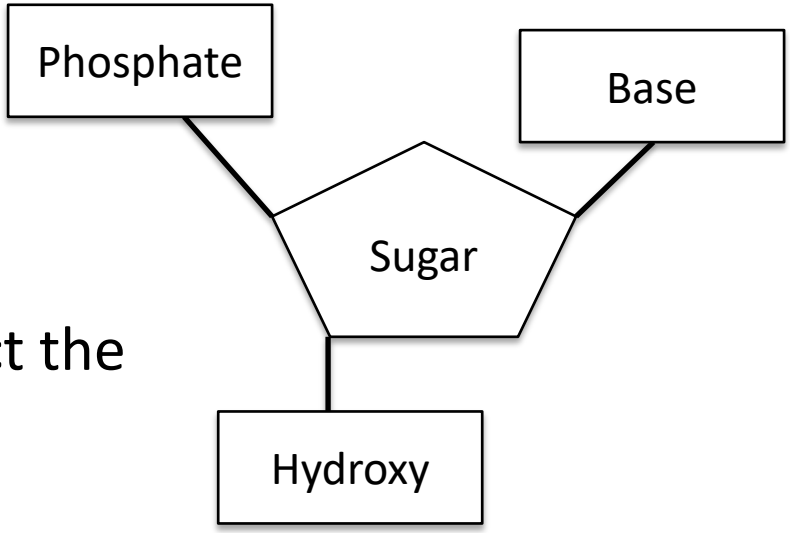


- In 1953, James Watson and Francis Crick discovered DNA molecule has two strands arranged in a double helix
- This was possible through the Xray diffraction data from Maurice Wilkins and Rosalind Franklin

http://www.chemheritage.org/discover/online-resources/chemistry-in-history/themes/biomolecules/dna/watson-crick-wilkins-franklin.aspx

Nucleotides

- DNA is a polymer
- Composed of repeating chemical units called *nucleotides*
- Nucleotide
  - Nitrogen containing base
  - 5 carbon sugar: deoxyribose
  - Phosphate group
  - Phosphate-hydroxy bonds connect the nucleotides
- Four nucleotides make DNA
  - adenine (A), cytosine (C), guanine (G) and thymine (T)



DNA stores the blue print of an organism

- The heredity molecule
- Has the information needed to make an organism
- Double strandedness of the DNA molecule provides stability, prevents errors in copying
  - one strand has all the information

Chromosomes

- All the DNA of an organism is divided up into individual *chromosomes*
- Each chromosome is really a DNA molecule
- Different organisms have different numbers of chromosomes

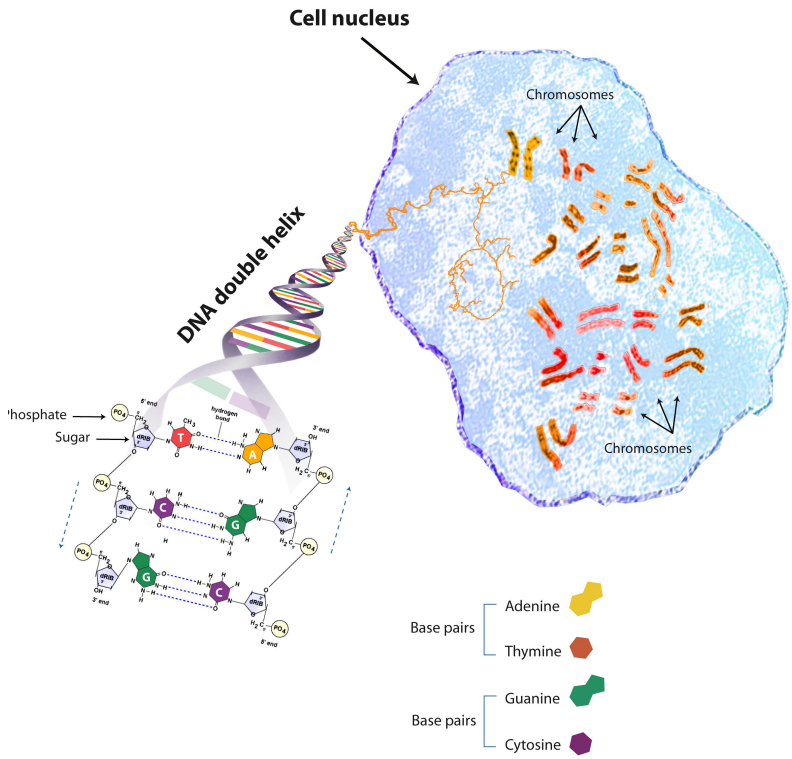
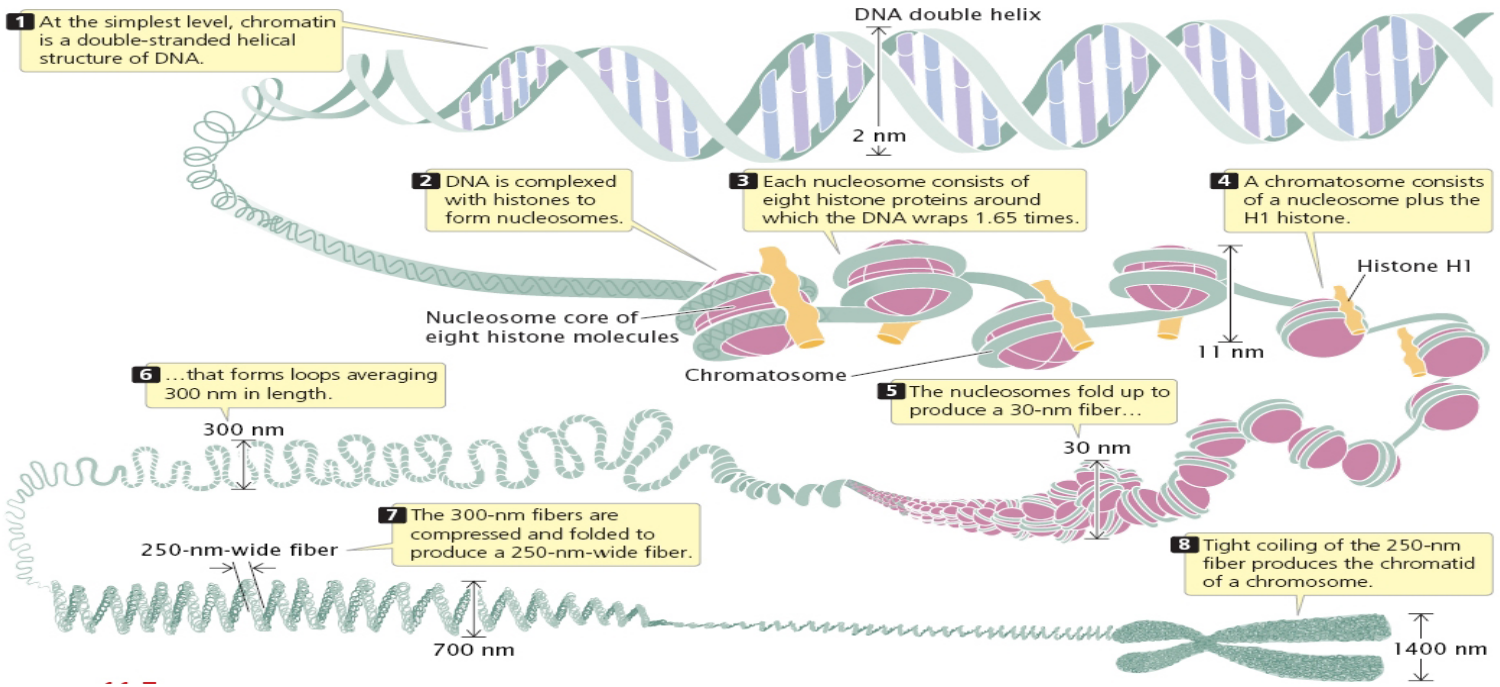


Image from www.genome.gov

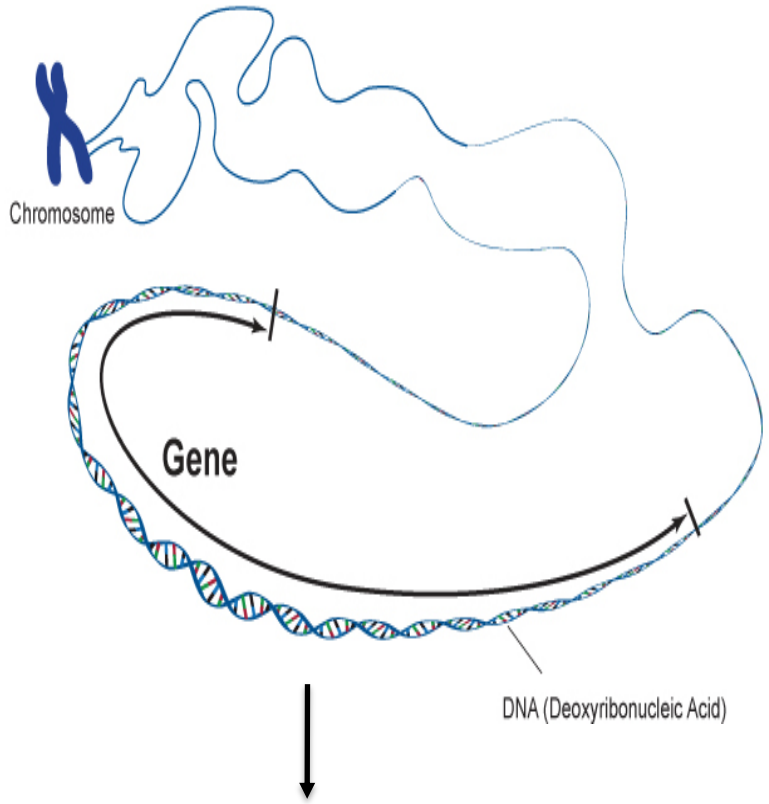
DNA packaging in Chromatin



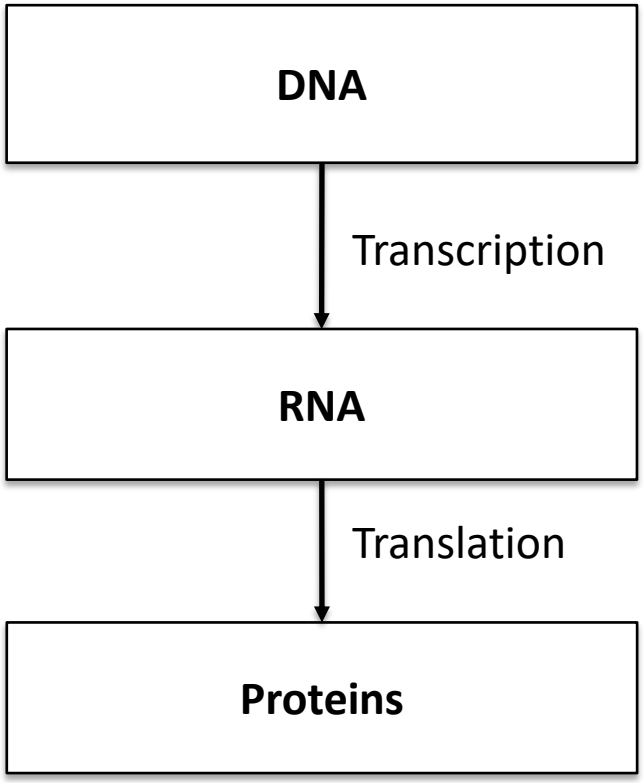
DNA is very long (3m in humans). The DNA is compressed and packaged inside a cell's nucleus with the help of a few key proteins (histones). Collection of DNA and proteins is called chromatin.

Genes

- Genes are the units of heredity
- A gene is a sequence of nucleotides which specifies a protein or RNA molecule
- The human genome has ~ 25,000 protein-coding genes (still being revised)
- One gene can have many functions
- One function can require many genes



The central dogma of Molecular biology



RNA: Ribonucleic acid

- RNA
  - Made up of repeating nucleotides
  - The sugar is ribose
  - U is used in place of T
- A strand of RNA can be thought of as a string composed of the four letters: A, C, G, U
- RNA is single stranded
  - More flexible than DNA
  - Can double back and form loops
  - Such structures can be more stable

Transcription

- In eukaryotes: happens inside the nucleus
- *RNA polymerase (RNA Pol)* is an enzyme that builds an RNA strand from a gene
- RNA Pol is recruited at specific parts of the genome in a condition-specific way.
- Transcription factor proteins are assigned the job of RNA Pol recruitment.
- RNA that is transcribed from a protein coding region is called *messenger RNA* (mRNA)

Translation

- Process of turning mRNA into proteins.
- Happens outside of the **nucleus** inside the **cytoplasm** in **ribosomes**
- *ribosomes* are the machines that synthesize proteins from mRNA

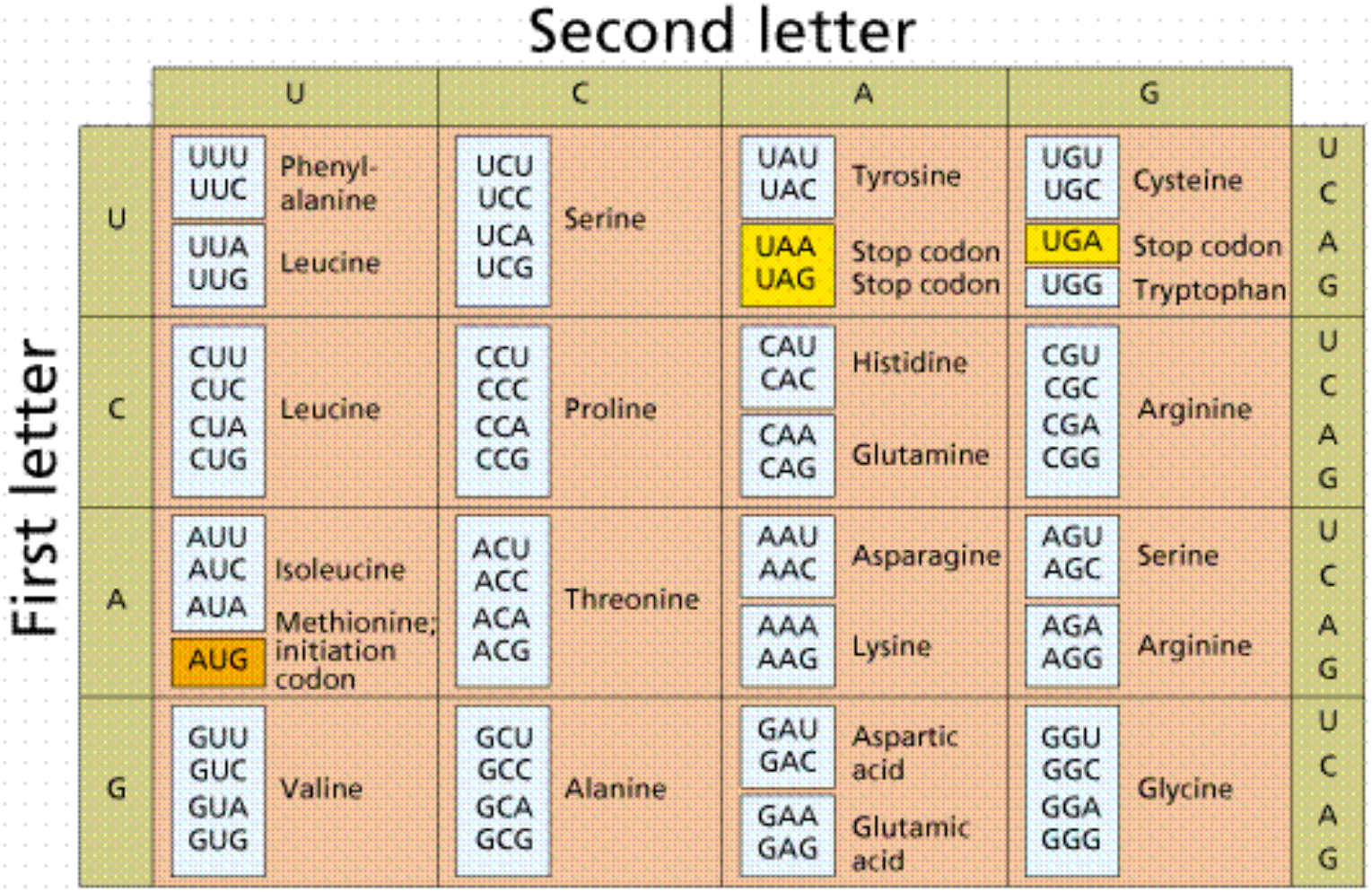
Proteins

- Proteins are polymers too
- The repeating units are *amino acids*
- There are 20 different amino acids known
- DNA sequence of a gene *codes* for a protein
- Some types of proteins are transcription factors and metabolic enzymes, signaling proteins

Amino Acids

Alanine	Ala	A
Arginine	Arg	R
Aspartic Acid	Asp	D
Asparagine	Asn	N
Cysteine	Cys	C
Glutamic Acid	Glu	E
Glutamine	Gln	Q
Glycine	Gly	G
Histidine	His	H
Isoleucine	Ile	I
Leucine	Leu	L
Lysine	Lys	K
Methionine	Met	M
Phenylalanine	Phe	F
Proline	Pro	P
Serine	Ser	S
Threonine	Thr	T
Tryptophan	Trp	W
Tyrosine	Tyr	Y
Valine	Val	V

The genetic code: specifies how mRNA is translated into protein



Genetic code is degenerate

A video about transcription and translation

Metabolites

- Small molecules that are essential to living systems
  - Water, sugars, fat
- Product or substrate of a metabolic process

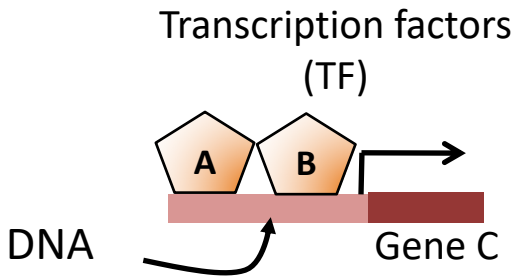
Goals for today

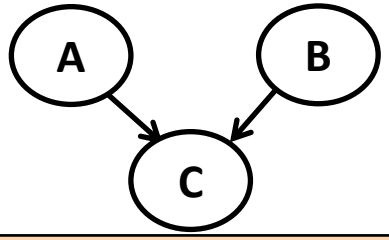
- Introductory Graph theory
- Molecules of life
- **Different types of molecular networks**

Graphs for representing molecular networks

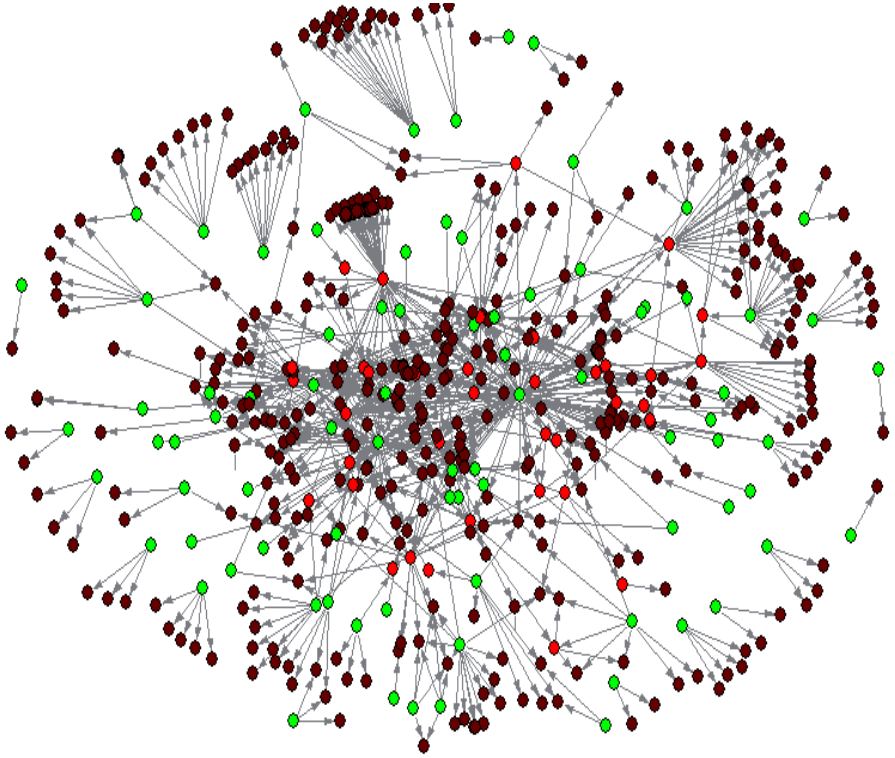
- Nodes are biological molecules
  - Genes, proteins, metabolites, etc
- Edges represent interaction between molecules
- Many different types of molecular networks exist
- They vary based upon the node and edge semantics

Transcriptional regulatory networks





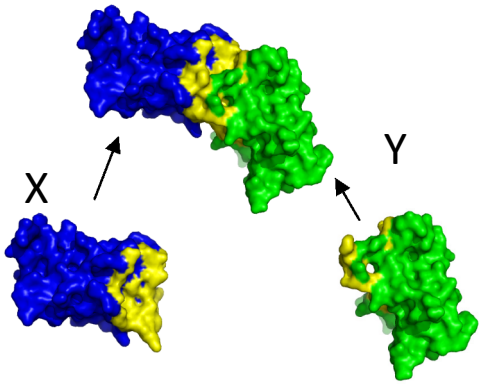
- Directed, signed, weighted graph
- Nodes: TFs and Target genes
- Edges: A regulates C's expression level

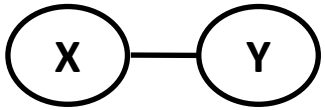


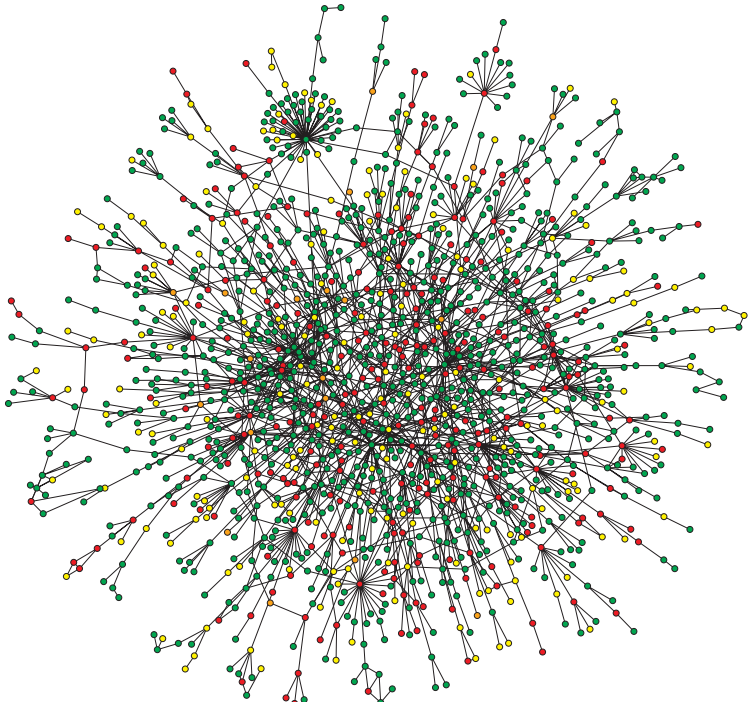
Regulatory network of *E. coli*.
153 TFs (green & light red), 1319 targets

Vargas and Santillan, 2008

Protein-protein interaction networks





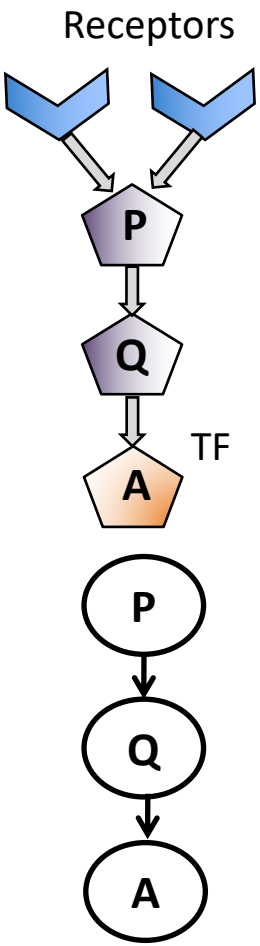


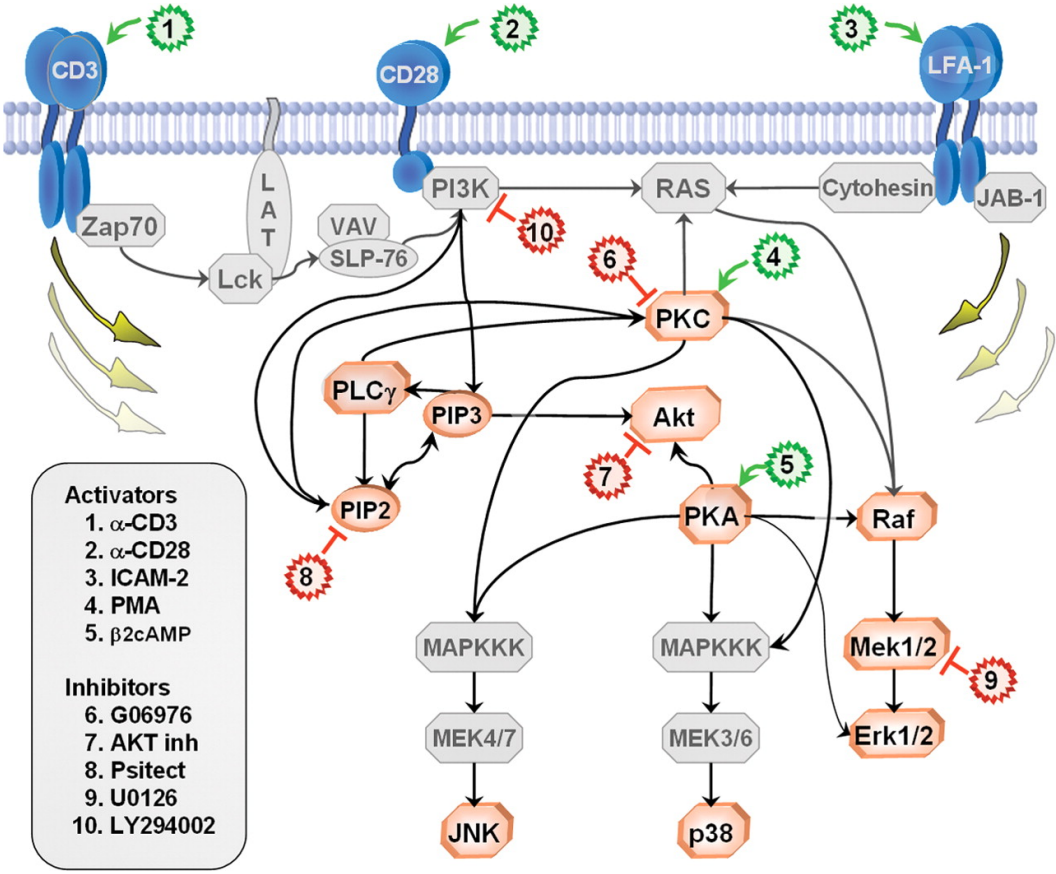
Yeast protein interaction network

- Undirected, may or may not be weighted graph
- Nodes: Proteins
- Edges: Protein X physically interacts with protein Y

Barabasi et al. 2003

Signaling networks

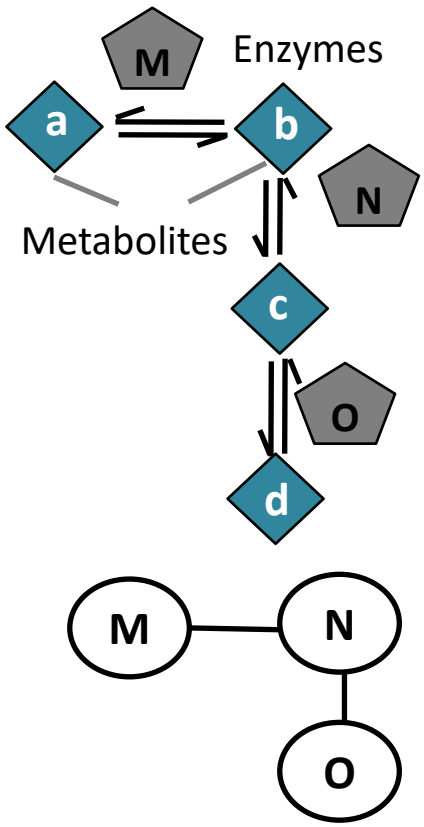


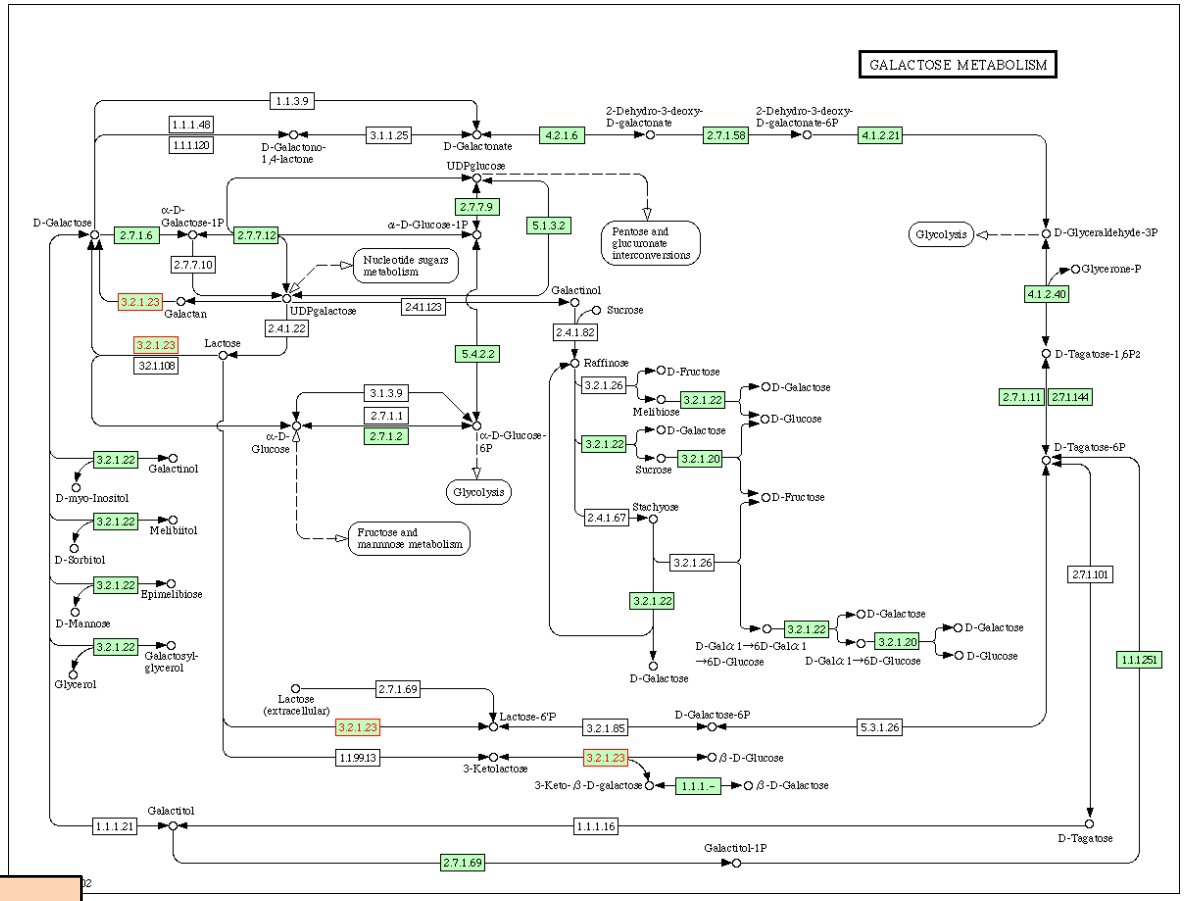


- Directed graph
- Nodes: Enzymes and other proteins
- Edges: Enzyme P modifies protein Q

Sachs *et al.*, 2005, Science

Metabolic networks





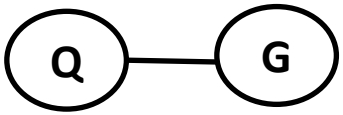
Reactions associated with Galactose metabolism

- Unweighted graph
- Nodes: Metabolic enzyme
- Edges: Enzymes M and N share a compound

KEGG

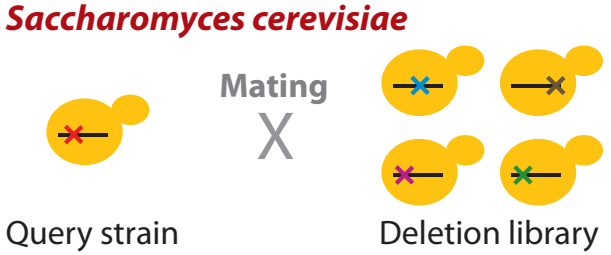
Genetic interaction networks

Genetic interaction: If the phenotype of double mutant is significantly different than each mutant alone

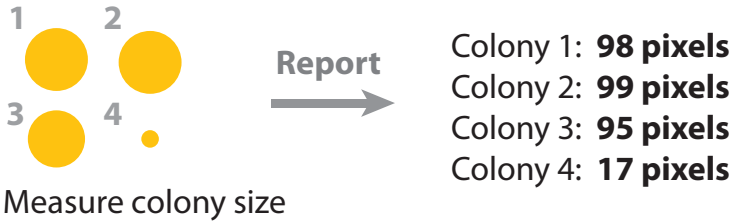


- Undirected graph
- Nodes: Genes
- Edges: Genetic interaction between query gene Q and gene G

Step 1:
Generate double mutant



Step 2:
Score phenotype and identify interactions

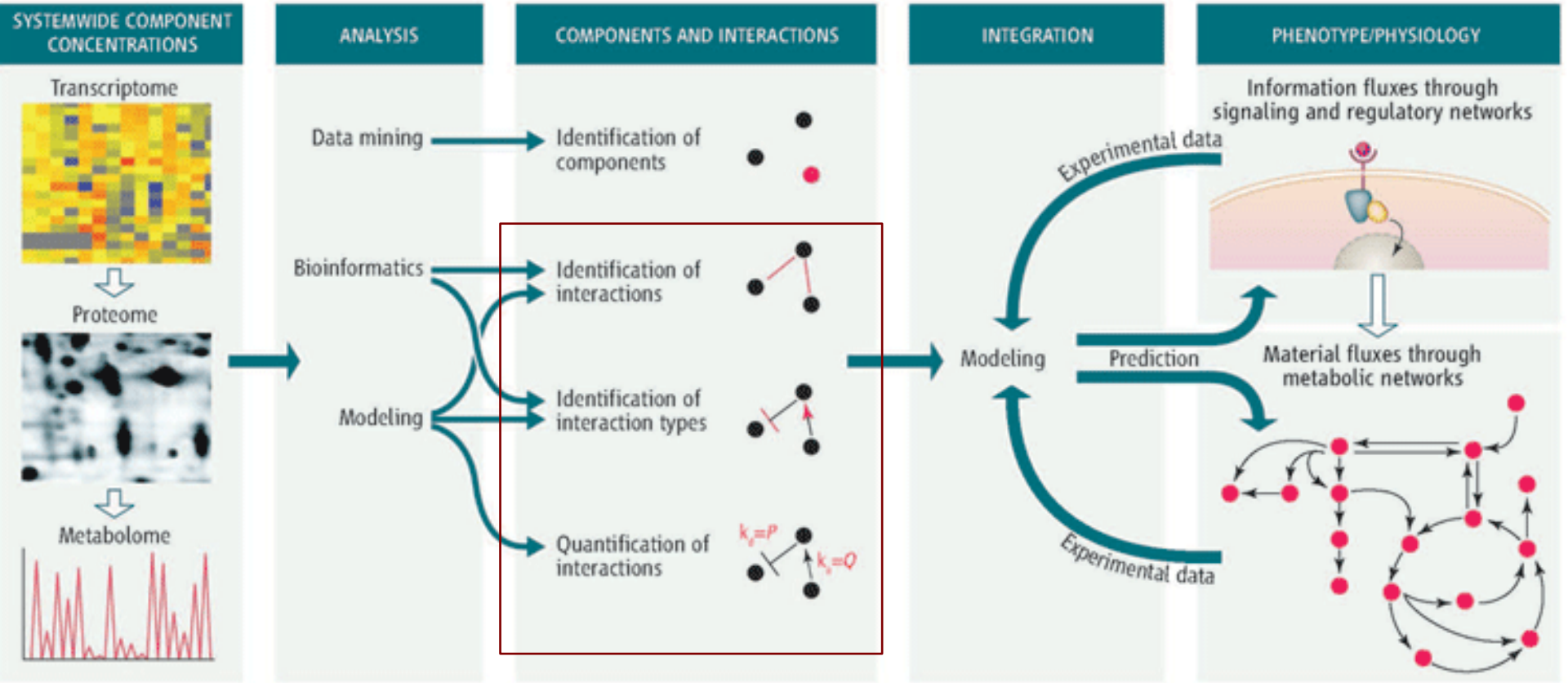


Dixon *et al.*, 2009, Annu. Rev. Genet

Different types of networks

- Physical networks
  - *Transcriptional regulatory networks:* interactions between regulatory proteins (transcription factors) and genes
  - *Protein-protein*: interactions among proteins
  - *Signaling networks*: protein-protein and protein-small molecule interactions to relay signals from outside the cell to the nucleus
- Functional networks
  - *Metabolic*: reactions through which enzymes convert substrates to products
  - *Genetic*: interactions among genes which when **perturbed together** produce a significant phenotype than when **individually perturbed**

“Omic” tools measure cellular molecular components in a high-throughput manner



Uwe Sauer, Matthias Heinemann, Nicola Zamboni, Science 2007

References

- Cormen, Thomas H., Clifford Stein, Ronald L. Rivest, and Charles E. Leiserson. *Introduction to Algorithms*. 2nd edition. McGraw-Hill Higher Education, 2001.
- Introductory lecture from Introduction to Bioinformatics, BMI/CS 576