

Sequence Analysis

- Some algorithms analyze biological sequences for patterns
 - RNA splice sites
 - Open reading frames (ORFs): stretch of codons
 - Amino acid propensities in a protein
 - Conserved regions in
 - AA sequences [possible active site]
 - DNA/RNA [possible protein binding site]
- Others make predictions based on sequence
 - Protein/RNA secondary structure folding

Bioinformatics

Sequence Driven Problems

- **Genomics**

- Fragment assembly of the DNA sequence.
 - Not possible to read entire sequence.
 - Cut up into small fragments using restriction enzymes.
 - Then need to do fragment assembly. Overlapping similarities to matching fragments.
 - N-P complete problem.
- Finding Genes
 - Identify open reading frames
 - Exons are spliced out.
 - Junk in between genes

- **Proteomics**

- Identification of functional domains in protein's sequence
 - Determining functional pieces in proteins.
- Protein Folding
 - 1D Sequence → 3D Structure
 - What drives this process?

Genome is Sequenced, What's Next?

- **Tracing Phylogeny**
 - Finding family relationships between species by tracking similarities between species.
- **Gene Annotation (cooperative genomics)**
 - Comparison of similar species.
- **Determining Regulatory Networks**
 - The variables that determine how the body reacts to certain stimuli.
- **Proteomics**
 - From DNA sequence to a folded protein.

Modeling

- Modeling biological processes tells us if we understand a given process
- Because of the large number of variables that exist in biological problems, powerful computers are needed to analyze certain questions
- **Protein modeling:**
 - Quantum chemistry imaging algorithms of active sites allow us to view possible bonding and reaction mechanisms
 - Homologous protein modeling is a comparative proteomic approach to determining an unknown protein's tertiary structure
- **Regulatory Network Modeling:**
 - Micro array experiments allow us to compare differences in expression for two different states
 - Algorithms for clustering groups of gene expression help point out possible regulatory networks (e.g. WGCNA)
 - Other algorithms perform statistical analysis to improve signal to noise contrast
- **Systems Biology Modeling:**
 - Predictions of whole cell interactions (Organelle processes, expression modeling)
 - Currently feasible for specific processes (eg. Flux Balance Analysis)

Back to the sequence alignment problem

Pairwise Sequence Alignment Methods

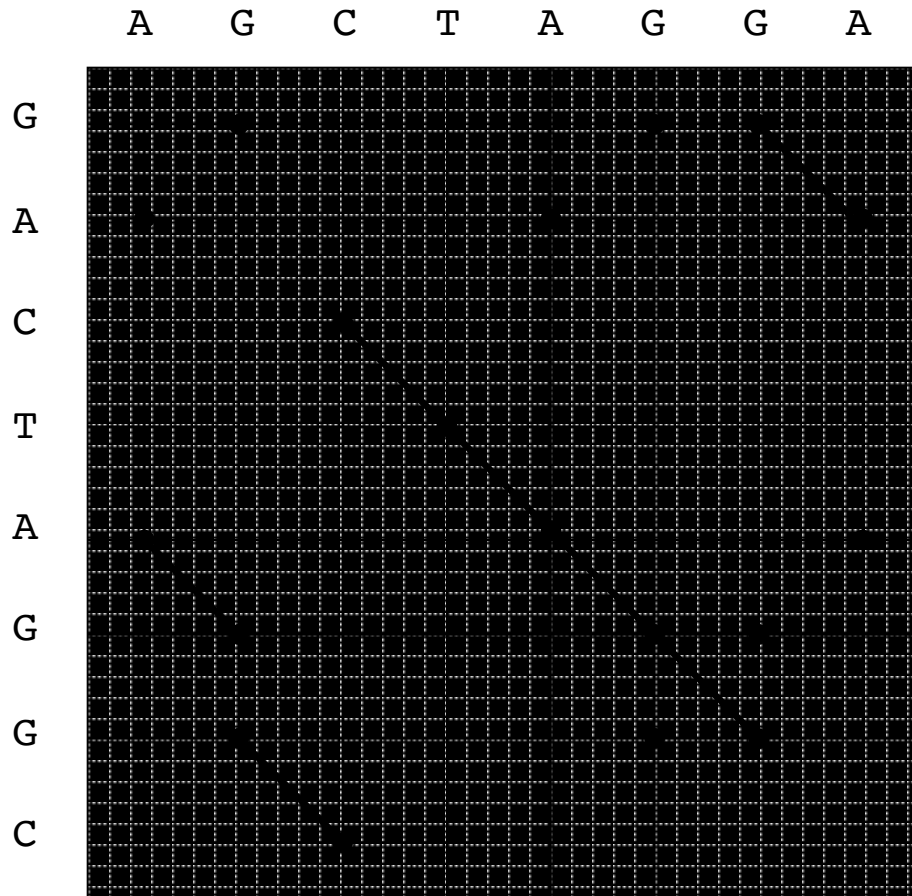
- Visual
- Brute Force
- Dynamic Programming
- Word-Based (k tuple)

Visual Alignments (Dot Plots)

- Matrix
 - Rows: Characters in one sequence
 - Columns: Characters in second sequence
- Filling
 - Loop through each row; if character in row, col match, fill in the cell
 - Continue until all cells have been examined

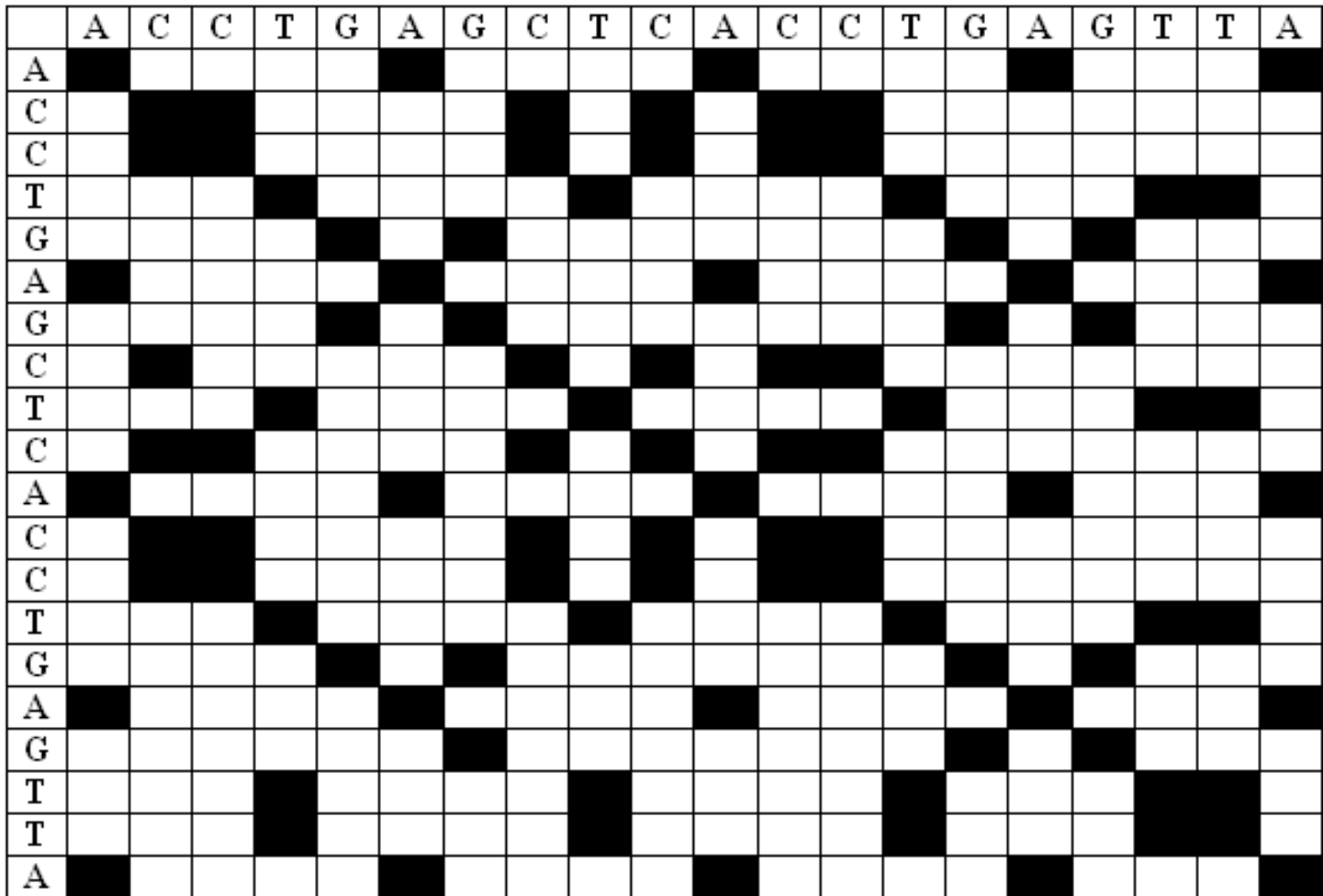
The Dot Matrix

- established in 1970 by A.J. Gibbs and G.A.McIntyre
- method for comparing two amino acid or nucleotide sequences

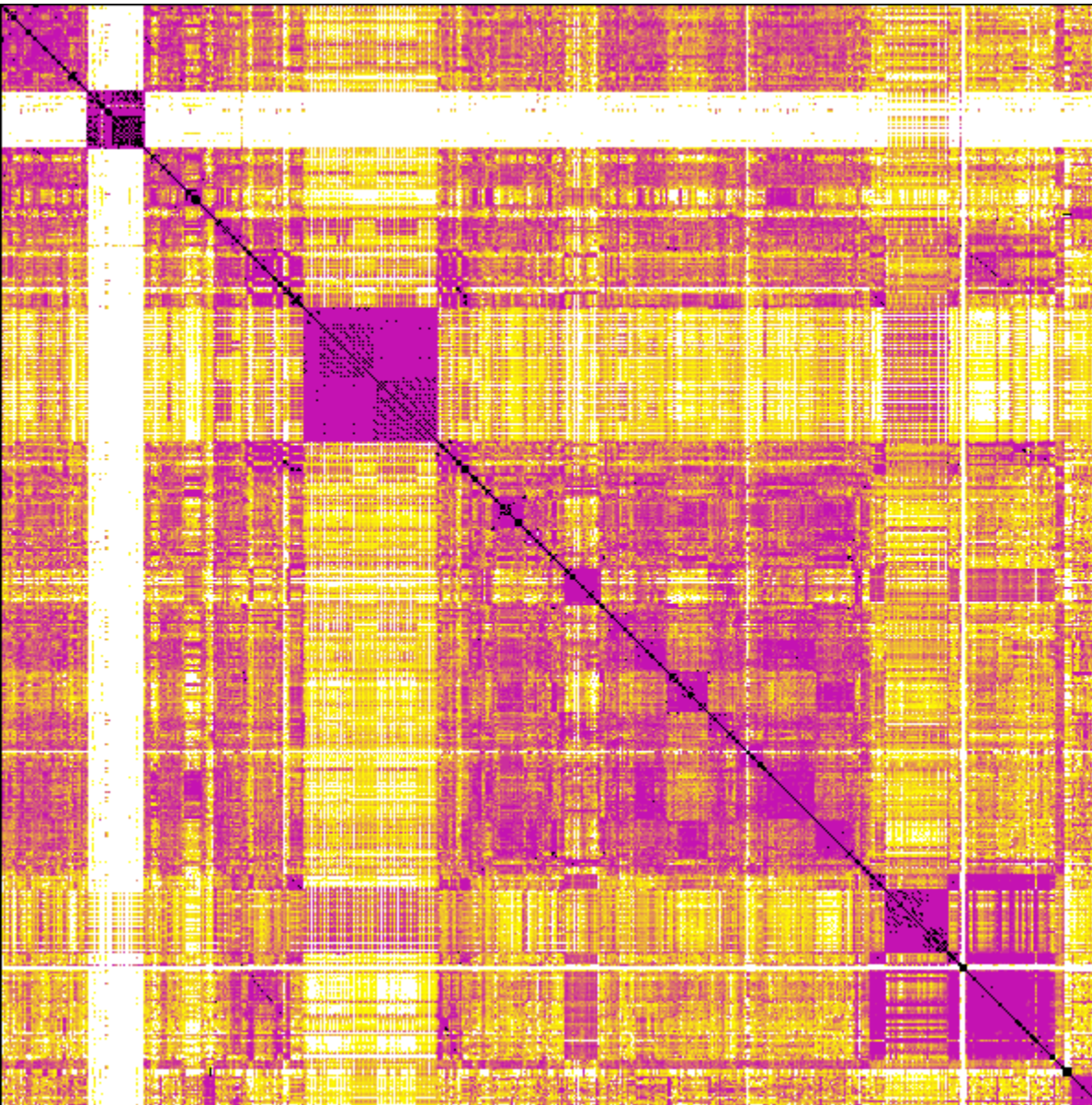


- each sequence builds one axis of the grid
- one puts a dot, at the intersection of same letters appearing in both sequences
- scan the graph for a series of dots
 - reveals similarity
 - or a string of same characters
- longer sequences can also be compared on a single page, by using smaller dots

Example Dot Plot



An entire software module of a telecommunications switch;
about two million lines of C



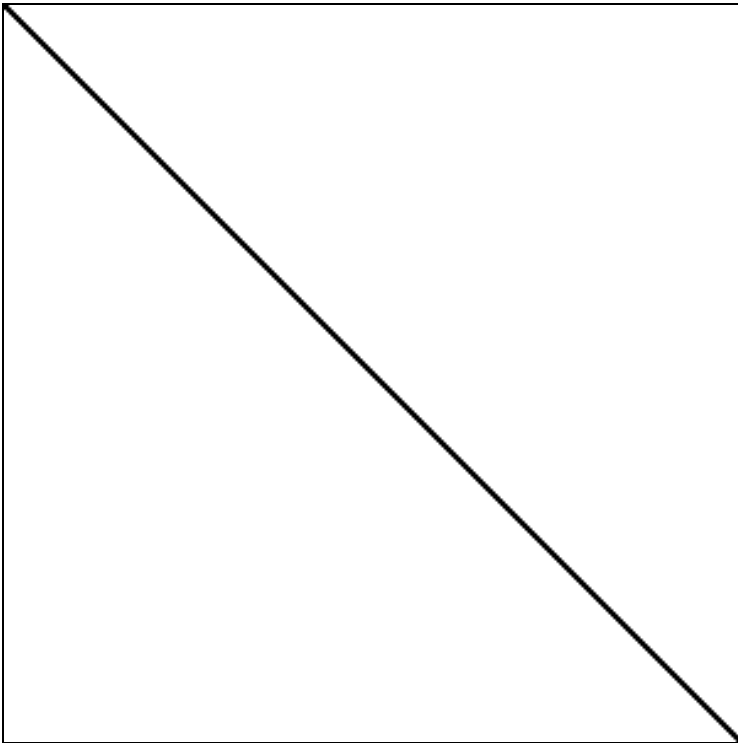
Darker areas indicate regions with a lot of matches (a high degree of similarity). Lighter areas indicate regions with few matches (a low degree of similarity). Dark areas along the main diagonal indicate sub-modules.

Dark areas off the main diagonal indicate a degree of similarity between sub-modules

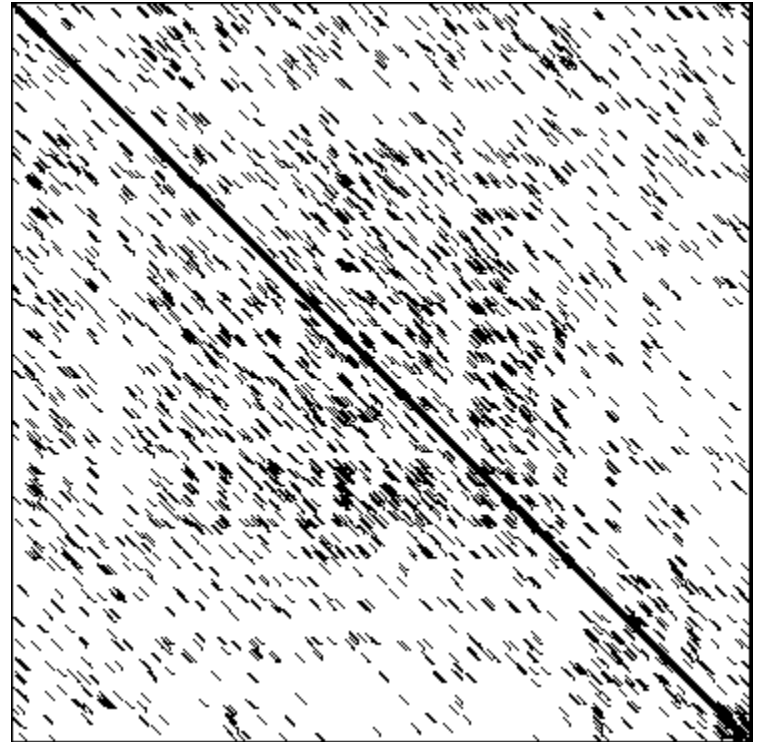
The largest dark squares are formed by redundancies in initializations of signal-tables and finite-state machines.

The Dot Matrix

The very stringent, self-dotplot:



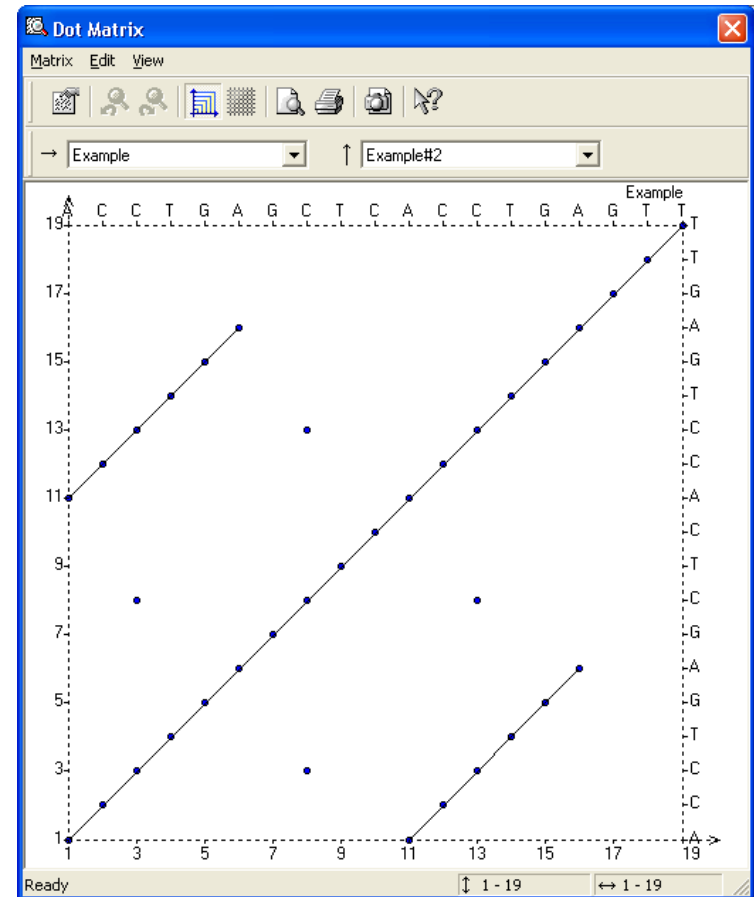
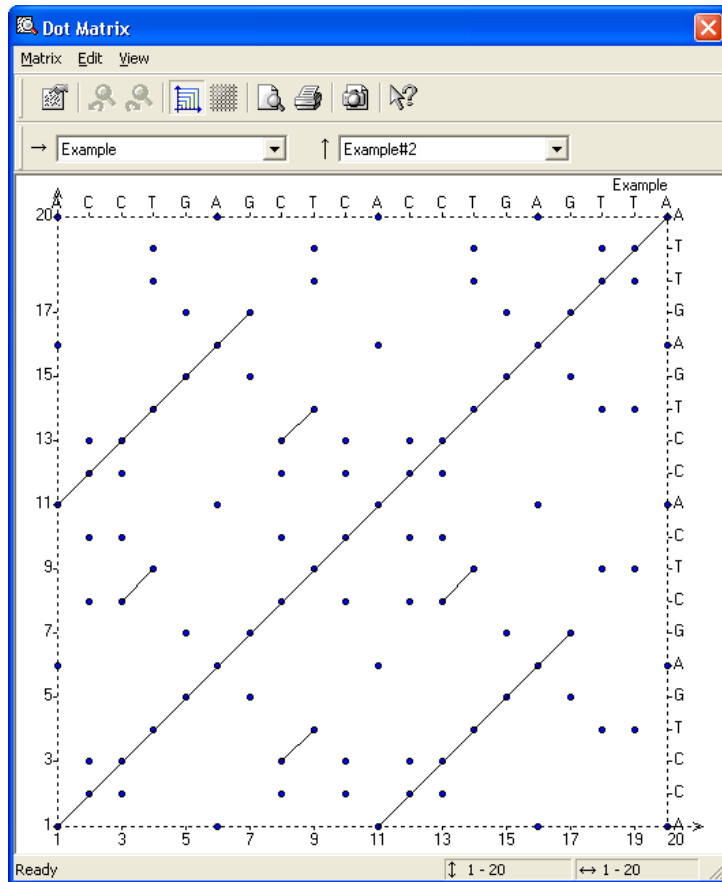
The non-stringent self-dotplot:



Noise in Dot Plots

- Nucleic Acids (DNA, RNA)
 - 1 out of 4 bases matches at random
- Stringency (The condition under which a DNA sequence can bind to related or non-specific sequences. For example, high temperature and lower salt increases stringency such that non-specific binding or binding with low melting temperature will dissolve)
 - Window size is considered
 - Percentage of bases matching in the window is set as threshold
- To filter out random matches, one uses sliding windows
- A dot is printed only if a minimal number of matches occur
- rule of thumb:
 - larger windows for DNAs (only 4 bases, more random matches)
 - typical window size is 15 and stringency of 10

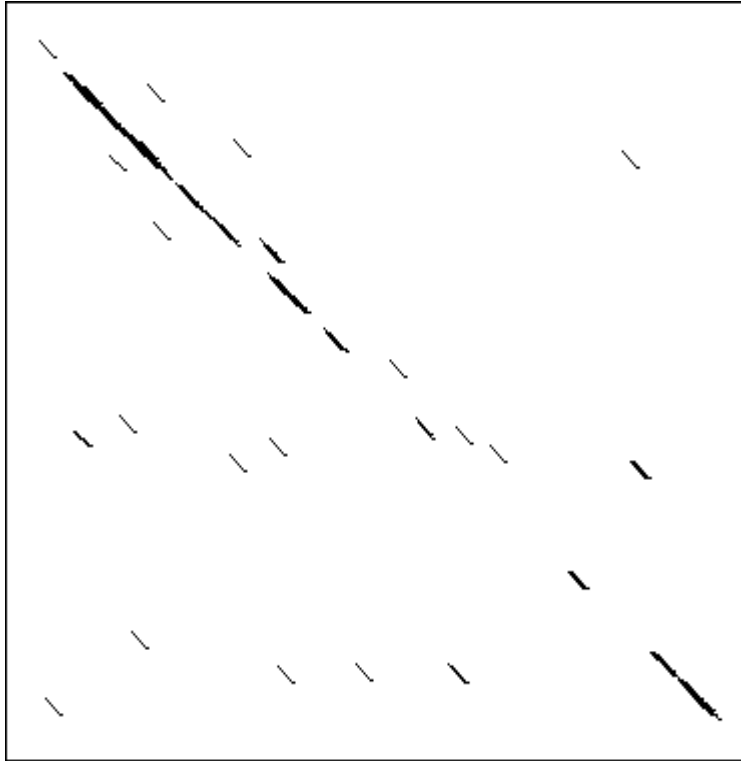
Reduction of Dot Plot Noise



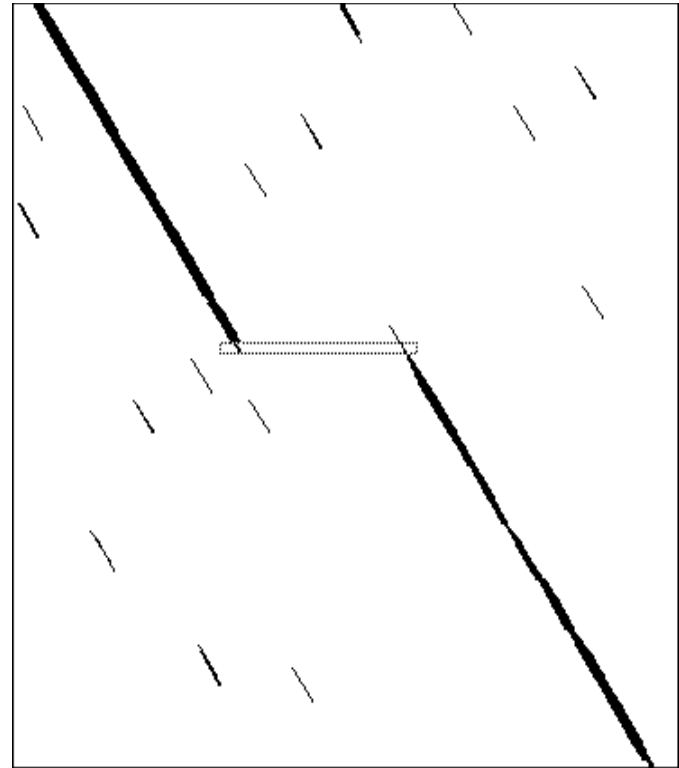
Self alignment of ACCTGAGCTCACCTGAGTTA

The Dot Matrix

Two similar, but not identical, sequences

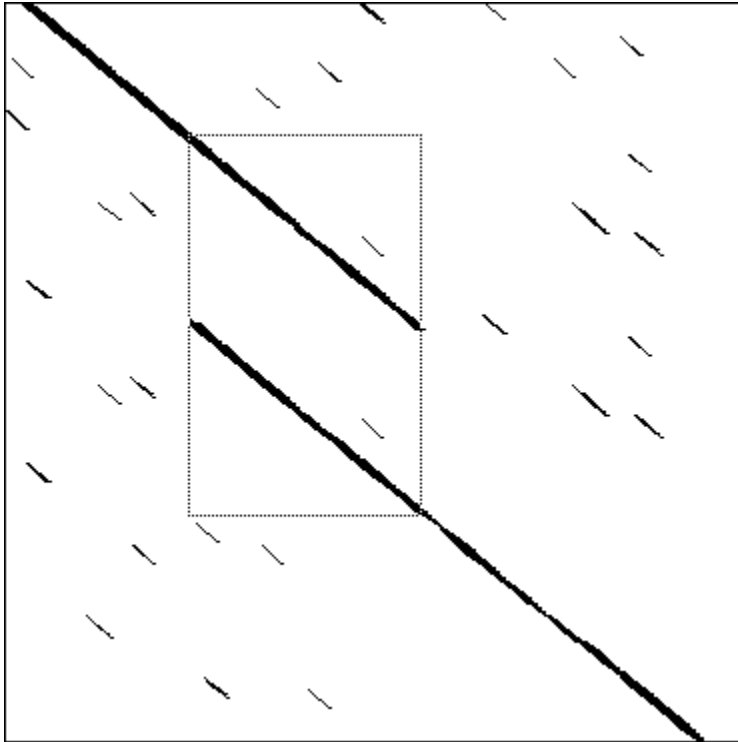


An indel (insertion or deletion):

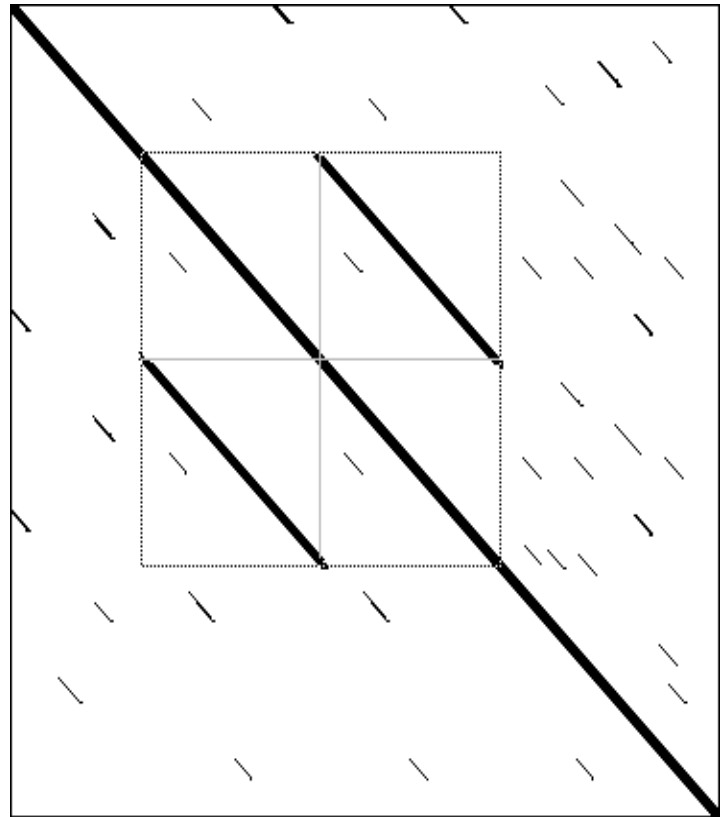


The Dot Matrix

A tandem duplication:

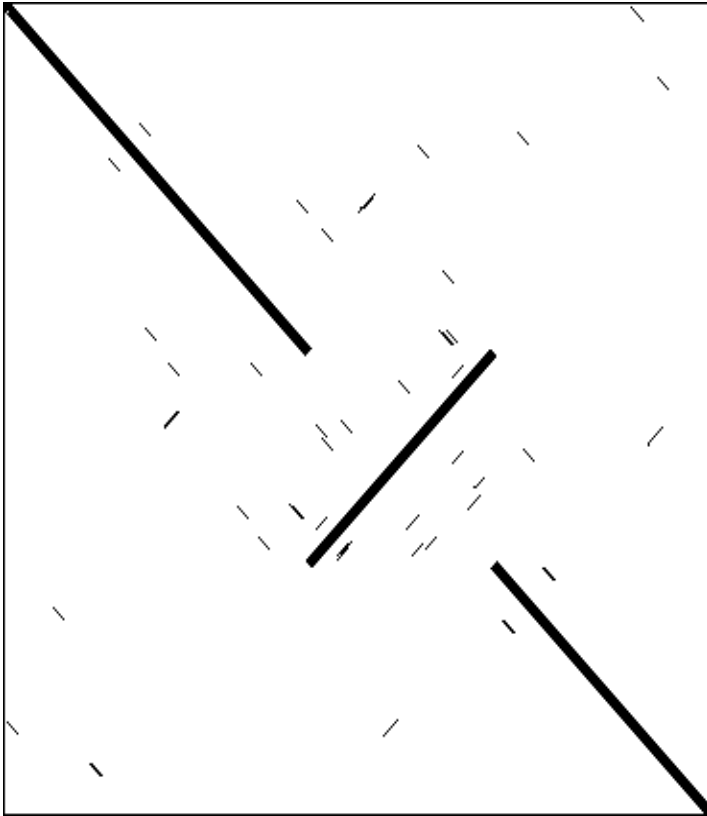


Self-dotplot of a tandem duplication:

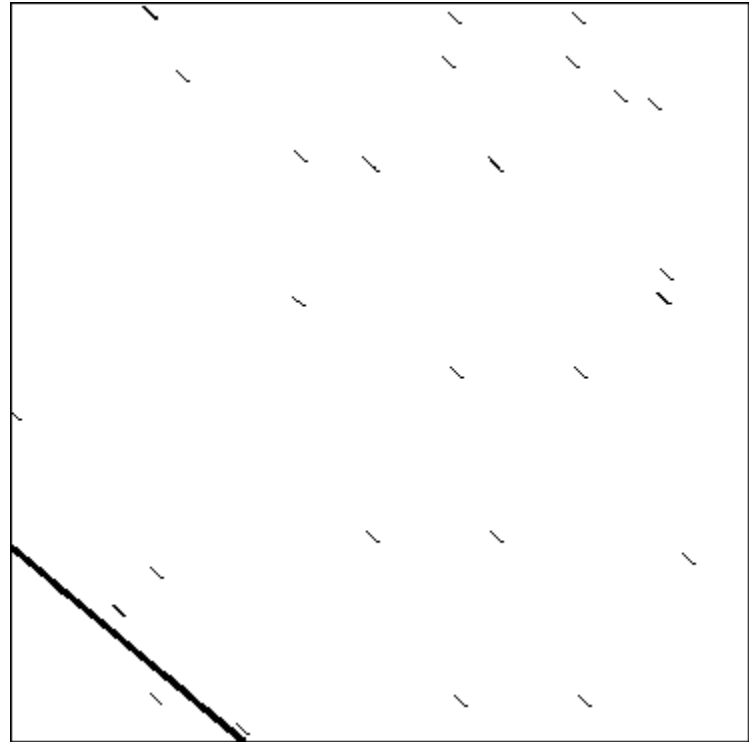


The Dot Matrix

An inversion:

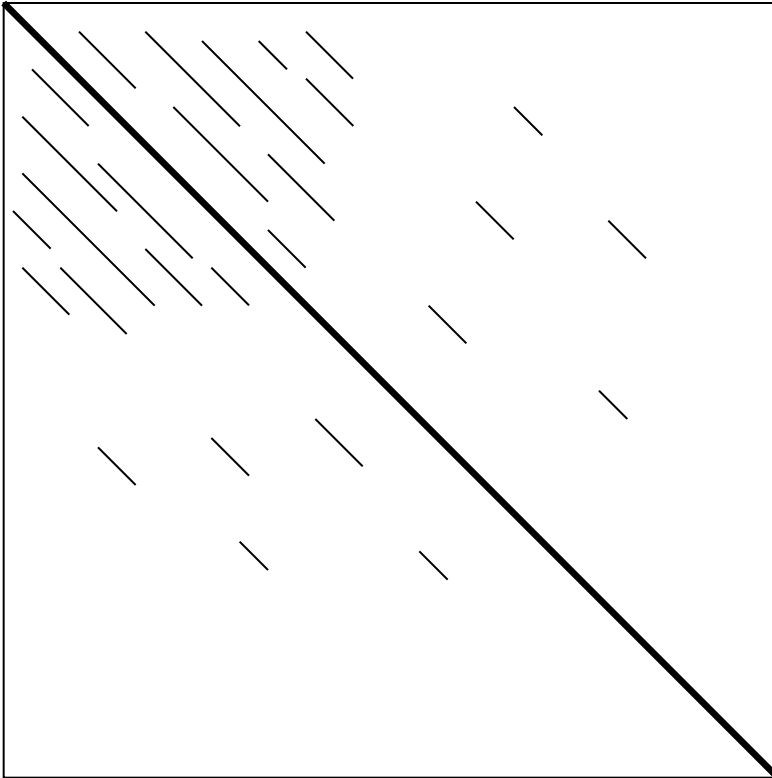


Joining sequences:



The Dot Matrix

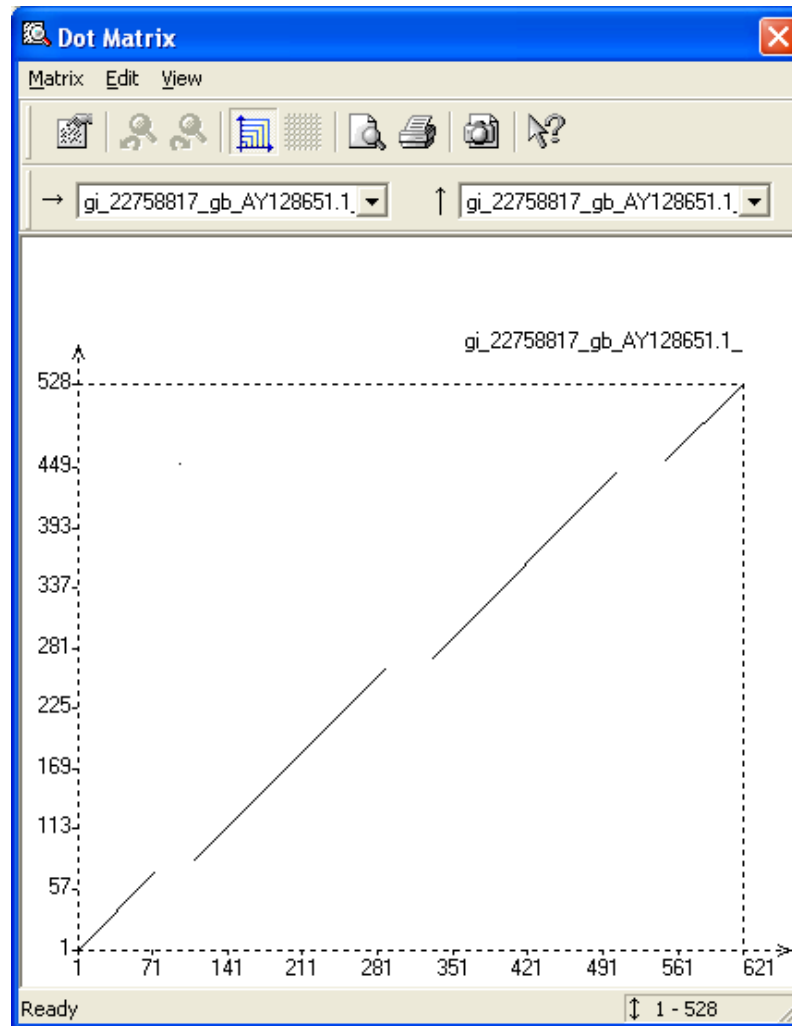
Self dotplot with repeats:



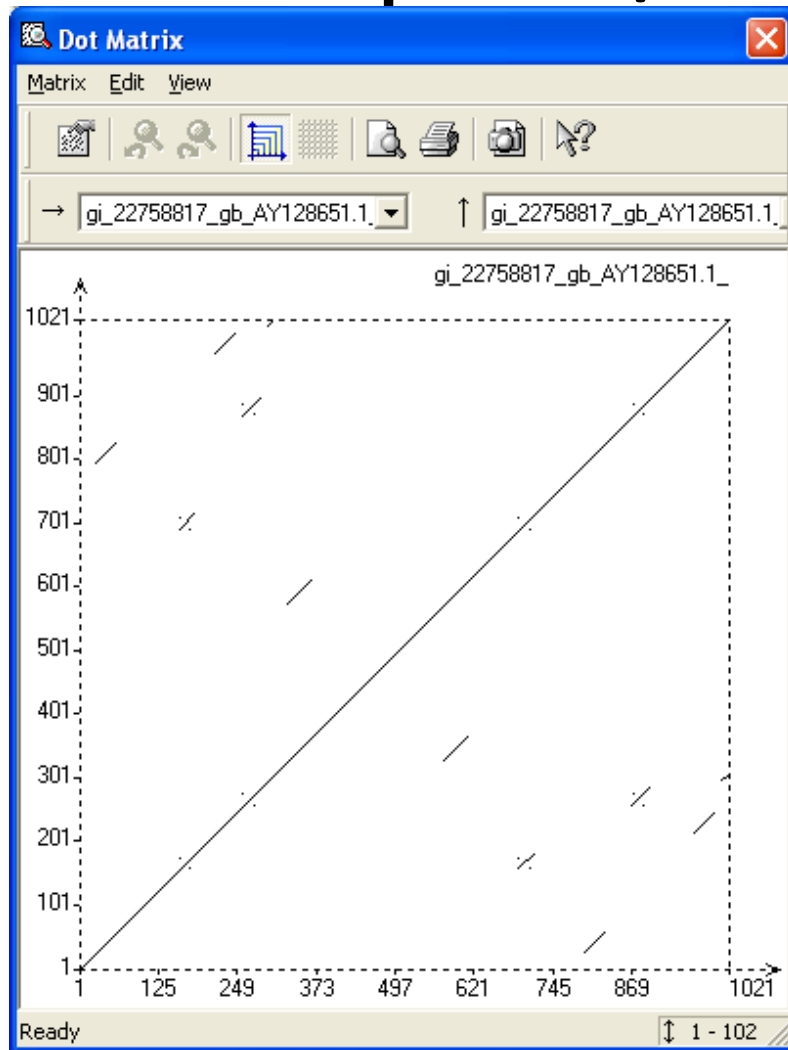
The Dot Matrix

- the dot matrix method reveals the presence of insertions or deletions
- comparing a single sequence to itself can reveal the presence of a repeat of a subsequence
 - Inverted repeats = reverse complement
 - Used to determine folding of RNA molecules
- self comparison can reveal several features:
 - similarity between chromosomes
 - tandem genes
 - repeated domains in a protein sequence
 - regions of low sequence complexity (same characters are often repeated)

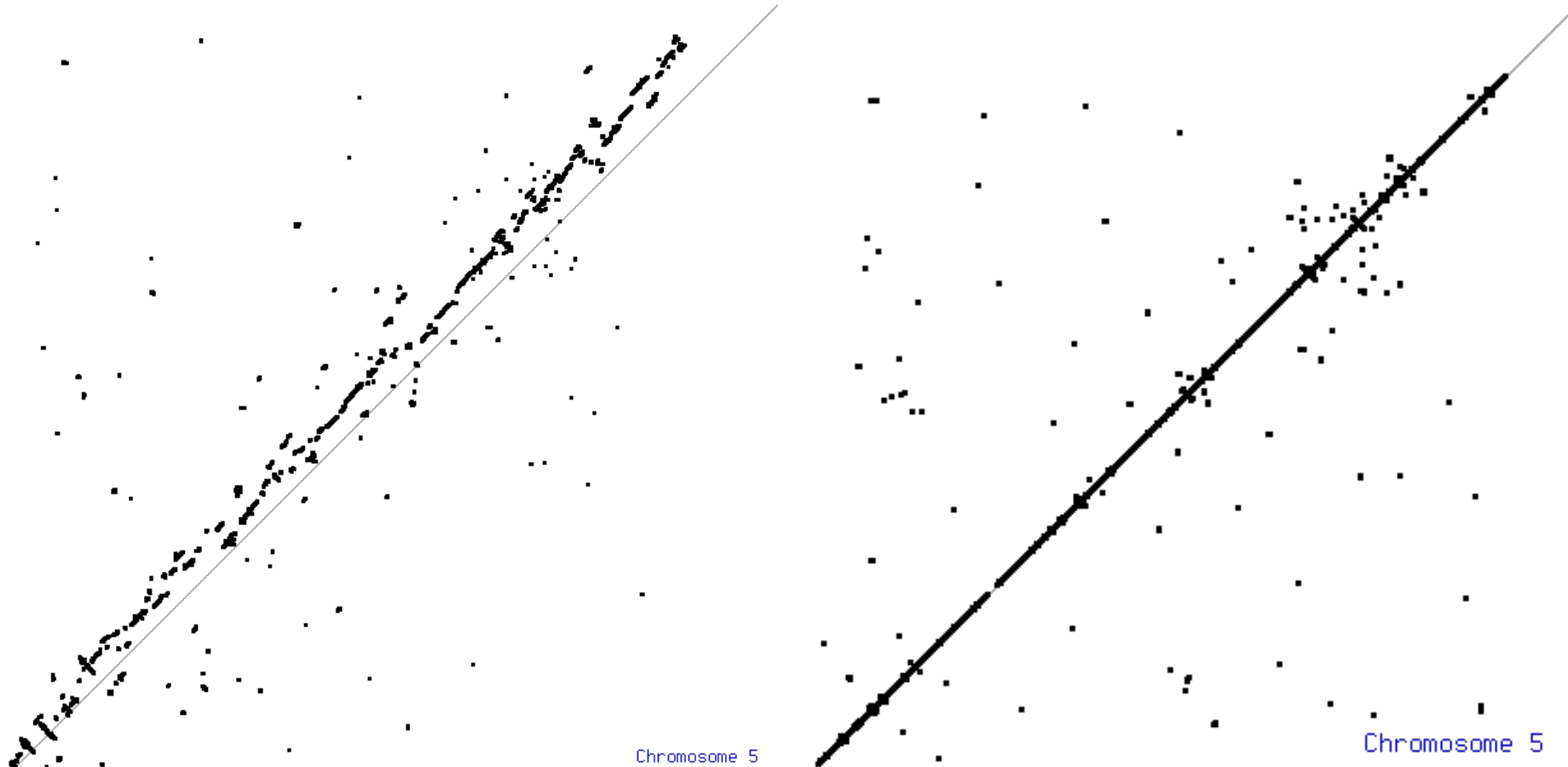
Insertions/Deletions



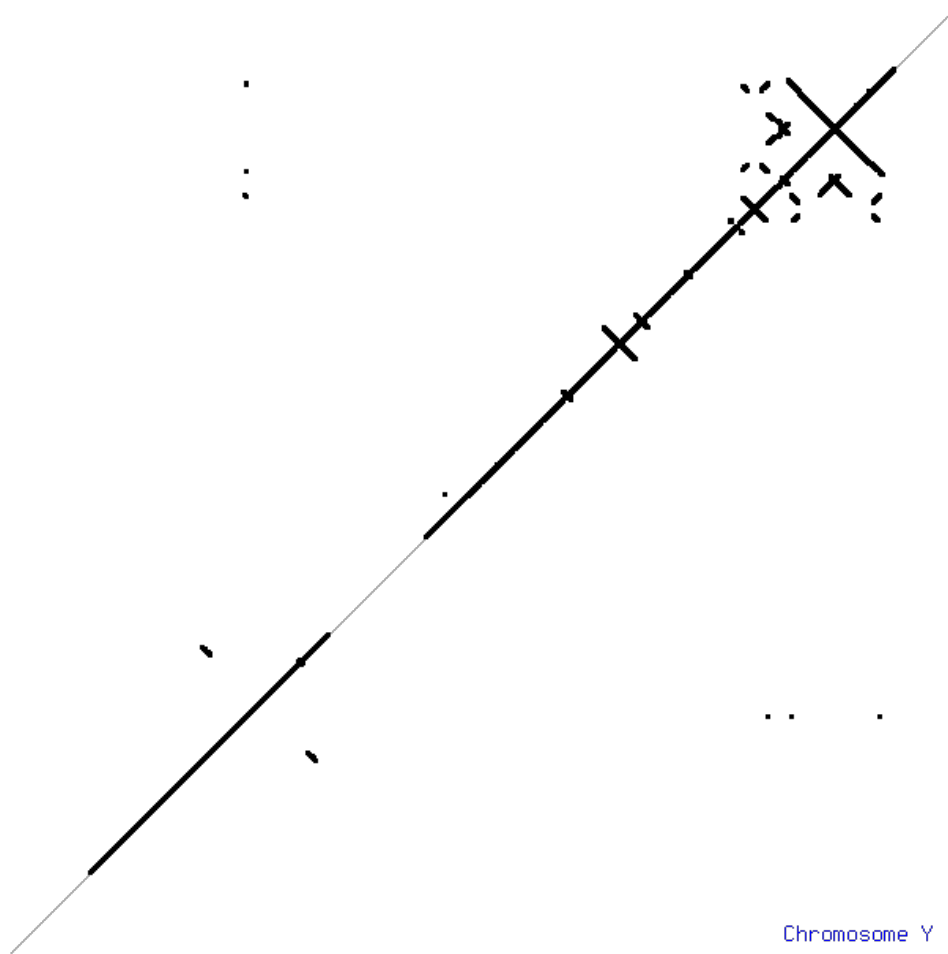
Repeats/Inverted Repeats



Comparing Genome Assemblies

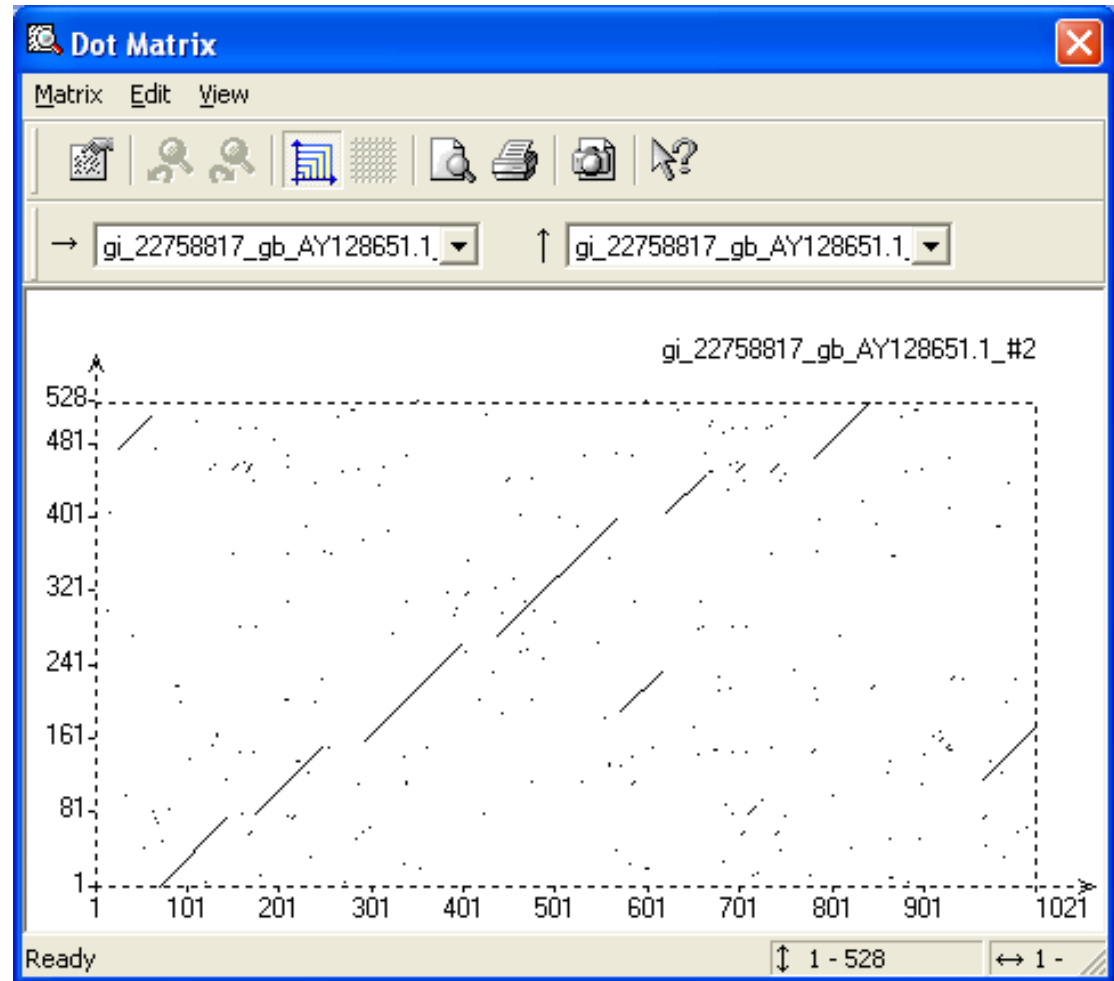


Chromosome Y self comparison



Available Dot Plot Programs

- Vector NTI software package (under AlignX)



Available Dot Plot Programs

- Vector NTI software package (under AlignX)

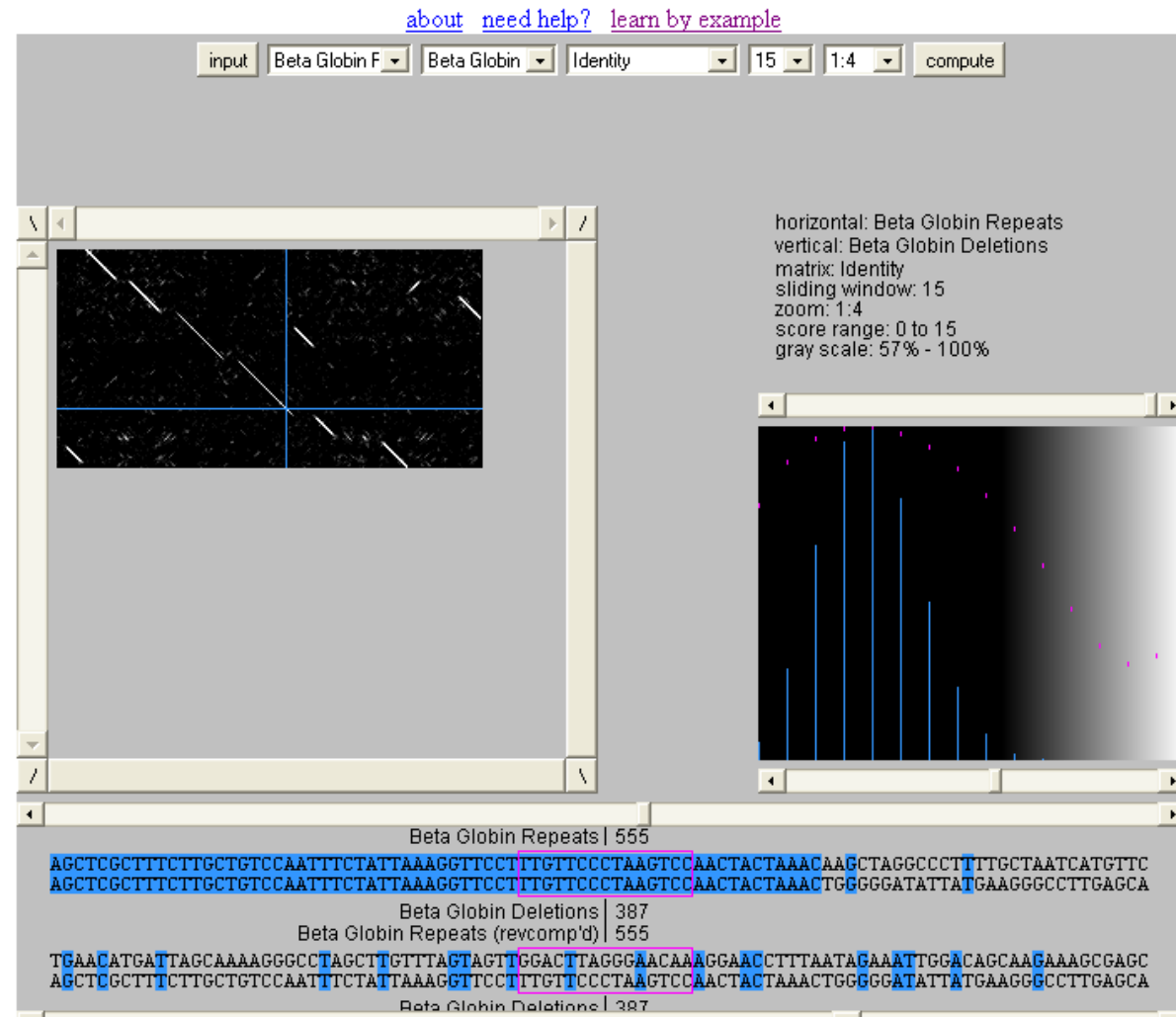
GCG software package:

- Compare <http://www.hku.hk/bruhk/gcgdoc/compare.html>
- DotPlot+ <http://www.hku.hk/bruhk/gcgdoc/dotplot.html>
- <http://www.isrec.isb-sib.ch/java/dotlet/Dotlet.html>
- <http://bioweb.pasteur.fr/cgi-bin/seqanal/dottup.pl>
- Dotter (<http://www.cgr.ki.se/cgr/groups/sonnhammer/Dotter.html>)

Available Dot Plot Programs

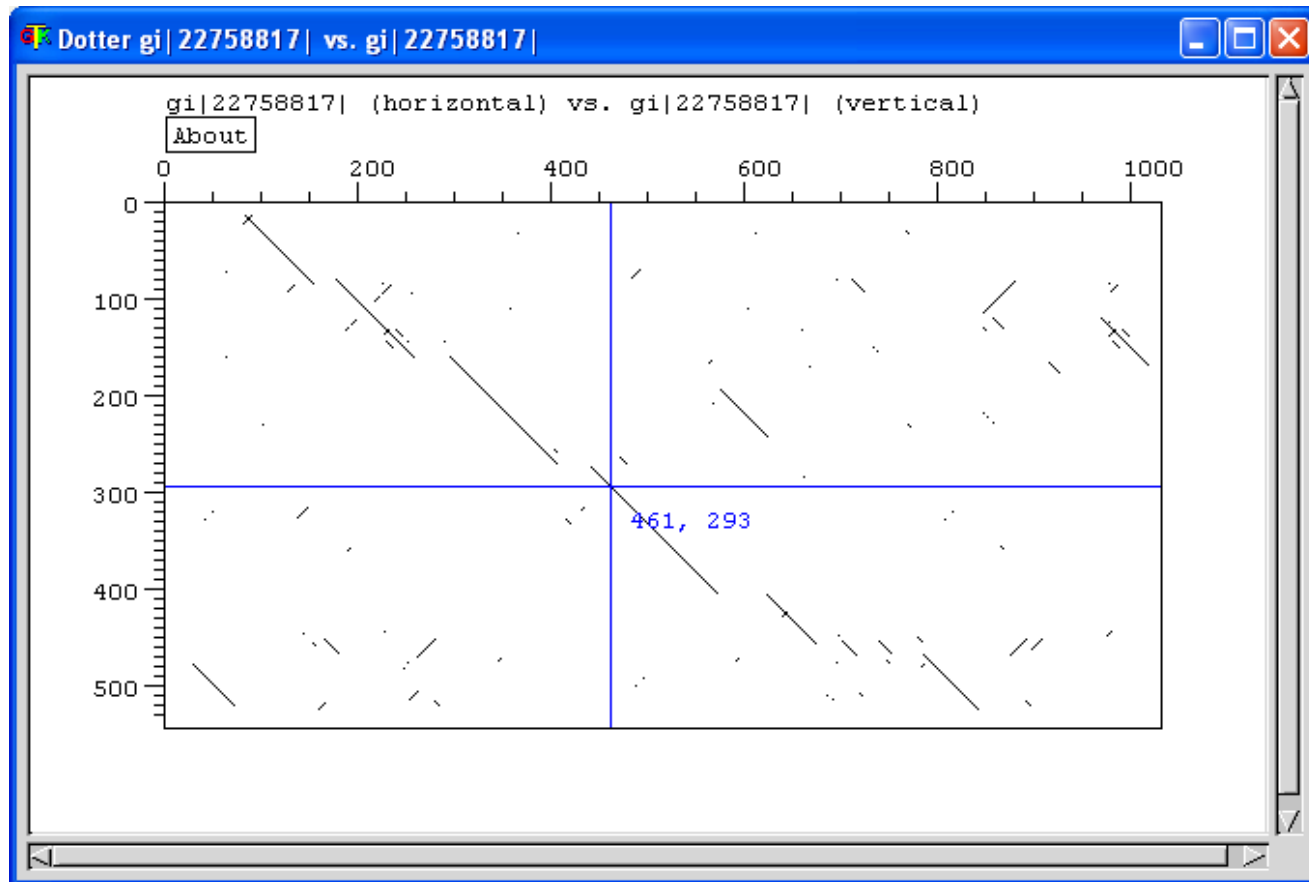
Dotlet (Java Applet)

<http://www.isrec.isb-sib.ch/java/dotlet/Dotlet.html>



Available Dot Plot Programs

Dotter (<http://www.cgr.ki.se/cgr/groups/sonnhammer/Dotter.html>)

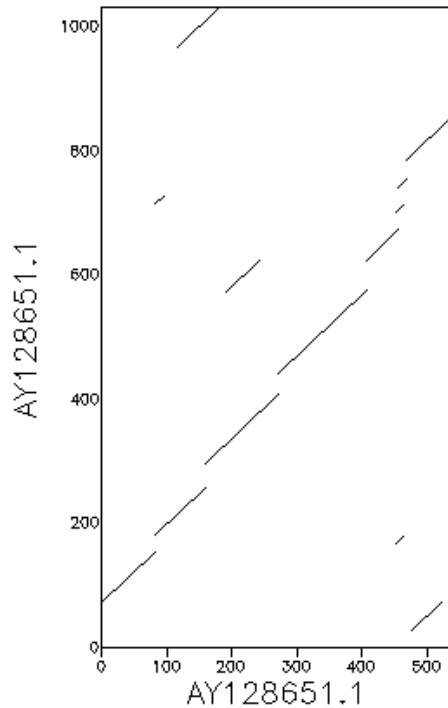


Available Dot Plot Programs

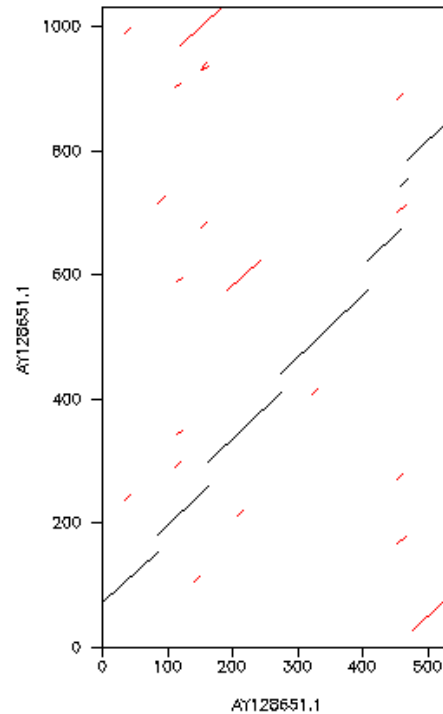
EMBOSS DotMatcher, DotPath, DotUp

Dotmatcher: AY128651.1 vs AY

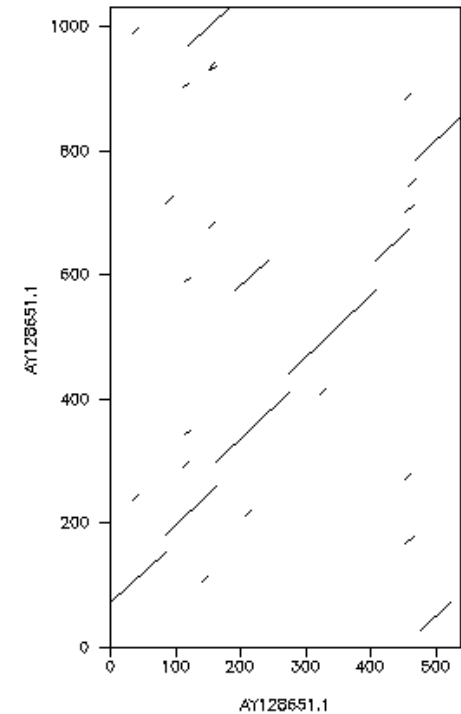
(window size = 10, threshold = 50.00 22/01/03)



dotpath (22/01/03)



dottup (22/01/03)



The Dot Matrix

- When to use the Dot Matrix method?
 - unless the sequences are known to be very much alike
- limits of the Dot Matrix
 - doesn't readily resolve similarity that is interrupted by insertion or deletions
 - Difficult to find the best possible alignment (optimal alignment)
 - most computer programs don't show an actual alignment

Dot Plot References

Gibbs, A. J. & McIntyre, G. A. (1970).

The diagram method for comparing sequences. its use with amino acid and nucleotide sequences.

Eur. J. Biochem. **16**, 1-11.

Staden, R. (1982).

An interactive graphics program for comparing and aligning nucleic-acid and amino-acid sequences.

Nucl. Acid. Res. **10** (9), 2951-2961.

Next step

We must define quantitative measures of sequence similarity and difference!

- *Hamming distance:*
 - *# of positions with mismatching characters*

AGTC
CGTA

Hamming distance = 2

- *Levenshtein (or edit) distance:*
 - *# of operations required to change one string into the other (deletion, insertion, substitution)*

AG-TCC
CGCTCA

Levenshtein distance = 3

Scoring

- +1 for a match -1 for a mismatch?
- should gaps be allowed?
 - if yes how should they be scored?
- what is the best algorithm for finding the optimal alignment of two sequences?
- is the produced alignment significant?

Scoring Matrices

- match/mismatch score
 - Not bad for similar sequences
 - Does not show distantly related sequences
- Likelihood matrix
 - Scores residues dependent upon likelihood substitution is found in nature
 - More applicable for amino acid sequences

Parameters of Sequence Alignment

Scoring Systems:

- Each symbol pairing is assigned a numerical value, based on a symbol comparison table.

Gap Penalties:

- Opening: The cost to introduce a gap
- Extension: The cost to elongate a gap

DNA Scoring Systems -very simple

Sequence 1

Sequence 2

actaccagttcatttgatacttctcaaa

 | | | | |
taccattaccgtgttaactgaaaggacttaaagact

	A	G	C	T
A	1	0	0	0
G	0	1	0	0
C	0	0	1	0
T	0	0	0	1

Match: 1
Mismatch: 0
Score = 5

Protein Scoring Systems

Sequence 1

Sequence 2

PTHPLASKTQILPEDLASEDLTI
 ||||| | | |
 PTHPLAGERAIGLARLAEEDFGM

T:G = -2

T:T = 5

Score = 48

Scoring
matrix

	C	S	T	P	A	G	N	D	.	.
C	9									
S	-1	4								
T	-1	1	5							
P	-3	-1	-1	7						
A	0	1	0	-1	4					
G	-3	0	-2	-2	0	6				
N	-3	1	0	-2	-2	0	5			
D	-3	0	-1	-1	-2	-1	1	6		
.										
.										

A scoring matrix is a table of values that describe the probability of a residue pair occurring in alignment.

Amino acid exchange matrices

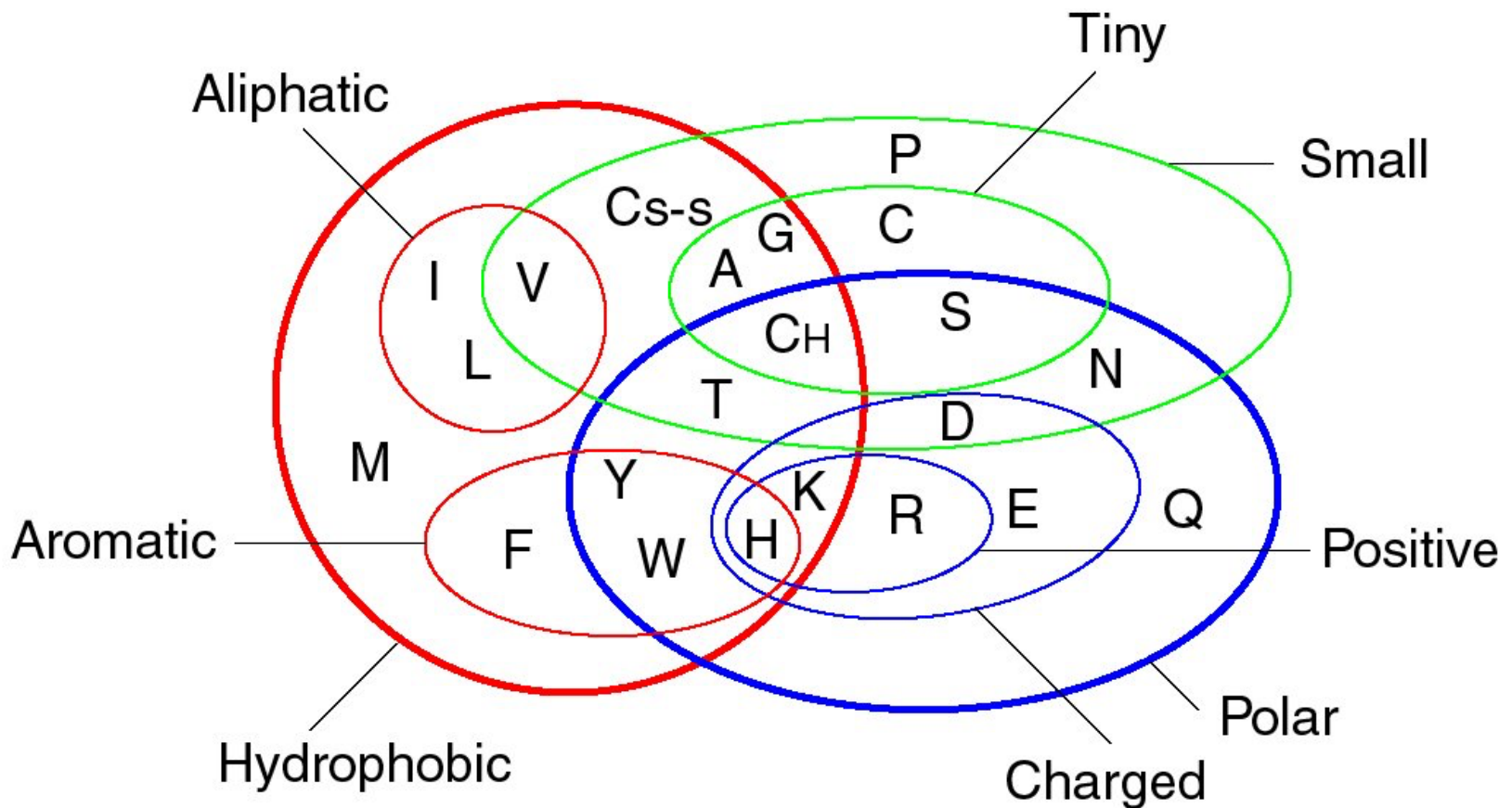
Amino acids are **not** equal:

1. Some are easily substituted because they have similar:
 - physico-chemical properties
 - structure
2. Some mutations between amino acids occur more often due to similar codons

The two above observations give us ways to define *substitution matrices*

Properties of Amino Acids

- Substitutions with similar chemical properties
- Amino acids have different biochemical and physical properties that influence their relative replaceability in evolution.



Scoring Matrices

- table of values that describe the probability of a residue pair occurring in an alignment
- the values are logarithms of ratios of two probabilities
 1. probability of random occurrence of an amino acid (diagonal)
 2. probability of meaningful occurrence of a pair of residues

BLOSUM62

Positive for chemically similar substitution

Common amino acids have low weights

Rare amino acids have high weights

A	4																				
R	-1	5																			
N	-2	0	6																		
D	-2	-2	1	6																	
C	0	-3	-3	-3	9																
Q	-1	1	0	0	-3	5															
E	-1	0	0	2	-4	2	5														
G	0	-2	0	-1	-3	-2	-2	6													
H	-2	0	1	-1	-3	0	0	-2	8												
I	-1	-3	-3	-3	-1	-3	-3	-4	-3	4											
L	-1	-2	-3	-4	-1	-2	-3	-4	-3	2	4										
K	-1	2	0	-1	-3	1	1	-2	-1	-3	-2	5									
M	-1	-1	-2	-3	-1	0	-2	-3	-2	1	2	-1	5								
F	-2	-3	-3	-3	-2	-3	-3	-3	-1	0	0	-3	0	6							
P	-1	-2	-2	-1	-3	-1	-1	-2	-2	-3	-3	-1	-2	-4	7						
S	1	-1	1	0	-1	0	0	0	-1	-2	-2	0	-1	-2	-1	4					
T	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	1	5				
W	-3	-3	-4	-4	-2	-2	-3	-2	-2	-3	-2	-3	-1	1	-4	-3	-2	11			
Y	-2	-2	-2	-3	-2	-1	-2	-3	2	-1	-1	-2	-1	3	-3	-2	-2	2	7		
V	0	-3	-3	-3	-1	-2	-2	-3	-3	3	1	-2	1	-1	-2	-2	0	-3	-1	4	
X	0	-1	-1	-1	-2	-1	-1	-1	-1	-1	-1	-1	-1	-1	-2	0	0	-2	-1	-1	-1
A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V	X	

Protein Scoring Systems

- Scoring matrices reflect:
 - # of mutations to convert one to another
 - chemical similarity
 - observed mutation frequencies
- Log odds matrices:
 - the values are logarithms of probability ratios of the probability of an aligned pair to the probability of a random alignment.
- Widely used scoring matrices:
 - PAM
 - BLOSUM

Scoring Matrices

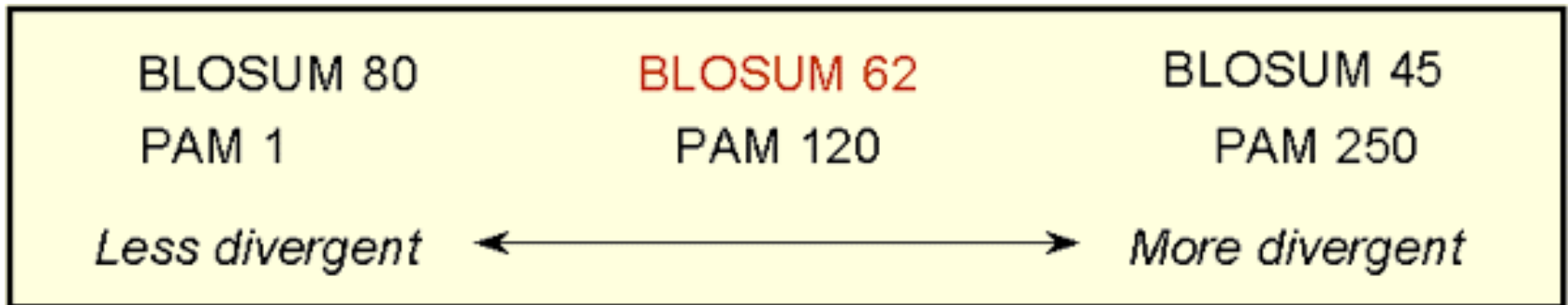
Widely used matrices

- **PAM** (Percent Accepted Mutation) / **MDM** (Mutation Data Matrix) / **Dayhoff**
 - Derived from *global* alignments of *closely* related sequences.
 - Matrices for greater evolutionary distances are extrapolated from those for lesser ones.
 - The number with the matrix (PAM40, PAM100) refers to the evolutionary distance; greater numbers are greater distances.
 - PAM-1 corresponds to about 1 million years of evolution
 - for distant (global) alignments, Blosum50, Gonnet, or (still) PAM250
- **BLOSUM** (Blocks Substitution Matrix)
 - Derived from *local, ungapped alignments of distantly related sequences*
 - *All matrices are directly calculated; no extrapolations are used*
 - *The number after the matrix (BLOSUM62) refers to the minimum percent identity of the blocks used to construct the matrix; greater numbers are lesser distances.*
 - *The BLOSUM series of matrices generally perform better than PAM matrices for local similarity searches.*
 - For local alignment, Blosum 62 is often superior
- **Structure-based matrices**
- **Specialized Matrices**

Scoring Matrices

The relationship between BLOSUM and PAM substitution matrices

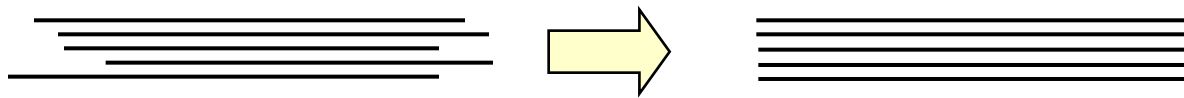
- BLOSUM matrices with higher numbers and PAM matrices with low numbers are designed for comparisons of closely related sequences.
- BLOSUM matrices with low numbers and PAM matrices with high numbers are designed for comparisons of distantly related proteins.



<http://www.ncbi.nlm.nih.gov/Education/BLASTinfo/Scoring2.html>

Percent Accepted Mutation (PAM or Dayhoff) Matrices

- Studied by Margaret Dayhoff (Dayhoff *et al.*, 1978).
- Amino acid substitutions
 - Alignment of common protein sequences
 - 1572 amino acid substitutions
 - 71 groups of protein, 85% similar



Derived from global alignments of protein families. Family members share at least 85% identity.

- “Accepted” mutations – do not negatively affect a protein’s fitness
- Similar sequences organized into phylogenetic trees
- Number of amino acid changes counted
- Relative mutabilities evaluated
- 20 x 20 amino acid substitution matrix calculated

Percent Accepted Mutation (PAM or Dayhoff) Matrices

- PAM 1: 1 accepted mutation event per 100 amino acids; PAM 250: 250 mutation events per 100 ...
- PAM 1 matrix can be multiplied by itself N times to give transition matrices for sequences that have undergone N mutations
- PAM 250: 20% similar; PAM 120: 40%; PAM 80: 50%; PAM 60: 60%

PAM1 matrix

normalized probabilities multiplied by 10000

	Ala	Arg	Asn	Asp	Cys	Gln	Glu	Gly	His	Ile	Leu	Lys	Met	Phe	Pro	Ser	Thr	Trp	Tyr	Val
	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
A	9867	2	9	10	3	8	17	21	2	6	4	2	6	2	22	35	32	0	2	18
R	1	9913	1	0	1	10	0	0	10	3	1	19	4	1	4	6	1	8	0	1
N	4	1	9822	36	0	4	6	6	21	3	1	13	0	1	2	20	9	1	4	1
D	6	0	42	9859	0	6	53	6	4	1	0	3	0	0	1	5	3	0	0	1
C	1	1	0	0	9973	0	0	0	1	1	0	0	0	0	1	5	1	0	3	2
Q	3	9	4	5	0	9876	27	1	23	1	3	6	4	0	6	2	2	0	0	1
E	10	0	7	56	0	35	9865	4	2	3	1	4	1	0	3	4	2	0	1	2
G	21	1	12	11	1	3	7	9935	1	0	1	2	1	1	3	21	3	0	0	5
H	1	8	18	3	1	20	1	0	9912	0	1	1	0	2	3	1	1	1	4	1
I	2	2	3	1	2	1	2	0	0	9872	9	2	12	7	0	1	7	0	1	33
L	3	1	3	0	0	6	1	1	4	22	9947	2	45	13	3	1	3	4	2	15
K	2	37	25	6	0	12	7	2	2	4	1	9926	20	0	3	8	11	0	1	1
M	1	1	0	0	0	2	0	0	0	5	8	4	9874	1	0	1	2	0	0	4
F	1	1	1	0	0	0	0	1	2	8	6	0	4	9946	0	2	1	3	28	0
P	13	5	2	1	1	8	3	2	5	1	2	2	1	1	9926	12	4	0	0	2
S	28	11	34	7	11	4	6	16	2	2	1	7	4	3	17	9840	38	5	2	2
T	22	2	13	4	1	3	2	2	1	11	2	8	6	1	5	32	9871	0	2	9
W	0	2	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	9976	1	0
Y	1	0	3	0	3	0	1	0	4	1	1	0	0	21	0	1	1	2	9945	1
V	13	2	1	1	3	2	2	3	3	57	11	1	17	1	3	2	10	0	2	9901

Log Odds Matrices

- PAM matrices converted to log-odds matrix
 - Calculate odds ratio for each substitution
 - Taking scores in previous matrix
 - Divide by frequency of amino acid
 - Convert ratio to \log_{10} and multiply by 10
 - Take average of log odds ratio for converting A to B and converting B to A
 - Result: Symmetric matrix

PAM 250

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V	B	Z
A	2	-2	0	0	0	0	0	1	-1	-1	-2	-1	-1	-3	1	1	1	-3	0	2	1	
R	-2	6	0	-1	1	-1	-3	2	-2	-3	3	0	-4	0	0	-1	-1	-4	-2	1	2	
N	0	0	2	2	-4	1	1	0	2	-2	-3	1	-2	-3	0	1	0	-4	-2	-2	4	3
D	0	-1	2	4	-5	2	3	1	1	-2	-4	0	-3	-6	-1	0	0	-7	-4	-2	5	4
C	-2	-4	-4	-5	12	-5	-5	-3	-3	-2	-6	-5	-5	-4	-3	0	-2	-8	0	-2	-3	-4
Q	0	1	1	2	-5	4	2	-1	3	-2	-2	1	-1	-5	0	-1	-1	-5	-4	-2	3	5
E	0	-1	1	3	-5	2	4	0	1	-2	-3	0	-2	-5	-1	0	0	-7	-4	-2	4	5
G	1	-3	0	1	-3	-1	0	5	-2	-3	-4	-2	-3	-5	0	1	0	-7	-5	-1	2	1
H	-1	2	2	1	-3	3	1	-2	6	-2	-2	0	-2	-2	0	-1	-1	-3	0	-2	3	3
I	-1	-2	-2	-2	-2	-2	-2	-3	-2	5	2	-2	2	1	-2	-1	0	-5	-1	4	-1	-1
L	-2	-3	-3	-4	-6	-2	-3	-4	-2	2	6	-3	4	2	-3	-3	-2	-2	-1	2	-2	-1
K	-1	3	1	0	-5	1	0	-2	0	-2	-3	5	0	-5	-1	0	0	-3	-4	-2	2	2
M	-1	0	-2	-3	-5	-1	-2	-3	-2	2	4	0	6	0	-2	-2	-1	-4	-2	2	-1	0
F	-3	-4	-3	-6	-4	-5	-5	-5	-2	1	2	-5	0	9	-5	-3	-3	0	7	-1	-3	-4
P	1	0	0	-1	-3	0	-1	0	0	-2	-3	-1	-2	-5	6	1	0	-6	-5	-1	1	1
S	1	0	1	0	0	-1	0	1	-1	-1	-3	0	-2	-3	1	2	1	-2	-3	-1	2	1
T	1	-1	0	0	-2	-1	0	0	-1	0	-2	0	-1	-3	0	1	3	-5	-3	0	2	1
W	-6	2	-4	-7	-8	-5	-7	-7	-3	-5	-2	-3	-4	0	-6	-2	-5	17	0	-6	-4	-4
Y	-3	-4	-2	-4	0	-4	-4	-5	0	-1	-1	-4	-2	7	-5	-3	-3	0	10	-2	-2	-3
W	0	-2	-2	2	-8	2	-2	-1	-2	4	2	-2	2	-1	-1	-1	0	17	2	4	0	0
	2	1	4	5	-8	3	4	2	3	-1	-2	2	-1	-3	1	2	2	2	0	6	5	
Z	1	2	3	4	-4	5	5	1	3	-1	-1	2	0	-4	1	1	1	-4	-3	0	5	6

A value of 0 indicates the frequency of alignment is random

$$\log(\text{freq}(\text{observed})/\text{freq}(\text{expected}))$$

PAM250 Log odds matrix

	C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W	
C	12																				C
S	0	2																			S
T	-2	1	3																		T
P	-3	1	0	6																	P
A	-2	1	1	1	2																A
G	-3	1	0	-1	1	5															G
N	-4	1	0	-1	0	0	2														N
D	-5	0	0	-1	0	1	2	4													D
E	-5	0	0	-1	0	0	1	3	4												E
Q	-5	-1	-1	0	0	-1	1	2	2	4											Q
H	-3	-1	-1	0	-1	-2	2	1	1	3	6										H
R	-4	0	-1	0	-2	-3	0	-1	-1	1	2	6									R
K	-5	0	0	-1	-1	-2	1	0	0	1	0	3	5								K
M	-5	-2	-1	-2	-1	-3	-2	-3	-2	-1	-2	0	0	6							M
I	-2	-1	0	-2	-1	-3	-2	-2	-2	-2	-2	-2	-2	2	5						I
L	-6	-3	-2	-3	-2	-4	-3	-4	-3	-2	-2	-3	-3	4	2	6					L
V	-2	-1	0	-1	0	-1	-2	-2	-2	-2	-2	-2	-2	2	4	2	4				V
F	-4	-3	-3	-5	-4	-5	-4	-6	-5	-5	-2	-4	-5	0	1	2	-1	9			F
Y	0	-3	-3	-5	-3	-5	-2	-4	-4	-4	0	-4	-4	-2	-1	-1	-2	7	10		Y
W	-8	-2	-5	-6	-6	-7	-4	-7	-7	-5	-3	2	-3	-4	-5	-2	-6	0	0	17	W
	C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W	

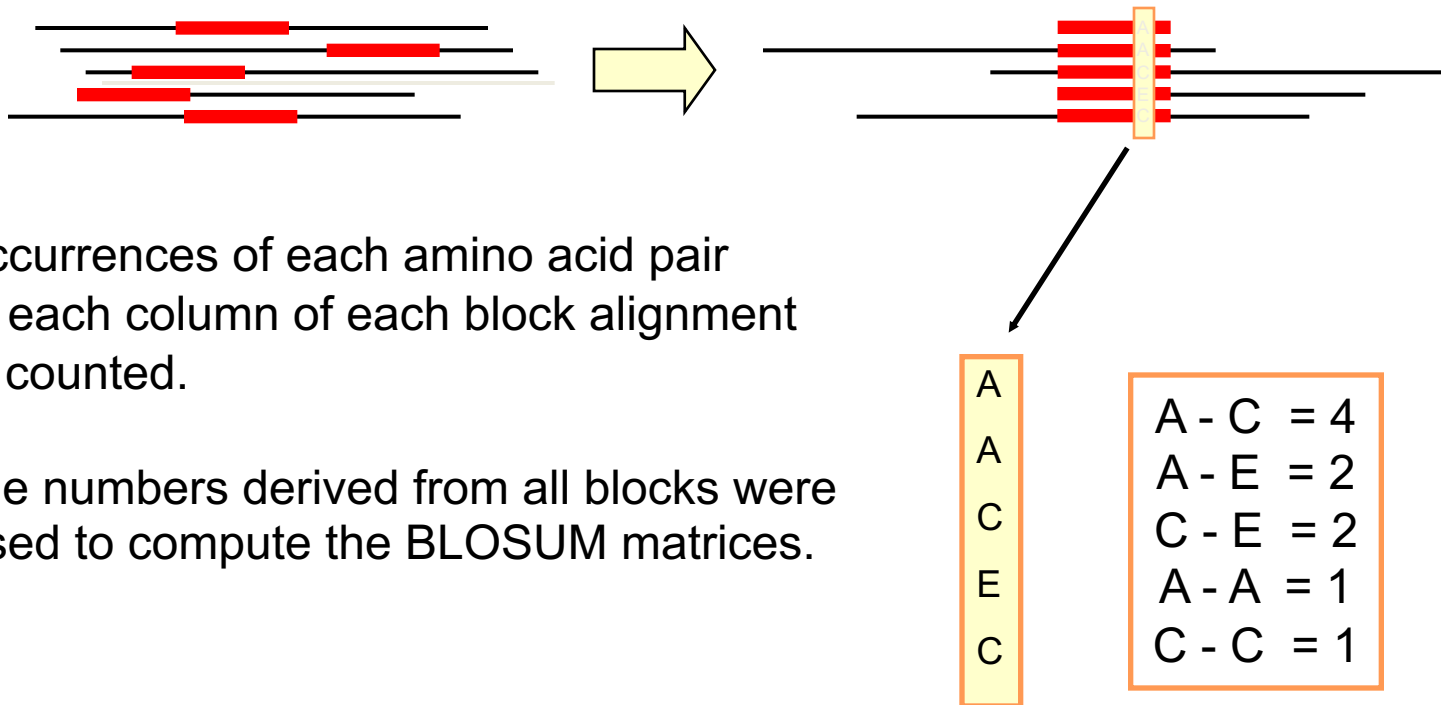
Blocks Amino Acid Substitution Matrices (BLOSUM)

- Larger set of sequences considered
- Sequences organized into signature blocks
- Consensus sequence formed
 - 60% identical: BLOSUM 60
 - 80% identical: BLOSUM 80

BLOSUM

(Blocks Substitution Matrix)

- Derived from alignments of domains of distantly related proteins (Henikoff & Henikoff, 1992).



- Occurrences of each amino acid pair in each column of each block alignment is counted.
- The numbers derived from all blocks were used to compute the BLOSUM matrices.

BLOSUM

(Blocks Substitution Matrix)

- Sequences within blocks are clustered according to their level of identity.
- Clusters are counted as a single sequence.
- Different BLOSUM matrices differ in the percentage of sequence identity used in clustering.
- The number in the matrix name (e.g. 62 in BLOSUM62) refers to the percentage of sequence identity used to build the matrix.
- Greater numbers mean smaller evolutionary distance.

AMINO ACID FREQUENCY

- Genetic information contained in mRNA is in the form of **codons**, which are translated into **amino acids** which then combine to form **proteins**.
- At certain sites in a protein's structure, amino acid composition is not critical.
- Yet certain amino acids occur at such sites up to six times more often than other amino acids.
- Are frequencies of particular amino acids simply a consequence of random permutations of the genetic code or instead a product of natural selection?

AMINO ACID FREQUENCY

- If a particular amino acid is in some way adaptive, then it should occur more frequently than expected by chance.
- This can easily be tested by calculating the **expected frequencies** of amino acids and comparing to **observed**.
- The codons and observed frequencies of particular amino acids are given in the next slide.

The codons and observed frequencies of amino acids

Amino Acids	Codons	Observed Frequency in Vertebrates
Alanine	GCU, GCA, GCC, GCG	7.4 %
Arginine	CGU, CGA, CGC, CGG, AGA, AGG	4.2 %
Asparagine	AAU, AAC	4.4 %
Aspartic Acid	GAU, GAC	5.9 %
Cysteine	UGU, UGC	3.3 %
Glutamic Acid	GAA, GAG	5.8 %
Glutamine	CAA, CAG	3.7 %
Glycine	GGU, GGA, GGC, GGG	7.4 %
Histidine	CAU, CAC	2.9 %

The codons and observed frequencies of amino acids

Isoleucine	AUU, AUA, AUC	3.8 %
Leucine	CUU, CUA, CUC, CUG, UUA, UUG	7.6 %
Lysine	AAA, AAG	7.2 %
Methionine	AUG	1.8 %
Phenylalanine	UUU, UUC	4.0 %
Proline	CCU, CCA, CCC, CCG	5.0 %
Serine	UCU, UCA, UCC, UCG, AGU, AGC	8.1 %
Threonine	ACU, ACA, ACC, ACG	6.2 %
Tryptophan	UGG	1.3 %
Tyrosine	UAU, UAC	3.3 %
Valine	GUU, GUA, GUC, GUG	6.8 %
Stop Codons	UAA, UAG, UGA	---

AMINO ACID FREQUENCY

- The frequencies of DNA bases in nature are
 - 22.0% uracil,
 - 30.3% adenine,
 - 21.7% cytosine,
 - 26.1% guanine.
- The expected frequency of a particular codon can then be calculated by
 - multiplying the frequencies of each DNA base comprising the codon.

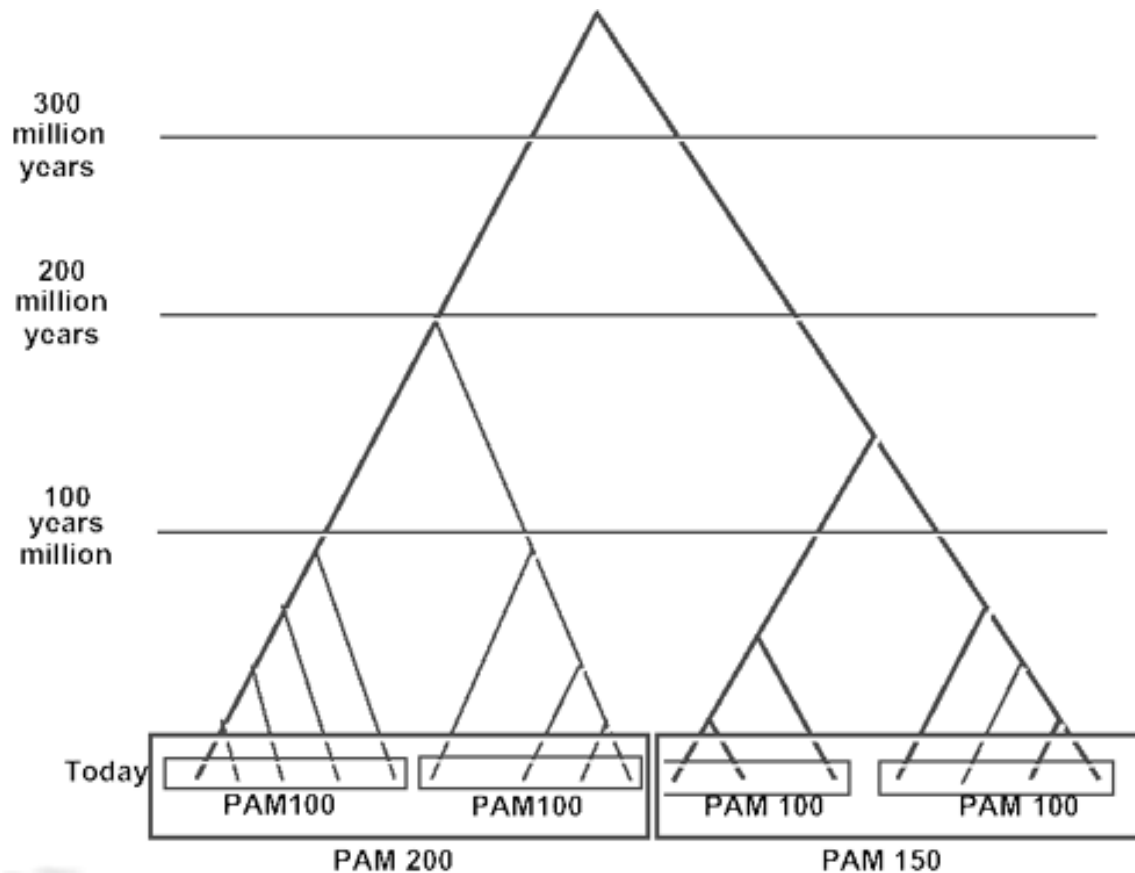
AMINO ACID FREQUENCY

- The expected frequency of the amino acid can then be calculated by adding the frequencies of each codon that codes for that amino acid.
- As an example,
 - the RNA codons for tyrosine are UAU and UAC, so the random expectation for its frequency is
 - $(0.220)(0.303)(0.220) + (0.220)(0.303)(0.217) = 0.0292$.
 - Since 3 of the 64 codons are nonsense or stop codons, this frequency for each amino acid is multiplied by a correction factor of 1.057.

TIPS on choosing a scoring matrix

- Generally, BLOSUM matrices perform better than PAM matrices for local similarity searches (Henikoff & Henikoff, 1993).
- When comparing **closely related** proteins one should use **lower PAM or higher BLOSUM** matrices,
- For **distantly related** proteins **higher PAM or lower BLOSUM** matrices.
- For database searching the commonly used matrix is BLOSUM62.

Use Different PAM's for Different Evolutionary Distances



Nucleic Acid Scoring Scheme

- Transition mutation (more common)

- purine \longleftrightarrow purine A \longleftrightarrow G
- pyrimidine \longleftrightarrow pyrimidine T \longleftrightarrow C

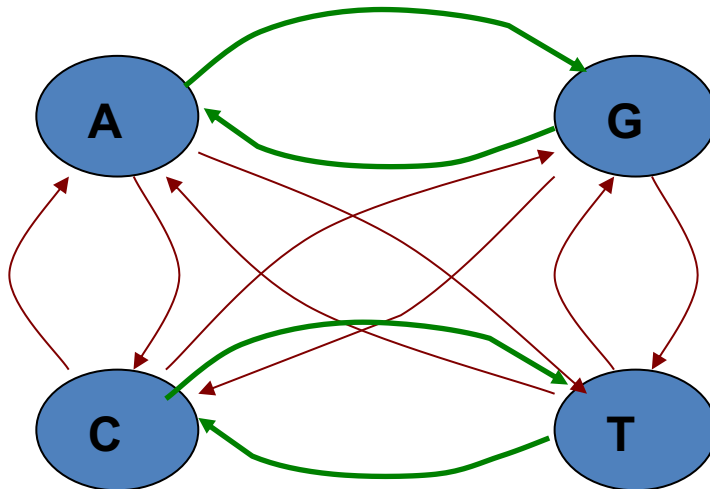
- Transversion mutation

- purine \longleftrightarrow pyrimidine A, G \longleftrightarrow T, C

	A	G	T	C
A	20	10	5	5
G	10	20	5	5
T	5	5	20	10
C	5	5	10	20

DNA Mutations

In addition to using a match/mismatch scoring scheme for DNA sequences, nucleotide mutation matrices can be constructed as well. These matrices are based upon two different models of nucleotide evolution: the first, the Jukes-Cantor model, assumes there are uniform mutation rates among nucleotides, while the second, the Kimura model, assumes that there are two separate mutation rates: one for transitions (where the structure of purine/pyrimidine stays the same), and one for transversions. Generally, the rate of transitions is thought to be higher than the rate of transversions.



PURINES: A, G
PYRIMIDINES C, T

Transitions: $A \leftrightarrow G$; $C \leftrightarrow T$
Transversions: $A \leftrightarrow C$, $A \leftrightarrow T$,
 $C \leftrightarrow G$, $G \leftrightarrow T$

Nucleic Acid Scoring Matrices

- Two mutation models:
 - Jukes-Cantor Model of evolution: α = common rate of base substitution
 - Kimura Model of Evolution: α = rate of transitions; β = rate of transversions
 - Transitions
 - Transversions

$$R = \begin{matrix} & \begin{matrix} A & C & G & U \end{matrix} \\ \begin{matrix} A \\ C \\ G \\ U \end{matrix} & \begin{pmatrix} * & 0.25\alpha & 0.25\alpha & 0.25\alpha \\ 0.25\alpha & * & 0.25\alpha & 0.25\alpha \\ 0.25\alpha & 0.25\alpha & * & 0.25\alpha \\ 0.25\alpha & 0.25\alpha & 0.25\alpha & * \end{pmatrix} \end{matrix}$$

$$R = \begin{matrix} & \begin{matrix} A & C & G & U \end{matrix} \\ \begin{matrix} A \\ C \\ G \\ U \end{matrix} & \begin{pmatrix} * & 0.25\beta & 0.25\alpha & 0.25\beta \\ 0.25\beta & * & 0.25\beta & 0.25\alpha \\ 0.25\alpha & 0.25\beta & * & 0.25\beta \\ 0.25\beta & 0.25\alpha & 0.25\beta & * \end{pmatrix} \end{matrix}$$

Nucleotide substitution matrices with the equivalent distance of 1 PAM

A. Model of uniform mutation rates among nucleotides.

	A	G	T	C
A	0.99			
G	0.00333	0.99		
T	0.00333	0.00333	0.99	
C	0.00333	0.00333	0.00333	0.99

B. Model of 3-fold higher transitions than transversions.

A	G	T	C	
A	0.99			
G	0.006	0.99		
T	0.002	0.002	0.99	
C	0.002	0.002	0.006	0.99

Nucleotide substitution matrices with the equivalent distance of 1 PAM

E.g. take $\sim \ln$ (logarithm base e) of the matrix in the previous slide (for non-diagonal entries):

A. Model of uniform mutation rates among nucleotides.

	A	G	T	C
A	2			
G	-6	2		
T	-6	-6	2	
C	-6	-6	-6	2

B. Model of 3-fold higher transitions than transversions.

	A	G	T	C
A	2			
G	-5	2		
T	-7	-7	2	
C	-7	-7	-5	2

Determining Optimal Alignment

- Two sequences: X and Y
 - $|X| = m$; $|Y| = n$
 - Allowing gaps, $|X| = |Y| = m+n$
- Brute Force
- Dynamic Programming

Brute Force

- Determine all possible subsequences for X and Y
 - 2^{m+n} subsequences for X, 2^{m+n} for Y!
- Alignment comparisons
 - $2^{m+n} * 2^{m+n} = 2^{(2(m+n))} = 4^{m+n}$ comparisons
- Quickly becomes impractical

Dynamic Programming

- Used in Computer Science
- Solve optimization problems by dividing the problem into independent subproblems
- Sequence alignment has optimal substructure property
 - Subproblem: alignment of prefixes of two sequences
 - Each subproblem is computed once and stored in a matrix

Dynamic Programming

- Optimal score: built upon optimal alignment computed to that point
- Aligns two sequences beginning at ends, attempting to align all possible pairs of characters

Dynamic Programming

- Scoring scheme for matches, mismatches, gaps
- Highest set of scores defines optimal alignment between sequences
- Match score: DNA – exact match; Amino Acids – mutation probabilities
- Guaranteed to provide optimal alignment given:
 - Two sequences
 - Scoring scheme

Steps in Dynamic Programming

- Initialization
- Matrix Fill (scoring)
- Traceback (alignment)

DP Example:

Sequence #1: GAATTCAGTTA; $M = 11$

Sequence #2: GGATCGA; $N = 7$

- $s(a_i b_j) = +5$ if $a_i = b_j$ (match score)
- $s(a_i b_j) = -3$ if $a_i \neq b_j$ (mismatch score)
- $w = -4$ (gap penalty)

View of the DP Matrix

- $M+1$ rows, $N+1$ columns

	-	G	A	A	T	T	C	A	G	T	T	A
-												
G												
G												
A												
T												
C												
G												
A												

Global Alignment (Needleman-Wunsch)

- Attempts to align all residues of two sequences
- ***INITIALIZATION***: First row and first column set
- $S_{i,0} = w * i$
- $S_{0,j} = w * j$

Initialized Matrix(Needleman-Wunsch)

	-	G	A	A	T	T	C	A	G	T	T	A
-	0	-4	-8	-12	-16	-20	-24	-28	-32	-36	-40	-44
G	-4											
G	-8											
A	-12											
T	-16											
C	-20											
G	-24											
A	-28											

Matrix Fill (Global Alignment)

$$S_{i,j} = \text{MAXIMUM}[$$
$$S_{i-1,j-1} + s(a_i, b_j) \text{ (match/mismatch in the diagonal),}$$
$$S_{i,j-1} + w \text{ (gap in sequence \#1),}$$
$$S_{i-1,j} + w \text{ (gap in sequence \#2)}$$
$$]$$

Matrix Fill (Global Alignment)

- $S_{1.1} = \text{MAX}[S_{0.0} + 5, S_{1.0} - 4, S_{0.1} - 4] = \text{MAX}[5, -8, -8]$

	-	G	A	A	T	T	C	A	G	T	T	A
-	0	-4	-8	-12	-16	-20	-24	-28	-32	-36	-40	-44
G	-4	5										
G	-8											
A	-12											
T	-16											
C	-20											
G	-24											
A	-28											

Matrix Fill (Global Alignment)

- $S_{1,2} = \text{MAX}[S_{0,1} - 3, S_{1,1} - 4, S_{0,2} - 4] = \text{MAX}[-4 - 3, 5 - 4, -8 - 4] = \text{MAX}[-7, 1, -12]$

	-	G	A	A	T	T	C	A	G	T	T	A
-	0	-4	-8	-12	-16	-20	-24	-28	-32	-36	-40	-44
G	-4	5	1									
G	-8											
A	-12											
T	-16											
C	-20											
G	-24											
A	-28											

Matrix Fill (Global Alignment)

	-	G	A	A	T	T	C	A	G	T	T	A
-	0	-4	-8	-12	-16	-20	-24	-28	-32	-36	-40	-44
G	-4	5	1	-3	-7	-11	-15	-19	-23	-27	-31	-35
G	-8											
A	-12											
T	-16											
C	-20											
G	-24											
A	-28											

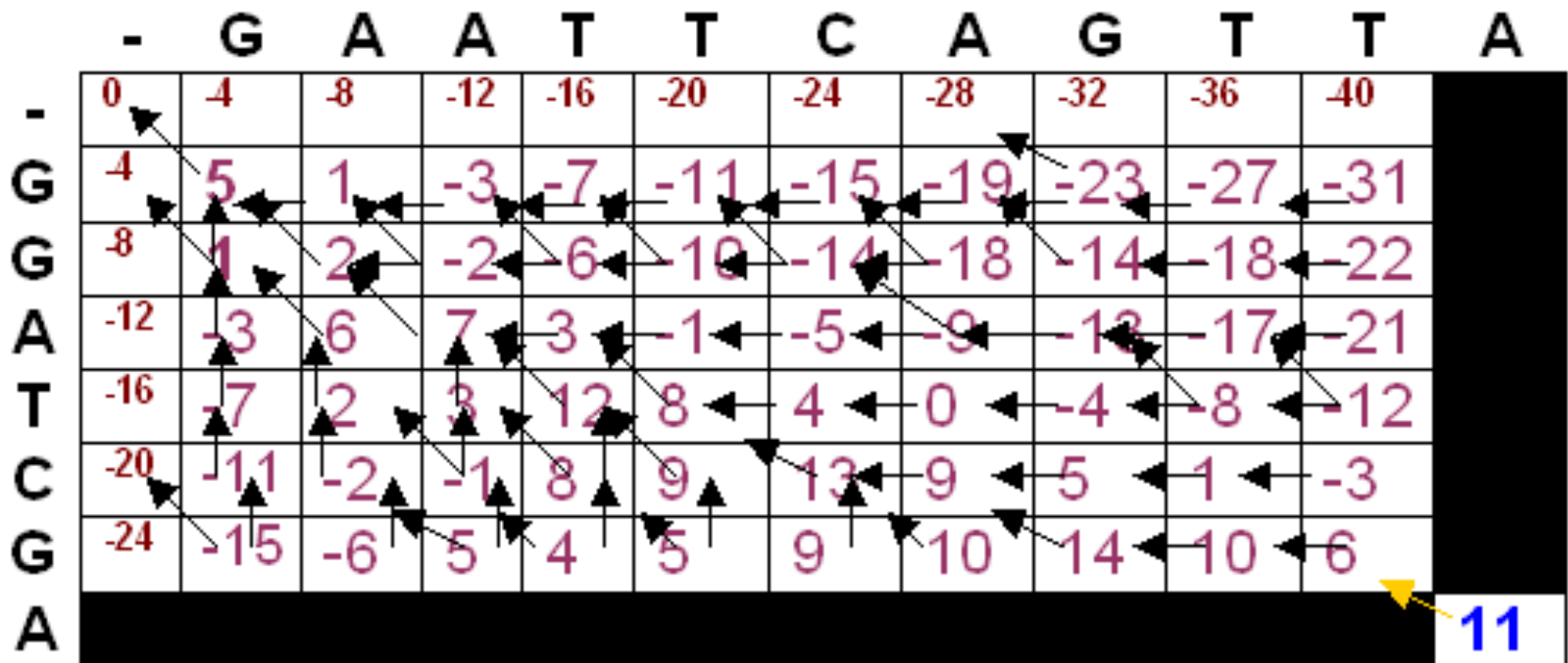
Filled Matrix (Global Alignment)

	-	G	A	A	T	T	C	A	G	T	T	A
-	0	-4	-8	-12	-16	-20	-24	-28	-32	-36	-40	-44
G	-4	5	1	-3	-7	-11	-15	-19	-23	-27	-31	-35
G	-8	1	2	-2	-6	-10	-14	-18	-14	-18	-22	-26
A	-12	-3	6	7	3	-1	-5	-9	-13	-17	-21	-17
T	-16	-7	2	3	12	8	4	0	-4	-8	-12	-16
C	-20	-11	-2	-1	8	9	13	9	5	1	-3	-7
G	-24	-15	-6	-5	4	5	9	10	14	10	6	2
A	-28	-19	-10	-1	0	1	5	14	10	11	7	11

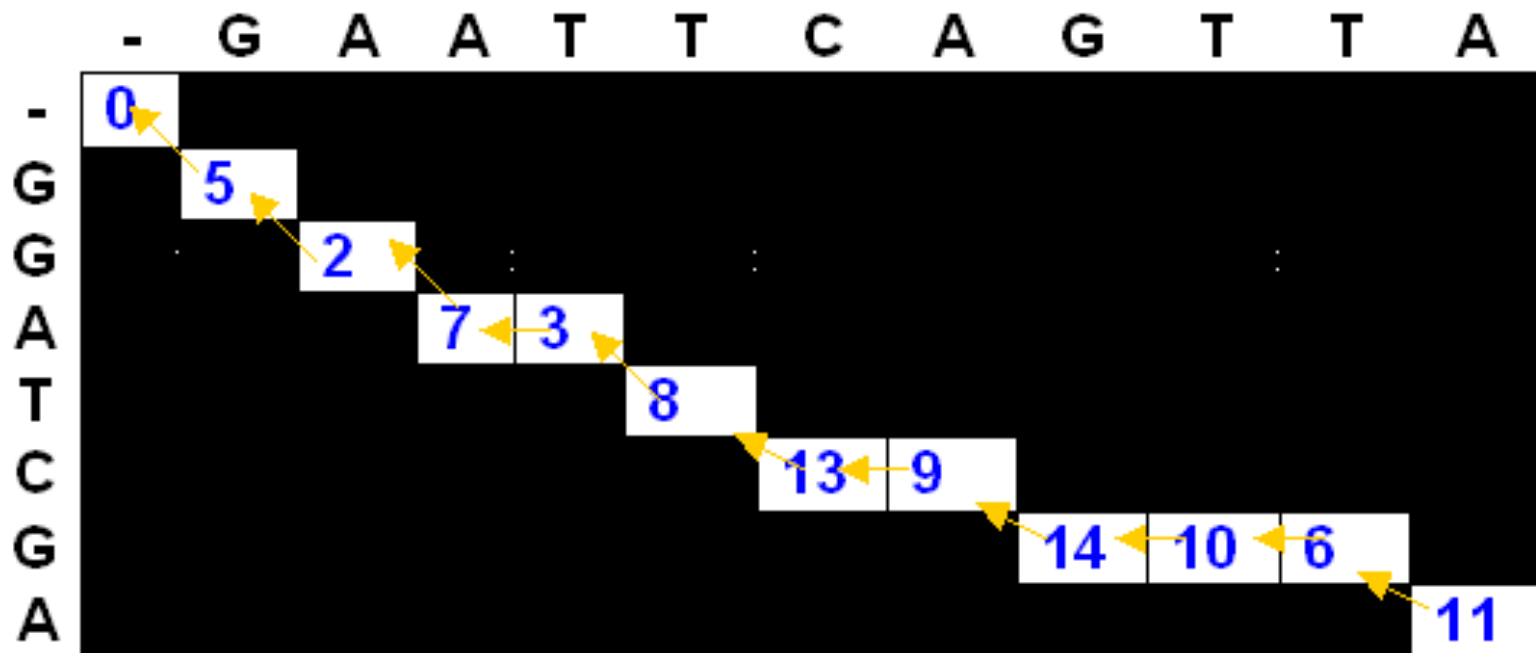
Trace Back (Global Alignment)

- maximum global alignment score = 11 (value in the lower right hand cell).
- Traceback begins in position $S_{M,N}$; i.e. the position where both sequences are globally aligned.
- At each cell, we look to see where we move next according to the pointers.

Trace Back (Global Alignment)



Global Trace Back



G A A T T C A G T T A
 | | | | | |
 G G A - T C - G - - A

Checking Alignment Score

G	A	A	T	T	C	A	G	T	T	A
G	G	A	-	T	C	-	G	-	-	A
+	-	+	-	+	+	-	+	-	-	+
5	3	5	4	5	5	4	5	4	4	5

$$5 - 3 + 5 - 4 + 5 + 5 - 4 + 5 - 4 - 4 + 5 = 11 \checkmark$$

Local Alignment

Local Alignment

- Smith-Waterman: obtain highest scoring local match between two sequences
- Requires a modification:
 - When a value in the score matrix becomes negative, reset it to zero (begin of new alignment)

Local Alignment Initialization

- Values in row 0 and column 0 set to 0.

	-	G	A	A	T	T	C	A	G	T	T	A
-	0	0	0	0	0	0	0	0	0	0	0	0
G	0											
G	0											
A	0											
T	0											
C	0											
G	0											
A	0											

Matrix Fill (Local Alignment)

$$S_{i,j} = \text{MAXIMUM} [$$
$$S_{i-1,j-1} + s(a_i, b_j) \text{ (match/mismatch in the diagonal),}$$
$$S_{i,j-1} + w \text{ (gap in sequence \#1),}$$
$$S_{i-1,j} + w \text{ (gap in sequence \#2),}$$
$$0]$$

Matrix Fill (Local Alignment)

$$S_{1,1} = \text{MAX}[S_{0,0} + 5, S_{1,0} - 4, S_{0,1} - 4, 0] = \text{MAX}[5, -4, -4, 0] = 5$$

	-	G	A	A	T	T	C	A	G	T	T	A
-	0	0	0	0	0	0	0	0	0	0	0	0
G	0	5										
G	0											
A	0											
T	0											
C	0											
G	0											
A	0											

Matrix Fill (Local Alignment)

$$S_{1,2} = \text{MAX}[S_{0,1} - 3, S_{1,1} - 4, S_{0,2} - 4, 0] = \text{MAX}[0 - 3, 5 - 4, 0 - 4, 0] = \text{MAX}[-3, 1, -4, 0] = 1$$

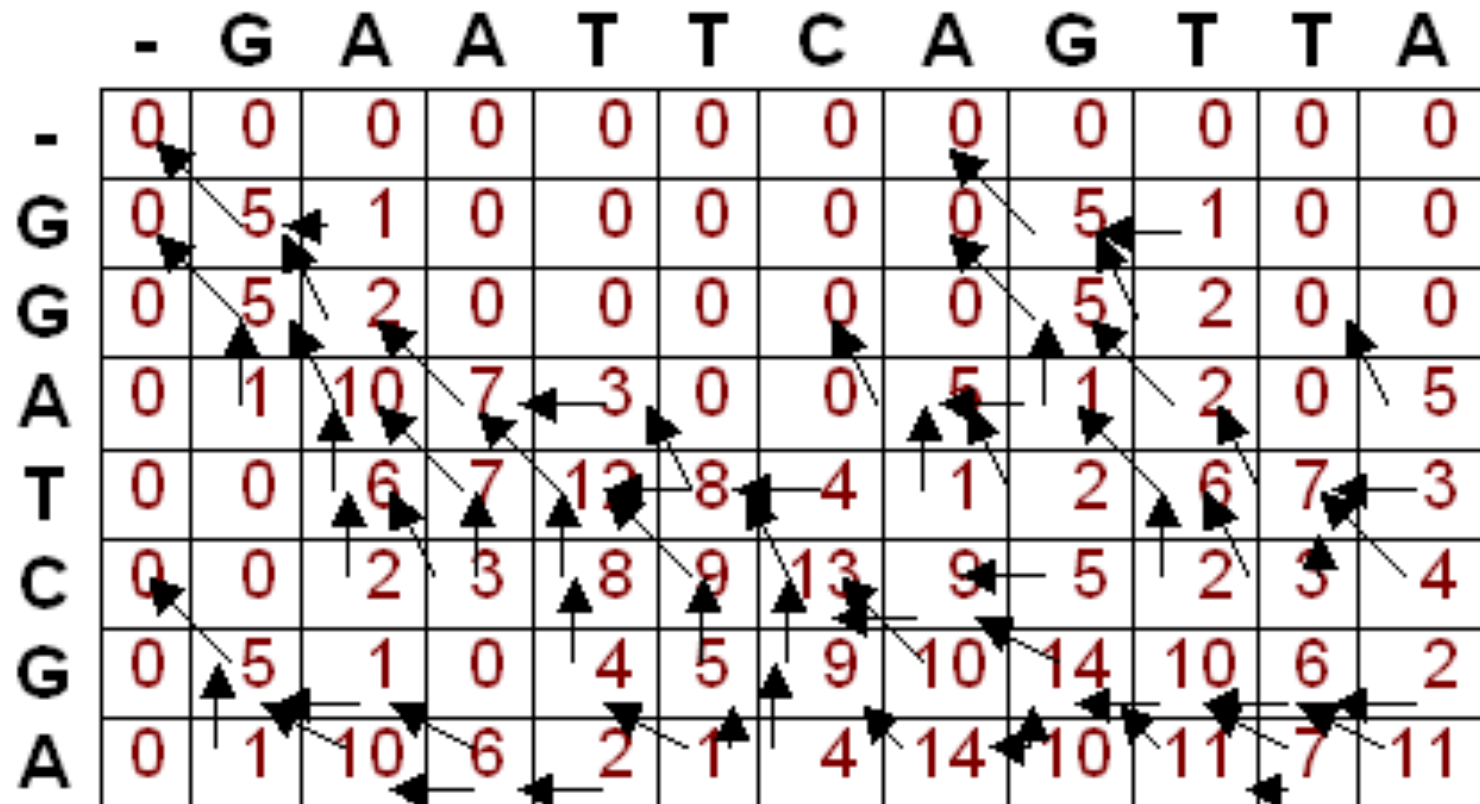
	-	G	A	A	T	T	C	A	G	T	T	A
-	0	0	0	0	0	0	0	0	0	0	0	0
G	0	5	1									
G	0											
A	0											
T	0											
C	0											
G	0											
A	0											

Matrix Fill (Local Alignment)

$$S_{1,3} = \text{MAX}[S_{0,2} - 3, S_{1,2} - 4, S_{0,3} - 4, 0] = \text{MAX}[0 - 3, 1 - 4, 0 - 4, 0] = \\ \text{MAX}[-3, -3, -4, 0] = 0$$

	-	G	A	A	T	T	C	A	G	T	T	A
-	0	0	0	0	0	0	0	0	0	0	0	0
G	0	5	1	0								
G	0											
A	0											
T	0											
C	0											
G	0											
A	0											

Filled Matrix (Local Alignment)



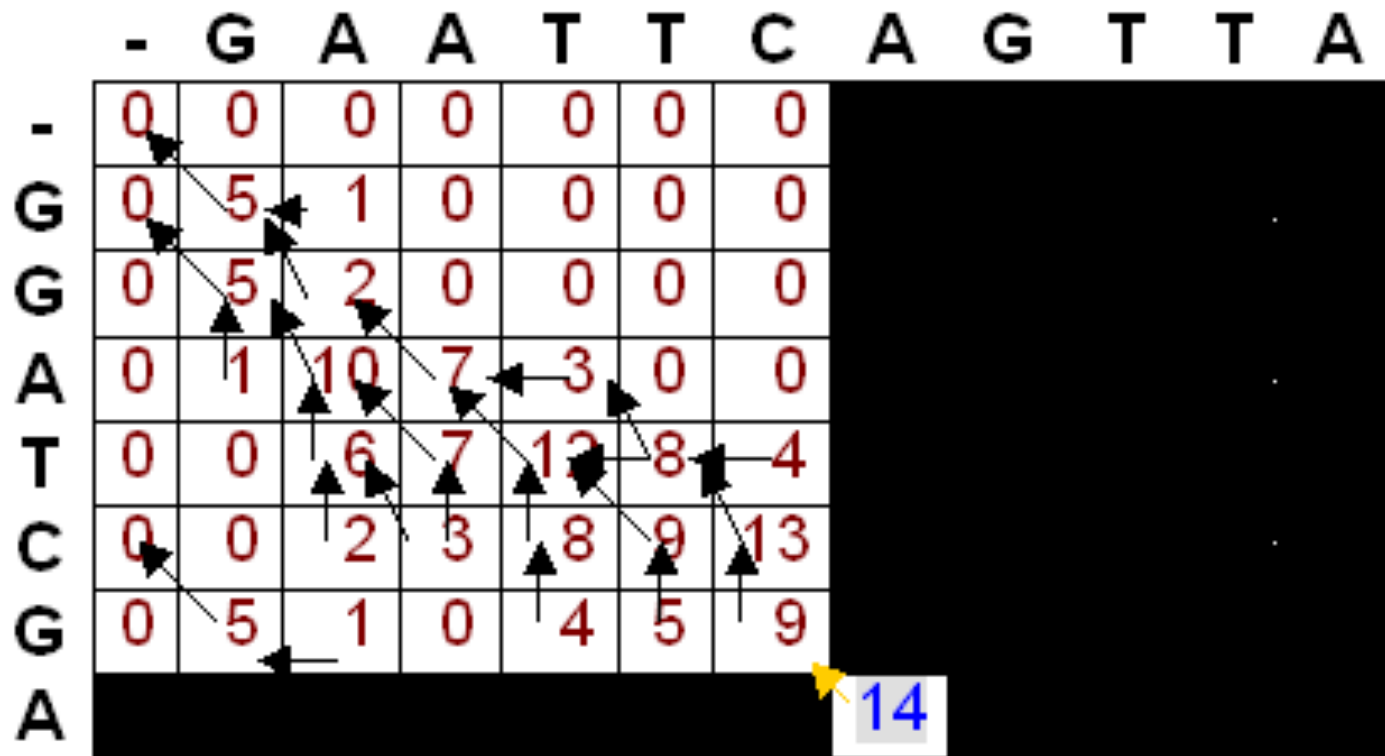
Trace Back (Local Alignment)

- maximum local alignment score for the two sequences is 14
- found by locating the highest values in the score matrix
- 14 is found in two separate cells, indicating multiple alignments producing the maximal alignment score

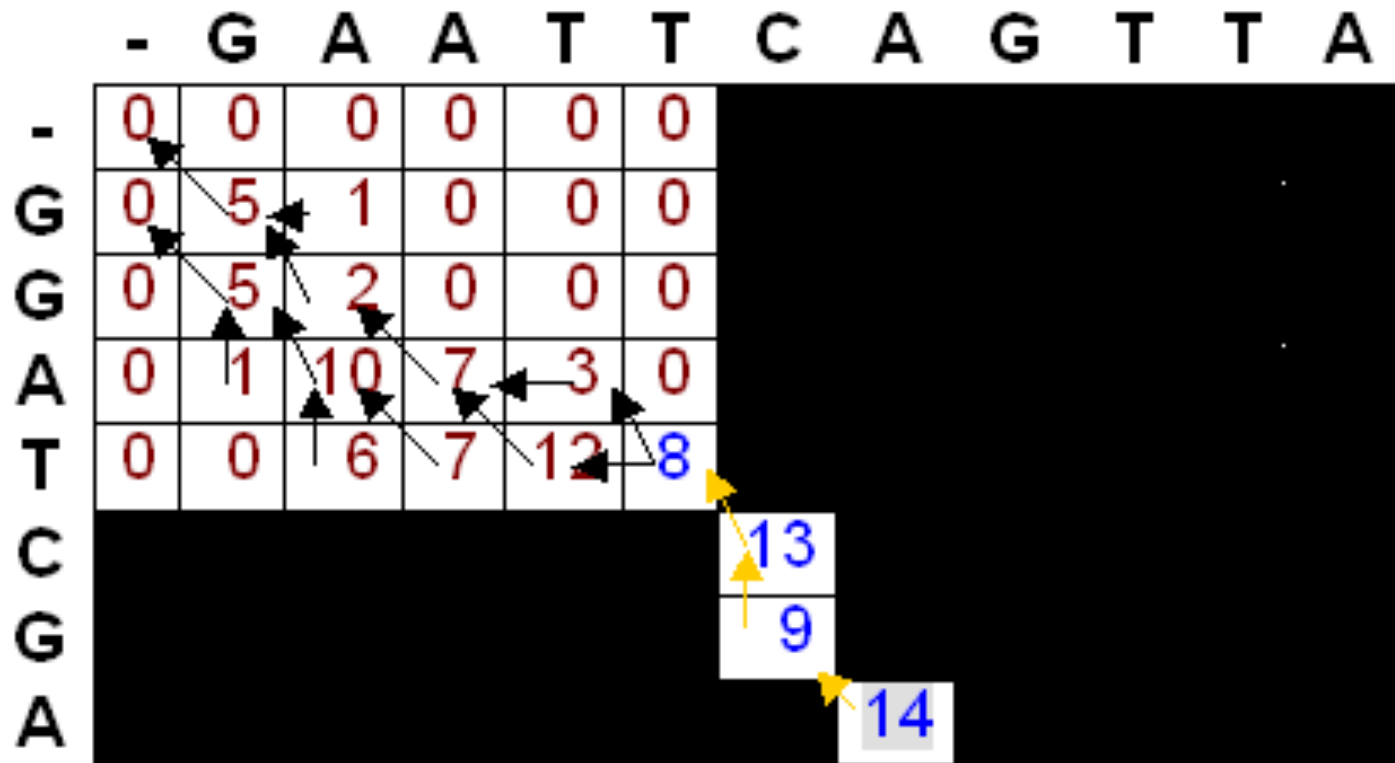
Trace Back (Local Alignment)

- Traceback begins in the position with the highest value.
- At each cell, we look to see where we move next according to the pointers
- When a cell is reached where there is not a pointer to a previous cell, we have reached the beginning of the alignment

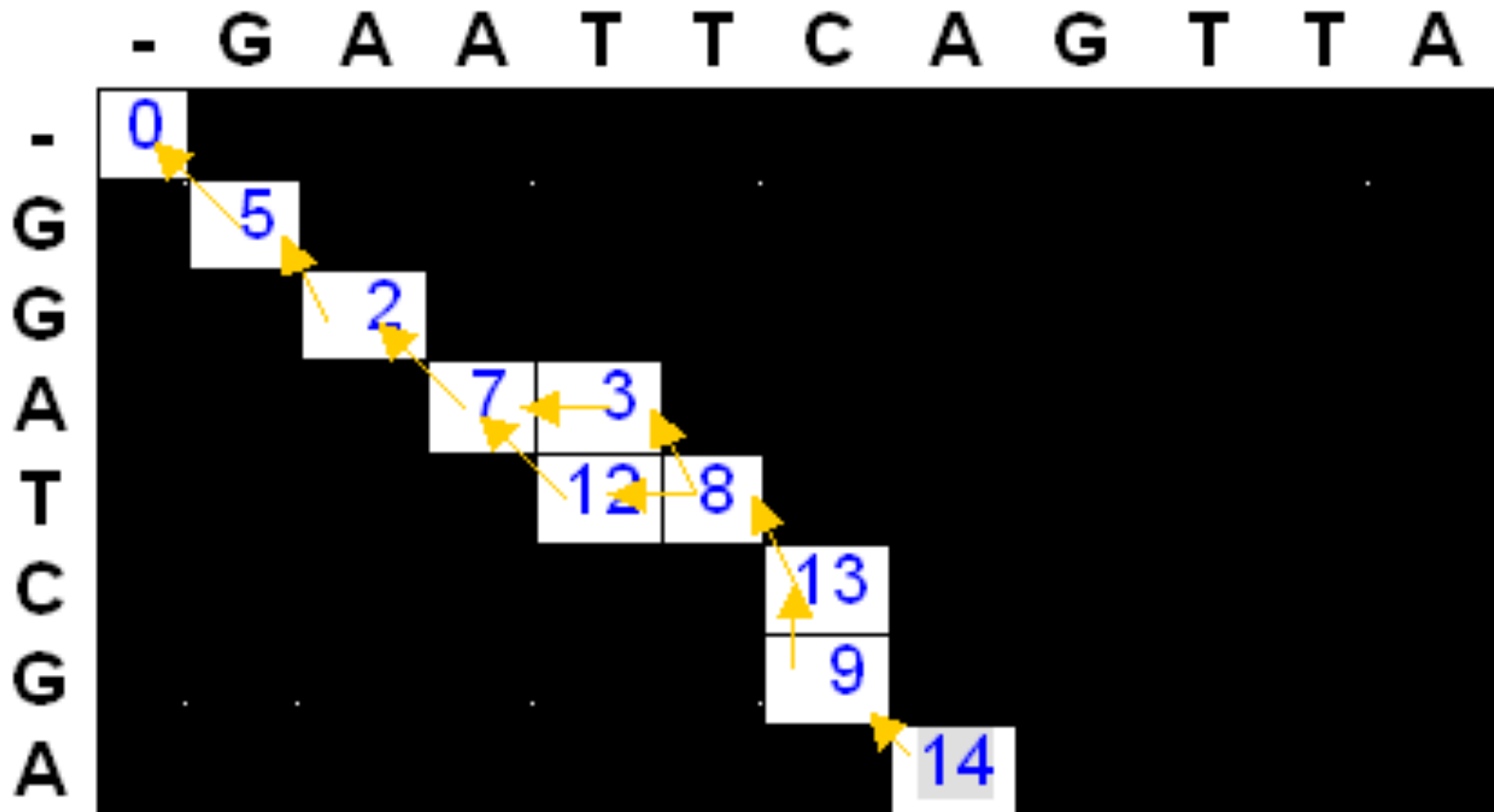
Trace Back (Local Alignment)



Trace Back (Local Alignment)



Trace Back (Local Alignment)



Maximum Local Alignment

G A A T T C - A

G A A T T C - A

| | | | |

| | | | |

G G A T - C G A

G G A - T C G A

+ - + + - + - +

+ - + - + + - +

5 3 5 5 4 5 4 5

5 3 5 4 5 5 4 5

Overlap Alignment

Overlap Alignment

Consider the following problem:

- ☼ Find the most significant **overlap** between two sequences?
- ☼ Possible overlap relations:

a.

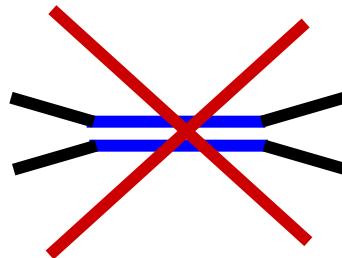


b.



Difference from **local** alignment:

Here we require alignment between the **endpoints** of the two sequences.



Overlap Alignment

Initialization: $S_{i,0} = 0$, $S_{0,j} = 0$

Recurrence: as in global alignment

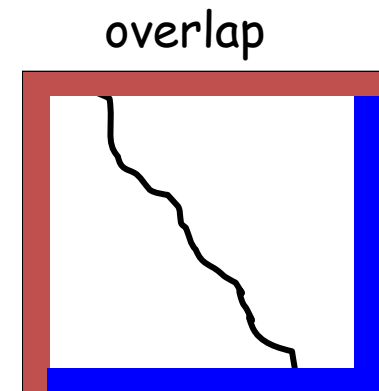
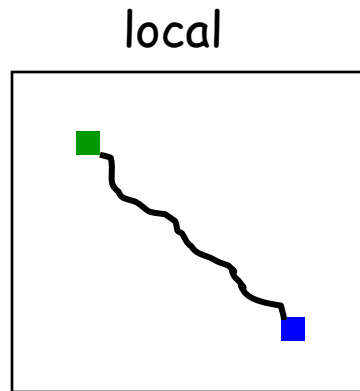
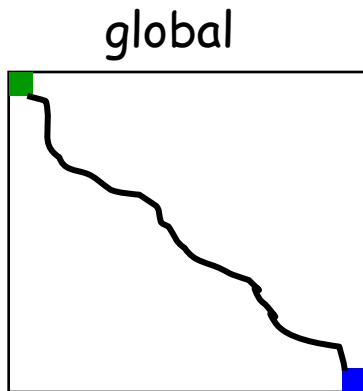
$S_{i,j} = \text{MAXIMUM [}$

$S_{i-1,j-1} + s(a_i, b_j)$ (match/mismatch in the diagonal),

$S_{i,j-1} + w$ (gap in sequence #1),

$S_{i-1,j} + w$ (gap in sequence #2)]

Score: maximum value at the bottom line and rightmost line



PAWHEAE
HEAGAWGHEE

Match: +4

Mismatch: -1

Gap penalty: -5

[illegible]

Overlap Alignment

PAWHEAE

HEAGAWGHEE

Scoring scheme :

Match: +4

Mismatch: -1

Gap penalty: -5

		H	E	A	G	A	W	G	H	E	E
	0	0	0	0	0	0	0	0	0	0	0
P	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
A	0	-1									
W	0	-1									
H	0	4									
E	0	-1									
A	0	-1									
E	0	-1									

Overlap Alignment

PAWHEAE
HEAGAWGHEE

Scoring scheme:

Match: +4

Mismatch: -1

Gap penalty: -5

		H	E	A	G	A	W	G	H	E	E
	0	0	0	0	0	0	0	0	0	0	0
P	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
A	0	-1	-2	3	-2	3	-2	-2	-2	-2	-2
W	0	-1	-2	-2	2	-2	7	2	-3	-3	-3
H	0	4	-1	-3	-3	1	2	6	6	1	-4
E	0	-1	8	3	-2	-3	0	1	5	10	5
A	0	-1	3	12	7	2	-2	-1	0	5	9
E	0	-1	3	7	11	6	1	-3	-2	4	9

Overlap Alignment

The best overlap is:

P A W H E A E - - - - -
- - - H E A G A W G H E E

Pay attention!

A different scoring scheme could yield a different result, such as:

Scoring scheme :

Match: +4

Mismatch: -1

Gap penalty: -2

- - - P A W - H E A E
H E A G A W G H E E -

Sequence Alignment Variants

- **Global** alignment (The Needleman-Wunsch Algorithm)
 - Initialization: $S_{i,0} = i * w$, $S_{0,j} = j * w$
 - Score: $S_{i,j} = \text{MAX} [S_{i-1,j-1} + s(a_i, b_j), S_{i,j-1} + w, S_{i-1,j} + w]$
- **Local** alignment (The Smith-Waterman Algorithm)
 - Initialization: $S_{i,0} = 0$, $S_{0,j} = 0$
 - Score: $S_{i,j} = \text{MAX} [S_{i-1,j-1} + s(a_i, b_j), S_{i,j-1} + w, S_{i-1,j} + w, 0]$
- **Overlap** alignment
 - Initialization: $S_{i,0} = 0$, $S_{0,j} = 0$
 - Score: $S_{i,j} = \text{MAX} [S_{i-1,j-1} + s(a_i, b_j), S_{i,j-1} + w, S_{i-1,j} + w]$

Linear vs. Affine Gaps

- The scoring matrices used to this point assume a linear gap penalty where each gap is given the same penalty score.
- However, over evolutionary time, it is more likely that a contiguous block of residues has become inserted/deleted in a certain region (for example, it is more likely to have 1 gap of length k than k gaps of length 1).
- Therefore, a better scoring scheme to use is an initial higher penalty for opening a gap, and a smaller penalty for extending the gap.

Linear vs. Affine Gaps

- Gaps have been modeled as linear
- More likely contiguous block of residues inserted or deleted
 - 1 gap of length k rather than k gaps of length 1
- Scoring scheme should penalize new gaps more

Affine Gap Penalty

- $w_x = g + r(x-1)$

- w_x : total gap penalty; g : gap open penalty; r : gap extend penalty; x : gap length
- gap penalty chosen relative to score matrix
 - Gaps not excluded
 - Gaps not over included
 - Typical Values: $g = -12$; $r = -4$

Affine Gap Penalty and Dynamic Programming

$$M_{i,j} = \max \{ D_{i-1,j-1} + \text{subst}(A_i, B_j) , \\ M_{i-1,j-1} + \text{subst}(A_i, B_j) , \\ I_{i-1,j-1} + \text{subst}(A_i, B_j) \}$$

$$D_{i,j} = \max \{ D_{i,j-1} - \text{extend}, M_{i,j-1} - \text{open} \}$$

$$I_{i,j} = \max \{ M_{i-1,j} - \text{open}, I_{i-1,j} - \text{extend} \}$$

where M is the match matrix, D is delete matrix,
and I is insert matrix

Drawbacks to DP Approaches

- Dynamic programming approaches are guaranteed to give the optimal alignment between two sequences given a scoring scheme.
- However, the two main drawbacks to DP approaches is that they are compute and memory intensive, in the cases discussed to this point taking at least $O(n^2)$ space, between $O(n^2)$ and $O(n^3)$ time.
- Linear space algorithms have been used in order to deal with one drawback to dynamic programming. The basic idea is to concentrate only on those areas of the matrix more likely to contain the maximum alignment.
- The most well-known of these linear space algorithms is the Myers-Miller algorithm.

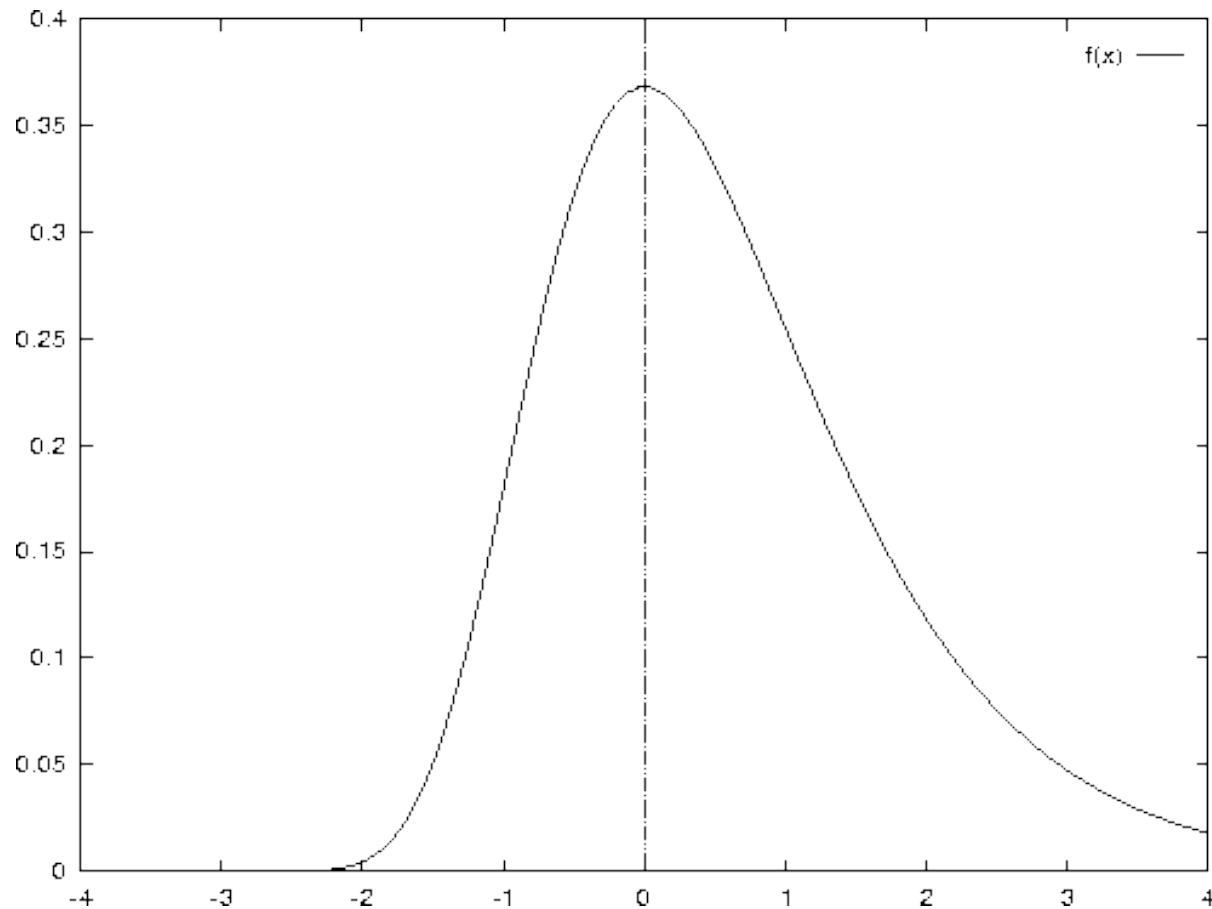
Alternative DP approaches

- Linear space algorithms Myers-Miller
- Bounded Dynamic Programming
- Ewan Birney's Dynamite Package
 - Automatic generation of DP code

Assessing Significance of Alignment

- When two sequences of length m and n are not obviously similar but show an alignment, it becomes necessary to assess the significance of the alignment.
- The alignment of scores of random sequences has been shown to follow a **Gumbel extreme value distribution**.

Gumbel Extreme Value Distribution



- <http://roso.epfl.ch/mbi/papers/discretechoice/node11.html>
- <http://mathworld.wolfram.com/GumbelDistribution.html>
- http://en.wikipedia.org/wiki/Generalized_extreme_value_distribution

Probability of Alignment Score

- Using a Gumbel extreme value distribution, the expected number of alignments (E-value) with a score at least S is:

$$E = Kmn e^{-\lambda S}$$

- m, n : Lengths of sequences
- K, λ : statistical parameters dependent upon scoring system and background residue frequencies

- Recall that the log-odds scoring schemes examined to this point normally use a $S = 10 \cdot \log_{10} x$ scoring system.
- We can normalize the raw scores obtained using these non-gapped scoring systems to obtain the amount of bits of information contained in a score (or the amount of **nats** of information contained within a score).

Converting to Bit Scores

A raw score can be normalized to a bit score using the form

$$S' = \frac{\lambda S - \ln K}{\ln 2}$$

- The E-value corresponding to a given bit score can then be calculated

$$E = mn 2^{-S'}$$

- Converting to **nats** is similar.
- We just substitute **e** for 2 in the above equations.
- Converting scores to either bits or **nats** gives a standardized unit by which the scores can be compared.

P-Value

- P values can be calculated as the probability of obtaining a given score at random.
- P-values can be estimated as:

$$P = 1 - e^{-E}$$

A quick determination of significance

- If a scoring matrix has been scaled to bit scores, then it can quickly be determined whether or not an alignment is significant.
- For a typical amino acid scoring matrix, $K = 0.1$ and λ depends on the values of the scoring matrix.
- If a PAM or BLOSUM matrix is used, then λ is precomputed.
- For instance, if the log odds matrix is in units of bits, then $\lambda = \log_e 2$, and the significance cutoff can be calculated as $\log_2(mn)$.

Significance of Ungapped Alignments

- PAM matrices are $10 * \log_{10}x$
- Converting to \log_2x gives **bits** of information
- Converting to $\log_e x$ gives **nats** of information

Quick Calculation:

- If bit scoring system is used, significance cutoff is:

$$\log_2(mn)$$

Example

- Suppose we have two sequences, each approximately 250 amino acids long that are aligned using a Smith-Waterman approach.
- Significance cutoff is:

$$\log_2(250 * 250) = 16 \text{ bits}$$

Example

- Using PAM250, the following alignment is found:

- F W L E V E G N S M T A P T G
- F W L D V Q G D S M T A P A G

PAM250 matrix

	C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W	
C	12																				C
S	0	2																			S
T	-2	1	3																		T
P	-3	1	0	6																	P
A	-2	1	1	1	2																A
G	-3	1	0	-1	1	5															G
N	-4	1	0	-1	0	0	2														N
D	-5	0	0	-1	0	1	2	4													D
E	-5	0	0	-1	0	0	1	3	4												E
Q	-5	-1	-1	0	0	-1	1	2	2	4											Q
H	-3	-1	-1	0	-1	-2	2	1	1	3	6										H
R	-4	0	-1	0	-2	-3	0	-1	-1	1	2	6									R
K	-5	0	0	-1	-1	-2	1	0	0	1	0	3	5								K
M	-5	-2	-1	-2	-1	-3	-2	-3	-2	-1	-2	0	0	6							M
I	-2	-1	0	-2	-1	-3	-2	-2	-2	-2	-2	-2	-2	2	5						I
L	-6	-3	-2	-3	-2	-4	-3	-4	-3	-2	-2	-3	-3	4	2	6					L
V	-2	-1	0	-1	0	-1	-2	-2	-2	-2	-2	-2	-2	2	4	2	4				V
F	-4	-3	-3	-5	-4	-5	-4	-6	-5	-5	-2	-4	-5	0	1	2	-1	9			F
Y	0	-3	-3	-5	-3	-5	-2	-4	-4	-4	0	-4	-4	-2	-1	-1	-2	7	10		Y
W	-8	-2	-5	-6	-6	-7	-4	-7	-7	-5	-3	2	-3	-4	-5	-2	-6	0	0	17	W
	C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W	

Example

- Using PAM250, the score is calculated:

- F W L E V E G N S M T A P T G

- F W L D V Q G D S M T A P A G

- $S = 9 + 17 + 6 + 3 + 4 + 2 + 5 + 2 + 2 + 6 + 3 + 2 + 6 + 1 + 5 = 73$

Significance Example

- S is in $10 * \log_{10}x$, so this should be converted to a bit score:
- $S = 10 \log_{10}x$
- $S/10 = \log_{10}x$
- $S/10 = \log_{10}x * (\log_2 10 / \log_2 10)$
- $S/10 * \log_2 10 = \log_{10}x / \log_2 10$
- $S/10 * \log_2 10 = \log_2 x$
- $1/3 S \sim \log_2 x$
- $S' \sim 1/3 S$

Significance Example

- $S' = 1/3S = 1/3 * 73 = 24.333$ bits
- The significance cutoff is:
 $\log_2(mn) = \log_2(250 * 250) = 16$ bits
- Since the alignment score is above the significance cutoff, this is a significant local alignment.

Estimation of E and P

- When a PAM250 scoring matrix is being used, K is estimated to be 0.09, while lambda is estimated to be 0.229.
- For PAM250, $K = 0.09$; $\lambda = 0.229$
- We can convert the score to a bit score as follows:
 - $S' = \lambda S - \ln Kmn$
 - $S' = 0.229 * 73 - \ln 0.09 * 250 * 250$
 - $S' = 16.72 - 8.63 = 8.09$ bits
 - $P(S' \geq x) = 1 - e^{(-e^{-x})}$
 - $P(S' \geq 8.09) = 1 - e^{(-e^{-8.09})} = 3.1 * 10^{-4}$
- Therefore, we see that the probability of observing an alignment with a bit score greater than 8.09 is about 3 in 1000.

Significance of Gapped Alignments

- Gapped alignments make use of the same statistics as ungapped alignments in determining the statistical significance.
- However, in gapped alignments, the values for λ and K cannot be easily estimated.
- Empirical estimations and gap scores have been determined by looking at the alignments of randomized sequences.

Biological Network Analysis and Gene-Gene Interaction Networks

Content

- Rationale of “**biological network analysis**”
- Inferring gene-gene interaction networks
- Co-regulation (co-expression) networks
- Weighted correlation network analysis of genes (WGCNA)
- Bayesian networks
- Real life examples of gene networks usage

Hypothesis



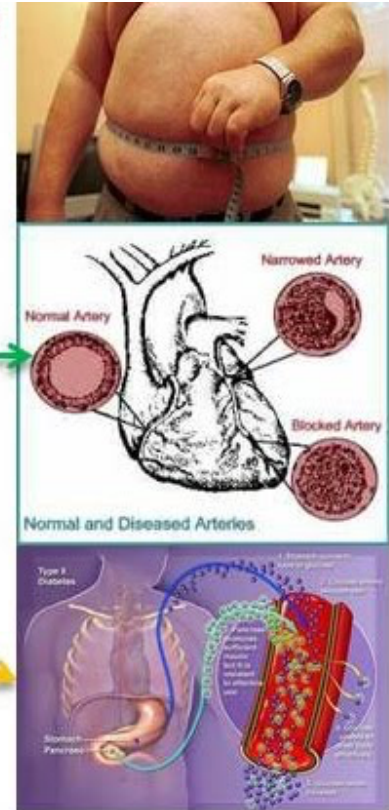
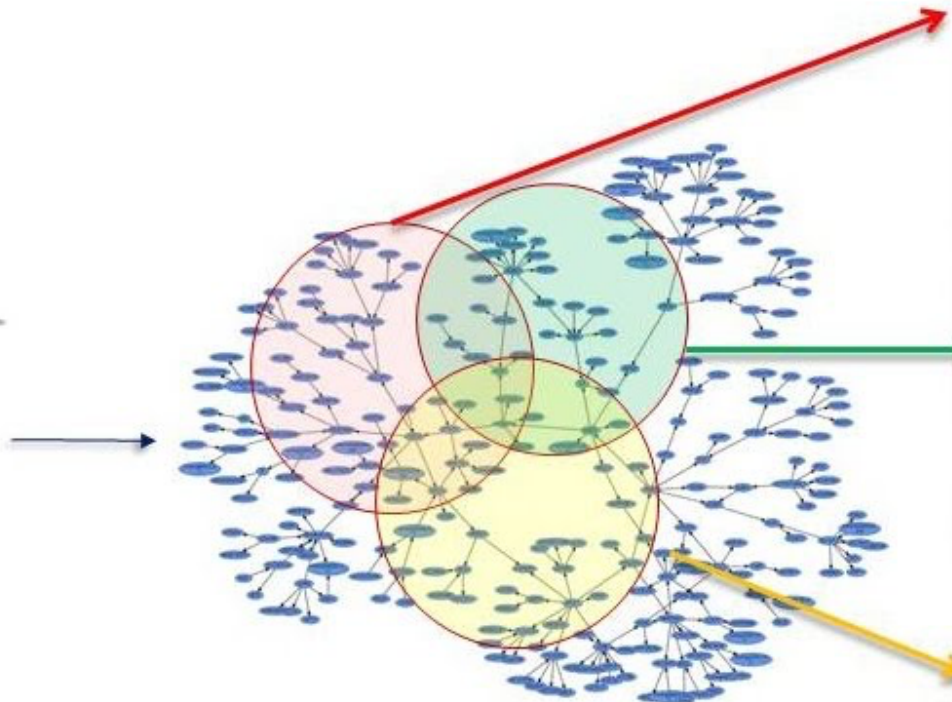
DNA variations/
environmental



Drive changes in
network states

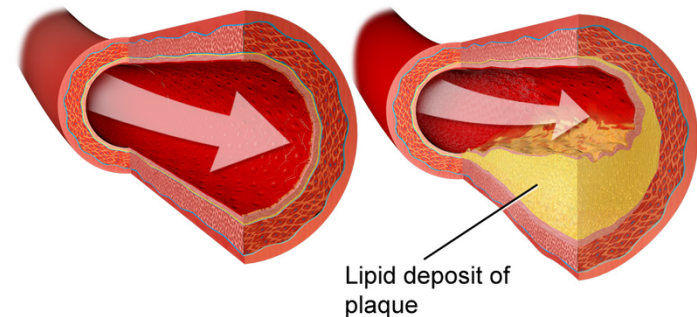
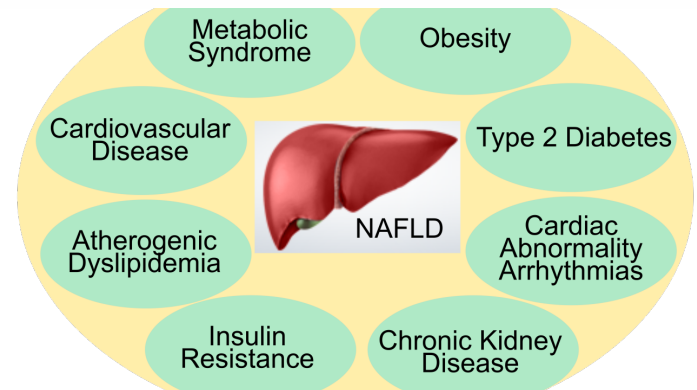
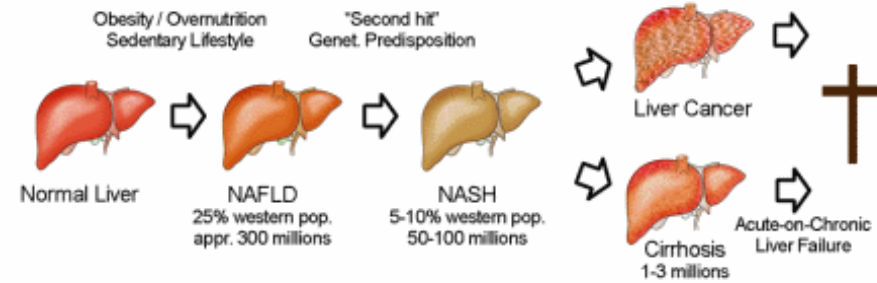
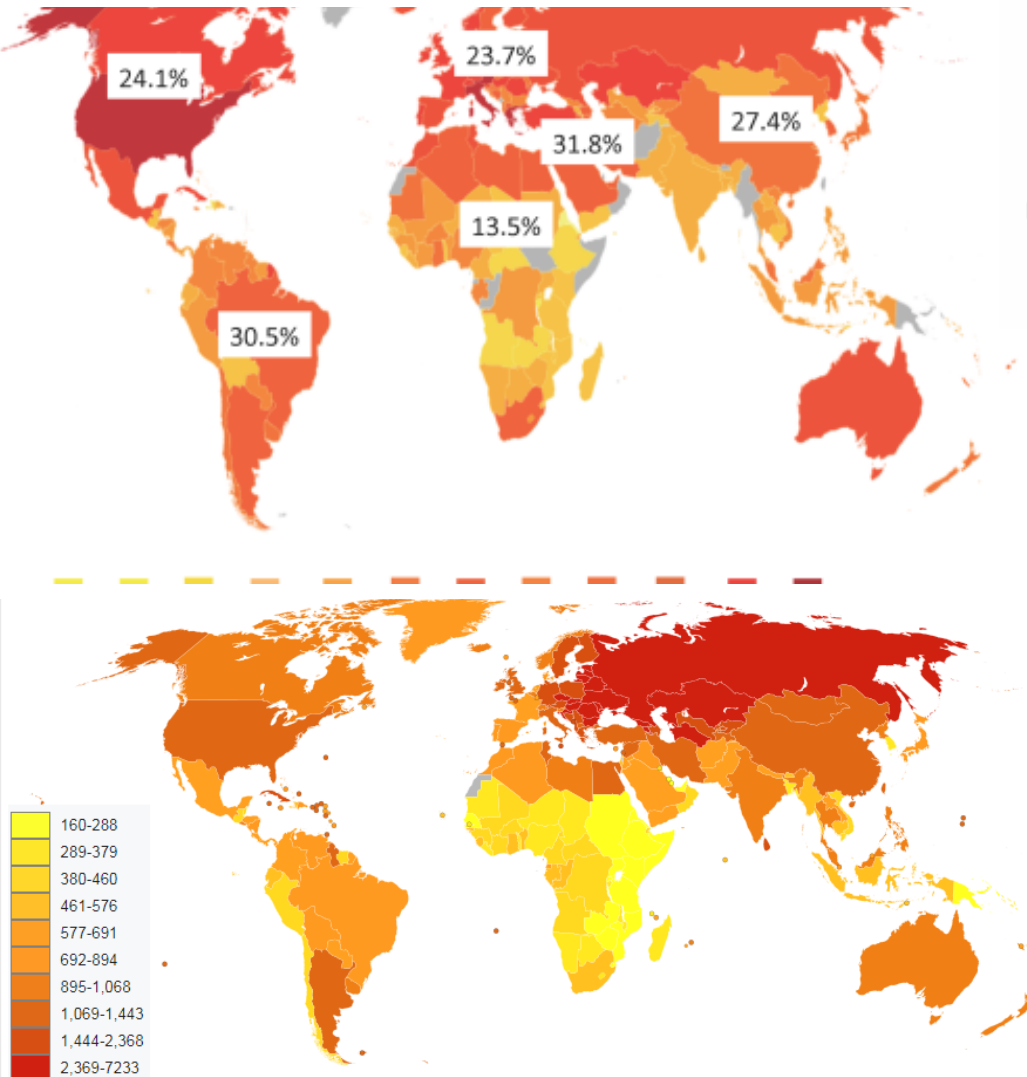


Variations in disease
phenotypes and drug



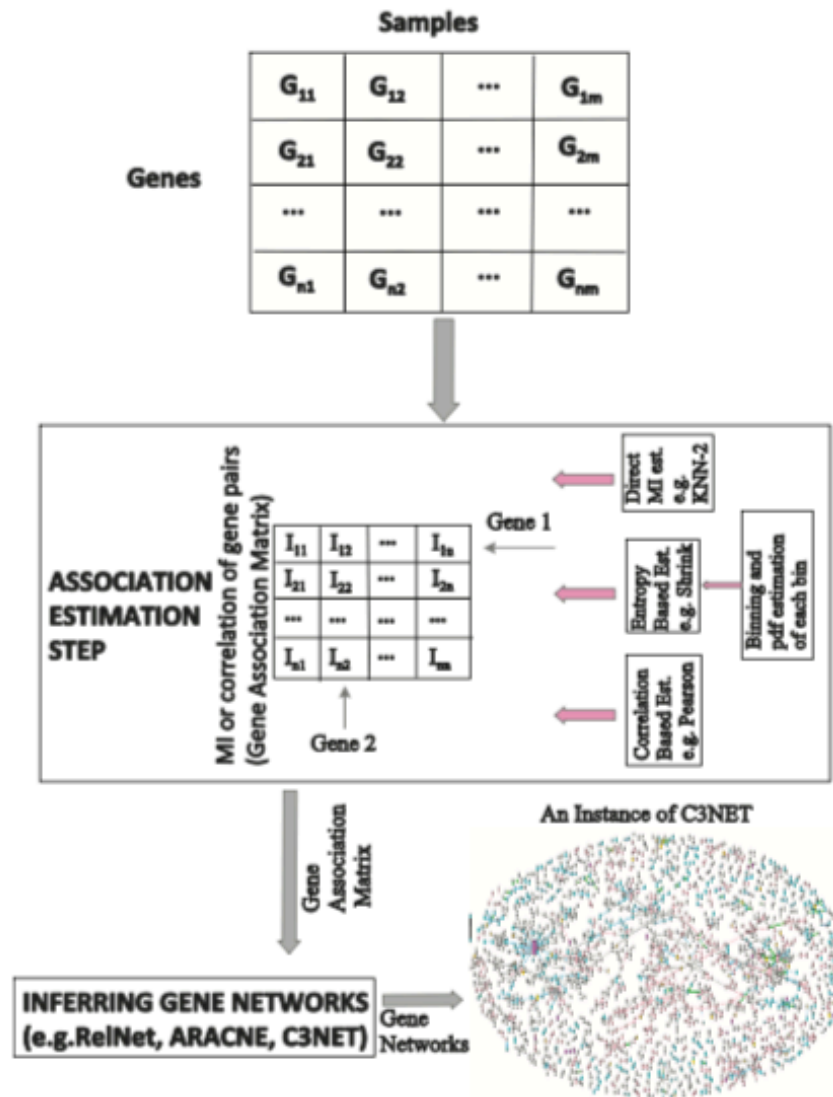
What can gene networks provide us?

To study common human diseases such as NAFLD and CAD..



Deaths from CAD in 2012 per million persons.
Statistics from WHO, grouped by deciles.

Dataset obtained from microarray data analysis



Weighted correlation network analysis (WGCNA)*

***Citation for WGCNA summary:**

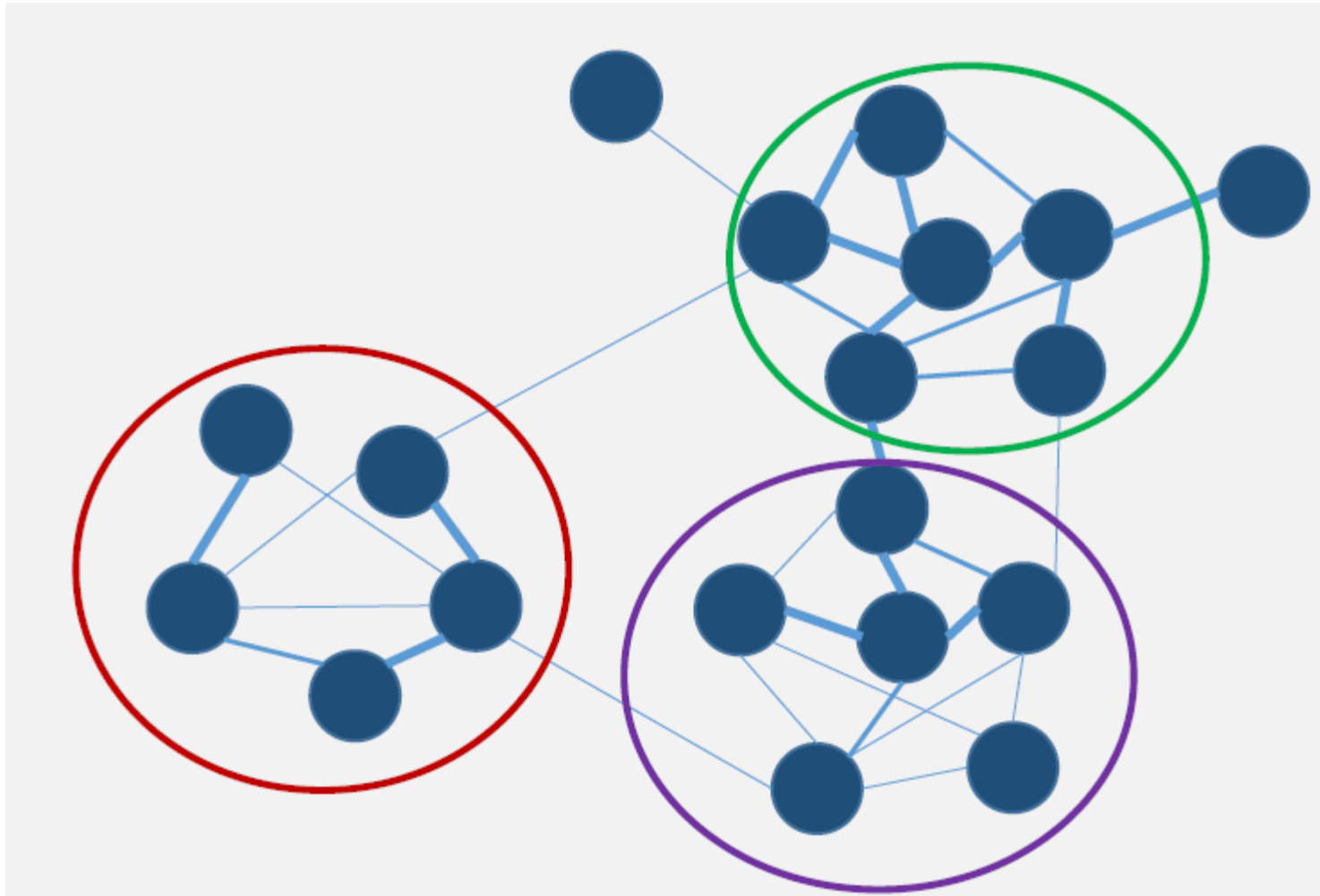
SIB course, Nov. 15-17, 2016, “Introduction to Biological Network Analysis” by Leonore Wigger and with Frédéric Burdet and Mark Ibberson

Overview of WGCNA

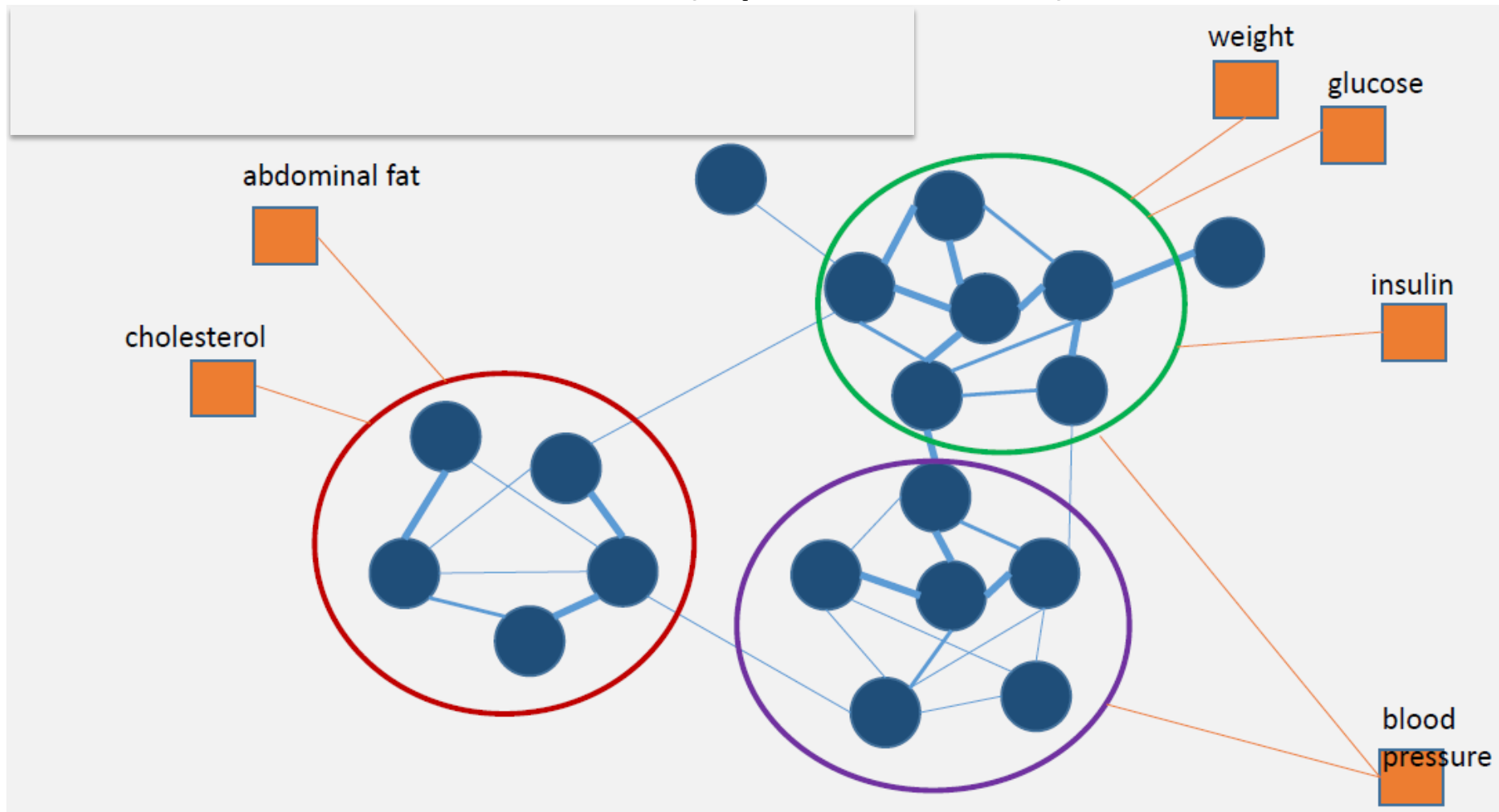
- **Theory 1:** Weighted correlation network, split into modules
- **Theory 2:** Identify modules and genes of interest
- **Input data:**
 - Gene expression data (microarray or RNA-Seq)
Recommendation: at least 20 individuals
 - Clinical/phenotypical traits from the same individuals (optional) **e.g. weight, insulin level, glucose level**

Aims of WGCNA:

Inferring a gene-gene similarity network

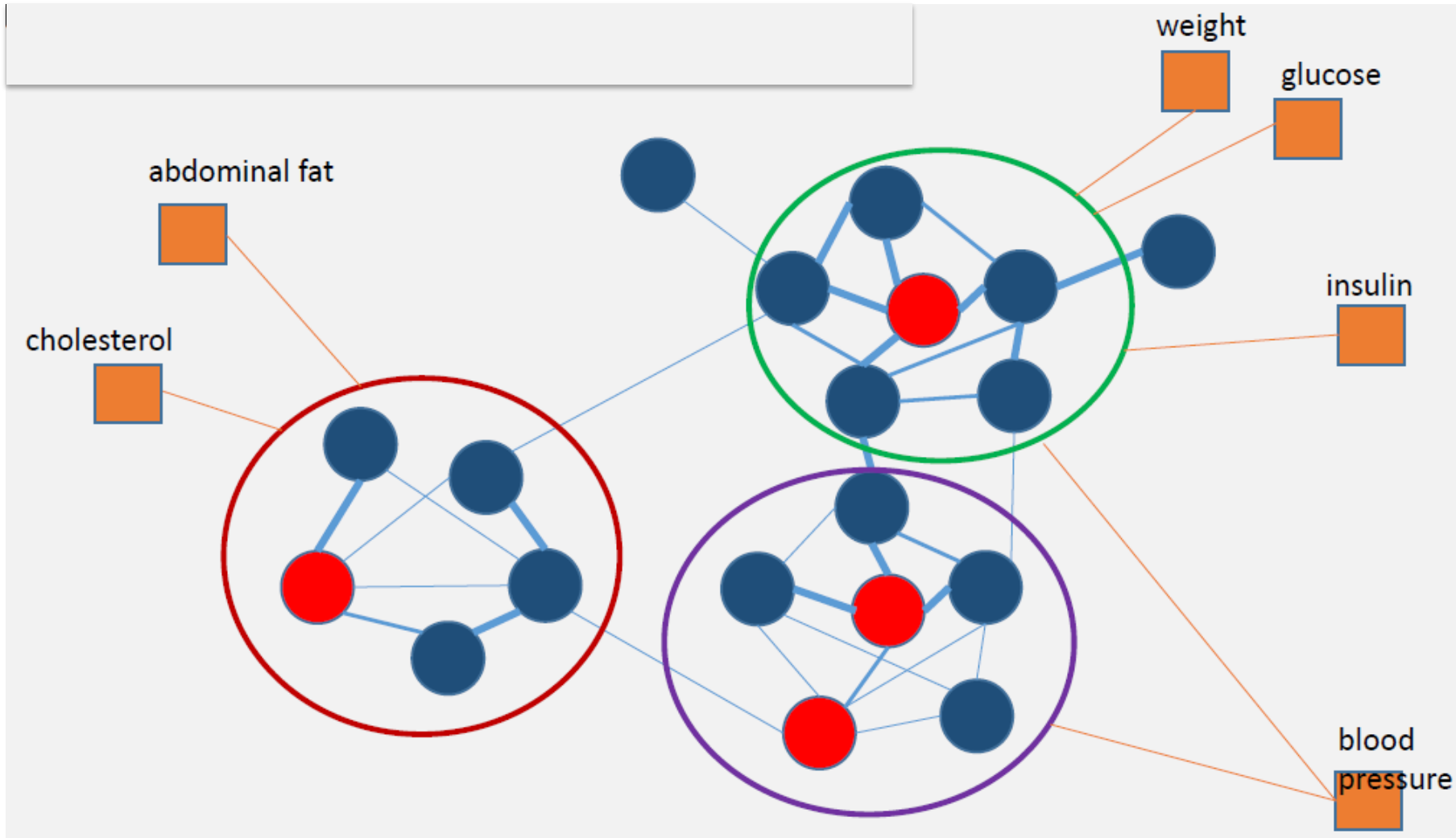


Aims: Correlate phenotypic traits to gene modules* (optional aim)



* **“Modules” found in WGCNA:** Groups or clusters of co-expressed genes with similar expression profiles over a large group of individuals

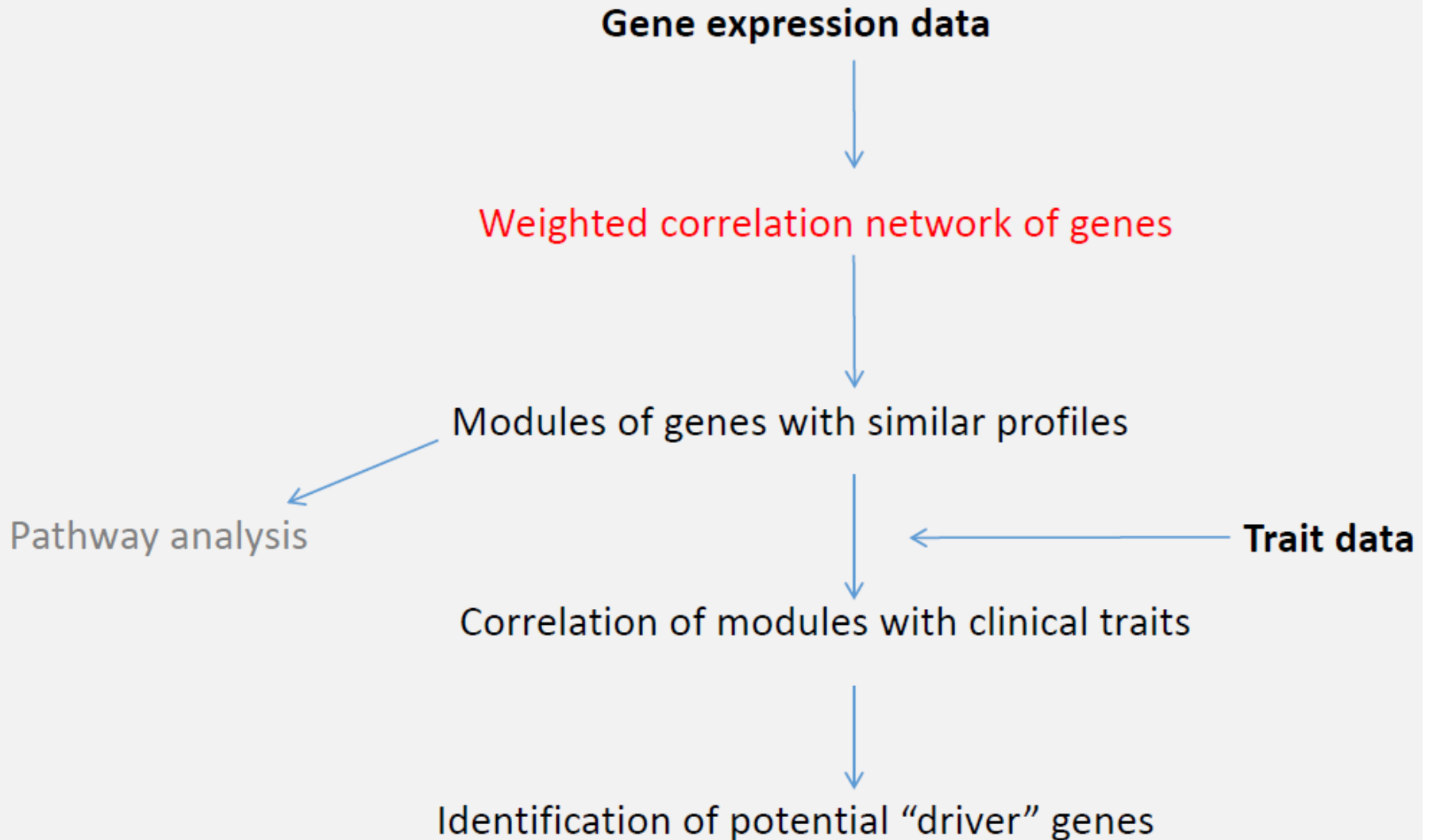
Aims: Identify “key driver” genes in modules



Rationale

- Genes with **similar expression patterns are of interest because they may be**
 - tightly co-regulated
 - functionally related
 - members of the same pathway
- WGCNA encourages hypotheses about genes based on their close network neighbors.

Workflow



But first: **preprocess** the gene expression data

- **Remove outlier samples e.g.** by creating dendrogram for samples (take transpose of the expression matrix for this) and identifying the outliers
- **Remove the lowly expressed genes** by eliminating the genes that do not have expression levels > 0 at least for 50% of all samples
- **Normalize gene expression by $\log_2(\text{expr}+1)$.** We need to transform expression values with logarithm base 2, since the correlation measures (e.g. Pearson coefficient) assume that the expression values of each gene are normally distributed (approximately) across the samples.
- **Extract expression table with the most variable probes/genes:** e.g. find the top $\sim 15,000$ highly variable genes based on the “**coefficient of variation**” (cv) measure.

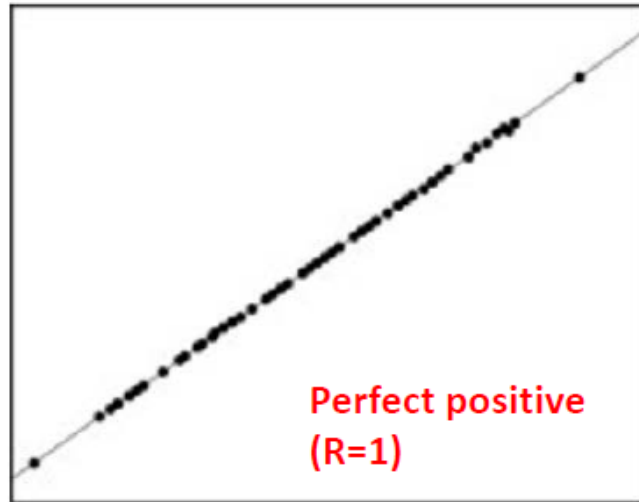
But first: **preprocess** the gene expression data

- Remove the lowly expressed genes:
 - `Min_Exp_level=0`
 - `Gene_present<-matrix()`
 - `for (i in 1:nrow(Expr_Mat))`
 `Gene_present[i]`
 `= (sum(Expr_Mat[i,] > Min_Exp_level) / ncol(Expr_Mat)) >= 0.5`
 - `Expr_Mat <- Expr_Mat[Gene_present,]`
- Transform expression values with logarithm base 2 (normalization):
 - `Expr_Mat = log2(Expr_Mat + 1)`
- Extract expression table with the most variable probes/genes:
 - `cv=NULL`
 - `a=Expr_Mat`
 - `for (m in 1:nrow(a)){`
 `cv_individual = cv(a[m,]) ; cv = rbind(cv,cv_individual)`
 `}`
 - `a_sort = with(data.frame(a), data.frame(a)[order(-cv),])`
 - # Keep the top e.g. 90 percentile, which provides us to have ~15,000 genes:
 - `perc <- 0.90`
 - `a_keep <- a_sort[1:(perc*nrow(a_sort)),]`
 - `Expr_Mat <- a_keep`

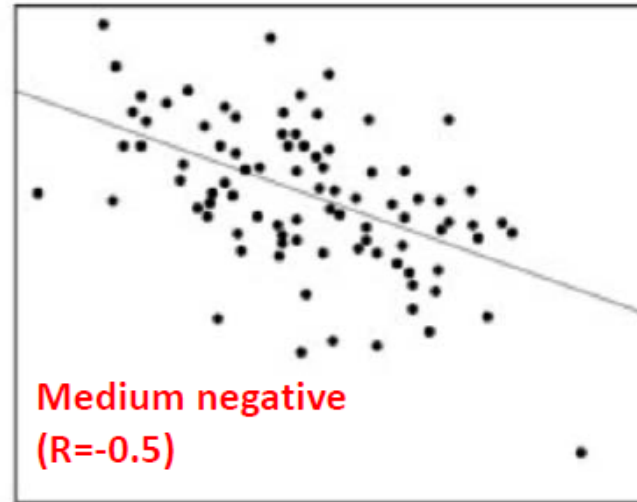
Construct weighted correlation network

- **Correlation:** A statistical measure for the extent to which two variables fluctuate together.
- **Positive correlation:** variables increase/decrease together
- **Negative correlation:** variables increase/decrease in opposing direction
- **Caution:** There is no causality here!!

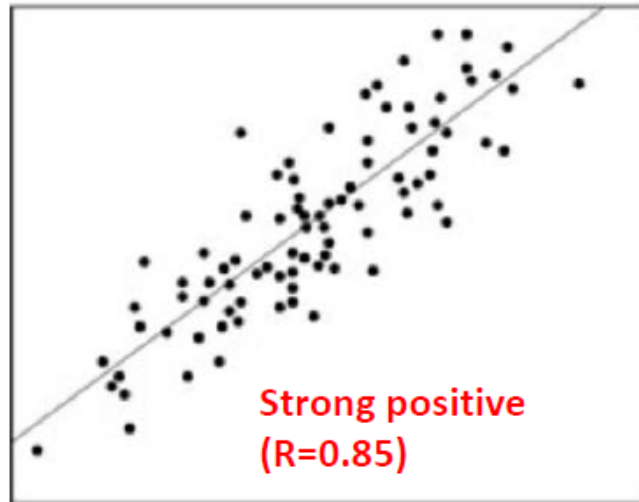
Correlation examples (Pearson correlation, R^2)



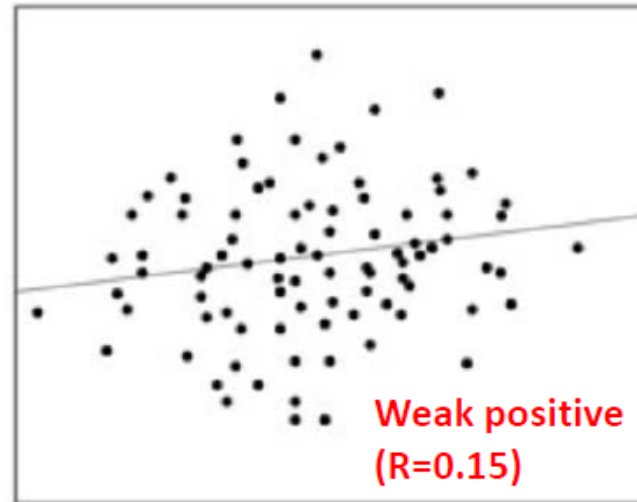
a



b



c



d

Scatterplots with correlations of a) $+1.00$; b) -0.50 ; c) $+0.85$; and d) $+0.15$.

Correlation measures implemented in WGCNA

- Pearson
- Spearman
- Kendall's tau
- Biweight midcorrelation (bicor)

Choosing a correlation method

$$a_{i,j} = |\text{cor}(i, j)|^\beta$$

- **Fastest, but sensitive to outliers:** Pearson correlation, `cor(x)`, “standard” measure of linear correlation
- **Less sensitive to outliers but much slower:** Biweight mid-correlation, `bicor(x)`, robust, recommended by the authors for most situations [needs modification for correlations involving binary/categorical variables]
- Spearman correlation, `cor(x, method=“spearman”)`, rank-based, works even if relationship is not linear (but the relationship expected to be monotonous), less sensitive to gene expression differences [can be used as-is for correlations involving binary/categorical variables]
- **Default correlation method in WGCNA:** `cor(Pearson)`.
- **Caveat:** use it only if there are no outliers, or for exercises/tutorials.

Adjacency matrix calculation

- Compute a correlation raised to a power between every pair of genes (i,j) : $a_{i,j} = |cor(i,j)|^\beta$
- Effect of raising correlation to a power:
 - Amplifies disparity between strong and weak correlations
 - Example: Power term $\beta = 4$

Correlations		Adjacencies	
$cor(i,j) = 0.8$	\rightarrow	$ 0.8 ^4 = 0.4096$	Strong corr.
$cor(k,l) = 0.2$	\rightarrow	$ 0.2 ^4 = 0.0016$	Weak corr.
0.8/0.2: 4-fold difference	\rightarrow	0.4096/0.0016: 256-fold difference	

Adjacency matrix calculation – cont'd

Adjacencies

Compute a correlation raised to a power between every pair of genes (i, j)

$$a_{i,j} = |\text{cor}(i, j)|^\beta$$

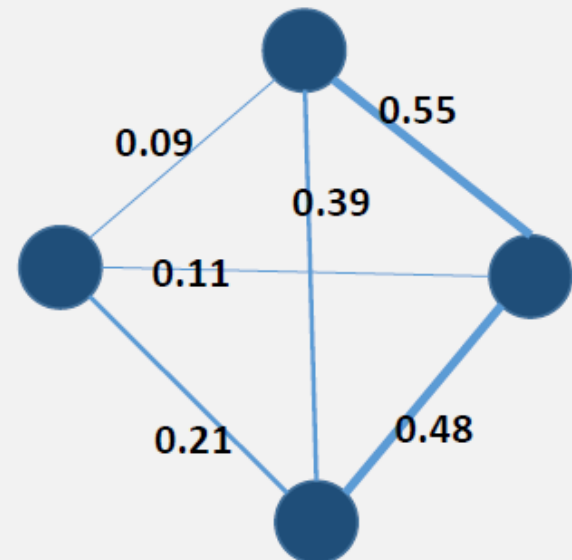
Network

Construct a fully connected network;
Genes as nodes, $a_{i,j}$ as edge weights.

high correlation – strong connection
low correlation – weak connection

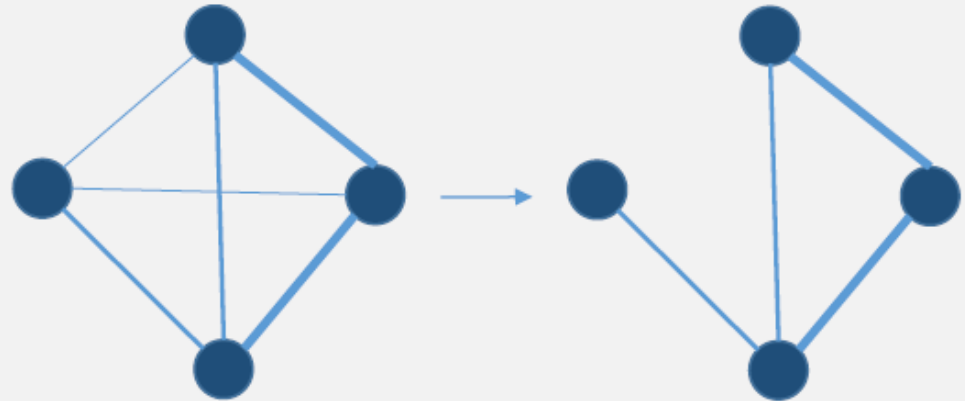
Adjacency matrix of 4 genes

$a_{i,j}$	gene1	gene2	gene3	gene4
gene1	1	0.55	0.39	0.09
gene2	0.55	1	0.48	0.11
gene3	0.39	0.48	1	0.21
gene4	0.09	0.11	0.21	1

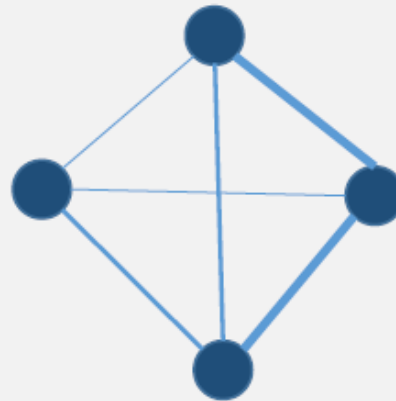


Adjacency matrix calculation – cont'd

For visualizations, set a threshold on edge weight and **remove the weakest links**.

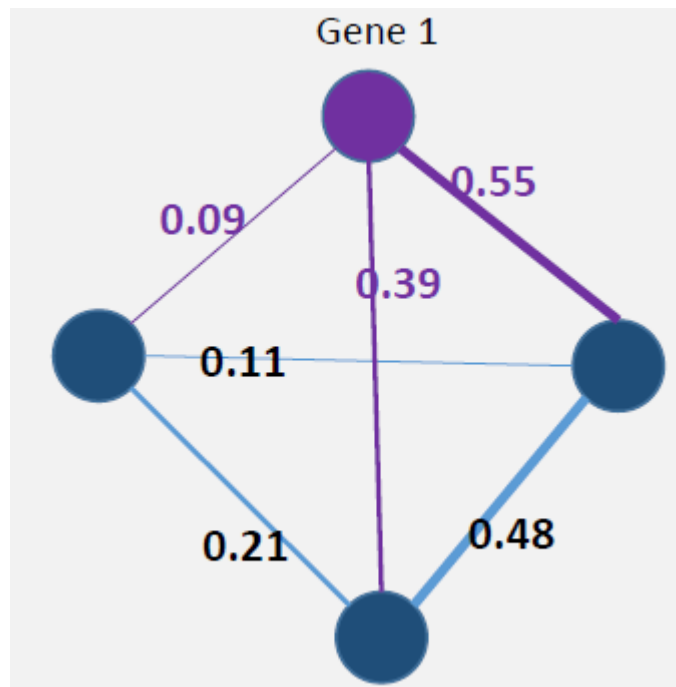


In most computations, work with all edges of the **fully connected network**.



Connectivity (degree) in a weighted network

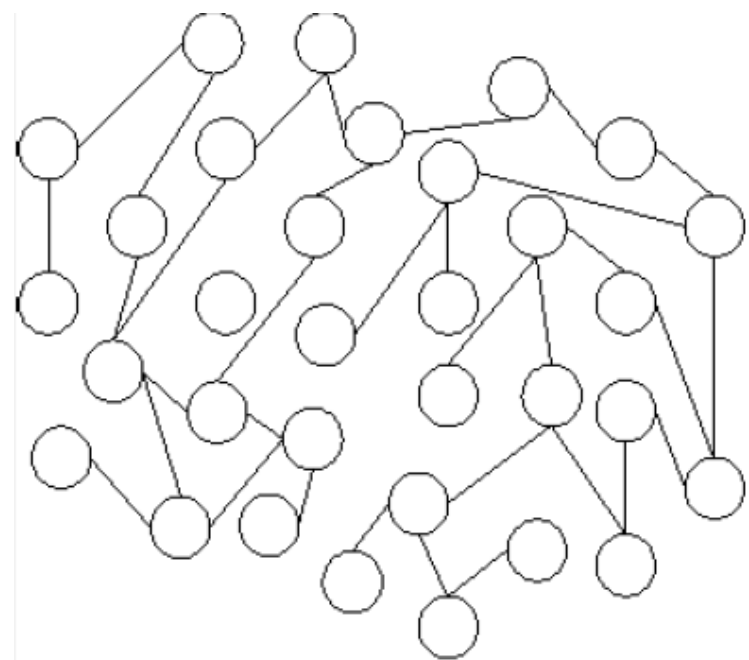
- **Connectivity of a gene:** Sum of the weights of all edges connecting to this gene
- **Example: Connectivity of gene 1 is:**
 $0.55 + 0.39 + 0.09 = 1.03$



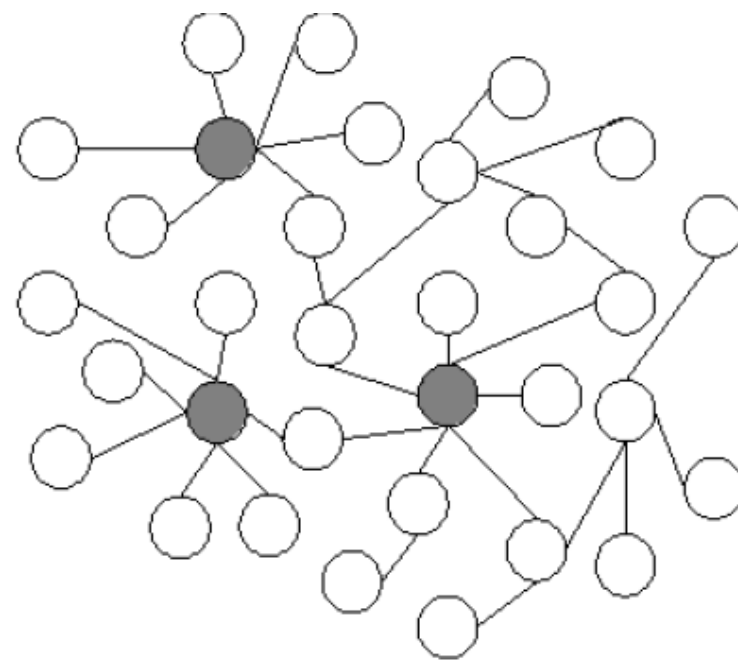
Picking a power term

$$a_{i,j} = |\text{cor}(i, j)|^\beta$$

- **Selection criterion:** Pick lowest possible β that leads to an approximately *scale-free network topology*:
 - few nodes with many connections ("hubs")
 - many nodes with few connections
- Degree distribution follows a power law:
 - the probability for a node of having k connections is $k^{-\gamma}$



(a) Random network



(b) Scale-free network

Source: Carlos Castillo: Effective Web Crawling, PhD Thesis, University of Chile, 2004

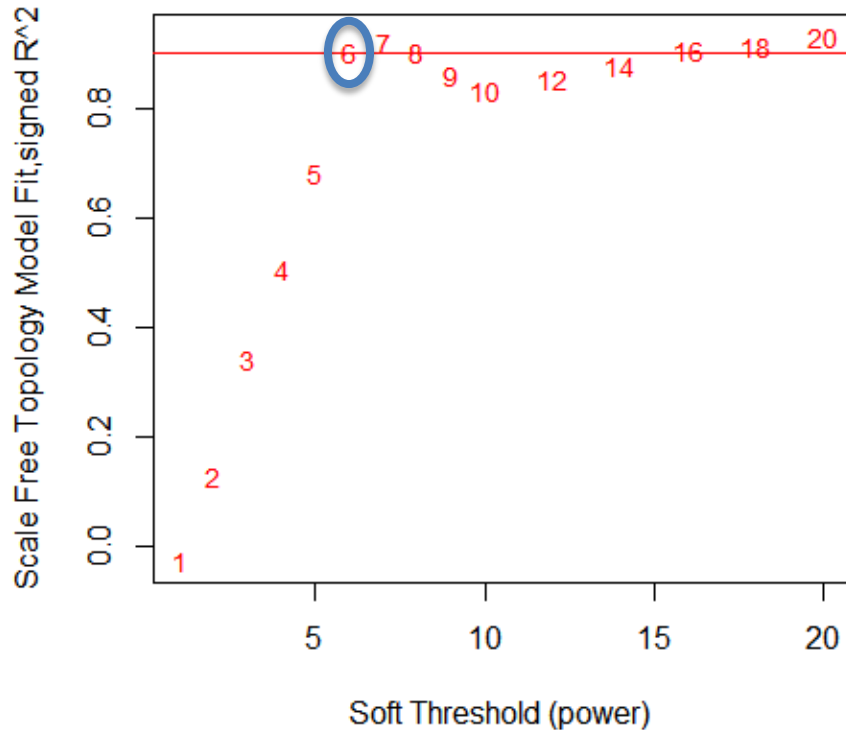
Why scale-free network topology?

- Barabási et al.* found many types of network in many domains to be approximately scale-free, including metabolic and protein interaction
- So, aim in WGCNA is: Building a biologically “realistic” network.

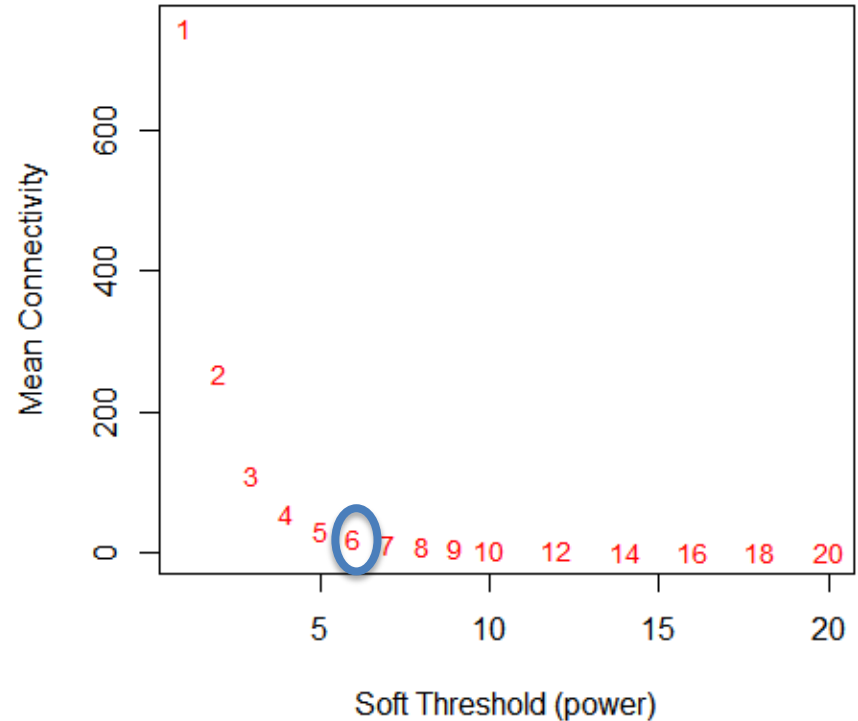
* Barabási, Albert-László; Bonabeau, Eric (May 2003). "Scale-Free Networks"(PDF). Scientific American. **288(5): 50–9.**

Pick a power term: Visual Aid in WGCNA

Scale independence

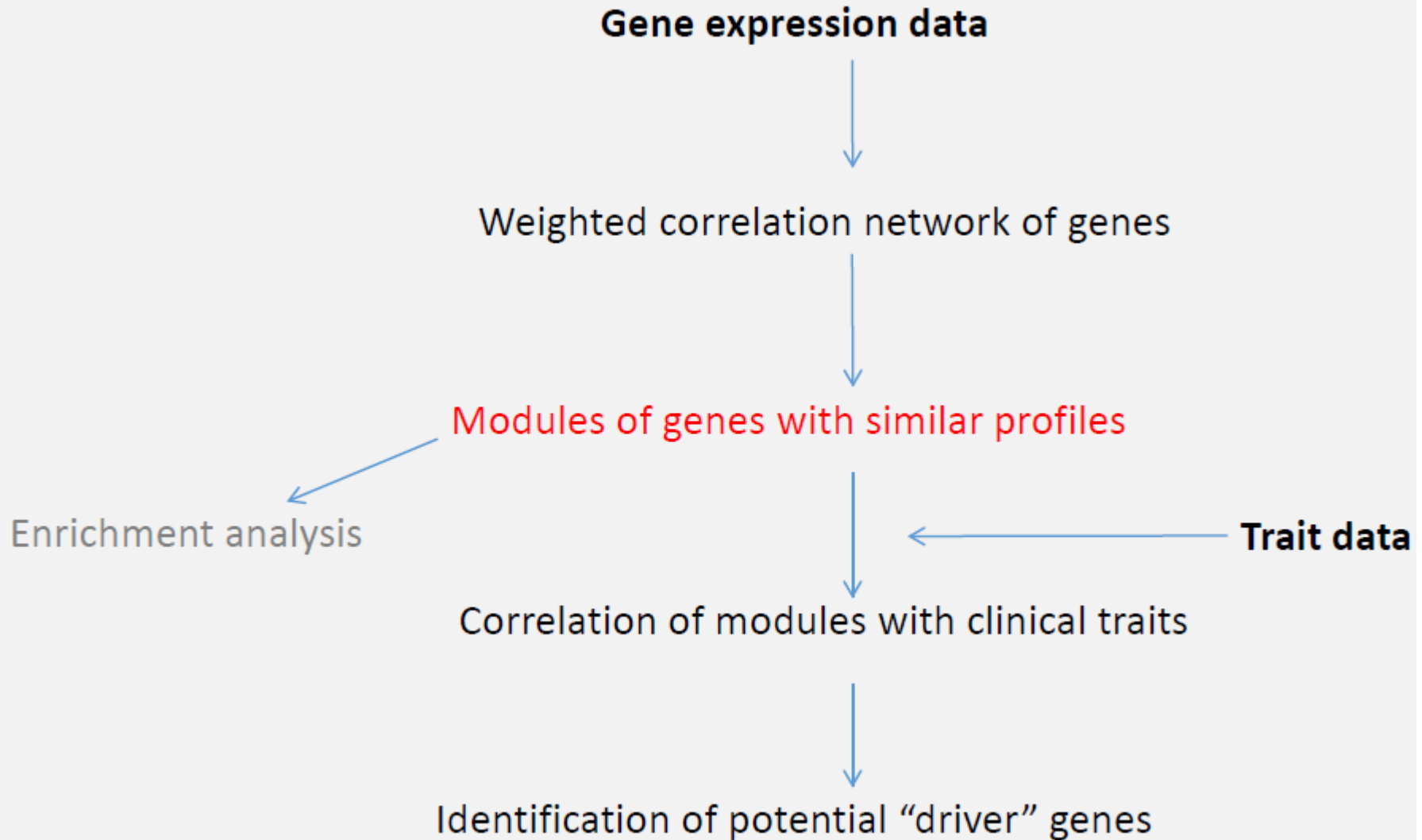


Mean connectivity



- **Left plot:** Choose power 6. Lowest possible power term where topology approximately fits a scale free network (on or above red horizontal line).
- **Right plot:** mean connectivity drops as power goes up. Must **not** drop too low

Next Step: Detect **modules** of co-expressed genes



4 steps to get from network to modules

1. Compute dissimilarity between genes:
“topological overlap measure dissimilarity”
2. Perform hierarchical clustering of genes: obtain tree structure
3. Divide clustered genes into modules: cut tree branches
4. **Optional:** Merge very similar modules: use module “eigengenes”

Step 1: Compute dissimilarity between genes

- **Why we use Topological Overlap Measure (TOM)?**
 - TOM is a pairwise similarity measure between network nodes (genes)
 - $TOM(i,j)$ is **high if genes i,j have many shared neighbors** because overlap of their network neighbors is large
- **So, a high $TOM(i,j)$ implies that genes have similar expression patterns**

- **How to calculate TOM similarity between two nodes:**

- 1. Count number of shared neighbors: “agreement” of the set of neighboring nodes

- 2. Normalize to [0,1]

$TOM(i,j) = 0$ means: no overlap of network neighbors

$TOM(i,j) = 1$ means: identical set of network neighbors

$$TOM_{ij} = \frac{\sum_u a_{iu} a_{uj} + a_{ij}}{\min(k_i, k_j) + 1 - a_{ij}}$$
$$DistTOM_{ij} = 1 - TOM_{ij}$$

- ***NOTE that: Generalized to the case of weighted networks in Zhang and Horvath (2005), first WGCNA paper***

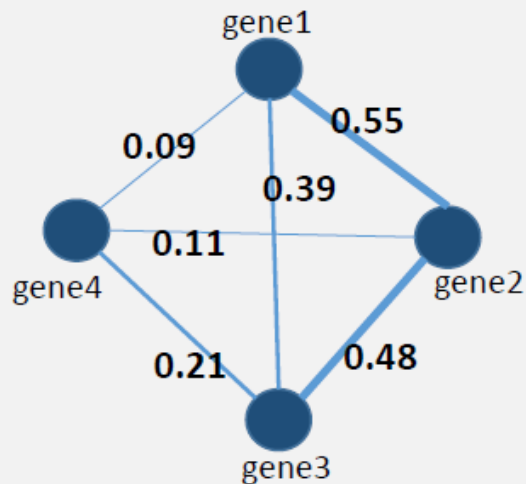
- All nodes are neighbors; counting them is not informative.
 - Compute agreement of the set of neighboring nodes based on edge strengths

- But, we need a **dissimilarity measure** for clustering!!
- TOM as a **similarity measure** can be transformed into a **dissimilarity measure**: **distTOM = 1-TOM**.

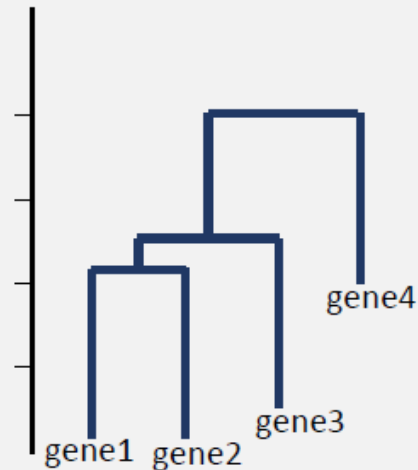
Step 2: Perform hierarchical clustering of genes

- **Compute gene dendrogram:**

Weighted correlation network
from gene expression data

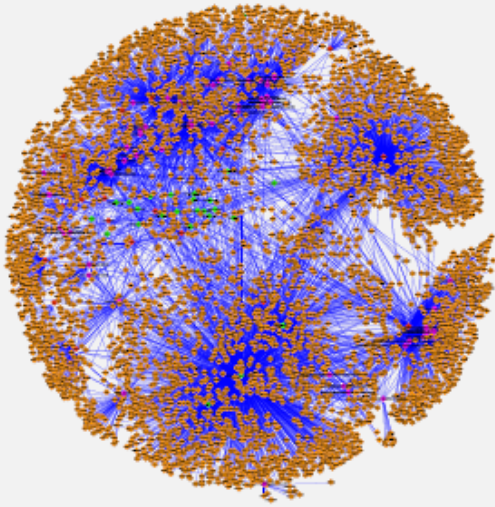


Clustering dendrogram



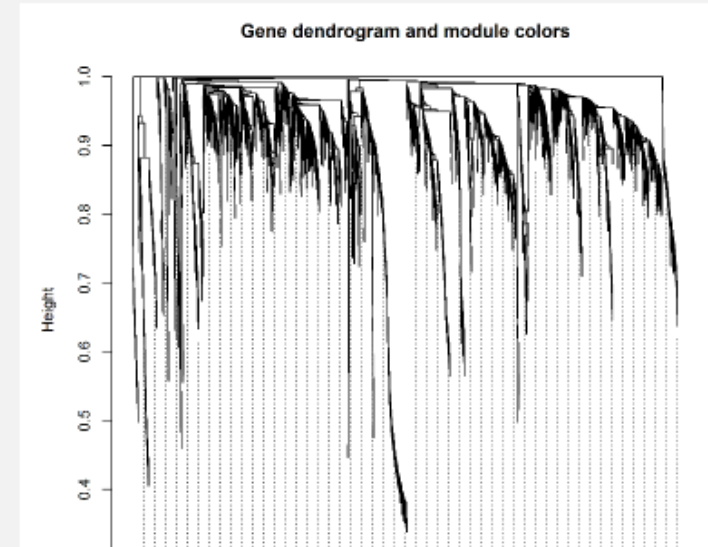
*(dis)similarity between genes:
Topological Overlap Measure TOM*

Weighted correlation network from gene expression data



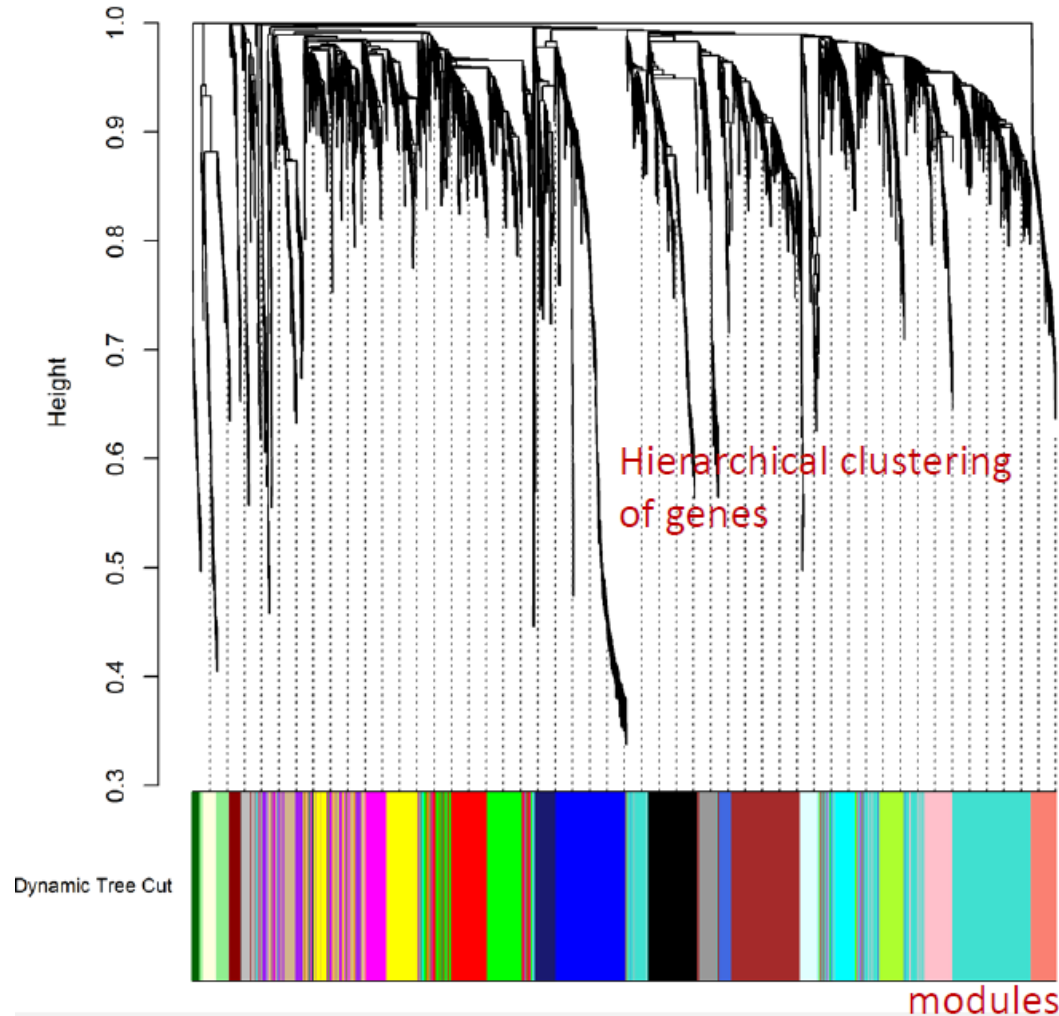
*(dis)similarity between genes:
Topological Overlap Measure TOM*

Clustering dendrogram



Step 3: Divide clustered genes into modules

- Gene dendrogram and detected modules

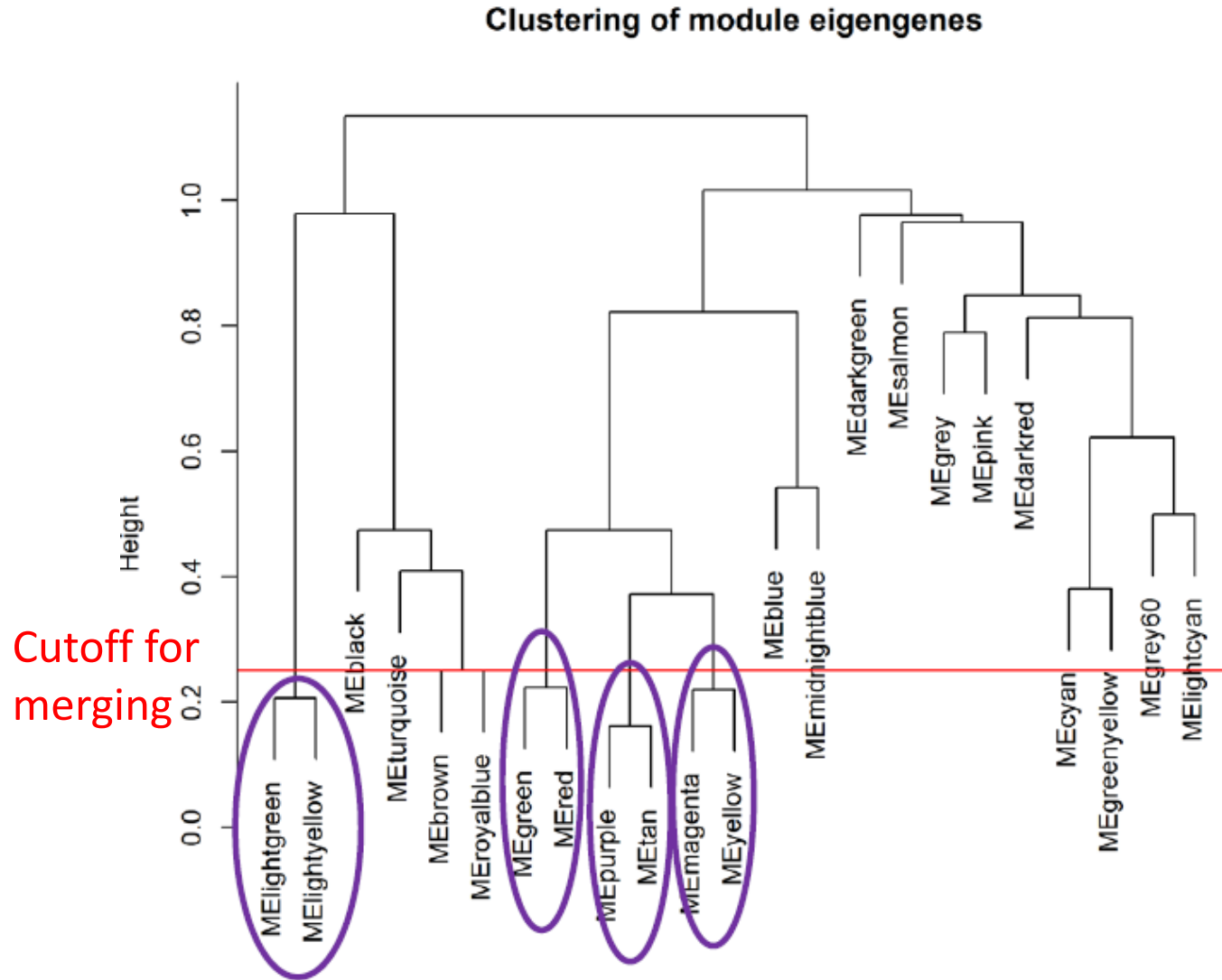


Dynamic tree cut algorithm groups genes into modules:
corFnc="pearson"; power=6; min.modulesize=30

Step 4 (Optional): Merge very similar modules

- **A module eigengene** is a 1-dimensional data vector, summarizing the expression data of the genes that form a module
- **How it is computed:** the 1st principal component of the expression data
- **What it is used for:** to represent the module in mathematical operations:
 - modules can be correlated with one another
 - modules could be **clustered together** (we can combine them)
 - modules can be correlated with external traits

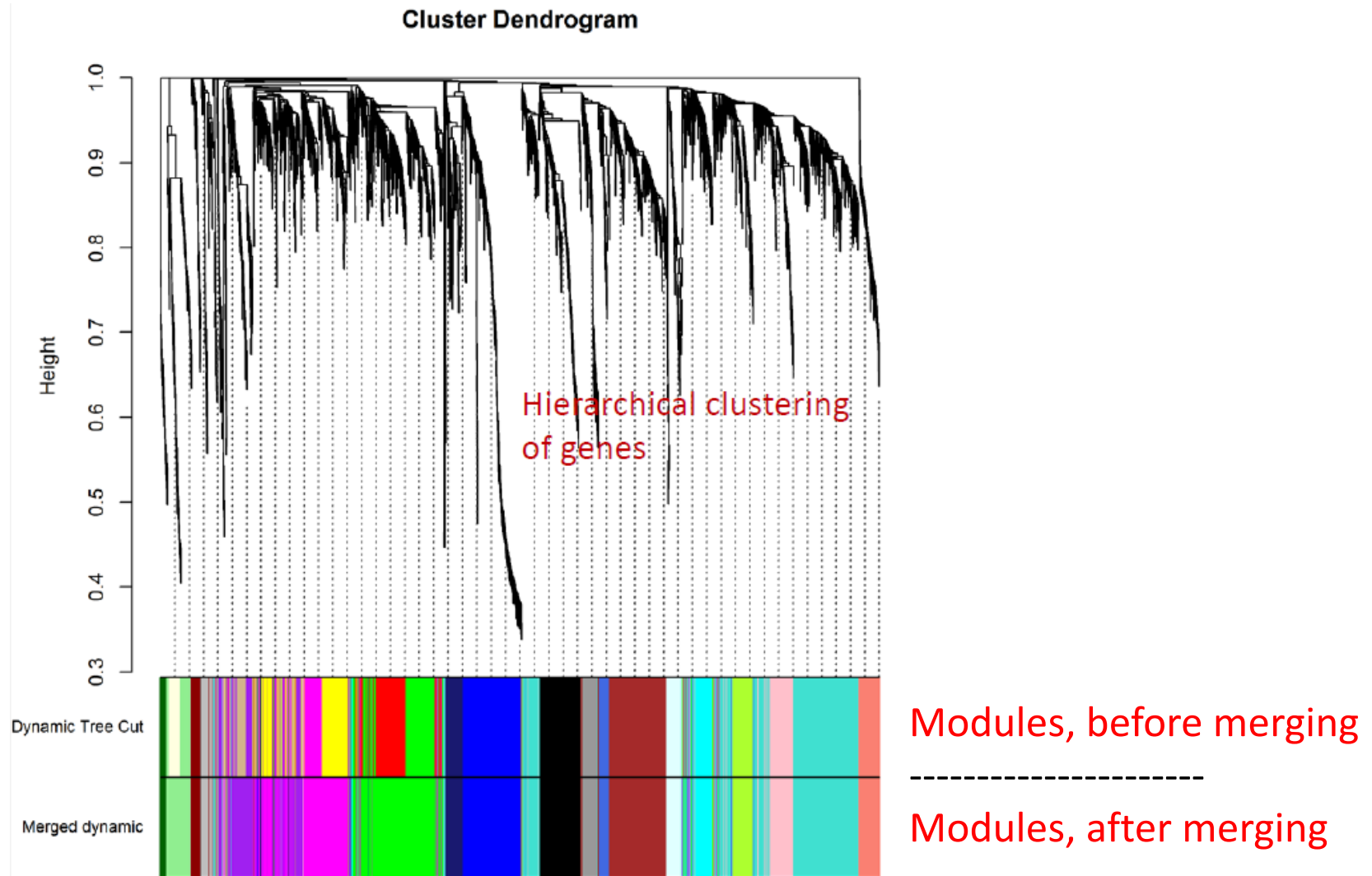
Clustering of module eigengenes



Dissimilarity measure: $1 - \text{cor}(\text{Meigengenes})$

Merge modules whose dissimilarity is below the merging cutoff

Gene dendrogram and detected modules, before and after merging



corFnc="pearson"; power=6; min.modulesize=30

Module Detection: Decisions to make

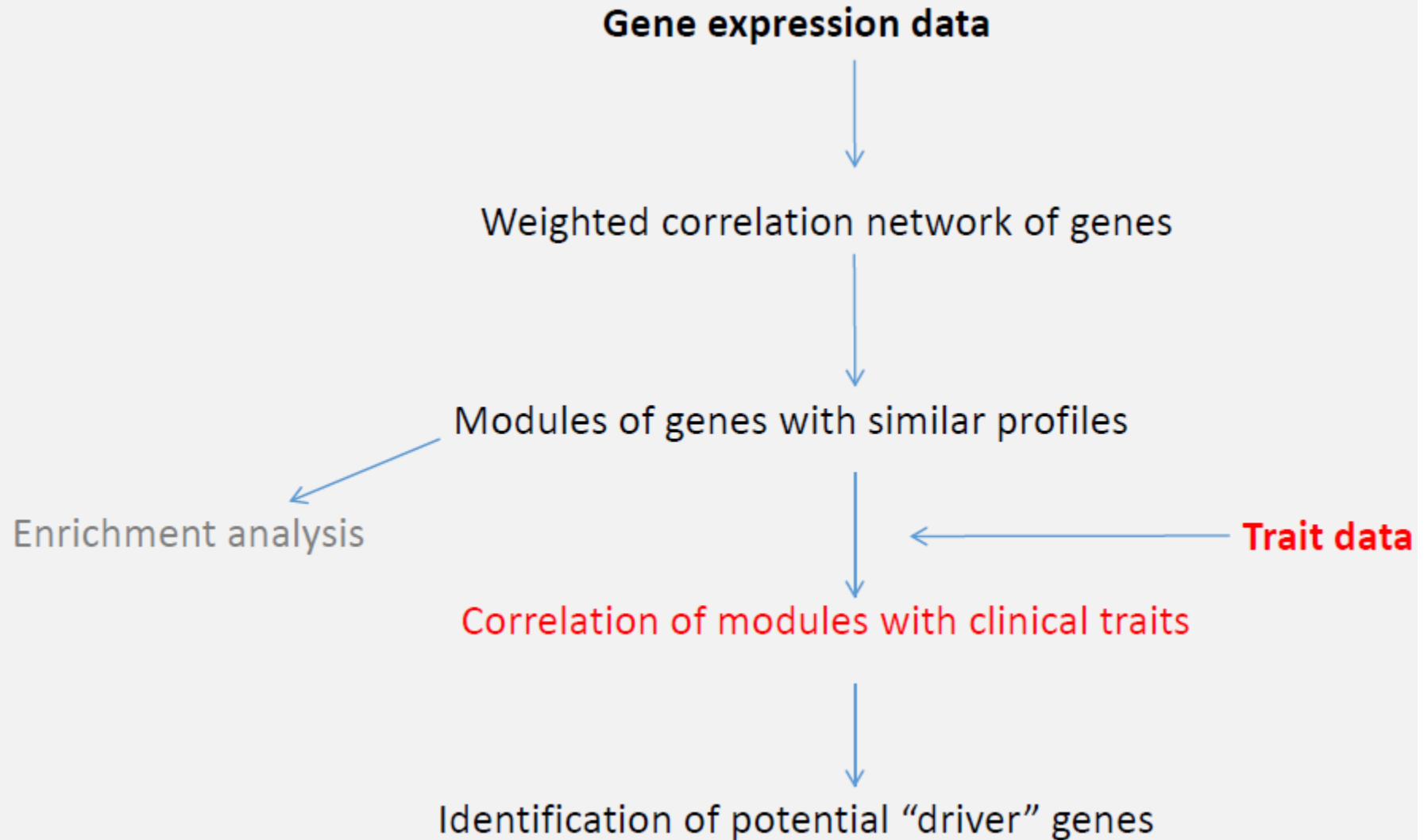
How to choose optimal parameters?

Dynamic Tree cut procedure

- **Minimal module size:** typically 20 or 30
- **CutHeight:** try different heights such as 0.95 or 0.9995, etc.

Module Merging procedure

- **Cutoff for module eigengene dendrogram:** typically between 0.15 and 0.25
 - *check if clusters look ok on dendrogram*
- **Merge once or several times?** Usually once, but merge step can be repeated:
 - *if some modules are very similar*
 - *if we want larger modules*



Correlate modules to external traits

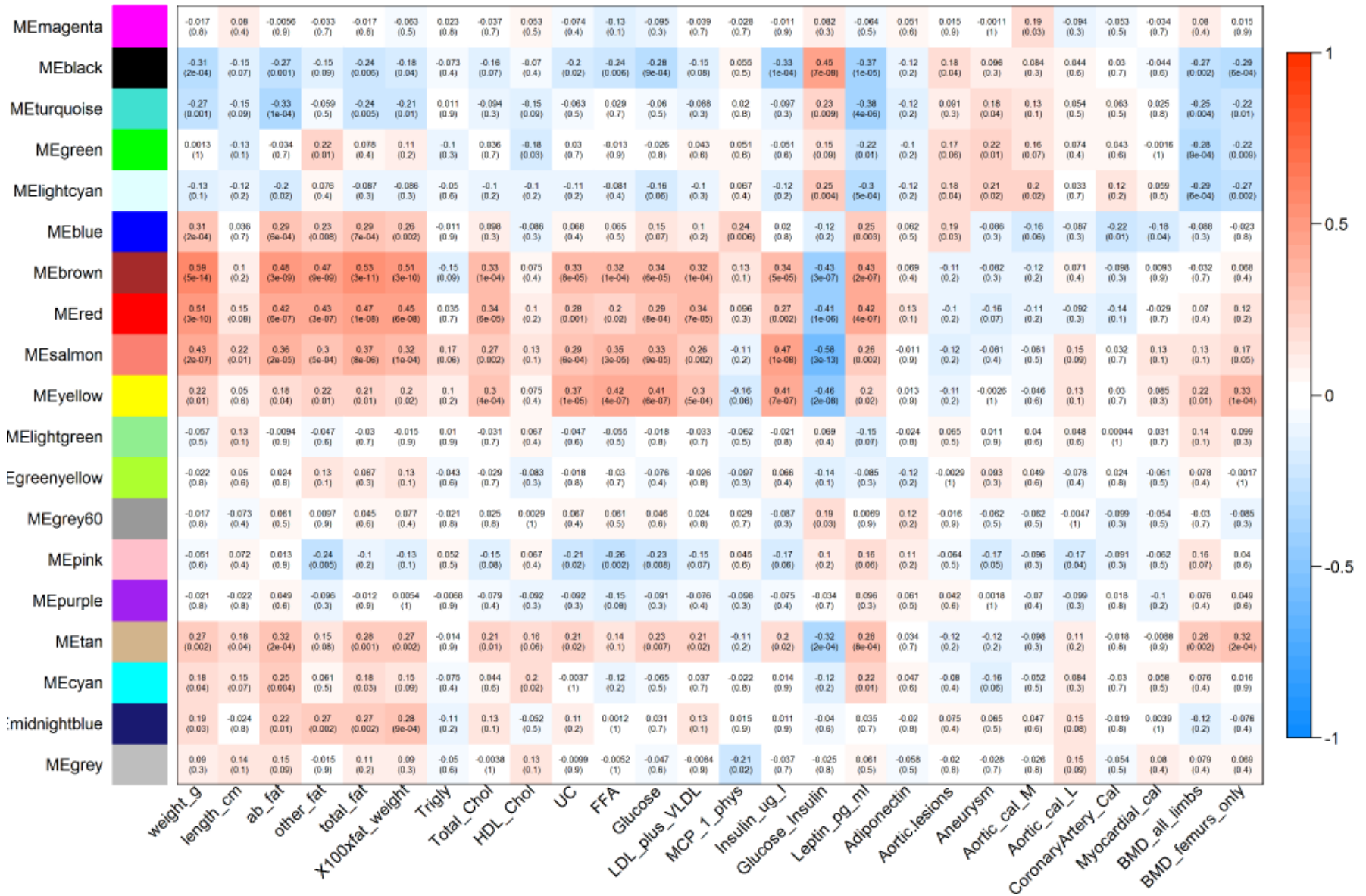
Examples: trait variables from obtained from some samples/individuals (preferably the same samples that we generated the expression data set):

- weight_g
- length_cm
- ab_fat
- other_fat
- total_fat
- Trigly
- Total_Chol
- HDL_Chol
- UC
- FFA
- Glucose
- LDL_plus_VLDL
- MCP_1_phys
- Insulin_ug_l
- Glucose_Insulin
- Leptin_pg_ml
- Adiponectin
- Aortic lesions
- Aneurysm
- Aortic_cal_M
- Aortic_cal_L
- CoronaryArtery_Cal
- Myocardial_cal
- BMD_all_limbs
- BMD_femurs_only

How to compute correlations: each module eigengene to each trait variable:
`cor(MEs, traitDat)`

Example

Module-trait relationships



Module preservation analysis in WGCNA

Is my network module preserved and reproducible?*

*** Langfelder et al PloS Comp Biol. 7(1): e1001057.**

Network-based module preservation statistics

- Input: module assignment in reference data.
- Adjacency matrices in **reference** A^{ref} and **test** data A^{test}
- Network preservation statistics assess preservation of
 - 1. network density: Does the module remain densely connected in the test network?
 - 2. connectivity: Is hub gene status preserved between reference and test networks?
 - 3. separability of modules: Does the module remain distinct in the test data?

Several connectivity preservation statistics

For general networks, i.e. input adjacency matrices

- $\text{cor.kIM} = \text{cor}(\text{kIM}^{\text{ref}}, \text{kIM}^{\text{test}})$
 - *correlation of intramodular connectivity across module nodes*
- $\text{cor.ADJ} = \text{cor}(\mathbf{A}^{\text{ref}}, \mathbf{A}^{\text{test}})$
 - *correlation of adjacency across module nodes*

For correlation networks, i.e. input sets are variable measurements

- $\text{cor.Cor} = \text{cor}(\text{cor}^{\text{ref}}, \text{cor}^{\text{test}})$
- $\text{cor.kME} = \text{cor}(\text{kME}^{\text{ref}}, \text{kME}^{\text{test}})$

One can derive relationships among these statistics in case of weighted correlation network

Choosing thresholds for preservation statistics based on permutation test

- For correlation networks, we study **4 density** and **3 connectivity** preservation statistics that take on values ≤ 1
- Challenge: Thresholds could depend on many factors (number of genes, number of samples, biology, expression platform, etc.)
- Solution: Permutation test. Repeatedly permute the gene labels in the test network to estimate the mean and standard deviation under the null hypothesis of no preservation.
- Next we calculate a **Z statistic**: $Z = (\text{observed} - \text{mean}) / \text{sd}$
- We have had **4 density** and **3 connectivity** preservation statistics:

$$Z_{\text{density}} = \text{median}(Z_{\text{meanCor}}, Z_{\text{meanAdj}}, Z_{\text{propVarExpl}}, Z_{\text{meanKME}}).$$

$$Z_{\text{connectivity}} = \text{median}(Z_{\text{cor.kIM}}, Z_{\text{cor.kME}}, Z_{\text{cor.cor}}).$$

Permutation test for estimating Z scores

- For each preservation measure we report the observed value and the permutation Z score to measure significance ($Z = (\text{observed} - \text{mean}) / \text{sd}$).
- Each Z score provides answer to “Is the module significantly better than a random sample of genes?”
- Summarize the individual Z scores into a composite measure called **Z.summary**

$$Z_{summary} = \frac{Z_{density} + Z_{connectivity}}{2}.$$

- **Z.summary < 2 indicates no preservation,**
- **2 < Z.summary < 10 weak to moderate evidence of preservation,**
- **Z.summary > 10 strong evidence**

Summary preservation

- Standard cross-tabulation based statistics are intuitive
 - Disadvantages: i) only applicable for modules defined via a module detection procedure, ii) ill suited for ruling out module preservation
- Network based preservation statistics measure different aspects of module preservation
 - Density-, connectivity-, separability preservation
- Two types of composite statistics: **Zsummary** and **medianRank**.
- Composite statistic **Zsummary** based on a permutation test
 - Advantages: thresholds can be defined, R function also calculates corresponding permutation test p-values
 - Example: $Z_{summary} < 2$ indicates that the module is *not* preserved
 - Disadvantages: i) **Zsummary** is computationally intensive since it is based on a permutation test, ii) often depends on module size
- Composite statistic **medianRank**
 - Advantages: i) fast computation (no need for permutations), ii) no dependence on module size.
 - Disadvantage: only applicable for ranking modules (i.e. relative preservation)

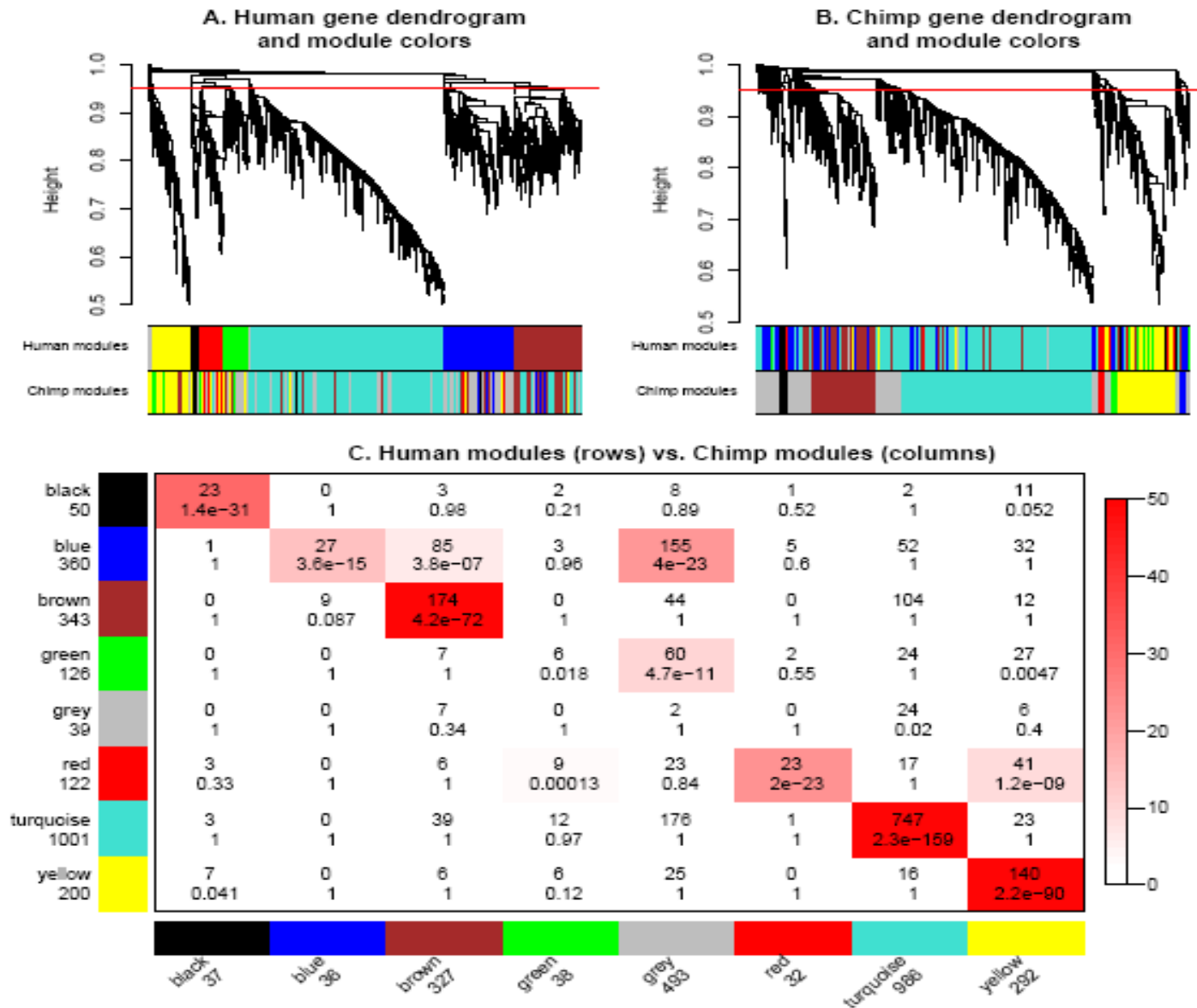
Application:

Studying the preservation of human brain co-expression modules in chimpanzee brain expression data.

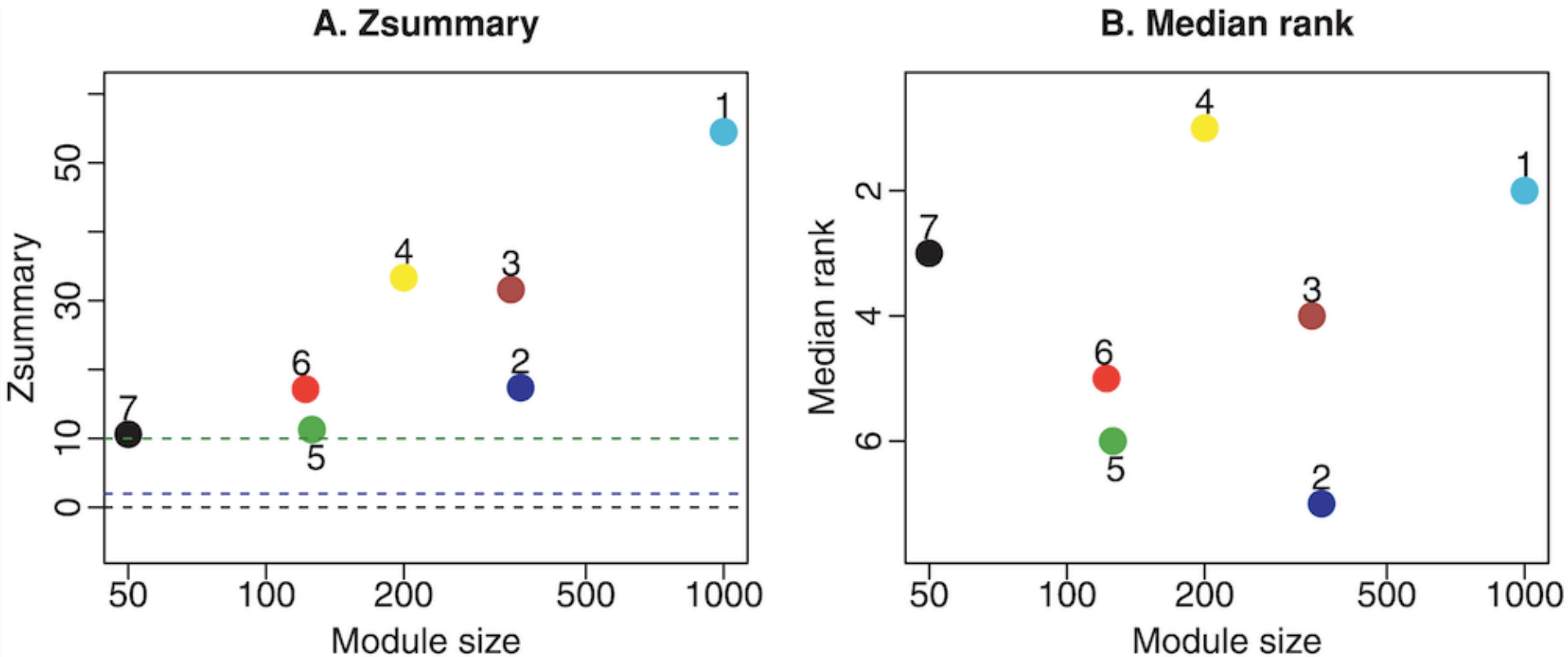
Modules defined as clusters
(branches of a cluster tree)

Data from Oldam et al 2006

Preservation of modules between human and chimpanzee brain networks



2 composite preservation statistics



- Zsummary is above the threshold of 10 (green dashed line), i.e. all modules are preserved.
- Zsummary often shows a dependence on module size which may or may not be attractive
- In contrast, the median rank statistic is not dependent on module size.
- It indicates that the yellow module is the most preserved one

Implementation and R software tutorials, WGCNA R library

- General information on weighted correlation networks
- Google search
 - “WGCNA”
 - “weighted gene co-expression network”
- R function `modulePreservation` is part of WGCNA package
- Tutorials: preservation between human and chimp brains

www.genetics.ucla.edu/labs/horvath/CoexpressionNetwork/ModulePreservation

Pseudocode for modulePreservation()

- `setLabels = c("Control", "Disease");`
- `multiExpr = list(Control = list(data = dataExp_Control), Disease = list(data = dataExp_Disease));`
- `multiLabel = list(Control = moduleLabelsControl, Disease=moduleLabelsDisease);`
- `mp = modulePreservation(multiExpr, multiLabel, referenceNetworks = c(1:2), nPermutations = 100, randomSeed = 1, quickCor = 0, verbose =3);`
- `save(multiExpr, multiLabel, mp, file =“modulepreservationLabel_Alldata.RData”)`
- `ref = 1; test = 2`
- `Zsummary1=mp$preservation$Z[[ref]][[test]][, 2]`
- `names(Zsummary1)=rownames(mp$preservation$Z[[ref]][[test]])`
- `low.preserved1=Zsummary1[which(Zsummary1<2)]`
- `ref=2; test=1`
- `Zsummary2=mp$preservation$Z[[ref]][[test]][, 2]`
- `names(Zsummary2)=rownames(mp$preservation$Z[[ref]][[test]])`
- `low.preserved2=Zsummary2[which(Zsummary2<2)]`

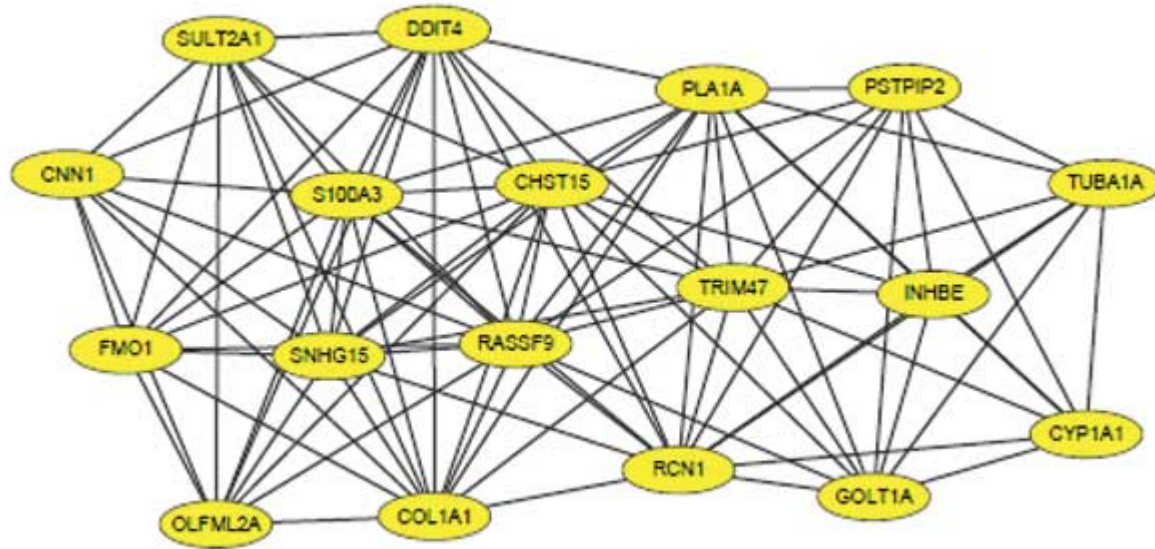
Introduction to Bayesian Networks

Content

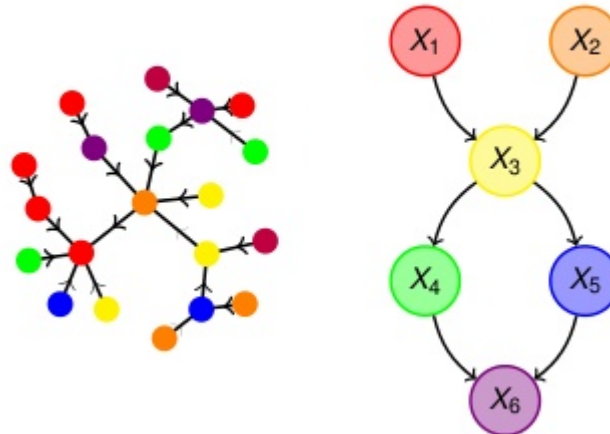
- Definition of Bayesian networks (BN)
- An example, Rimbanet...
- Mandatory and optional input files
- Comparison of BN tools

Gene-gene interaction networks

Co-expression networks

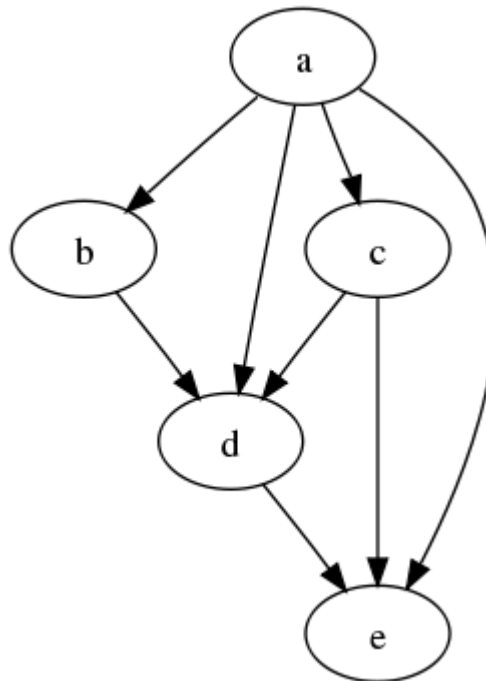


Bayesian networks



What is a DAG

- A directed acyclic graph (DAG) is a finite directed graph with no directed cycles.
- There is no way to start at any vertex v and follow a consistently-directed sequence of edges that eventually loops back to v again.



Bayes' theorem

- Bayes' theorem describes the probability of an event, based on prior knowledge of conditions that might be related to the event.
- Bayes' theorem is stated as:



The picture can't be displayed.

Bayes' theorem - an example

- Suppose a test to detect a disease in human is 99% sensitive and 95% specific, means:

For the patients, the test gives a positive result with 99% (TP) and gives a negative result with 1% (FN): $P(+ | C)=0.99$ and $P(- | C)=0.01$


For the healthy samples, test gives a negative result with 95% (TN) and a positive result with 5% (FP): $P(- | H)=0.95$ and $P(+ | H)=0.05$

- Q: If a randomly selected individual's test result is positive, what is the probability that he/she carries the disease? Hence, $P(C | +) = ?$**

$$\begin{aligned} P(C|+) &= \frac{P(+|C)P(C)}{P(+)} = \frac{P(+|C)P(C)}{\sum_i P(+|Option_i)P(Option_i)} \\ &= \frac{P(+|C)P(C)}{P(+|C)P(C) + P(+|H)P(H)} = \frac{0.99 \times 0.02}{0.99 \times 0.02 + 0.05 \times 0.98} \approx 29\% \end{aligned}$$


Bayesian networks (BNs)

- BNs are directed acyclic graphs (DAGs) in which the edges of the graph are defined by conditional probabilities that characterize the distribution of the states of each node given the state of its parents.


 The picture can't be displayed.


, where M: network model, D: observation data

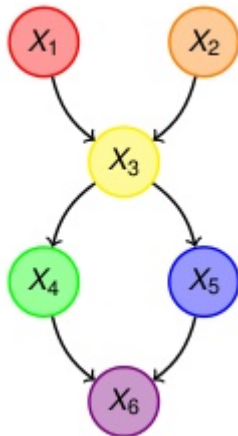
- We can define a partitioned joint probability distribution over all nodes:

 The picture can't be displayed.

where

 The picture can't be displayed.

is parent set of 



$$P(X_1, \dots, X_6) = P(X_1)P(X_2)P(X_3|X_1, X_2) \\ P(X_4|X_3)P(X_5|X_3)P(X_6|X_4, X_5)$$

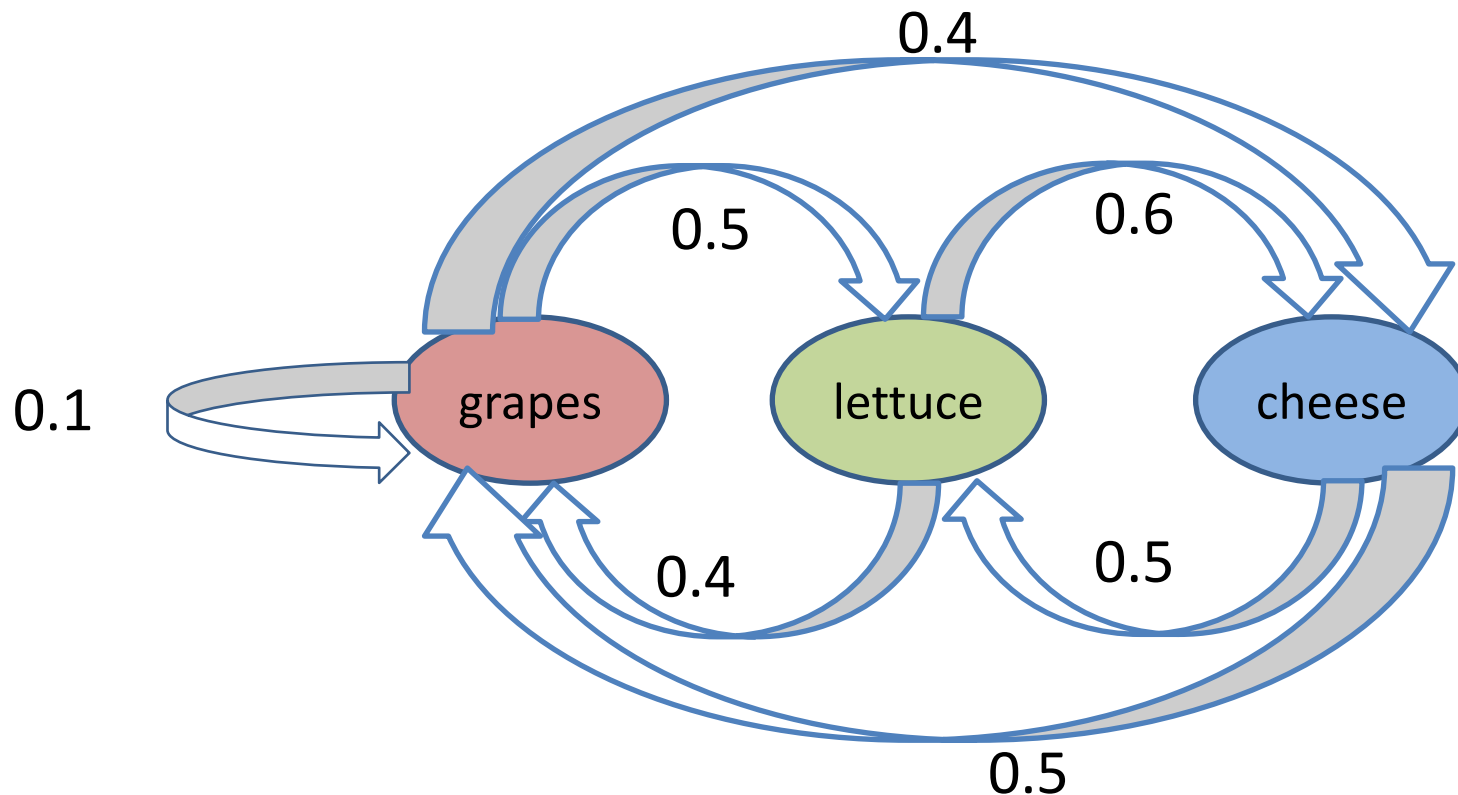
Application of Bayes' theorem in gene regulatory network inference

$$P(M | D) = P(X) = \prod_i P(X^i | \text{Pa}(X^i))$$

- The numbers of the possible network structures (M) \uparrow with the # of nodes \uparrow .
- Search of all possible structures to find the best supported one by the data is not feasible.
- Solution: We can **use Markov chain Monte Carlo (MCMC)** simulation to identify a particular amount (e.g. 1,000) of plausible networks.
- A consensus network is obtained by combining these plausible networks.

Markov chain


- State of the system at time “t+1” is predicted by solely using the state at time “t”, i.e. previous states (t-1, t-2,..) will not be used for predicting t+1.



Markov chain Monte Carlo (MCMC)

- MCMC methods are a class of sampling algorithms.
- It is used for sampling from a probability distribution based on a **Markov chain** model that has the desired distribution as its equilibrium distrib.
- The state of the chain after a number of steps is used as a sample of the desired distribution.
- The quality of the sample \uparrow as the number of steps \uparrow .
- MCMC methods mainly used for numerical approximations of the integrals

MCMC in the BN problem

- MCMC algorithm to identify 1000 of networks:
 - Start with a null network (for each 1000 cases) including prior edges.
 - Make random changes on each network such as:
 - Flip,
 - Add, and
 - Delete individual edges;
 - Accept the random changes made above that lead to an improvement in the fit of the network to the data. $P(M | D) \approx P(D | M) \times P(M)$
 - The fitness is assessed by **Bayesian information criterion (BIC)**, which also penalizes the network if the #of the parameters (complexity) \uparrow .
 - **Note:**  The picture can't be displayed. where n is # of data points in D (sample #); k = # of the parameters to be estimated (parent # of node i).
- Create a consensus network by combining above 1000 networks and check for the DAG feature of the final network.

Comparison of BN tools

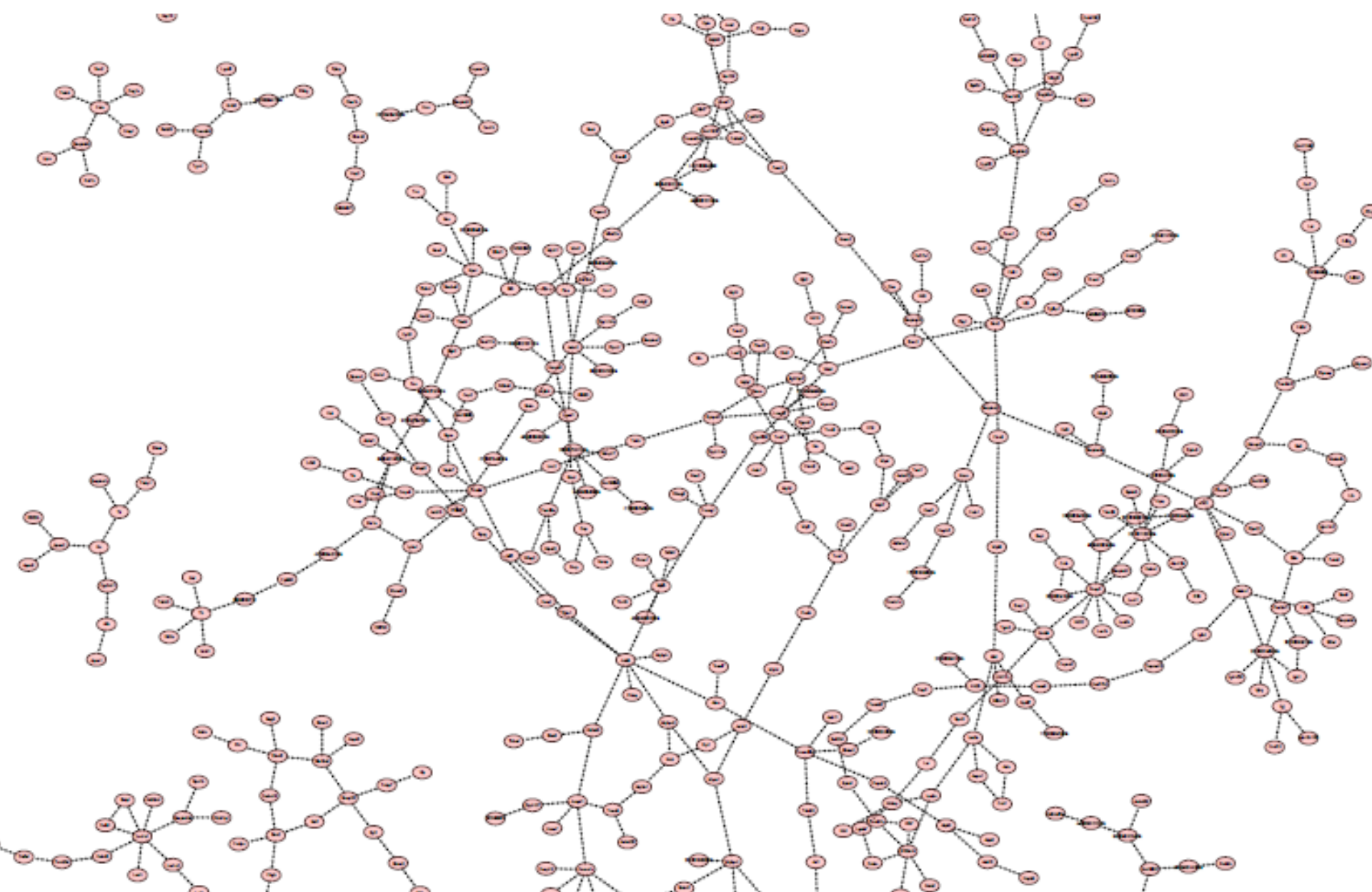
Software packages

Tool	Platform	Scoring method(s)	Data type	Multi processing?	Prior info support?
RIMBANet	Perl+C	BIC	Continuous + Discrete	No	Yes
Bnfinder	Python	BIC , BDE, MIT	Continuous + Discrete	Yes	Yes
sparsebn	R	BIC+CV	Continuous + Discrete	No	No
bnlearn	R	BIC, AIC, BDE	Continuous + Discrete	Yes	Yes
catnet	R	BIC, AIC	Continuous + Discrete	Yes	Yes

- BIC: Bayesian information criterion
- BDE: Bayesian Dirichlet equivalence
- MIT: χ^2 distance based Mutual information test
- AIC: Akaike information criterion
- CV: cross validation

Example - 500 genes 100 samples from an aorta tissue expression dataset

Tool	Scoring method	Runtime (sec)	Edge #
RIMBANet	BIC	5,586 sec (~1.30 hour) on 1 node	502
Bnfinder	BIC	12,710 sec (~3.30 hours) on 8 nodes (Could not finish in 24h on 1 node)	479
sparsebn	BIC	2,409 sec on 1 node Prior info integration is not available*	5,019
bnlearn	BIC	Could not finish in 24h (on 8 nodes)	--
catnet	BIC	Could not finish in 24h (on 8 nodes)	--



Summary

- Gene networks provide us to:
 - Identify biological mechanisms and molecular subnetworks underlying common human diseases.
 - Integrating diverse type of multi-dimensional biological datasets.
 - Predicting the key driver genes in disease-related subnets.
- After presenting our data-driven findings, experimentalists can conduct *in vivo* and/or *in vitro* experiments to test our candidate genes on a given tissue, for certain hallmarks of a disease/disorder.