

# Introduction to Bioinformatics

# Goal

**Goal of molecular cell biology** - to understand the physiology of living cells in terms of the information that is encoded in the **genome** of the cell

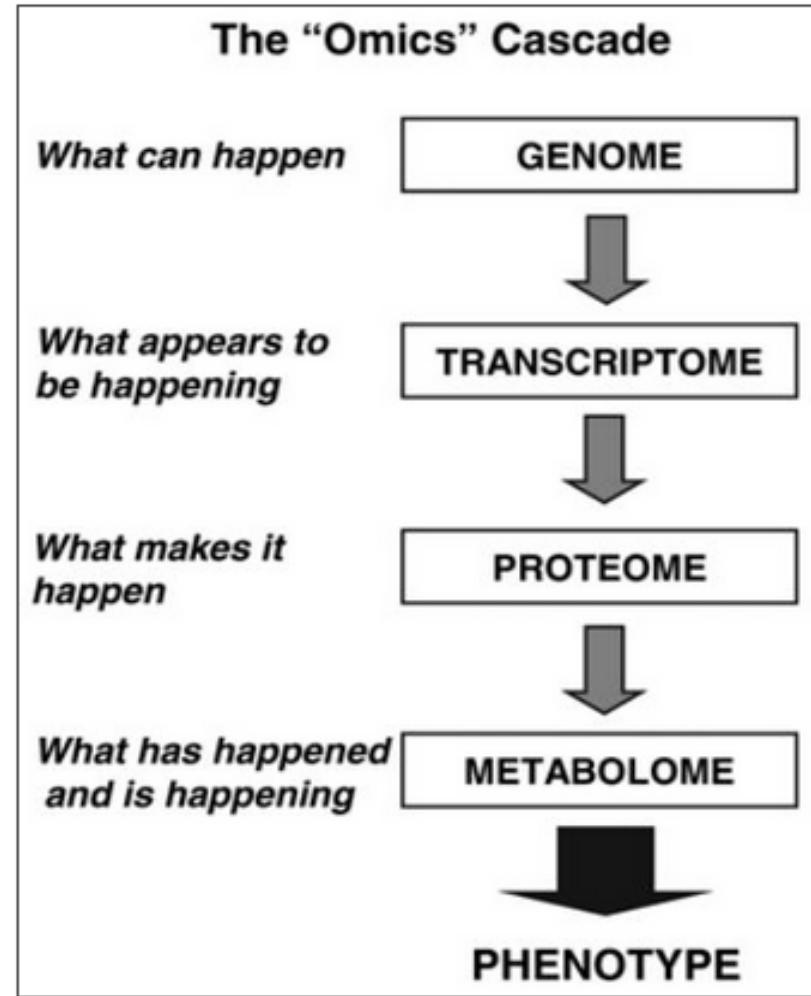
**How computer science can help  
in achieving this goal?**

⇒ **Location of the genes in the genome, its function, what factors affect its expression, in normal vs disease state, etc.**

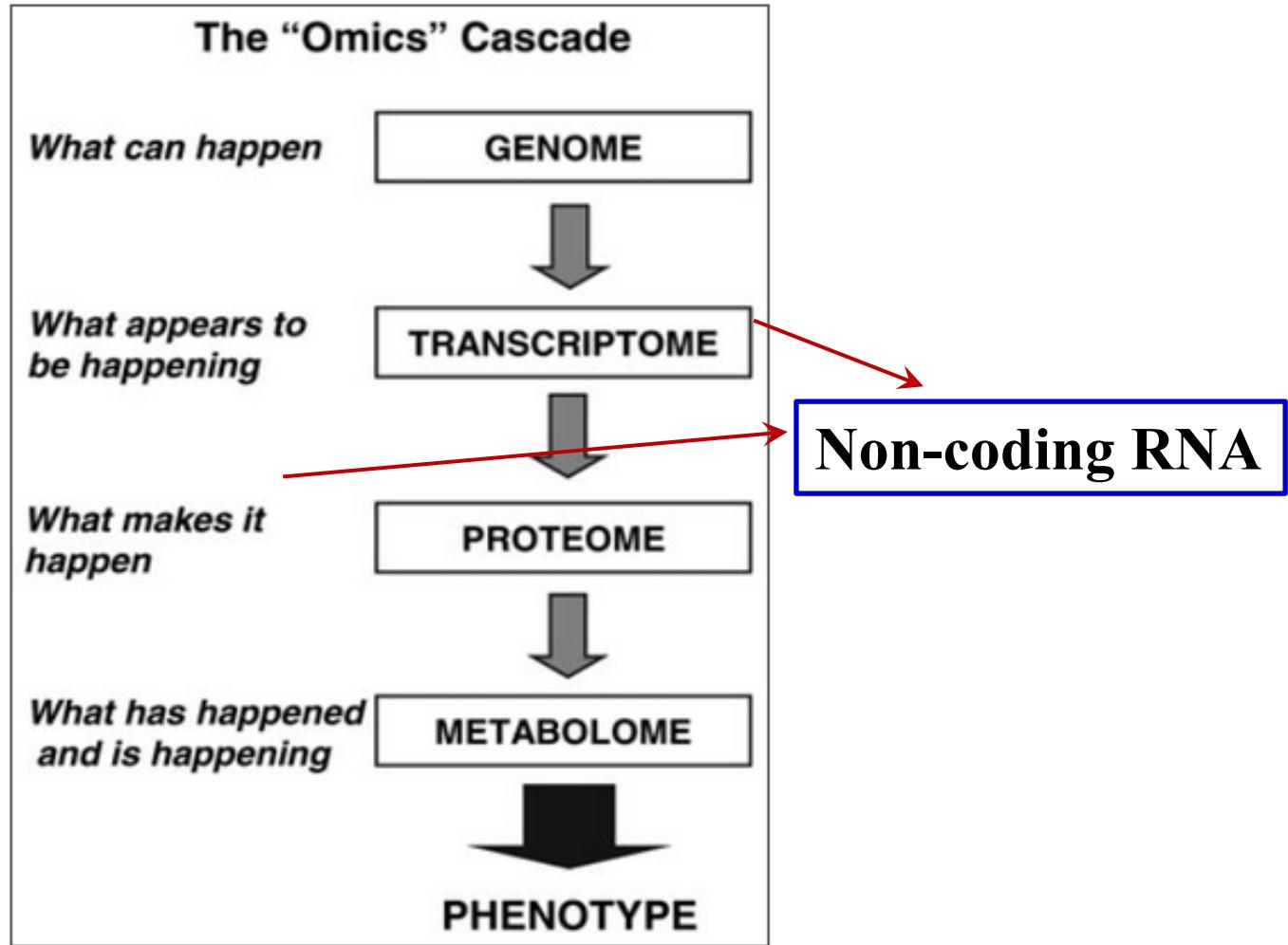
‘Bioinformatics’ was coined by Paulien Hogeweg in 1979, for the study of informatic processes in biological systems.

- it’s a **data-driven** field to gain insight into what happens in the living cell of an organism using various types of biological data

Various Omics studies, *viz.*, Genomics, Transcriptomics, Proteomics & Metabolomics are **data-driven fields** that aim to answer the question of how genomes code for living organisms.



Major inputs from CS – develop algorithms for mining meaningful information from biological data, and develop efficient data storage and data retrieval systems for managing large volumes of data



**Major inputs from CS – develop algorithms for mining meaningful information from biological data, and develop efficient data storage and data retrieval systems for managing large volumes of data**

# Biological Data: Levels of Organization

Central Dogma of MB

DNA  
↓  
mRNA

...TACCCCGATGGCGAAATGC...

Sequence/Structure Alignment, Pattern Recognition,

Central Dogma of Bioinformatics

Protein  
↓  
Enzyme

...AUGGGCUACCGCUUUACG...

Pattern Recognition, Molecular Modeling

Metabolic Pathways, Interacting Networks

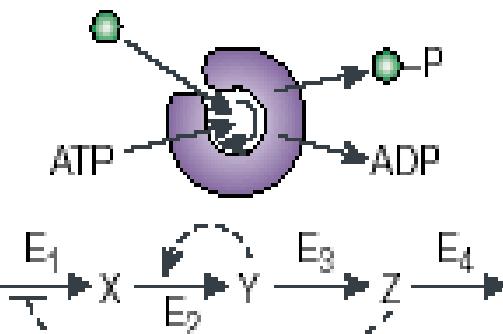
↓  
Enzyme

...Met-Gly-Tyr-Arg-Phe-Thr...

Molecular Modeling

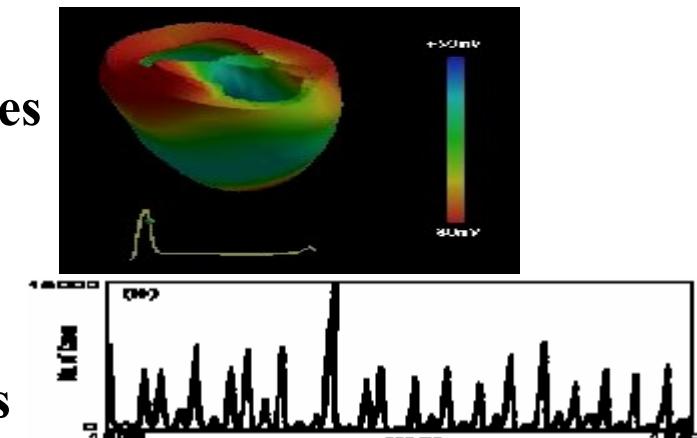
Intercellular Interactions

↓  
Reaction network



Network Modeling, Dynamical Systems

Organs/Tissues Physiology



Modeling, Differential Eqns,

Graph Theory, Chaos Theory,

Pattern Formation, Characterization,

Time-series data analysis

Ecological modeling  
Inter-species interactions

If it were required, in a single model, to span all the scales from

**molecular motions**       $\Rightarrow$       **cell responses**  
(nanometers/picoseconds)      (micrometers/secs)

- theoretical approach to cell physiology would be beyond grasp, both computationally & intellectually

**Fortunately, considerable progress can be made – at any given level of hierarchy – independent of the successes or failures at levels above/below**

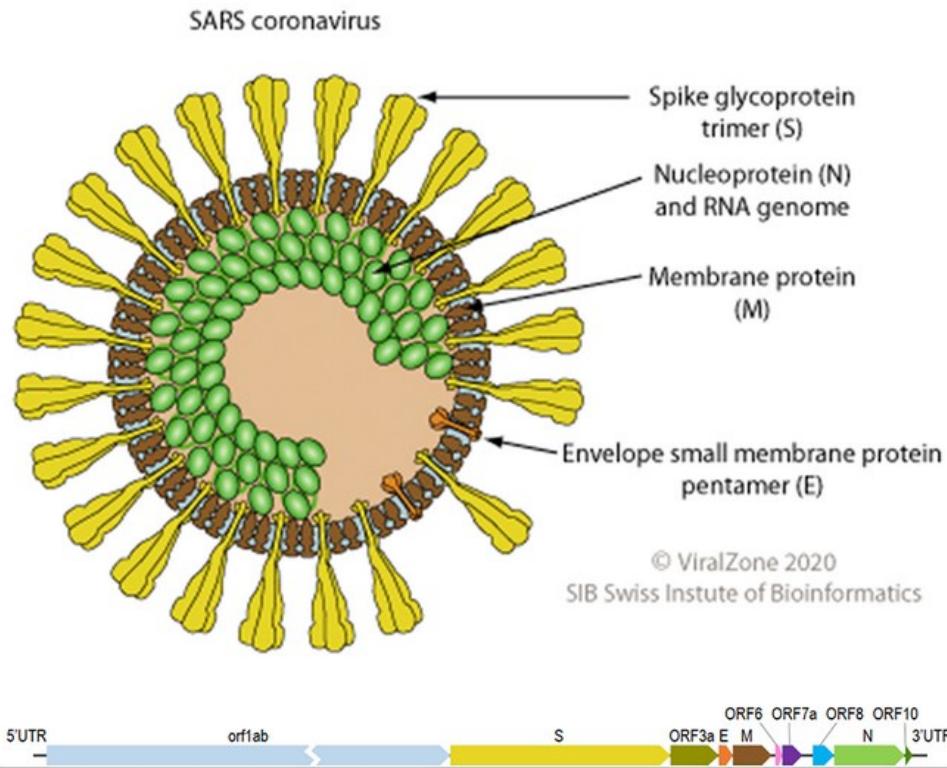
Systems biology is aiming towards achieving this goal of understanding the functional behaviour of a cell/tissue as a whole.

## Disease - COVID19

- When a new virus strikes the population, such as we saw in the case of COVID-19, no specific treatment is available

**What kind of sequence analysis can help in combating the disease?**

## SARS-CoV-2



# **What kind of Bioinformatics analysis can we carry out to know about the virus causing COVID-19?**

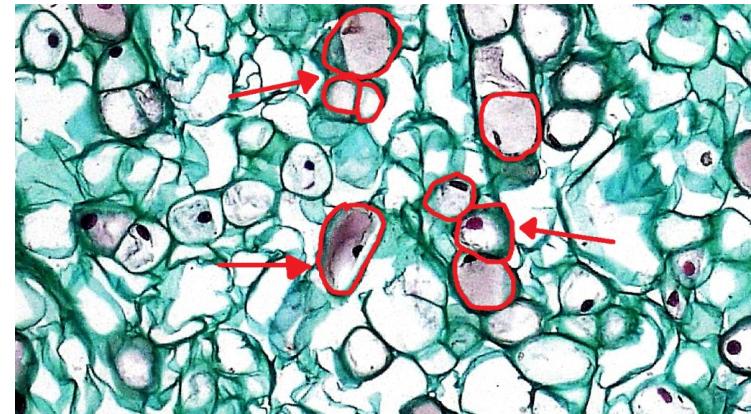
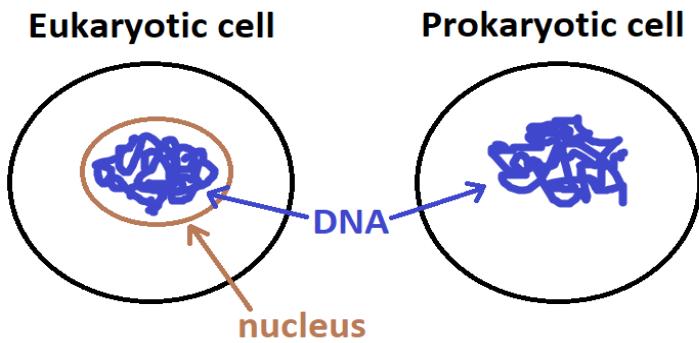
- How to identify if a person is infected with SAR-COV-2?
- Is it the only known human coronavirus?
- Comparing its genome with other viral genomes – to identify its closest relative
- What proteins aid in its transmission and infection?
- Identifying drug targets and develop vaccines
- What organs/tissues are affected by its infection?
- Its rate of propagation
- Is it mutating and becoming more virulent, or milder with time
- etc.

# The Cell

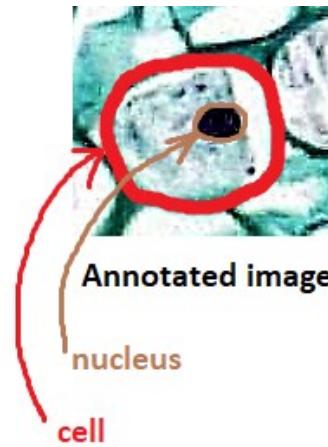
**Cell** - the basic building block of all living creatures.

All forms of “omics” measure large quantities of something found inside cells.

Cells are of two types:



Original image



Apart from prokaryotes and eukaryotes, there is a third category called **Viruses**, acellular entities. May contain DNA or RNA as their nuclear material.

# Cells and Chromosomes

E. B. Wilson: “the key to every biological problem must be sought in the cell; for every living organism is, or at some time has been, a cell.”

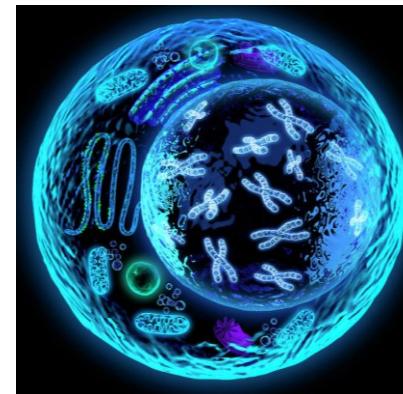
~  $10^{13}$  cells that form a human body, the whole organism has been generated by cell divisions from a single cell

Cells are the fundamental units of life - the vehicle for all the hereditary information that defines each species.

**Genome** – the total DNA content of an organism

**Chromosomes** – are physically separate molecules that range in length from ~ 50 - 250Mbp

In mammals and many other eukaryotes, the chromosomes occur in homologous pairs, called **diploids**, except for sex chromosomes.



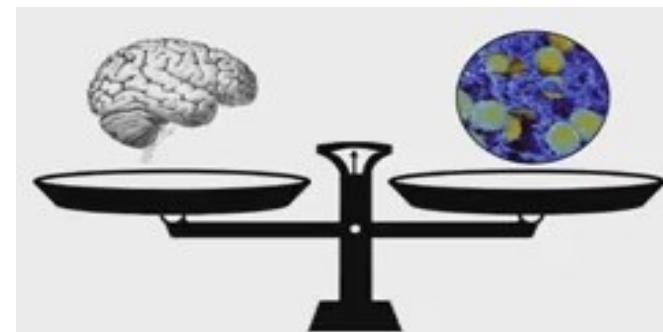
Organism	Number of chromosomes
pea plant	14
sun flower	34
cat	38
puffer fish	42
human	46
dog	78

# Cells and Genomes

**Do we carry cells of any other organism within us,  
apart from human cells?**

# How human are we?

- We have 10 trillion human cells and 100 trillion microbial cells: with respect to cell count we are just 10% human
- Our genome has 20-30K genes, our microbiome has 2-20M genes: with respect to genes we are 0.1-1% human
- Our microbiome weighs ~ 3 pounds, about the same weight as our brain, and maybe as important to our well being, if not more!
- We share 99.9% of our genome with other individuals, but we share only 10% of our microbiome
- Microbiota include bacteria, archaea, viruses, eukaryota



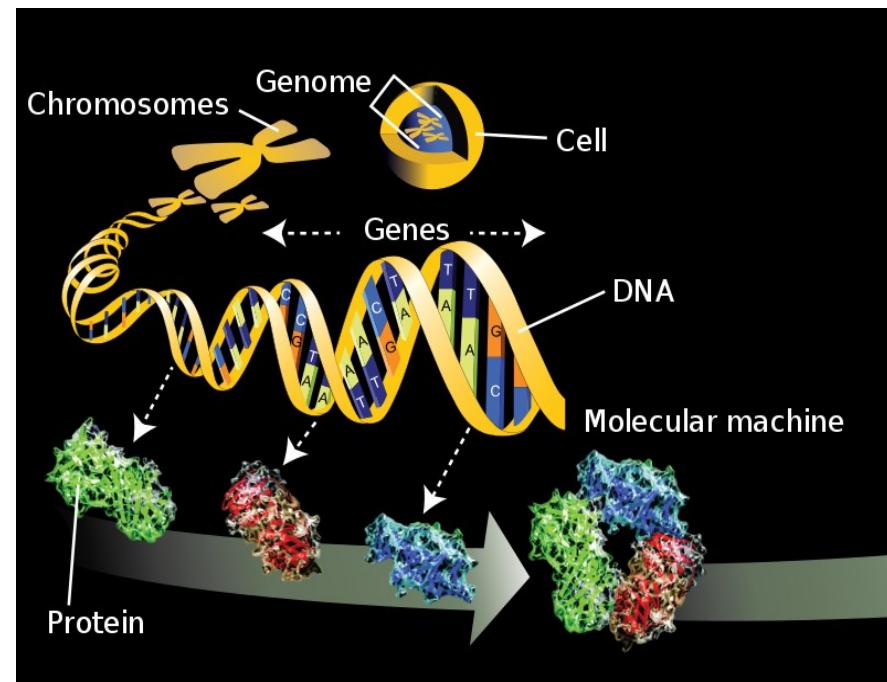
# DNA and Genes

**Human DNA - a long sequence of 3 billion letters, inside every cell in our body. There are ~ 37.2 trillion cells in our body.**

**A “gene” is a particular segment of DNA that encodes instructions for making a protein, hence genes are referred to as the “coding” part of the genome.**

**~ 25,000 genes covering ~ 2-3% of the human genome.**

**Function of remaining 98% of “non-coding” part of the genome?**



# DNA

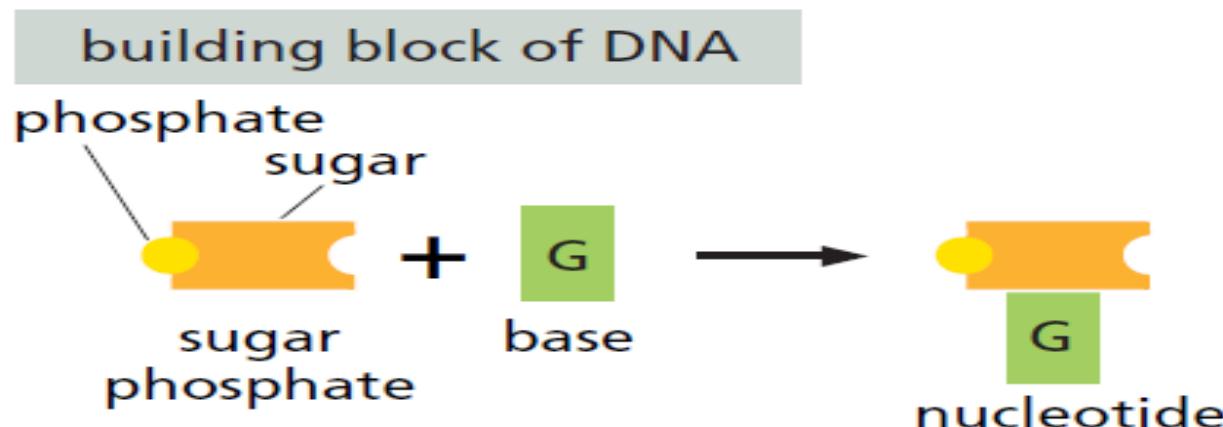
**DNA (Deoxyribonucleic acid):**

Composed of four basic units - called **nucleotides**

Each nucleotide contains - a **sugar**, a **phosphate** and one of the four bases:

**Adenine (A), Thymine (T),**

**Guanine (G), Cytosine (C).**

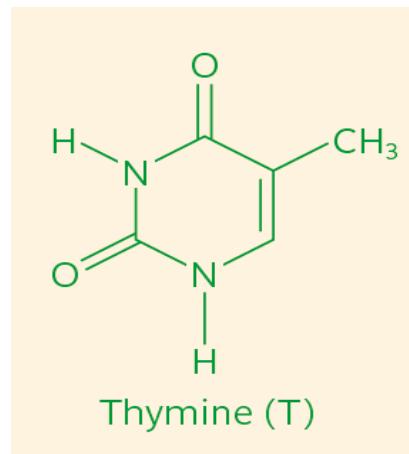
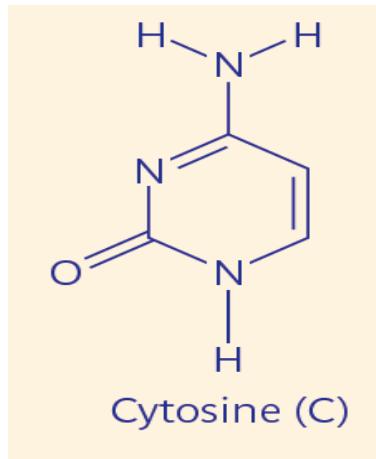


# DNA

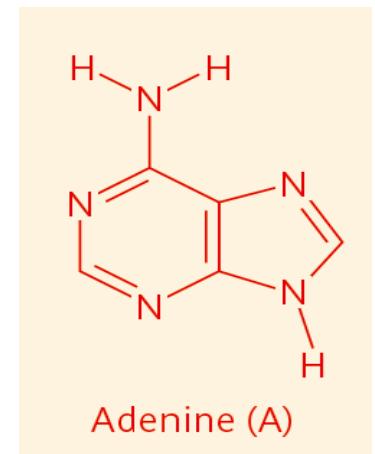
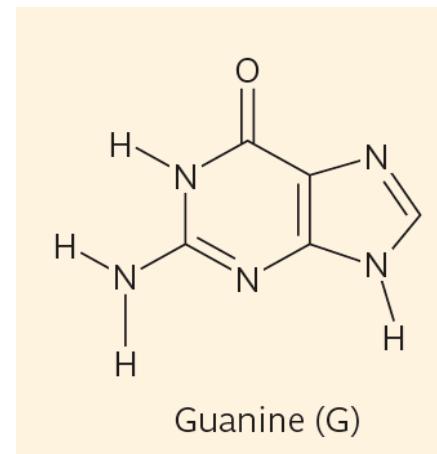
**Bases:** are ring-shaped and come in four types which fit together in pairs - this pairing forms the basis of information carrying capacity of DNA.

These are categorized as:

## Pyrimidines



## Purines



Which of these form base-pairs?

# DNA

DNA is **double-stranded** - the two strands of DNA wind around each other to form a double helix.

Information in one strand is a “**mirror copy**” of the information in the other strand, achieved by base-pairing:

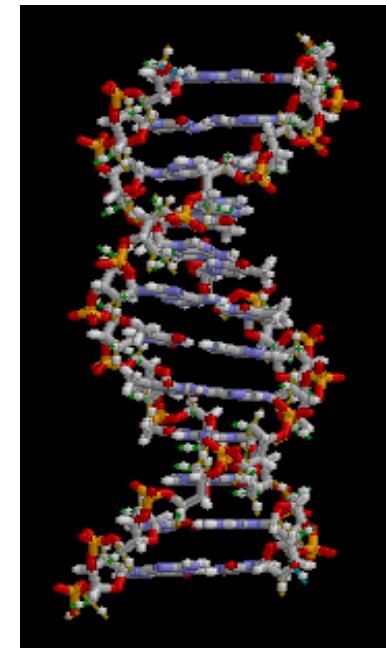
$$A \Leftrightarrow T, \quad G \Leftrightarrow C$$

So, if the sequence on one strand is GATTACA,  
what is the sequence of the other strand?

Importance of double-stranded nature of DNA:

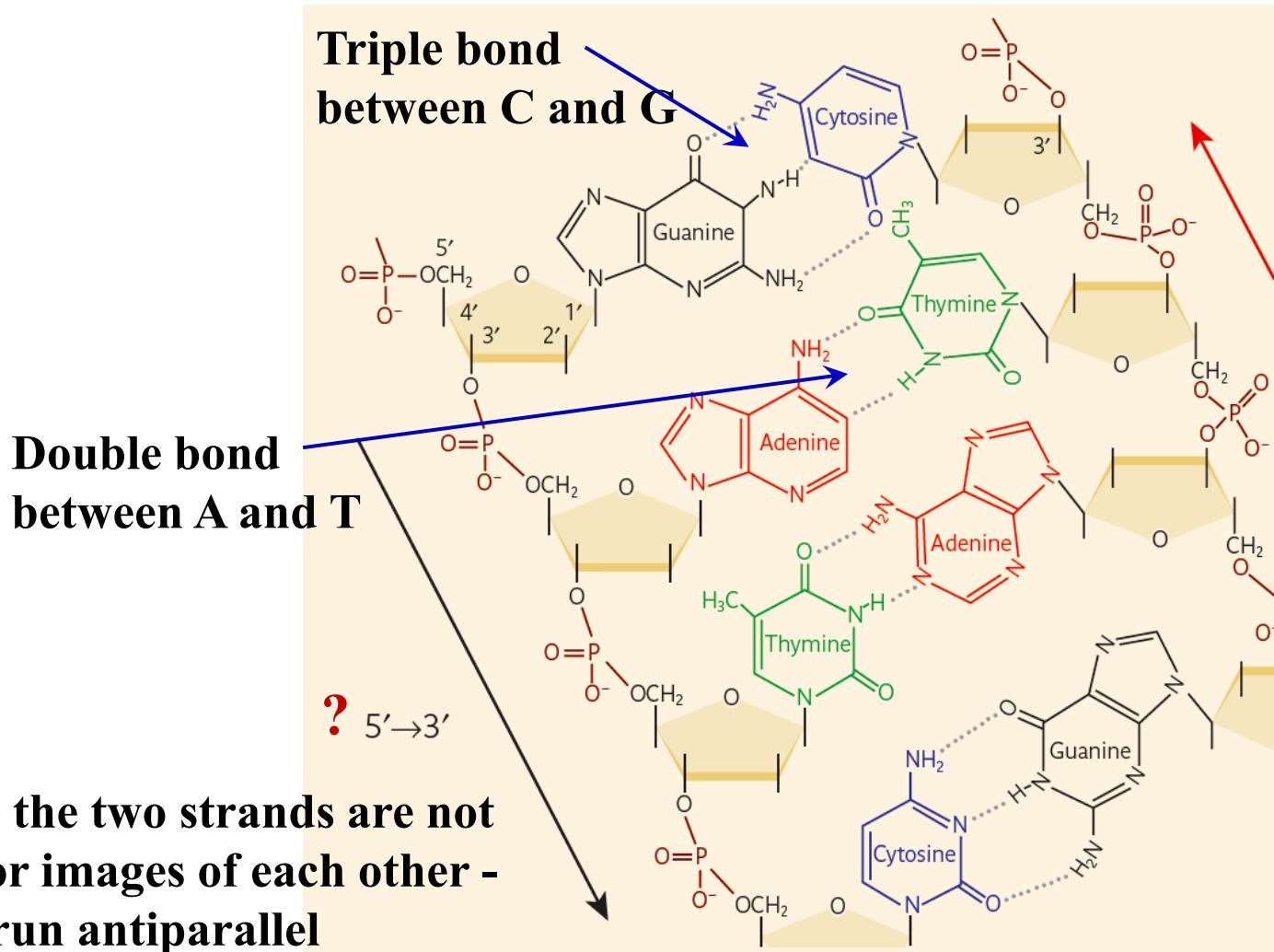
- Facilitates DNA replication
- Error-correct the genome
- Provides stability

From a computational perspective, the sequence of only one strand is needed.



# DNA

**Base Pairing:** If two polynucleotide strands face each other, sugar-phosphate backbone runs down each side, and complementary pairs of bases in the middle spontaneously form hydrogen bonds:



**Double-Stranded DNA:** If the sequence in the forward strand in **5' to 3'** direction is:

**5' CATTGCCAGT 3'**

Then what is the sequence on the reverse strand when read in **5' to 3'** orientation?

# DNA

If the sequence in the forward strand in **5' to 3'** direction is:

**5' CATTGCCAGT 3'**

Then what is the sequence on the reverse strand when read in **5' to 3'** orientation?

First write its complement:

**5' CATTGCCAGT 3'**

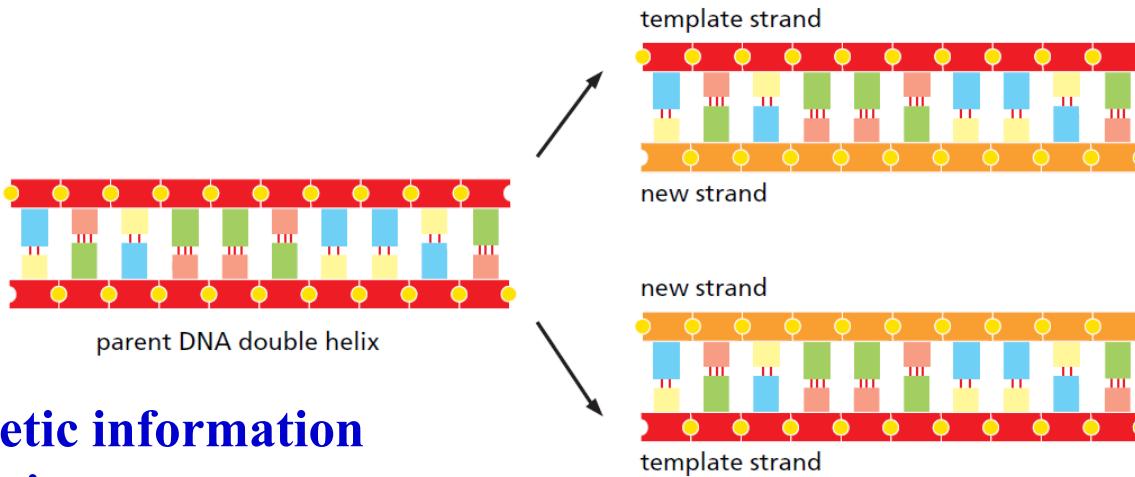
**3' GTAACGGTCA 5'**

When read in **5' to 3'** orientation, the sequence on the reverse strand is:

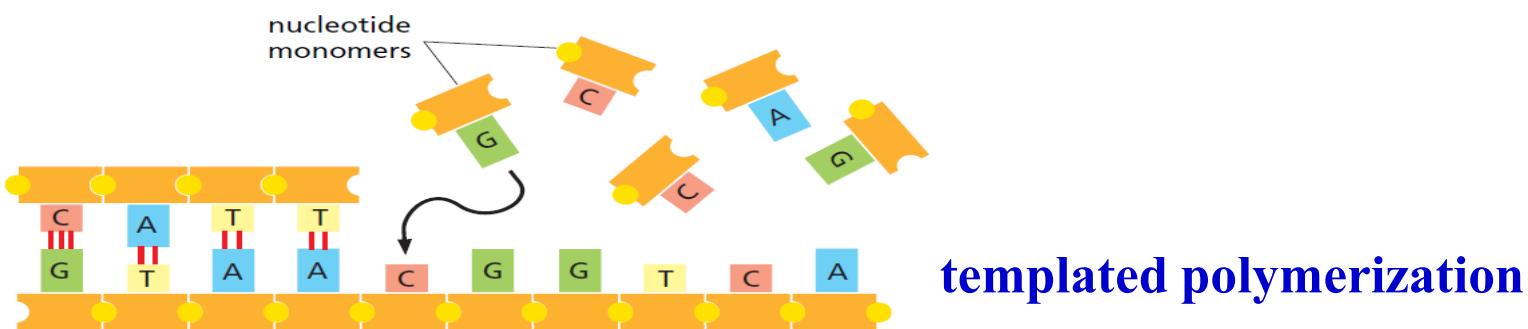
**5' ACTGGCAATG 3'**

# DNA Replication

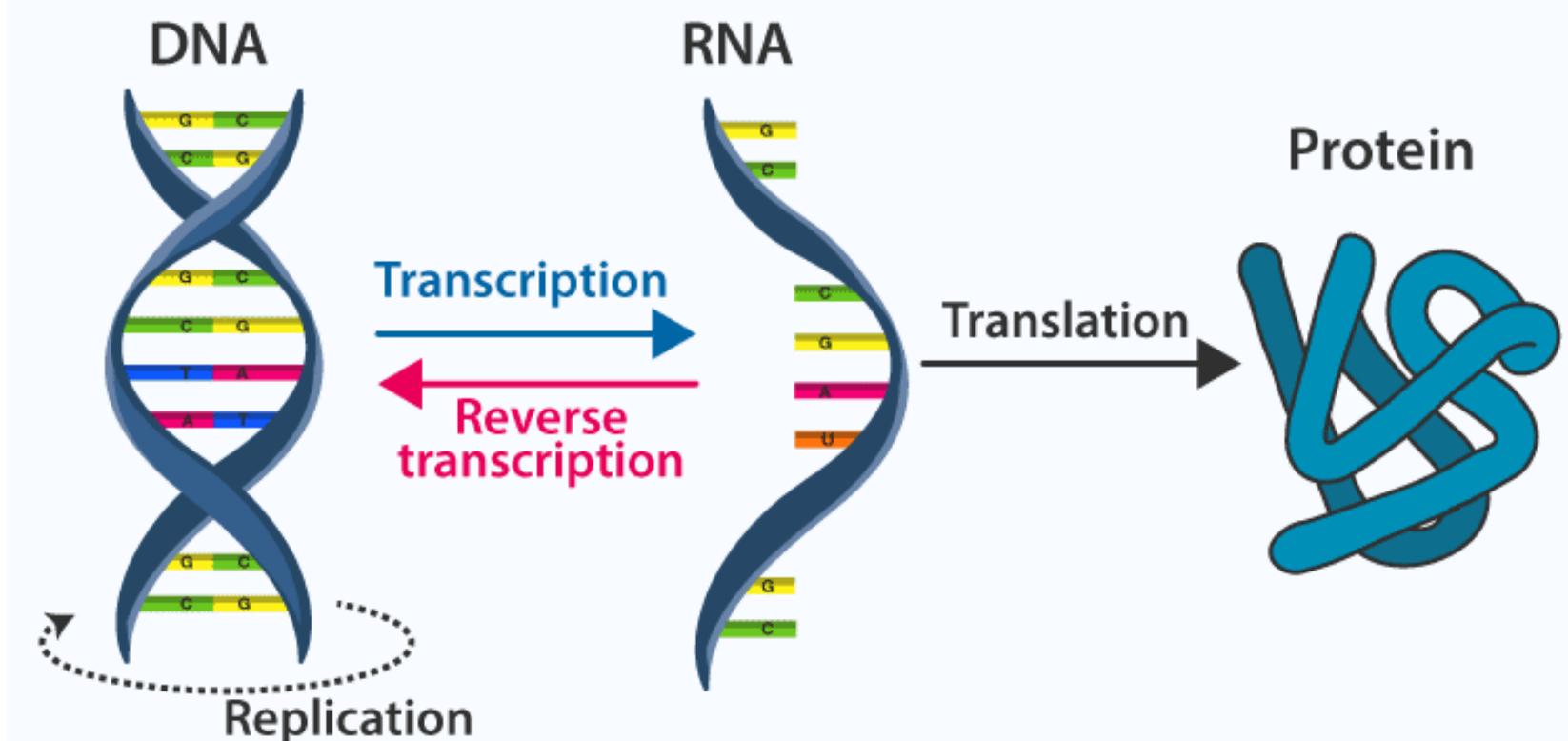
In living cells DNA is not synthesized as a free strand in isolation, but on a template formed by a pre-existing DNA strand.



Copying of genetic information  
by DNA replication



# Central Dogma of Molecular Biology



## Ribonucleic Acid (RNA):

It is **single-stranded** molecule

Composed of four basic units - called **nucleotides**:

Each nucleotide contains - a sugar (ribose), a phosphate and one of the four bases: Adenine (A), **Uracil (U)**, Guanine (G), Cytosine (C)

RNA polynucleotide strand is built by creating a **phosphodiester bond** between nucleotides.

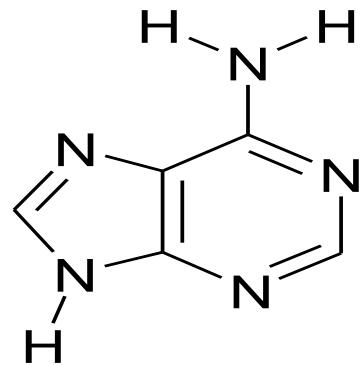
**Intra-strand base pairing** is a characteristic feature of RNA

Base Pairing – formed by weak H-bonds and follows the following complementarity rule:

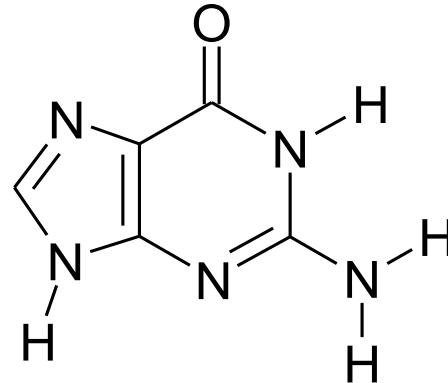
G  $\longleftrightarrow$  C,     A  $\longleftrightarrow$  U, and     G  $\longleftrightarrow$  U

# Ring Structure of Nucleic Acid bases

**Adenine (A)**

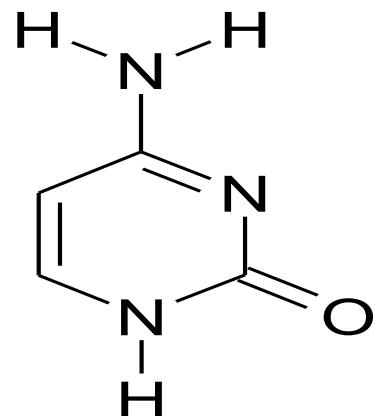


**Guanine (G)**



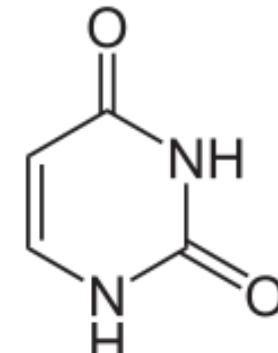
**Purines**

**Cytosine (C)**



**Pyrimidines**

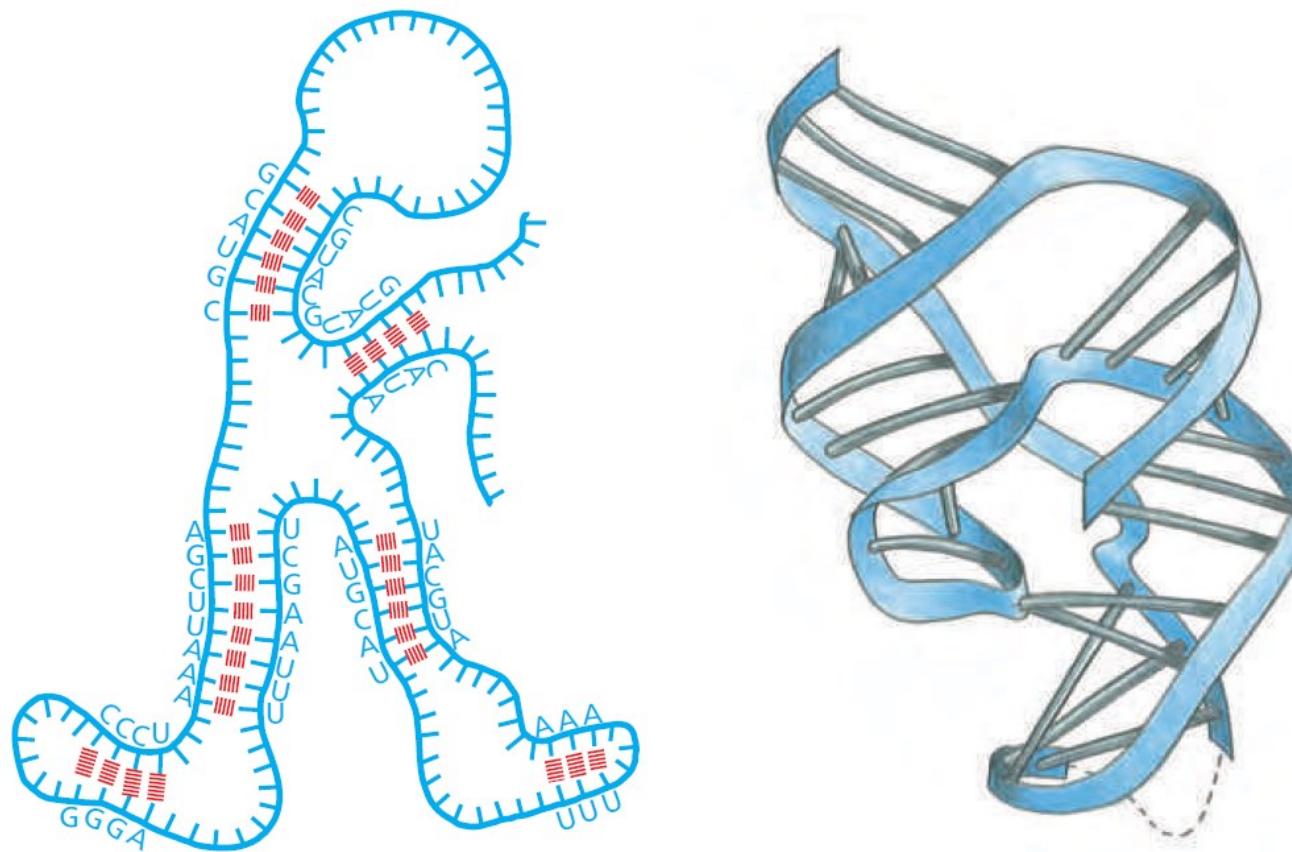
**Uracil (U)**



**Note: No CH3 in Uracil as in Thymine**

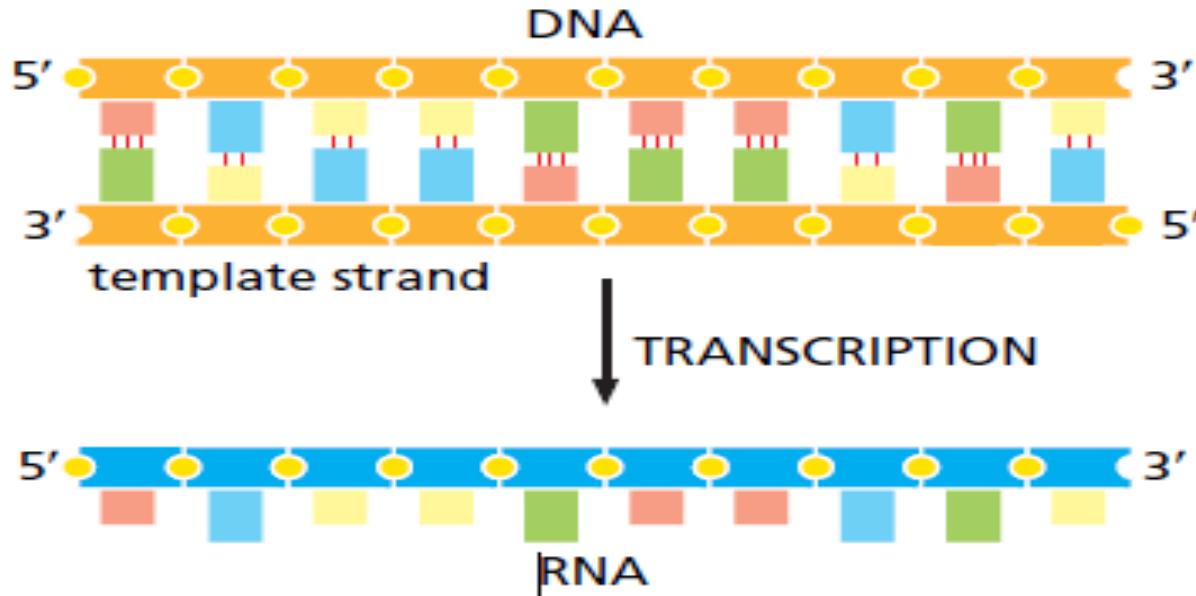
# RNA

**Nucleotide pairing between different regions of the RNA polymer chain causes the molecule to adopt a distinctive shape - enables it to recognize other molecules by selective binding, or, catalyze chemical changes in the molecules that are bound.**



## RNA Synthesis:

RNA is also read in the 5' to 3' orientation.



RNA molecules that are copied from the genes (which ultimately direct the synthesis of proteins) are called messenger RNA (mRNA) molecules.

# RNA Synthesis

1. If the following DNA sequence is the **forward strand**:

5' CATTGCCAGT 3'

What will be the sequence of the RNA strand synthesized?

2. If the following DNA sequence is used as **template** for RNA synthesis:

5' CATTGCCAGT 3'

Give the sequence of the RNA strand read in 5' to 3' orientation.

# RNA Synthesis

1. If the DNA sequence in the **forward strand** is given:

**5' CATTGCCAGT 3'**

Template sequence used for RNA synthesis is its complement:

**5' CATTGCCAGT 3'**

**3' GTAACGGTCA 5'** template

The synthesized RNA sequence is the reverse complement of the template:

**3' GTAACGGTCA 5'** template

**5' CAUUGGCCAGU 3'** RNA

- i.e., synthesized RNA sequence is basically the DNA sequence in the forward strand with T replaced by U

# RNA Synthesis

If the following DNA sequence is used as template for RNA synthesis:

5' CATTGCCAGT 3'

First write its complement:

5' CATTGCCAGT 3'

3' GUAACGGUCA 5' complement

Then the synthesized RNA sequence in 5' to 3' orientation is:

5' ACUGGCAAUG 3' RNA

- i.e., synthesized RNA sequence is basically the complement of the template DNA sequence with T replaced by U, when read in the 5' to 3' orientation

## RNA Synthesis:

There are other RNA molecules also obtained from genes. The final product in such cases is RNA.

- these are known as **noncoding RNAs** because they do not code for protein.

e.g., in yeast *Saccharomyces cerevisiae*, over 1200 genes (~15%) produce RNA as their final product; Humans may produce on the order of 10,000 noncoding RNAs.

These RNAs, like proteins, serve as enzymatic, structural, and regulatory components for a wide variety of processes in the cell.

TABLE 6-1 Principal Types of RNAs Produced in Cells

Type of RNA	Function
mRNAs	Messenger RNAs, code for proteins
rRNAs	Ribosomal RNAs, form the basic structure of the ribosome and catalyze protein synthesis
tRNAs	Transfer RNAs, central to protein synthesis as adaptors between mRNA and amino acids
snRNAs	Small nuclear RNAs, function in a variety of nuclear processes, including the splicing of pre-mRNA
snoRNAs	Small nucleolar RNAs, help to process and chemically modify rRNAs
miRNAs	MicroRNAs, regulate gene expression by blocking translation of specific mRNAs and cause their degradation
siRNAs	Small interfering RNAs, turn off gene expression by directing the degradation of selective mRNAs and the establishment of compact chromatin structures
piRNAs	Piwi-interacting RNAs, bind to piwi proteins and protect the germ line from transposable elements
lncRNAs	Long noncoding RNAs, many of which serve as scaffolds; they regulate diverse cell processes, including X-chromosome inactivation

**Note: rRNA, tRNA and snRNA play an important role in protein synthesis**

# Protein Synthesis

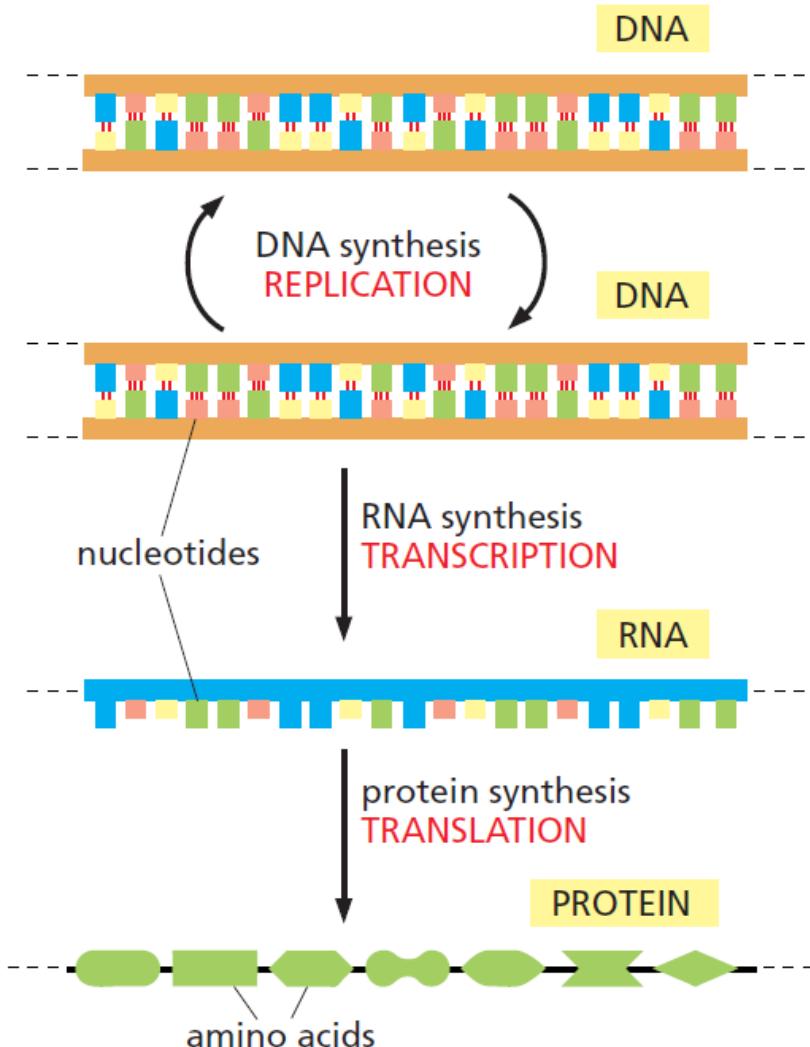
Proteins are synthesized from DNA in a two-step process:

Each chromosome has several genes that code for various traits in the body.

- from enzymes to the color of eye

RNA molecules direct synthesis of proteins in a complex process called translation.

- information in mRNA is read out in groups of three nucleotides, called codons.



# The Genetic Code

		Second letter					
		U	C	A	G		
		UUU UUC UUA UUG	UCU UCC UCA UCG	UAU UAC UAA UAG	UGU UGC UGA UGG	C A G	
First letter	U	Phe	Ser	Tyr Stop Stop	Cys Stop		
	C	Leu			Trp		
	A	Leu	Pro	His Gln	Arg		
	G	AUU AUC AUA AUG Met	Thr	Asn Lys	Ser Arg		
Start codon						Third letter	
		Val	Ala	Asp Glu	Gly		

The genetic code is degenerate

# Protein Synthesis

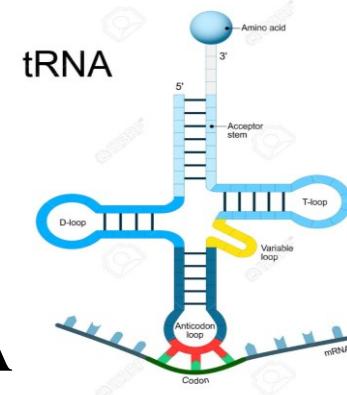
Using the genetic code, the amino acid sequence synthesized from the following mRNA sequence is:

5' ACU GGC AAU 3'

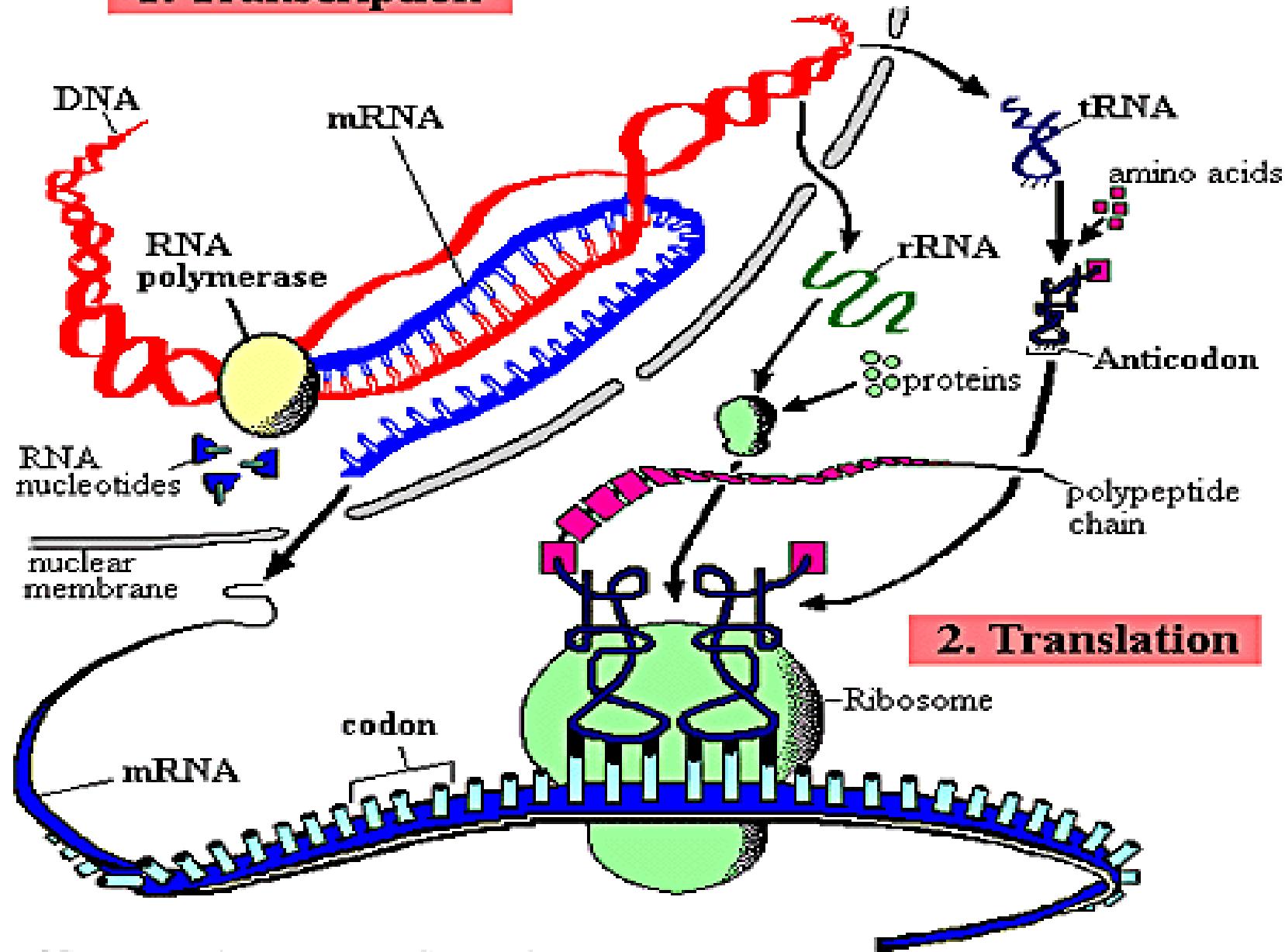
Thr    Gly    Asn

This genetic code is read out by a class of small RNA molecules, called **transfer RNAs (tRNAs)**.

- each type of tRNA attaches at one end a specific amino acid and at its other end has a specific sequence of 3 nucleotides, an **anticodon** that enables it to recognize, through base-pairing, a particular codon in the mRNA sequence.
- This process occurs on **ribosome**, a large multi-molecular machine composed of both proteins and ribosomal RNA.



## 1. Transcription



## 2. Translation

Protein synthesis

# Proteins

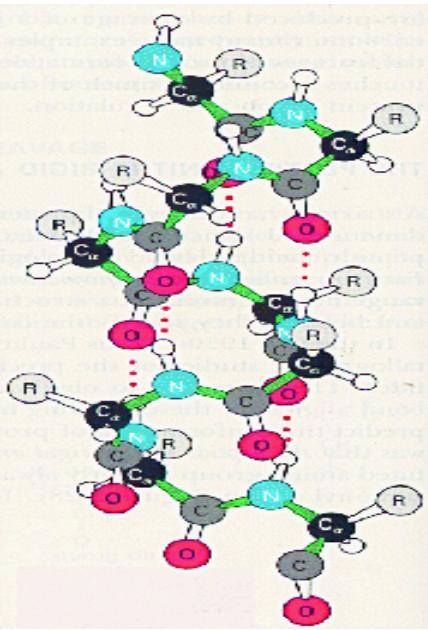
Like DNA and RNA, Proteins carry information in linear sequence on a 20-letter alphabet, called **amino acids**:

ATRVGTCWPRA

Protein structure is divided in 4 hierarchical levels:

- **Primary structure** - represented by AA sequences
- **Secondary structure** -  $\alpha$ -helices &  $\beta$ -sheets
- **Tertiary and Quaternary structures** - represented by 3D structures

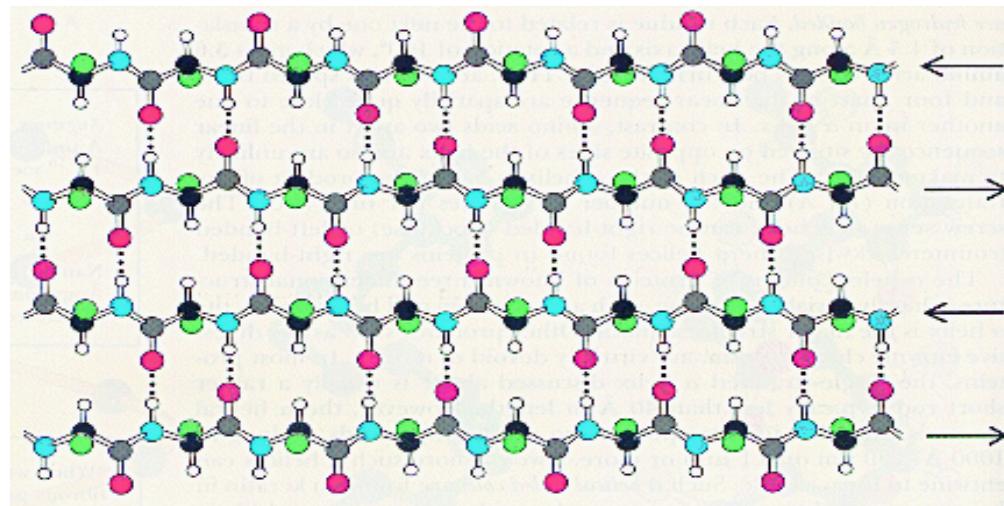
# Primary Structure: ATRVGTCWPRA



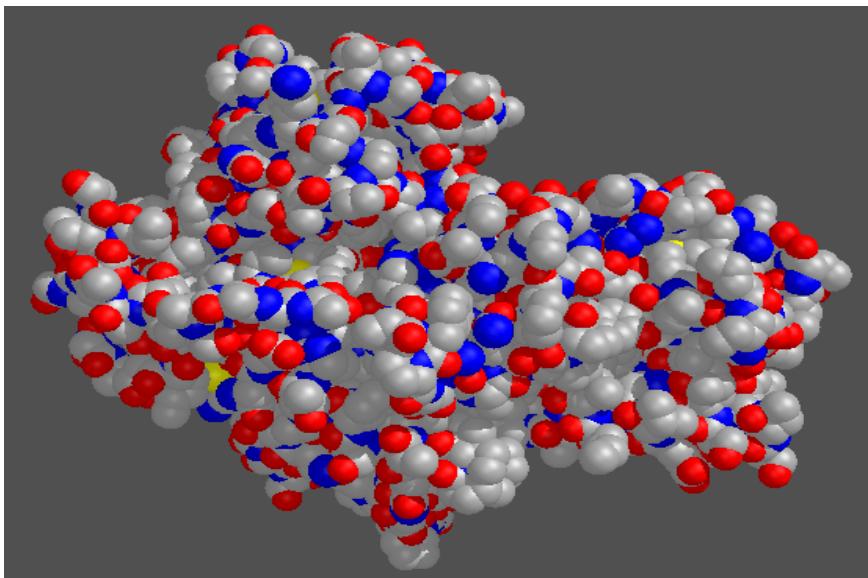
$\alpha$ -helix

Secondary  
Structures

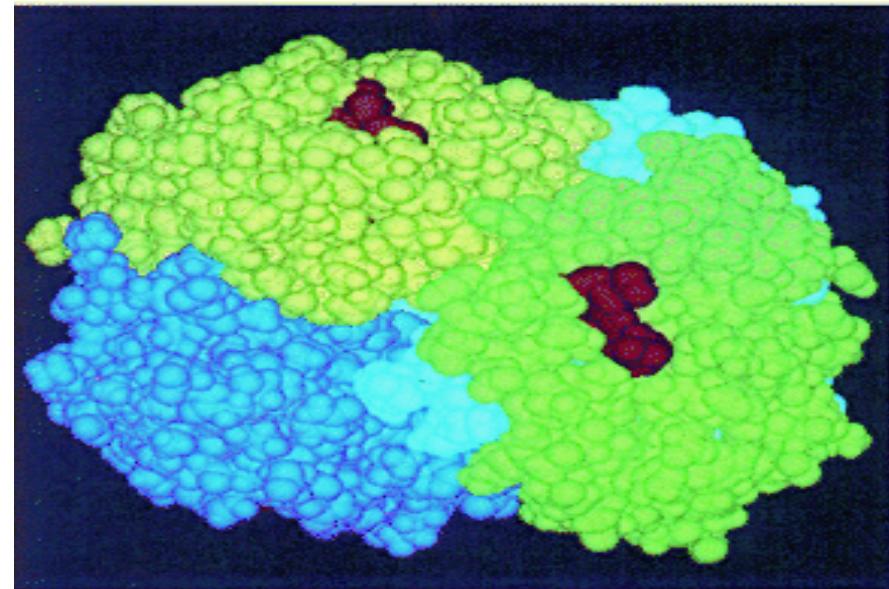
$\beta$ -sheets



Tertiary Structure



Quaternary Structure



# Function of Proteins

- Proteins make up much of the **cellular structure** – hair, skin, fingernails, etc.
- **Enzymes** – catalyze chemical reactions within the cell
- **Transcription factors** – regulate the manner in which genes direct production of other proteins
- **Receptors** – proteins on the surface of cells act as receptors for hormones and other signaling molecules
- **Recognize and bind** to Nucleic acids (DNA, RNA) and Proteins – to carry out their functions in the cell

# Genes

Special sequences in the DNA code for **genes**:

- **Protein-coding genes**, for which the final product is a protein.
  - same gene may give rise to more than one protein (~ 6 per gene in humans).
- **Non-coding RNA genes** - for which the final product is RNA

**Genotype** – An organism's genotype is the set of **genes** that it carries.

**Phenotype** – An organism's phenotype is all of its **observable characteristics** which are influenced both by its genotype and by the environment e.g., height, hair colour, levels of hormones, etc.

# Differences in the genotypes can produce different phenotypes

Genes for ear form are different, causing one of the cats to have normal ears and the other to have curled ears



Change in environment can also affect the phenotype. Pinkness is not encoded in the genotype of flamingos - the food they eat makes their phenotype white or pink - a natural pink dye, canthaxanthin, obtained from their diet of brine shrimp and blue-green algae



# Genes

The biological function of a gene is to preserve and express the genetic information encoded within it

Genes are normally very **stable entities**

Genetic stability is not **absolute**, however.

Genes may occasionally become **altered**; these changes called **mutations** create new **alleles**.

Mutant genes are also **stable entities** and are inherited in the same way as normal, wild-type genes.

# Genes

Normal diploid cells such as somatic cells of humans contain **two** sets of genes – one set inherited from each parent.

- corresponding genes derived from each parent are called **alleles**.

Together the two alleles govern the **phenotype** of an organism.

**What is the percentage of genes in a genome?**

# Genes

**Gene-fraction varies from ~70% in prokaryotes to ~2 - 3% in humans**

- does this imply prokaryotes have more gene content than eukaryotes?
- Size of a prokaryotic genome? Eukaryotic genome?

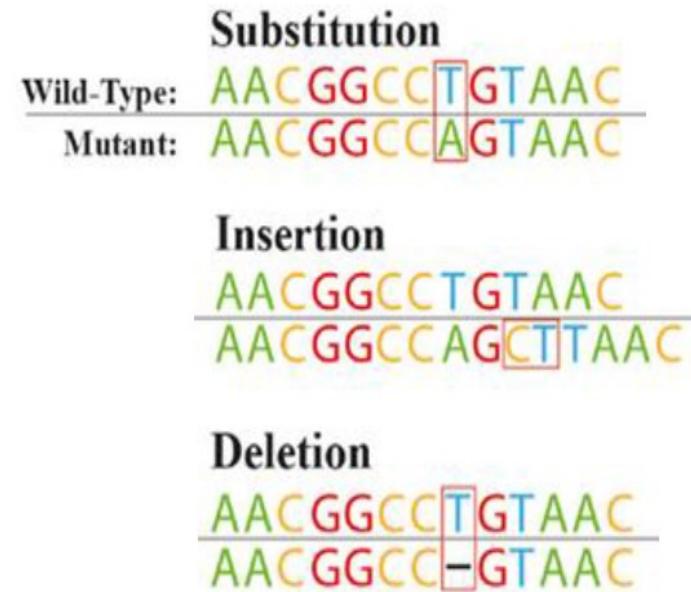
**What's the function of remaining ~97-98% of human genome?**

**The remaining part of the genome consists of noncoding regions, whose functions may include providing chromosomal structural integrity and regulating where, when, and in what quantity proteins are made, and repeats.**

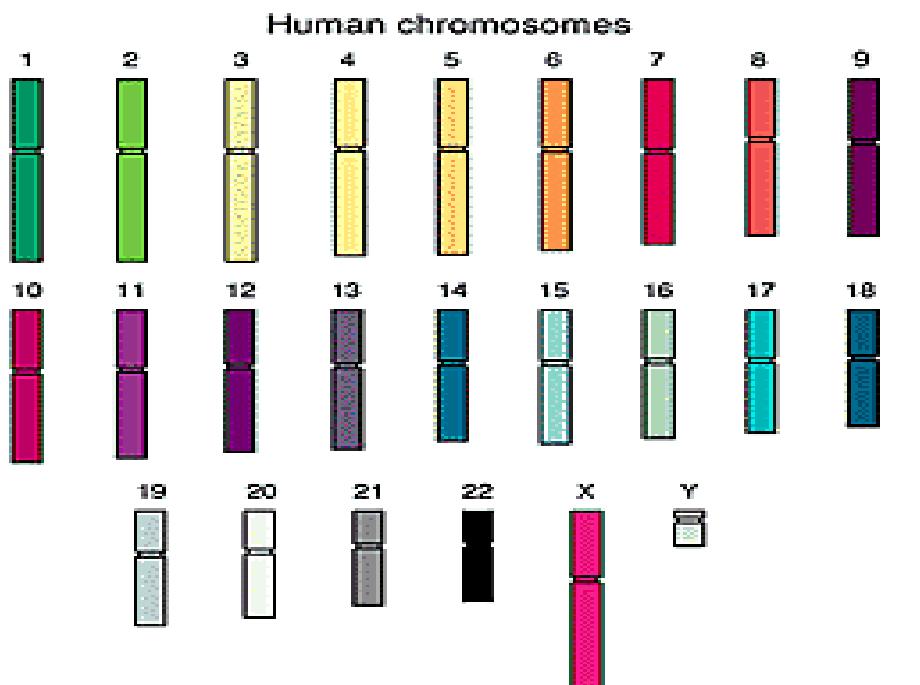
# Mutations

**Mutations** - are local changes in the DNA content, caused by inexact replication and are of various kinds:

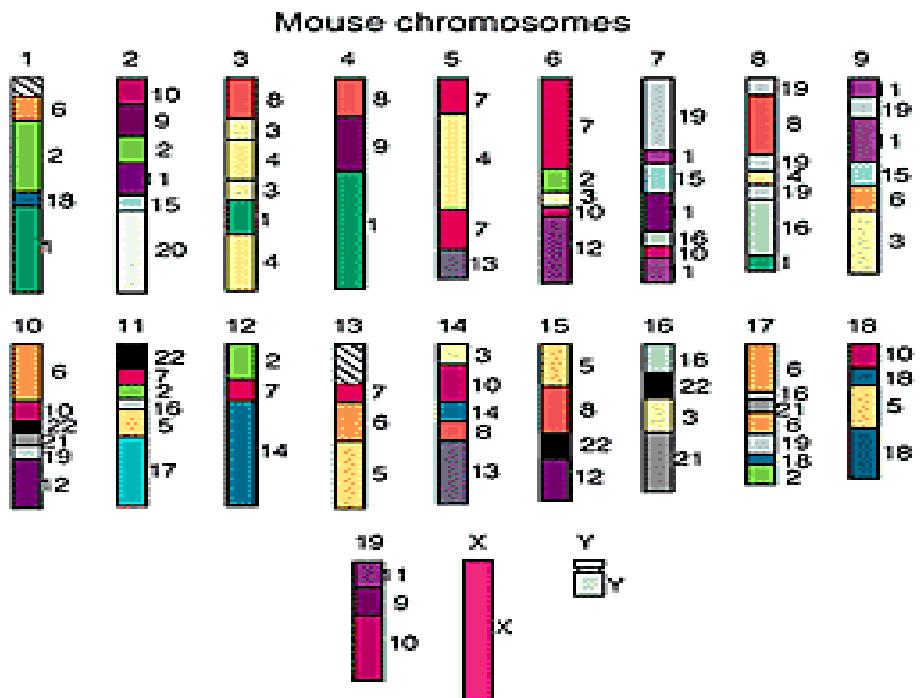
- **Substitution** - a base is replaced by another - may or may not alter the protein sequence depending on the place it occurs.
- **Insertion/Deletion** – addition/removal of one or more bases – results in a frame-shift in coding regions.
- **Rearrangement** - a change in the order of complete segments along a chromosome.



**Chromosomal rearrangements occur both within and between chromosomes during evolution**



**The colors on the mouse chromosomes and the numbers alongside indicate the human chromosomes containing homologous segments.**



# Mutations

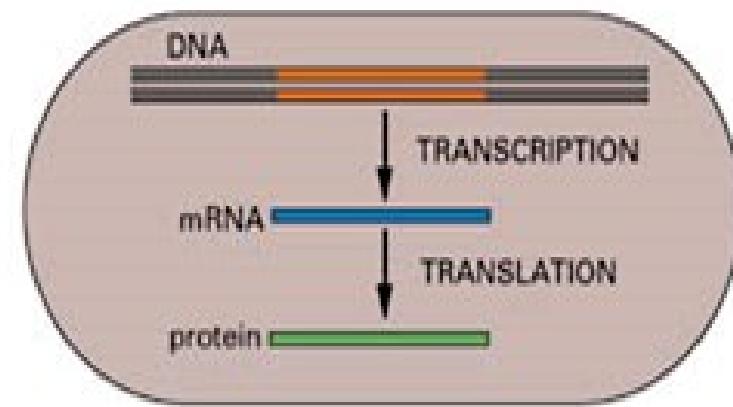
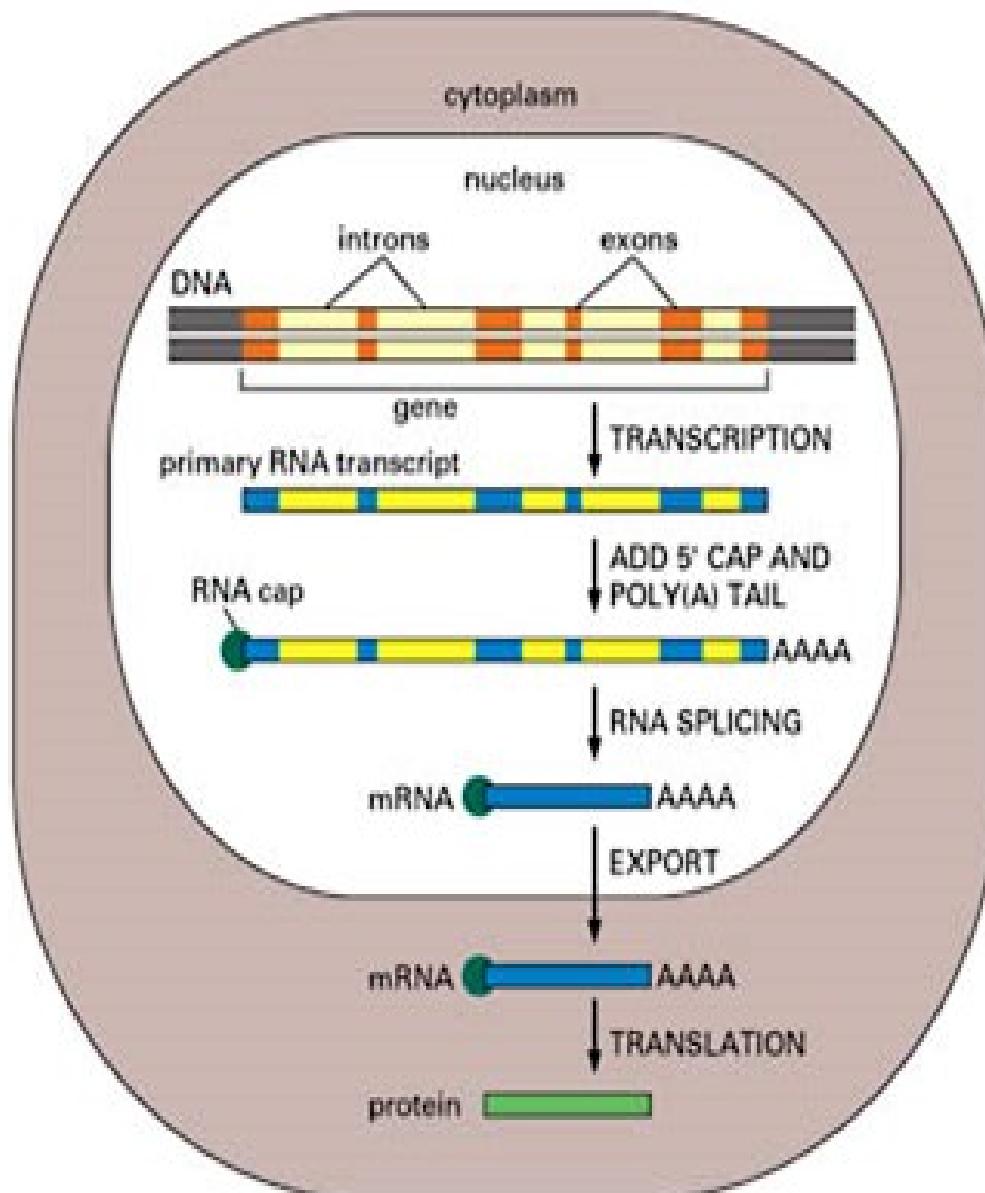
## Role of Mutations:

- Mutations are the source of **phenotypic variation** on which natural selection acts, creating species & changing them.  
e.g., the human and mouse genome are very similar – major difference being the **internal order** of DNA segments.

**Without mutations there wouldn't be any evolution!**

- They are responsible for **inherited disorders and diseases**, which involve alterations in gene.

# Steps Leading from Gene to Protein

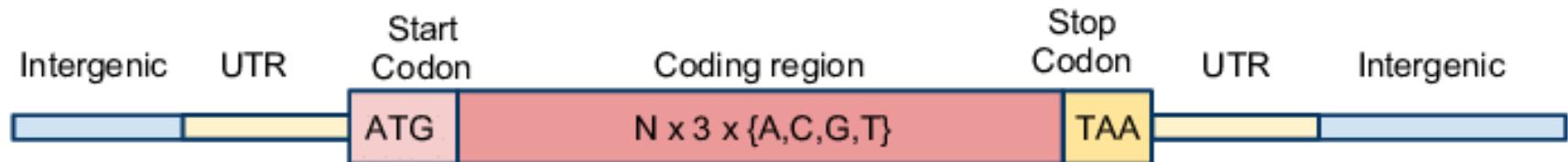


**Prokaryotes**

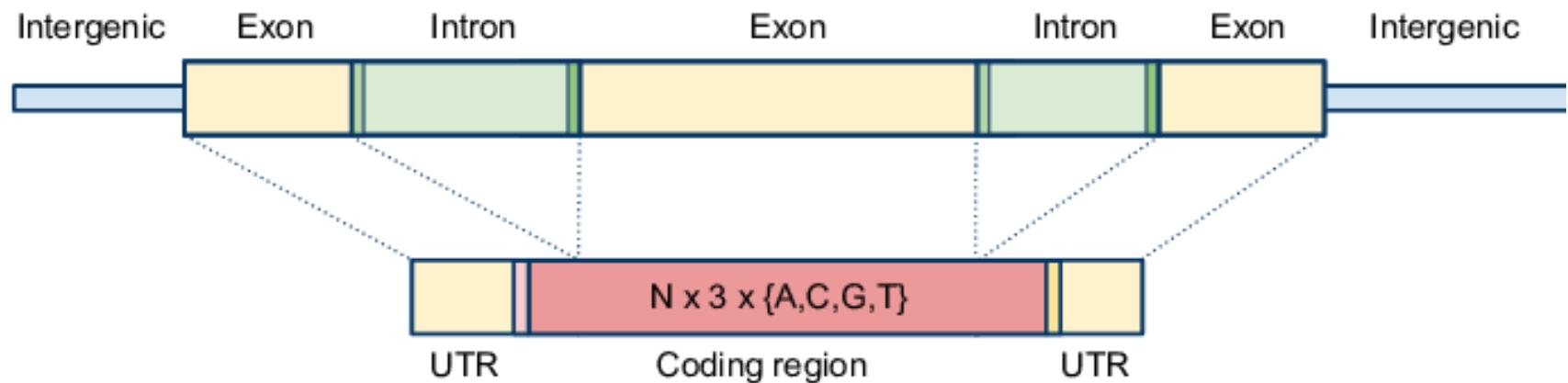
**Eukaryotes**

# Gene Structure

## A) Prokaryotic Gene

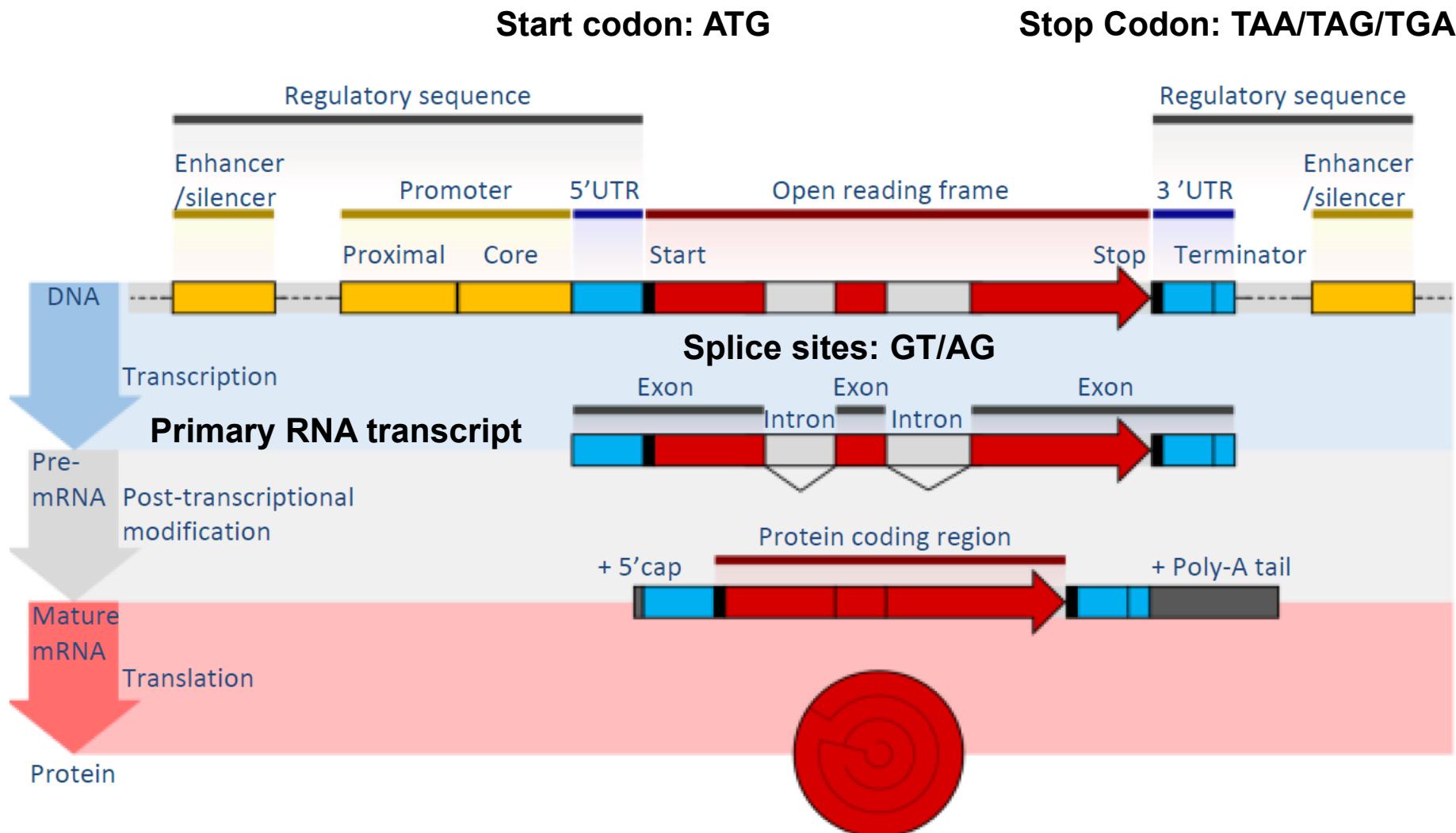


## B) Eukaryotic Gene



**UTRs – Untranslated Regions – are transcribed, but not translated**

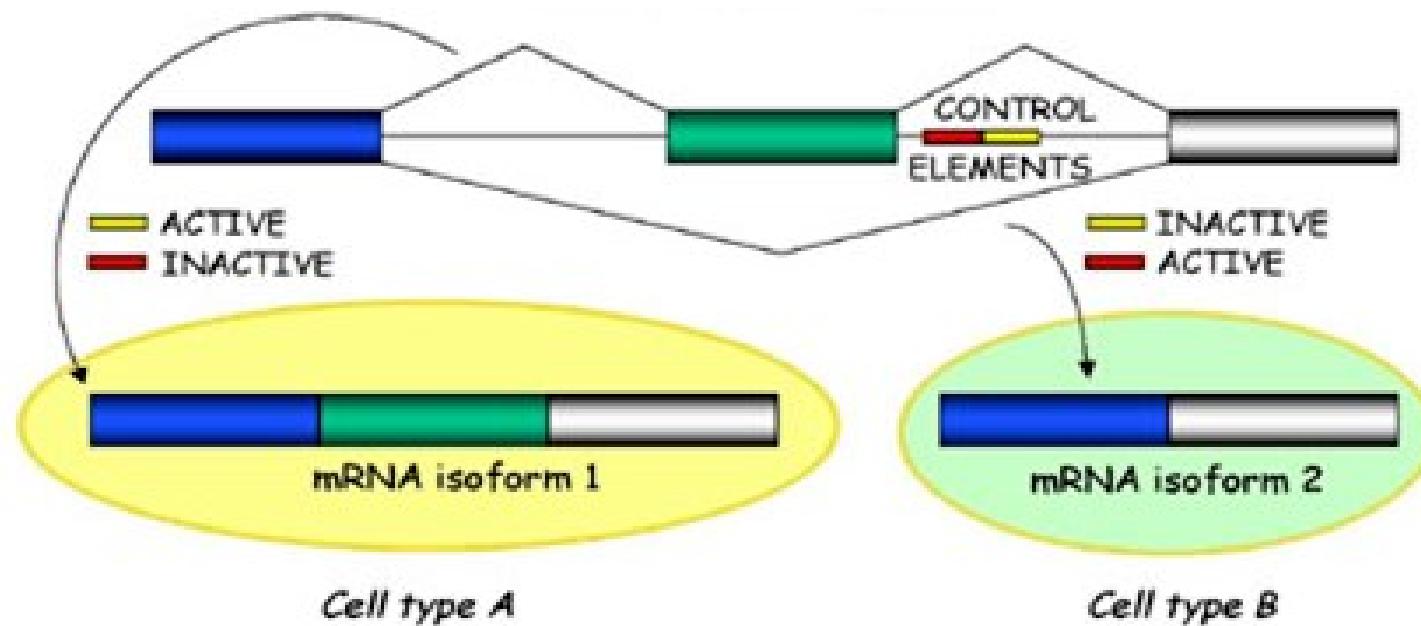
# Eukaryote Gene Structure



Transcription is initiated only at certain specific positions in the sequence, signaling the beginning of genes, called **promoters**.

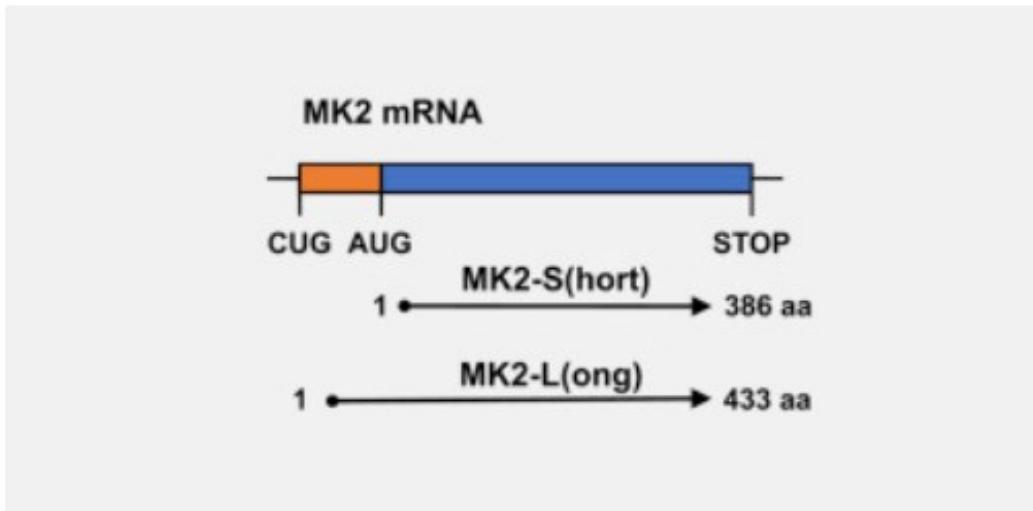
# Alternative Splicing

- In many cases, the pattern of splicing can vary depending on the tissue in which the transcription occurs.  
e.g., an exon maybe spliced in the gene transcribed in liver **but retained** when transcribed in the brain.
- This variation called **alternative splicing**, contributes to the overall protein diversity in the organism



# Alternative Initiation

- Another type of variation that contributes to protein diversity is **alternative initiation**



- **Alternative translation** is an important mechanism of post-transcriptional gene regulation leading to the expression of different protein isoforms originating from the same mRNA

# Data Representation

**DNA - a complex, dynamic, three-dimensional molecule represented as a string of alphabets**



**- a perfect representation for computer analysis**

**Aim:** to find grammar & syntax rules of DNA language based on this 4-letter alphabet

**- similar to English Grammar to form meaningful sentences**

# Biological Sequence Analysis

## Pattern Recognition:

**Assumption in biological sequence analysis:**

**- strings carrying information will be different from random strings**

**If a hidden pattern can be identified in a string, it must be carrying some functional information**

# Biological Sequence Analysis

**Order of occurrence of bases:  
not completely random**

- Different regions of the genome exhibit different patterns of the four bases, A, T, G, C

e.g., protein coding regions, regulatory regions, intron/exon boundaries, repeat regions, etc.

**Aim:** Identify various patterns to infer their functional roles

# Example

# This is a lecture on bioinformatics

asjd lkjfl jdjd sjftye nvcrow nzcdjhspu

# Frequency of letters

- |          |         |
|----------|---------|
| A. 7.3%  | N. 7.8% |
| B. 0.9%  | O. 7.4% |
| C. 3.0%  | P. 2.7% |
| D. 4.4%  | Q. 0.3% |
| E. 13.0% | R. 7.7% |
| F. 2.8%  | S. 6.3% |
| G. 1.6%  | T. 9.3% |
| H. 3.5%  | U. 2.7% |
| I. 7.4%  | V. 1.3% |
| J. 0.2%  | W. 1.6% |
| K. 0.3%  | X. 0.5% |
| L. 3.5%  | Y. 1.9% |
| M. 2.5%  | Z. 0.1% |

# Other statistics

**Frequencies of the most common first letter of a word, last letter of a word, doublets, triplets, etc.**

## **20 most used words in written English**

**- the of to in and a for was is that on at he with by be it an as his**

## **20 most used words in spoken English**

**- the and I to of a you that in it is yes was this but on well he have for**

# Parallels in DNA language

ATGGTGGTCATGGCGCCCCGAACCCCTCTTCCTGCTG  
CTCTCGGGGGCCCTGACCCCTGACCGAGACCTGGGCG  
GGTGAGTGCAGGGTCAGGAGGGAAACAGCCCCCTGC  
GCGGAGGGAGGGAGGGGCCGGCCGGCGGG

GTCTCAACCCCTCCTCGCCCCCAGGCTCCACTCCA  
TGAGGTATTCAGCGCCGCCGTGTCCCAGGCCCCGGCC  
GCGGGGAGCCCCGCTTCATGCCATGGGCTACGTGG  
ACGACACGCAGTTCGTGCAGGTTC

# Parallels in DNA language

**ATG GTG GTC ATG GCG CCC CGA ACC CTC TTC**  
**CTG CTG CTC TCG GGG GCC CTG ACC CTG ACC**  
**GAG ACC TGG GCG GGT GAG TGC GGG GTC AGG**  
**AGG GAA ACA GCC CCT GCG CGG AGG AGG GAG**  
**GGG CCG GCC CGG CGG...**

**GTC TCA ACC CCT CCT CGC CCC CAG GCT CCC ACT**  
**CCA TGA GGT ATT TCA GCG CCG CCG TGT CCC**  
**GGC CCG GCC GCG GGG AGC CCC GCT TCA TCG**  
**CCA TGG GCT ACG TGG ACG ACA CGC AGT TCG**  
**TGC GGT TC...**

**1<sup>st</sup> exon and 1<sup>st</sup> intron of Human HLA gene**

**This task needs to be automated because of the large genome sizes:**

**Smallest genome:**

**Mycoplasma genitalium  $0.5 \times 10^6$  bp**

**Human genome:  $3 \times 10^9$  bp – not the largest!**

**~ 10-100 times the Britannica Encyclopedia**

**Plant genomes are even larger.**

# DNA Sequence Analysis

- Evolution has operated on every sequence that we see today
  - genes and sequences involved in gene regulation are **conserved**.
- these are transferred, like code modules, from one organism to another. Because of evolution, similar sequences have similar functions.
- Algorithms for comparing sequences and finding similar regions are at the heart of computational biology.

# **Syllabus**

**Unit 1: Overview – Bioinformatics, Gene & Genome structure**

**Gene Technology – Restriction Endonucleases, Cloning vectors**

**DNA sequencing – PCR, cDNA and Whole Genome sequencing, NGS and third generation sequencing technologies**

## **Unit 2: BioDatabases**

- **Major Bioinformatics Resources – NCBI, EBI, PubMed,**
- **Primary Nucleotide and Proteins Databases - GenBank, UniProt, PDB,**
- **Genome Browsers – Ensembl, UCSC**
- **k-mer analysis and their significance in biological sequences**

# Syllabus

## Unit 3: Sequence Alignment:

- **Pairwise Alignment** – Types of pairwise alignments – Global, Local and Overlap alignments, Dot Plots, dynamic programming (DP) algorithm,
- **Scoring matrices for nucleotides and proteins and gap penalties,**
- **Sequence-based Database Search algorithms** – BLAST, FASTA,
- **Multiple Alignment, Algorithms for Global and Local MSA** – DP, Progressive based (ClustalX), Iterative methods, Motif search-based methods

# Syllabus

## **Unit 4: Modeling Molecular Evolution – Phylogeny:**

- **Markov models of base substitution,**
- **Computing Phylogenetic Distances,**
- **Phylogenetic Tree Construction Methods, PHYLIP**

## **Unit 5: Gene Prediction:**

### **Gene Prediction approaches –**

- **Open Reading Frames,**
- **Homology search,**
- **Content-based methods,**
- **Markov models**

# Gene Technology

For all computational purposes, DNA is represented as a string of 4-letter alphabets - A, T, C, G:

attgctacgttacatcgctgca

How do we get this string representation from a dynamic double-stranded molecule?

**DNA Sequencing** - determine the precise sequence of nucleotides in a sample of DNA

To carry out this task we need to be able to chop the DNA, store it, make copies of it.

Let's consider the example of detecting if a person is infected by the novel coronavirus SARS-CoV-2

- uses Real Time RT-PCR Nucleic Acid Detection Kit based on the PCR method which uses a fluorescent probe and a specific primer to detect three specific regions within the SARS-CoV-2 nucleocapsid protein N gene.
- How is the SARS-CoV-2 genome sequenced?
- How does one identify the coordinates of N gene on it? i.e., how to construct a physical map of a genome?
- How does one select which regions in this gene would give specificity for the presence of SARS-CoV-2?\*
- How are the specific probe regions extracted and amplified for detection?
- Is it possible to store the DNA sample for re-testing? How?

To sequence a gene, we need to

- Identifying the **region of interest**
- Isolate it from the organism – **DNA fragmentation**
- moving it to another easily manageable organism such as a bacterium for obtaining multiple copies – **cloning**

Such manipulations are conducted by a toolkit of enzymes:

**Restriction endonucleases** - used as molecular scissors

**DNA ligase** - to bond pieces of DNA together

- a variety of additional enzymes that modify DNA are used to facilitate the process.

**Restriction endonucleases** are enzymes that make **site-specific** cuts in the DNA – **chemical scissors**

Ability to cut DNA into discrete fragments allows to understand

- how genetic material of an organism is **organized**
- how expression of genetic information is **controlled**
- how **alteration** of genetic information can give rise to genetically inherited disorders, etc.
- in **bulk production** of pharmaceutically important proteins

First restriction enzyme was isolated from *H. influenzae* in 1970 by Daniel Nathans and Kathleen Danna  
- awarded the Nobel Prize for Medicine in 1978

**Restriction endonucleases** are enzymes that make **site-specific** cuts in the DNA – **chemical scissors**

First restriction enzyme was isolated from *H. influenzae* and used to cleave SV40 DNA ( a tumor virus):



- 11 distinct DNA bands were visible on polyacrylamide gel electrophoresis, indicating that the enzyme always cut SV40 resulting in the same 11 pieces

# Background

## How were these restriction endonucleases identified?

Bacteria are under constant attack by bacteriophages – a virus that infects and replicates within a bacterium

To protect themselves, bacteria have developed a method to chop up any foreign DNA - such as that of an attacking phage

These bacteria build an **endonuclease** - an enzyme that cuts DNA - it circulates in the bacterial cytoplasm, waiting for phage DNA.

These endonucleases are termed “restriction enzymes” because they **restrict** the infection of bacteriophages.

## Why the restriction enzymes do not chew up the genomic DNA of their host?

# Background

A bacterium that makes a particular restriction endonuclease, also synthesizes a companion **DNA methyltransferase**,

- which methylates the DNA target sequence for that restriction enzyme, thereby protecting it from cleavage.

DNA from an attacking bacteriophage will not have these protective methyl groups and will be destroyed.

Methyl groups block the binding of restriction enzymes, but do not block the normal reading and replication of the genomic information stored in the host DNA.

# DNA Fragmentation

Different endonucleases present in different bacteria recognize **different** nucleotide sequences

Naming of restriction enzymes - after their host of origin, e.g.,

- EcoRI - *Escherichia coli*
- Hind II & Hind III - *Haemophilus influenzae*
- XhoI - *Xanthomonas holcicola*

When cut with a restriction enzyme (RE), the ends of the cut DNA fragment can be **cohesive or blunt-ended** depending on the enzyme.

Enzyme	Recognition Sequence
EcoRI	G <sup>↓</sup> AATTC
HindIII	A <sup>↓</sup> AGCTT
BamHI	G <sup>↓</sup> GATCC
BglII	GCCNNNN <sup>↓</sup> NGGC
PvuI	CGATC <sup>↓</sup> G
HaeIII	GG <sup>↓</sup> CC
MboI	GAT <sup>↓</sup> C

# Generation of Cohesive & Blunt-ended Fragments

Cutting with Eco R I

5'... G ↓ AATTC... 3'  
3'... CTTAA ↑ G ... 5'

5'... G

AATTC... 3'

3'... CTTAA

G... 5'

Cohesive or  
“Sticky” Ends

(a)

Cutting with Pst I

5'... CTGCA ↓ G... 3'  
...G ↑ ACGTC... 5'

5'... CTGCA

G... 3'

3'... G

ACGTC... 5'

Cohesive or  
“Sticky” Ends

Cutting with Sma I

↓

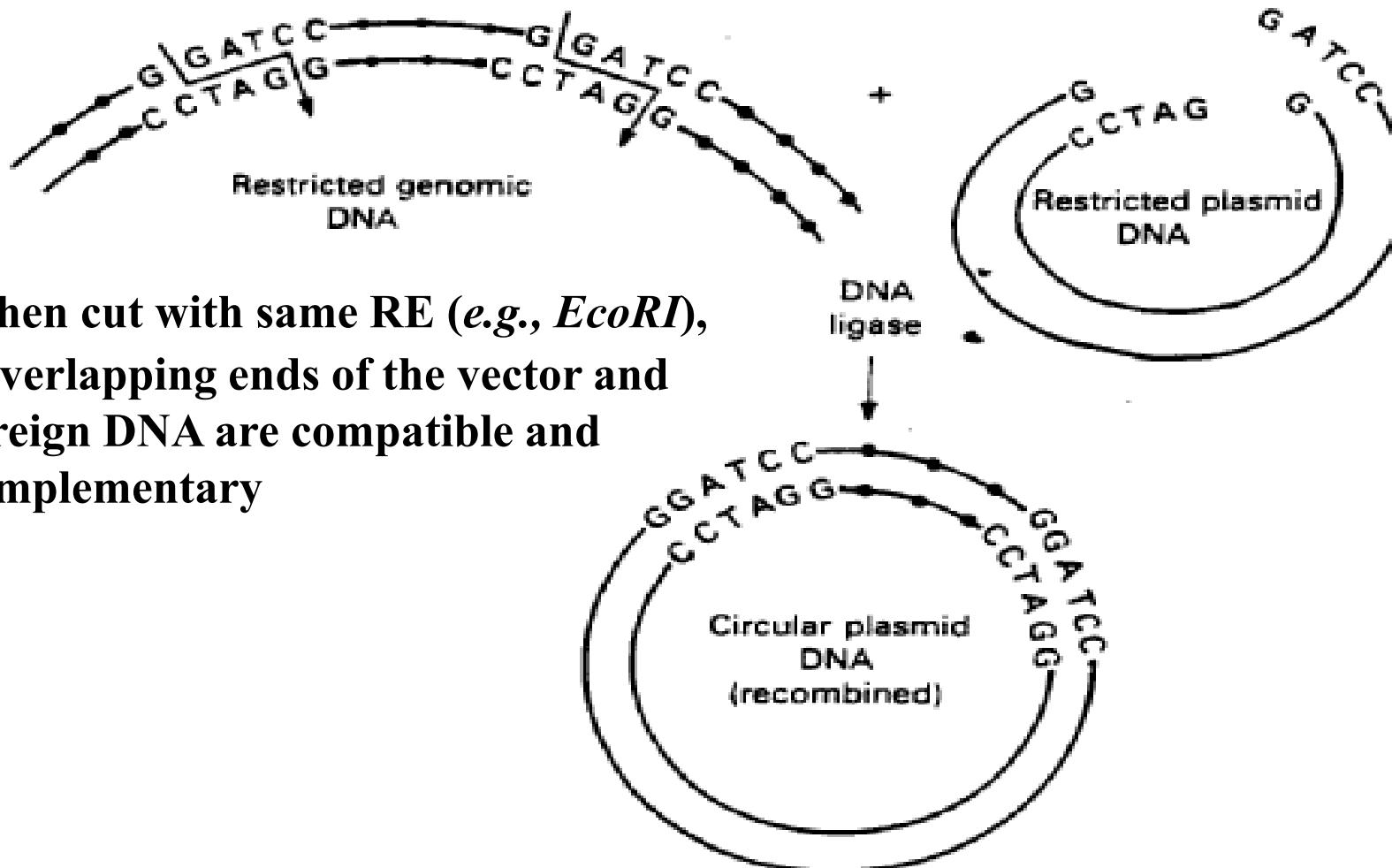
5'... CCC GGG... 3'  
3'... GGG CCC... 5'

5'... CCC  
3'... GGG

Blunt Ends

GGG... 3'  
CCC... 5'

# Restriction enzyme digestion of genomic DNA and plasmid vector DNA



When cut with same RE (e.g., *EcoRI*),  
- overlapping ends of the vector and  
foreign DNA are compatible and  
complementary

**How does one cut a DNA if it **doesn't** contain desired RE sites?**

**Or**

**If the RE site is **present** within the DNA of interest?  
(say, within SARS-CoV-2 N gene )**

**Or**

**If the RE result in **blunt-ended** DNA fragments, how do we insert the fragment in a cloning vector?**

**Cutting with Sma I**



5'... CCC GGG... 3'

3'... GGG CCC... 5'

5'... CCC  
3'... GGG

**Blunt Ends**

GGG... 3'  
CCC... 5'

## How to clone a **blunt-ended** DNA fragment?

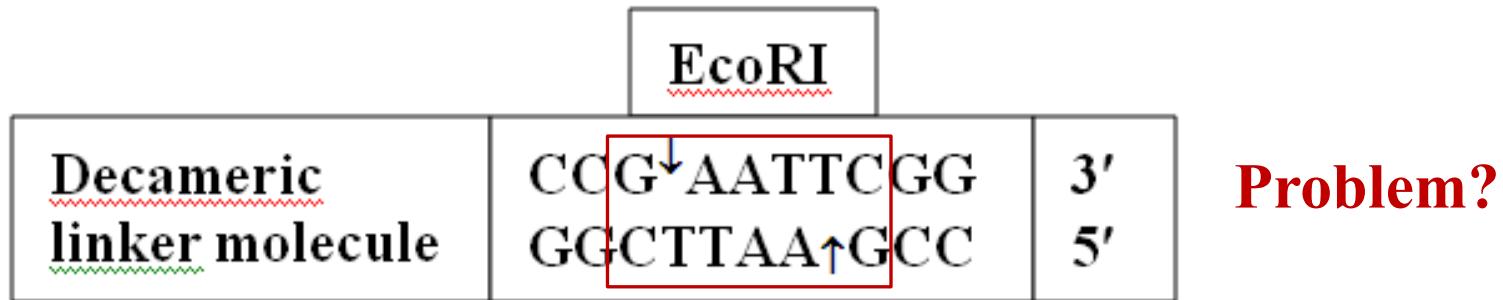
- a **linker** molecule can be ligated on either side by **DNA ligase**, cut with the RE contained in the linker molecule to obtain cohesive ends.

**How does one cut a DNA if it **doesn't** contain desired RE sites?**

- the DNA maybe be cut with whatever RE sites are available, and then **linker or adaptor** molecules maybe added to enable ligating it to the vector.

# Linkers & Adaptors

**Linkers** - short, double-stranded DNA molecules (~ 8-14bp) with one **internal site** for RE (~ 3-8bp)



- the sites for the enzyme used to generate cohesive ends may be present in the target DNA fragment, limiting its use for cloning.
- This problem can be solved using adaptors.

# Linkers & Adaptors

**Adaptors** - chemically synthesized DNA molecules with pre-formed cohesive ends

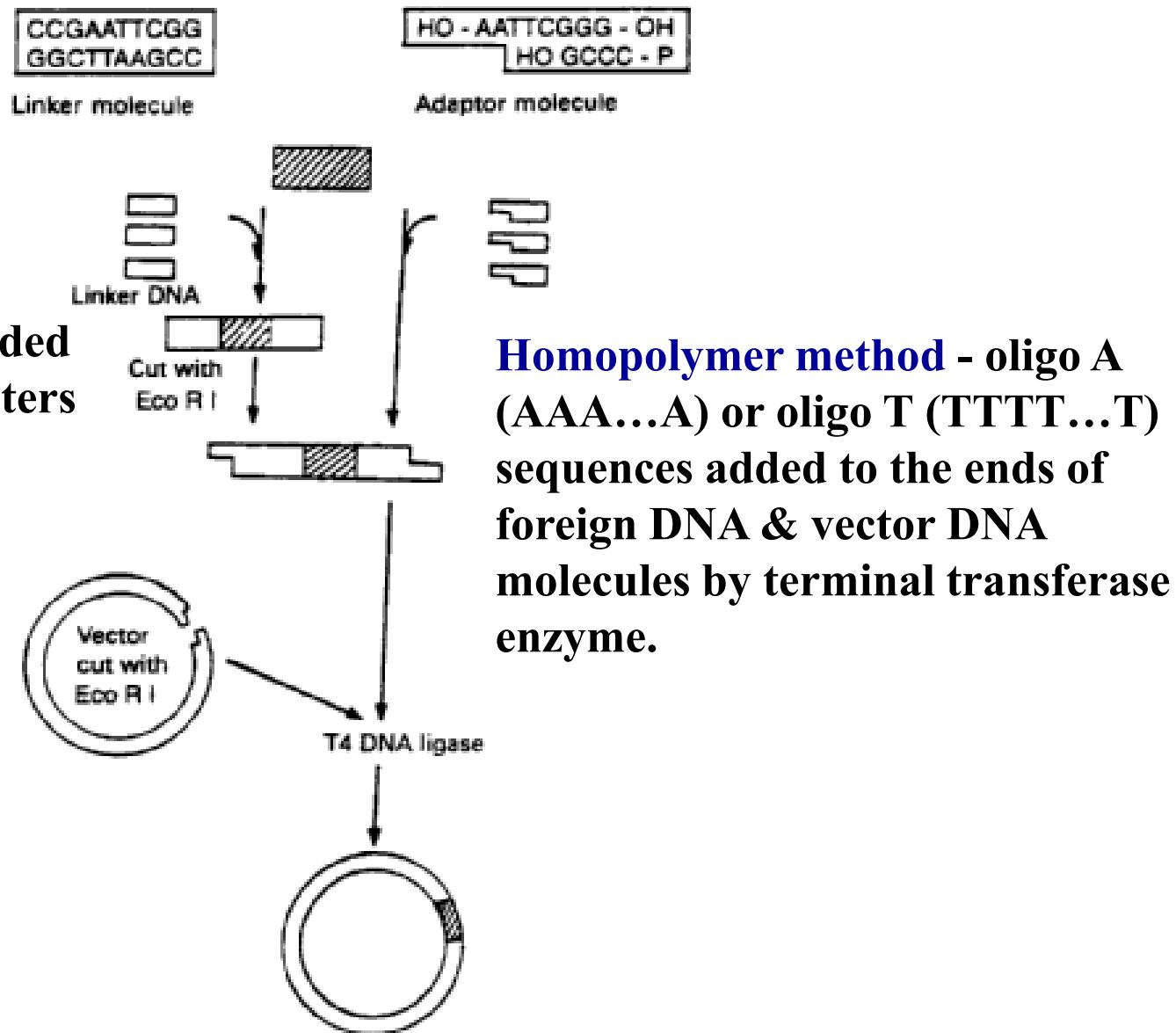
- it has **one blunt end** bearing a 5' phosphate group and another **cohesive end** for a specific RE which is not phosphorylated to prevent self ligation.



**Adaptor molecule**

- reduces the need for restriction digestion following ligation

# Use of Linker & Adaptor Molecules in the Formation of Recombinant Plasmids



# Features of Restriction Enzymes

- Length of recognition sequence dictates how frequently the enzyme will cut a DNA sequence

Which of the recognition sites - of length, 4, 6, or 8, will occur at higher frequency? At what distances will they occur?

- Different REs can have the same recognition site and are called isoschizomers, e.g., *SacI* & *SstI* : GAGCTC
- Restriction recognitions sites can be unambiguous, e.g., *BamH I* recognizes the sequence GGATCC and no other, or ambiguous, e.g., *Hinf I* has a recognition site, GANTC.

Recognition sites for *Hinf I* will occur at what frequency?

# Features of Restriction Enzymes

- Recognition site for one enzyme may contain the restriction site for another, e.g., *BamH* I recognition site (GGATCC) contains the recognition site for *Sau3A* I (GATC).

*Sau3A* I recognizes the sequence GATC and produces the same sticky ends as *BamH* I upon cutting

Will the two REs give the same results? If not, which one will give larger number of fragments?

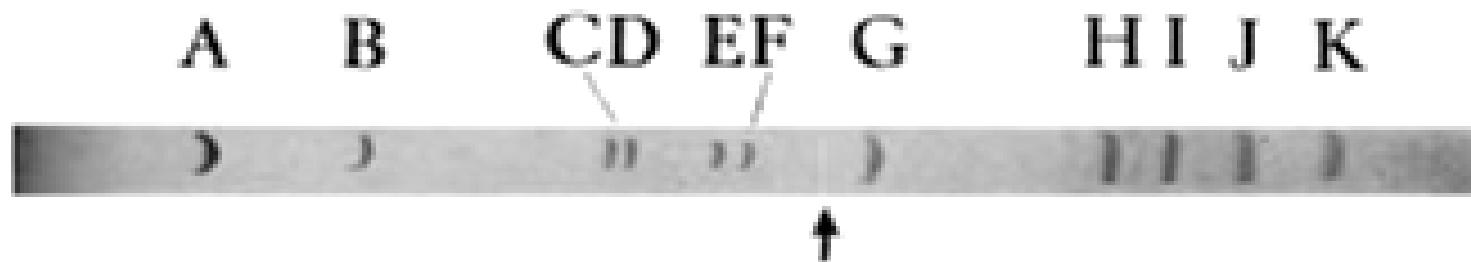
- Most recognition sequences are palindromes - they read the same forward and backward

Can we use the property of palindrome sequence to identify restriction recognition sites?

# Applications of Restriction Enzymes

Danna & Nathans showed that it was possible:

- to prepare a **physical map** of the SV40 genome
- to localize the **origin of replication**
- to position **early & late genes** of SV40 onto this “restriction map”
- that any individual gene could be mapped by **testing for biological activity** during transformation experiments
- **informative mutants** could be made by deleting one or more of the specific fragments



# Applications of Restriction Enzymes

- **Variations** in DNA sequences, *viz.*, mutations in recognition sites, copy number variation of VNTRs, insertions, deletions, inversions and translocations, can be identified by RE analysis
  - The length variations is known as **restriction fragment length polymorphisms (RFLPs)**.
- In **genetic engineering** - using REs DNA may be cut at precise locations & using DNA ligase, reassembled in any desired order, allowing the researchers to assemble **customized genomes**; create designer bacteria that make insulin, or growth hormones, or add genes for disease resistance to agricultural plants, etc.
- in **DNA sequencing** – first step is to cut the DNA in manageable pieces

# Restriction Map

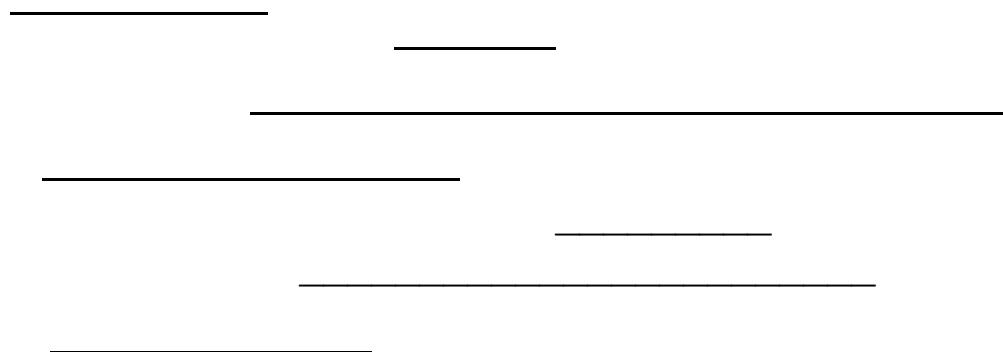
**Restriction map** is a description of restriction endonuclease cleavage sites within a piece of DNA

- generating such a map is the first step in **characterizing** an unknown DNA

**Multiple Complete Digest Mapping** – creates a map by digesting DNA with multiple REs

- each recognizing a different specific short DNA sequence and producing a separate **fingerprint** for each clone

Because of the frequent occurrence of these sites, restriction mapping produces a relatively **fine scale** of physical map.



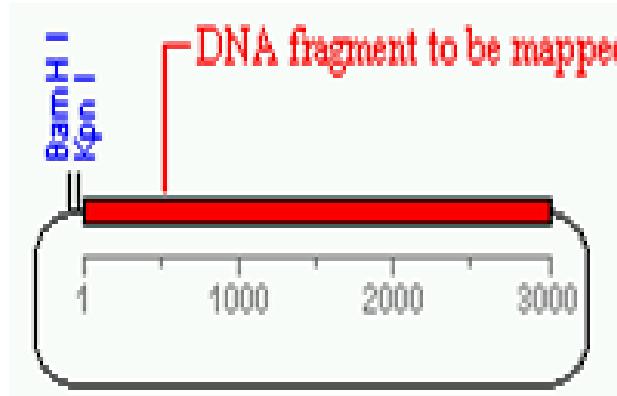
**How do you order the fragments in the correct order?**



**The fragments can be arranged in the correct order by finding the overlapping fragments**

# Restriction Mapping

Ex: Consider a plasmid that contains a 3000 bp fragment of unknown DNA & unique recognition sites for enzymes **Kpn I** & **BamH I**.



Consider first separate digestions with **Kpn I** & **BamH I**:

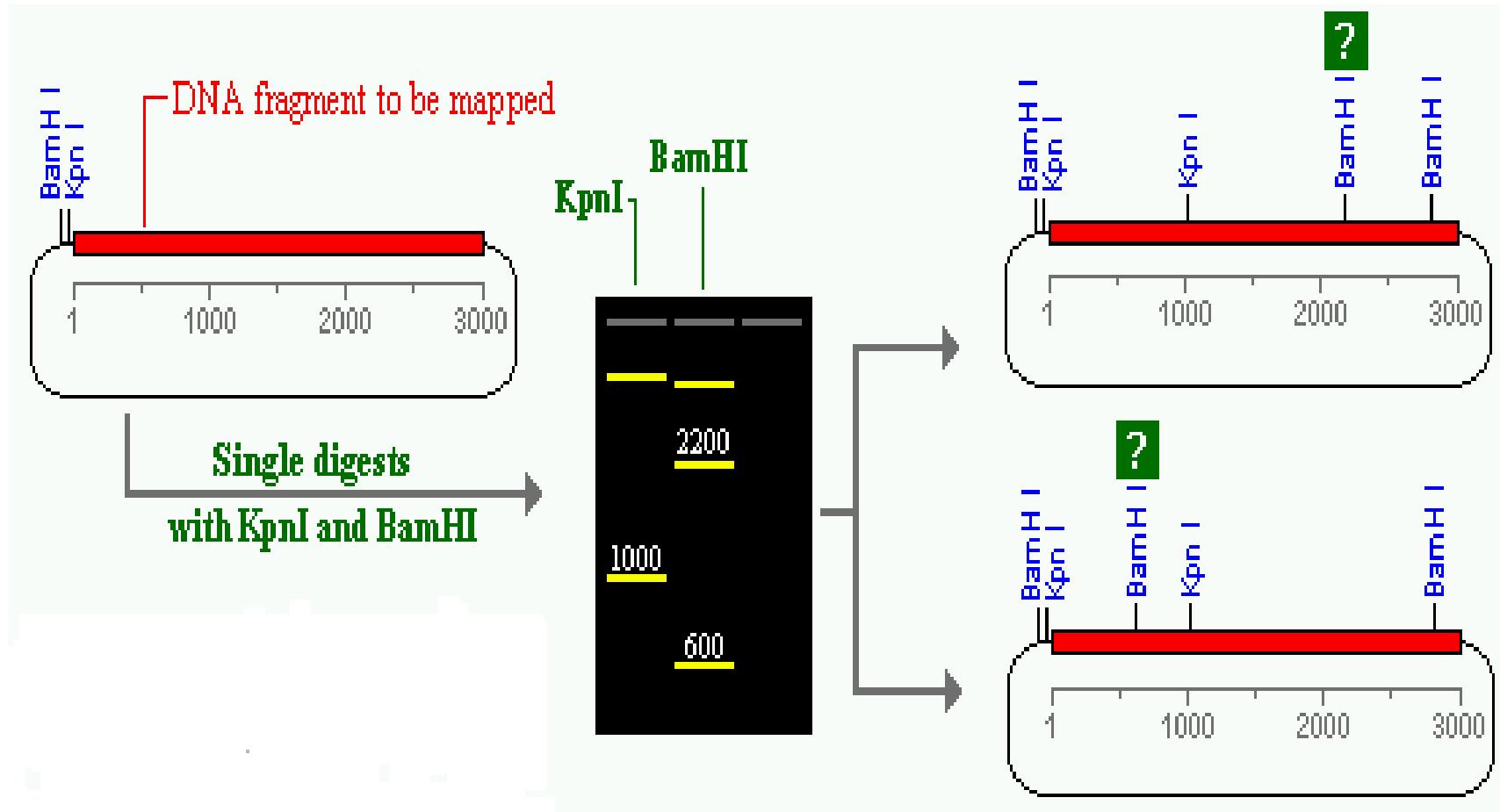
**Kpn I** yields 2 fragments: 1000bp & “big”

**BamH I** yields 3 fragments: 600, 2200 & “big”

big – part of unknown DNA sequence + vector

⇒ one **Kpn I** site & two **BamH I** sites are present in the unknown DNA sequence, given 1 each on the vector sequence

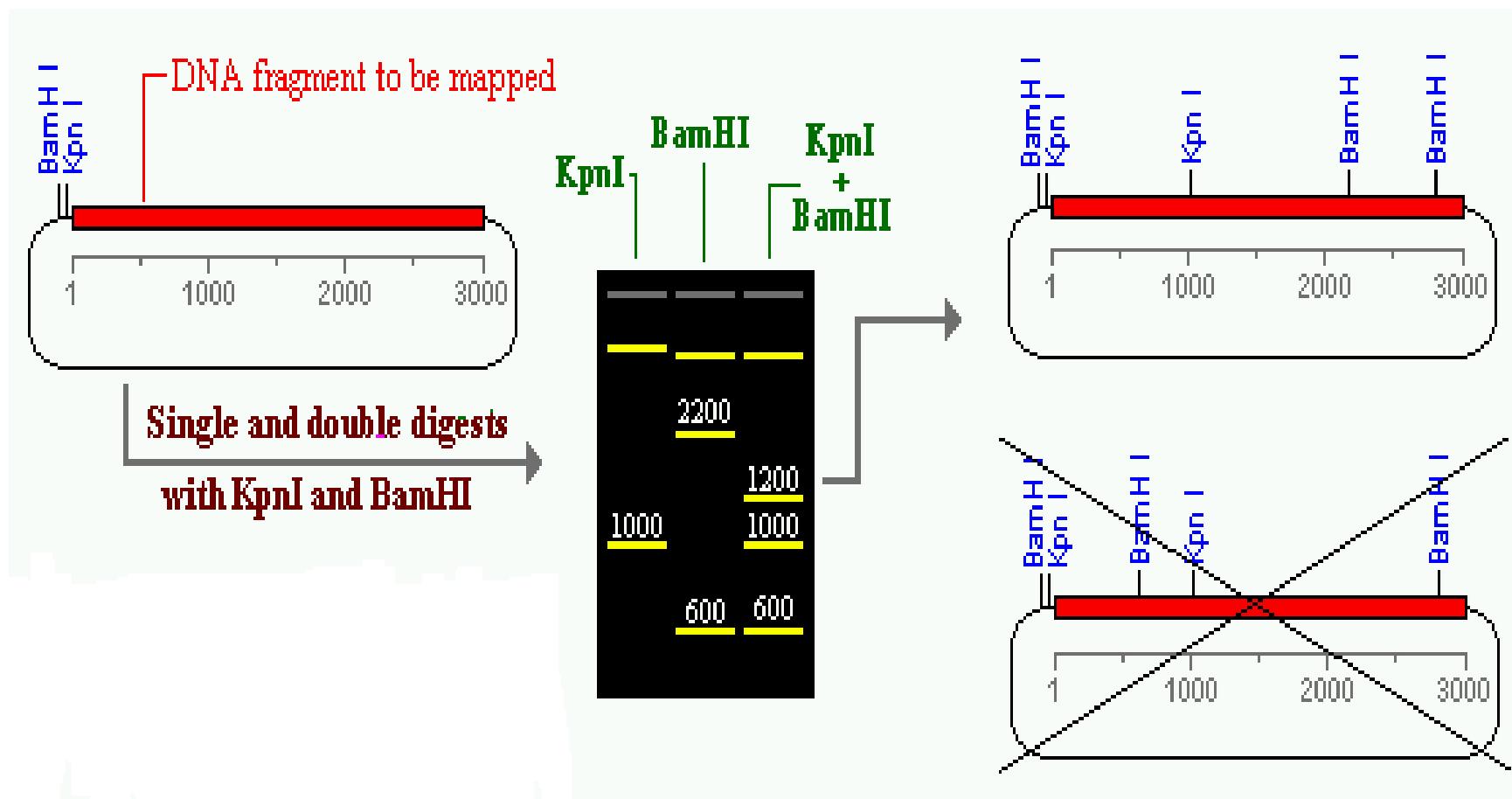
# Restriction Mapping



One BamH I site is at **2800 bp**. Trick to determine the location of 2<sup>nd</sup> BamH I site is to digest the plasmid with **Kpn I & BamH I** together

# Restriction Mapping

Double digest yields fragments of **600, 1000 & 1200 bp** (plus the "big" fragment).



# Restriction Mapping

If the above process is conducted with a larger set of enzymes, a much more complete map would result

**single digests** - are used to determine which fragments are in the unknown DNA, and

**multiple digests** - to order and orient the fragments correctly.

For any novel genome, e.g., SARS-CoV-2, can a physical map be constructed computationally?

# Restriction Mapping

## Using a Computer to Generate Restriction Maps

If the sequence is known, feed it to computer programs, which will search the sequence for various RE recognition sites and build a map.

- **Mapper** - available as part of Molecular Toolkit  
<http://arbl.cvmbs.colostate.edu/molkit/mapper/>
- **Webcutter**  
<http://www.firstmarket.com/cutter/cut2.html>
- **RebSite** – as part of the REBASE Tools  
<http://tools.neb.com/REBsites/index.php3>

# REBASE

## The **R**estriction **E**nzyme **data****B**ASE

**A comprehensive database containing information:**

- **restriction enzymes, methylases & related proteins involved in restriction-modification processes**
- **recognition and cleavage sites, isoschizomers, neoschizomers, commercial availability, methylation sensitivity, crystal & sequence data.**

**All newly sequenced genomes are analyzed for the presence of putative restriction systems and these data included in REBASE**

**It is updated daily (<http://rebase.neb.com/>)**

**Ref: Robert et al, *Nucl. Acids Res.* 43: D298-D299 (2015)**

[Back to...](#)

# REBsites

[Program Guide](#)[Help](#)

This tool will take a DNA sequence and digest it with one example of each of the known Type 2 restriction enzyme specificities.

The maximum size of the input file is 2 MByte, and the maximum sequence length is 200 Kbases.

Local sequence file:  [Browse...](#)

GenBank number:  ([Browse GenBank](#))

Name of sequence: **NC\_045512** (optional)

or Paste in your DNA sequence: (plain or FASTA format)

The sequence is:  Linear  Circular

Input sites:  All specificities  Defined oligonucleotide sequences:

[Clear the table below](#)

Name	Oligonucleotide sequence

Standard sequences:

- Lambda
- pBR322
- PhiX174
- Ad2

[Submit](#)

**theoretical digest with all REBASE prototypes**

[\[New DNA\]](#)

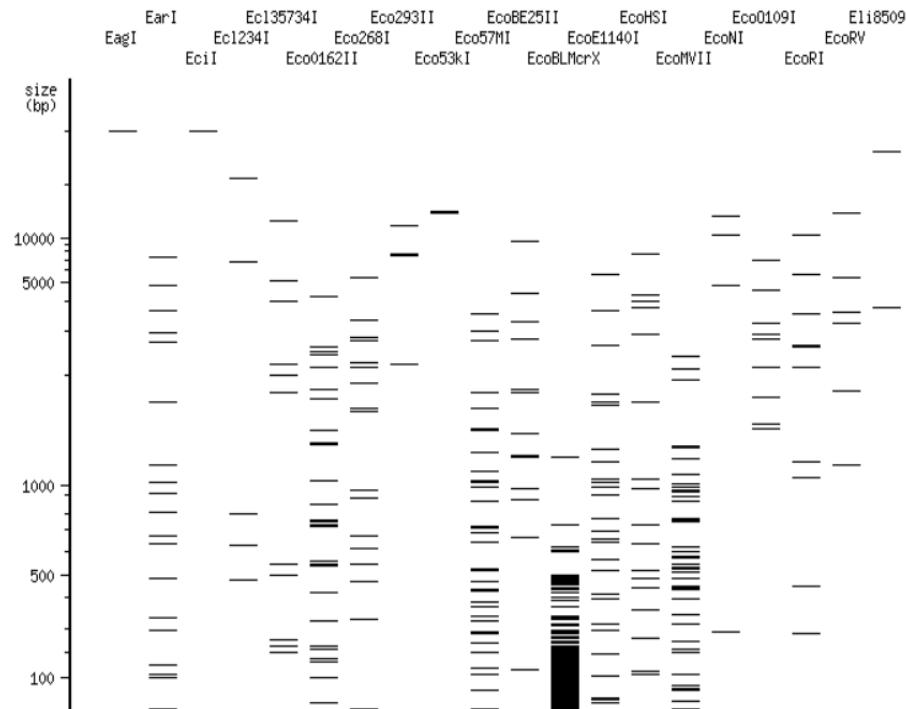
## REBsites

NC 045512

Gel:     
Order by:

[\[<< Prev\]](#) [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [\[Next >>\]](#)

[\[Print\]](#)



Click on an enzyme name for a list of fragments/sites.

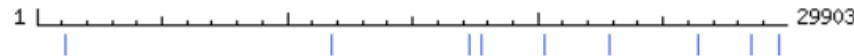
[Print](#)

## Fragment list

[Close](#)

NC 045512 digested with EcoRI

[\[Sites with flanks\]](#)



#	Location	Size [bp]
1	1162-11734	10573
2	11735-17280	5546
3	22871-26439	3569
4	20279-22870	2592
5	17729-20278	2550
6	26440-28551	2112
7	1-1161	1161
8	28552-29620	1069
9	17281-17728	448
10	29621-29903	283

# Assignment

- Write a program to generate a restriction map for Wuhan isolate-1 genome (Acc. Id.: NC\_045512) using EcoRI as RE compare your results with REBsites.
- Write a program to identify restriction recognition sites in the given DNA sequence.

# Cloning

# What is cloning?

The process of cloning involves the production of **multiple copies of a DNA fragment of interest by amplification *in vivo***

- depends upon the ability of vectors to continue their life cycles in bacterial or yeast cells in spite of having foreign DNA inserted into them.

**Cloning vector** - a DNA molecule that carries foreign DNA into a host cell, replicates inside a bacterial (or yeast) cell and produces many copies of itself and the foreign DNA

**- a vector containing foreign DNA is termed recombinant vector**

## Features of Cloning Vectors:

- sequences that permit the propagation of itself in bacteria (or yeast)
- a cloning site to insert foreign DNA; the most versatile vectors contain a site that can be cut by many REs
- a method of selecting for bacteria (or yeast) containing a vector with foreign DNA; usually accomplished by selectable markers for drug resistance

Major requirement of all vectors - an origin of replication for a given host cell in order that they may replicate autonomously (i.e., independently of the host's chromosome)

# Types of Vectors

Vector	Insert size (kb)
Plasmids	<10 kb
Bacteriophage	9 - 20 kb
Cosmids	33 - 47 kb
Bacterial artificial chromosomes (BACs)	75 - 125 kb
Yeast artificial chromosomes (YACs)	100-1000 kb

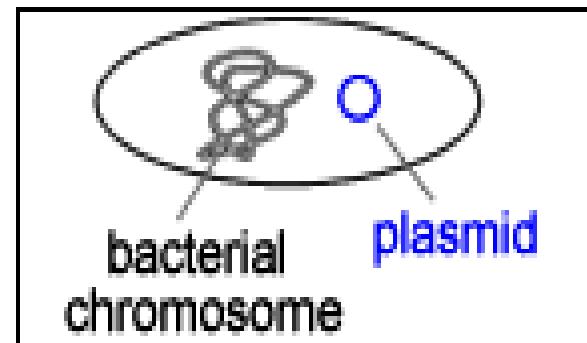
# Types of Vectors

**Plasmids** - an **extra-chromosomal** double-stranded **circular DNA** molecules that replicates autonomously inside the bacterial cell

Plasmids are important as one can:

- (i) isolate them in large quantities,
- (ii) cut & splice them, add DNA of choice,
- (iii) put them back into bacteria, where they replicate along with the bacteria's own DNA,
- (iv) isolate them again to get billions of copies of inserted DNA

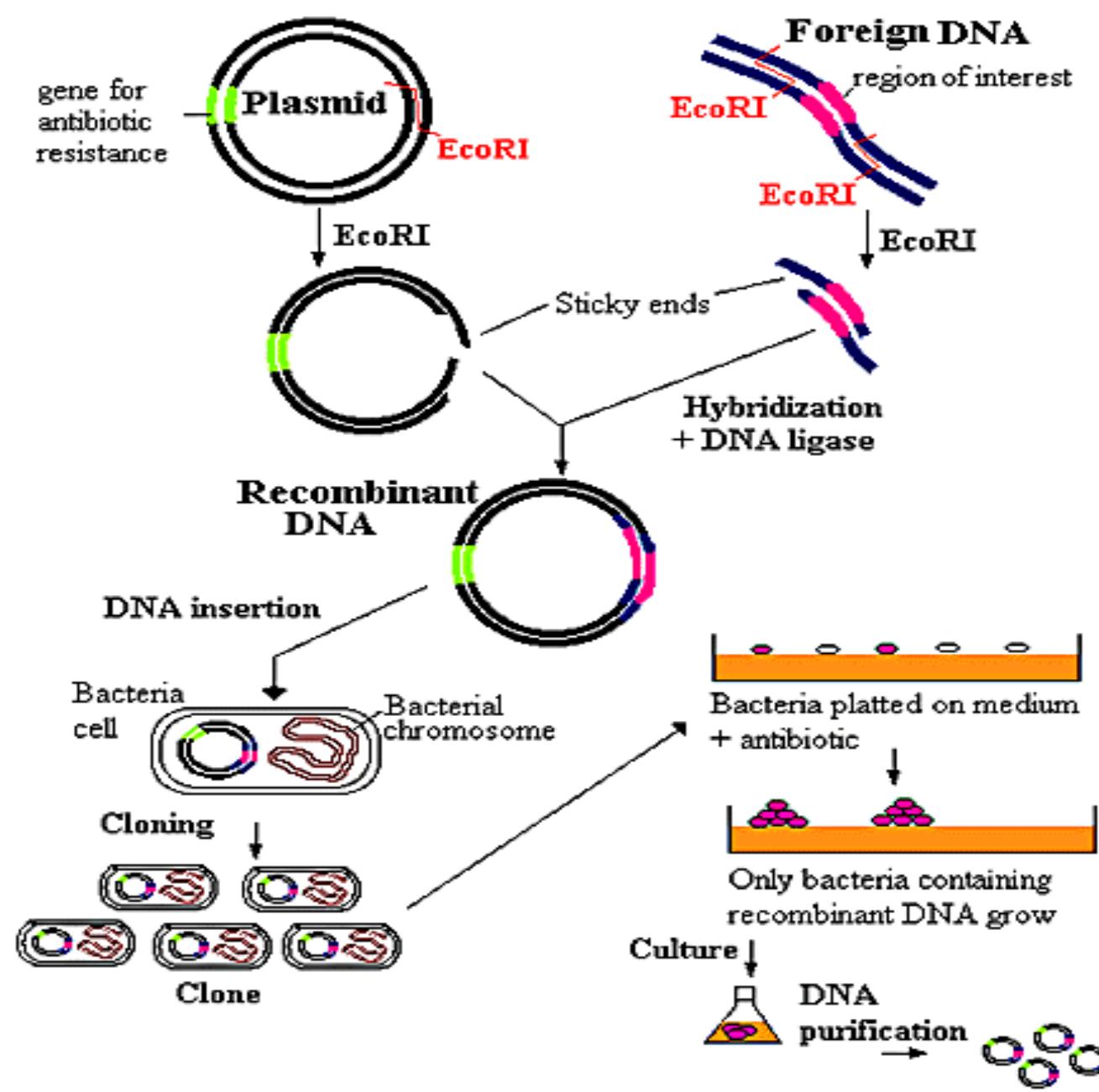
**Limitation:** size of DNA that can be introduced into the cell by transformation ( $\sim 2 - 10\text{kb}$ )



Plasmid vectors are derived from **naturally occurring plasmids** of *E. coli* such as **ColE1** or from related plasmid **pMB1**

**pBR322** – most widely used cloning vectors of *E. coli*, is a hybrid between **ColE1** & genes coding for **resistance to antibiotics tetracycline & ampicillin**

**What's the advantage of inserting genes coding for resistance to antibiotics into a vector?**



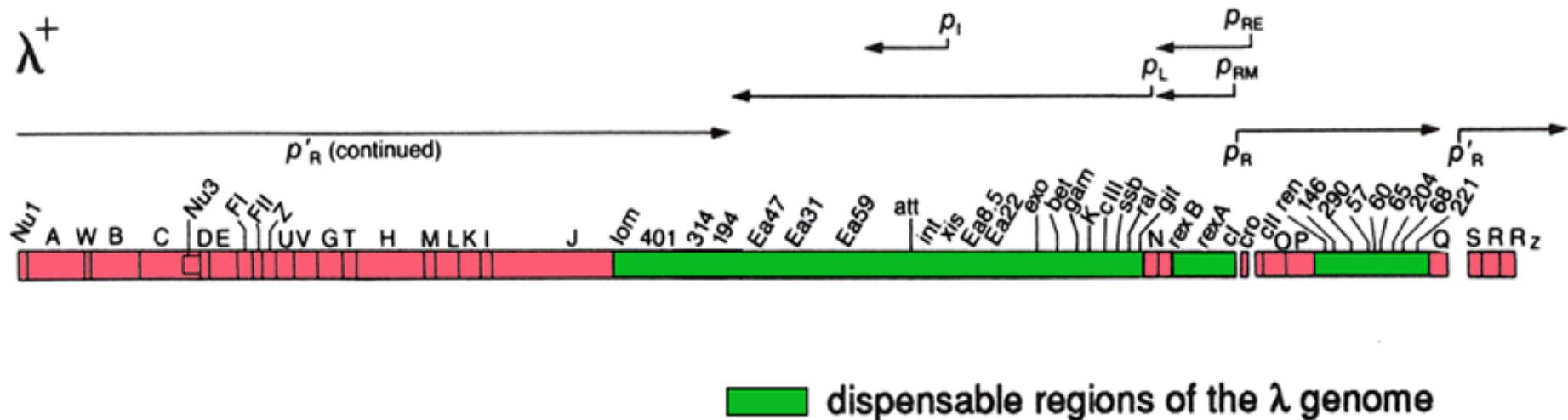
## Cloning into a plasmid

# Types of Vectors

# Bacteriophage Vectors

- a **double-stranded linear** molecule of size **49.5Kbp**

## Cloning limit: 9 - 20 kb



**Enterobacteria phage  $\lambda$  is a bacterial virus, or bacteriophage, that infects the bacterial species *E. coli*.**

# Artificially Constructed Vectors

**Cosmids** - an **extra-chromosomal circular DNA** molecule that combines features of plasmids and cos gene of phage lambda

**Cloning limit: 35 - 50 kb**

**BAC - Bacterial Artificial Chromosome**

- based on naturally occurring F-factor plasmid found in the bacterium *E. coli*.

**Cloning limit: 100-300 kb**

**YAC - Yeast Artificial Chromosomes**

- it is a vector constructed from yeast DNA, used to clone large DNA fragments

**Cloning limit: 100-1000 kb**

**Useful for cloning long segments of eukaryotic DNA**

**YAC** - a functional self-replicating artificial chromosome. It includes three specific DNA sequences that enable it to propagate from one cell to its offspring:

- **TEL:** The telomere which is located at each chromosome end, protects the linear DNA **from degradation** by nucleases
- **CEN:** The centromere which is the attachment site for mitotic spindle fibers, "pulls" **one copy of each duplicated chromosome into each new daughter cell.**
- **ORI:** Replication origin sequences, specific DNA sequences that **allow the DNA replication machinery to assemble on the DNA and move at the replication forks**

It also contains few other specific sequences like:

- **A and B:** **selectable markers** that allow easy isolation of yeast cells that have taken up the artificial chromosome.
- **Recognition site** for two REs: **EcoRI & BamHI**

## **Why is it important to be able to clone large sequences?**

**To map the entire human genome ( $3 \times 10^9$  bps) would require  
more than 1000,000 plasmid clones (~10Kb limit).**

**In principle, the human genome could be represented in  
about 10,000 YAC clones (~1Mb limit)**

## What determines the choice vector?

- **insert size**
- **vector size**
- **restriction sites**
- **copy number**
- **cloning efficiency**
- **ability to screen for inserts**

# DNA Sequencing

**DNA Sequencing** - determine the precise sequence of nucleotides in a sample of DNA – **the order of A, T, G, C**

Various types of sequencing:

- Sequencing a **region of interest**, e.g., gene.
- **Whole Genome/Exome Sequencing**
- **cDNA Sequencing** – sequencing cDNA libraries of the expressed genes
- **High-throughput sequencing** – next-generation, 3<sup>rd</sup> & 4<sup>th</sup> generation sequencing - **whole Genome/Exome/targeted**
- **Metagenome sequencing** - sequencing of environmental samples
  - depending on the nature of analysis, type of sample, or type of sequencer used

# Sequencing a Region of Interest

First requirement in sequencing a region of DNA is

- to have **enough starting template for sequencing.**

This is achieved by **PCR - Polymerase Chain Reaction**

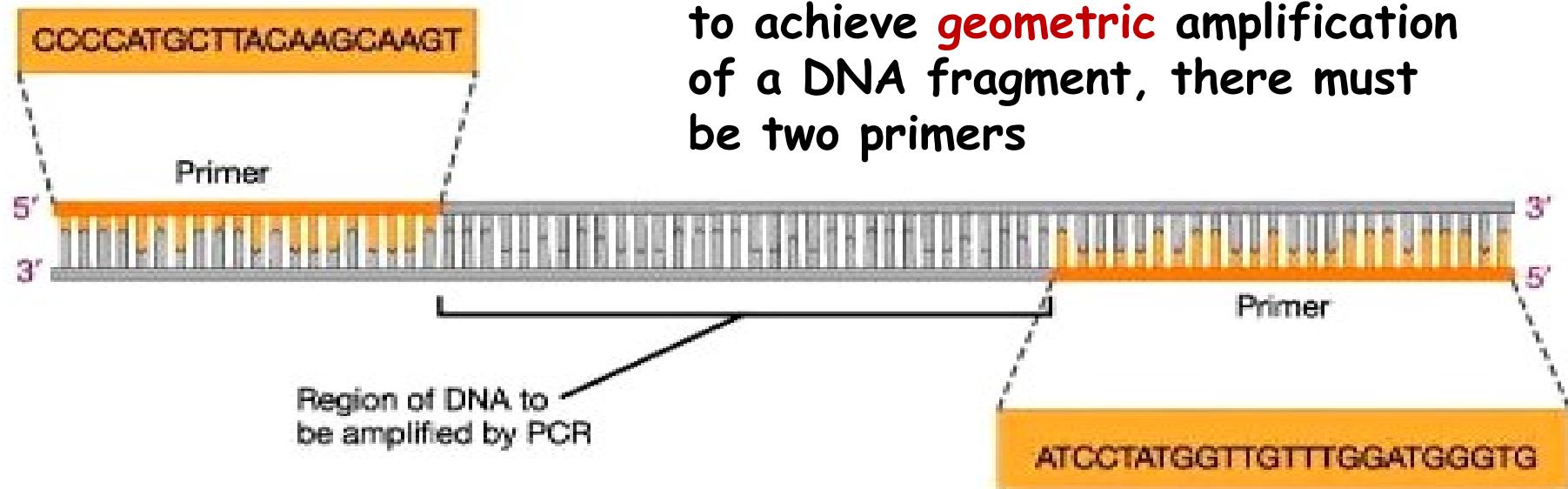
- carried out in an automated cycler for 30 - 40 cycles.

Essential requirements for a PCR:

- a mixture of 4 deoxy-nucleotides in ample quantities:  
dATP, dGTP, dCTP, dTTP
- Taq DNA polymerase
- Primers ?
- Genomic DNA of interest

**What is the advantage of using PCR over traditional gene cloning?**

## Region of DNA to be amplified by PCR



**Primers** - short single-stranded oligonucleotides which anneal to the DNA template and serve as a starting point for DNA synthesis

Why are primers required?

# The Cycling Reactions

## Step-1: Denaturation at 94°C

- opens up double stranded DNA, all enzymatic reactions stop.

## Step-2: Annealing at 54°C

- Primers jiggling around because of Brownian motion, binds to single stranded template once an exact match is found; the polymerase then attaches and start copying the template.

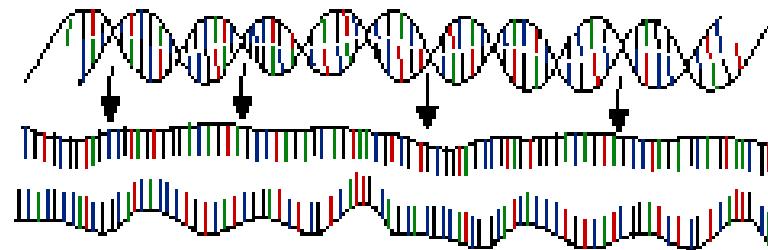
## Step-3: Extension at 72°C

- ideal working temperature for the polymerase. Bases complementary to the template are coupled to the primer on 3' side (reading the template from 3' to 5' side)

# Different Steps in PCR

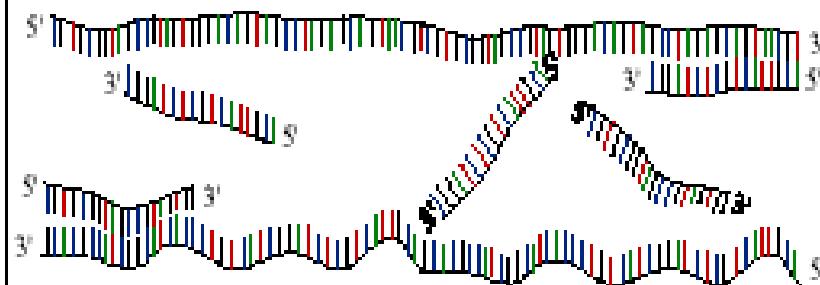
## PCR : Polymerase Chain Reaction

30 - 40 cycles of 3 steps :



Step 1 : denaturation

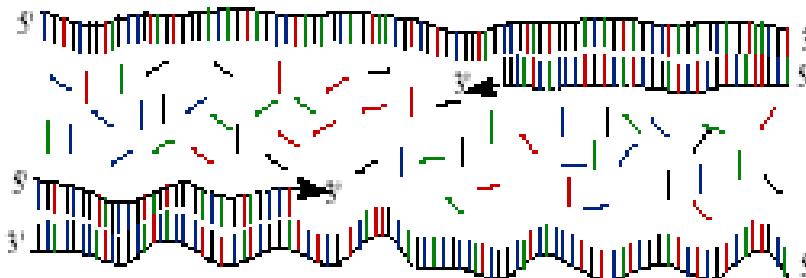
1 minut 94 °C



Step 2 : annealing

45 seconds 54 °C

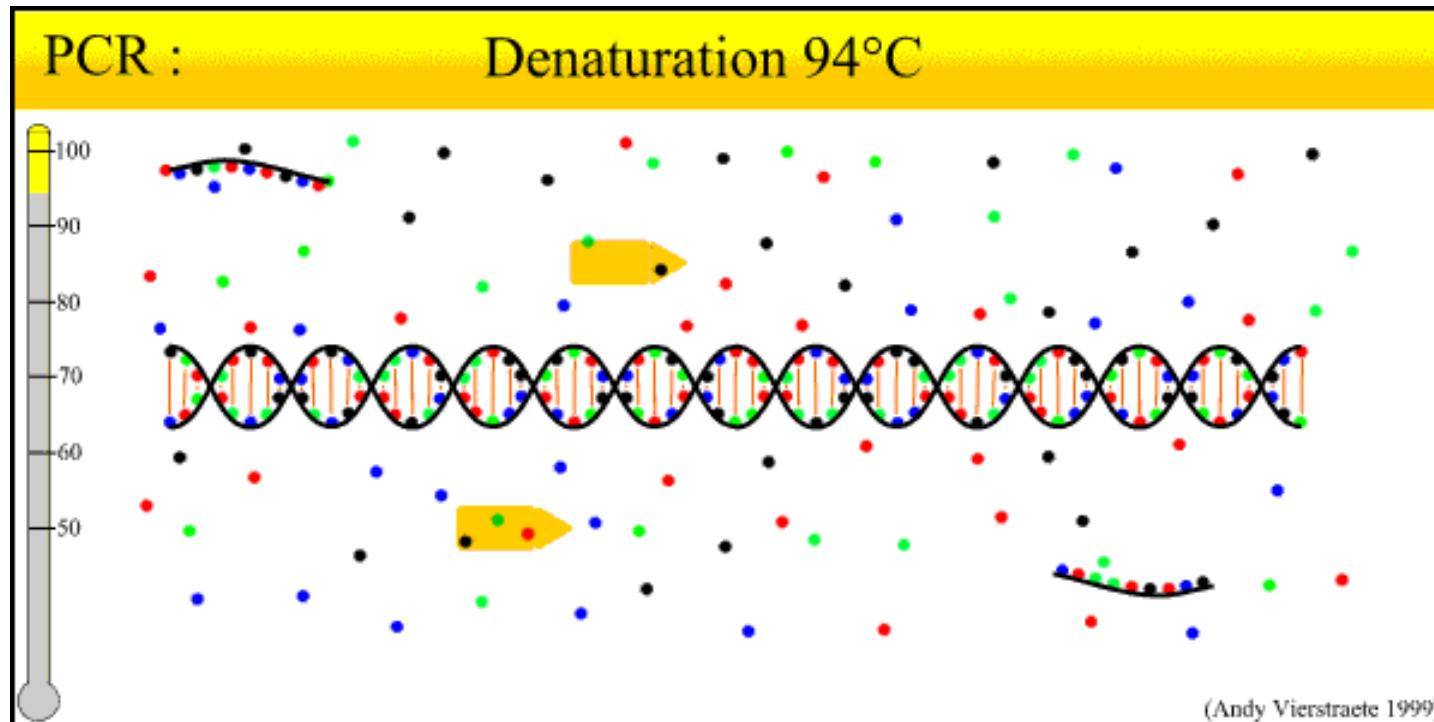
forward and reverse  
primers !!!



Step 3 : extension

2 minutes 72 °C  
only dNTP's

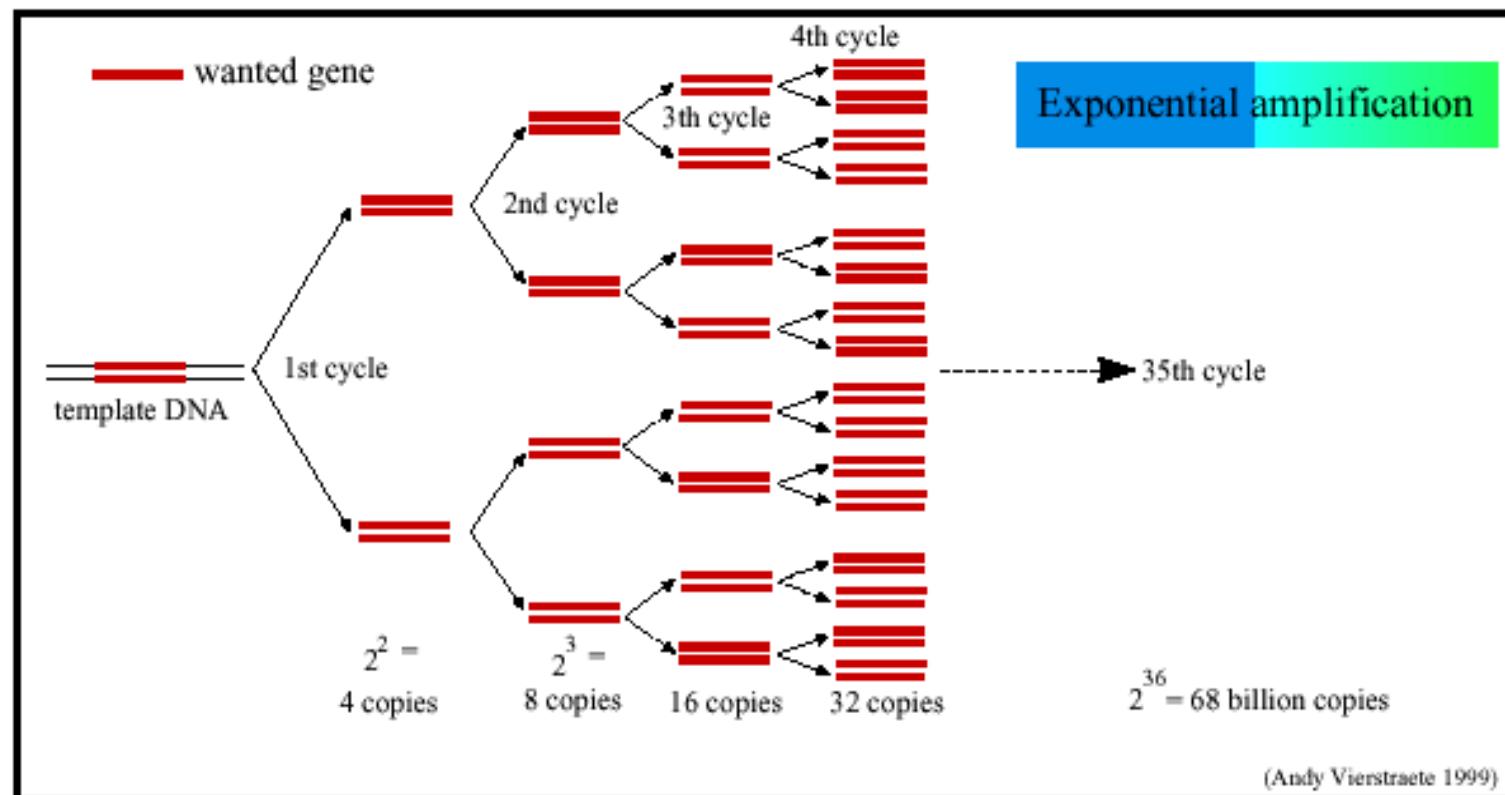
# Different Steps in PCR



# Exponential amplification of region of interest

Both strands are copied during PCR

- leading to an **exponential increase** of the number of copies of the region of interest.



# Verification of PCR Product

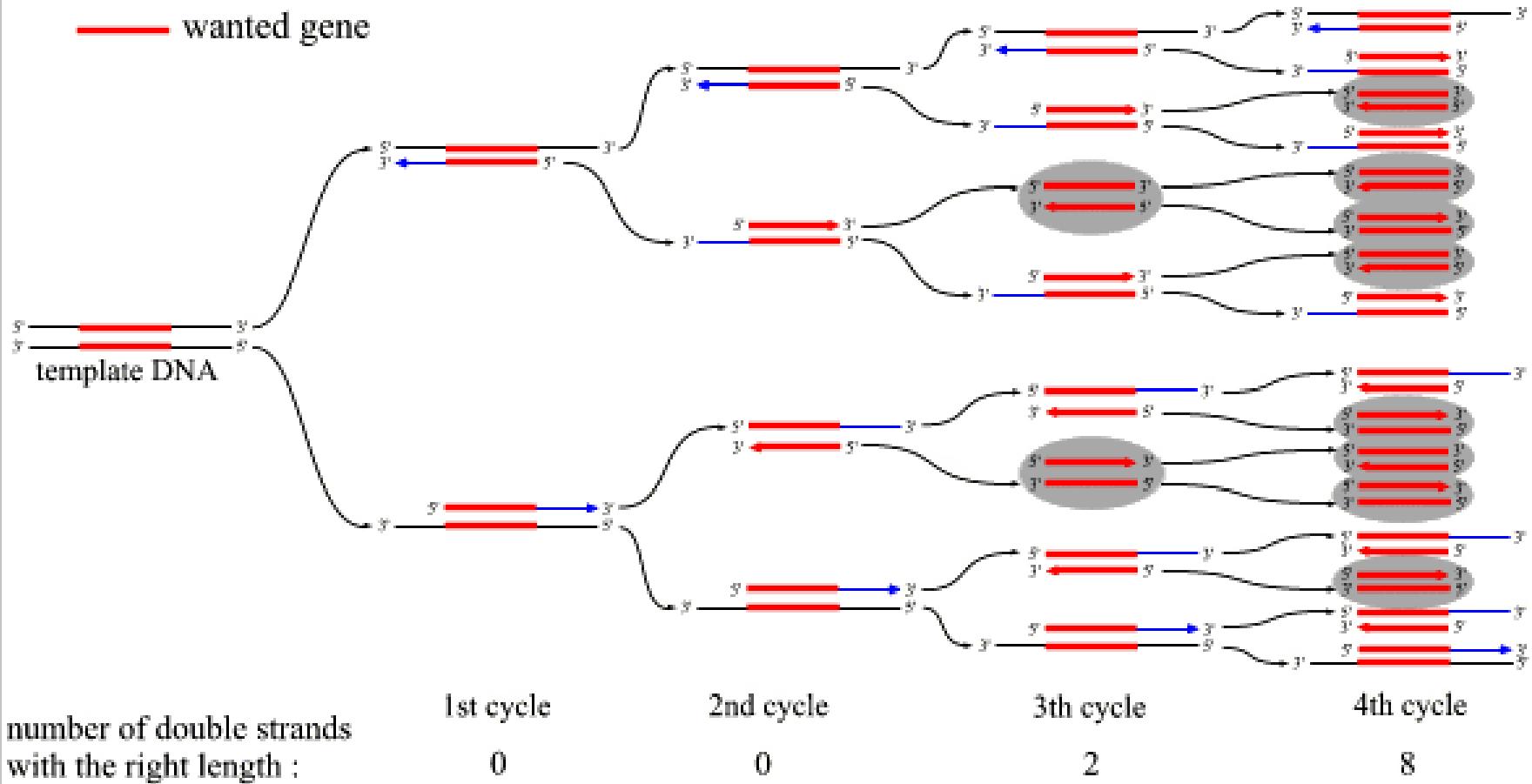
**Is the template copied during PCR and is it the right size?**

**Before the PCR product is used in further applications, it has to be checked if:**

- 1. A product is formed**
- 2. The product is of the right size**
- 3. Only one band is formed**

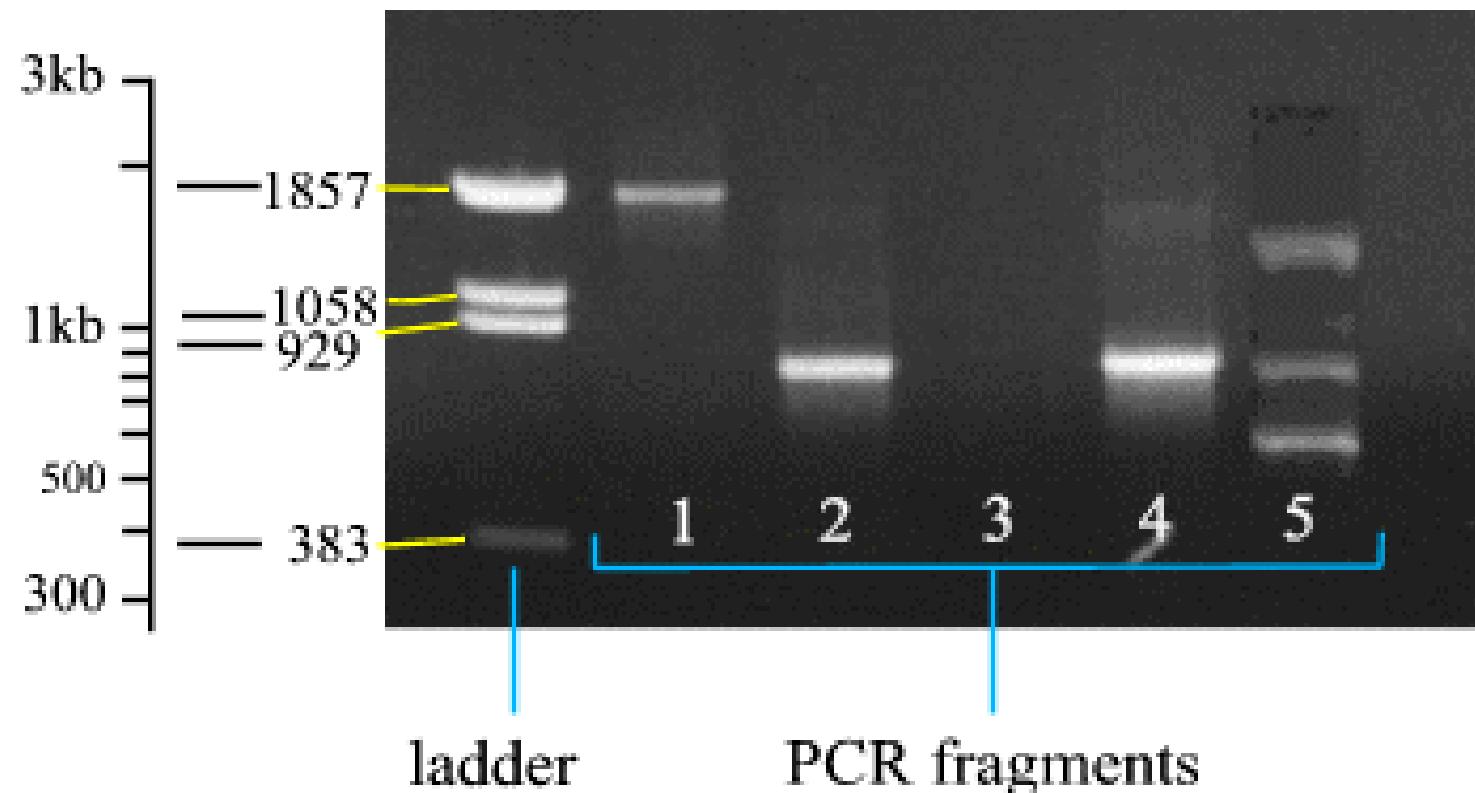
# First 4 cycles of a PCR reaction

## The first 4 cycles of PCR in detail



# Verification of the PCR product

Verification of PCR product on  
agarose or separeide gel

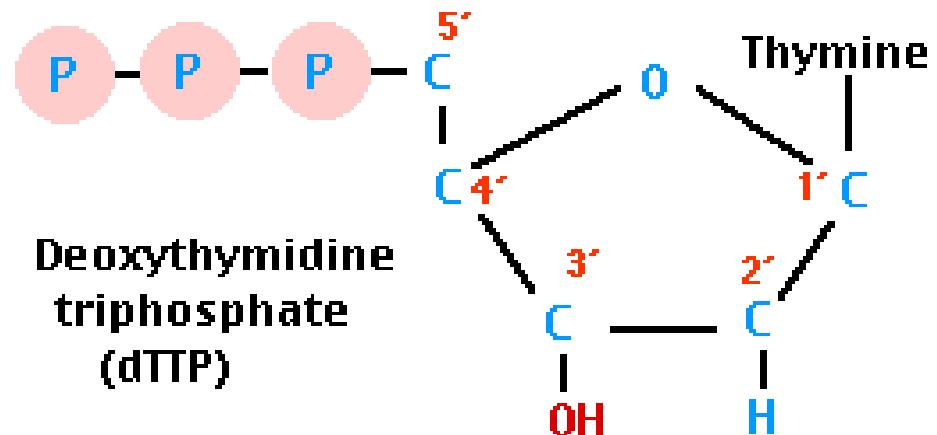


# PCR Sequencing

For sequencing, we don't start from gDNA (like in PCR) but mostly from PCR fragments or cloned genes.

Amplified PCR product is supplied with

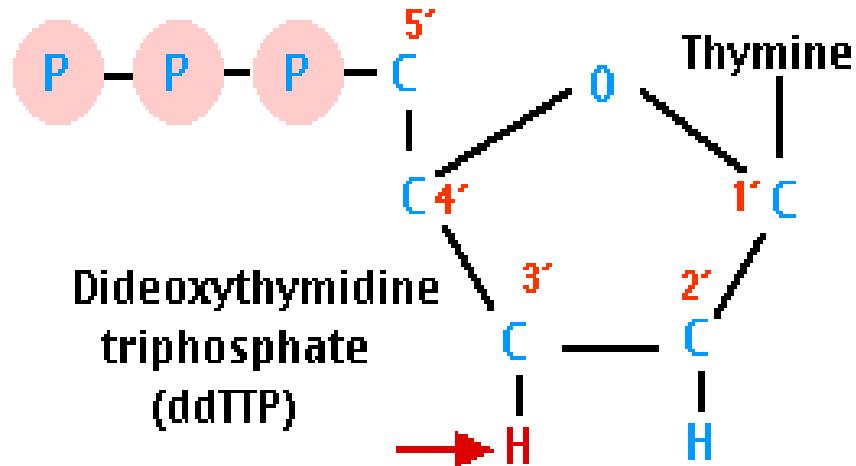
- a mixture of all four normal (deoxy) nucleotides in ample quantities
  - dATP
  - dGTP
  - dCTP
  - dTTP
- *Taq* DNA polymerase



# PCR Sequencing

- a mixture of all four dideoxynucleotides, each present in limiting quantities and each labeled with a "tag" that **fluoresces** a different color:

- ddATP
- ddGTP
- ddCTP
- ddTTP



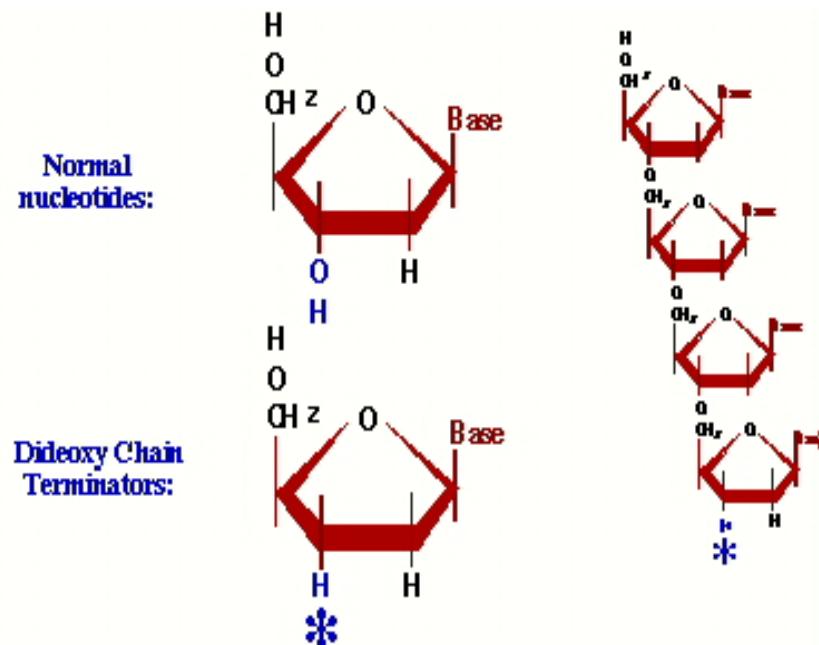
This method of DNA sequencing is called **dideoxy method**, or **chain termination method**, or **Sanger's method**.

# PCR Sequencing

**Dideoxy method:** DNA is synthesized from four deoxynucleotide triphosphates.

Each new nucleotide is added to 3' -OH group of the last nucleotide added.

When a dideoxynucleotide, **ddNTP is added to the growing DNA strand, chain elongation stops** because there is no 3'-OH for the next nucleotide to be attached to.



# Steps in PCR Sequencing

## I The sequencing reaction

- Denaturation at 94°C
- Annealing at 50°C
- Extension at 60°C ← instead of 72°C

## II Separation of the fragments

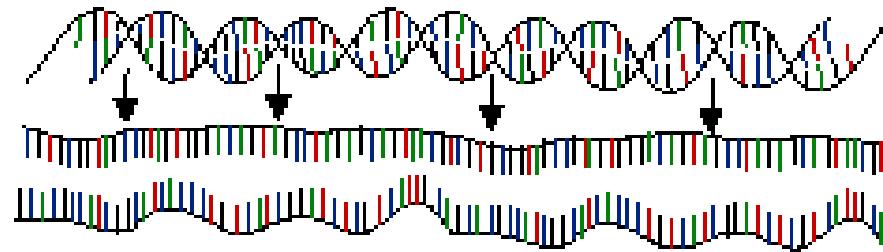
## III Detection on an automated sequencer

## IV Assembling the sequenced parts

# Different steps in Sequencing

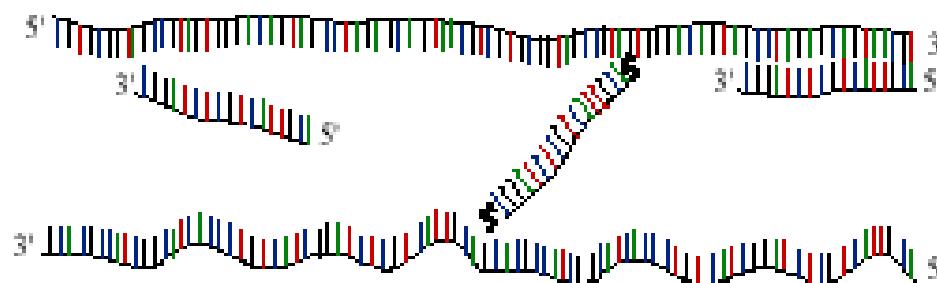
## Sequencing

30 cycles of 3 steps :



Step 1 : denaturation

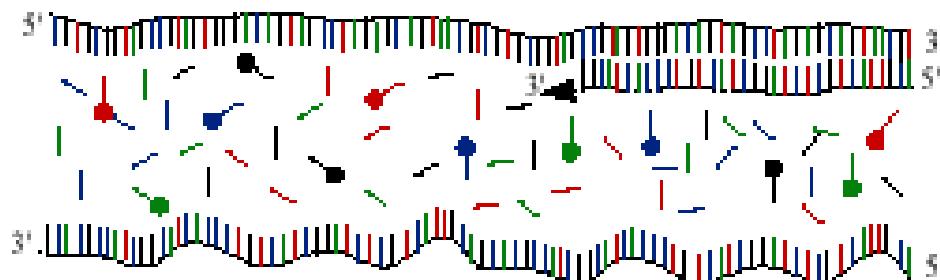
1 minut 94 °C



Step 2 : annealing

15 seconds 50 °C

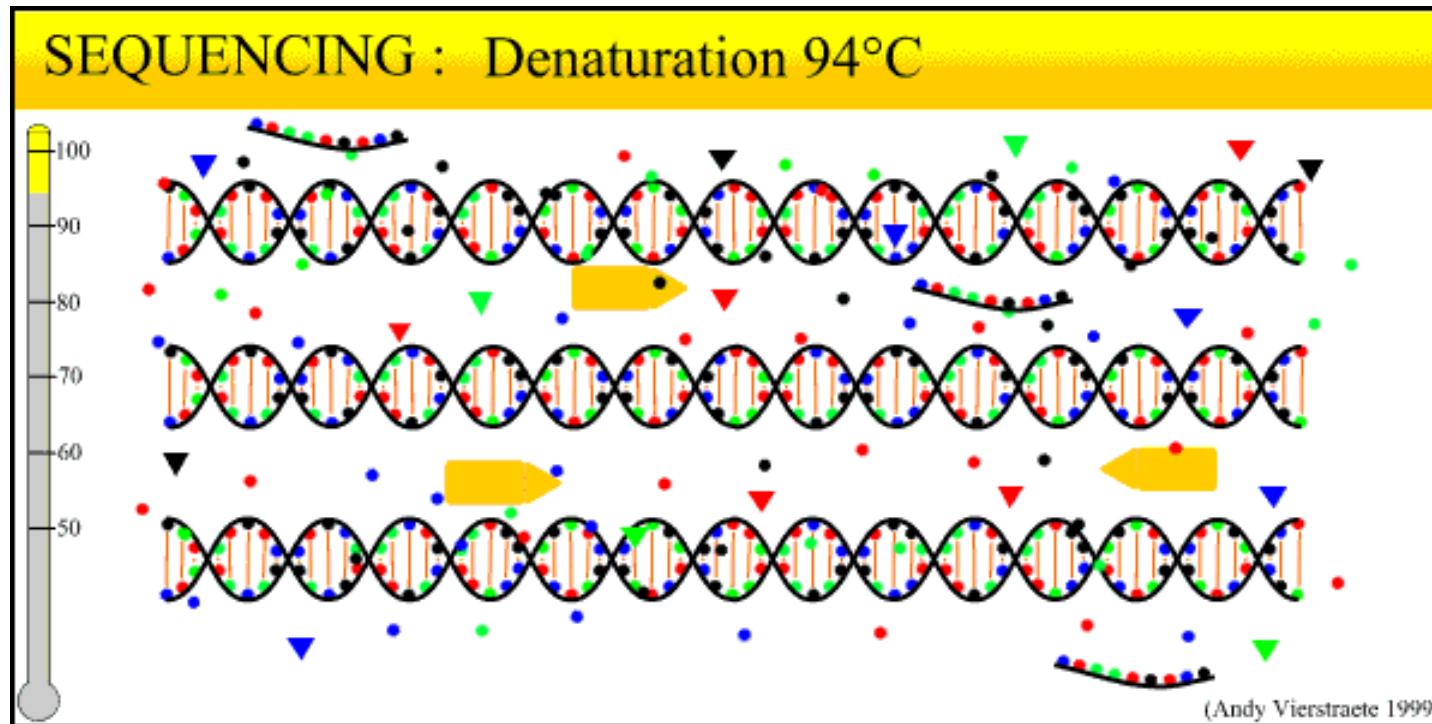
1 primer !!!!



Step 3 : extension

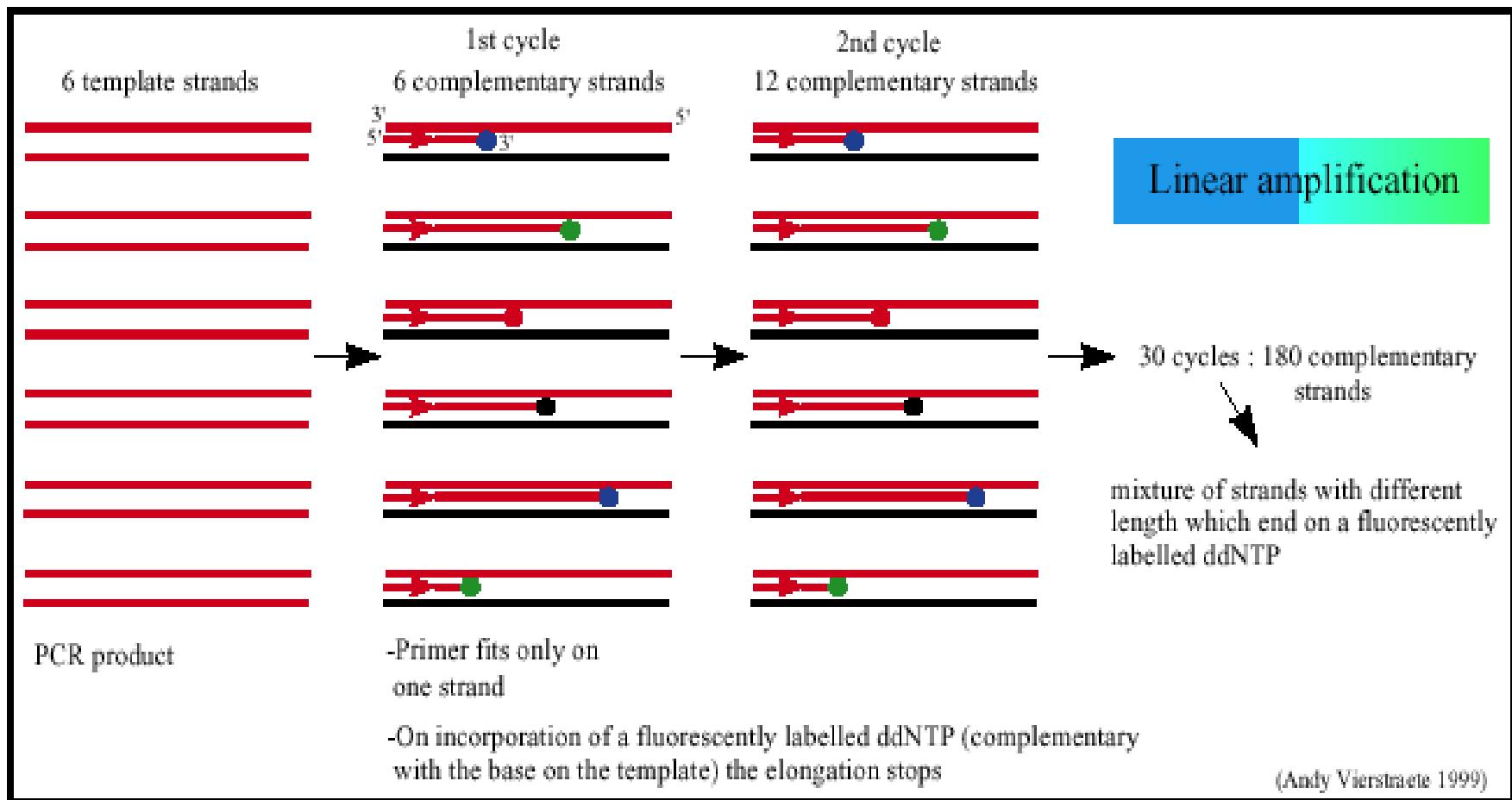
4 minutes 60 °C  
mixture of dNTP's |  
and ddNTP's |

# Different steps in Sequencing



# PCR Sequencing

Since only one primer is used, only one strand is copied during sequencing – resulting in a **linear increase** of the number of copies of one strand of the gene. Hence, a large amount of DNA in the **starting mixture for sequencing is required**.



# PCR Sequencing

## II Separation of the molecules:

After the sequencing reactions, the mixture of strands of different lengths, all ending on a fluorescently labeled ddNTP, need to be separated

- done by loading the mix on an acrylamide gel - gel electrophoresis.

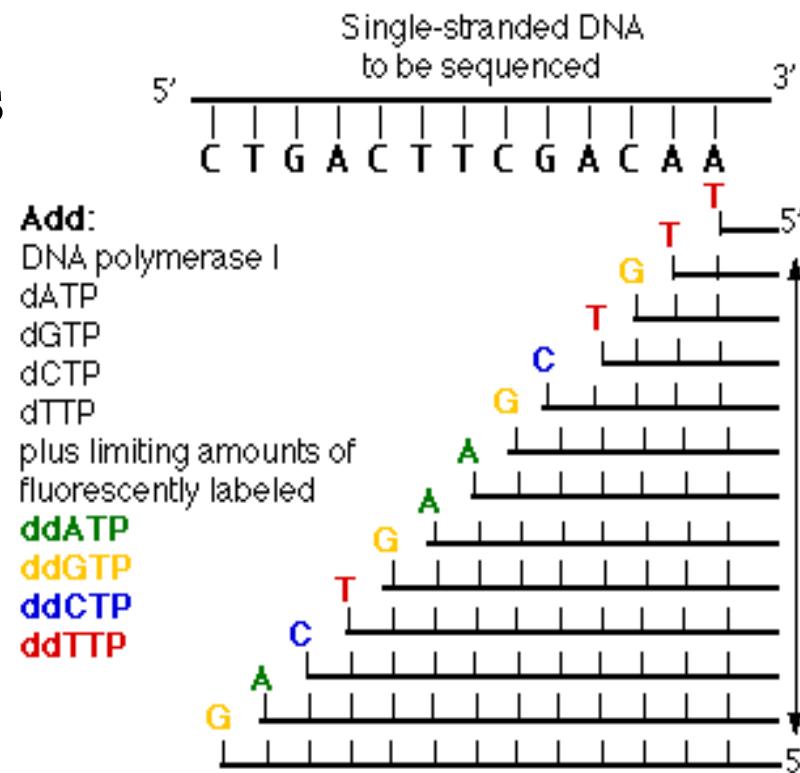
During electrophoresis, a voltage is created across the gel making one end positive and the other negative. DNA being –vely charged, migrates to the positive side.

DNA strands of different length migrate at different rates and thus can be separated based on their size - the smallest strand travels the fastest.

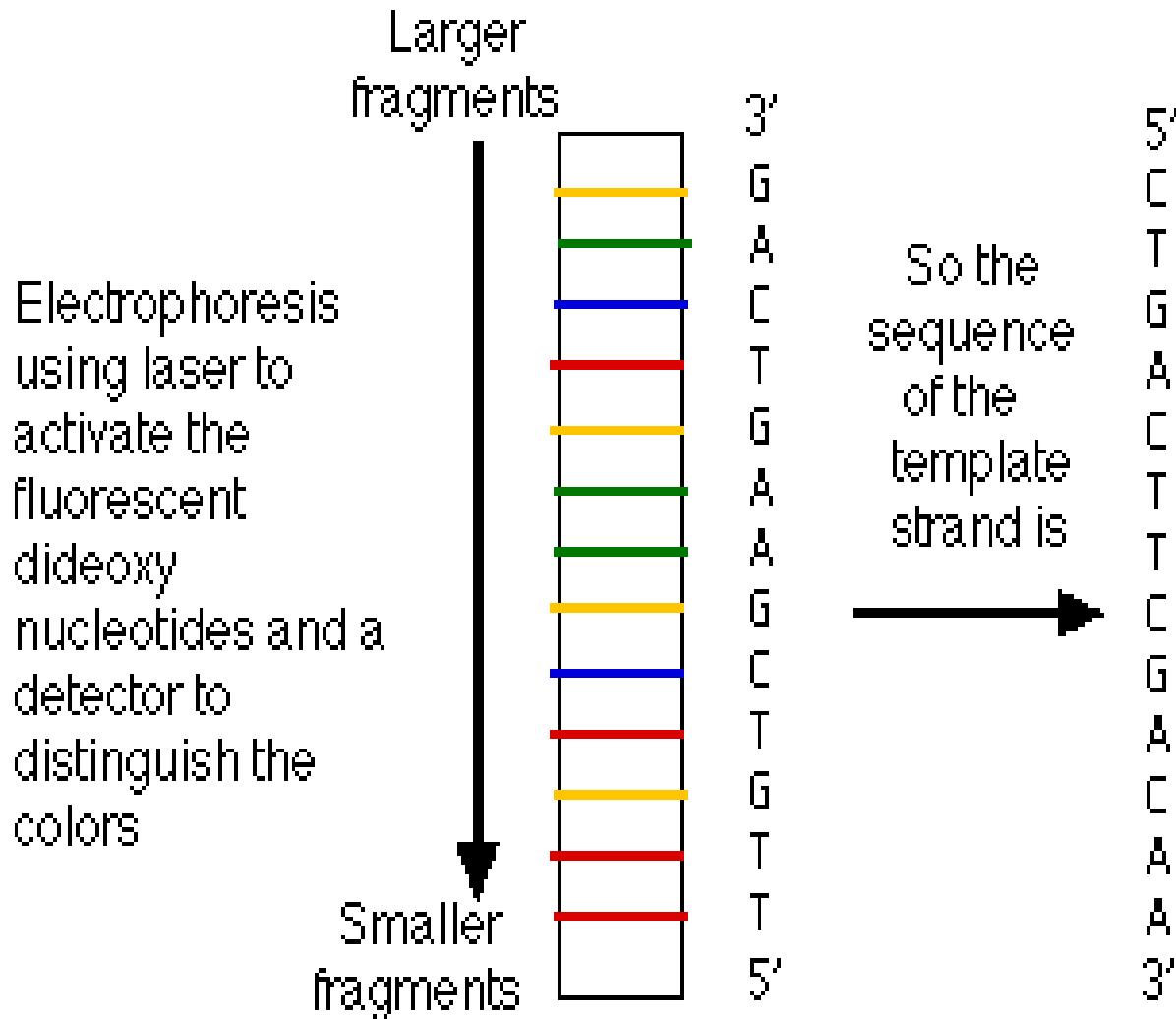
# Separation of molecules with electrophoresis

**Very good resolution** - a difference of even **one** nucleotide is enough to separate a strand from the next shorter or longer strand.

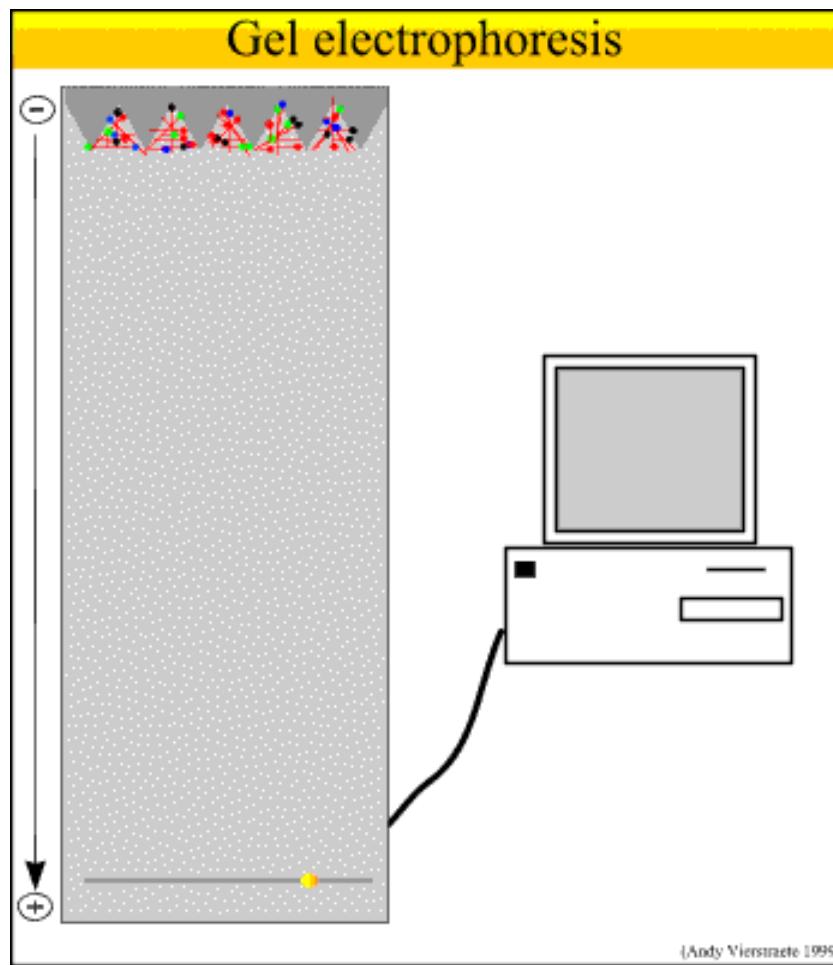
**Four dideoxynucleotides fluoresces a different color when illuminated by a laser beam and an automatic scanner provides a printout of the sequence.**



# Separation of Molecules with Electrophoresis



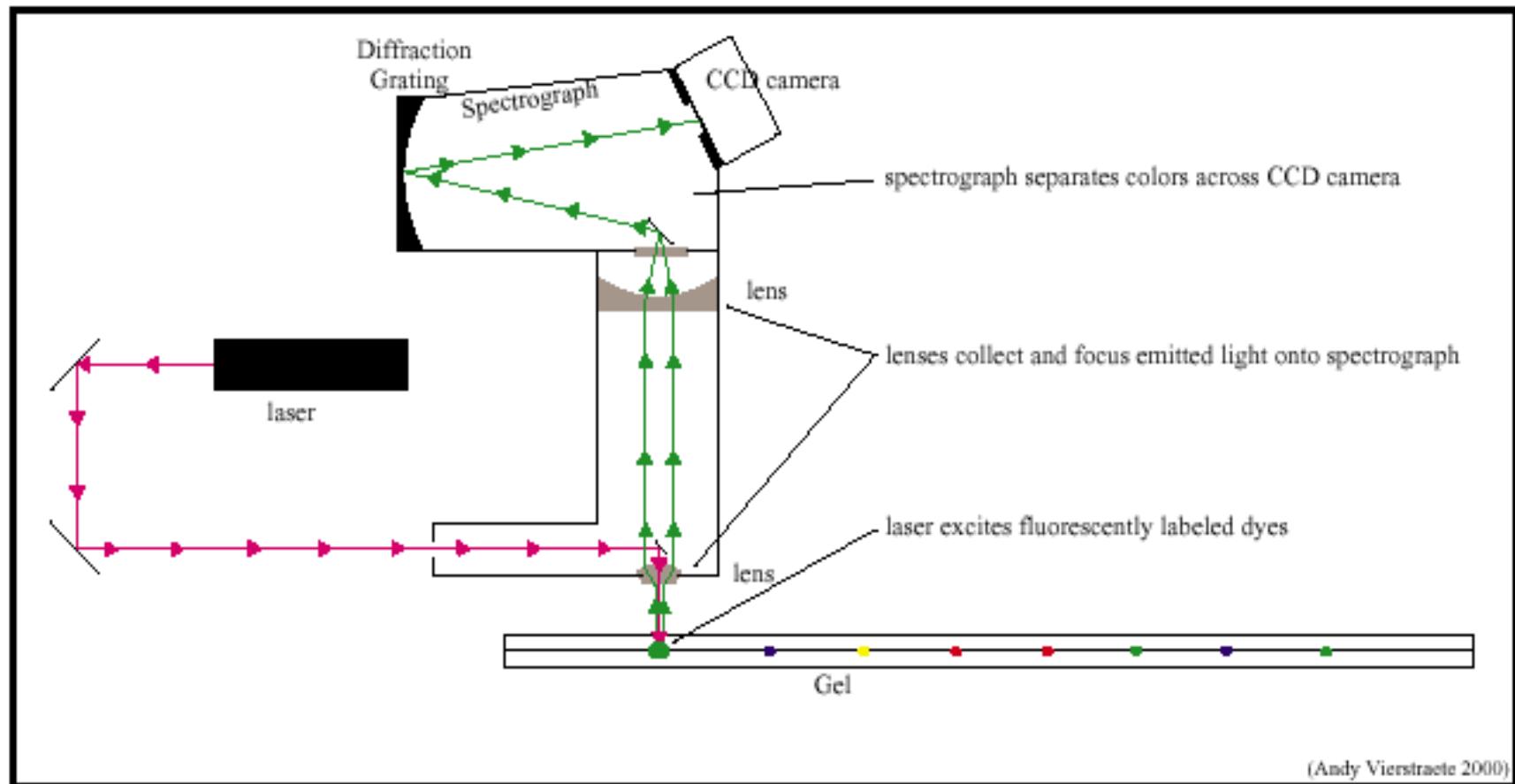
# Separation of the Molecules with Electrophoresis



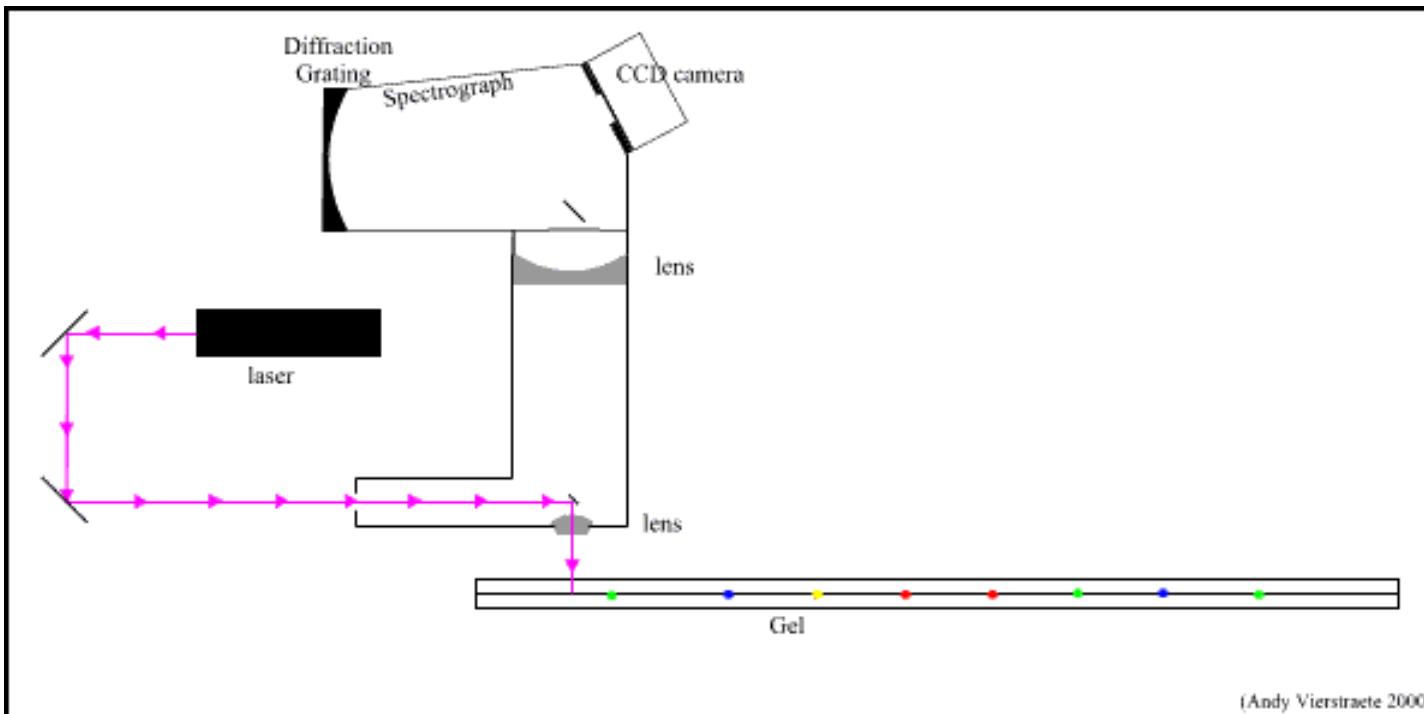
# PCR Sequencing

## III Detection on an automated sequencer:

Fluorescently labeled fragments that migrate through the gel pass a laser beam at the bottom of the gel.



# Scanning & Detection System on a Sequencer



# PCR Sequencing

**Plot of the colors detected in a 'lane' of the gel (one sample),  
scanned from smallest fragments to largest.**

**The computer interprets the colors by printing the nucleotide sequence across the top of the plot.**

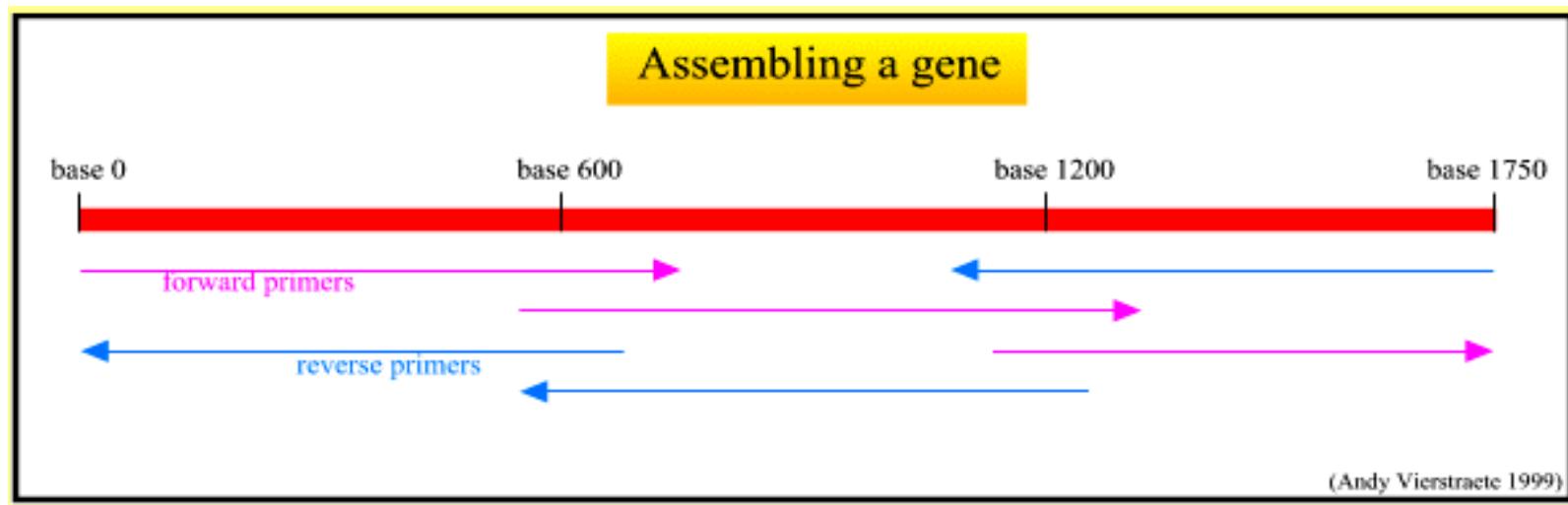
# PCR Sequencing

## IV Assembling the sequenced parts of a gene:

**For publication, a gene sequence has to be confirmed in both directions using forward & reverse primers**

Since it is only possible to sequence ~700-800 bases in one run, a gene of, say, 1800 bases, is sequenced with **internal primers**.

- the sequenced fragments are assembled using a computer program to obtain complete gene sequence.



# Genome Sequencing

# Genome Sequencing

By Sanger's method, we can sequence a fragment of DNA ~ 1000bp long.

But what about longer pieces?

Human genome is 3 billion bases long, arranged on 23 pairs of chromosomes.

Sequencing machine reads just a drop in the ocean!

# Genome Sequencing

**Solution:** Break the entire genome into manageable pieces and sequence them.

Two approaches were used for sequencing Human genome:

- Publicly funded Human Genome Project (HGP) – **clone-by-clone** or hierarchical shotgun sequencing method
- Privately Funded Sequencing Project - Celera Genomics – **whole genome shotgun** sequencing method

# Genome Sequencing

## Hierarchical shotgun sequencing approach:

- genomic DNA is cut into pieces of about 150 Mb
- inserted into BAC vectors,
- transformed into *E. coli* where they are replicated and stored.

BAC inserts are isolated & mapped to determine the order of each cloned 150 Mb fragment - referred to as the **Golden Tiling Path**

*Begun formally in 1990, Human Genome Project was a 13-yr effort coordinated by the U.S. DAE and NIH.*

*- completed in 2003*

# Genome Sequencing

Each BAC fragment in the **Golden Path** is

- fragmented randomly into smaller pieces,
- each piece is cloned into a **plasmid** and sequenced on both strands.

These sequences are aligned so that identical regions overlap.

Contiguous pieces are then assembled into finished sequence once each strand had been sequenced about **5** times to produce **10× coverage** of high-quality data.

# Genome Sequencing

## Whole genome shotgun sequencing (WGS)

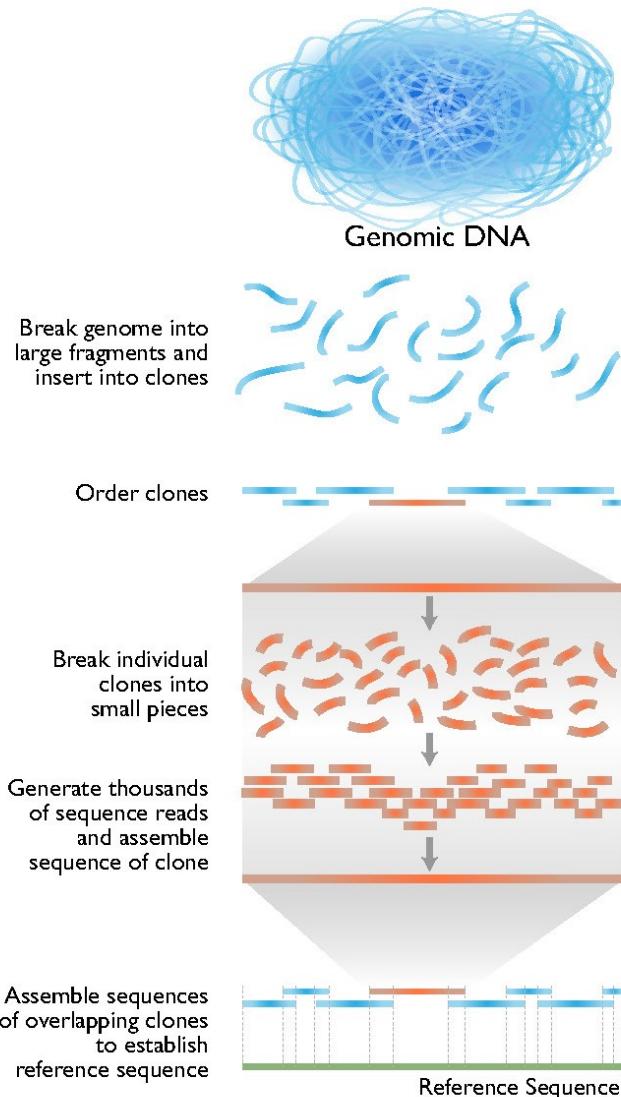
- method developed and preferred by Celera Genomics
- skips the entire step of making libraries of BAC clones

Blast apart entire human genome into fragments of 2 - 10 kb and sequence them.

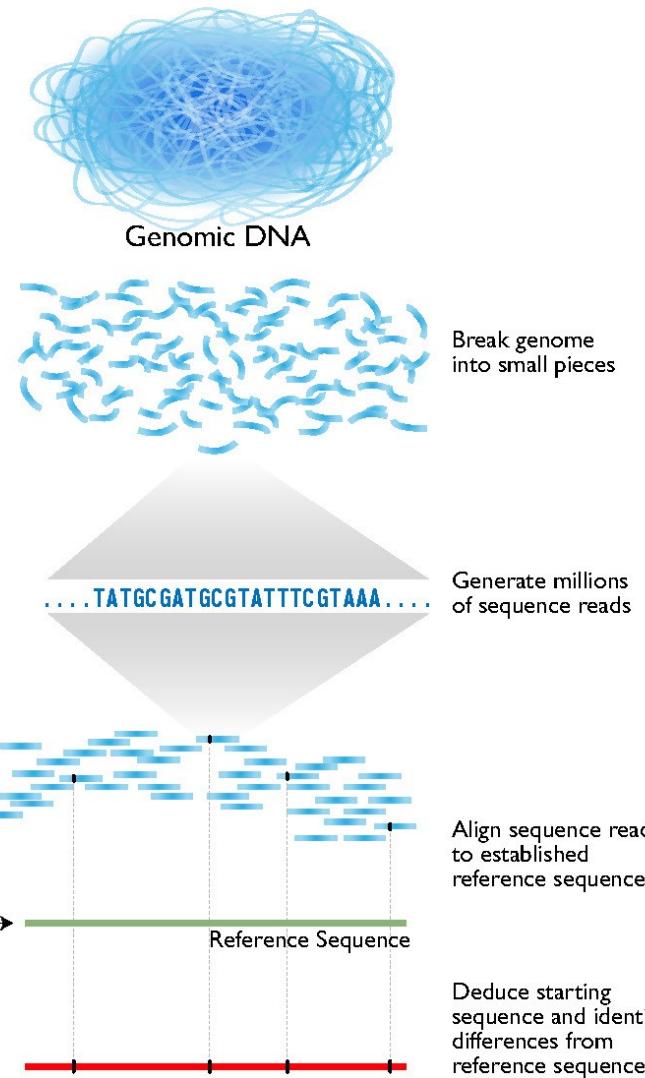
Challenge is then to assemble these fragments into the whole genome sequence.

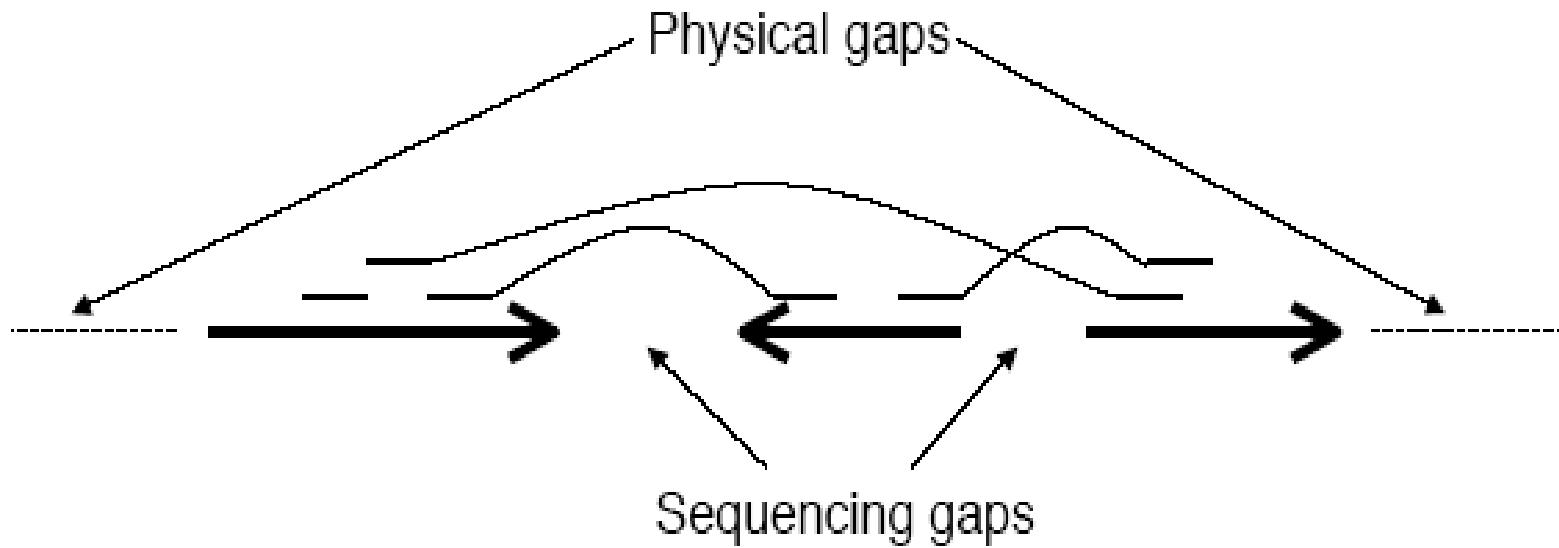
# Human Genome Sequencing

## Generating a Reference Genome Sequence (e.g., Human Genome Project)



## Generating a Person's Genome Sequence (e.g., Circa ~2016)





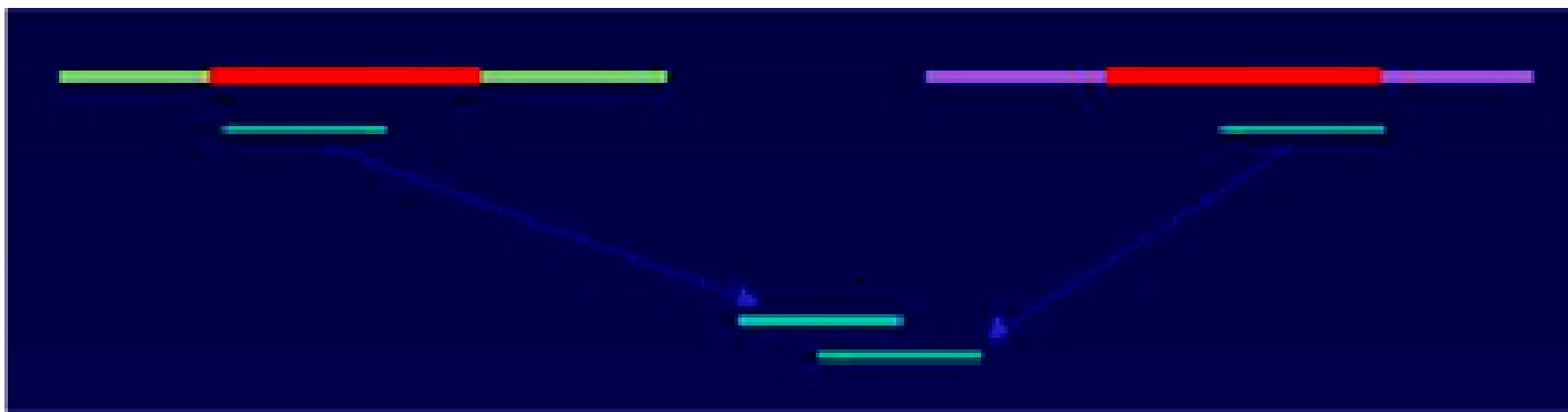
**sequencing gap** - we know the order and orientation of the contigs and have at least one clone spanning the gap

**physical gap** - no information known about the adjacent contigs, nor about the DNA spanning the gap

# Whole Genome Shotgun Method

What makes the task of assembling the genome fragments especially challenging

- repeats in the genome (~ 50% in human genome).



Because of the various ways a fragment could align with a repeat, and the different areas adjacent to the repeats in the original genome, assemblers need to be designed so as not to incorrectly join fragments

# Whole Genome Shotgun Method

**Adding to the challenge is the sheer computational complexity of the task.**

e.g., human genome is 3 billion base pairs long and if the length of one read is **500 bps** and the desired coverage is **10x**, then  **$6 * 10^7$**  reads would be required:

$$\text{GenomeLength} * \text{DesiredCoverage} / \text{ReadLength} = \text{RequiredReads}$$

With **60** million reads to assemble, we need algorithms that run in near linear time ( $O(n \log n)$ )

# **Whole Genome Shotgun Method**

**Which method is better?**

**Depends on the size and complexity of the genome**

**Note: Celera had access to the HGP data but the HGP did not have access to Celera data.**

**Which method is preferable for sequencing the genome of a novel coronavirus – SAR-CoV-2? Why?**

# cDNA Sequencing

# Sequencing cDNA Libraries of Expressed Genes

Two common goals in sequence analysis are

- to identify sequences that **encode proteins**, which determine all cellular metabolism, and
- to discover sequences that **regulate** the expression of genes or other cellular processes.

Genomic sequencing meets both the goals.

However, only a small percentage of the genomic sequence actually encodes proteins

# cDNA Sequencing

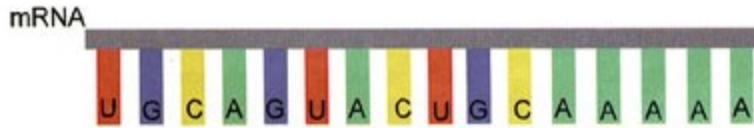
**Computational methods for analyzing genomic sequences and finding protein-encoding regions are not completely reliable**

**cDNA libraries are prepared that have the sequences of the mRNA molecules expressed in the cells, or else cDNA copies are sequenced directly by RT-PCR**

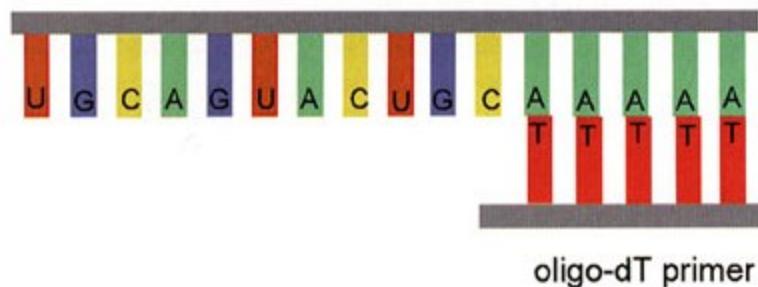
**Reverse transcription polymerase chain reaction (RT-PCR) - is used to qualitatively detect gene expression through creation of complementary DNA (cDNA) transcripts from RNA.**

# RT-PCR

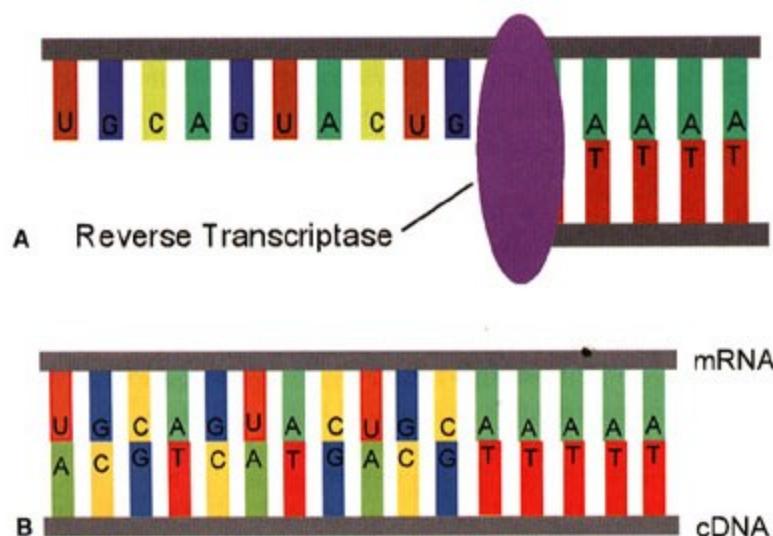
## RNA Template



## Priming for Reverse Transcription



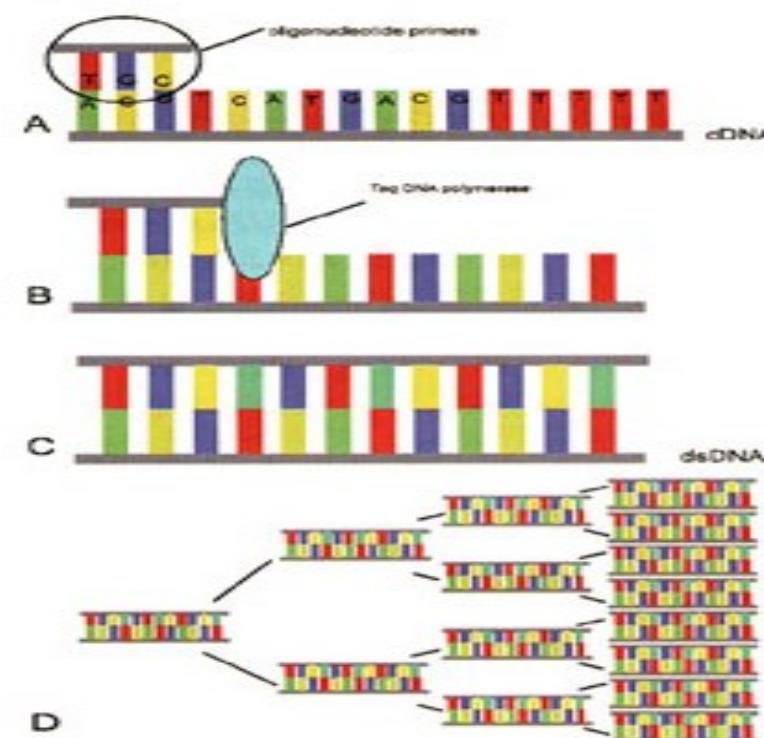
## First Strand Synthesis



## Removal of RNA



## The PCR Reaction



## **COVID-19 Testing**

**This is how the coronavirus is detected in real time RT-PCR diagnostic test, by amplifying certain regions in the viral genome:**

- Targets include the RdRp (RNA-dependent RNA polymerase) gene ORF1ab and N (nucleoprotein).**

# cDNA Sequencing

**Can all protein-coding genes of an organism be identified by cDNA sequencing?**

# cDNA Sequencing

**Can all protein-coding genes of an organism be identified by cDNA sequencing?**

**Difficulty with this approach** - a gene of interest may be developmentally expressed or regulated in such a way that the mRNA is not present

**This problem is circumvented by pooling mRNA from a variety of tissues & developing organs, or subjecting the organism to several environmental influences**

**Current gold standard for protein-coding gene annotation** is EST or full-length cDNA sequencing followed by alignment to a reference genome.

**EST – expressed sequence tag**

# EST Sequencing

An important development in computational approaches was by Craig Venter - to prepare databases of partial sequences of expressed genes, called **expressed sequence tags or ESTs**.

- which are long enough to give a pretty good idea of the protein sequence.

To identify the function of the cloned gene, translated EST sequence can be compared to a database of protein sequences - to find its homologs with known function.

Corresponding cDNA clone of the gene of interest can then be obtained and the gene completely sequenced.

# **High-throughput / Next-Generation Sequencing**

## Cost per Human Genome



13yrs, \$3 billion

S.S.

8days, \$10,000

15min, <\$1,000

**DNA sequencing beating Moore's law**

# HTS/NGS Sequencing

**High-throughput sequencing (HTS) technologies have revolutionized the way biologists acquire and analyze genomic data.**

**- massively parallel sequencing**

**HTS instruments such as**

- 454 from Roche Diagnostics,**
  - Illumina Genomic Analyzer,**
  - Applied Biosystems SOLiD System,**
  - Helico's Single-molecule sequencing platform**
  - MinION, Oxford Nanopore Technologies**
- can generate tens of gigabases per week, at a cost 200-fold less than previous methods, potentially enabling the routine sequencing of human and other genomes.**

# Sequencing Machines: Overview

	Roche GS FLX+	Illumina HiSeq 2000	SOLiD™ 4	Ion Torrent PGM
Bases per run	700Mb	600 Gb	100 GB	1 Gb
Time per run	23h	~11 days	~14 days	4.5 h
Reads per run	1 Million	6 Billion (paired-end) 3 Billion (single)	1.4 Billion	Millions
Read length	~700 bp	2 x 100 bases	2 x 50 bases	35–400 bases

**Single-molecule sequencing technology - ≥1500bp**

**MinION – 10-100Kb read lengths, high error rates (~10-15%)**

# Sequencing Machines: Overview



Roche GS-FLX

1. Pyrosequencing



Life Technologies SOLiD

3. Sequence by ligation



Illumina HiSeq

2. Sequence by Synthesis



Life Technologies Ion Torrent

4. Proton Detection



5. Nanopore sequencing

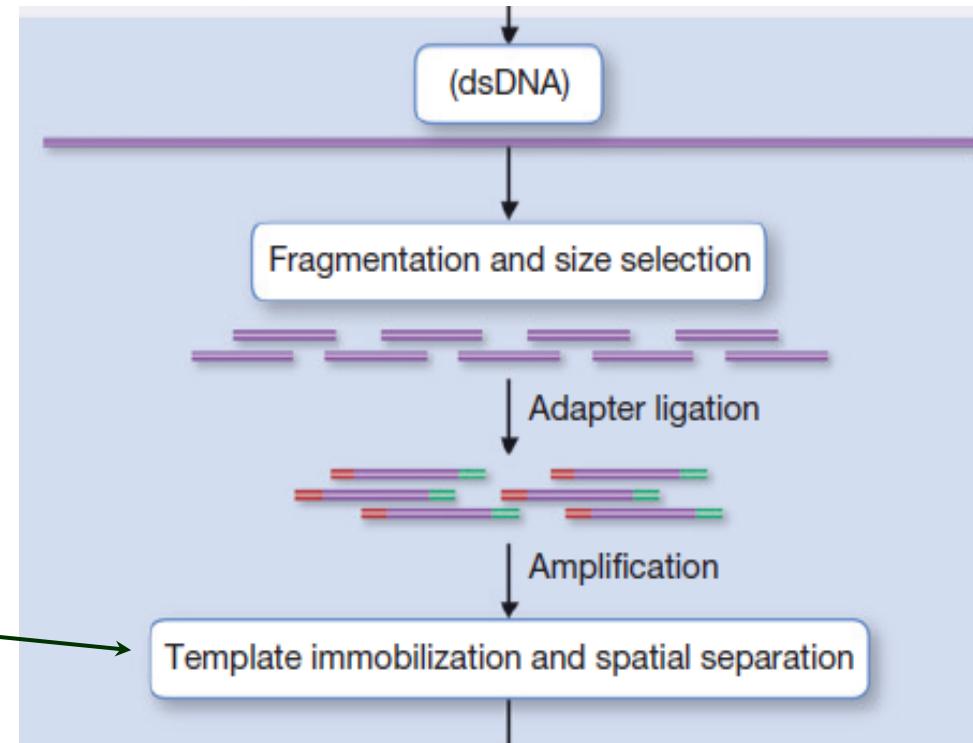
# Basic workflow: Template Generation

**Sequence library** – convert starting material into a library of sequencing reaction templates.

**Require common steps:**

- **Fragmentation**
- **Size selection**
- **Adapter ligation**

by attachment to solid surfaces or beads

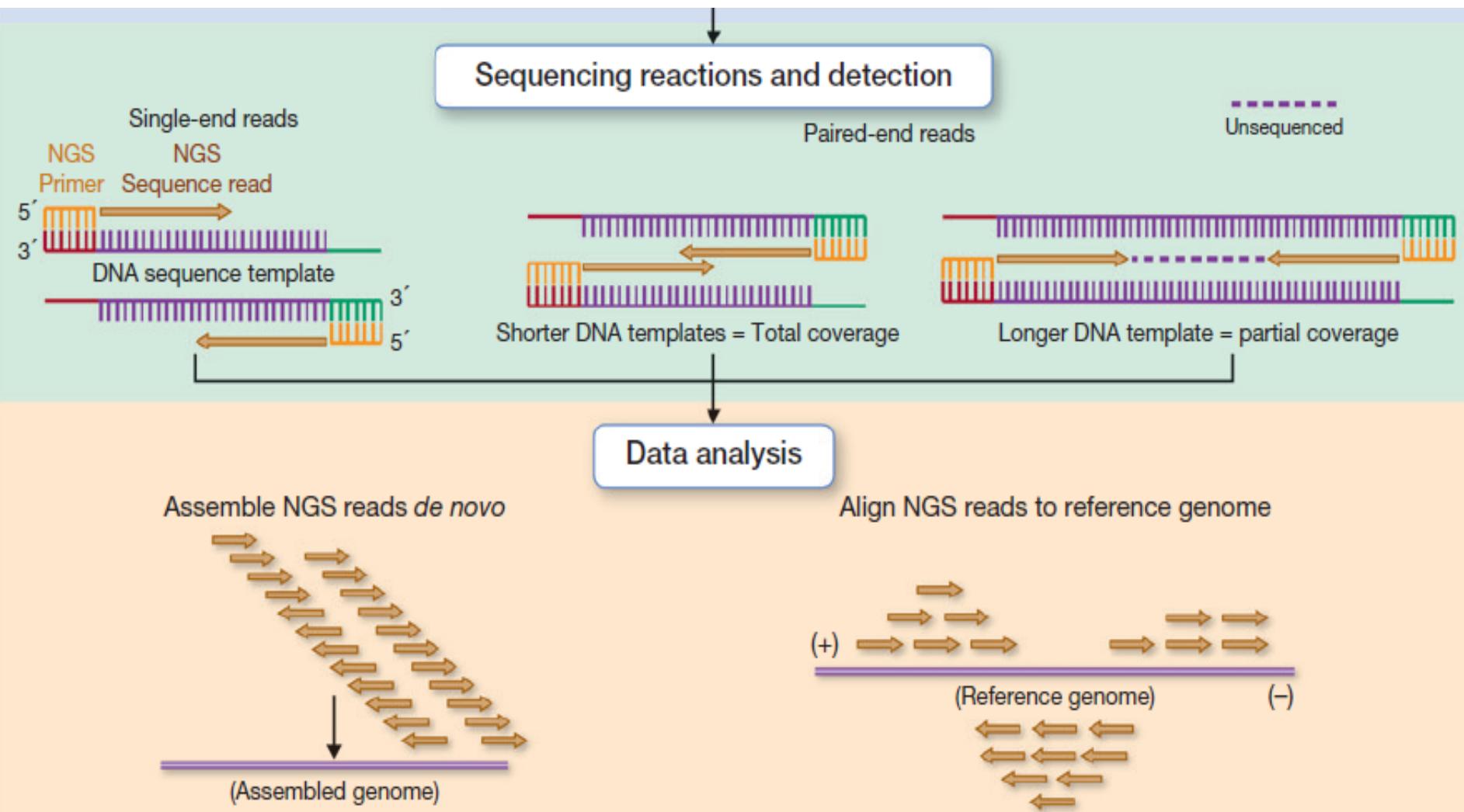


**Amplification-based** - “second-generation” sequencing technology

**Single-molecule** - “third-generation” sequencing technology

A library is either sequenced directly - Single-molecule templates, or amplified then sequenced - Clonally amplified templates

# Basic workflow: Detection & Data Analysis



# Data Analysis

The scale and nature of data produced by all NGS platforms place substantial demands on IT at all stages of sequencing, including data tracking, storage, and quality control.

- read lengths: 50 – 100bp, No. of reads: ~ GBs

Initial analysis or **base calling** - by proprietary software on the sequencing platform.

After base calling, sequencing data are **aligned** to a reference genome if available or a *de novo* assembly is conducted.

Once the sequence is aligned to a reference genome, the data needs to be **analyzed** in an experiment-specific fashion.

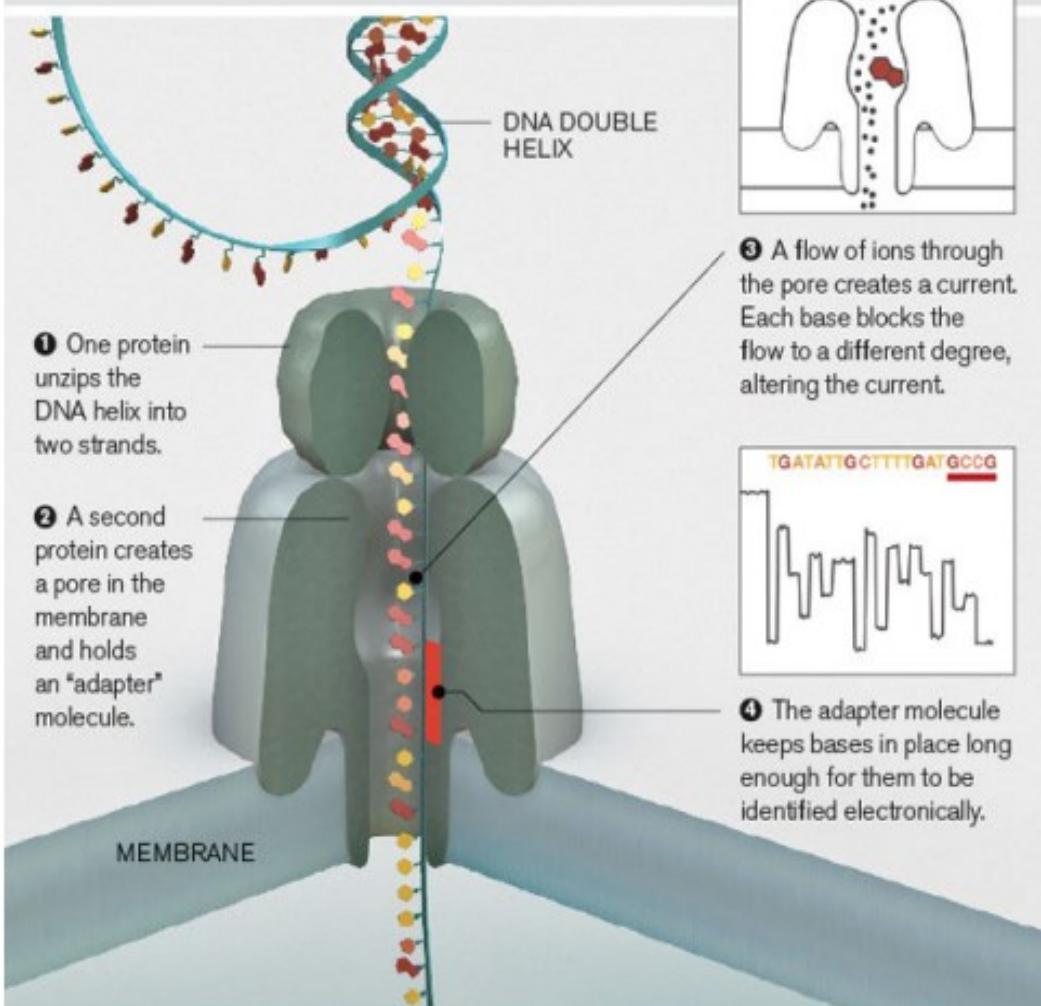
**Sequence alignment & assembly is an active area of computational research**

# Third Generation Sequencing (TGS)

- ‘Long read sequencing’ – read length:  $\sim 10 - 60\text{Kb}$
- Single molecule sequencing
- No PCR step involved
- Faster and portable
- Under active development
- e.g., PacBio Single molecule real time sequencing (SMRT) and Oxford Nanopore

# Oxford Nanopore - MinION

DNA can be sequenced by threading it through a microscopic pore in a membrane. Bases are identified by the way they affect ions flowing through the pore from one side of the membrane to the other.



# HTS Applications

genome	<p><i>de novo</i> sequencing: the initial generation of large eukaryotic genomes</p> <h3>De novo, whole-genome and targeted sequencing</h3> <p><i>whole-genome</i> resequencing: comprehensive SNP, indels, copy number and structural variations in individual human genomes</p> <p><i>targeted</i> resequencing: targeted polymorphism and mutation discovery</p>	<p>Velasco et al., 2007</p> <p>Digustini et al., 2009</p> <p>Huang et al., 2009</p> <p>Li et al., 2010</p> <p>Bentley, 2006</p> <p>Ossowski et al., 2008</p> <p>Denver et al., 2009</p> <p>Xia et al., 2009</p> <p>Hodges et al., 2007</p> <p>Porreca et al., 2007</p> <p>Harismendy et al., 2009</p>
transcriptome	<p>quantification of gene expression and alternative splicing; transcript annotation; discovery of transcribed SNPs or somatic mutations</p> <h3>Deep sequencing of RNA transcripts</h3> <p>small RNA profiling</p>	<p>Axtell et al., 2006</p> <p>Sultan et al., 2008</p> <p>Sugabaker et al., 2008</p> <p>Jacquier, 2009</p> <p>Berezikov et al., 2006</p> <p>Houwing et al., 2007</p>
epigenome	<p>transcription factor with its direct targets</p> <h3>Deep sequencing of DNA fragments pulled down by Chip-Seq</h3> <p>genomic profiles of histone modifications</p> <p>DNA methylation</p> <h3>Deep sequencing of bisulfite-treated DNA</h3> <p>genomic profiles of nucleosome positions</p>	<p>Johnson et al., 2007</p> <p>Robertson et al., 2007</p> <p>Impey et al., 2004</p> <p>Mikkelsen et al., 2007</p> <p>Cokus et al., 2008</p> <p>Costello et al., 2009</p> <p>Fierer et. al., 2006</p> <p>Johnson et al., 2006</p>
metagenome	<p>environmental</p> <h3>Species classification by metagenomics &amp; pangenomics</h3> <p>human microbiome</p>	<p>Edwards et al., 2007</p> <p>Hubert et al., 2007</p> <p>Turnbaugh et al., 2007</p> <p>Qin et al., 2010</p>

# HTS Applications

**One of the most prominent applications of NGS is re-sequencing:**

- **whole genome resequencing**
  - **target-region resequencing**
  - **exome resequencing**
- **genome-wide analysis of single nucleotide variations and other structural variations in multiple individuals, or strains, cancer sequencing, population-based sampling of a species, migration patterns of a virus, e.g., SARS-CoV-2, etc.**

Any human individual's genome available in NCBI?

# HTS Applications

**RNA sequencing** – has several applications, including RNA expression, *de novo* transcriptome sequencing for non-model organisms and novel transcript discovery

*viz.*, mRNAs, noncoding RNAs, small RNAs, miRNA

For RNA and microRNA expression profiling, NGS has significant advantages compared to microarray methods in better quantification of common & rare transcripts.

Transcriptome - the complete set of transcripts in a cell, and their quantity, for a specific developmental stage or physiological condition.

# NGS Applications

**Epigenomic Analysis** – NGS technologies have been applied in several epigenomic areas, *viz.*,

- characterization of DNA methylation patterns,
- posttranslational modifications of histones,
- interaction between transcription factors and their direct targets, and
- nucleosome positioning on a genome-wide scale.

Epigenetics is the study of heritable gene regulation that does not involve the DNA sequence itself but its modifications and higher-order structures.

# HTS Applications

**Metagenome Sequencing** – sequencing the bacterial 16S rRNA gene across a number of species, for studying phylogeny and taxonomy, particularly in diverse metagenomic samples

e.g., cataloging human gut microbial genes by metagenomic sequencing (Qin *et al*, 2010).

~ 570Gb of sequence data from 124 individuals was generated, assembled and characterized 3.3 million non-redundant microbial genes.

This helped scientists, for the first time, to define the minimal human gut metagenome.

Metagenomics involves genomic analysis of microorganisms by direct extraction of DNA from uncultured ensemble of microbial communities

# PCR Sequencing

**How would you go about sequencing SARS-CoV-2 genome, 29903 bases long?**

**What technique is used for diagnostic testing of COVID-19?**

**While sequencing a novel genome for the first time, how are primers identified?**

**Can we now answer these Qs:**

- **How is the SARS-CoV-2 genome sequenced?**
- **How does one identify the coordinates of N gene on it? i.e., how to construct a physical map of a genome?**
- **How does one select which regions in this gene would give specificity for the presence of SARS-CoV-2?\***
- **How is the specific probe regions extracted and amplified for detection?**
- **Is it possible to store the DNA sample for re-testing? How?**

## References:

1. **Concepts in Biotechnology**, ed. D. Balasubramanyam
2. **Restriction Endonucleases and DNA Modifying Enzymes**  
<http://arbl.cvmbs.colostate.edu/hbooks/genetics/biotech/enzymes/index.html>
3. **REBASE: restriction enzymes and methyltransferases**,  
**Nucleic Acids Research**, Vol. 31 (1), 418–420 (2003)

# Assignment

**Q1. It is known that the initiation of replication is mediated by DnaA, a protein that binds to a short segment within the ori known as a DnaA box, which is a 9-mer. Is it possible to say which of the sequences given below is likely to be a DnaA box sequence? Give reasons.**

**CTCTTGATC      ATGATCAAG,      TCTTGATCA, and  
CTTGATCAT**

# Assignment

**Q2. Two individuals are taken from two different species, say A and B. If the similarity between the individuals of species is A is more compared to that between species B, what can you say about the population size of the two species? Give reason to support your answer.**

# Sequence Comparison

## DotPlots & Alignments

# Computational Molecular Biology

## Genome Analysis/ Sequence Analysis

- involves identifying characteristic features in a genome

Some important analytical approaches involve:

- **Sequence Alignment** - to identify regions of similarity (Pairwise & Multiple)
- **Pattern search** - identifying repeats, motifs, CDS, etc.
- **Database search** - sequence/pattern-based search to identify similar sequences in the database
- **Statistical measures** – *ab initio* methods based on certain characteristic features of sequence (e.g., gene prediction), evaluating significance of alignment/motifs in Db search.

# Types of Mutations

- **Mutations** - are local changes in DNA content, caused by inexact replication. There are various kinds of mutations:
- **Substitution** - a wrong base is incorporated instead of a true copy. A substitution may or may not alter the protein sequence depending on the place it occurs, e.g., GUU, GUC, GUA, GUG all code for Valine, GGU – Glycine, CUU – Leucine; Val & Leu – non-polar, Gly - polar
- **Insertion / Deletion** - addition/removal of one or more bases - leads to frame-shift in coding regions.
- **Rearrangement** - a change in the order of complete segments along a chromosome, e.g., human and mouse genome are very similar – major difference being the internal order of DNA segments.

## **Mutations are important for several reasons:**

- are the source of phenotypic variation on which natural selection acts, creating species & changing them, allowing them to adapt to changes in the environment, etc.
- are responsible for inherited disorders and diseases including cancer, which involve alterations in gene.

**To understand evolution** we need to know the various types of mutations that occur, frequency/distribution of their occurrence, and their effect.

**For disease diagnosis**, we need to understand the types of mutations, their inheritance pattern, their phenotype, etc.

# Sequence Comparison

**Why compare sequences?**

## Why Compare Sequences?

**Sequencing of genomes** – has outputted an enormous amount of sequence data on new proteins

**Fundamental problem** – determination of the function of a new protein

If there is significant **sequence similarity** between a pair of sequences, we can extrapolate the **functional annotation** of one sequence to the other.

**Any other reasons for Sequence Comparison?**

# Comparison of Sequences

- **Identifying species** – as in the case of DNA barcoding
- **Phylogenetic analysis** – to find evolutionary relatedness between species
- **Genome comparison between individuals in a population** – for structural variation analysis
- **Genome comparison - diseased (e.g., cancer) vs normal cells** – for identifying variations responsible for the disease
- **Genome comparison between species** – for understanding genome evolution
- **Identifying overlapping regions** – for **genome assembly**
- **Identifying repeats, multiple copies of domains**
- **Identifying self-complementary regions in RNA sequences** for structure prediction

# Computational Methods in Sequence Comparison:

- **Graphical methods** - visual /qualitative comparison - **dotplots**
- **Sequence Alignment:** Determine residue-residue comparison to identify patterns of conservation and variability.
  - **pairwise alignment**  
e.g., identify genes/proteins belonging to the same family.
- **Database Search:** Look for homologs of query genes/proteins in the database
- **Knowledge-based prediction:** extract **empirical rules** from known examples representing **sequence-structure or sequence-function relationships.**
  - **multiple alignment**  
e.g., motif identification, identifying remote homologs

# Dot Plots - Graphical Comparison of Sequences

One of the simplest method for comparing two sequences,  
described by Gibbs & McIntyre (1970)

A dot plot is a visual representation of the regions of similarity  
within a sequence/between two sequences.

A dot plot can identify

- **regions of similarity**
  - **overlap regions**
  - **rearrangement events**
  - **internal repeats, multiple copies of domains**
  - **self-complementary regions in RNA sequences**
- Comparing  
two/more sequences
- Self-comparison

# Dot Plots

Sequence 2 along:  $b \rightarrow$  (Add a “guard” row and column.)

Sequence 1 down:  $a \downarrow$

		A	C	A	C	A	C	T	A
		A							
			•						
		G							
		C							
		A				•			
		C				•			
		A					•		
		C						•	
		A							

A dot goes where the two sequences match

Connect the dots along diagonals.

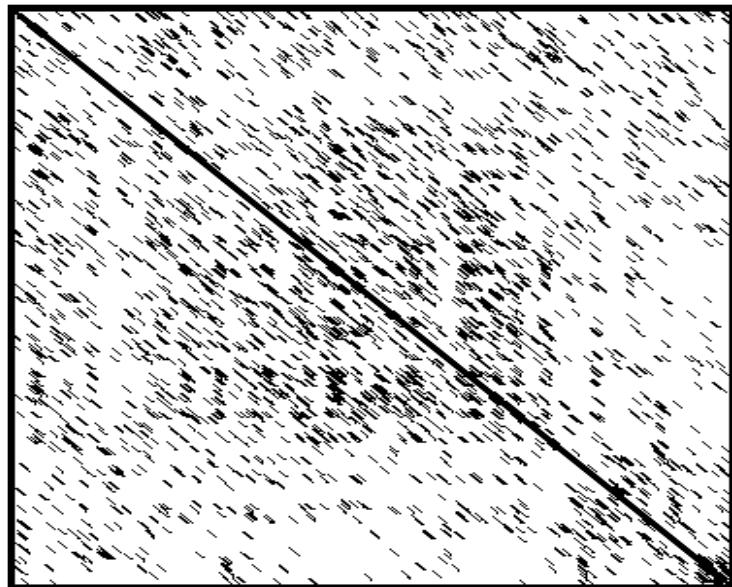
a dot is drawn for residue-residue match

Where the two sequences have substantial regions of similarity, many dots align to form diagonal lines

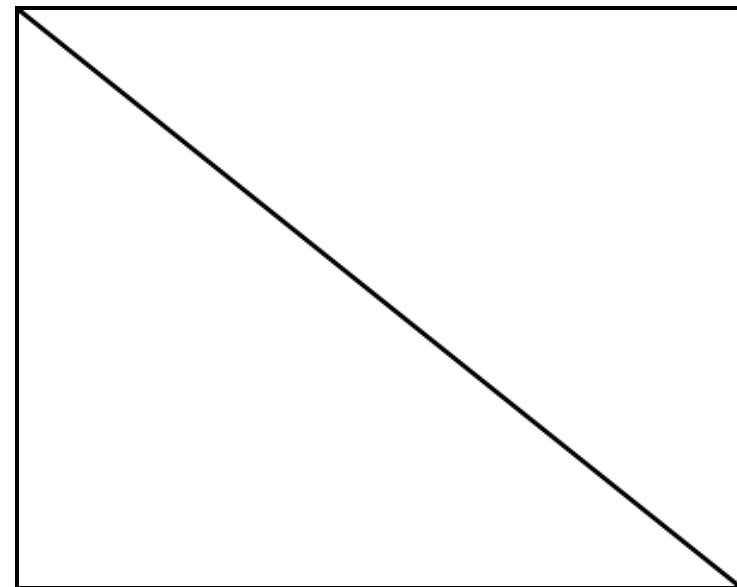
# Dot Plots

When two sequences share similarity over their entire length, a diagonal line will extend from one corner of the dot plot to the diagonally opposite corner.

Non-stringent, self-dot plot



Very stringent, self-dot plot

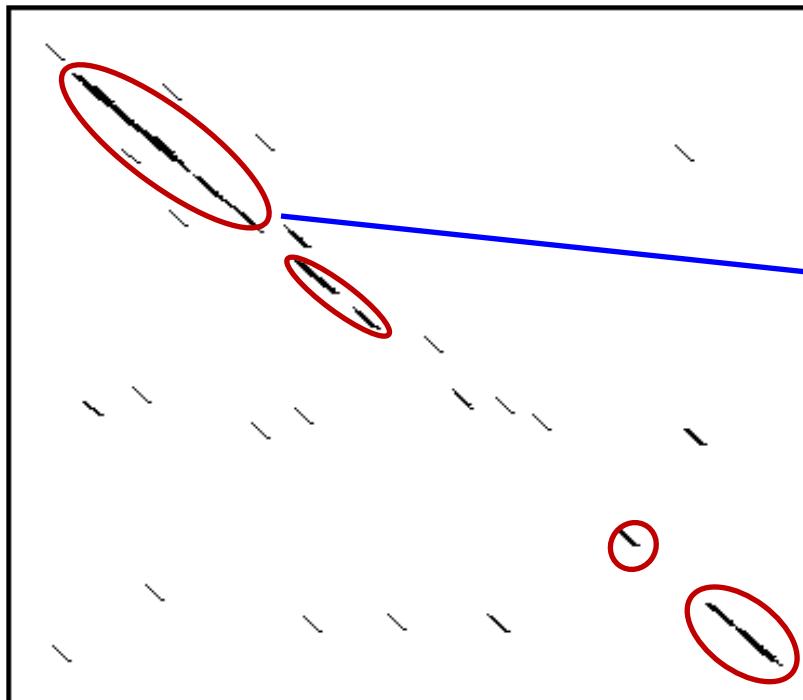


Every residue in one sequence is compared to every residue in the other sequence - nothing is missed

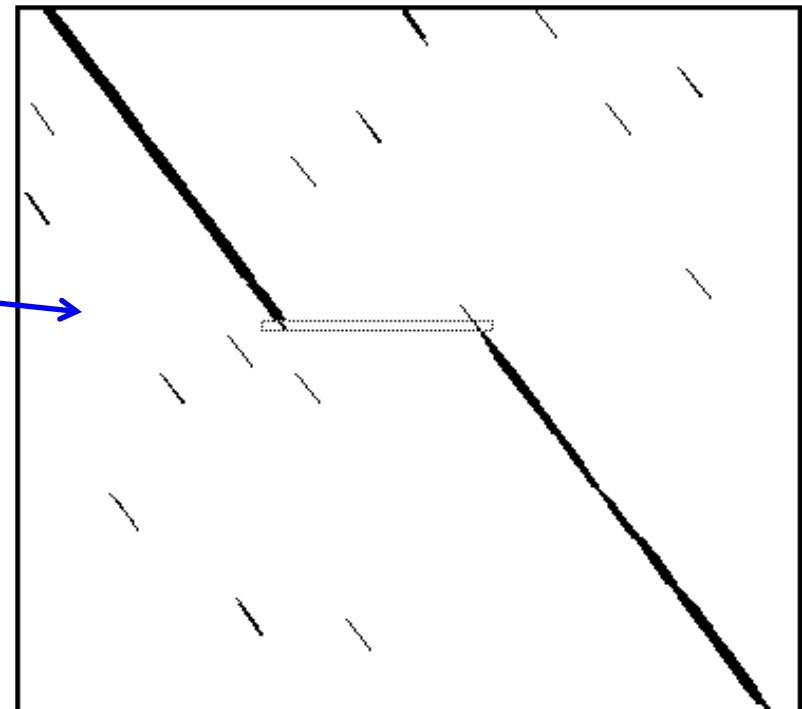
# Dot Plots

If two sequences only share patches of similarity this will be revealed by **short diagonal stretches**.

Two similar, but not identical sequences

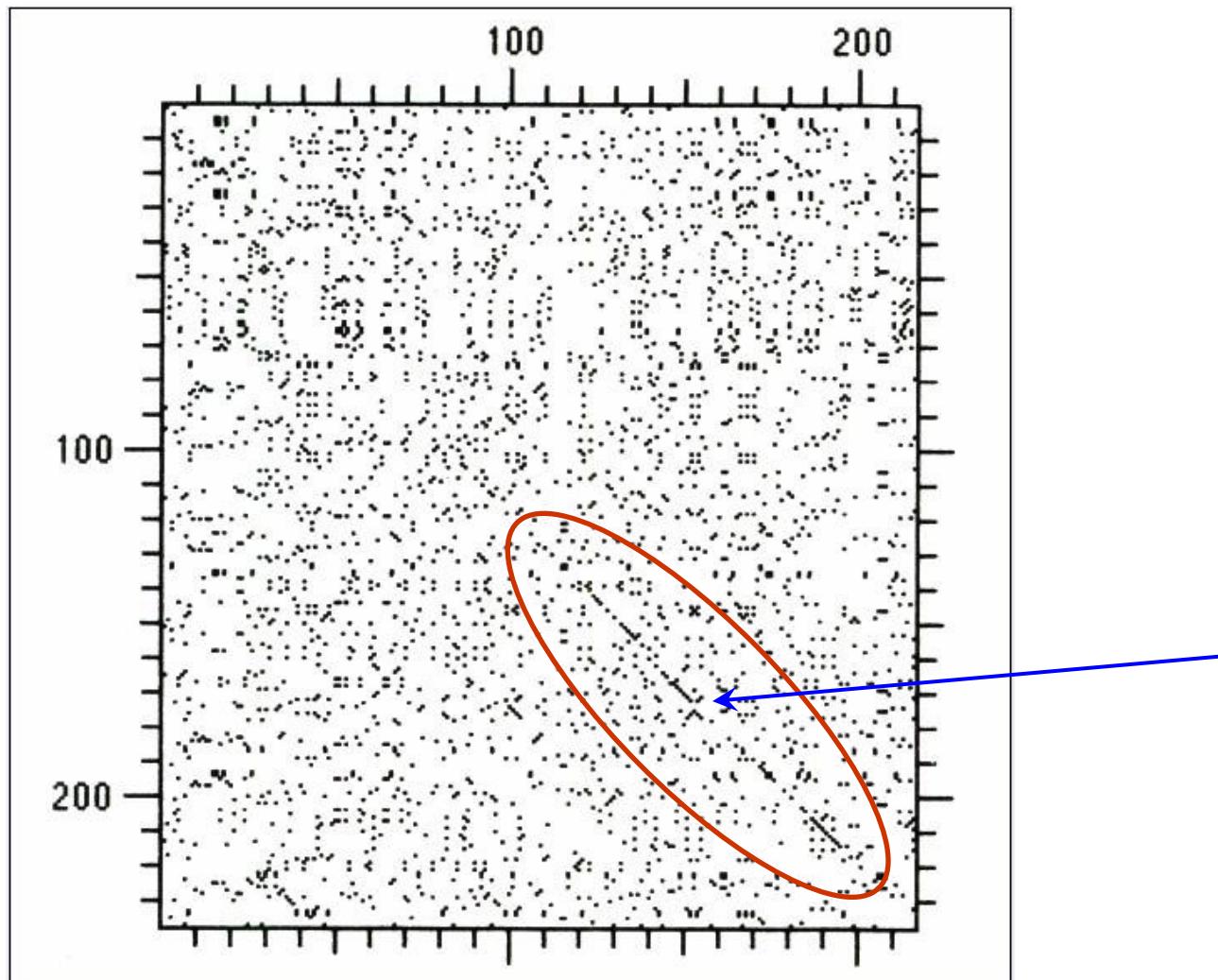


Insertion or Deletion



Homologous sequence comparison

# Dot matrix analysis of amino acid sequences of the phage $\lambda$ cI and phage P22 c2 repressors



# Dot Plots

- Major advantage of dot matrix method for finding sequence alignment - all possible matches of residues between two sequences are found, leaving investigator choice of identifying the most significant ones
  - Based on the dot plot, user can decide whether one is dealing with a case of **global** (end-to-end), **local**, or **overlapping** (similarity at the ends) similarity

## Global alignment

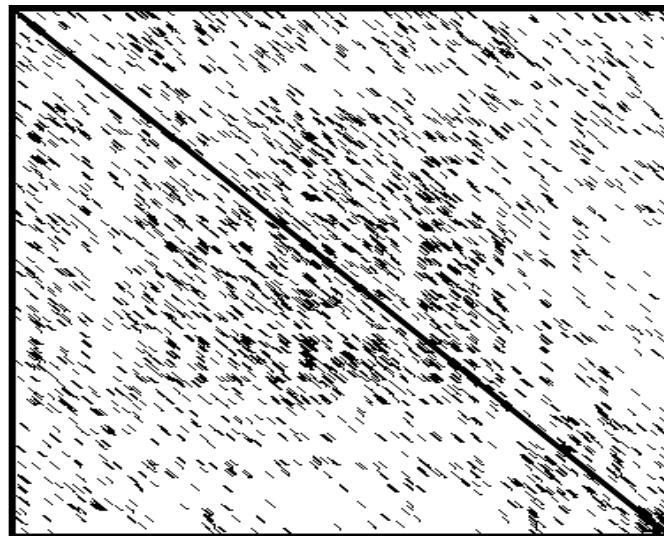
----- T G K G -----  
|||  
----- A G K G -----

## Local alignment

# Dot Plots

Detection of matching region is improved by filtering out random matches in a dot matrix - by using a **sliding window** to compare the two sequences.

Instead of comparing every base, a window of adjacent positions in the two sequences is compared and a dot is printed only if a **certain minimal number** of matches occur.



# Extensions of Dot Plots

Thus, for window analysis of dot plots we define:

- **Window:** size of diagonal strip centered on an entry, over which matching is accumulated, and
- **Stringency:** the extent of agreement required over the window, before a dot is placed at the central entry.
  - increasing window size would result in a faster search, but at the cost of sensitivity

# Dot Plots

A large window size is generally used for DNA sequences.

- typically a window size of **15** and a suitable match requirement of **10**.

For protein sequences, the matrix is often not filtered, but a window size of **2 or 3** and a match requirement of **1 or 2** will highlight matching regions.

Why?

# Dot Plots

**A large window size is generally used for DNA sequences.**

**- typically a window size of 15 and a suitable match requirement of 10.**

**For protein sequences, the matrix is often not filtered, but a window size of 2 or 3 and a match requirement of 1 or 2 will highlight matching regions.**

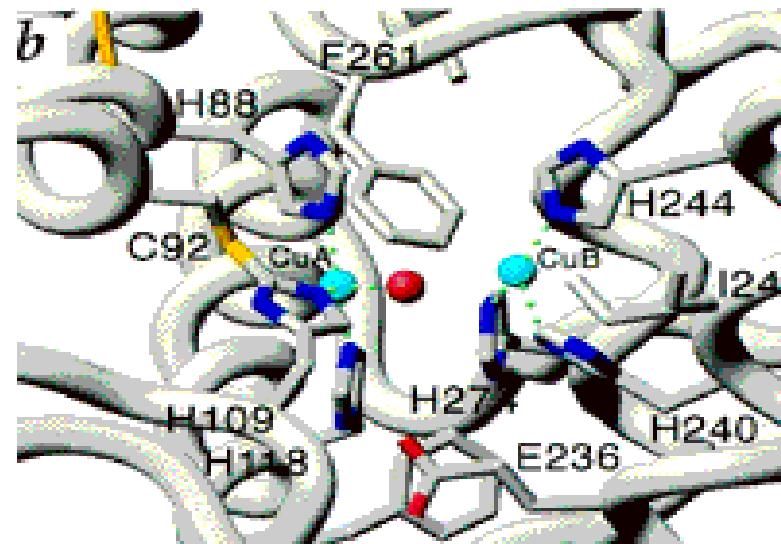
**Why?**

**- the no. of random matches is more in case of DNA due to the use of 4 nucleotides symbols as compared to 20 amino acid symbols for proteins.**

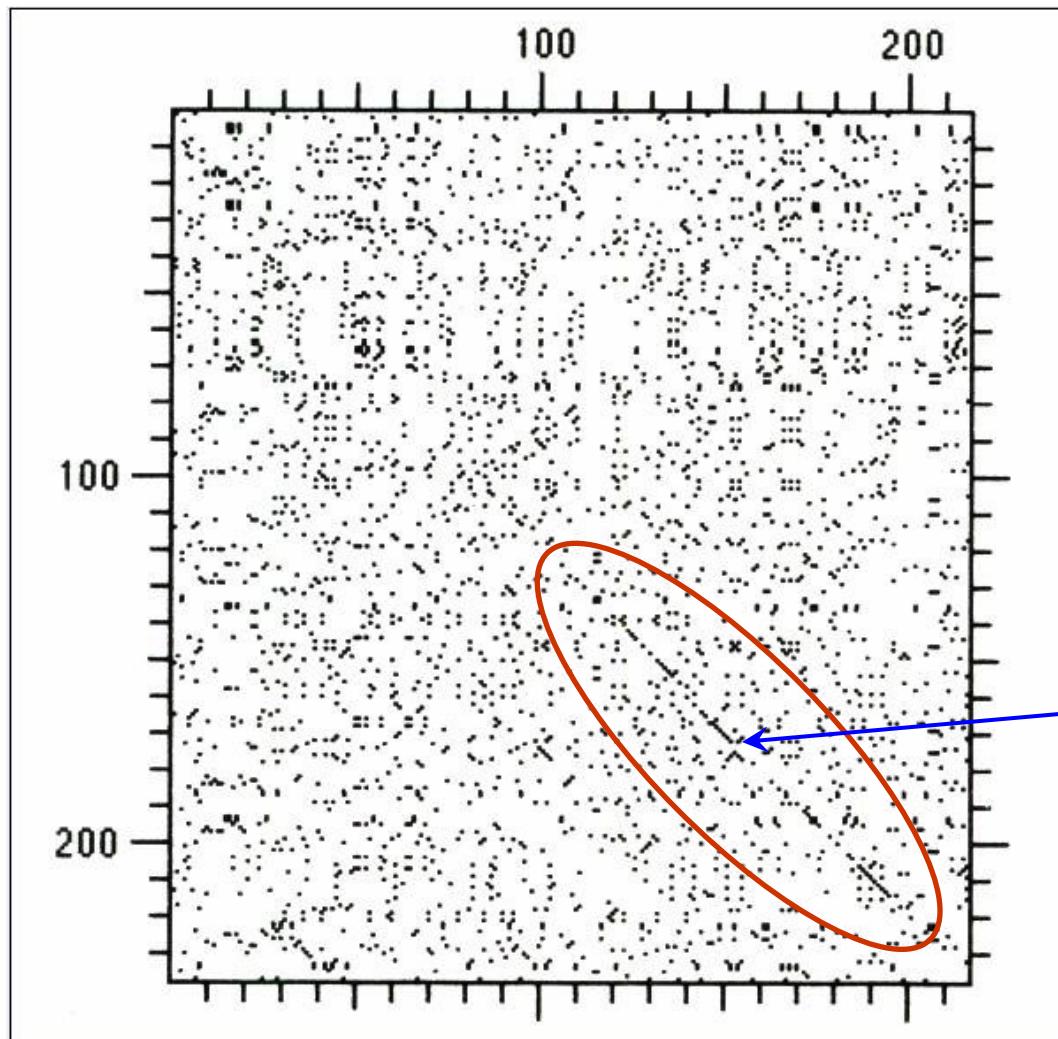
# Dot Plots

If two proteins are expected to be related but have long regions of dissimilar sequence with only a small proportion of identities, such as similar active sites,

- a large window, e.g., 20, and a small stringency, e.g., 5, should be useful for seeing any similarity.
- the reason being, residues in an active site are **not** necessarily **contiguous** in the sequence, and only the positions involved in interaction are conserved.

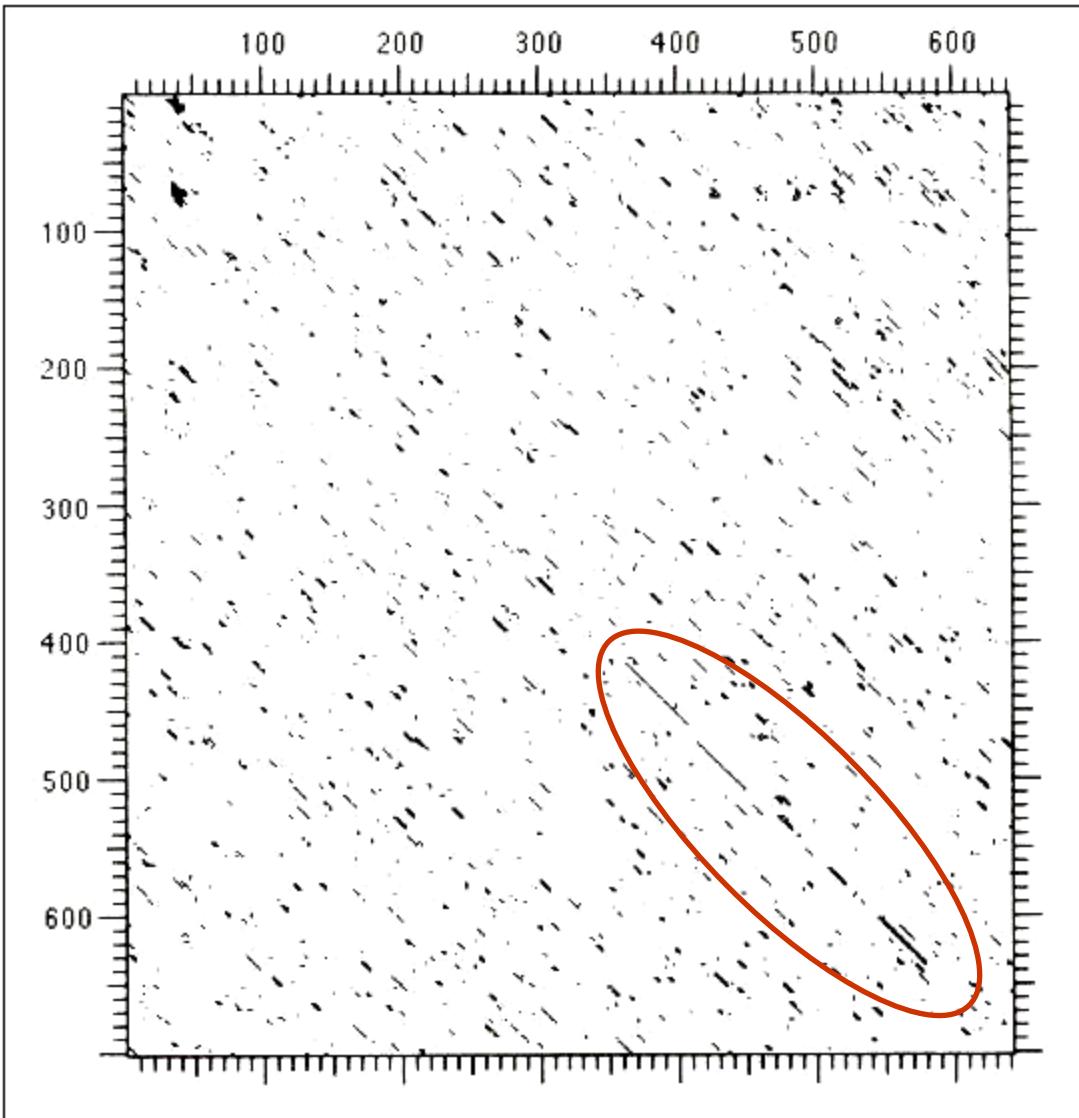


# Dot matrix analysis of amino acid sequences of the phage $\lambda$ cI and phage P22 c2 repressors



Window size: 1  
Stringency: 1

# Dot matrix analysis of DNA sequences encoding the E. coli phage $\lambda$ cI (horizontal) & phage P22 c2 (vertical) repressors



Window size: 11  
Stringency: 7

similarity in the C-terminal domains of the encoded proteins clearly seen

**There are three types of variations in the analysis of protein sequences by the dot matrix method.**

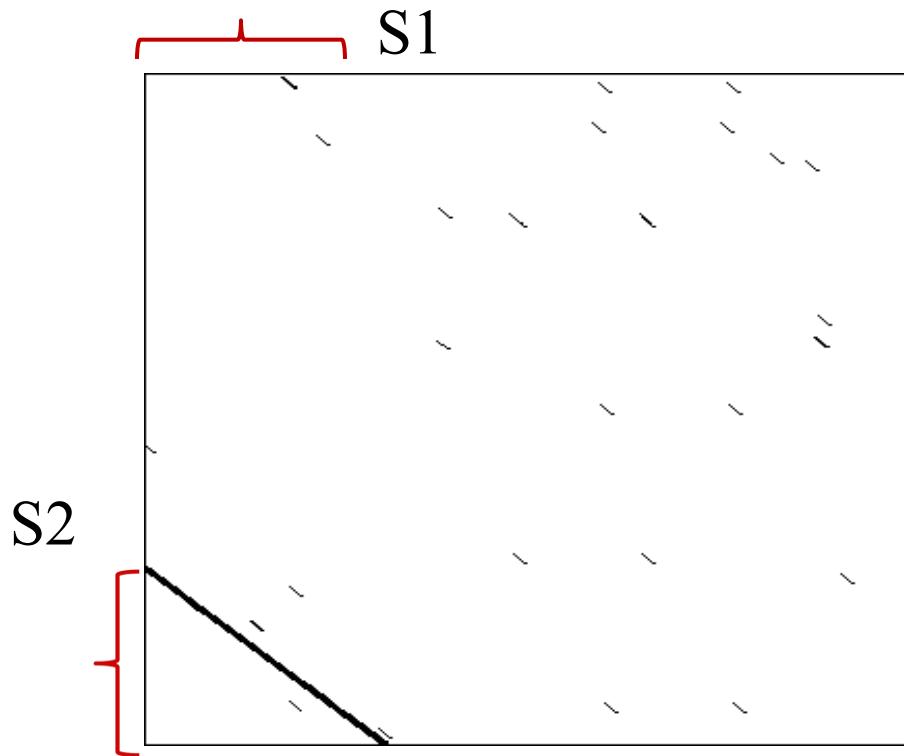
- First, chemical similarity or some other feature for distinguishing AAs may be used to score similarity.
- Second, scoring matrices may be used to provide scores for matches based on their occurrence in aligned protein families.

**When these tables are used, a dot is placed in the matrix only if a minimum similarity score is found.**

These table values may also be used in a sliding window option, which averages the score within the window, and prints a dot only above a certain average score.

**- improves the sensitivity of a dotplot while comparing protein sequences**

# Identifying Overlapping Sequences Dot Plots



**When do we expect to find overlapping sequences?**

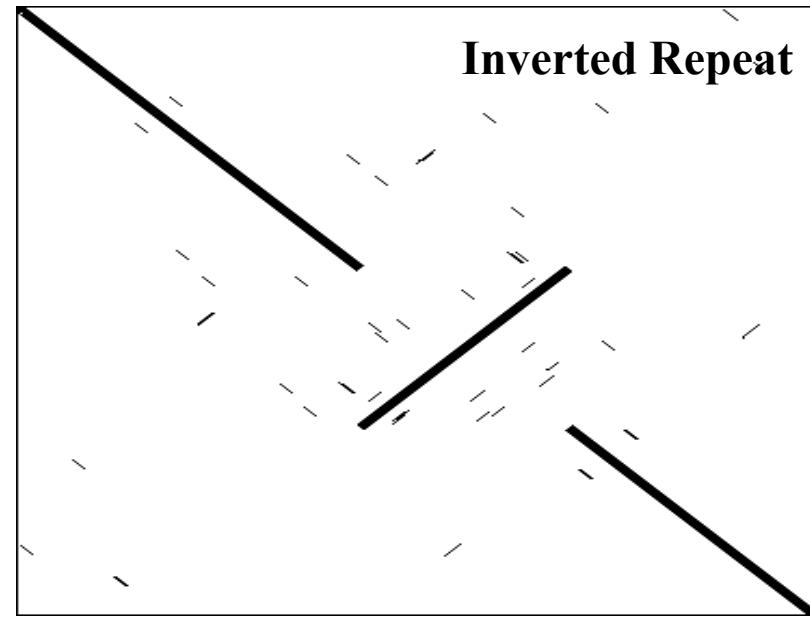
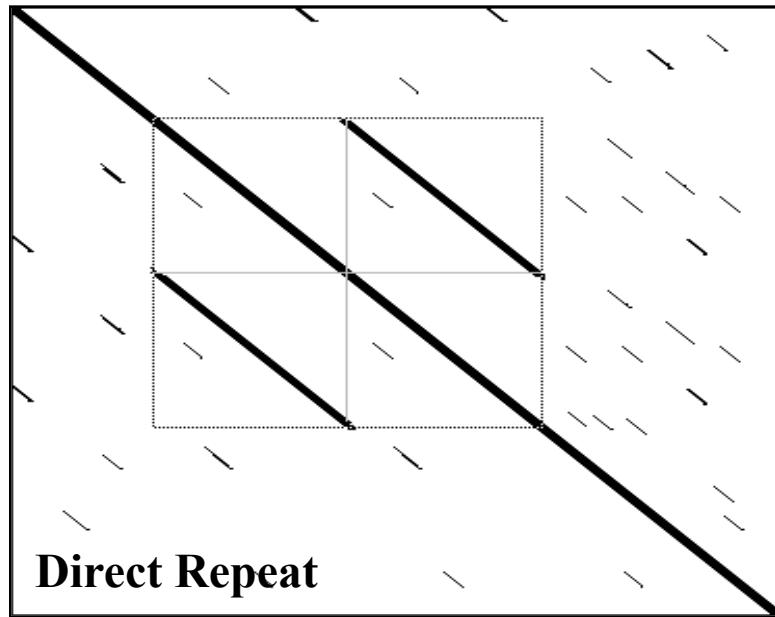
- during sequence assembly, aligning ESTs to gene / genomic sequences

# Dot Plots

- Sequences may contain regions of self-similarity termed **internal repeats**. A dot plot comparison of the sequence with itself will reveal internal repeats by displaying **several parallel diagonals**.
- Presence of repeats of the same character many times (**low-complexity regions**) appear as - **horizontal or vertical rows of dots** that sometimes merge into **rectangular or square patterns**

# Dot Plots

## Self-dot plot of a tandem duplication



We can compare a sequence to itself - it reveals repeat regions in the sequence

# Sequence Repeats

Identifying direct and inverted repeats within sequences using Dot matrix analysis.

Sequence is aligned **against itself** and the presence of repeats is revealed by rows of dots **parallel** to the diagonal

	A	G	G	C	G	C	G	C
A	•							
G		•	•		•		•	
G		•	•		•		•	
C				•		•		•
G				•		•		•
C				•		•		•
G				•		•		•
C				•		•		•

	G	A	T	T	A	G
G	•					•
A		•			•	
T			•	•		
T			•	•		
A		•			•	
G	•					•

# Repeats of a Single Sequence Symbol

A dot matrix analysis can also reveal the presence of repeats of the same sequence character many times.

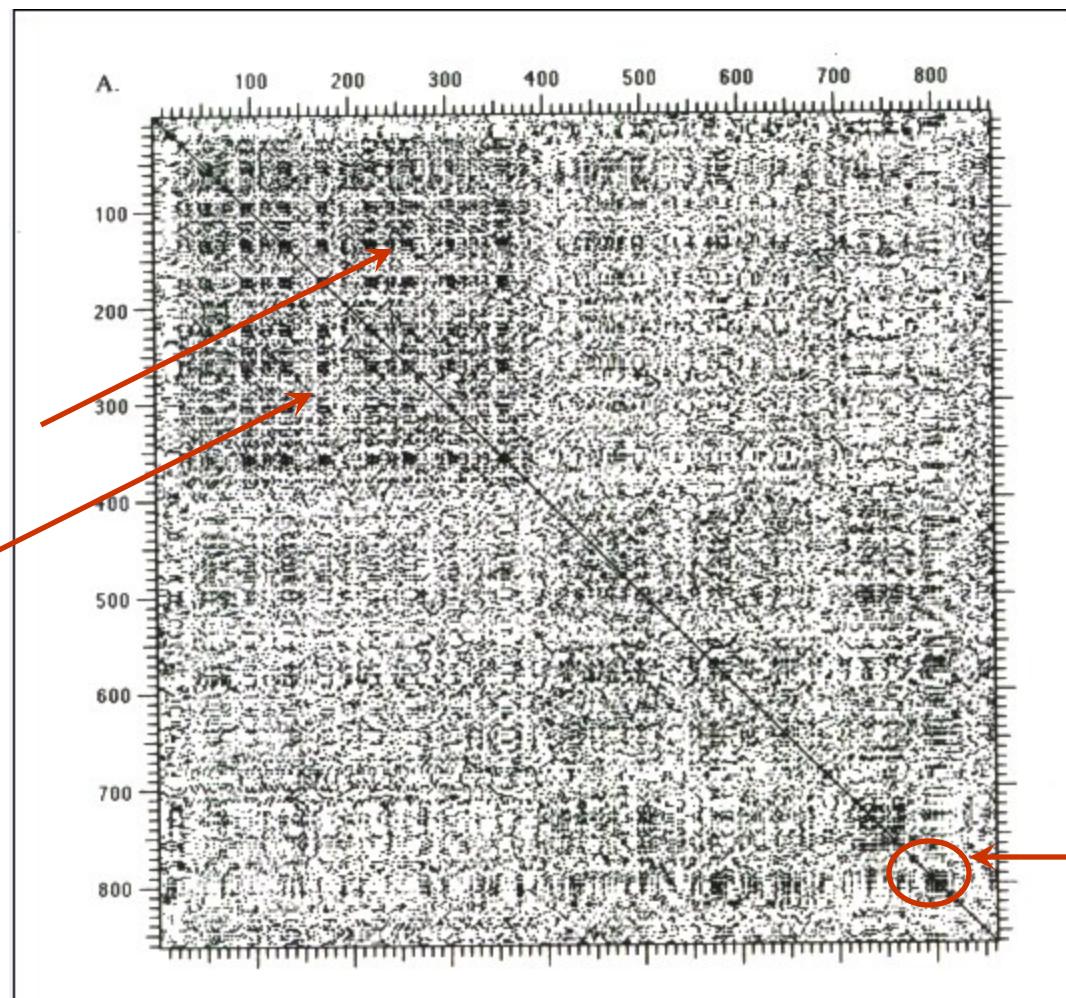
- these repeats become apparent on the dot matrix as horizontal or vertical rows of dots, merging into rectangular or square patterns.
- as seen in the lower-right regions of the dot matrix of the human LDL receptor

Occurrence of such repeats of the same character increases the difficulty of aligning sequences as they create alignments with artificially high scores

- Mask these repeats during database searches

Programs: DUST (DNA), SEG (Protein)

# Dot matrix analysis of the human LDL receptor against itself

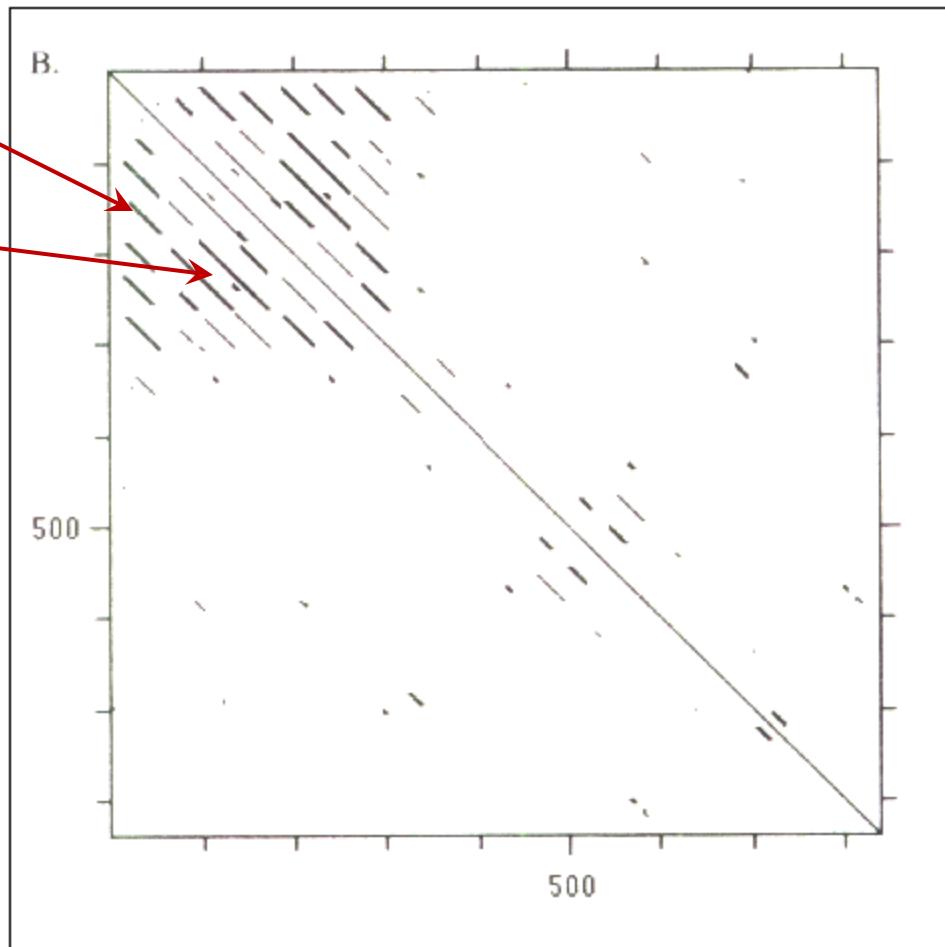


Window size: 1  
Stringency: 1

Low-complexity  
region

## Dot matrix analysis of the human LDL receptor against itself

Repeats of  
different  
lengths

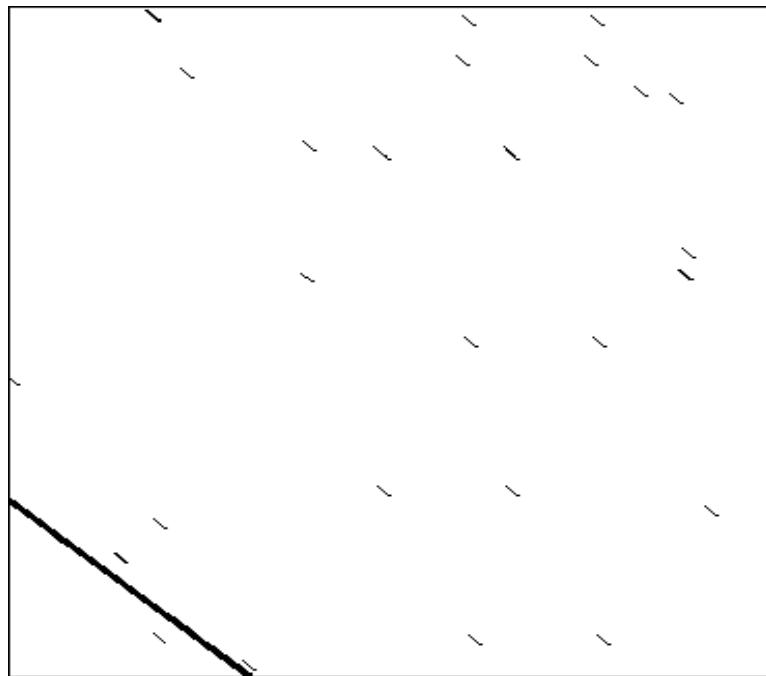


Window size: 23  
Stringency: 7

Proteins composed of multiple copies of a single domain  
can be identified by dot plots

# Dot Plots

## Overlapping Sequences

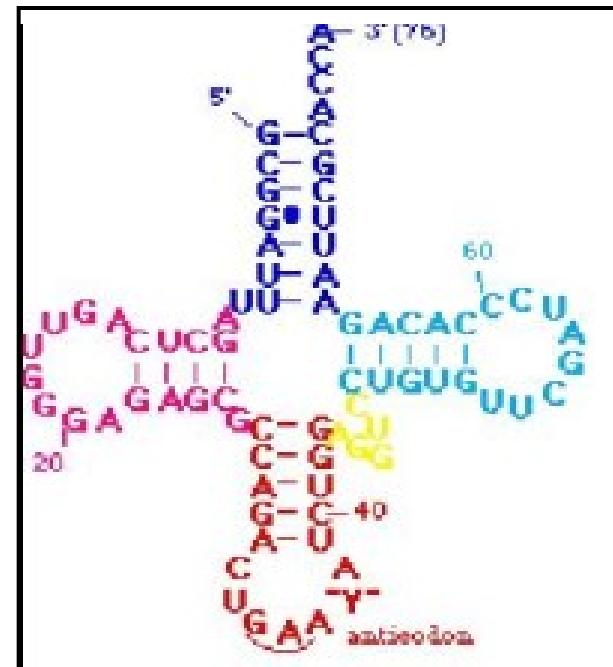


**When do we expect to find overlapping sequences?**

# Self-Complementary Regions in RNA Sequences

RNA secondary structure analysis begin with the identification of self-complementary regions

- these represent regions that can potentially self-hybridize to form RNA double strands
- once identified, the compatible regions may be used to predict a minimum free-energy structure.
- simplest way of identifying stretches of self-complementary regions in RNA sequence is a **dot plot** analysis
- there are two approaches.



# Self-Complementary Regions in RNA Sequences

**Method-1:** Sequence is listed in 5' to 3' direction along the horizontal axis and its **complementary sequence** is listed along the vertical axis, also in the 5' to 3' direction.

Matrix is then scored for identities

Self-complementary regions appear as rows of dots going from upper left to lower right.

For RNA, these regions represent sequences that can potentially form A/U and G/C base pairs

- G/U base pairs not included in this simple analysis because they play a less significant role in base-pairing.

G	A	U	C	G	G
C				•	
C				•	
G		•			•
A			•		
U				•	
C					•

# Self-Complementary Regions in RNA Sequences

As with matching DNA sequences, there are many random matches between the four bases in RNA, and the diagonals are difficult to visualize.

A long nucleotide window and a requirement for a large number of matches within this window are used to filter out the random matches.

# Self-Complementary Regions in RNA Sequences

**Method-2:** Alternative approach - list the RNA sequence along the horizontal axis and also along the vertical axis,

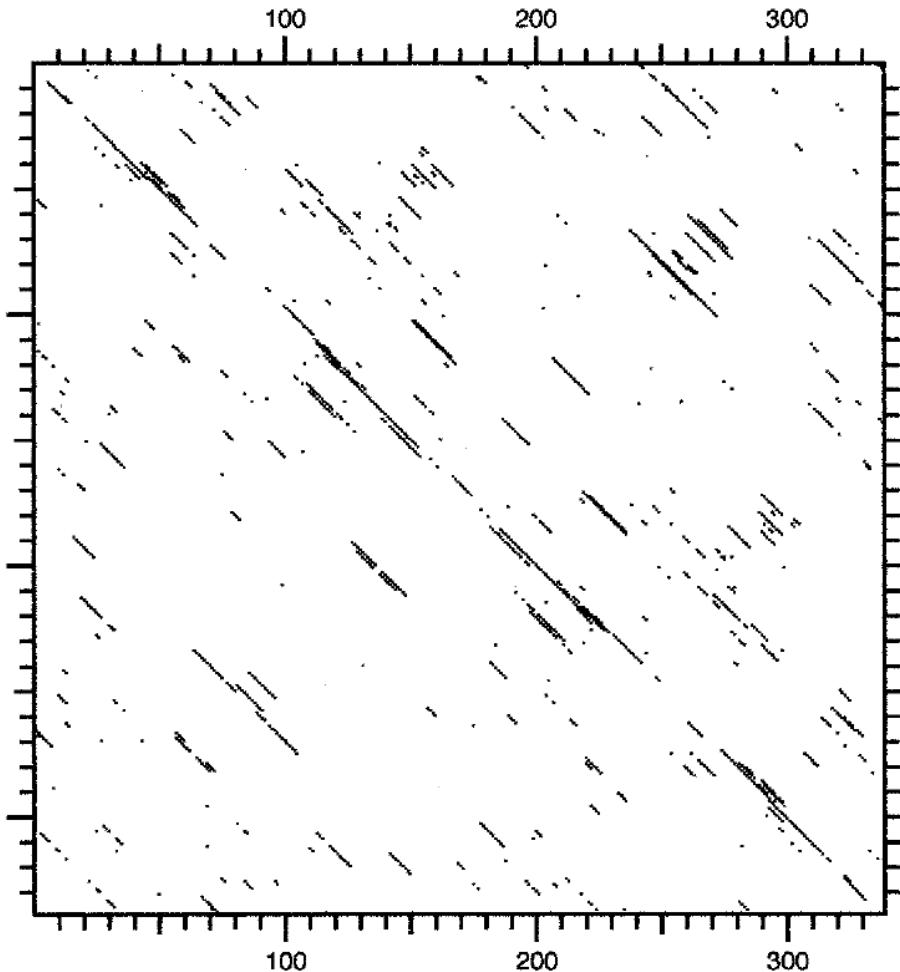
- Score matches of complementary bases G/C, A/U, and G/U instead of identities (as in the earlier method)

Diagonals indicating complementary regions will go from upper right to lower left in this matrix.

This type of matrix is used to produce an **energy matrix** for RNA secondary structure prediction.

	G	A	U	C	G	G
G				•		
A			•			
U	•					
C	•				•	•
G				•		
G				•		

# Dot matrix Analysis of Potato Spindle Tuber Viroid for RNA Secondary Structure Analysis



Window: 15  
Stringency: 11

Note: mirror image of diagonal  
from center to upper left and  
from center to lower right

# Tools for Dot Plots

- **Dotter**
- **Dottup** - EMBOSS (dotmatcher, dotpath, polydot)
- **Diagon**
- **Compare & dotplot** - GCG package

# EMBOSS

## **EMBOSS - European Molecular Biology Open Software Suite**

**- is a suite of free software tools for sequence analysis. It consists of a wide variety of programs ranging in application from database search to presentation of sequence data.**

<https://www.ebi.ac.uk/Tools/emboss/>

# dottup

**EMBOSS dottup - displays a wordmatch dotplot of two sequences**

**It looks for places where words (tuples) of a specified length have an exact match in both sequences and draws a diagonal line over the position of these words.**

**Using a longer tuple size displays less random noise, runs extremely quickly, but is less sensitive.**

**Shorter word sizes are more sensitive to shorter or fragmentary regions of similarity, but also display more random points of similarity (noise) and runs slower**

**For what tasks is this program suitable?**

# dottup

## For what tasks is dottup program suitable?

- When comparing a cDNA sequence (mRNA sequence converted to double stranded DNA sequence) to the genomic sequence, we expect an exact match, and dottup is suitable in such situations.
- Comparing very closely related sequences, when we expect a large no. of exact matches.

## Other Dot Plot programs in EMBOSS:

- **dotmatcher** – displays a **threshold dotplot of 2 Seqs**
  - a sliding window analysis along the diagonal; displays a line over the window if the sum of the comparisons (using a substitution matrix) exceeds a threshold. It is slower but much more sensitive.
- **dotpath** - Displays a **non-overlapping wordmatch dotplot of two sequences**
  - suitable for moderately distant sequences, for multiple domains in a protein, etc.
- **polydot** - Displays **all-against-all dotplots of a set of sequences**

Difference between dottup and dotpath?

**Assignment:**

**Find out the functionalities of the various dotplot programs in EMBOSS.**



About • Applications • GUIs • Servers • Downloads • Licence • User docs • Developer docs •  
Administrator docs • Get involved • Support • Meetings • News • Credits

## About EMBOSS

[Overview](#) • [Uses](#) • [FAQ](#) [Citing EMBOSS](#)

A high-quality package of free, Open Source software for molecular biology ... [more >](#)

## Applications

[EMBOSS](#) • [EMBASSY](#) • Groups [Proposed](#)

Hundreds of useful, well documented applications for molecular sequence and other analyses ... [more >](#)

## GUIs

[Jemboss](#) • [GUIs](#) • [Web](#) • [Others](#)

We support the Jemboss GUI but many others are available... [more >](#)

## Servers

[Portals](#) • [Servers](#) • [Mirrors](#) • [Misc](#)

Many EMBOSS portals, servers and mirrors are available ... [more >](#)

## Downloads

[Stable release](#) • [Developers \(CVS\) version](#) • [Getting started](#)

EMBOSS is open source software and is freely available to all ... [more >](#)

## Licence

[Licensing terms](#)

EMBOSS uses the General Public Licence (GPL) and Library GPL ... [more >](#)



[ sort alphabetically ]

## ALIGNMENT CONSENSUS

cons  
megamerger  
merger

## ALIGNMENT DIFFERENCES

diffseq

## ALIGNMENT DOT PLOTS

dotmatcher  
dotpath  
dottup  
polydot

## ALIGNMENT GLOBAL

alignwrap  
est2genome  
needle  
stretcher

## ALIGNMENT LOCAL

# DOTTUP

*(Displays a wordmatch dotplot of two sequences)*



Fields with a coloured background are optional and can safely be ignored...

[ Hide optional fields ]

### 1. SET THE PARAMETERS FOR THE RUN (OR ACCEPT THE DEFAULTS...)

input section

Select an input sequence.

Use one of the following three fields:

1. To access a sequence from a database, enter the USA path here: (dbname:entry)

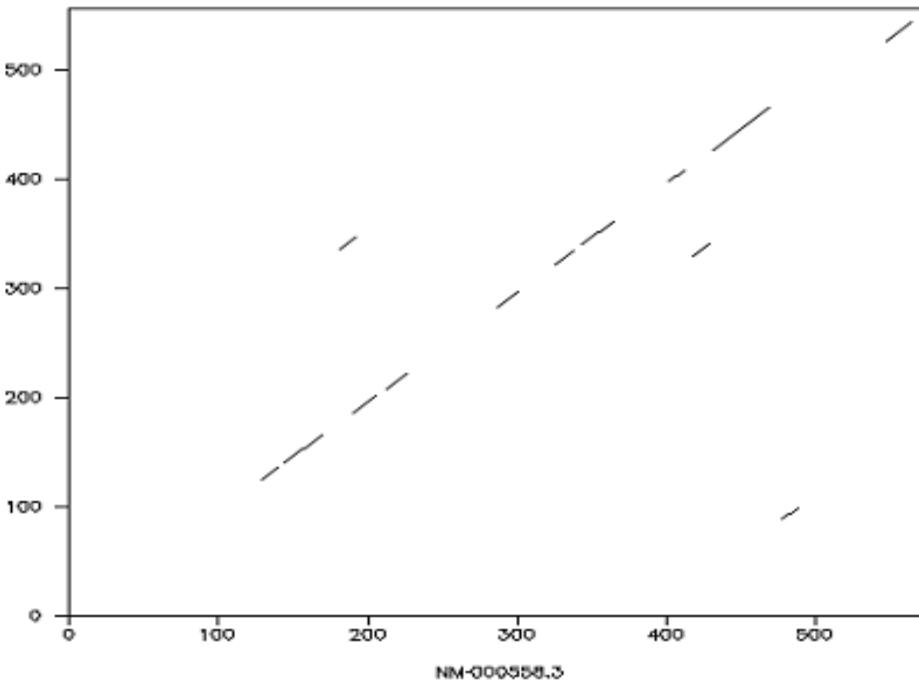
2. Or, upload a sequence file from your local computer here:

 Browse...

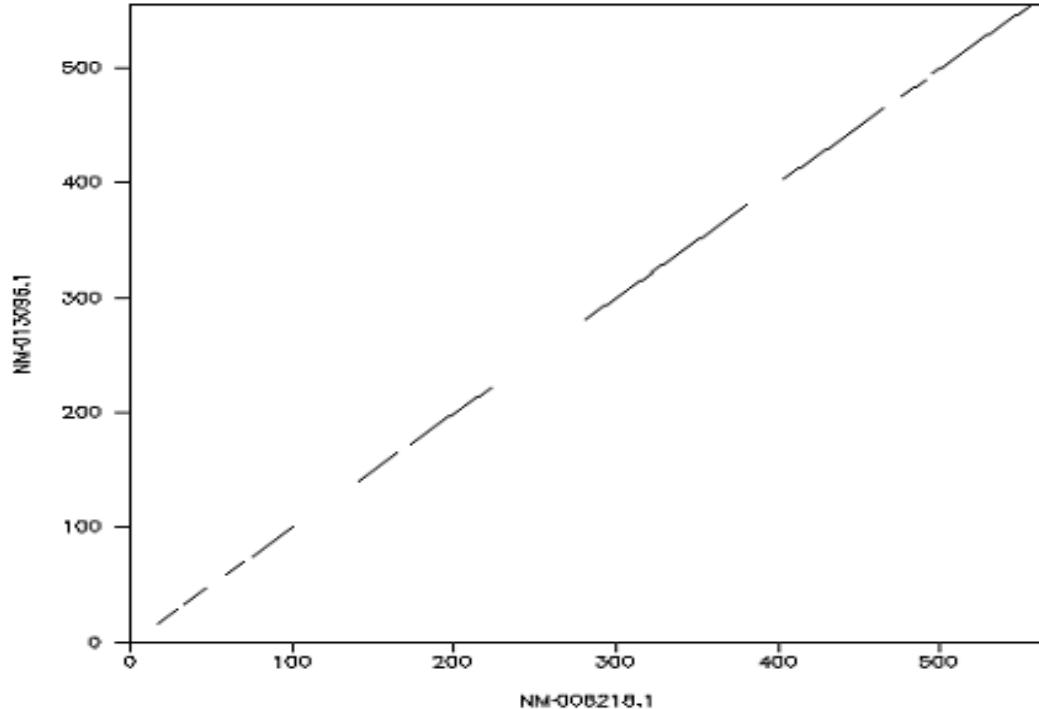
3. Or enter the sequence data manually here:

1. >gi|14456711|ref|NM\_000558.3| **Homo sapiens**  
**hemoglobin, alpha 1 (HBA1), mRNA**
2. >gi|6981009|ref|NM\_013096.1| **Rattus norvegicus**  
**hemoglobin alpha, adult chain 1 (Hba-a1), mRNA**
3. >gi|6680174|ref|NM\_008218.1| **Mus musculus**  
**hemoglobin alpha, adult chain 1 (Hba-a1), mRNA**

NM-013695.1



*Homo sapiens*  
vs  
*Mus Musculus*



*Mus Musculus*  
vs  
*Rattus norvegicus*

# Summarize

By analyzing the diagonal segments, dot plots can be used:

- to find local regions of similarity, i.e., conserved and less conserved parts of homologous proteins
  - as long diagonal lines
- to identify domain homologies between proteins not homologous overall
- to identify overlapping sequences, e.g., in sequence assembly
  - as a diagonal on a corner of the plot
- to identify internal repeats and duplications
  - as lines parallel to the diagonal
- to identify insertions and deletions
  - as breaks or discontinuities in the diagonal lines
- to identify self-complementary regions
  - in RNA secondary structure analysis

# Summarize

- For DNA sequence dot matrix comparisons, use long windows and high stringencies, e.g., 11 & 7, 15 & 11.
- For protein sequences, use short windows, e.g., 2 & 1 for window and stringency, respectively.
- When looking for a short domain of partial similarity in otherwise not-similar protein sequences, e.g. sharing similar active sites
  - use a longer window and a small stringency, e.g., 15 & 5, for window and stringency, respectively.

# Assignment

**Q1. Obtain a dotplot when a sequence A is completely contained in sequence B.**

**Q2. Obtain a self-dotplot of the sequence:**

**ATGCGCGCTG**

# Sequence Alignment

**Sequence alignment** - a scheme of writing one sequence on top of another where the **residues in one position** are deemed to have a **common evolutionary origin**

If the same letter occurs in both sequences then this position has been **conserved in evolution**.

If the letters differ it is assumed that the two **derive from an ancestral letter** (could be one of the two or neither)

# Comparison of Sequences

**Sequence alignment of two sequences basically involves**

- identifying regions of similarity, i.e., *conserved regions*, between them
- to find out if the two sequences are related or not
- enable us to extrapolate knowledge of the known sequence, or family, to the unknown query sequence

**Any other reasons for Sequence Comparison?**

# Comparison of Sequences

**Sequence alignment of two sequences basically involves**

- identifying regions of similarity, i.e., *conserved regions*, between them
- to find out if the two sequences are **related or not**
- enable us to extrapolate knowledge of the known sequence, or family, to the unknown query sequence
- identifying species, evolutionary analysis

**Statistical measures have been proposed to evaluate the significance of alignment, i.e.,**

- decide whether the alignment is more likely to have occurred because they are **related**, or just by **chance**

# Sequence Alignment

A letter or a stretch of letters may be paired up with dashes in the other sequence to signify an insertion or deletion event.

Since an **insertion** in one sequence can always be seen as a **deletion** in the other, one frequently uses the term "*indel*"

I

**B**ANANA-  
-ANANAS

Score: 10

**B**ANANA  
**P**ANAMA

II

Score: 2

# Sequence Alignment

Using a simple evolutionarily motivated scoring scheme, an alignment mediates the definition of a **distance** for two sequences:

Assign 0 to a match, some positive number (say, +1) to a mismatch and a larger positive number (say, +5) to an *indel*.

By adding these values along an alignment one obtains a **score** for this alignment:

BANANA-

- ANANAS

Score: 10

BANANA

PANAMA

Score: 2

# Sequence Alignment

A **distance function** for two sequences can be defined by looking for the alignment which yields the ***minimum score***

Using **dynamic programming** this minimization can be effected without explicitly enumerating all possible alignment of two sequences.

The idea of assigning a **score** to an alignment and then **minimizing** over all alignments is at the heart of all biological sequence alignments.

# Sequence Alignment

**Note:** one may either define a **distance or a similarity function** to an alignment.

- difference lies mainly in the interpretation of the values

A **distance function** defines 0 for a match and positive values for mismatches or gaps, and then aims at **minimizing this distance**

A **similarity function** assigns high positive values to matches and negative values to mismatches and gaps, and then **maximize the resulting score**.

Basic structure of the algorithm is the **same** for both cases.

When would you use a **distance function and a similarity function** for scoring an alignment?

# Sequence Alignment

Thus, an alignment is:

- a mutual arrangement of two sequences
- It exhibits where the two sequences are similar, and where they differ
- An 'optimal' alignment is one that exhibits the most correspondences, and the least differences
- 'Optimal' alignment need not reflect the true evolutionary relationship between two sequences, though it usually does

Similarity  $\Rightarrow$  Homology

Why is this not true?

# Sequence Alignment

## Differences between similarity and homology:

- o Similarity is simply a measure of expression how alike two sequences are
- o Homology means there is an evolutionary relationship between two sequences - there are no degrees of homology.
- o Extending this to individual residues they are 'identical' or 'similar' residues - similar implies that they share certain physicochemical properties
- o Homology cannot be observed, it is only an inference

# Differences between similarity and homology

**Identical protein sequences result in identical 3-D structures - similar sequences may result in similar structures, and this is usually the case.**

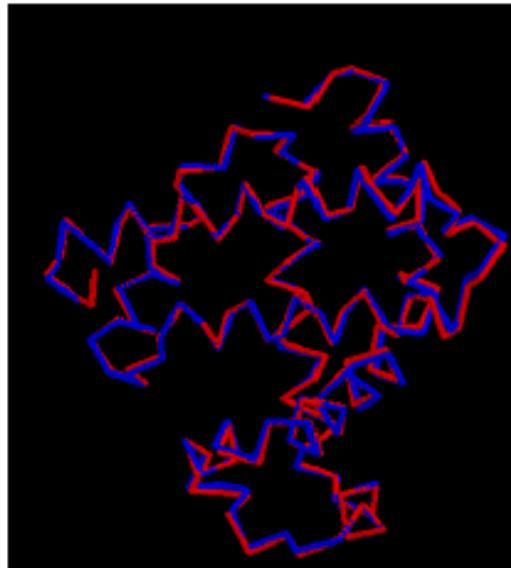
**The converse is not true: identical 3-D structures do not necessarily indicate identical sequences. It is because of this that there is a distinction between “homology” and “similarity”.**

**There are examples of proteins in the databases that have nearly identical 3-D structures, and are therefore homologous, but do not exhibit significant (or detectable) sequence similarity**

# Sequence identity and rmsd of Sperm Whale myoglobin

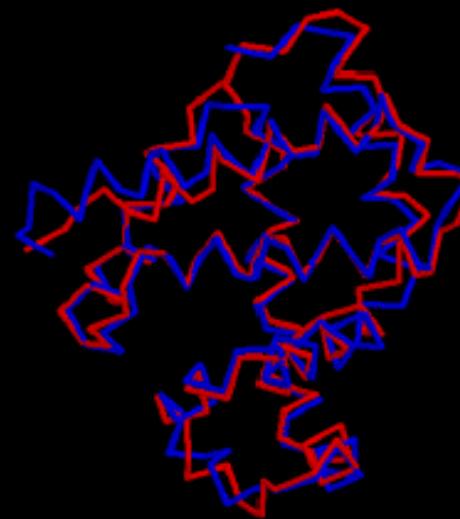
myoglobin  
pig

rmsd = 0.5 Å  
id = 86%



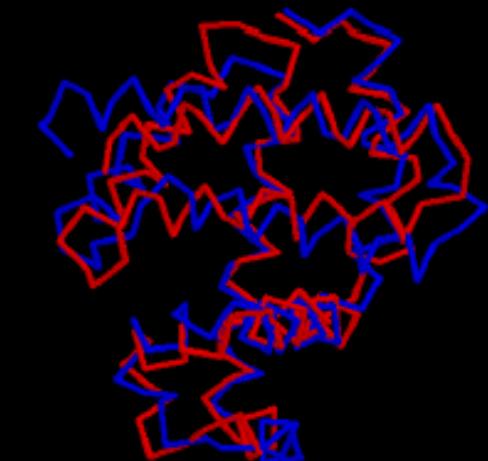
haemoglobin  
pig

rmsd = 1.5 Å  
id = 28%



globin-3  
*P. piclitum*

rmsd = 2.2 Å  
id = 18%



phycocyanin  
*F. diplosiphon*

rmsd = 3.3 Å  
id = 8%



# Summarize

- Comparison of an unknown sequence to an annotated sequence permits us to infer structural, functional & evolutionary relationships
- Wherever possible use the protein sequence since this confers more information
- Substitutions, deletions and insertions all occur as part of the natural evolutionary process
- Homology implies an evolutionary relationship between two sequences

# Pairwise Sequence Alignment

contd....

# Heuristic Alignment Algorithms

Dynamic programming algorithms though very sensitive, are **not the fastest** available sequence alignment methods

- time complexity  $\sim O(nm)$ , and in many cases **speed is the issue**, e.g., database searches

Protein database contain  $\sim 100M$  residues, this requires  $\sim 10^{11}$  matrix cells to be evaluated to search complete database for a query of length 1000; DNA Dbs are even larger (over 100 Giga bases)

At 10 million matrix cells per second this would be  $10^4$  secs, or  $\sim 3$  hrs for a single search

(for DNA Db search  $\sim 3000$ hrs = 115days)

# Heuristic Alignment Algorithms

For homology-based gene identification and annotation purposes one routinely screens genes against a database of proteins.

- need for **algorithms faster than pure DP**

Approx. methods can detect close relationships well and quickly but fail to identify very distant relationships.

# Heuristic Alignment Algorithms

Goal of such methods - to search as small a fraction as possible of the cells in the DP matrix, while still looking at all high scoring alignments.

- take a small integer  $k$ , and determine all instances of each  $k$ -tuple of residues in the probe sequence that occur in any sequence in the database

A candidate sequence in the database would contain many matching  $k$ -tuples, with equivalent spacing in probe and candidate sequences.

- perform DP only for probable homologies

# Heuristic Alignment Algorithms

For selected candidate sequences, approximate optimal alignment calculations are carried out, with the “time” and “space” saving restriction

- paths through the matrix are restricted to bands around the diagonals containing the matching  $k$ -tuples.

e.g. of heuristic approaches: BLAST, FASTA

Another computational resource that can limit DP alignment is **memory usage**, which is also of order  $\sim O(nm)$

For two protein sequences (few hundred residues long) - manageable on a desktop computers

But if one or both the sequences are genomic DNA sequences (hundreds of thousands of bases long), required memory for the full matrix can exceed machine's physical capacity.

e.g., comparison of Chimp and Human Chr 1 ( $\sim 249\text{Mb}$ ), memory requirement is  $\sim 6 \times 10^{16}$

# Assignment

Find out the size of protein database, UniProt, and nucleotide database, GenBank.

Compute No. of matrix cells to be computed using DP for:

- (1) searching protein database, UniProt, and nucleotide database, GenBank and the time required assuming query sequence of length 1000 bases.
- (2) Comparing Human Chr 1 ~249Mbp with a query sequence of 1000 bases using DP, and comparing it with Chr 1 of Mouse (~195Mbp)? What is the space requirement in the two cases?

Consider computation time as 10M matrix cells per second (or operation time of your machine)

# Linear Space Alignments

Fortunately, situation is better with **memory** than **speed**:

- there are techniques that give the **optimal alignment** in limited memory, of order  $n + m$  rather than  $nm$ , but comes with a cost of **doubling** of time.

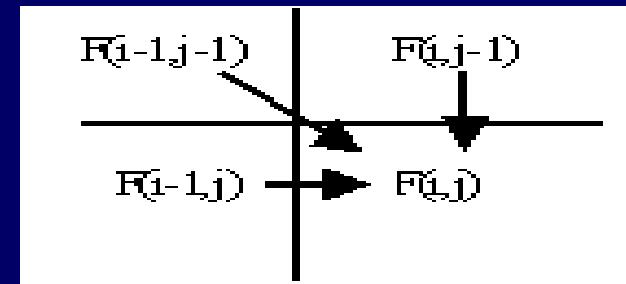
These methods are commonly referred to as **linear space alignment methods**.

# Linear Space Alignments

If only score of the alignment is needed, the problem is very simple:

Recurrence relation for  $F(i, j)$  depends only on entries **one row back** - one can throw away rows that are further than one back from the current point.

For **local alignment**, the maximum score in the whole matrix is required - easy to **keep track of the maximum value** as the matrix is being built.



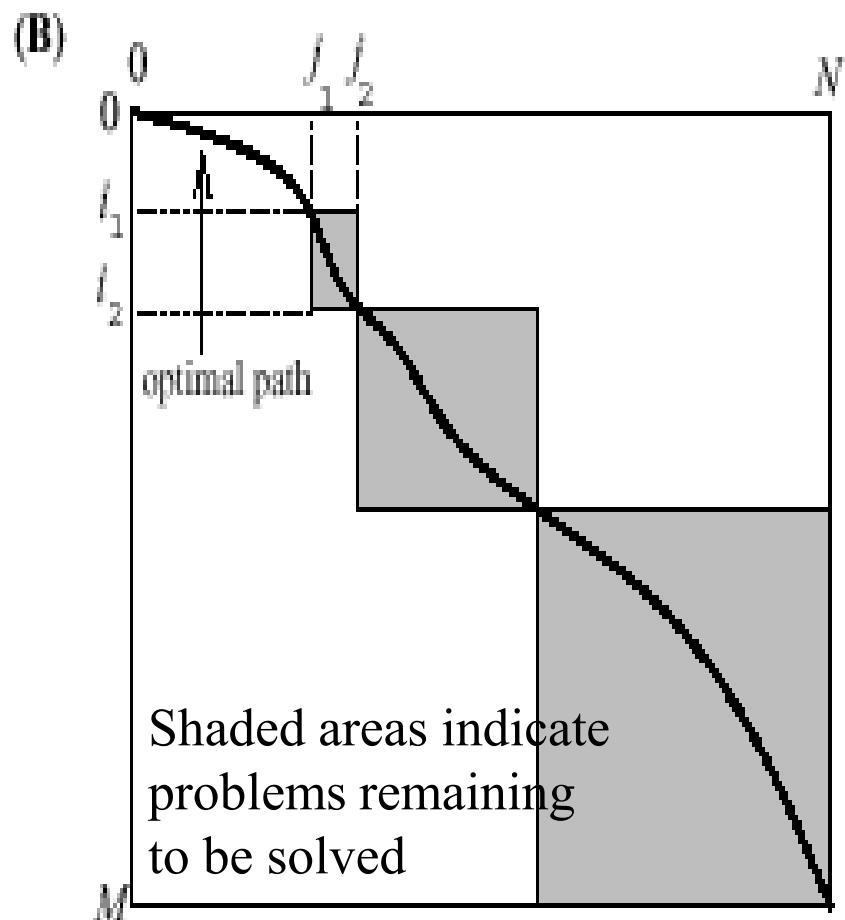
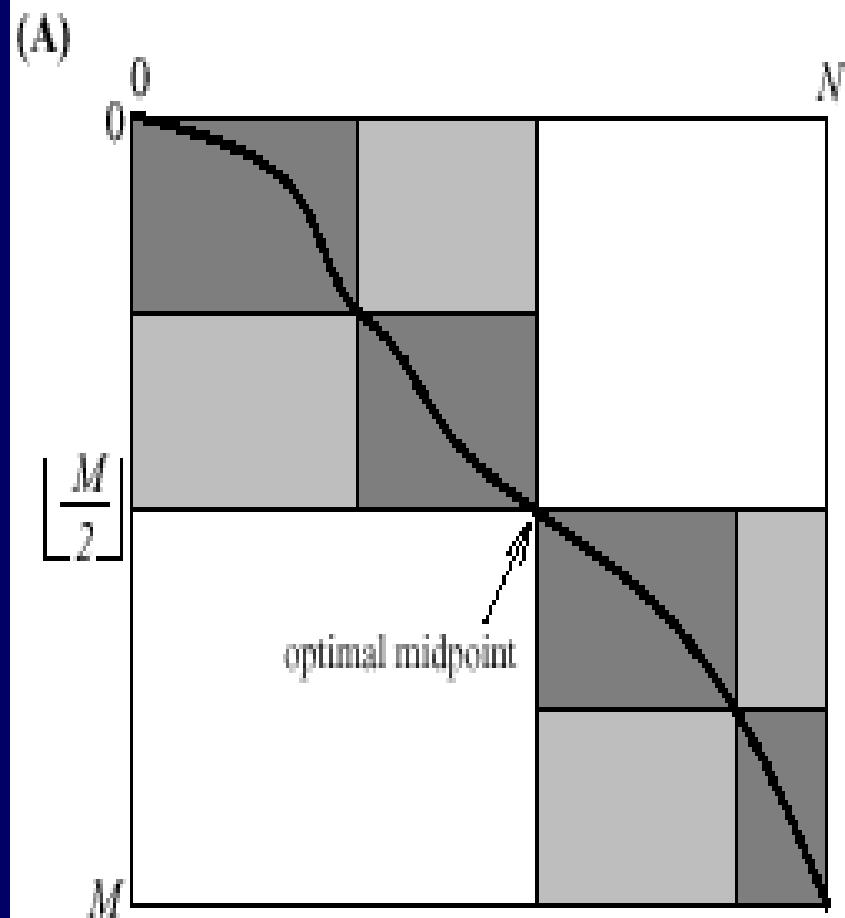
# Linear Space Alignments

While this gives us the score, it will not find the Alignment!

By throwing away rows to avoid  $O(nm)$  storage, we loose the traceback pointers.

An approach used to obtain the alignment uses the principle of **divide and conquer**.

# Snapshot of Execution of Hirschberg's algorithm



# Linear Space Alignments

## Steps:

- Let  $u = m/2$  (integer part)
- Identify a  $v$  such that the cell  $(u, v)$  is on the optimal alignment, i.e.,  $v$  is the column where the alignment crosses the  $u = m/2$  row of the matrix
  - this splits the DP problem into two subproblems: from  $(0, 0)$  to  $(u, v)$ , & from  $(u, v)$  to  $(m, n)$
  - full alignment will be concatenation of the optimal alignments for these two separate submatrices

# Linear Space Alignments

But how do we find  $v$ ?

- By combining the results of “forward” and “backward” DP passes at row  $u$ , for each point along the middle row, i.e.,
  - optimal score from  $(0, 0)$  to each point in row  $u$  and optimal score from that point to  $(m, n)$ .
- A sweep along the middle row, checking these sums, determines a point  $(m/2, v)$  where an optimal path crosses the middle row.
- This is then done **recursively**, by successively halving each region.

# Linear Space Alignments

**EMBOSS Programs:**

**matcher:** linear-space version of local alignment  
algorithm by M. S. Waterman & M. Eggert

**stretcher:** global alignment algorithm using linear space

# Database Search - Need for Heuristics

- Dynamic prog'g algorithms are computer intensive & time consuming  $\sim O(nm)$
- Searching large databases using these algorithms is not feasible

e.g., if the time to compare two 1kb sequences is  $\sim$  milliseconds, comparison of 1kb sequence with the human genome will take  $\sim$ 1hr.

# Database Search - Need for Heuristics

- **Solution 1:** implement the algorithm in hardware
  - **expensive**

SW algorithm has been implemented on Hybrid Core machines

- **Solution 2:** distribute the job to several processors to do in parallel mode
  - **gets expensive as no. of processors  $\sim 1000$**
- **Solution 3:** use heuristics - **gives approx. solutions**

# Heuristic Algorithms

Basic idea: first locate high-scoring short stretches and then extend them

Two well known programs:

~ 50 times  
faster than DP

- FASTA: Fast Alignment Tool
- BLAST: Basic Local Alignment Search Tool

Heuristic algorithms aim at speeding up the database search at the price of possibly **missing** the best scoring alignment

# FASTA

- Compares sequences pairwise
- Heuristics: A good alignment will have exact matching subsequences
- Gain on speed but loss of sensitivity
- Best suited for global alignment
- Works better with DNA sequences

<http://www.ebi.ac.uk/Tools/sss/fasta>

Pearson & Lipman, 1988

# FASTA: The algorithm

- Based on the logic of the dot matrix method
- View sequences as sequences of short words ( $k$ -tuple)
  - DNA:  $k = 2-6$ , proteins:  $k = 1, 2$

## Motivation:

- Good alignments should contain many exact matches
- Hashing can find exact matches in  $O(n)$  time
- Diagonals can be formed from exact matches quickly and sorted
- Apply more precise alignment to small search space at the end

# FASTA: The algorithm

- Look for all  $k$ -tuple matches between query & database sequences
  - For DNA  $k = 2-6$ , for proteins,  $k = 1, 2$
- This is done by a lookup table, or hash
- For each  $k$ -tuple the database is pre-processed to identify its positions
- Query is scanned - for each  $k$ -tuple in the query, lookup the table/hash to identify matches in the database

# Lookup method for finding an alignment

Position 1 2 3 4 5 6 7 8 9 10 11

Seq A n c s p t a . . . . .

Seq B . . . . . a c s p r k

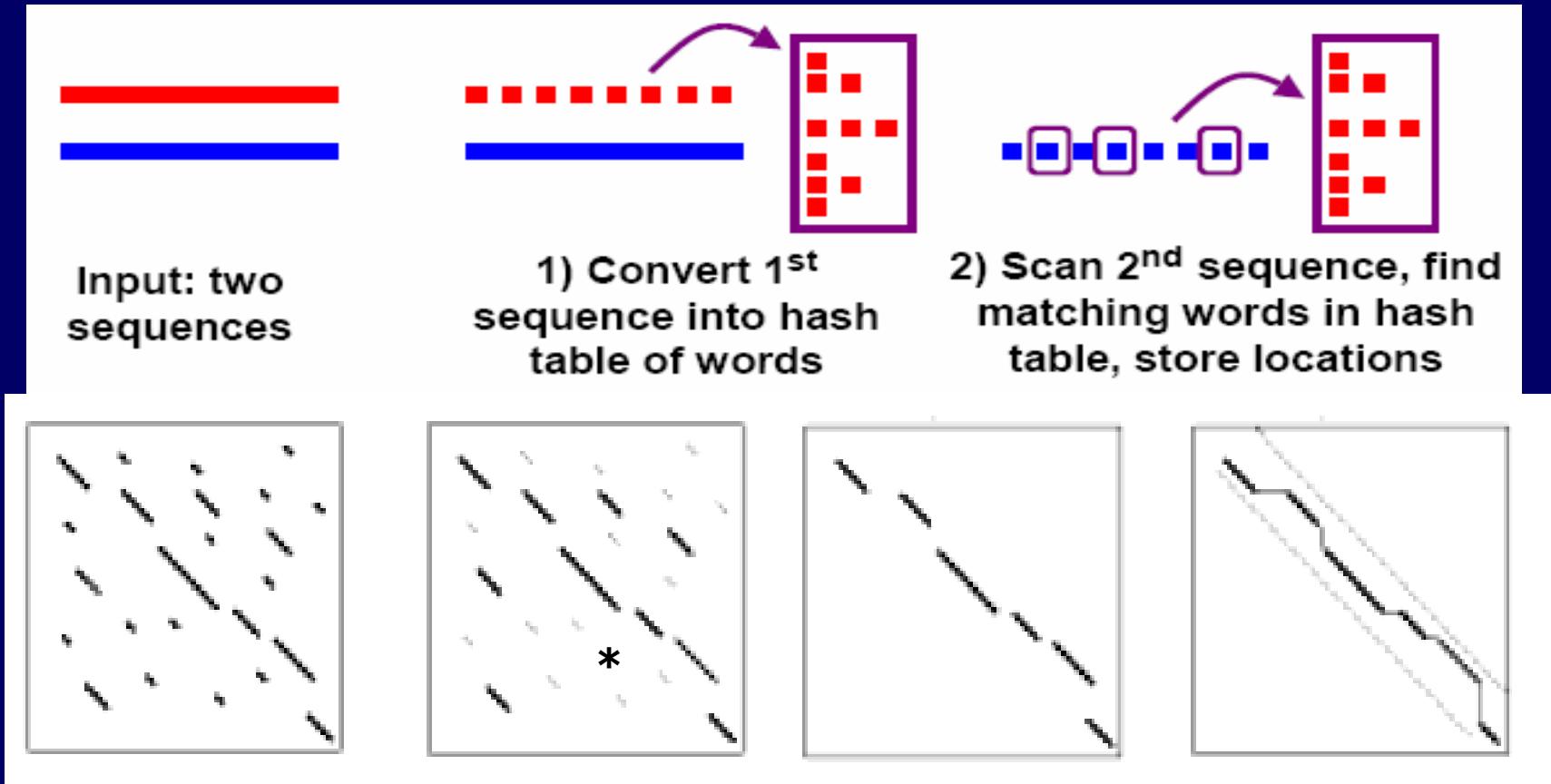
Dot matrix method

	n	c	s	p	t	a					
a										•	
c		•									
s			•								
p				•							
r											
k											

$\sim O(n)$

Possible alignment:

n c s p t a  
| | | | |  
a c s p r k



Identify regions of identity (hot spots).

Scan regions using a scoring matrix & save the best initial regions, regions with score < threshold are discarded, \* marks best region

Optimally join initial regions with score > threshold (INITN),

Recalculate to obtain an OPT score by performing DP around the highest scoring region

# FASTA: The algorithm

- Look for adjacent hot spots (substrings of exact matches) on the same diagonal
  - join them to form larger segments
  - space between hot spots gets -ve score
- Find 10 highest scoring diagonals
- Find the best diagonal run and filter out low scoring runs - by recomputing the score for each diagonal using scoring matrix and threshold score
- Combine close diagonal runs, including indels
- Compute alternative local alignments, using a band around the best diagonal run
- Finally use DP to align the query against best ranking resulting sequences

# FASTA: Programs

- **FASTA**: Search DNA/protein database with a DNA/protein query
- **FASTX/FASTY**: compare a DNA sequence to a protein database (fasty in both forward & reverse frames)
  - useful for gene finding by homology
- **TFASTX/TFASTY**: compare a protein sequence to a DNA database
  - to find all genes coding for a protein

[http://fasta.bioch.virginia.edu/fasta\\_www2/](http://fasta.bioch.virginia.edu/fasta_www2/)

# FASTA: Programs

- If a DNA sequence has a high possibility of errors (e.g., ESTs), the translated sequence may be inaccurate due to a.a. changes or frameshifts.
- FASTA programs are designed to go around such errors by allowing gaps and frameshifts in the alignments.
- FASTX & TFASTX allow frameshifts **between** codons, whereas FASTY and TFASTY allow substitutions or frameshifts **within** a codon.
  - can handle errors up to 10% in query sequences.

# FASTA: Programs

## Search Databases with FASTA

[Search Proteomes/Genomes](#)

[Statistical Significance from Shuffles](#)

[Find Internal Duplications \(lalign/plalign\)](#)

[Hydropathy/Secondary-Structure/seg](#)

Retrieve result **RID:**

This page provides searches against comprehensive databases, like **SwissProt** and **NCBI RefSeq**. The **PIR1 Annotated** database can be used for small, demonstration searches. The **NCBI nr** database is also provided, but should be your last choice for searching, because its size greatly reduces sensitivity. The best first choice for searching is a genome database from a closely related organism (e.g. **RefSeq Human** for vertebrates).

The [Individual Proteomes/Genomes](#) page provides searches against selected prokaryotes.

**New:** Annotation features available for Uniprot/SwissProt/PIR1 library searches.

[Show recent searches](#)

**Choose: (A) Program, (B) Query (sequence/accession), (C) Database and (D) Start Search:**

**(A) Program:**

FASTA: protein:protein

SSEARCH: local protein:protein

GGSEARCH: global protein:protein

GLSEARCH: global/local protein:protein

FASTX: DNA vs protein

FASTY: DNA vs protein

FASTA: DNA:DNA

TFASTX: protein vs DNA

TFASTY: protein vs DNA

FASTS: unordered peptides vs protein

TFASTS: unordered peptides vs DNA

FASTF: mixed peptides vs protein

TFASTF: mixed peptides vs DNA

FASTM: ordered oligonucleotides vs DNA

FASTS: unordered oligonucleotides vs :DNA

Or upload query:

Protein  DNA

Compare your own sequences:

Range:   Use Subset range

Annotate Query Sequence (SwissProt accessions)

No file chosen

[Entrez protein](#) / [Entrez DNA sequence browser](#)

[Uniprot sequence browser](#)

DNA (rev-comp only)

**(C) Database:**

Protein

DNA

**(D) Start Search**

Upload annotation file:  No file chosen

[Entrez protein](#) / [Entrez DNA](#) sequence browser

[Uniprot sequence browser](#)

Or upload query from file:  No file chosen

Protein  DNA (both-strands)  DNA (forward only)  DNA (rev-comp only)

**(C) Database:**

Protein

DNA

**(D) Start Search**

Annotations:

Exclude low complexity (seg)

Comments (optional):

**Other search options:**

Output limits:   Show Histogram

Scoring matrix:

Hide Alignments

Alignment Options: Highlight  similarities  differences  compact differences. Output format:

[FASTA program information](#) | [Download FASTA](#) | [About the Author](#)

Copyright © 1996, 1997, 1998, 1999, 2002, 2014, 2015 by William R. Pearson and The Rector & Visitors of the University of Virginia

The FASTA package is open source software, licensed under the Apache License, Version 2.0 (the "License"); you may not copy this software except in compliance with the License. You may obtain a copy of the License at: <http://www.apache.org/licenses/LICENSE-2.0>

# FASTA: Programs

## UVa FASTA Server

New: Annotation features available for Uniprot/SwissProt/PIR1 library searches.

### About

- Getting started
- [fasta\\_guide.pdf](#)

### Other FASTA Servers

- EMBL-EBI
- KEGG (Japan)

### References

- FASTA
- FASTX/FASTY
- Statistics
- FASTS/FASTF

### Software

- FASTA v36 ChangeLog
- Downloads
- Sequence Libraries

### Other resources

- PSI-Search2
- BLASTP with annotations
- CHAPS - Convert HMMs and Profiles
- FASTA Exercises
- NCBI BLAST server
- EMBL-EBI Server

The **FASTA** programs find regions of local or global similarity between Protein or DNA sequences, either by searching Protein or DNA databases, or by identifying local duplications within a sequence. Other programs provide information on the statistical significance of an alignment. Like **BLAST**, **FASTA** can be used to infer functional and evolutionary relationships between sequences as well as help identify members of gene families.

### Protein

- Protein-protein **FASTA**
- Protein-protein Smith-Waterman ([sssearch](#))
- Global Protein-protein (Needleman-Wunsch) ([ggsearch](#))
- Global/Local protein-protein ([glsearch](#))
- Protein-protein with unordered peptides ([fasts](#))
- Protein-protein with mixed peptide sequences ([fastf](#))

### Nucleotide

- Nucleotide-Nucleotide (DNA/RNA [fasta](#))
- Ordered Nucleotides vs Nucleotide ([fastm](#))
- Un-ordered Nucleotides vs Nucleotide ([fasts](#))

### Translated

- Translated DNA (with frameshifts, e.g. ESTs) vs Proteins ([fastx/fasty](#))
- Protein vs Translated DNA (with frameshifts) ([tfastx/tfasty](#))
- Peptides vs Translated DNA ([tfasts](#))

### Statistical Significance

- Protein vs Protein shuffle ([prss](#))
- DNA vs DNA shuffle ([prss](#))
- Translated DNA vs Protein shuffle ([prfx](#))

### Local Duplications

- Local Protein alignments ([lalign](#))
- Plot Protein alignment "dot-plot" ([plalign](#))
- Local DNA alignments ([lalign](#))
- Plot DNA alignment "dot-plot" ([plalign](#))

# Basic Local Alignment Search Tool (BLAST)

## Motivation:

- Good alignments should contain many close matches
- Statistics can determine which matches are significant - more sensitive than % identity
- Designed to work best for local ungapped alignments - now incorporates gaps
- Extending matches in both directions finds alignment - yields high-scoring/maximum segment pairs (HSPs/MSPs)

Altshul *et al.*, 1990

# BLAST: Fastest alignment tool

- View sequences as sequence of short words (*k*-tuples)
  - DNA: 7, 11, 15 (default 11),
  - protein: 2, 3, 6 (default 6)
- Create hash table of neighborhood (closely-matching) words
- Use statistics to set threshold for “closeness”

# BLAST: Fastest alignment tool

- First look for short matching segments
- Choose matching segments above a threshold score, hits
- Overlapping hits form a larger segment
- Large segment extended in either direction if its score is above a threshold
- Extension of alignment continues until the score falls below the drop-off threshold from maximum value

PQG - 18

PEG - 15

PRG - 14

PSG - 13

PQA - 12

# BLAST: Example

- Sequence: ASTNC
- Word ASTN has score of 9 for exact match using PAM250 scoring matrix
- If threshold score is 17, ASTN will not be used for querying the database
- STNC has score 19 for exact match using PAM250, and will be used
- Note: not all exact matches are used if the score is below the threshold.

# BLAST: The Algorithm

Look for high scoring segments pairs (HSPs)

- looks for similar instead of identical pairs
- uses scoring matrix to score aligned pairs
- only those pairs which score above a threshold are considered for extension
- the extension is without gaps

L P	P Q G	L L	query	
M P	P E G	L L	Db seq	HSP score
<word>				= 9 + 15 + 8
←		→		= 32
Ext. to Left		Ext. to Rt		

L-L: 4, P-P: 7, L-M: 2

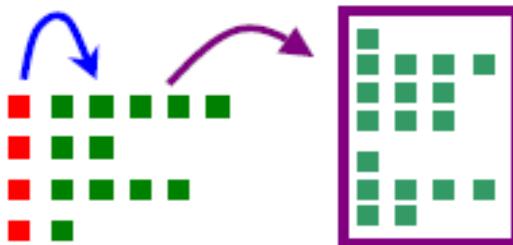
# BLAST: The Algorithm

- Ungapped extension of HSPs with scores  $> T$  (threshold) - identifies maximal segment pairs.
- Extension continues until the score drops below a threshold drop-off from the maximum score encountered
- Highest scoring segment pair, MSP (Maximal scoring Segment Pair) identified
- For gapped alignment, BLAST uses the same strategy as FASTA of joining segments on different diagonals.

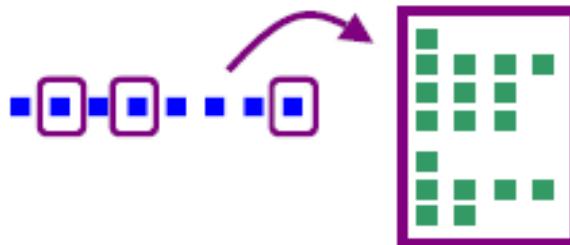
# BLAST: Algorithm



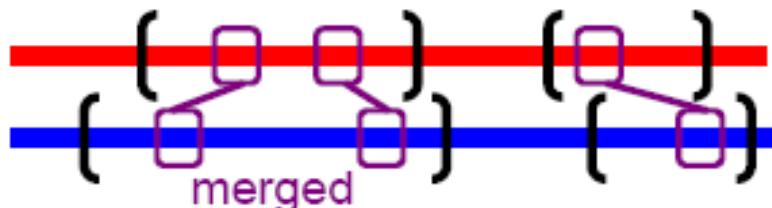
1) Convert 1<sup>st</sup> sequence into words (using all frames for given word size)



2) Calculate for each word list of “neighborhood” words (scoring threshold  $T$ ) and enter in dictionary



3) Scan 2<sup>nd</sup> sequence, find matching words in dictionary, store locations



4) For each match, extend alignment in both directions while score above threshold  $S$ , merge segments



5) Align best segments using dynamic programming, report statistically significant matches

# Selecting a BLAST program

- BLAST is suite of programs, the most popular being
- **blastn**: compares nucleotide sequence with nucleotide database
- **blastp**: compares protein sequence with a protein sequence database

## Basic Local Alignment Search Tool

**BLAST** finds regions of similarity between biological sequences. The program compares nucleotide or protein sequences to sequence databases and calculates the statistical significance.

[Learn more](#)

NEWS

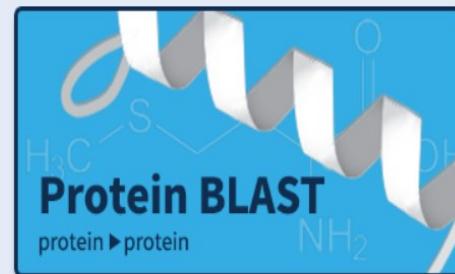
End of updates for BLAST+ version 4 databases (dbV4)

Start moving to the new version 5 databases!

Fri, 27 Sep 2019 16:00:00 EST

[More BLAST news...](#)

## Web BLAST



## BLAST Genomes

[Search](#)[Human](#)[Mouse](#)[Rat](#)[Microbes](#)

blastn  blastp  blastx  tblastn  tblastxBLASTN programs search nucleotide databases using a nucleotide query. [more...](#)

Reset page Bookmark

## Enter Query Sequence

Enter accession number(s), gi(s), or FASTA sequence(s) [?](#)

Or, upload file

 [Browse...](#) [?](#)

Job Title

 Align two or more sequences [?](#)

## Choose Search Set

Database

 Human genomic + transcript  Mouse genomic + transcript  Others (nr etc.):Nucleotide collection (nr/nt) [?](#)Organism  
Optional

Enter organism name or id-completions will be suggested

 Exclude [+](#)Enter organism common name, binomial, or tax id. Only 20 top taxa will be shown. [?](#)Exclude  
Optional Models (XM/XP)  Uncultured/environmental sample sequencesEntrez Query  
OptionalEnter an Entrez query to limit search [?](#)

## Program Selection

Optimize for

- Highly similar sequences (megablast)
- More dissimilar sequences (discontiguous megablast)
- Somewhat similar sequences (blastn)

Choose a BLAST algorithm [?](#)

## BLAST

Search database Nucleotide collection (nr/nt) using Megablast (Optimize for highly similar sequences)

 Show results in a new window

- Megablast is intended for comparing a query to closely related sequences and works best if the target percent identity is 95% or more but is very fast.
- Discontiguous megablast uses an initial seed that ignores some bases (allowing mismatches) and is intended for cross-species comparisons.
- BlastN is slow, but allows a word-size down to seven bases.

## Algorithm parameters

Enter coordinates for a **subrange** of the query sequence. The BLAST search will apply only to the residues in the range. Sequence coordinates are from 1 to the sequence length. The range includes the residue at the **To** coordinate. [more...](#)

This title appears on all BLAST results and saved searches.

## Genomic plus Transcript

Human genomic plus transcript (Human G+T)  
Mouse genomic plus transcript (Mouse G+T)

## Other Databases

- Nucleotide collection (nr/nt)
- Reference mRNA sequences (refseq\_rna)
- Reference genomic sequences (refseq\_genomic)
- NCBI Genomes (chromosome)
- Expressed sequence tags (est)
- Non-human, non-mouse ESTs (est\_others)
- Genomic survey sequences (gss)
- High throughput genomic sequences (HTGS)
- Patent sequences (pat)
- Protein Data Bank (pdb)
- Human ALU repeat elements (alu\_repeats)
- Sequence tagged sites (dbsts)
- Whole-genome shotgun reads (wgs)
- Environmental samples (env\_nt)

## Algorithm parameters

### General Parameters

#### Max target sequences

100

Select the maximum number of aligned sequences to display 

Maximum number of aligned sequences to display (the actual number of alignments may be greater than this).

#### Short queries

Automatically adjust parameters for short input sequences 

Automatically adjust word size and other parameters to improve results for short queries.

#### Expect threshold

10

 **statistical significance threshold**

Expected number of chance matches in a random model. [more...](#)  [Expect value tutorial](#)

#### Word size

28

The length of the seed that initiates an alignment. [more...](#)

#### Max matches in a query range

0

 Limit the number of matches to a query range. This option is useful if many strong matches to one part of a query may prevent BLAST from presenting weaker matches to another part of the query. The algorithm is based upon <http://www.ncbi.nlm.nih.gov/pubmed/10890403>

### Scoring Parameters

#### Match/Mismatch Scores

1,-2

Gap Costs [more...](#) 

Linear

1,-2  
1,-3  
1,-4  
2,-3  
4,-5  
1,-1

Existence: 5 Extension: 2  
Existence: 2 Extension: 2  
Existence: 1 Extension: 2  
Existence: 0 Extension: 2  
Existence: 3 Extension: 1  
Existence: 2 Extension: 1  
Existence: 1 Extension: 1

### Filters and Masking

#### Applied only to query seq

#### Filter

Low complexity regions 

Mask regions of low compositional complexity that may cause spurious or misleading results. [more...](#)

Species-specific repeats for: [Homo sapiens \(Human\)](#)  

#### Mask

Mask for lookup table only 

Mask query while producing seeds used to scan database, but not for extensions. [more...](#)

Mask lower case letters 

[Home](#)[Recent Results](#)[Saved Strategies](#)[Help](#)

► NCBI/BLAST/blastp suite

[blastn](#)[blastp](#)[blastx](#)[tblastn](#)[tblastx](#)

Enter Query Sequence

Enter accession number, gi, or FASTA sequence [?](#)[Clear](#)Query subrange [?](#)From To Or, upload file [Browse...](#) [?](#)

Job Title

Enter a descriptive title for your BLAST search [?](#) Align two or more sequences [?](#)

## Choose Search Set

Database

Non-redundant protein sequences (nr) [?](#)

Organism

Optional

Enter organism name or id--completions will be suggested

Enter organism common name, binomial, or tax id. Only 20 top taxa will be shown. [?](#)Entrez Query  
OptionalEnter an Entrez query to limit search [?](#)

Non-redundant protein sequences (nr)

Reference proteins (refseq\_protein)  
Swissprot protein sequences(swissprot)  
Patented protein sequences(pat)  
Protein Data Bank proteins(pdb)  
Environmental samples(env\_nr)

## Program Selection

Algorithm

 blastp (protein-protein BLAST) PSI-BLAST (Position-Specific Iterated BLAST) PHI-BLAST (Pattern Hit Initiated BLAST)Choose a BLAST algorithm [?](#)**blastp - compares a protein query to a protein database****PSI-BLAST - allows the user to build a PSSM using the results of the first BlastP run****PHI-BLAST - performs the search but limits alignments to those that match a pattern in the query****BLAST**Search database nr using **Blastp (protein-protein BLAST)** Show results in a new window► [Algorithm parameters](#)

## ▼ Algorithm parameters

### General Parameters

Max target sequences

100

Select the maximum number of aligned sequences to display 

Short queries

Automatically adjust parameters for short input sequences 

Expect threshold

10



Word size

3



- PAM30
- PAM70
- BLOSUM80
- BLOSUM62**
- BLOSUM45

### Scoring Parameters

Matrix

BLOSUM62



- Existence: 9 Extension: 2
- Existence: 8 Extension: 2
- Existence: 7 Extension: 2
- Existence: 12 Extension: 1
- Existence: 11 Extension: 1**
- Existence: 10 Extension: 1

Gap Costs

Existence: 11 Extension: 1



Compositional  
adjustments

Conditional compositional score matrix adjustment



to compensate for the compositions of  
the two sequences being compared

### Filters and Masking

Filter

Low complexity regions 

Mask

Mask for lookup table only 

Mask lower case letters 



Difference between the two options?

**BLAST**

Search **database nr** using **Blastp (protein-protein BLAST)**

# Selecting a BLAST program

- **blastn**: compares nucleotide sequence with nucleotide database
- **blastp**: compares protein sequence with a protein sequence database
- **blastx**: compares a nucleotide sequence translated in all 6 frames with protein sequence database
- **tblastn**: compares a protein sequence with a nucleotide database dynamically translated in all 6 frames

# Selecting a BLAST program

- **tblastx**: compares the 6-frame translations of a nucleotide sequence with a 6-frame translation of a nucleotide database. Can no longer be used with nr database on BLAST web pages as it is computationally very intensive
- **Megablast**: fast comparison of large genomic sequences
- **bl2seq**: compare two sequences

## Specialized searches

### SmartBLAST



Find proteins highly similar to your query

### Primer-BLAST



Design primers specific to your PCR template

### Global Align



Compare two sequences across their entire span (Needleman-Wunsch)

### CD-search



Find conserved domains in your sequence

### IgBLAST



Search immunoglobulins and T cell receptor sequences

### VecScreen



Search sequences for vector contamination

### CDART



Find sequences with similar conserved domain architecture

### Multiple Alignment



Align sequences using domain and protein constraints

### MOLE-BLAST



# Specialized BLAST: PSI-BLAST

- Position Specific Iterated (PSI) BLAST
- Designed to find remote homologues (15 - 25% identity levels)
- Construct scoring matrices by multiple alignment of hits obtained
- Search the database with the new scoring matrix for every iteration
- Iterate until convergence is reached

# Specialized BLAST: PSI-BLAST

The idea of constructing a scoring matrix from the hits is that the new scoring matrix is tailor-made to find sequences similar to the query.

- allowing detection of homologues in the range of 15%-25% sequence identity levels.

# Specialized BLAST: PHI-BLAST

- Pattern Hit Iterated (PHI) BLAST
- Given a protein sequence and a motif/pattern within it, the program finds all proteins which carry that pattern and have similar surrounding residues
- Combines regular expression with local alignment around the matching region
- Designed to find protein motifs

# BLAST: Statistics

BLAST uses a heuristic approach to accelerate aligning two sequences - it becomes essential to compute statistical significance of the alignment.

- Compute the probability that the alignment is obtained by "chance".

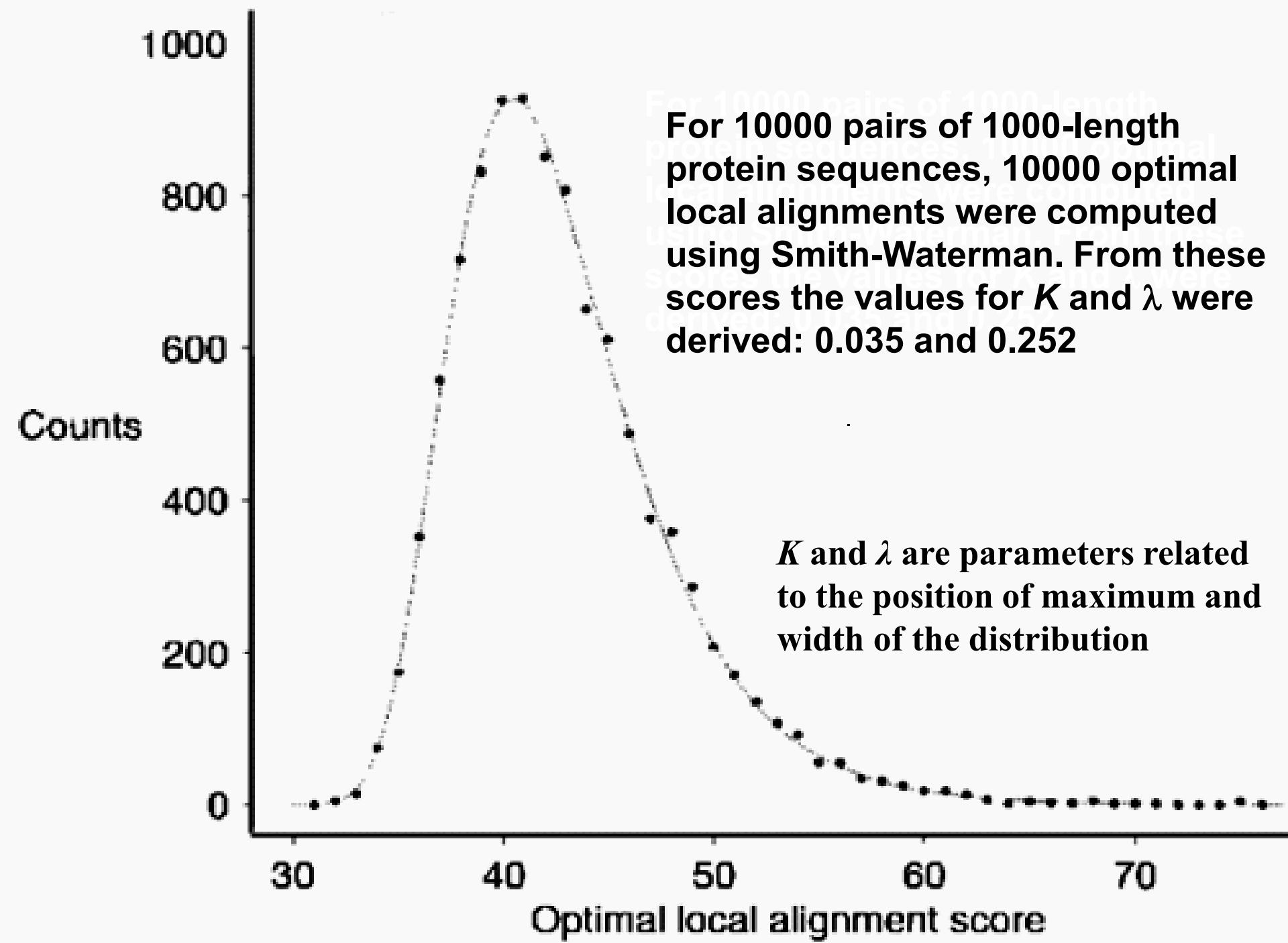
Strategy adopted by BLAST - compute distribution of alignment scores for random sequences.

- very little known about the random distribution of optimal global alignment scores

Statistics for ungapped local alignment score is quite well understood

# BLAST: Statistics

- Model random sequences generated: for proteins choose amino acids randomly with specific probabilities
- Align the random sequences using Smith-Waterman algorithm and compute optimal alignment scores
- It is known that the distribution of the maximum of independent identically distributed (i.i.d.) random variables is an extreme value distribution
  - scores of optimal alignment of random sequences falls in this category



For 10000 pairs of 1000-length protein sequences, 10000 optimal local alignments were computed using Smith-Waterman. From these scores the values for  $K$  and  $\lambda$  were derived: 0.035 and 0.252

$K$  and  $\lambda$  are parameters related to the position of maximum and width of the distribution

# BLAST: Statistics

In the limit of large sequence lengths  $m$  and  $n$ ,  
expected number of HSPs with score at least  $S$ :

$$E = K(mn)e^{-\lambda S}$$

Dependent on  
database size

$K$  and  $\lambda$  are empirical parameters derived from the distribution

E-value depends on the size of query sequence as well as that of database.

⇒ for same aligned pair of sequences, E-value may be different depending on the database searched

Rule of thumb - search a larger database whenever possible

# Significance of a hit: example

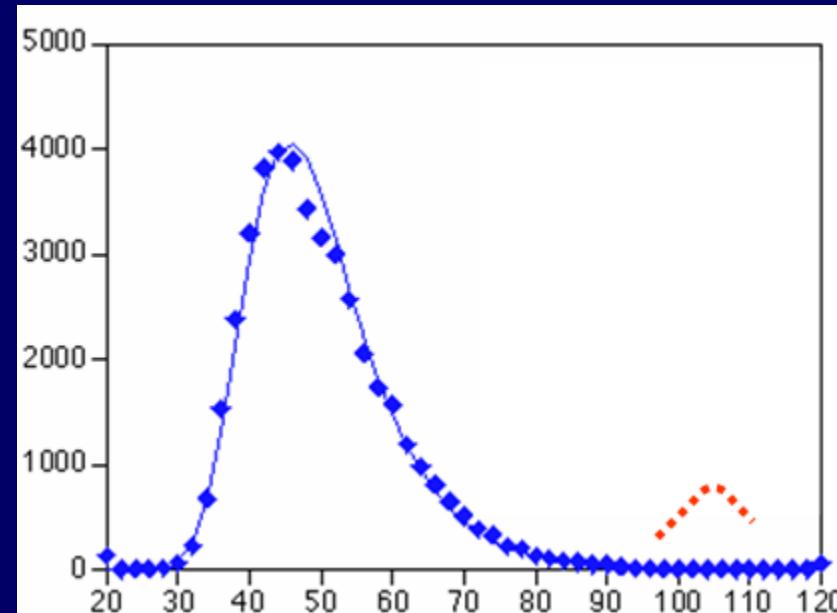
Search against a database of 10,000 sequences.

An extreme-value distribution (blue) is fitted to the distribution of all scores.

It is found that 99.9% of the blue distribution has a score below 112.

This means that when searching a database of 10,000 sequences you'd expect to get  $0.1\% * 10,000 = 10$  hits with a score of 112 or better for random reasons

10 is the E-value of a hit with score 112. You want E-values well below 1!



# BLAST: Bit Scores

Raw scores have little meaning without detailed knowledge of the scoring system used, or more simply its statistical parameters  $K$  and  $\lambda$

Query: 1 SGLKSLVGKTALLSGTSSKL 20  
Sbjct: 1 SGLKSLVGKTALLSGTSSKL 20

Score = 91

Query: 1 CQHMWYQWMIQCIWMYHCMQ 20  
Sbjct: 1 CQHMWYQWMIQCIWMYHCMQ 20

Score = 138

Based on scores  $\Rightarrow$  alignment 2 is better than alignment 1, but both the alignments are of the same length and have 100% identity

# BLAST: Bit Scores

Bit score,  $S'$  is calculated from the raw score by normalizing with the statistical variables that define a given scoring system:

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

- effect of normalization is to change the score distribution into standard normal distribution
- ⇒ bit scores from different alignments, even those employing different scoring matrices can be compared.

$E$ -value corresponding to a given bit score is

$$E = m n 2^{-S'}$$

# Using BLAST: Inferring Homology

How do you know when a certain level of similarity implies homology?

Does doing multiple searches help?

# Using BLAST: Inferring Homology

Rules of thumb expressed in terms of percent identity in the optimal alignment:

- ~ 45% identity - the proteins will have very similar structures and are very likely to have a common or at least a similar function.
- ~ 25% identity - they are likely to have a similar general folding pattern.
- 18-25% identity - defined as 'twilight zone' - lower degree of sequence similarity cannot rule out homology.

# Using BLAST: Inferring Homology

- In general, two sequences significantly similar over the entire length are likely to be homologous
- 50% similarity over a short sequence often occurs by chance
- Low complexity regions can be highly similar without being homologous
- Homologous sequences are not always highly similar
- Suggested BLAST cutoffs
  - For nucleotide-based searches, look for hits with  $e\text{-value} \leq 10^{-6}$  and sequence identity  $\geq 70\%$
  - For protein-based searches, look for hits with  $e\text{-value} \leq 10^{-3}$  and sequence identity  $\geq 25\%$

# Substitution Matrices

## Importance of scoring matrices

- Scoring matrices appear in all analysis involving sequence comparison.
- Choice of matrix can strongly influence the outcome of the analysis.
- Scoring matrices implicitly represent a particular theory of evolution.
- Understanding theories underlying a given scoring matrix can aid in making a proper choice

# Substitution Matrices for Nucleotides

**BLAST Matrix:** In general, different substitution matrices are tailored to detect similarities among sequences that are diverged by differing degrees, e.g.

	A	T	C	G
A	2	-3	-3	-3
T	-3	2	-3	-3
C	-3	-3	2	-3
G	-3	-3	-3	2

default – blastn

	A	T	C	G
A	1	-3	-3	-3
T	-3	1	-3	-3
C	-3	-3	1	-3
G	-3	-3	-3	1

default - megablast

- Default scoring scheme for blastn target sequences that are 90% identical, while the default scoring matrix for megablast is appropriate for sequences that are 99% identical.

# Substitution Matrices for Nucleotides

Distance matrix:

Another measure, called **edit distance, or cost** is also used, e.g.,

**Transition/Transversion Matrix:**

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

Using a Transition/Transversion matrix reduces noise in comparisons of distantly related sequences

# Substitution Matrices for Nucleotides

ATGCCGTGATAGTCGAT  
ACGGCTCGATCTACTAC

**Identity Matrix:** Score: 8 1's = 8

**Blast Matrix:** Score:  $8 \times 1 - 3 \times 9 = 8 - 27 = -19$

**Transition/Transversion Matrix:** (distance/cost)

Score:  $8 \times 0 + 3 \times 1 + 6 \times 5 = 33$

Which is better?

It is clear that raw alignment scores are meaningless without specific knowledge of the scoring matrix used.

# PAM Matrices

- developed by Margaret Dayhoff and coworkers.

They examined **1572** accepted mutations between **71** families of closely related sequences of proteins and noticed that the **substitutions were not random**.

⇒ evolutionarily related proteins need not have same AA at every position: can have a **comparable** one.

PAM stands for “Point Accepted Mutations” or “Percent of Accepted Mutations” (“accepted” refers to mutations that have become fixed in the population)

PAM matrices refer to various degrees of sensitivity depending on the evolutionary distance between sequence pairs

# PAM Matrices

**PAM units - measure the amount of evolutionary distance between two protein sequences.**

**One PAM of evolution means that the total number of substitutions is 1% of the sequence length**

After 100 PAMs of evolution, not every position would have changed, because some positions will have mutated several times, perhaps returning to their original state

**- even after 250 PAMs, proteins are sufficiently similar that sequence homology can frequently be detected.**

# PAM Matrices

- If changes were purely random
  - Frequency of each possible substitution would be proportional to background frequencies
- In related proteins:
  - Observed substitution frequencies called the target (replacement) frequencies are biased toward those that do not disrupt the protein's function
  - These point mutations are “accepted” during evolution
- Log-odds approach:
  - Scores proportional to the natural log of the ratio of target frequencies to background frequencies

# PAM Matrices

Score matrix entry for time  $t$  given by:

$$s(a, b|t) = \log \frac{P(a|b, t)}{q_a q_b}$$

Conditional probability that  $b$  is substituted by  $a$  in  $t$

Frequency of AAs  $a$  &  $b$

# PAM Matrices Construction

- Based on the hypothesis that proteins diverge by accumulating uncorrelated mutations
- Align closely related sequences ( $> 85\%$  identity)
  - considering very similar sequences allow the correct alignments to be determined with high certainty
- Observe the probability of AA changes & compute the log-odds ratio
- Normalize the matrix (relative frequencies of various mutations multiplied by a carefully chosen constant) to give an average change of 1% of all positions to obtain PAM-1 matrix

# PAM Matrices Construction

How to derive scoring matrices for distantly related sequences from data about closely related sequences?

- Scoring matrices to any PAM distance can be determined by **extrapolating** from PAM 1 – by successive iteration of the reference mutation matrix:

$$M_n = (M_1)^n$$

e.g., for PAM 250, multiply PAM-1 250 times with itself.

Assumption in this evolutionary model is that AA substitutions observed over short periods of evolutionary history can be extrapolated to longer distances.

# PAM Matrices Construction

$$M_n = (M_1)^n$$

$M_1$  - matrix reflecting 99% sequence conservation and one accepted point mutation (PAM 1) per 100 residues

$M_n$  - substitution probabilities after  $n$  PAMs

**PAM250** – most frequently used matrix that is geared to very distant, but still detectable homologies

**PAM matrices are derived from global alignments of closely related sequences**

Table 1 - The log odds matrix for 250 PAMs (multiplied by 10)

	A	C	D	E	F	G	H	I	K	L	M	N	P	Q	R	S	T	V	W	Y
A $\rightarrow$	2	-2	0	0	-4	1	-1	-1	-1	-2	-1	0	1	0	-2	1	1	0	-6	-3
C $\rightarrow$	12	-5	-5	-4	-3	-3	-2	-5	-6	-5	-5	-4	-3	-5	-4	0	-2	-2	-8	0
D	4	3	-6	1	1	-2	0	-4	-3	2	-1	2	-1	0	0	-2	-7	-4		
E	4	-5	0	1	-2	0	-3	-2	1	-1	2	-1	0	0	-2	-7	-4			
F	9	-5	-2	1	-5	2	0	-4	-5	-5	-5	-4	-3	-3	-1	0	7			
G	5	-2	-3	-2	-4	-3	0	-1	-1	-3	1	0	-1	-7	-5					
H	6	-2	0	-2	-2	2	0	3	2	-1	-1	-2	-3	0						
<b>Most / Least Mutable AAs?</b>		I	5	-2	2	2	-2	-2	-2	-2	-2	-1	0	4	-5	-1				
		K	5	-3	0	1	-1	1	3	0	0	-2	-3	-4						
		L	6	4	-3	-3	-2	-3	-3	-3	-2	2	-2	-2	-1					
		M	6	-2	-2	-1	0	-2	-1	2	-4	-2								
<b>Elements of matrix are multiplied by 10</b>		N $\rightarrow$	2	-1	1	0	1	0	-2	-4	-2									
		P	6	0	0	1	0	-1	-6	-5										
		Q	4	1	-1	-1	-2	-5	-4											
		R	6	0	-1	-2	2	-4												
		S $\rightarrow$	2	1	-1	-2	-3													
		T	3	0	-5	-3														
		V	4	-6	-2															
		W $\rightarrow$	17	0																
		Y	10																	

+ve values represent evolutionarily  
conservative replacements

# PAM Matrices

## Salient Points:

- Matrices for greater evolutionary distances are extrapolated from those for lesser ones
- Number with the matrix (PAM40, PAM100, PAM250) refers to the evolutionary distance; larger numbers represent greater distances
- No clear correspondence between PAM distance and evolutionary time, since different protein families evolve at different rates
- Does not take into account different evolutionary rates between conserved & non-conserved regions.

# BLOSUM – BLocks Substitution Matrix

- Uses an alternative approach to determine a family of scoring matrices
- Uses structurally conserved protein domains from **BLOCKS database**, which contains ungapped multiple alignments, called blocks, of core regions from hundreds of proteins
- Directly tabulates frequencies  $p(x,y)$  for distantly related proteins, instead of having a need to extrapolate from observation
  - makes them more suitable for finding remote homologies and functionally related proteins but yields no evolutionary model.

# BLOSUM Matrices

Each matrix is tailored to a particular evolutionary distance, *viz.*, for BLOSUM62 matrix,

In each block, sequences sharing at least 62% AA identity are clustered together

Then frequencies of aligned pairs  $p(x, y)$  counted

Sequences more identical than 62% are represented by a single sequence in the alignment so as to avoid over-weighting closely related family members.

For each AA pair substitution scores are computed as:

$$\log \frac{\text{Pair\_freq (obs)}}{\text{Pair\_freq (expected)}}$$

$S_{i,j}$  are scaled to give an integer value

		BLOSUM50 matrix																			
		BLOSUM50 matrix																			
		BLOSUM50 matrix																			
A	5																				
R	-2	7																			
N	-1	-1	7																		
D	-2	-2	2	8																	
C	-1	-4	-2	-4	13																
Q	-1	1	0	0	-3	7															
E	-1	0	0	2	-3	2	6														
G	0	-3	0	-1	-3	-2	-3	8													
H	-2	0	1	-1	-3	1	0	-2	10												
I	-1	-4	-3	-4	-2	-3	-4	-4	-4	5											
L	-2	-3	-4	-4	-2	-2	-3	-4	-3	2	5										
K	-1	3	0	-1	-3	2	1	-2	0	-3	-3	6									
M	-1	-2	-2	-4	-2	0	-2	-3	-1	2	3	-2	7								
F	-3	-3	-4	-5	-2	-4	-3	-4	-1	0	1	-4	0	8							
P	-1	-3	-2	-1	-4	-1	-1	-2	-2	-3	-4	-1	-3	-4	10						
S	1	-1	1	0	-1	0	-1	0	-1	-3	-3	0	-2	-3	-1	5					
T	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-1	-2	-1	2	5			
W	-3	-3	-4	-5	-5	-1	-3	-3	-3	-2	-3	-1	1	-4	-4	-3	15				
Y	-2	-1	-2	-3	-3	-1	-2	-3	2	-1	-1	-2	0	4	-3	-2	-2	2	8		
V	0	-3	-3	-4	-1	-3	-3	-4	-4	4	1	-3	1	-1	-3	-2	0	-3	-1	5	
A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V		

## BLOSUM50 matrix:

- Positive scores on diagonal (identities)
  - Similar residues get positive scores (marked in red)
  - Dissimilar residues get smaller (negative) scores

# BLOSUM 62 Matrix

Table 2 – The log odds matrix for BLOSUM 62

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

Default matrix in BLAST

# BLOSUM Matrices

## Salient Points:

- Derived from **local, ungapped alignments** of distantly related sequences
- Uses **blocks of protein sequence fragments** from different families (the BLOCKS database)
- Blocks represent **structurally conserved** regions
- Amino acid pair frequencies calculated by **summing** over all possible pairs in block
- Different evolutionary distances are incorporated into this scheme with a **clustering procedure** (identity over particular threshold = same cluster)

# BLOSUM Matrices

- All matrices are **directly calculated**; no extrapolations are used – no explicit model
- Number after the matrix (BLOSUM62) refers to the minimum percent identity of the blocks used to construct the matrix; greater numbers represent lesser distances.
- **BLOSUM 62** is the default matrix in BLAST
- For database searches, Blosum matrices have often been the better performers
  - the reason being these matrices are based on the replacement patterns found in more highly conserved regions of the sequences.

# BLAST: Nucleotide Scoring Matrix

- Scoring Matrices for match (M)/mismatch (N):
  - 1, -2
  - 1, -3
  - 1, -4
  - 2, -3 ← default
  - 4, -5
  - 1, -1

Relative magnitudes of M & N determines the No. of nucleic acid PAMs (point accepted mutations per 100 residues) for which they are most sensitive at finding homologs.

# BLAST: Nucleotide Scoring Matrix

**A ratio of 0.33 (1/-3) is appropriate for sequences that are about 99% conserved**

**A ratio of 0.5 (1/-2) is best for sequences that are 95% conserved**

**A ratio of one (1/-1) is best for sequences that are 75% conserved**

**- the (absolute) reward/penalty ratio should be increased as one looks at more divergent sequences**

# Qs

1. A query sequence is searched in two databases of different lengths,  $L_1$  and  $L_2$ , with  $L_2 > L_1$ . The significance of the alignment of a certain query sequence 'S' is observed to  $E_1$  and  $E_2$  respectively, in the two databases. What can we say about relationship between  $E_1$  and  $E_2$ :
  - A.  $E_1 < E_2$
  - B.  $E_1 > E_2$
  - C.  $E_1 = E_2$ , since it is the alignment of same pair of sequences
  - D. Not enough information
2. Two alignments of lengths  $L_1 = 20$ , and  $L_2 = 100$  with 100% identity are obtained on querying a database, with e-value  $E_1$  and  $E_2$  respectively. Is
  - (A)  $E_1 = E_2$
  - (B)  $E_1 > E_2$
  - (C)  $E_1 < E_2$
  - (D) Not enough information

# Multiple Sequence Alignment (MSA)

# Multiple Sequence Alignment

Most important contribution of MB to **evolutionary analysis** is the discovery that DNA sequences of different organisms are often **related**.

i.e., genes are **conserved** across widely divergent species, often performing a **similar** or even identical function, and at other times, mutating or rearranging to perform an **altered** function.

Through **simultaneous alignment** of gene sequences, sequence patterns that have been subject to alteration may be analyzed.

# Multiple Sequence Alignment

Aligning more than two sequences

A	C	-	-	B	C	D	B
-	C	A	D	B	-	D	-
A	C	A	-	B	C	D	-

In an MSA, homologous residues among a set of sequences are aligned together in columns.

'Homologous' is meant in both the structural and evolutionary sense.

# Motivation for MSA

- MSA helps identify conserved regions and those allowed to vary - regions resistant to change are functionally most important to the molecule.
- Carries more information than mere pair-wise alignment
- Multiple sequence similarity suggests common structure for the protein, a common function or evolutionary origin
- MSA requirements are different in the various applications

# MSA for DNA Sequences

In DNA sequences MSA is used in

- **Genome sequence assembly - shotgun sequencing**
- **Discovering new regulatory elements**
- **Inferring evolutionary relationships**
- **DNA barcoding**
- **SNP identification**
- **Develop primers & probes - use conserved regions to develop**
  - **Primers for PCR of related sequences**
  - **Probes for DNA microarrays**

**In which of these applications do we look for similarity/differences?**

# MSA for DNA Sequences

Can you think of an application of MSA in  
the analysis of *SARS-CoV-2*?

# MSA for Protein Sequences

In protein sequences, MSA is used in

- Homology modeling of proteins
- Building phylogenetic tree
- Constructing scoring matrices - PAM, BLOSUM
- Predicting secondary & tertiary structures of new sequences
- Identifying conserved patterns, motifs, blocks in protein sequences - to characterize protein families
- Identify related proteins in database searches, e.g., Profiles, PSI-BLAST, HMMs

# Motivation for MSA

Multiple alignments can improve pairwise alignments:

(A)	p110 $\alpha$	TFILGIGDRHNSNIMVKDDG-QLFHIDFGHFLDHKKKKFGYKRERVPFVLT--QDFLIVI	142
	cAMP-kinase	QIVLTFEYLHSLDLIYRDLKPENLLIDQQGYIQVTDFGFAKRVKGRTWXLCGTPPEYLAPE	179
(B)	p110 $\beta$	SYVLGIG-----DRHSDNINVKKTGQLFHIDFGHILGNFKSKFGIKRERVPFILT	136
	p110 $\delta$	TYVLGIG-----DRHSDNIMIRESGQLFHIDFGHFLGNFKTKFGINRERVPFILT	136
	p110 $\alpha$	TFILGIG-----DRHNSNIMVKDDGQLFHIDFGHFLDHKKKKFGYKRERVPFVLT	135
	p110 $\gamma$	TFVLGIG-----DRHNDNIMITETGNLFHIDFGHILGNYKSFLGINKERVPFVLT	135
	p110_dicti	TYVLGIG-----DRHNDNLMVTKGGRLFHIDFGHFLGNYKKFGFKRERAPFVFT	135
	cAMP-kinase	QIVLTFEYLHSLDLIYRDLKPENLLIDQQGYIQVTDFGFAKRVKGRTWXLCG--TPEYLA	177

Catalytic domains of 5 P13-kinases and cAMP-dependent protein kinase

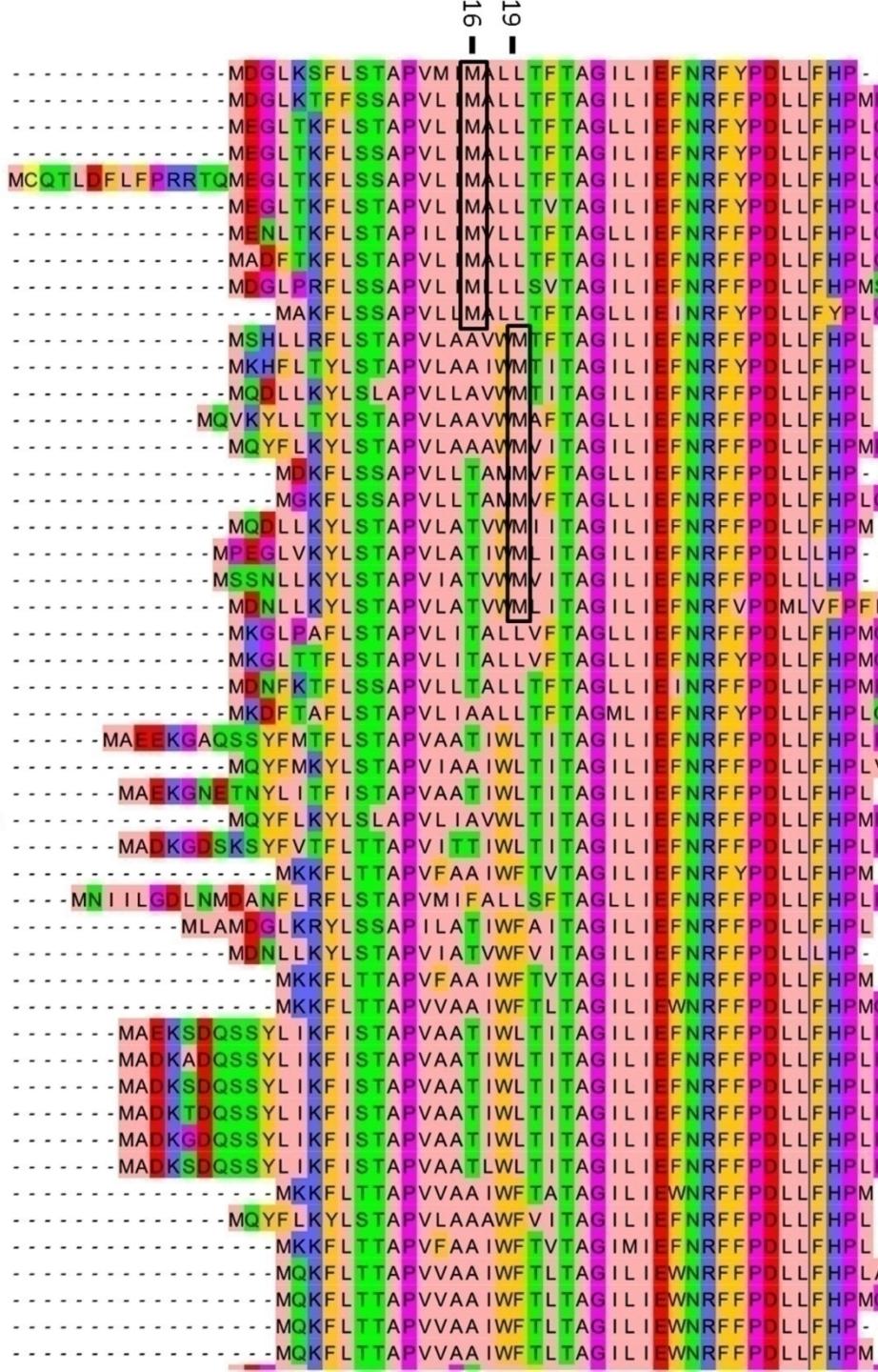
# MSA

Visual alignment of MSA tables use different colours for displaying AAs of different physico-chemical type - aids in identifying conserved patterns:

Colour	Residue Type	Amino acids
Yellow	Small nonpolar	Gly, Ala, Ser, Thr
Green	Hydrophobic	Cys, Val, Ile, Leu, Pro, Phe, Tyr, Met, Trp
Magenta	Polar	Asn, Gln, His
Red	Negatively charged	Asp, Glu
Blue	Positively charged	Lys, Arg

Structure prediction tools also give more reliable results when based on MSAs than on single sequences.

*Synechocystis\_sp.\_PCC\_6803/1-40*  
*Halothece\_sp.\_PCC\_7418/1-42*  
*Cyanothece\_sp.\_PCC\_8802/1-42*  
*Microcystis\_aeruginosa\_DIANCHI905/1-42*  
*Microcystis\_aeruginosa\_NIES-843/1-56*  
*Cyanothece\_sp.\_ATCC\_51472/1-42*  
*cyanobacterium\_UCYN-A/1-42*  
*Crocospaera\_watsonii\_WH\_8501/1-42*  
*Microcoleus\_sp.\_PCC\_7113/1-42*  
*Cyanothece\_sp.\_PCC\_7822/1-39*  
*Chamaesiphon\_minutus\_PCC\_6605/1-41*  
***Thermosynechococcus\_elongatus\_BP-1/1-41***  
*Moorea\_producens\_3L/1-41*  
*Synechococcus\_sp.\_PCC\_6312/1-43*  
*Oscillatoria\_nigro-viridis\_PCC\_7112/1-42*  
*Synechococcus\_sp.\_PCC\_7002/1-37*  
*Leptolyngbya\_sp.\_PCC\_7376/1-39*  
*Trichodesmium\_erythraeum\_IMS101/1-41*  
*Leptolyngbya\_sp.\_PCC\_7375/1-41*  
*Synechococcus\_sp.\_PCC\_7335/1-41*  
*Pseudanabaena\_sp.\_PCC\_7367/1-42*  
*Cyanobacterium\_stanieri\_PCC\_7202/1-42*  
*Cyanobacterium\_aponinum\_PCC\_10605/1-42*  
*Dactylococcopsis\_salina\_PCC\_830/1-42*  
*Gloeocapsa\_sp.\_PCC\_73106/1-42*  
*Nodularia\_spumigena\_CCY9414/1-50*  
*Oscillatoriales\_cyanobacterium\_JSC-12/1-43*  
*Anabaena\_sp.\_90/1-48*  
*Coleofasciculus\_chthonoplastes\_PCC\_7420/1-42*  
*Calothrix\_sp.\_PCC\_7507/1-49*  
*Synechococcus\_sp.\_CB0101/1-38*  
*Calothrix\_sp.\_PCC\_6303/1-52*  
*Synechococcus\_elongatus\_PCC\_7942/1-44*  
*Leptolyngbya\_sp.\_PCC\_6406/1-40*  
*Synechococcus\_sp.\_CB0205/1-38*  
*Synechococcus\_sp.\_CC9311/1-39*  
*Nostoc\_sp.\_PCC\_7107/1-49*  
*Nostoc\_sp.\_PCC\_7120/1-49*  
*Nostoc\_azollae'\_0708/1-49*  
*Nostoc\_sp.\_PCC\_7524/1-49*  
*Nostoc\_punctiforme\_PCC\_73102/1-49*  
*Cylindrospermum\_stagnale\_PCC\_7417/1-49*  
*Synechococcus\_sp.\_BL107/1-38*  
*Lyngbya\_sp.\_PCC\_8106/1-41*  
*Synechococcus\_sp.\_RCC307/1-38*  
*Synechococcus\_sp.\_WH\_7805/1-39*  
*Synechococcus\_sp.\_RS9917/1-39*  
*Synechococcus\_sp.\_WH\_7803/1-37*  
*Synechococcus\_sp.\_RS9916/1-38*



# MSA

To be informative a MSA should

- contain a distribution of **closely-** and **distantly-** related sequences.

If all **closely-related** - information contained is largely redundant

⇒ **few inferences can be drawn.**

If all **very distantly-related** - difficult to construct an accurate alignment

⇒ **quality of results & inferences might be questionable**

Ideally, one should have a **complete range of similarities**, including distantly-related examples linked through chains of close relationships

# Inferences from MSA

Some examples:

- Identify highly conserved regions - likely to be essential sites for structure/function, e.g. active site
- Regions rich in insertions/deletions - may correspond to loops/turns in proteins
- Build gene/protein families - use conserved regions to guide search
- Basis for phylogenetic analysis - infer evolutionary relationships between genes

# Inferences from MSA

## In Secondary Structure Prediction:

- Conserved pattern of hydrophobicity with spacing 2 with intervening residues more variable and including hydrophilic residues - suggests  $\beta$ -strand
- Conserved pattern of hydrophobicity with spacing 4 suggests a helix

# 12 glutamyl tRNA<sup>10</sup> reductase sequences

13 - 16

E. coli  
Axc  
Synts  
Cypar  
Oleg  
Horvu  
Arab  
Cupep  
B. subt.  
Chvib  
Cjoj

	20	30	40	
MT	L L A L G I	N H K T A P V S L	R V T F S P D T L	D Q A L D S
MT	L L A L G I	N H K T A P V S L	R V S F S P D K L	D Q A L D S
MT	L W V L G L	N H Q T A P V D L	R A A F A G D A L	P R A L E S
MNI	A V V V G L	S H K T A P V E I	K L S I Q E A K L	E E A L T H
MNI	I V V V G L	S H K T A P V D F	I P K V R I G E A I	R E
S A A D R Y M K E K S S I	A V I V G L	S V H T A P V E M	M R E K L A V A E E L	W P R A I S E
S A A D R Y I K E K S S I	A V I V G L	S V H T A P V D M	M R E K L A V A E E L	W P R A I S E
S A A D R Y T K E R S S I	V V I V G L	S I H T A P V E E	M R E K L A I P E A E W P R A I A E	
S S V N R Y T K E R I S I	V V I V G L	N V H T A P V E L L	R E K L A I P E A Q W P P G I G E	
M H I	L V V V G V D Y K S A P I	E I I R E K V S F Q P N E L A E A M V Q		
M N I	I S V V G V N H K T A P I	E I I R E R I A L S E V Q N K E F V T D		
S I K K R F R M Y I L S I	S A S L D Y K S A A I	D I R E R F S Y T S T R I R E I L R R		

38 - 45  
α-helix  
i, i+3,  
i+4, i+7

PHD Sec. Pred.



PHD Acc. Pred.



SOPMA Sec. Pred.



SSPRED Sec. Pred.



Conservation



Consensus



hydrophobic

Non-hydrophobic (active sites)

## JPRED server

E. coli  
Axc  
Synts  
Cypar  
Oleg  
Horvu  
Arab  
Cupep  
B. subt.  
Chvib  
Cjoj

	50	60	70	80
L L A Q P M V	Q G G V V L	S T C N R T	E L Y L S V E E	R W L C D Y H N
L L A Q P M V	Q G G V V L	S T C N R T	E L Y L S V E E	R W L C D Y H N
L R A L P Q V S	E A A L L	S T C N R T	E L Y A M A E E A H	S L V T W L E T H
L R S Y P H I	E E V T V I	S T C N R L	E I Y Y A V V T D T E K G V V E I T Q F L	S E T G N
L C N Y P H I	E E V A I L	S T C N R L	E I Y Y V V A L S D T Y Q G I R E A T Q F L	A D S S D
L T S L N H I	E E A A V L	S T C N R M	E I Y Y V V A L S D T Y Q G I R E A T Q F L	A D S S D
L T S L N H I	E E A A V L	S T C N R M	E I Y Y V V A L S D T Y Q G I R E A T Q F L	A D S S D
L C G L N H I	E E A A V L	S T C N R M	E I Y Y V V A L S D T Y Q G I R E A T Q F L	A D S S D
L C A L N H I	E E A A V L	S T C N R I	E I Y Y V V A L S D Q L H T G R Y Y I K K F L	A D W F Q
L K E E K S I L E	E N I I V	S T C N R T	E I Y Y A V V D Q L H T G R Y Y I K K F L	A D W F Q
L V S S G L A S E	A M V V	S T C N R T	E L Y V V P Q M P E V N C D Y L K D Y I I S Y K D	
I K A A D G V S	G A V L L	C T C N R T	E L Y I S G D N I E N M N P A L L C Q L S G E E D	

PHD Sec. Pred.



PHD Acc. Pred.



SOPMA Sec. Pred.



SSPRED Sec. Pred.



Conservation



Consensus



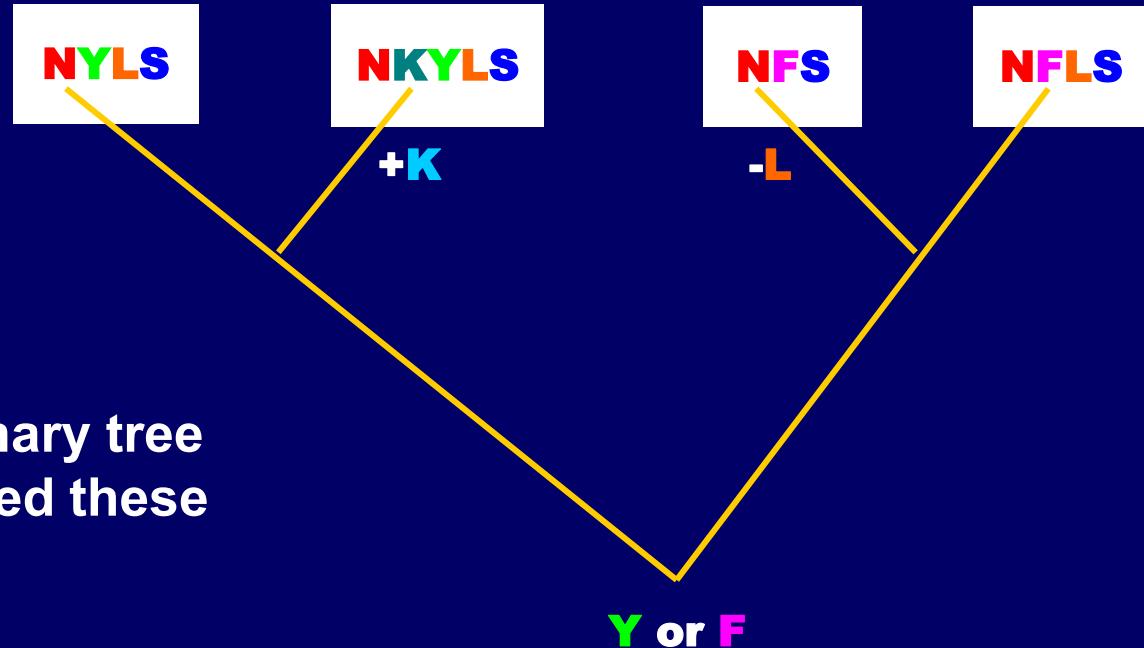
# Relationship between MSA & Phylogenetic Analysis

SeqA N – F L S

SeqB N – F – S

SeqC N K Y L S

SeqD N – Y L S



A hypothetical evolutionary tree  
that could have generated these  
sequence changes.

## MSA of 8 fragments of immunoglobulin sequences

VTISCTGSSSNIGAG-NHVKWYQQQLPG
VTISCTGTSSNIIGS--ITVNWYQQQLPG
LRLSCSSSGFIFSS--YAMYWVRQAPG
LSLTCTVSGTSFDD--YYSTWVRQPPG
PEVTCVVVVDVSHEDPQVKFNWYVDG--
ATLVCLISDFYPGA--VTVAWKADS--
AALGCLVKDYFPEP--VTVSWNSG---
VSLTCLVKGFYPSD--IAVEWESNG--

Conserved residues, regions, patterns

alternating hydrophobicity pattern -  $\beta$ -strand,  
conserved Cys, W

# Multiple Alignment: Evaluation

VTISCTGSSNIG-AGNHVKWYQQLPG

VTISCTGTSSNIG--SITVNWYQQLPG

LRLSCSSSGFIFS--SYAMYWVRQAPG

LSLTCTVSGTSFD--DYYSTWVRQPPG

PEVTCVVVDVSHDPQVKFNW--YVDG

ATLVCLISDFYPG--AVTVAW--KADS

AALGCLVKDYFPE--PVTVSW--NS-G

VSLTCLVKGFYPS--DIAVEW--ESNG

VTISCTGSSNIGAG-NHVVKWYQQLPG

VTISCTGTSSNIGS--ITVNWYQQLPG

LRLSCS-SSGFIFSS-YAMYWVRQAPG

LSLTCT-VSGTSFDD-YYSTWVRQPPG

PEVTCVVVDVSHDPQVKFNWYVDG--

ATLVCLISDFYPGA--VTVAWKADS--

AALGCLVKDYFPEP--VTWSWNSG---

VSLTCLVKGFYPSD--IAVEWESNG--

*It is not enough to just get a multiple alignment;  
we need to score the alignment*

# Multiple Alignment: Evaluation

A simple way to evaluate a multiple alignment is to evaluate the cost column by column

Sum of Pairs (SP)

$$= \sum_{i < j} D(S_i, S_j)$$

Using unit cost:

mismatch costs 1,  
match 0, and  
indel costs 1

$$ColumnCost \begin{pmatrix} L \\ L \\ A \\ P \\ G \\ S \\ - \\ G \end{pmatrix} = ?$$

Summing the scores of all possible combinations of AA pairs in a column of MSA

# Multiple Alignment: Evaluation

$$ColumnCost \left( \begin{array}{c} L \\ L \\ A \\ P \\ G \\ S \\ - \\ G \end{array} \right) = 26.$$

Assumes a model for evolutionary change in which any of the sequence could be the ancestor of others

# SP Scoring Method

- There are problems with SP scoring system as illustrated in the example:

Sequence	Col. A	Col. B	Col. C
1	...N...	...N...	...N...
2	...N...	...N...	...N...
3	...N...	...N...	...N...
4	...N...	...N...	...C...
5	...N...	...C...	...C...
Score	60	24	9

(Using Blosum62):

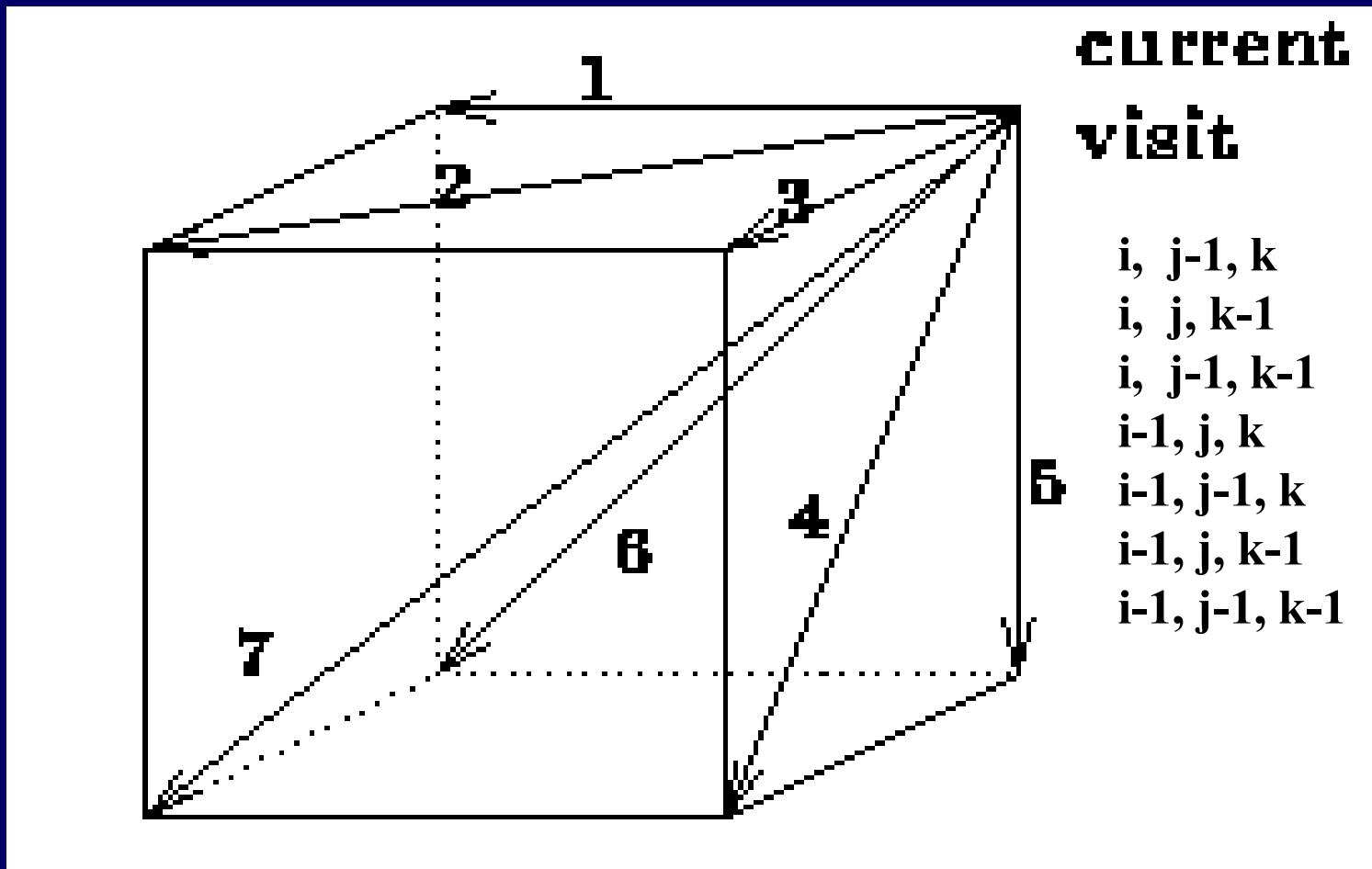
N-N: 6, N-C: -3, C-C: 9

What's the problem?  
Score for N = 10 seq

# Multiple Alignment: DP

- Pair-wise alignment: DP involves a  $L \times L$  ( $L^2$ ) matrix
- Multiple alignment: DP involves an  $L^N$  matrix, an  $N$ -dimensional hyper-lattice,  $N$  - no. of sequences of length  $L$  each, to align simultaneously
- Computationally not feasible: for 5 sequences, each  $\sim 100$  bp long,  $10^{10}$  matrix elements need to be computed
  - equivalent to a pairwise alignment of two 100,000 bp sequences

# The Recursive Relation



For 3 sequences, to assign a value to a node  $(i, j, k)$ , we need to consider 7 values; for 2 seqs, we needed only 3!

# Multiple Sequence Alignment

$\alpha_{i_1, i_2, \dots, i_N}$  - maximum score of an alignment up to the subsequences ending with  $x_{i_1}^1, x_{i_2}^2, \dots, x_{i_N}^N$

Recursive relation for multiple sequences:

$$\alpha_{i_1, i_2, \dots, i_N} = \max \left\{ \begin{array}{ll} \alpha_{i_1-1, i_2-1, \dots, i_N-1} & + S(x_{i_1}^1, x_{i_2}^2, \dots, x_{i_N}^N), \\ \alpha_{i_1, i_2-1, \dots, i_N-1} & + S(-, x_{i_2}^2, \dots, x_{i_N}^N), \\ \alpha_{i_1-1, i_2, i_3-1, \dots, i_N-1} & + S(x_{i_1}^1, -, \dots, x_{i_N}^N), \\ \cdot & \\ \cdot & \\ \cdot & \\ \alpha_{i_1-1, i_2-1, \dots, i_N} & + S(x_{i_1}^1, x_{i_2}^2, \dots, -), \\ \alpha_{i_1, i_2, i_3-1, \dots, i_N-1} & + S(-, -, \dots, x_{i_N}^N), \\ \cdot & \\ \cdot & \\ \cdot & \\ \alpha_{i_1, i_2-1, \dots, i_{N-1}-1, i_N} & + S(-, x_{i_2}^2, \dots, -), \\ \cdot & \\ \cdot & \\ \cdot & \end{array} \right.$$

# Multiple Sequence Alignment

To calculate each entry, need to maximize over all  $2^N - 1$  combinations of gaps in a column, excluding the case where all  $\Delta_k$  are zero.

Introducing the notation  $\Delta_i$ , which is 0 or 1 and define the 'product'

$$\Delta_i \cdot x = \begin{cases} x & \text{if } \Delta_i = 1, \\ - & \text{if } \Delta_i = 0. \end{cases}$$

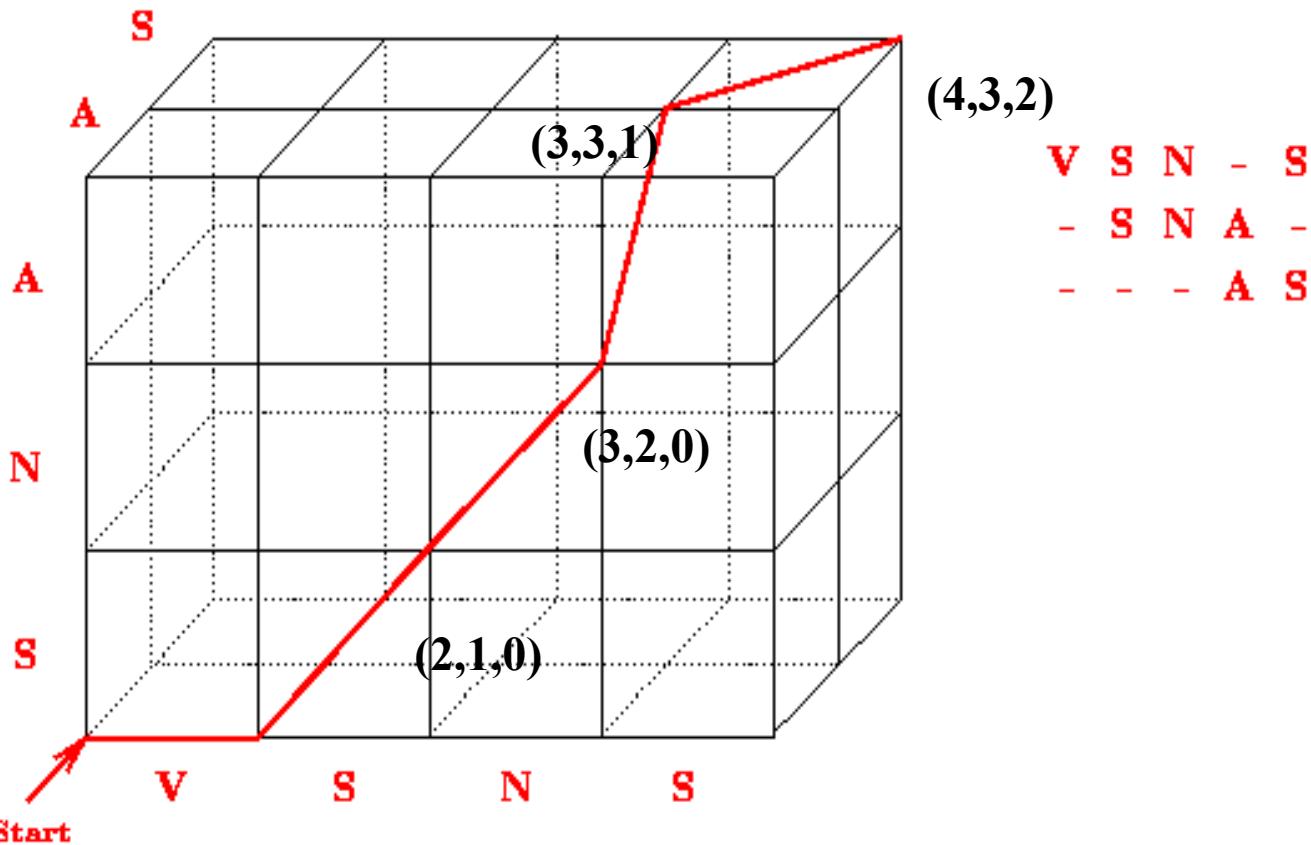
Recursion relation can now be written as

$$a_{i_1, i_2, \dots, i_N} = \max_{\Delta_1 + \dots + \Delta_N > 0} \left\{ a_{i_1 - \Delta_1, i_2, -\Delta_2, \dots, i_N - \Delta_N} + S(\Delta_1 \cdot x_{i_1}^1, \Delta_2 \cdot x_{i_2}^2, \dots, \Delta_N \cdot x_{i_N}^N) \right\}$$

For  $N = 3, 4, 5$ , and  $10$ ,  $2^N - 1 = ?$

# DP Hyperlattice: Example

Figure 1: Alignment Path for 3 Sequences.



Overall score  $S(m)$  for an alignment is defined as a sum of scores  $S(m_i)$  for each column  $i$ :  $S(m) = \sum_i S(m_i)$

# Time and Space complexity

Assuming all sequences of roughly same length  $L$ , memory complexity of the multi-dimensional DP algorithm is  $O(L^N)$  and time complexity is  $O(2^NL^N)$

- impractical for more than a few sequences.

For 6 sequences, each 100bp long, time taken will be  $2^6 \times 100^6 \times 10^{-9} = 64000$  seconds ( $\sim 18$  hrs)

Add 2 more sequences of same length and the no. is  $2.56 \times 10^9$  seconds (over 81 yrs)

Even worse is memory space requirement -  $10^{12}$  for 6 sequences!

# Carrillo-Lipman/MSA Algorithm

- Carrillo and Lipman proposed a heuristic method for accelerating the search, implemented in the program called **MSA**.
- It is based on the property that if the strings are relatively **similar**, the alignment path would be **close to the main diagonal**.
  - ⇒ not all values in the multi-dimensional cube need to be calculated.
- Their central idea is - every multiple alignment imposes a pw-alignment on each pair of sequences
- For each pair, it sets the upper bound equal to the cost of the imposed alignment.

MSA can be projected onto the sides of cube, defining no. of positions within the cube to be evaluated

sequence B

sequence C

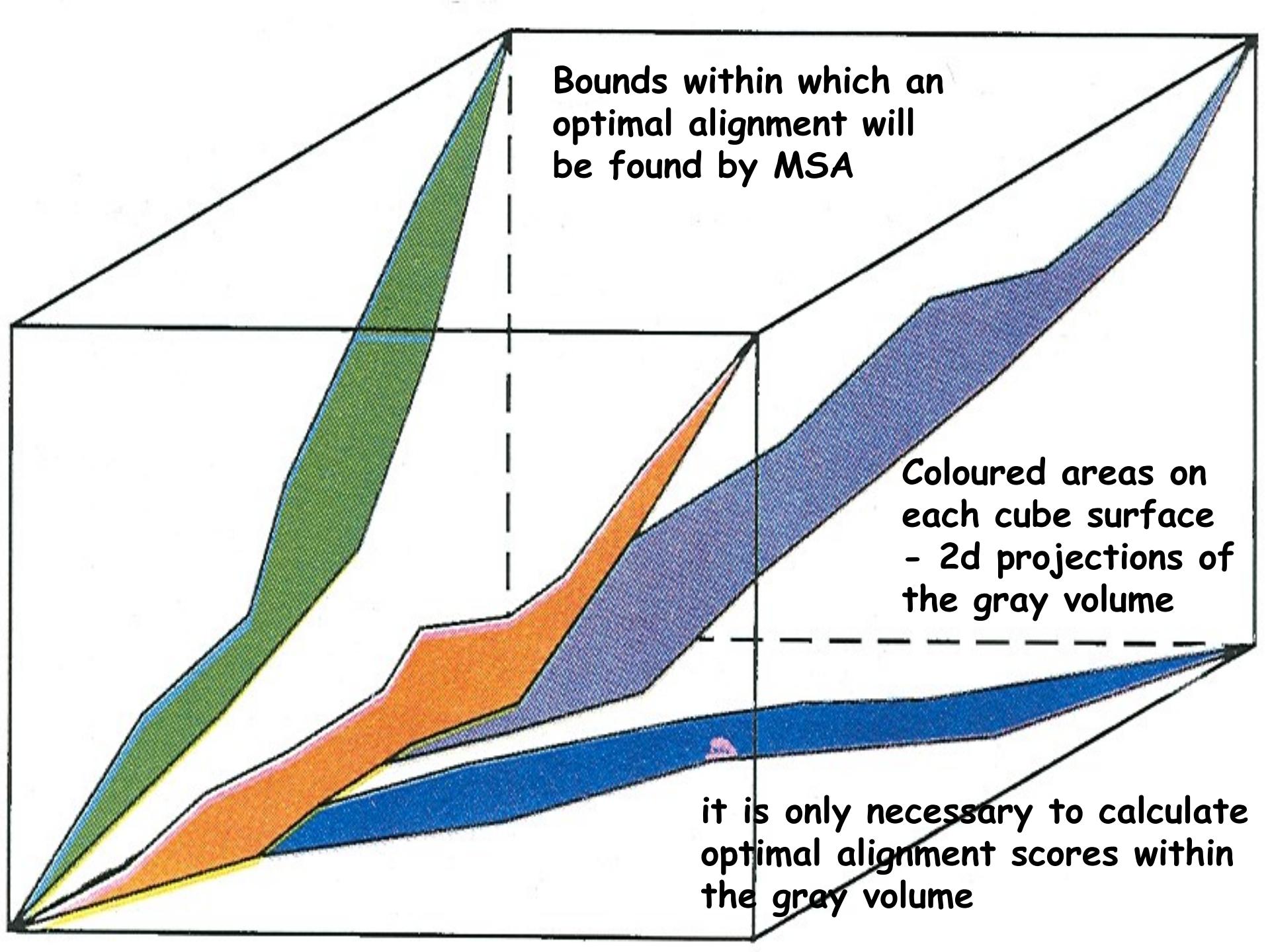
Pairwise alignment

B-C

A-B

A-C

sequence A



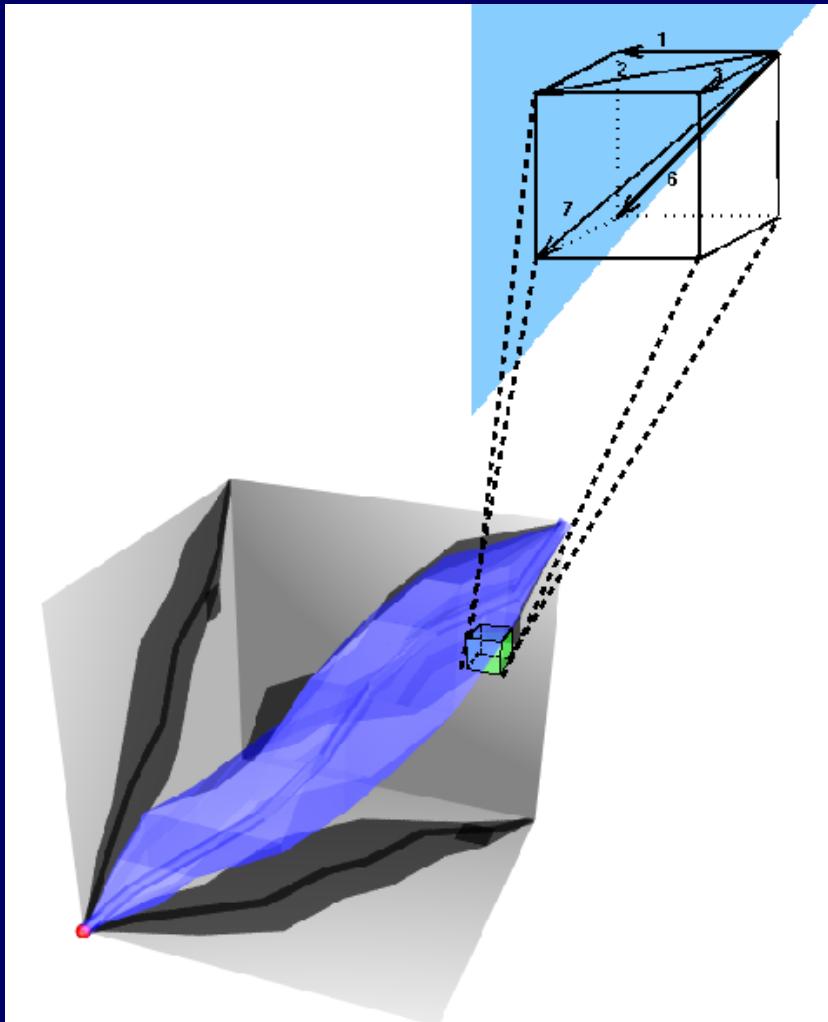
Bounds within which an optimal alignment will be found by MSA

Coloured areas on each cube surface - 2d projections of the gray volume

it is only necessary to calculate optimal alignment scores within the gray volume

# Multiple alignment DP: Heuristics

Optimal alignment path will lie around the main diagonal



While visiting a node and looking for the minimum along all the incoming edges - ignore the edges “coming from outside polyhedron”. Edges 1, 2, 3, 6, 7 are coming from inside and edges 4 and 5 (not shown) can be ignored

useful if the sequences are similar

(Carillo-Lipman, 1988)

# Multiple alignment DP: Heuristics

- MSA uses a heuristic approach to choose upper bounds for the pairs.
- Works for very similar sequences
- MSA can find optimal multiple alignments for ~ 6-8 sequences, 200-300 length

# Alignment by Consensus

- Aim of alignment by Consensus: to minimize the consensus error
- Align sequence 1 & 2 and obtain a consensus sequence
- Building consensus: an example

A	V	-	C	-	D
A	V	C	C	-	D
A	V	C	D	E	-
<hr/>					
A	V	C	S	E	D

- As is clear from above, the consensus need not be similar to any of the aligned sequences

# Consecutive consensus

- Align sequence 3 to the consensus of first two sequences to obtain a new consensus
- Continue until the consensus alignment converges to a global consensus
- Problem - Depends on the order in which the sequences are introduced

Solution ?

# Consecutive consensus

- Align sequence 3 to the consensus of first two sequences to obtain a new consensus
- Continue until the consensus alignment converges to a global consensus
- Problem - Depends on the order in which the sequences are introduced
- Solution - use a guide tree obtained by p-w alignment scores to build the consensus

# Progressive approach

- Align each sequence to every other pair-wise
- Compute distances between each aligned pair (e.g., no. of mismatches)
- Construct a phylogenetic tree
- Cluster closely related sequences
- Align closely related sequences first
- Gaps inserted in closely related sequences are propagated throughout

Progressive alignment - involves constructing a succession of pairwise alignments.

Tools: *ClustalW, T-Coffee, MUSCLE*

# Pairwise alignments

Sheep STCVLSAYWKDLNNTYH

Cattle STCVLSAYWKDLNNTYH

Sheep STCVL9AYWK-DLNNYH

Pig STCVLSAYWKNELNNTYH

Sheep STCVLSAYWKDLNNTYH

Human STCKLGY - QDFNKFH

Sheep STCVLSAYWKDLNNTYH

Rat STCKLGY - QDLNKFH

Sheep STCVLSAYWKD-LNNYH

Salmon STCVLGKL-SQELHKLQ

Pig STCVLSAYWKNELNNEH

Human STCKLGY - QD-ENKFH

Pig

Rat

Pig

Salmon

Human

Rat

Human

Salmon

Rat

Salmon

STCVLSAYWKNELNNTYH

STCKLGY - QD-LNKFH

STCVLSAYWKNELNNTYH

STCVLGKL-SQELHKLQ

STCKLGY QD**E**NKFH

STCKLGY QD**L**NKFH

STCKLGY - QDFNKFH

STCVLGKL-SQELHKLQ

STCKLGY - QDLNKFH

STCVLGKL-SQELHKLQ

STCKLGY - QD-ENKFH

# Hierarchy of Addition

<b>Sheep-Cattle</b>	<b>0</b>	<b>Pig-Rat</b>	<b>8</b>
<b>Sheep-Pig</b>	<b>4</b>	<b>Pig-Salmon</b>	<b>10</b>
<b>Sheep-Human</b>	<b>8</b>	<b>Human-Rat</b>	<b>1</b>
<b>Sheep-Rat</b>	<b>7</b>	<b>Human-Salmon</b>	<b>9</b>
<b>Sheep-Salmon</b>	<b>11</b>	<b>Rat-Salmon</b>	<b>8</b>
<b>Pig-Human</b>	<b>9</b>	...	

- Align Sheep and Cattle first
- Align Human and Rat
- Align Pig to Sheep and Cattle
- Align these two clusters to each other
- ...
- Align Salmon to large alignment last

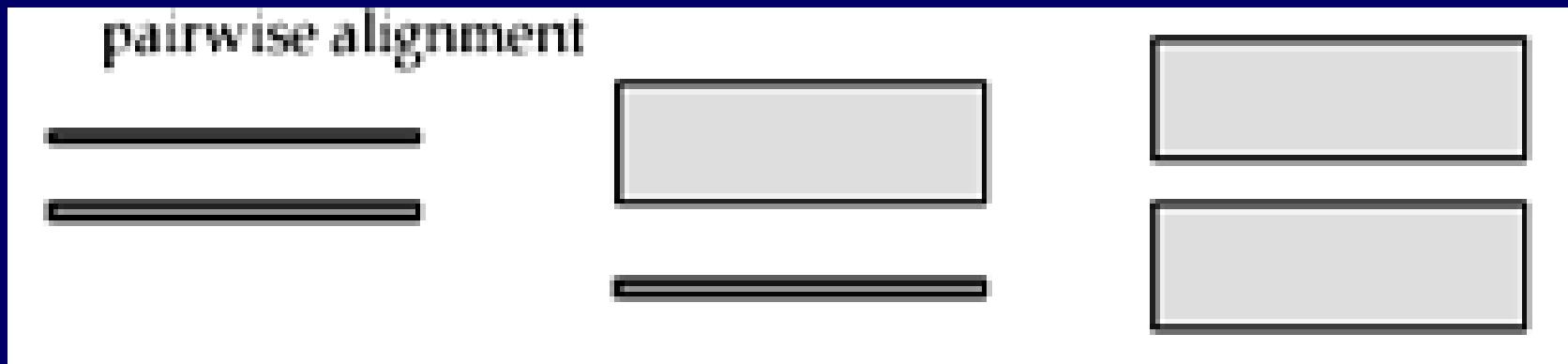
# Progressive alignment

step 1	Sheep	STCVLSAYWKDLNNYH
	Cattle	STCVLSAYWKDLNNYH
step 2	Human	STCMLGTYQDENKFH
	Rat	STCMLGTYQDLNKFH
step 3	Sheep	STCVLSAYWK-DLNNYH
	Cattle	STCVLSAYWK-DLNNYH
	Pig	STCVLSAYWRNELNNFH
...	Salmon	STCVLGKLSQE-LHKLQ

# Aligning Alignments

All possible cases that arise in progressive alignment approach:

- Align two sequences to each other
- Align a sequence to an existing alignment
- Align two alignments to each other



Note - computationally this is always a PW

# Aligning Alignments

Pairwise alignment of alignments is also called profile alignment

Again, we can use DP

$$S(i, j) = \max[S(i-1, j-1) + m(i, j), S(i-1, j) + g, S(i, j-1) + g]$$

$m(i, j)$  - similarity score averaged over characters at that position, here  $i$  no longer stands for a single sequence, but an average of several, and similarly for  $j$  when aligning alignments

$g$  - gap penalty

W T C V L S A YWKD-LNNYH

Sheep

U N T E C Y L U S A Y W K D - L N N Y H

## Cattle

S T C V L S A Y N B N E L N N F H

Pig

SS							
TT							
CC							
MM							
LL							
GG							
TT							
YY							
QQ							
DD							
FL							
NN							

# Aligning Alignments

Alignment 1:	ATA
	CCA
Alignment 2:	TCAFE
	TAT-E
	TATF-
	AGTFD



Score 1<sup>st</sup> column of 1<sup>st</sup> alignment against 2<sup>nd</sup> column in the other alignments using:

$$= \frac{1}{8} (\text{score}(A, C) + \text{score}(A, A) + \text{score}(A, A) + \text{score}(A, G) + \text{score}(C, C) + \text{score}(C, A) + \text{score}(C, A) + \text{score}(C, G))$$

# Aligning Alignments

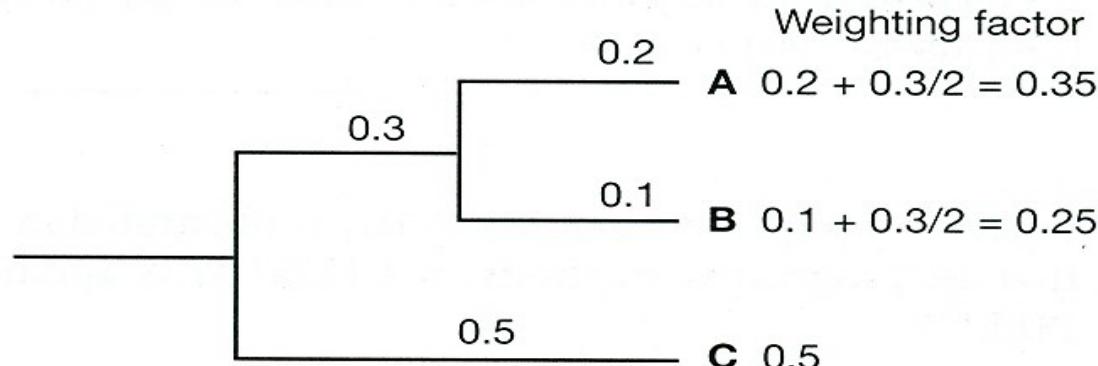
- Once sequences are aligned & gaps introduced, these are not altered - the alignment method is hierarchical
- ClustalW finds a local optimum as early alignment decisions are “locked in” by the “greedy” algorithm
- Early errors will be propagated and can cause the final alignment to be poor

# ClustalX/ClustalW

- Performs pair-wise alignments of all the sequences
  - k-tuple based alignment (fast/approx.), full DP (slow/accurate)
- Uses alignment scores to produce a phylogenetic tree
  - Genetic distance: no. of mismatches/no. of matches (positions against gaps not considered)
- Aligns sequences sequentially, guided by the phylogenetic relationships indicated by tree.
  - Sequence contributions are 'weighted' by their relationship on the predicted tree, weights based on the distance of each sequence from root

## Weighting scheme used by CLUSTALW

### A. Calculation of sequence weights



### B. Use of sequence weights

Column in alignment 1

Sequence A (weight a) ..... K.....  
Sequence B (weight b) ..... I.....

Column in alignment 2

Sequence C (weight c) ..... L.....  
Sequence D (weight d) ..... V.....

Score for matching these two column in an msa =

[ a x c x score (K,L) +  
a x d x score (K,V) +  
b x c x score (I,L) +  
b x d x score (I,V) ] / 4

## Basic idea in Progressive Heuristic Approach:

- compute pairwise alignments and merge alignments consistently

Consider alignment of 3 sequences:

acg, cga, gac

Get optimal pairwise alignments:

a c g -

- a c g

c g a -

- c g a

g a c -

- g a c

1&2

a	c	g	-
-	c	g	a

1&3

-	a	c	g
g	a	c	-

c	g	a	-
-	g	a	c

2&3

Merge using  
alignments  
with 1<sup>st</sup> sequence

-acg-  
--cga  
gac--

Merge using  
alignments  
with 3<sup>rd</sup> sequence

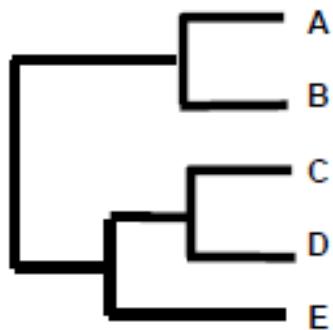
--acg  
cga--  
-gac-

Merge using  
alignments  
with 2<sup>nd</sup> sequence

acg--  
-cga-  
--gac

**Order of merging matters !**  
**Note once a gap, always a gap ...**

# Example



C PADKTNVKAAWGKVGAHAGEYGA

D AADKTNVKAAWSKVGGHAGEYGA

A PEEKSAVTALWGKVNVDEYGG

B GEEKAAVLALWDKVNEEEYGG

C PADKTNVKAAWG\_KVGAHAGEYGA

D AADKTNVKAAWS\_KVGGHAGEYGA

E AA\_\_TNVKTAWSSKVGGHAPA\_\_A

A PEEKSAV\_TALWG\_KVN\_VDEYGG

B GEEKAAV\_LALWD\_KVN\_EEEYGG

C PADKTNVKAA\_WG\_KVGAHAGEYGA

D AADKTNVKAA\_WS\_KVGGHAGEYGA

E AA\_\_TNVKTA\_WSSKVGGHAPA\_\_A

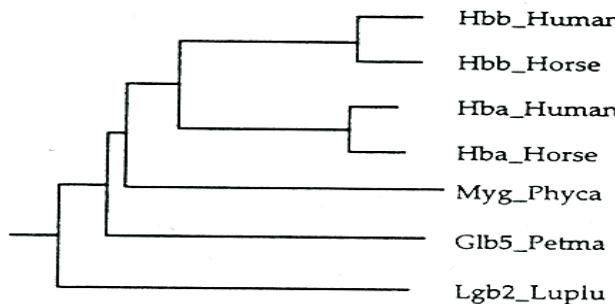
Once a gap, always a gap

# ClustalW: Refinements

- Different BLOSUM matrices used based on similarity of sequences - to reflect evolutionary changes better
- Recall: BLOSUM80/ BLOSUM62 are based on sequences that are 80% / 62% identical, i.e., lower numbers for more distant sequences
- ClustalW calculates gaps in a way to place them between conserved domains, based on observation that structurally related proteins align in a way to have gaps between secondary structural elements
- Frequency of gaps next to each amino acid computed in these regions is used (Pascarella and Argos)

Hbb_Human	1	-				
Hbb_Horse	2	.17	-			
Hba_Human	3	.59	.60	-		
Hba_Horse	4	.59	.59	.13	-	
Myg_Phyc	5	.77	.77	.75	.75	-
Glb5_Petma	6	.81	.82	.73	.74	.80
Lgb2_Luplu	7	.87	.86	.86	.88	.93 .90

## MSA of 7 globins by CLUSTALW



-----VHLTPEEKSAVTALWGKVN-----VDEVGGEALGRLLVVYWTQRFFESFGDLST  
 -----VQLSGEEKAAVLALWDKVN-----EEEVGGEALGRLLVVYWTQRFFESFGDLSN  
 -----VLSPADKTNVKAAWGKVGAHAGEYGAELERMFLSFPTTKTYFPHFDSL-----  
 -----VLSAADKTNVKAAWSKVGGHAGEYGAELERMFLGFPTTKTYFPHFDSL-----  
 -----VLSGEGEWQLVLHVWAKVEADVAHGQDILIRLFKSHPETLEKFDRFKHLKT  
 PIVDTGSVAPLSAAEKTKIRSAWAPVYSTYETSGVDILVKFFTSTHAAQEFFPKFKGLTT  
 -----GALTESQAAALVKSSWEETNANIPKHTHRRFFILVLEIAAAKDLFSFLKGTE

PDAVMGNPKVKAHKKVLGAFSDGIAHLD-----NLKGTFATLSELHCDKLHVDPENFRL  
 PGAVMGNPKVKAHKKVLHSFGEVHHLD-----NLKGTFAALESSELHCDKLHVDPENFRL  
 -----HGSAQVKGHGKVKADALTNAVAHVD-----DMPNALSALSSDLHAHKLRLDPVNFKL  
 -----HGSAQVKAHGKKVGDACTLAVGHLD-----DLPGALSNLSDLHAHKLRLDPVNFKL  
 EAEMKASEDLKKHGVTVLTAAGAILKKKG-----HHEAEELKPLAQSHATKHKIPIKYLEF  
 ADQLKKSSADVRVWHAERIINAVNDAVASMDTT-----EKMSMKLRLDSGKHAKSFQVDPQYFKV  
 VP-----QNNPELOQAHAGKVFKLVYEAATIQLQVTGVVVT-----DATLKNLGSVHVSKG-VADAHFPV

LGNVLVCVLAHFGKEFTPVQAYQKVVAGVANALAHKYH-----  
 LGNVLVCVLAHFGKDFTPELQASYQKVVAGVANALAHKYH-----  
 LSHCLLVTLAALHPLAEFTPVAHSAASLDKFLASVSTVLTSKYR-----  
 LSHCLLSTLAVHLPNDFTPAVHASLDKFLSSVSTVLTSKYR-----  
 ISEAIIHVLHSRHPGDFGADAQGAMNKAELFRKDIAAKYKELGYQG  
 LAAVIADTVAAAG-----DAGFEKLMMSMICILLRSAY-----  
 VKEAILKTIKEVVGAKWSEELNSAWTIAYDELAIVIKKEMNDAA-----

Pairwise alignment:  
 Calculate distance matrix

Rooted Neighbor Joining  
 tree (guide tree)

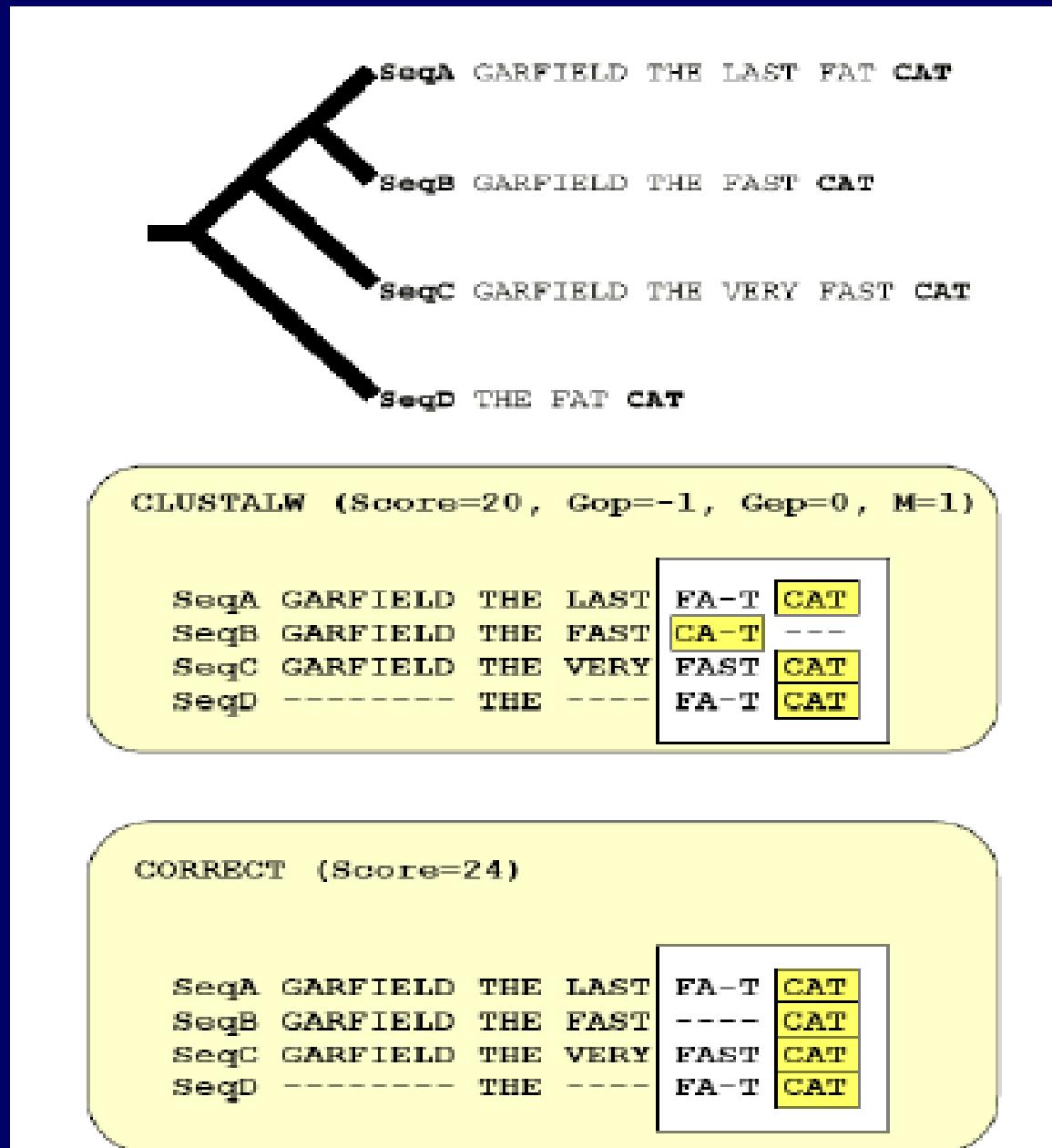
Progressive  
 alignment:  
 Align following  
 the guide tree

Known locations of 7  $\alpha$ -helices in the structure  
 of this group shown in  
 boxes

# Problems with Progressive Alignment

Gaps at the ends are penalized less, so CAT is aligned with FAT in sequence 2

The greedy approach results in efficiency of the algorithm at the cost of accuracy



# Problems with Progressive Alignment

## Local Minimum Problem:

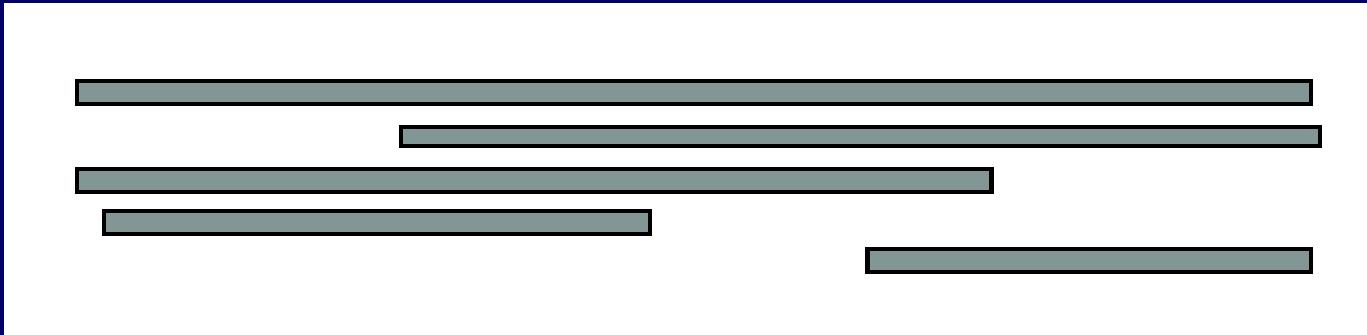
- Depends on the very first closely related sequences used for constructing the multiple alignment
- If these sequences align well, fewer errors
- More distantly related these sequences are, more errors will get propagated through the alignment

## Solution:

- Using Stochastic or Iterative Methods
- Using Bayesian methods such as HMMs
  - for aligning more distantly related sequences.

# ClustalW Misapplied

ClustalW and other algorithms that include an initial pair-wise comparison step should not be used to align sequences that do not all share a common block.



⇒ MSA of protein sequences should be done at domain level

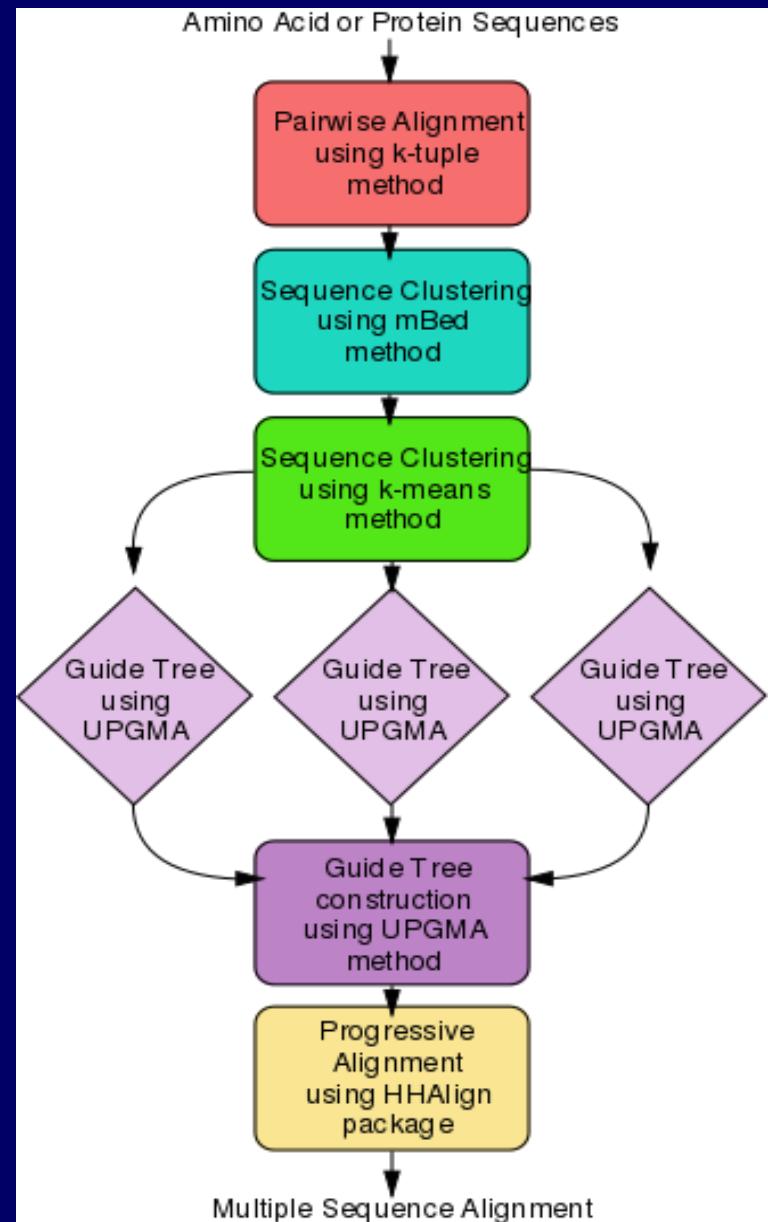
# ClustalΩ

- EBI:  
<https://www.ebi.ac.uk/Tools/msa/>

The latest version is called *Clustal Omega* - uses seeded guide trees and HMM profile-profile techniques to generate alignments

- one of the fastest multiple sequence alignment tools

- DDBJ:  
<http://clustalw.ddbj.nig.ac.jp/>



# Iterative Methods of MSA

- To correct for errors introduced by initial alignment, use iterative methods: re-align sub-groups of sequences and then align these sub-groups into a global alignment
- Objective – to improve overall alignment score
- Selection of groups may be based on phylogenetic tree, separation of one or two sequences from the rest, or a random selection of the groups.
- Programs using iterative methods – MultiAlin, PRRP and DIALIGN

# Other Methods

Genetic Algorithm: It is a machine learning approach, that produces alignments by attempted simulation of the evolutionary changes in sequences.

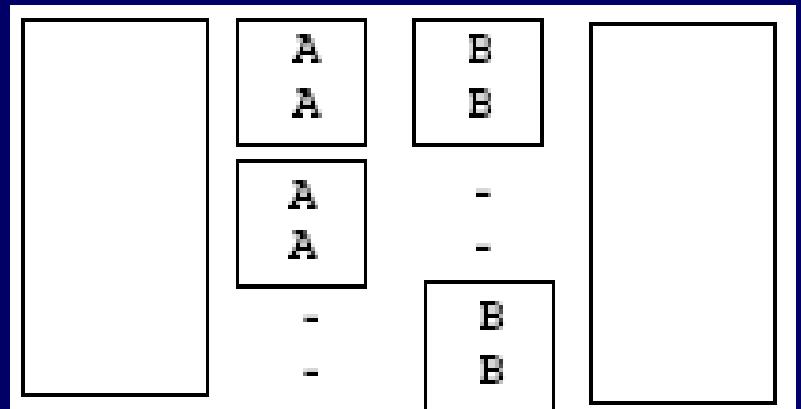
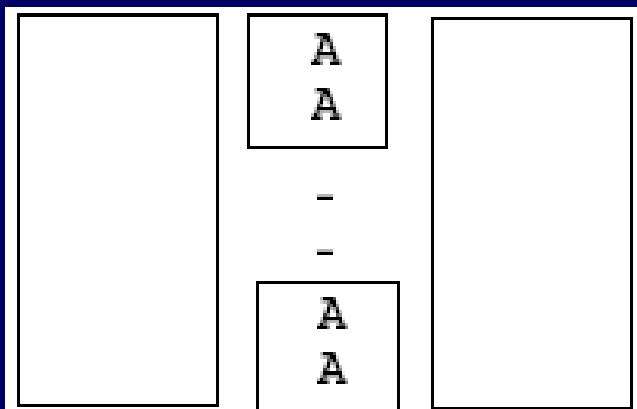
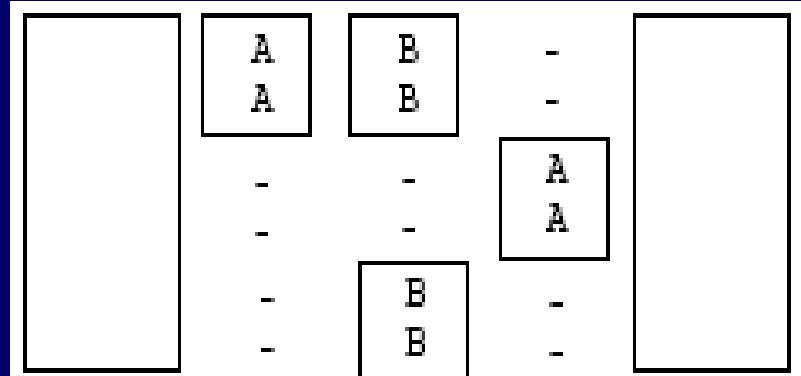
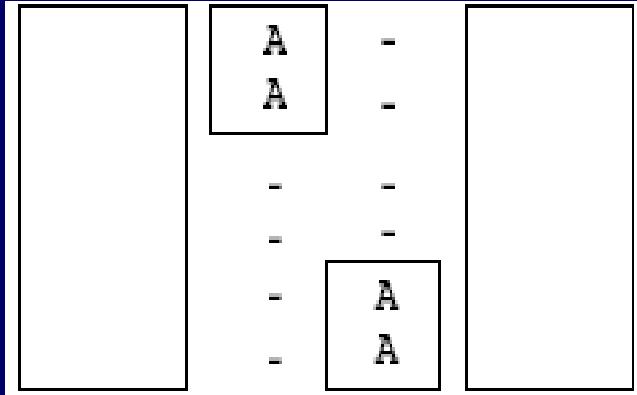
Basic idea behind this method is to try to generate many different MSAs by rearrangements that simulate gap insertion & recombination events during replication in order to generate a higher and higher score for the MSA.

SAGA, Notredame & Higgins (1996)

Hidden Markov Models (HMM): HMMs are trained to recognize a specific family of proteins; require different HMM for each protein family

# Examine alignments by eye

Some artifacts observed in the output of algorithms:



Always examine your alignment manually to see if it can be improved.

# Edit alignments manually

## Multiple sequence alignment tools

- **Viewers**
  - ClustalX, Jalview, Cinema, Sequence logos
- **Editors / annotation**
  - SeqVu, MACAW, BioEdit
- **BioEdit available at:**  
<https://bioedit.software.informer.com/7.2/>

# Work with proteins

- Twenty symbols to match as against four for DNA
- No noise resulting from the degeneracy of genetic code
- More sensitive scoring matrices
- Requires less of manual editing

# Choose genes judiciously

- When inferring phylogeny choose genes carefully
- For closely related organisms choose genes which mutate fast
- For distantly related species choose slowly mutating genes
- Compare orthologous genes between species and paralogous ones within an organism

# Summary

- Treat the output of multiple alignment programs as a first alignment
- Examine it by eye and edit it manually to improve it
- Get rid of low confidence or highly divergent regions
- Ensure you have started with a sensible evolutionary hypothesis

# Summary

How does one perform an MSA?

- By hand: too hard!
- Automated alignment: Fast, but doesn't necessarily produce the "correct" alignment

Best approach = Automated alignment with  
manual editing

# References

- David W. Mount, Bioinformatics: Sequence and Genome Analysis, CBS Publishers & distributors, New Delhi, and references therein.
- Thompson, J.D., Higgins, D.G. and Gibson. T.J. "CLUSTALW: improving the sensitivity of multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice" *Nuc. Acids Res.* 22, 4673-80 (1994).
- Carrillo, H. and Lipman, D. "The multiple sequence alignment problem in biology" *SIAM J. Appl. Math.* 48, 1073-82 (1988).

# **Local MSA**

## **Motif and Profile Analysis**

# Localised MSA

- Local MSA methods align only the most similar region in a sequence, ignoring dissimilarities.
- Local MSA is useful in identifying signatures of protein families and can often be used as tools for the prediction of protein function.
- Local MSA is also used as anchors in constructing global MSA of divergent sequences

# Local MSA

## Motivation:

- Find local regions of high similarity (motifs)
- Align based on motifs/blocks

## Approach:

- Find Motifs
  - Patterns
  - Blocks
  - Statistical Profiles (PSSM, HMM)
- Align Sequences
  - Preserve motifs as much as possible

# Localised MSA

- The results of local MSA are usually displayed in three main forms:
  - Regular expression
  - Position-Specific Scoring Matrix
  - Direct Alignment

# Pattern and Profile

- **Pattern (Motif)**
  - Deterministic syntax describing well-conserved region, an exact word or regular expression
- **Profile**
  - Probabilistic syntax describing well-conserved region
  - Score-based representations
    - Position-specific scoring matrix (PSSM)
    - Hidden Markov model (HMM)
- **Usefulness of Patterns & profiles**
  - to search for motifs/domains of biological significance that characterize a protein family

# Significance of Patterns / Motifs

- **DNA**
  - Recognition sites of restriction endonucleases
  - Codons specifying the Aas, start/stop codons
  - Intron splice sites
  - Promoter elements
  - Binding sites for regulatory proteins which activate or repress transcription
- **Proteins**
  - Active sites
  - Binding/interaction sites
  - Prediction of protein secondary structure
  - Presence of signals used to localize the protein in the cell

## When do we use motifs, patterns, blocks and profiles?

- In database search
- Sequence database search typically miss 10 - 20% of “true hits”, this area of similarity known as twilight zone
- Proportion of missed similarities are even greater when searching modular proteins
- Tools based on motifs, patterns, blocks and profiles for database searches can help as these use family information to improve sensitivity to distant family members

# Consensus Sequences

Often biologically functions are carried out by related, but not identical, sequences.

Following sequences are all known to bind MEF2 (Myocyte enhancer factor 2) transcription factor:

CTAAAAAATAA  
TTAAAAAATAA  
TTTAAAATAA  
CTATAAAATAA  
TTATAAAATAA  
CTTAAAATAG  
TTTAAAATAG

# Consensus Sequences

Simplest form of a consensus sequence is obtained by picking the most frequent base at each position in the set of aligned sequences:

TTAAAAATAA

Using IUB (International Union of Biochemistry and Molecular Biology) nucleotide codes, set of sequences could be represented by the motif :

YTWWAAATAR

where Y=[CT], W=[AT] and R=[AG].

Which one is more informative?

CTAAAAATAA  
TTAAAAATAA  
TTTAAAATAA  
CTATAAATAA  
TTATAAATAA  
CTTAAAATAG  
TTTAAAATAG

## *Summary of single-letter code recommendations*

Symbol	Meaning	Origin of designation
G	G	Guanine
A	A	Adenine
T	T	Thymine
C	C	Cytosine
R	G or A	puRine
Y	T or C	pYrimidine
M	A or C	aMino
K	G or T	Keto
S	G or C	Strong interaction (3 H bonds)
W	A or T	Weak interaction (2 H bonds)
H	A or C or T	not-G, H follows G in the alphabet
B	G or T or C	not-A, B follows A
V	G or C or A	not-T (not-U), V follows U
D	G or A or T	not-C, D follows C
N	G or A or T or C	aNy

# Regular Expression

A string that describes or matches a set of strings according to certain syntax rules.

Regular expressions extend the alphabet further.

Among the new symbols of this extended alphabet, there are symbols denoting alternative occurrence of a no. of nucleotides at a given position, and symbols denoting that a given position may not be present.

[LI]-x-E-[LIVM](2)-x(4,5)-[LIVM]-[TL]-x(5,7)-C-x(4)-[IVA]-x-[DNS]-[LIVMA]

# Regular Expression

## Syntax rules:

AA - single residue

[ ] - set of observed residues

{ } - Excluded residues

( ) - No. of repetition of any expression before it

x - Wildcard (any AA)

x(3) - Wildcard length, x(3,6) - Varying lengths

Ex: Docking of a kinase to a receptor

x(3)-[DE]-[AVLI]-x(4)-[RKH]-[VFWH]-x(3)

x	x	x	d	a	x	x	x	x	r	y	x	x	x
e	v								k	f			
									h	w			
l													
i													

Database:  
PROSITE

# Motifs

- A sequence motif is a short conserved element of a sequence alignment.
- Its function or structure may be known, or its significance may be unknown.
- One way to get functional or structural information about a sequence is to determine what motifs it contains.

**Motif databases:** Prosite, Pfam, Prints, Smart  
- contains functional sites of protein families.

**DNA motifs db:** TRANSFAC - db of eukaryotic cis-acting regulatory elements and trans-acting factors.

**Algorithms:** MOTIF, ASSET, BLOCKMAKER

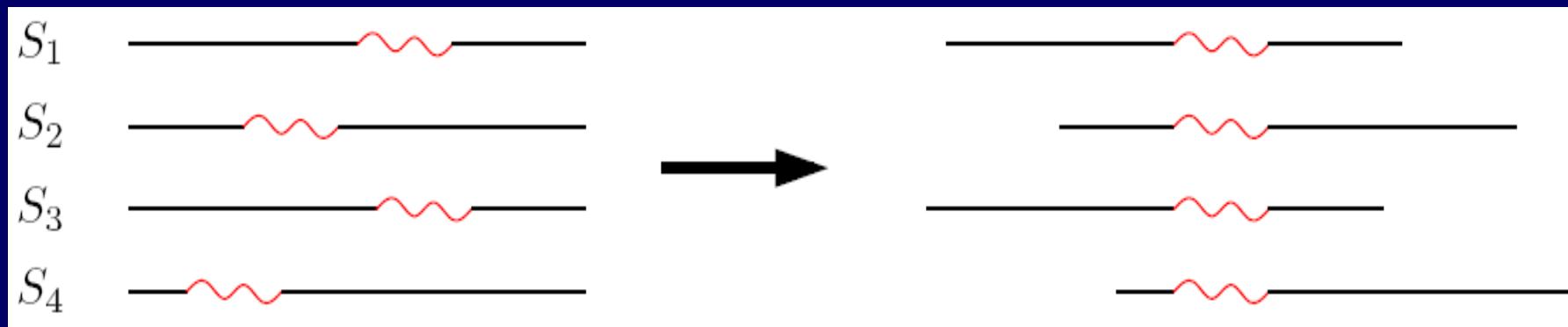
# Pattern

Two types:

- Pattern (1): a small motif
- Pattern (2): a region containing several motifs and can also contain gaps

# Motif Based Method for MSA

- These methods try to find local motifs and use them as anchors.



- Motifs are identified either using a PROSITE database or by motif search algorithms.
- Pattern-driven approach: MOTIF
- Sequence-driven approach: ASSET, BLOCKMAKER

# Profiles

- Profile - position specific scoring matrix built from multiple alignment of a set of sequences
  - matrix represents the distribution of residues at each position in the sequence alignment.
- It stores the number / probability of a specific residue at each position in the motif.
  - entries usually stored in log-odds form
- The latest approaches to building profiles are based on hidden Markov models (HMMs). These methods try to maximize the likelihood in a probabilistic model of multiple alignment.

# Profile Analysis

- Align the sequences in the family
- Extract the most conserved regions from the MSA to create a profile
- Compute the log-odds ratio of each AA / NT to appear in each position

Sequence	Position						Sequence	Position						
	1	2	3	4	...	$l$		1	2	3	4	5	...	$l$
1	$a_{11}$	$a_{12}$	$a_{13}$	...	...	$a_{1l}$	L	$P_{L1}$	$P_{L2}$	$P_{L3}$	...	...	...	$P_{Ll}$
2	$a_{21}$	$a_{22}$	$a_{23}$	...	...	$a_{2l}$	V	$P_{V1}$	$P_{V2}$	$P_{V3}$	...	...	...	$P_{Vl}$
3	$a_{31}$						F	$P_{F1}$						
.							.							
.							.							
N	$a_{N1}$	$a_{N2}$	$a_{N3}$	...	...	$a_{Nl}$	.							

**MSA**

$\text{Score}_{ij} = \log(P_{ij})$ ,  
 $P_{ij} = f_{ij} / b_i$

No. of columns is determined by the letters in the alphabet (20 for AA & 4 for NT)

# PSSM

Information about the relative occurrence of each symbol at each position is lost in the motifs, e.g., A/G on last column

This information can be explicitly captured by providing the relative frequency or probability of each symbol at each position along the alignment.

These probabilities are - Position Weight Matrices, or Position Specific Scoring Matrices (PWMS/PSSMs)

CTAAAAAATAA  
TTAAAAAATAA  
TTTAAAATAA  
CTATAAAATAA  
TTATAAAATAA  
CTTAAAATAG  
TTTAAAATAG

# Profile Analysis

Step-1: Align members of family:

LEVK

LDIR

LEIK

LDVE

Step 2: Compute  $f_{i,j}$  = % of a.a.  $i$  in column  $j$ ;  $b_i$  = % of a.a.  $i$  with “background” freq.;

$$\Rightarrow p_{i,j} = f_{ij} / b_i \quad b_i = ?$$

e.g.,  $p_{E,2} = (2/4) / (1/20) = 10$

assuming uniform background frequency

$\Rightarrow 20 \times l$  array of propensities of a.a.  $i$  in col.  $j$

# Profile Analysis

Step-3: Now to score an **I** long sequence,  
say LEVE, compute

$$p_{L,1} \times p_{E,2} \times p_{V,3} \times p_{E,4}$$

- If this is greater than some cutoff, then we say it is a “member of the family”, otherwise not.
- In practice, compute

$$\log(p_{L,1} \times p_{E,2} \times p_{V,3} \times p_{E,4}) \quad \text{Why?}$$

$$= \log(p_{L,1}) + \log(p_{E,2}) + \log(p_{V,3}) + \log(p_{E,4})$$

$$\Rightarrow \text{score}_{i,j} = \log(p_{i,j})$$

# Profile Analysis

To use a profile to score a new sequence, slide a window of length  $l$  over the sequence

e.g., for sequence LEVEER, find if it contains a motif

Score each  $l$ -long window:

LEVE, EVEE, VEER

Score of LEVE =  $score_{L,1} + score_{E,2} + score_{V,3} + score_{E,4}$

Score of EVEE =  $score_{E,1} + score_{V,2} + score_{E,3} + score_{E,4}$

Score of VEER =  $score_{V,1} + score_{E,2} + score_{E,3} + score_{R,4}$

If any of these is larger than cutoff, we have found the motif & its position in sequence

# Profile of a set of heat shock proteins

Cons	A	B	C	D	E	F	G	H	I	K	L	M	N	P	Q	R	S	T	V	W	Y	Z	Gap	Len	
I	8	3	-2	5	4	5	5	-4	<u>24</u>	0	15	13	1	1	1	-7	2	22	21	-18	-6	?	4	100	100
T	13	19	-5	24	18	-18	19	7	1	7	-7	-4	14	11	10	-1	9	<u>29</u>	3	-28	-14	15	100	100	
L	5	5	-5	3	4	13	4	2	8	-4	<u>14</u>	12	8	-5	0	-10	0	10	10	-1	5	2	<u>22</u>	22	
S	17	14	17	13	10	-12	29	-5	-5	6	-14	-9	12	10	0	-2	<u>34</u>	19	1	-8	-15	4	100	100	
T	15	3	22	0	-1	-5	12	-2	7	-3	-8	-6	5	7	-8	-7	16	<u>29</u>	9	-22	6	-4	100	100	
T	8	-1	12	-2	0	5	6	-4	19	-4	8	5	-1	2	-8	-8	7	<u>22</u>	19	-15	4	-3	100	100	
C	17	0	<u>24</u>	-1	-3	11	8	-1	7	-10	1	-2	1	-3	-8	-14	8	5	9	-5	14	-7	100	100	
V	11	0	18	-1	-2	2	14	-10	26	-4	9	7	-3	7	-7	-7	21	10	<u>31</u>	-19	-5	-5	100	100	
C	10	-8	<u>15</u>	-11	-11	6	8	-7	11	-10	4	3	-7	0	-11	-4	11	5	15	-22	14	-11	100	100	
V	7	7	-3	8	8	-3	11	1	20	-1	14	10	4	2	8	-5	0	5	<u>26</u>	-24	-6	8	100	100	

Consensus AA at each position in the profile

# Profile Analysis

- Disadvantage: Contains no more information than in the MSA
- Missing AA/NT could be due to small sample size
- Several similar sequences produce a bias in the profile towards themselves - this can be circumvented by defining sequence weighting
- When constructing a PSSM, larger the no. of sequences in the alignment, reliable will be the PSSM in representing the motif.
- Profile making/searching programs:  
ProfileMake/ProfileSearch (GCG),  
prophecy/prophet/profit (EMBOSS)

# BLOCKS analysis

- Similar to profiles using conserved regions of MSA
- Different because regions with gaps not considered
- Blocks may also be formed using pattern searching tools to look for similar short sequences in larger sequences
- These blocks may be whole domains, short seq. motifs, key parts of enzyme active sites, etc.
- Programs:

BLOCKS (<http://blocks.fhcrc.org>),  
eMOTIFs (<http://dna.stanford.edu/emotif>)

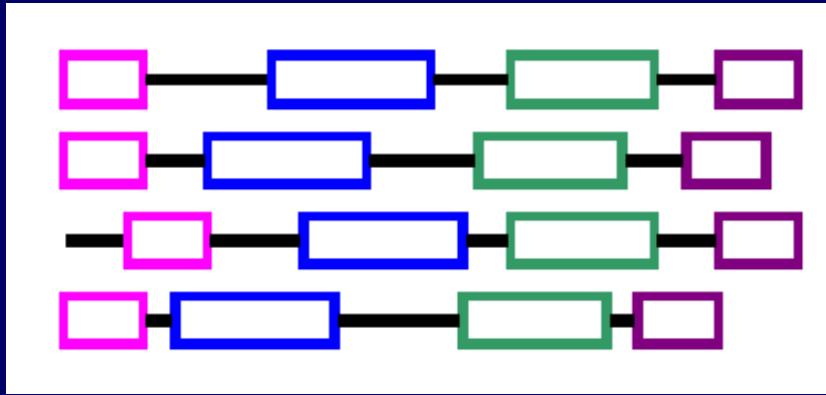
BLOCKS represent a conserved region in the msa that is lacking in gaps - i.e. no insertions/deletions

# BLOCKS analysis

- A single protein contains numerous such BLOCKS separated by stretches of intervening sequences that can differ in length and composition.
- These methods do not use substitution matrices to score matches.
- BLOCKS are typically anywhere from 3-60 a.a. long, based on exact a.a. matches, and have the same spacing in at least some of the input sequences

Seq1	GVDVLVATPG	RLLLDLEHQNA .. VKLDQV	EILVLDEADR
Seq2	GPDALVSTPG	RYLTLEHRNV .. LKPDIV	TIRVLDEADR
Seq3	ADEVIVSTPG	RLWDLHHQNA .. VQLSQD	ELLDLDEADK
.....			
Seqn	GCDKLNATPG	RLMDLKHQGA .. VKLLFV	SILVMDEADR

# BLOCK-based Global MSA

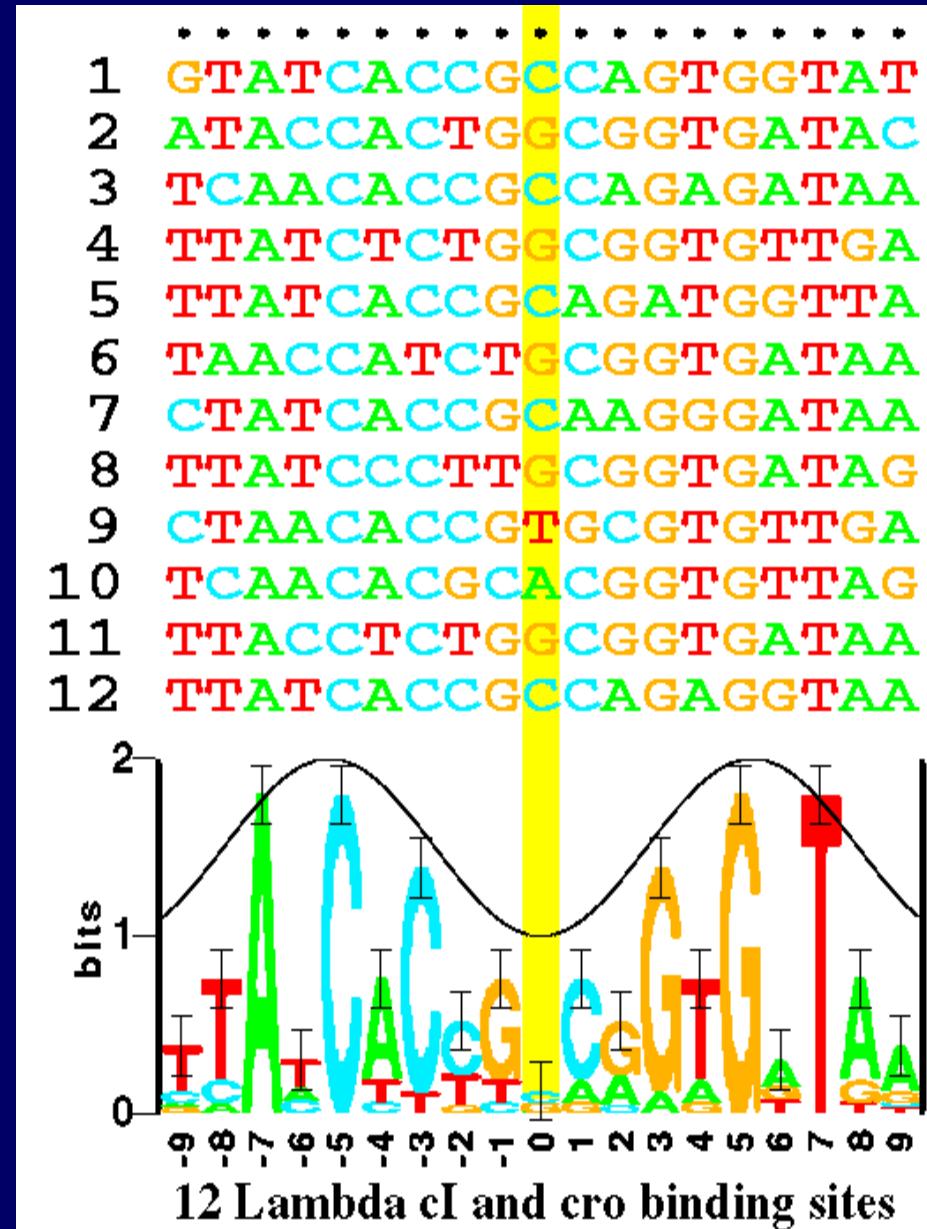


- Search for ungapped conserved regions (blocks)
  - Pairwise alignment → weighted diagonals (DIALIGN)
  - Suffix trees → common subsequences
  - Dot matrix plots → diagonals
- Use blocks as anchors to align segments
  - Find consistent set of uniform / near-uniform blocks
  - Align sequences to produce maximum SP weight

Ex: DIALIGN

# Viewing MSA – Sequence Logos

- Graphical representation of conserved region
  - Good for short sequences
  - More information than just consensus sequence
- Total height of a stack of letters (measured in bits)
  - Degree of sequence conservation
- Relative letter heights
  - Frequencies of bases or residues at each position



# Protein Secondary / Pattern Databases

- **PROSITE** - ExPasy (SIB), HGMP (UK)
- **PRINTS** - EBI, Manchester (UK), NCBI (USA)
- **SMART** - EMBL, Heidelberg (Germany)
- **BLOCKS** - FHCRC, Seattle (USA)
- **Pfam** - Sanger Institute (UK)
- **ProDom** - INRA & CNRS, France

# Protein Secondary / Pattern Databases

- **TIGRFAMS** - TIGR, USA
- **PIR Superfamilies** - Georgetown Univ., USA
- **DOMO** - (France)
- **ProtoMap** - Stanford, USA
- **SYSTERS** - Germany
- **ProClass** - Georgetown University (USA)
- **InterPro** - (France)
- **MetaFam** - University of Minnesota (USA)

## Importance of protein family databases:

Allows for

- better detection of family members,
- identification of conserved residues,
- distinguishing orthologues (which are related by descent) from paralogues (which derive from gene duplication)
- structure modeling.

# InterPro

InterPro - is an integrated documentation resource for protein families, domains and functional sites.

InterPro combines a number of databases - member databases - that use different methodologies and a varying degree of biological information on well-characterized proteins to derive protein signatures

By uniting the member databases, InterPro capitalizes on their individual strengths, producing a powerful integrated diagnostic tool

<http://www.ebi.ac.uk/interpro/>

# InterPro

Member databases use two main approaches:

➤ **Sequence-motif methods**

- PROSITE - regular expressions and profiles;
- Gene3D, PANTHER, PIRSF, Pfam, SMART, SUPERFAMILY, and TIGRFAMs - HMM profiles
- PRINTS - provider of fingerprints (groups of aligned, un-weighted motifs)

➤ **Sequence-cluster methods**

- ProDom uses PSI-BLAST to find homologous domains, that are clustered in the same ProDom entry
- **Cross-referencing between BLOCKS database**

# InterPro

While all the resources share a common interest in protein sequence classification, the focus of each database is different. For e.g.,

- Pfam focuses on divergent domains,
- ProDom to facilitate domain identification,
- PROSITE on functional sites,
- PRINTS focuses on families, specializing in hierarchical definitions from superfamily down to sub-family levels in order to describe specific functions.
- Blocks provide ungapped multiple alignments for protein families.

# InterPro

Diagnostically, these resources have different areas of optimum application owing to **different strengths & weaknesses of their underlying analysis methods**, e.g.,

- Regular expressions are likely to be unreliable in the identification of members of highly divergent superfamilies (where profiles and HMMs excel)
- Fingerprints perform relatively poorly in the diagnosis of very short motifs (where regular expressions do well)
- Profiles & HMMs are less likely to give specific sub-family diagnosis (where fingerprints excel)

# **Modeling Molecular Evolution**

**What are the basic processes of molecular evolution?**

**What is one looking for while comparing sequences?**

**Basic mutational processes are - substitution and insertions and deletions**

**- introduced at the molecular level through random changes as the molecules are copied into new generations**

**It potentially affects the function of the gene, which can either be beneficial, or lead to reduction in functionality & adaptability of the protein**

**Natural selection comes into play – allowing mutations that are either evolutionarily advantageous or, occur in non-functional regions of the sequence**

**- it has the effect of screening the mutations; some changes are seen more often than others**

Let's try to develop mathematical models of molecular evolution using the **language of probability** to describe random mutations

- the concept of **phylogenetic distance** as a measure of sequence similarity will emerge from these probabilistic models.

When base substitutions occur in evolution, the probability of a particular base appearing at a site in the descendent sequence **might** depend on the ancestral base.

e.g., if the ancestral base is T, then

- probability of seeing a T in the descendant sequence is higher – why?
- probability of seeing a C in the descendant sequence is lower than T – why?
- probability of seeing an A or G in the descendant sequence is lowest – why?

To formalize this we shall use the concept of **conditional probability**

When base substitutions occur in evolution, the probability of a particular base appearing at a site in the descendent sequence **might** depend on the ancestral base.

e.g., if the ancestral base is T, then

- probability of seeing a T in the descendant sequence is higher – **No change**
- probability of seeing a C in the descendant sequence is lower than T – **Transition**
- probability of seeing an A or G in the descendant sequence is lowest – **Transversion**

A transition is less likely than a “no change” and a transversion is even lower.

# Conditional Probability

If E and F are two events, then conditional probability of F given E is defined by

$$p(F | E) = \frac{p(F \cap E)}{p(E)}$$

The concept of conditional probability also clarifies the notion of independence of events.

Events E and F are independent if knowledge that one has occurred gives no information as to whether the other occurred, i.e.,

$$p(F|E) = p(F) \text{ and } P(E|F) = p(E)$$

$\Rightarrow P(F \cap E) = p(F)p(E)$ , if events E & F are independent

# Conditional Probability

Ex: Taking into account the likelihood of transitions and transversions, which of the following is likely to be the smallest? Which is likely to be the largest?

- (i)  $p(S_1 = C | S_0 = C)$  “no change”,
- (ii)  $p(S_1 = T | S_0 = C)$ , “transition”,
- (iii)  $p(S_1 = A | S_0 = C)$  “transversion”, and
- (iv)  $p(S_1 = G | S_0 = C)$  “transversion”.

What is the sum of the above four probabilities?

**Other mutations observed include:**

- **Deletion of a base or consecutive bases,**
- **Insertion of a base or consecutive bases,**
- **Inversion (reversal) of a section of the sequence**

**These mutations are seen more rarely in natural populations.**

**- not surprising, since these mutations have a dramatic effect on the protein.**

**Ignore such possibilities to make our modeling task both clearer & mathematically tractable.**

Focusing solely on base substitutions, a basic problem is how to deduce the amount of mutation during evolution:

S0: ACCTGCGCTA

S1: ACGTGCACTA

S2: ACGTGC~~G~~CTA

If G → A → C at the 7<sup>th</sup> position?

Comparing S0 & S2 - 1/10 mutations per site

Comparing S0, S1 & S2 - 3/10 mutations per site from S0 to S2

– a simple ratio of mutations per site obtained from comparing 1<sup>st</sup> & 3<sup>rd</sup> sequences gives a lower estimate of the mutation that actually occurred.

Assuming that mutations are rare,

- ignore the probability of hidden mutations having occurred ( $G \rightarrow A \rightarrow G$ )

we can reconstruct a mathematical model for the no. of mutations that are likely to have occurred from those observed in comparing only the initial and final DNA sequences.

# **Matrix Models of Base Substitution**

# Matrix Models of Base Substitution

Model ancestral sequence probabilistically:

Assume each site in the sequence is one of the 4 bases chosen randomly with probabilities  $p_A, p_G, p_C, p_T$ ,

- these probabilities describe the ancestral base distribution in a vector as

$$\mathbf{p}_0 = (p_A, p_G, p_C, p_T), \quad p_A + p_G + p_C + p_T = ?$$

Model the mutation process over one-time step, assuming that only base substitutions can occur.

# Matrix Models of Base Substitution

Model the mutation process over one-time step, assuming that only base substitutions can occur.

⇒ 16 conditional probabilities of observing base substitutions,  $p(S_1 = i | S_0 = j)$  for  $i, j = A, G, C, \text{ and } T$ :

$$M = \begin{pmatrix} & \text{Descendent base} \\ \begin{matrix} p_{A|A} & p_{A|G} & p_{A|C} & p_{A|T} \\ p_{G|A} & p_{G|G} & p_{G|C} & p_{G|T} \\ p_{C|A} & p_{C|G} & p_{C|C} & p_{C|T} \\ p_{T|A} & p_{T|G} & p_{T|C} & p_{T|T} \end{matrix} & \\ \text{Ancestral base} \end{pmatrix}$$

Assuming only base substitutions – is it reasonable for coding regions of DNA?

Column sum ?  
Row sum?

# Example

For the 40-base ancestral sequence,  $S_0$ :

ACTTGTCTGGATGATCAGCGGTCCATGCACCTGACAACGGT

and its descendent sequence,  $S_1$ :

ACATGTTGCTTGACGACAGGTCCATGCGCCTGAGAACGGC

$$p_0 = (p_A, p_G, p_C, p_T) = (.225, .275, .275, .225)$$

$$M = \begin{pmatrix} .778 & 0 & .091 & .111 \\ .111 & .818 & .182 & 0 \\ 0 & .182 & .636 & .222 \\ .111 & 0 & .091 & .667 \end{pmatrix}$$

No. of A's in  $S_0$  = 9, probability of A in  $S_0$  = 9/40 = 0.225, ...

First entry in  $M$  is  $p_{A|A} = 7/9 = 0.778$ , ...

# Matrix Models of Base Substitution

Multiplying:

$$\mathbf{M}\mathbf{p}_0 = \begin{pmatrix} \mathbf{p}_{A|A} & \mathbf{p}_{A|G} & \mathbf{p}_{A|C} & \mathbf{p}_{A|T} \\ \mathbf{p}_{G|A} & \mathbf{p}_{G|G} & \mathbf{p}_{G|C} & \mathbf{p}_{G|T} \\ \mathbf{p}_{C|A} & \mathbf{p}_{C|G} & \mathbf{p}_{C|C} & \mathbf{p}_{C|T} \\ \mathbf{p}_{T|A} & \mathbf{p}_{T|G} & \mathbf{p}_{T|C} & \mathbf{p}_{T|T} \end{pmatrix} \begin{pmatrix} \mathbf{p}_A \\ \mathbf{p}_G \\ \mathbf{p}_C \\ \mathbf{p}_T \end{pmatrix}$$
$$= \begin{pmatrix} \mathbf{p}_{A|A}\mathbf{p}_A + \mathbf{p}_{A|G}\mathbf{p}_G + \mathbf{p}_{A|C}\mathbf{p}_C + \mathbf{p}_{A|T}\mathbf{p}_T \\ \mathbf{p}_{G|A}\mathbf{p}_A + \mathbf{p}_{G|G}\mathbf{p}_G + \mathbf{p}_{G|C}\mathbf{p}_C + \mathbf{p}_{G|T}\mathbf{p}_T \\ \mathbf{p}_{C|A}\mathbf{p}_A + \mathbf{p}_{C|G}\mathbf{p}_G + \mathbf{p}_{C|C}\mathbf{p}_C + \mathbf{p}_{C|T}\mathbf{p}_T \\ \mathbf{p}_{T|A}\mathbf{p}_A + \mathbf{p}_{T|G}\mathbf{p}_G + \mathbf{p}_{T|C}\mathbf{p}_C + \mathbf{p}_{T|T}\mathbf{p}_T \end{pmatrix} = \begin{pmatrix} p(S_1 = A) \\ p(S_1 = G) \\ p(S_1 = C) \\ p(S_1 = T) \end{pmatrix}$$
$$\Rightarrow \mathbf{M} \mathbf{p}_0 = \mathbf{p}_1$$

$\mathbf{p}_1$  - the vector of base probabilities in sequence  $S_1$

Note:  $P(F \cap E) = p(F|E)p(E)$

# Matrix Models of Base Substitution

$M$  - is a transition matrix, gives how probabilities of each base in ancestral seq  $S_0$  are transformed into probabilities of each base in the descendent seq  $S_1$ , one-time step later.

What would be the meaning of  $Mp_1$ ?

$$p_1 = Mp_0 = \begin{pmatrix} .225 \\ .275 \\ .300 \\ .200 \end{pmatrix}, \quad p_2 = Mp_1 = \begin{pmatrix} .222 \\ .274 \\ .320 \\ .183 \end{pmatrix}$$

What is the sum of the entries in  $p_1$ ? In  $p_2$ ?

Why must this be the case? Why use same  $M$ ?

# Matrix Models of Base Substitution

To make sense biologically, we must assume that the probabilistic mutation process over the first time step is identical to that over the next time step.

Using the same transition matrix  $M$  of conditional probabilities means each type of base substitution has the same likelihood of occurring as it did before.

- reasonable assumption for small time intervals.

# Matrix Models of Base Substitution

Furthermore, what happens during the second step depends only on:

- what the base was at time  $t = 1$  (the information in  $p_1$ ), and
- the conditional probabilities (the information in  $M$ )

Whether that site experienced a substitution during the previous time step is irrelevant.

This is an example of a **Markov model**.

# Markov Models

- In a Markov model, a system is described in one of  $n$  different states and may switch from one state to another with time.
- In our DNA substitution model, the system is a **site** in a DNA sequence, which is initially in one of the 4 states (A, G, C, or T) according to the base that occupies it.
- Initial probabilities that the system is in one of the states is given by a vector,  $p_0$ .
- Conditional probabilities of the switch from every state to every other state over one-time step is given by a  $4 \times 4$  transition matrix,  $M$ .

# Markov Models

An important assumption is made in any Markov model:

What happens to the system over a given time step depends only on the state the system is in at the start of that step and the transition probabilities.

- there is “no memory” of what changes might have occurred during earlier time steps, i.e., conditional probabilities are independent of the past history.

Can a Markov model be used to identify spatial patterns in a DNA sequence, e.g. CpG islands, protein-coding regions?

# Markov Models

**Q. For a DNA substitution model, is it reasonable to assume this independence?**

In our DNA model we also assumed that each site in the sequence behaves **identically** and **independently** of every other site to find various probabilities from sequence data, by considering each site as an independent trial of the same probabilistic process.

**Q. How reasonable is this assumption?**

# Markov Models

This assumption may not be a very reasonable one for gene sequences:

- Genetic code allows for many changes in the **third site of each codon to have no effect on the gene product, as a consequence, substitutions in the third sites might be more likely than in the first two, violating the assumption that each site behaves identically**
- Since genes lead to the production of proteins, the likelihood of change at one site may well be tied to changes at another, violating the assumption of **independence**.

# Markov Models

**Can we find ways to go around these assumptions?**

- **allowing for different conditional probabilities for various sites.**
- **be careful to take assumptions into account when using the tools on real data**
  - **for instance, we might ignore the third base of each codon in estimating information from our data, so that it is more reasonable to treat sites as independent and following identical processes.**

# Markov Models

**Markov matrix has all entries  $\geq 0$  and its columns sum to 1.**

**Theorem-1:** A Markov matrix always has  $\lambda_1 = 1$  as its **largest eigenvalue** and has all eigenvalues satisfying  $|\lambda| \leq 1$ . Eigenvector corresponding to  $\lambda_1$  has all nonnegative entries.

**There will be only one eigenvector associated with  $\lambda_1 = 1$ .**

**Theorem-2:** A Markov matrix, all of whose entries are positive (i.e., nonzero), always has 1 as a strictly dominant eigenvalue.

# Markov Models

**We will now discuss a few special Markov models of base substitutions:**

- **Jukes-Cantor Model**
- **Kimura Models**

# Jukes-Cantor model

- the simplest Markov model of base substitution.

Additional assumptions made to basic Markov model:

First, all bases occur with equal probability in the ancestral sequence, i.e.,

$$p_0 = (1/4, 1/4, 1/4, 1/4)$$

Second, all the 16 conditional probabilities of base substitutions are **same**, i.e., all possible substitutions are equally likely:

$$A \leftrightarrow G,$$

$$C \leftrightarrow T,$$

$$A \leftrightarrow C,$$

$$A \leftrightarrow T,$$

$$C \leftrightarrow G,$$

$$T \leftrightarrow G$$

i.e., it assumes transitions & transversions occur at the same rate.

# Jukes-Cantor model

If we define  $\alpha/3$  as the conditional probability of a base substitution of any type:

$$p(S_1 = i | S_0 = j) = \alpha/3, \quad \text{for all } i, j$$

i.e., the 12 off-diagonal entries of the matrix  $M$  will be  $\alpha/3$ .

Since the entries in any column of  $M$  add to 1, what would be the diagonal entries?

# Jukes-Cantor model

Transition matrix for Jukes-Cantor model:

$$M = \begin{pmatrix} 1-\alpha & \alpha/3 & \alpha/3 & \alpha/3 \\ \alpha/3 & 1-\alpha & \alpha/3 & \alpha/3 \\ \alpha/3 & \alpha/3 & 1-\alpha & \alpha/3 \\ \alpha/3 & \alpha/3 & \alpha/3 & 1-\alpha \end{pmatrix}$$

Value of  $\alpha$  depends on the **time step** we use and **features** of the particular DNA sequence being modeled.

# Jukes-Cantor model

Although  $\alpha$  is a probability, we can interpret it as a rate:

- it is the rate at which observable base substitutions occur over one time step and is measured in units of

$$\alpha = (\text{substitutions per site}) / \text{time step}$$

Mutational rates  $\alpha$  for DNA in real organisms is not easily found.

# Jukes-Cantor model

Estimates of  $\alpha$ :

- **$1.1 \times 10^{-9}$  mutations per site per year for certain sections of chloroplast DNA of maize & barley**
- **$10^{-8}$  mutations per site per year for mtDNA in mammals**
- **$0.01$  mutations per site per year for influenza A virus**
- **Rate of mutation is generally a bit lower in coding regions than in noncoding DNA**

**After 1 million years, compute the amount of mutation in the descendant sequence if  $\alpha = 10^{-8}$  & length of sequence is 1000 bases.**

# Jukes-Cantor model

In the development of our model, we shall treat  $\alpha$  as an unknown **constant**.

In reality, the mutation rate **may not be constant**; it may change with time, or with location within the DNA.

For shorter periods of time and for DNA serving a fixed purpose, the assumption of a constant mutational rate is reasonable.

When mutation rates are constant, there is said to be a **molecular clock** operating.

# Jukes-Cantor model

**Ex-1: For the Jukes-Cantor model, in what proportion of the sites will each base appear after one time-step?**

**Ex-2: What proportion of the sites will have a base A in the ancestral sequence and a T in the descendent one time-step later:**

$$p(S_0 = A \text{ and } S_1 = T) ?$$

**Ex-3: What is the probability that a base A in the ancestral sequence will have mutated to become a base T in the descendent sequence 100 time-steps later:  $p(S_{100} = T | S_0 = A)$ ?**

# Jukes-Cantor model

Ex-1: For the Jukes-Cantor model, in what proportion of the sites will each base appear after one time-step?

$$\mathbf{p}_1 = M \mathbf{p}_0 = \begin{pmatrix} 1-\alpha & \alpha/3 & \alpha/3 & \alpha/3 \\ \alpha/3 & 1-\alpha & \alpha/3 & \alpha/3 \\ \alpha/3 & \alpha/3 & 1-\alpha & \alpha/3 \\ \alpha/3 & \alpha/3 & \alpha/3 & 1-\alpha \end{pmatrix} \begin{pmatrix} 1/4 \\ 1/4 \\ 1/4 \\ 1/4 \end{pmatrix} = \begin{pmatrix} 1/4 \\ 1/4 \\ 1/4 \\ 1/4 \end{pmatrix}$$

**Base composition of the sequence does not change under Jukes-Cantor model**

$(1/4, 1/4, 1/4, 1/4)$  is an equilibrium base distribution for sequences under the Jukes-Cantor model

# Jukes-Cantor model

**Ex-2: What proportion of sites will have a base A in ancestral sequence and a T in descendent one time-step later,  $p(S_0 = A \text{ and } S_1 = T)$ ?**

$$\begin{aligned} p(S_0 = A \text{ and } S_1 = T) &= p(S_1 = T \mid S_0 = A) p(S_0 = A) \\ &= (\alpha/3)(1/4) = \alpha / 12 \end{aligned}$$

# Jukes-Cantor model

**Ex-3: What is the probability that a base A in the ancestral sequence will have mutated to become a base T in descendent sequence 100 time-steps later, i.e., compute the probability  $p(S_{100} = T | S_0 = A)$ ?**

$$p_{100} = M^{100} p_0$$

**What is the (4,1) entry of  $M^{100}$ ?**

# Jukes-Cantor model

Generalizing to any  $t$ , let's find all entries of  $M^t$  – using the eigenvectors approach.

**Why do we need to compute eigenvalues and eigenvectors?**

How do we compute the eigenvectors & eigenvalues of a matrix?

# Jukes-Cantor model

**Theorem:** If  $A$  is an  $n \times n$  matrix,  $v$  a non-zero vector, and  $\lambda$  a scalar such that  $Av = \lambda v$ , then  $v$  is an eigenvector of  $A$  with eigenvalue  $\lambda$ .

Equilibrium base distribution is one eigenvector with eigenvalue  $\lambda = 1$ , there are 3 more that can be found by trial and error or a long computation

$$p_1 = M p_0 = \begin{pmatrix} 1-a & a/3 & a/3 & a/3 \\ a/3 & 1-a & a/3 & a/3 \\ a/3 & a/3 & 1-a & a/3 \\ a/3 & a/3 & a/3 & 1-a \end{pmatrix} \begin{pmatrix} 1/4 \\ 1/4 \\ 1/4 \\ 1/4 \end{pmatrix} = \begin{pmatrix} 1/4 \\ 1/4 \\ 1/4 \\ 1/4 \end{pmatrix}$$

# Jukes-Cantor model

Eigenvectors & eigenvalues of Jukes-Cantor matrix are:

$$\mathbf{v}_1 = (1, 1, 1, 1)$$

$$\lambda_1 = 1$$

$$\mathbf{v}_2 = (1, 1, -1, -1)$$

$$\lambda_2 = 1 - 4/3 \alpha$$

$$\mathbf{v}_3 = (1, -1, 1, -1)$$

$$\lambda_3 = 1 - 4/3 \alpha$$

$$\mathbf{v}_4 = (1, -1, -1, 1)$$

$$\lambda_4 = 1 - 4/3 \alpha$$

Check by multiplying  $M\mathbf{v}_i$  for each  $i$ .

# Jukes-Cantor model

**Theorem:** If  $v$  is an eigenvector of  $A$  with eigenvalue  $\lambda$ , then for any scalar  $c$ ,  $cv$  is also an eigenvector of  $A$  with the same eigenvalue  $\lambda$ , i.e.,

If  $Av = \lambda v$ , then

$$A(cv) = cAv = c \lambda v = \lambda(cv)$$

**Theorem:** Let  $A$  be a  $n \times n$  matrix with  $n$  eigenvectors  $v_1, v_2, \dots, v_n$ , whose corresponding eigenvalues are  $\lambda_1, \lambda_2, \dots, \lambda_n$ .

Let these  $n$  eigenvectors form columns of the matrix  $S$ .

# Jukes-Cantor model

If  $S$  has an **inverse**, then any vector can be written as a sum of column eigenvectors. Expressing initial eigenvector as

$$x_0 = c_1 v_1 + c_2 v_2 + \dots + c_n v_n$$

$$\text{Then, } x_1 = A x_0 = A(c_1 v_1 + c_2 v_2 + \dots + c_n v_n)$$

$$= c_1 A v_1 + c_2 A v_2 + \dots + c_n A v_n$$

$$= c_1 \lambda_1 v_1 + c_2 \lambda_2 v_2 + \dots + c_n \lambda_n v_n$$

$$x_2 = Ax_1 = A(c_1 \lambda_1 v_1 + c_2 \lambda_2 v_2 + \dots + c_n \lambda_n v_n)$$

$$= c_1 \lambda_1 A v_1 + c_2 \lambda_2 A v_2 + \dots + c_n \lambda_n A v_n$$

$$= c_1 (\lambda_1)^2 v_1 + c_2 (\lambda_2)^2 v_2 + \dots + c_n (\lambda_n)^2 v_n$$

And so on, we obtain

$$x_t = c_1 (\lambda_1)^t v_1 + c_2 (\lambda_2)^t v_2 + \dots + c_n (\lambda_n)^t v_n$$

# Jukes-Cantor model

To find all entries of  $M^t$ : Let's first focus on the first column of  $M^t$ , which can be isolated by taking the product

$$M^t \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \text{first column of } M^t$$

Expressing  $(1,0,0,0)$  in terms of the eigenvectors:

$$(1,0,0,0) = \frac{1}{4}v_1 + \frac{1}{4}v_2 + \frac{1}{4}v_3 + \frac{1}{4}v_4$$

**Theorem: If  $\lambda$  is an eigenvalue for an  $n \times n$  matrix  $A$ , then it satisfies the  $n^{\text{th}}$  degree polynomial equation  $\det(A - \lambda I) = 0$ . Thus, there are at most  $n$  eigenvalues for  $A$ .**

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} ax + by \\ cx + dy \end{pmatrix} = \begin{pmatrix} a \\ c \end{pmatrix} \quad \text{for } x = 1, y = 0$$

**The first column of  $A$**

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} a^2 + bc & ab + bd \\ ca + dc & cb + d^2 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} a^2 + bc \\ ca + dc \end{pmatrix} \quad \text{for } x = 1, y = 0$$

**The first column of  $A^2$**

**$x = ?, y = ?$  to obtain 2<sup>nd</sup> col of  $A$**

# Jukes-Cantor model

$$\begin{aligned}
 \mathbf{M}^t \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} &= \frac{1}{4} \mathbf{M}^t \mathbf{v}_1 + \frac{1}{4} \mathbf{M}^t \mathbf{v}_2 + \frac{1}{4} \mathbf{M}^t \mathbf{v}_3 + \frac{1}{4} \mathbf{M}^t \mathbf{v}_4 \\
 &= \frac{1}{4} \mathbf{1}^t \mathbf{v}_1 + \frac{1}{4} (1 - 4/3 \alpha)^t \mathbf{v}_2 + \frac{1}{4} (1 - 4/3 \alpha)^t \mathbf{v}_3 + \frac{1}{4} (1 - 4/3 \alpha)^t \mathbf{v}_4
 \end{aligned}$$

Substituting in the vectors  $\mathbf{v}_i$ ,

$$\begin{array}{ll}
 \mathbf{v1} = (1, 1, 1, 1) & \lambda_1 = 1 \\
 \mathbf{v2} = (1, 1, -1, -1) & \lambda_2 = 1 - 4/3 \alpha \\
 \mathbf{v3} = (1, -1, 1, -1) & \lambda_3 = 1 - 4/3 \alpha \\
 \mathbf{v4} = (1, -1, -1, 1) & \lambda_4 = 1 - 4/3 \alpha
 \end{array}$$

$$\mathbf{M}^t \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} \frac{1}{4} + \frac{3}{4} \left(1 - \frac{4}{3} \alpha\right)^t \\ \frac{1}{4} - \frac{1}{4} \left(1 - \frac{4}{3} \alpha\right)^t \\ \frac{1}{4} - \frac{1}{4} \left(1 - \frac{4}{3} \alpha\right)^t \\ \frac{1}{4} - \frac{1}{4} \left(1 - \frac{4}{3} \alpha\right)^t \end{pmatrix}$$

# Jukes-Cantor model

Other columns of  $M^t$  are found similarly:

$$M^t = \begin{pmatrix} \frac{1}{4} + \frac{3}{4} \left(1 - \frac{4}{3}\alpha\right)^t & \frac{1}{4} - \frac{1}{4} \left(1 - \frac{4}{3}\alpha\right)^t & \frac{1}{4} - \frac{1}{4} \left(1 - \frac{4}{3}\alpha\right)^t & \frac{1}{4} - \frac{1}{4} \left(1 - \frac{4}{3}\alpha\right)^t \\ \frac{1}{4} - \frac{1}{4} \left(1 - \frac{4}{3}\alpha\right)^t & \frac{1}{4} + \frac{3}{4} \left(1 - \frac{4}{3}\alpha\right)^t & \frac{1}{4} - \frac{1}{4} \left(1 - \frac{4}{3}\alpha\right)^t & \frac{1}{4} - \frac{1}{4} \left(1 - \frac{4}{3}\alpha\right)^t \\ \frac{1}{4} - \frac{1}{4} \left(1 - \frac{4}{3}\alpha\right)^t & \frac{1}{4} - \frac{1}{4} \left(1 - \frac{4}{3}\alpha\right)^t & \frac{1}{4} + \frac{3}{4} \left(1 - \frac{4}{3}\alpha\right)^t & \frac{1}{4} - \frac{1}{4} \left(1 - \frac{4}{3}\alpha\right)^t \\ \frac{1}{4} - \frac{1}{4} \left(1 - \frac{4}{3}\alpha\right)^t & \frac{1}{4} - \frac{1}{4} \left(1 - \frac{4}{3}\alpha\right)^t & \frac{1}{4} - \frac{1}{4} \left(1 - \frac{4}{3}\alpha\right)^t & \frac{1}{4} + \frac{3}{4} \left(1 - \frac{4}{3}\alpha\right)^t \end{pmatrix}$$

Note: all diagonal entries (prob. of a base remaining unchanged) are identical, also all non-diagonal entries (prob. of a base undergoing substitution) are identical.

# Jukes-Cantor model

This formula for  $M^t$  is of the Jukes-Cantor form itself, with the Jukes-Cantor parameter being

$$\frac{3}{4} - \frac{3}{4} \left(1 - \frac{4}{3}\alpha\right)^t$$

Ex-3: Can we now answer the question of the probability that a base A in ancestral sequence will have mutated to become a base T in the descendent sequence 100 time-steps later?

This is the (4,1) entry of  $M^{100}$  which is

$$\frac{1}{4} - \frac{1}{4} \left(1 - \frac{4}{3}\alpha\right)^{100}$$

# Jukes-Cantor Model

**Jukes-Cantor model is a one-parameter model of mutation,**

- it depends on the single parameter  $\alpha$  to specify the mutation.**

**Other models use several different parameters to specify mutation rates for several different types of mutations, e.g., Kimura 2-parameter and Kimura 3-parameter models**

# The Kimura Models

Kimura 2-parameter model allows for **different rates** for transitions ( $\beta$ ) and transversions ( $\gamma$ ).

If we assume these rates are **independent** of initial base, then off-diagonal entries of the transition matrix are given by:

$$M = \begin{pmatrix} * & \beta & \gamma & \gamma \\ \beta & * & \gamma & \gamma \\ \gamma & \gamma & * & \beta \\ \gamma & \gamma & \beta & * \end{pmatrix}$$

Since the columns sum to 1, this means all the diagonal entries must be  $1 - \beta - 2\gamma$ .

# The Kimura Models

Kimura 3-parameter model assumes a transition matrix of the form

$$M = \begin{pmatrix} * & \beta & \gamma & \delta \\ \beta & * & \delta & \gamma \\ \gamma & \delta & * & \beta \\ \delta & \gamma & \beta & * \end{pmatrix}$$

The equilibrium base distribution vector

$$p_0 = (1/4, 1/4, 1/4, 1/4)$$

is an eigenvector with eigenvalue 1 for both Kimura 2- and 3- parameter models, i.e., sequences evolving according to these models have uniform base distribution at all times.

# Phylogenetic Distances

# Phylogenetic Distances

With a model of DNA mutations, we can better understand how to **relate** the amount of mutation that we **observe** in comparing an ancestral sequence and descendent sequence to the amount of mutation that must have **actually** occurred.

i.e., uncover the amount of **hidden mutation** that was obscured by subsequent mutations at the same site.

# Phylogenetic Distances

Considering Jukes-Cantor model of sequence mutation:

$$M = M(\alpha) = \begin{pmatrix} 1 - \alpha & \alpha/3 & \alpha/3 & \alpha/3 \\ \alpha/3 & 1 - \alpha & \alpha/3 & \alpha/3 \\ \alpha/3 & \alpha/3 & 1 - \alpha & \alpha/3 \\ \alpha/3 & \alpha/3 & \alpha/3 & 1 - \alpha \end{pmatrix}$$

Compute the entries of  $M^t$  for  $t = 0, 1, 2, 3, \dots$

Diagonal entries of  $M^t$  – probability of observing no change at a site are

$$\frac{1}{4} + \frac{3}{4} \left( 1 - \frac{4}{3} \alpha \right)^t$$

# Phylogenetic Distances

Fraction of sites that agreed with their initial base are given by

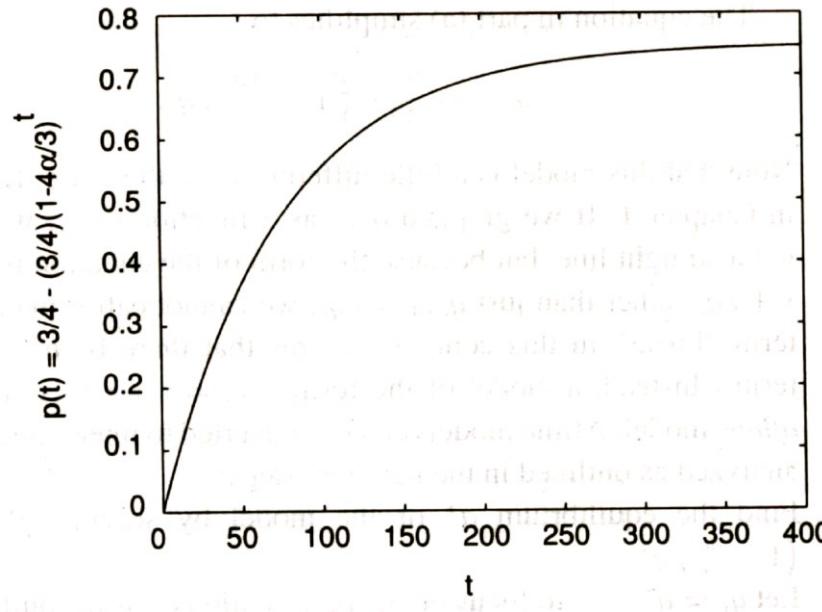
$$q(t) = \frac{1}{4} + \frac{3}{4} \left( 1 - \frac{4}{3} \alpha \right)^t$$

Fraction of sites that are different will be

$$p(t) = 1 - q(t) = \frac{3}{4} - \frac{3}{4} \left( 1 - \frac{4}{3} \alpha \right)^t$$

# Phylogenetic Distances

Fraction of sites that differ from original base gradually increases with  $t$ , approaching the value  $\frac{3}{4}$ , and never exceeds  $\frac{3}{4}$ . Why?



Jukes Cantor model  
with alpha = 0.01

Q. Even if so much mutation has occurred that the two sequences appear to be completely unrelated, you would expect to find agreement at  $1/4$  of the sites. Why?

# Jukes-Cantor Distance

For each time  $t$ ,  $p(t)$  has a different value, i.e., given any value  $0 \leq p \leq 3/4$ , we can find a  $t$  with  $p(t) = p$ , corresponding to the proportion of sites that differ between two sequences

⇒ We should be able to recover the number of elapsed time steps (assuming we know  $\alpha$ )

For real sequence data,  $p$  is easily estimated, although the elapsed time  $t$  and the mutation rate  $\alpha$  usually are not known.

Recovering them from data is our goal.

# Jukes-Cantor Distance

Suppose we have records of an original DNA sequence and a mutated version of it at a later time, but do not know either the mutation rate  $\alpha$  nor the number of elapsed time steps  $t$ .

- we can estimate  $p = p(t)$  by comparing the two sequences and using the proportion of sites that disagree in the two sequences as an estimate.
- if the original & mutated sequences are ATTGAC and ATGGCC, our estimate is

$$p(t) = 2/6 = 0.333$$

# Jukes-Cantor Distance

With  $p = p(t)$  estimated, how do we recover information on the mutation rate  $\alpha$  and the amount of elapsed time  $t$ ?

$$p(t) = \frac{3}{4} - \frac{3}{4} \left(1 - \frac{4}{3} \alpha\right)^t$$

Solving for  $t$ ,

$$t = \frac{\ln(1 - 4/3 p)}{\ln(1 - 4/3 \alpha)}$$

Note: Choice of a step size for time in formulating our model affects both the value of mutation rate  $\alpha$ , and the elapsed time between ancestor and descendent.

We cannot really expect to recover both of these.

# Jukes-Cantor Distance

Product of the two does have a meaning which is more intrinsic to what we are modeling:

$d = ta = (\text{no. of time steps})(\text{mutation rate})$

$= (\text{no. of time steps})(\text{no. of substitutions per site/time step})$

$= (\text{expected no. of substitutions per site during the elapsed time})$

$$\ln\left(1 - \frac{4}{3}a\right) \approx -\frac{4}{3}a \quad t \approx \frac{\ln(1 - 4/3p)}{-4/3a} \approx -\frac{3}{4a} \ln\left(1 - \frac{4}{3}p\right)$$

$$d = ta \approx -\frac{3}{4} \ln\left(1 - \frac{4}{3}p\right)$$

Using the approximation  $\ln(1+x) \sim x$ , for small x

# Jukes-Cantor Distance

Jukes-Cantor distance between DNA sequences  $S_0$  &  $S_1$  is defined as

$$d_{JC}(S_0, S_1) = t\alpha \approx -\frac{3}{4} \ln\left(1 - \frac{4}{3}p\right)$$

$p$  - fraction of sites that disagree in comparing  $S_0$  &  $S_1$

Provided that Jukes-Cantor model accurately describes the evolution of one sequence into another, it is an estimate of the total number of substitutions per site that occurred during the evolution

“Distance” here is an abstract notion of how different two sequences are because of mutations

# Jukes-Cantor Distance

**Ex: If between two 40-base sequence 11 sites have undergone a substitution, then  $p = 11/40 = 0.275$**

$$d_{JC}(S_0, S_1) = -\frac{3}{4} \ln \left( 1 - \frac{4}{3} \frac{11}{40} \right) \approx .3426$$

**while we observe .275 substitutions per site, we estimate that in the course of evolution 0.3426 substitutions per site occurred**

**- Hidden mutations account for the difference.**

# Jukes-Cantor Distance

If molecular clock hypothesis holds, distance computed is proportional to the amount of elapsed time; the constant of proportionality being the **mutation rate**

⇒ the distance can be thought of as a measure of how much time was required for one sequence to mutate into the other.

If molecular clock hypothesis does not hold, it is still a reconstruction of the average number of substitutions that occurred at any one site.

- the larger it is, greater the evolutionary change

# Jukes-Cantor Distance

If there is some other data (such as geological record) suggesting the time evolved, then the mutation rate can be found from  $d_{JC}$ .

- this is one way that real DNA mutation rates are estimated.

For e.g., if  $t = 10$  Myrs by some geological records, then

$$d = t\alpha = 10^7 \times \alpha = 0.33 \text{ (from } d_{JC})$$

$$\Rightarrow \alpha = 0.33 \times 10^{-7}$$

For the sequences: ATTGAC & ATGGCC,  $p(t) = 2/6 = 0.333$

# The Kimura distances

For Kimura 3-parameter model,

$$d_{K3} = -\frac{1}{4}(\ln(1 - 2\beta - 2\gamma) + \ln(1 - 2\beta - 2\delta) + \ln(1 - 2\gamma - 2\delta))$$

If  $\gamma = \delta$ , this expression gives the distance for the Kimura 2-parameter model, with  $\beta$  being the probability of transition and  $\gamma + \delta = 2\gamma$ , the probability of transversion.

If from sequence data we estimate probability of transition as  $p_1$  and transversion as  $p_2$

$$d_{K2} = -\frac{1}{2}\ln(1 - 2p_1 - p_2) - \frac{1}{4}\ln(1 - 2p_2)$$

# References

- **Mathematical Models in Biology: An Introduction, E.S. Allman and J.A. Rhodes**
- **Bioinformatics Sequence & Genome Analysis, David W. Mount**
- **Biological Sequence Analysis, Probabilistic Models of Proteins and Nucleic Acids, R. Durbin, S.R. Eddy, A. Keoghs and G. Mitchison**

# **Phylogenetic Tree Construction**

- In constructing a phylogenetic tree, the taxa we wish to relate are usually ones **currently living**.
- We have information, such as DNA sequences, from the terminal taxa and **no information** from the ones represented by internal vertices.
- We do not even know which internal vertices **should exist**, because we do not yet know the tree topology.
- Distance methods attempt to build trees using information that we believe describes the total distances between terminal taxa along the tree.

**Let's try to find evolutionary relationship between four species S1, S2, S3, and S4**

- choose a particular orthologous gene from their genomes and align the sequences**
- compute Jukes-Cantor distances between each pair of sequences.**

**These are our estimates of distances along the tree.**

	S <sub>1</sub>	S <sub>2</sub>	S <sub>3</sub>	S <sub>4</sub>
S <sub>1</sub>		.45	.27	.53
S <sub>2</sub>			.40	.50
S <sub>3</sub>				.62

# Methods for Phylogeny

## ➤ **Distance Methods**

- **UPGMA**
- **Fitch-Margoliash Algorithm**
- **Neighbour-Joining Algorithm**

## ➤ **Character-based Methods**

- **Maximum Parsimony Methods**
- **Maximum Likelihood Methods**

# Distance Methods

- **Uses the number of changes between pairs of sequences in a group to construct a tree**
- **Sequences with **fewest** changes are neighbours, *i.e.* they share a node to which they are joined by a branch**
- **Aim is to position neighbours correctly and to compute branch lengths that best fit the data**

# **Tree Construction: UPGMA**

## **Unweighted Pair-Group Method with Arithmetic Means (UPGMA)**

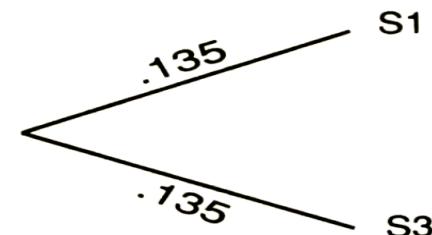
- is the simplest method for tree construction**
- assumes that the rate of change along the branches of a tree is **constant**, i.e., it assumes a molecular clock.**

**This method produces a **rooted tree**.**

# Tree Construction: UPGMA

## Table: Distances Between Taxa

	S <sub>1</sub>	S <sub>2</sub>	S <sub>3</sub>	S <sub>4</sub>
S <sub>1</sub>		.45	.27	.53
S <sub>2</sub>			.40	.50
S <sub>3</sub>				.62



**Step -1:** pick the two closest taxa: S<sub>1</sub> and S<sub>3</sub>

Draw the edges equidistant from the common ancestor:  $0.27/2 = 0.135$

# UPGMA

**Combine S1 & S3 into a group, compute distance of remaining sequences from this group, e.g., distance between S1-S3 and S2 is**

$$(.45 + .40)/2 = .425$$

	S <sub>1</sub>	S <sub>2</sub>	S <sub>3</sub>	S <sub>4</sub>
S <sub>1</sub>		.45	.27	.53
S <sub>2</sub>			.40	.50
S <sub>3</sub>				.62

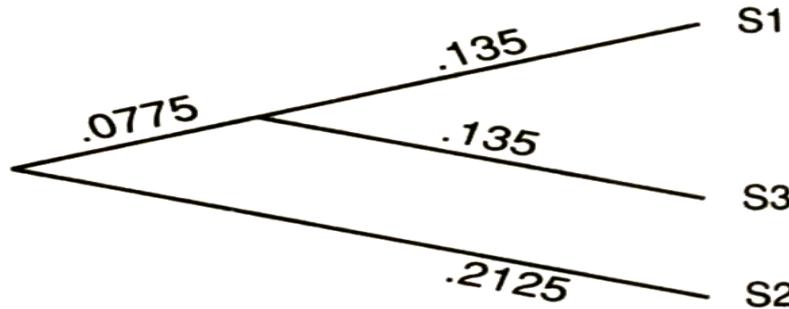
**The table then collapses to:**

**Table: Distances between Groups: UPGMA Step-1**

	S <sub>1</sub> -S <sub>3</sub>	S <sub>2</sub>	S <sub>4</sub>
S <sub>1</sub> -S <sub>3</sub>		.425	.575
S <sub>2</sub>			.50

# UPGMA

**Step-2:** Repeat the process, using the collapsed table. In the new table S1-S3 and S2 are closest:



**Edge to S2 will have length  $.425/2 = .2125$ , while the other new edge will be  $.2125 - .135 = .0775$**

	S1-S3	S2	S4
S1-S3		.425	.575
S2			.50

# UPGMA

**Step-3:** Again combining taxa, we form a group **S1-S2-S3**, and compute its distance from **S4**:  $(.53 + .5 + .62)/3 = .55$

	S1-S2-S3
S4	.55
S1	
S2	
S3	

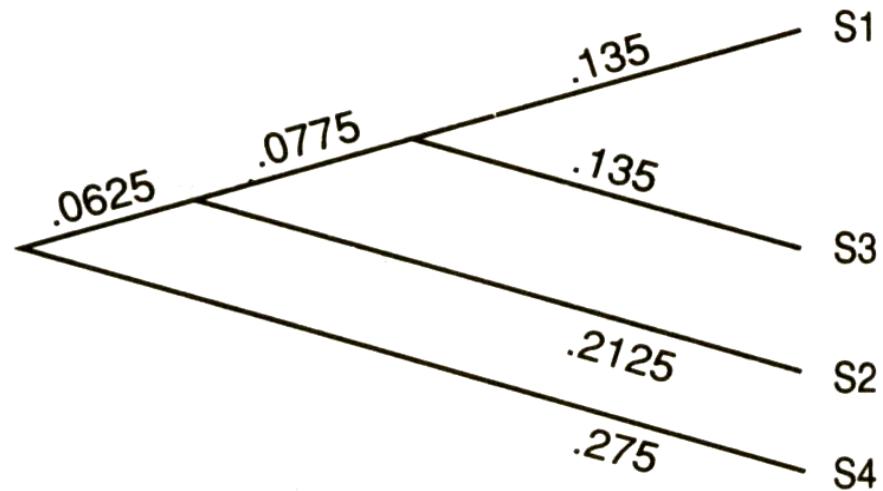
	S <sub>1</sub>	S <sub>2</sub>	S <sub>3</sub>	S <sub>4</sub>
S <sub>1</sub>		.45	.27	.53
S <sub>2</sub>			.40	.50
S <sub>3</sub>				.62

**Final tree is drawn by estimating S4 as  $.55/2 = .275$  from the root.**

**The other edge has length  $0.275 - 0.2125 = .0625$ , since that places all other taxa .275 from the root as well.**

# UPGMA

	$S_1$	$S_2$	$S_3$	$S_4$
$S_1$		.45	.27	.53
$S_2$			.40	.50
$S_3$				.62



**Does the constructed tree exactly fit the data?**

**Distance on the tree from S3 to S4 = .55, while according to the original data, it is .62!**

**Tree constructed for the data does not exactly fit the data.**

**However, the tree distances are reasonably close to the distances given by the data**

# UPGMA

**Note that the molecular clock assumption is implicit in UPGMA.**

**In this example, when we placed S1 & S3 at the ends of equal length branches, we assumed that the amount of mutation each underwent from their common ancestor was equal.**

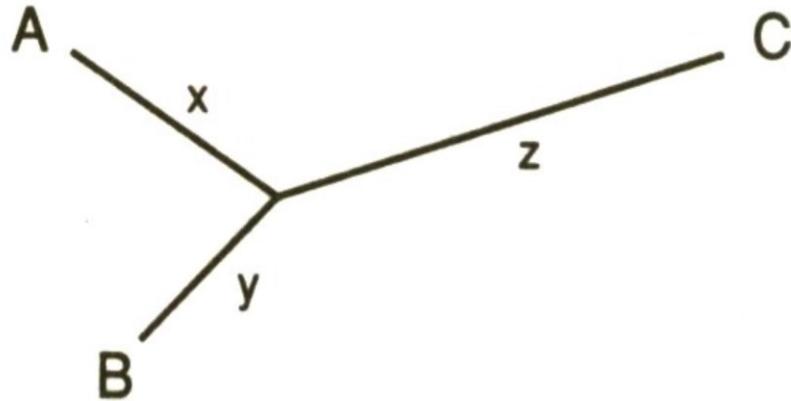
**UPGMA always places all the taxa at the same distance from the root, so that the amount of mutation from the root to any taxon is identical.**

# Fitch-Margoliash Algorithm

More complicated than UPGMA, but builds on the same basic approach.

It attempts to drop the molecular clock assumption of UPGMA.

First, let's put 3 taxa on an unrooted tree:



Distance data defined as:

$$x + y = d_{AB}$$

$$x + z = d_{AC}$$

$$y + z = d_{BC}$$

For 3 taxa, we can assign lengths to the edges to fit data exactly

# **Fitch-Margoliash Algorithm**

**These equations can be solved to give**

$$x = (d_{AB} + d_{AC} - d_{BC})/2$$

$$y = (d_{AB} + d_{BC} - d_{AC})/2$$

$$z = (d_{AC} + d_{BC} - d_{AB})/2$$

**- 3-point formula for fitting taxa to a tree**

**Fitch-Margoliash algorithm uses the 3 taxa case to handle more taxa.**

# Fitch-Margoliash Algorithm

	S1	S2	S3	S4	S5
S1	-	0.31	1.01	0.75	1.03
S2	-	-	1.00	0.69	0.90
S3	-	-	-	0.61	0.42
S4	-	-	-	-	0.37
S5	-	-	-	-	-

**As in UPGMA, choose the closest pair of taxa to join (S<sub>1</sub> & S<sub>2</sub> in this case)**

# Fitch-Margoliash Algorithm

## Step – 1:

- **Join  $S_1$  &  $S_2$  **without** placing them at an equal distance from a common ancestor**
  - **reduce to 3-taxa case by combining all other taxa into a group (i.e., group  $S_3-S_4-S_5$ )**
- **Compute the distance of  $S_1$  and  $S_2$  from the group as the average of their respective distances from  $S_3$ ,  $S_4$ , and  $S_5$**

# Fitch-Margoliash Algorithm

**Distance from  $S_1$  to  $S_3-S_4-S_5$ :**

$$d(S_1, S_3-S_4-S_5) = (1.01+.75+1.03)/3 = 0.93$$

**Distance from  $S_2$  to  $S_3-S_4-S_5$ :**

$$d(S_2, S_3-S_4-S_5) = (1.00+.69+.90)/3 = 0.863$$

**This gives us the table:**

	S1	S2	S3-S4-S5
S1	-	0.31	0.93
S2	-	-	0.863

# Fitch-Margoliash Algorithm

**Fit the data in the table to obtain the tree using 3-point formula:**

$$x + y = dS_1S_2,$$

$$x + z = dS_1G,$$

$$y + z = dS_2G,$$

$$G: S_3-S_4-S_5$$

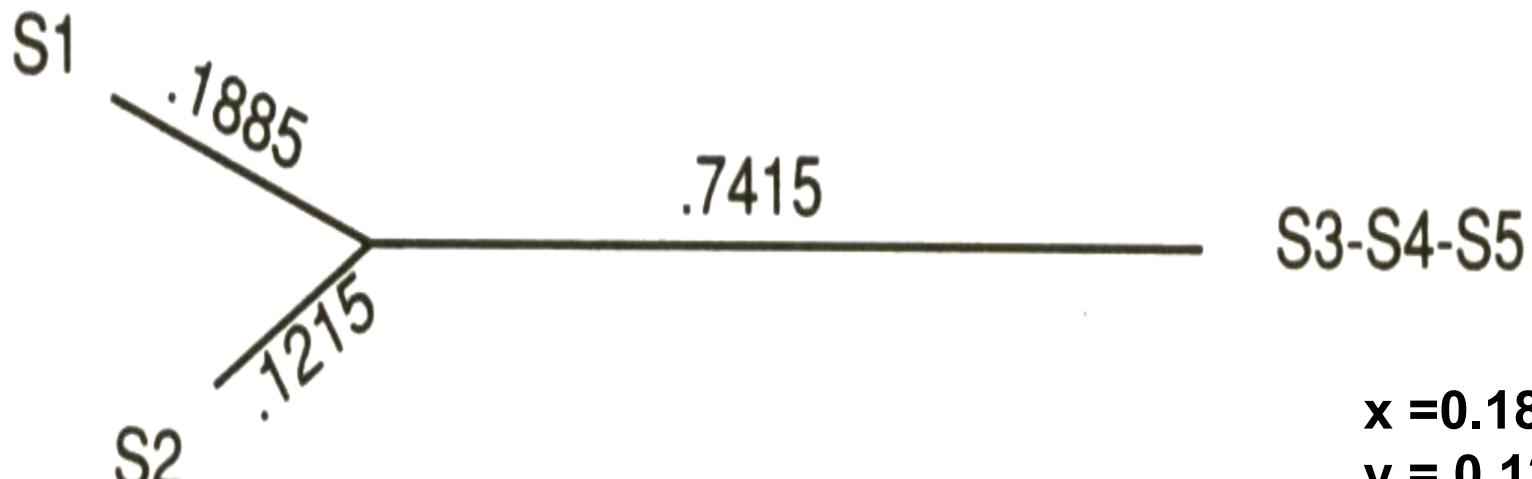
**Then,**

$$\begin{aligned} x &= (dS_1S_2 + dS_1G - dS_2G)/2 \\ &= (0.31 + 0.93 - 0.863)/2 = 0.1885 \end{aligned}$$

$$\begin{aligned} y &= (dS_1S_2 + dS_2G - dS_1G)/2 \\ &= (0.31 + 0.863 - 0.93)/2 = 0.1215 \end{aligned}$$

$$\begin{aligned} z &= (dS_1G + dS_2G - dS_1S_2)/2 \\ &= (0.93 + 0.863 - 0.31)/2 = 0.7415 \end{aligned}$$

# Fitch-Margoliash Algorithm



## FM algorithm: Step 1

**Note: S1 & S2 are not equidistant from the internal node**

**Also note that  $x + y = 0.31 = d_{S_1 S_2}$**

# Fitch-Margoliash Algorithm

## Step-2:

- **Keep only edges ending at  $S_1$  &  $S_2$  and return to original table**
- **Join  $S_1$  &  $S_2$  into a group, compute distances of remaining taxa from the group  $S_1-S_2$ :**

**Distance from  $S_3$  to  $S_1-S_2$ :**

$$d(S_3, S_1-S_2) = (1.01+1.00)/2 = 1.005$$

**Distance from  $S_4$  to  $S_1-S_2$ :**

$$d(S_4, S_1-S_2) = (0.75+0.69)/2 = 0.72$$

**Distance from  $S_5$  to  $S_1-S_2$ :**

$$d(S_5, S_1-S_2) = (1.03+0.90)/2 = 0.965$$

	S1	S2	S3	S4	S5
S1	-	0.31	1.01	0.75	1.03
S2	-	-	1.00	0.69	0.90
S3	-	-	-	0.61	0.42
S4	-	-	-	-	0.37
S5	-	-	-	-	-

# Fitch-Margoliash Algorithm

On collapsing S1-S2 into a group:

	S1-S2	S3	S4	S5
S1-S2	-	1.005	0.72	0.965
S3	-	-	0.61	0.42
S4	-	-	-	0.37

Again look for the closest pair.

# Fitch-Margoliash Algorithm

## Step 3:

- **Join the closest pair  $S_4$  &  $S_5$**
- **Compute the distances of  $S_4$  &  $S_5$  from a single temporary group  $S_1-S_2-S_3$ :**

$$d(S_4, S_1-S_2-S_3) = (.75+.69+.61)/3 = 0.683$$

$$d(S_5, S_1-S_2-S_3) = (1.03+.90+.42)/3 = 0.783$$

**This gives us the table:**

	$S_1-S_2-S_3$	$S_4$	$S_5$
$S_1-S_2-S_3$	-	0.683	0.783
$S_4$	-	-	0.37

# Fitch-Margoliash Algorithm

Applying the 3-point formula to the table:

$$x + y = dS_4S_5, \quad x+z = dS_4G, \quad y+z=dS_5G, \quad G:S_1-S_2-S_3$$

$$\begin{aligned}x &= (dS_4S_5 + dS_4G - dS_5G)/2 \\&= (0.37 + 0.683 - 0.783)/2 = 0.135\end{aligned}$$

$$\begin{aligned}y &= (dS_4S_5 + dS_5G - dS_4G)/2 \\&= (0.37 + 0.783 - 0.683)/2 = 0.235\end{aligned}$$

$$\begin{aligned}z &= (dS_4G + dS_5G - dS_4S_5)/2 \\&= (0.683 + 0.783 - 0.37)/2 = 0.548\end{aligned}$$

# Fitch-Margoliash Algorithm

$x = 0.135$

$y = 0.235$

$z = 0.548$

S1-S2-S3



**FM algorithm: Step 2**

# Fitch-Margoliash Algorithm

**Keep the edges joining  $S_4$  &  $S_5$  and discard the edge leading to the temporary group  $S_1-S_2-S_3$ .**

**So far we have joined two groups,  $S_1-S_2$  &  $S_4-S_5$ .**

**Next, compute a new table containing these two groups:**

$$d(S_1-S_2, S_4-S_5) = (0.75+1.03+0.69+0.90)/4 = 0.8425$$

$$d(S_3, S_4-S_5) = (0.61+0.42)/2 = 0.515$$

**From step-2,  $d(S_1-S_2, S_3) = 1.005$ .**

	S1-S2	S3	S4-S5
S1-S2	-	1.005	0.8425
S3	-	-	0.515

# Fitch-Margoliash Algorithm

Applying the 3-point formula again:

$$x + y = dG_1S_3, \quad x+z = dG_2S_3, \quad y+z = dG_1G_2,$$

$$G_1: S_1-S_2 \quad G_2: S_4-S_5.$$

$$\begin{aligned} x &= (dG_1S_3 + dG_2S_3 - dG_1G_2)/2 \\ &= (1.005 + 0.515 - 0.8425)/2 = 0.33875 \end{aligned}$$

$$\begin{aligned} y &= (dG_1S_3 + dG_1G_2 - dG_2S_3)/2 \\ &= (1.005 + 0.8425 - 0.515)/2 = 0.66625 \end{aligned}$$

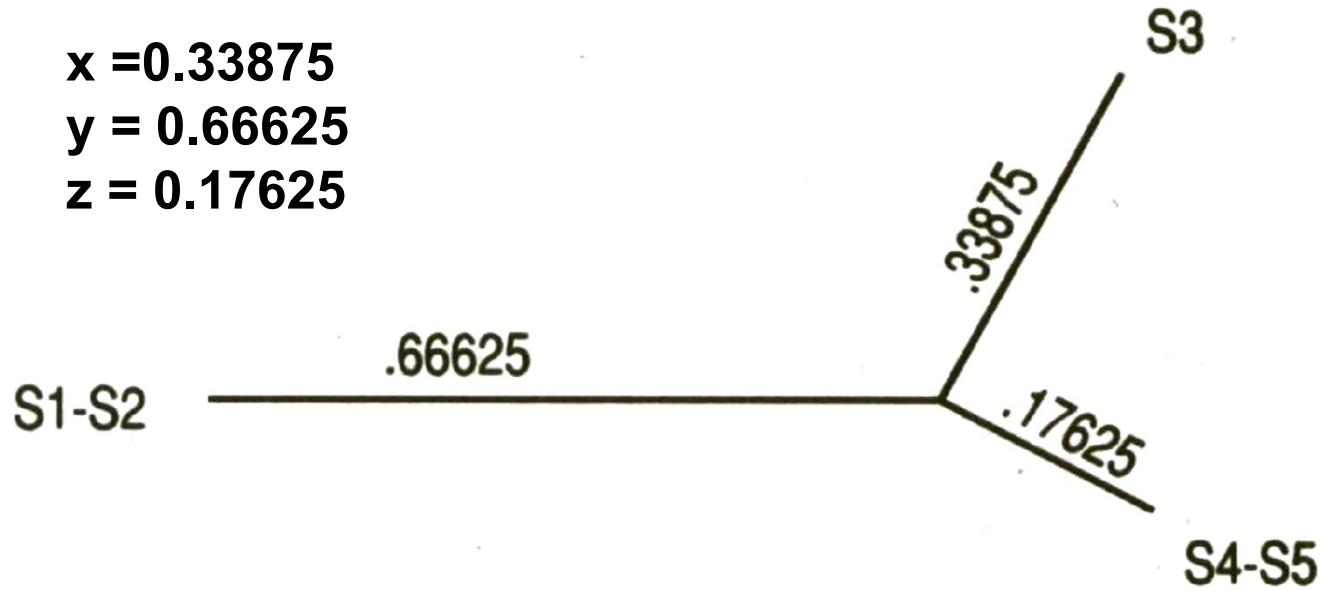
$$\begin{aligned} z &= (dG_2S_3 + dG_1G_2 - dG_1S_3)/2 \\ &= (0.515 + 0.8425 - 1.005)/2 = 0.17625 \end{aligned}$$

# Fitch-Margoliash Algorithm

$x = 0.33875$

$y = 0.66625$

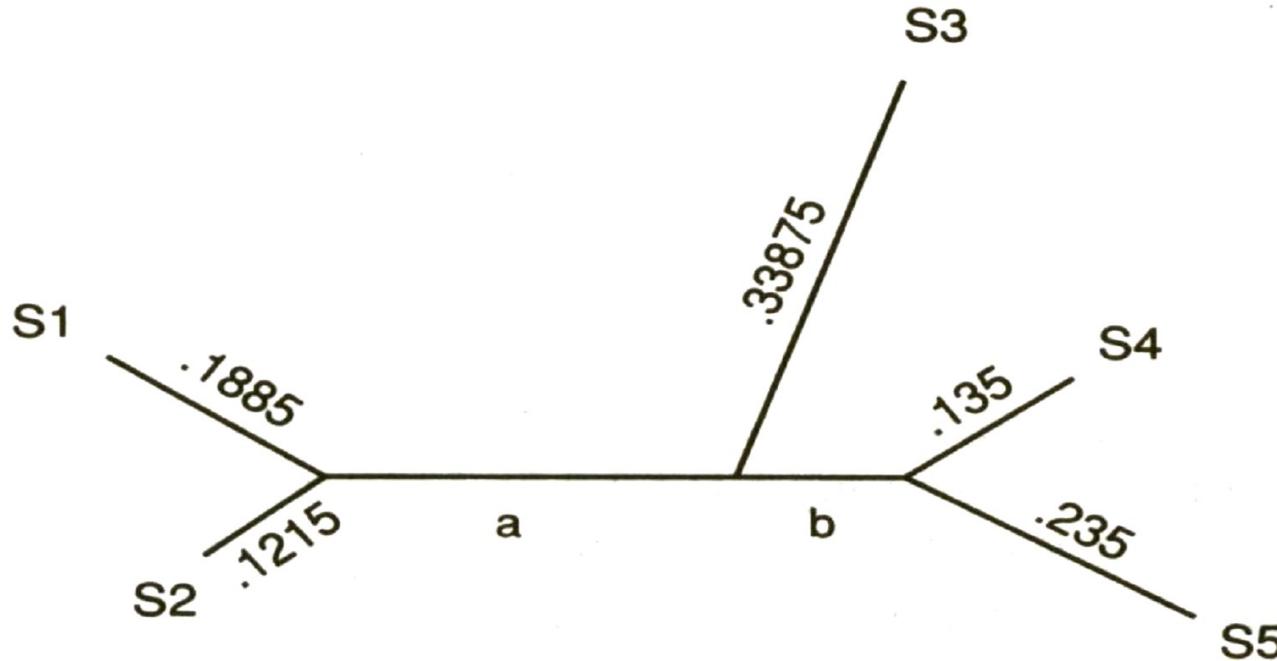
$z = 0.17625$



**FM algorithm: Step 3**

# Fitch-Margoliash Algorithm

Replacing the groups already determined in the earlier steps gives us the following tree:



**FM algorithm: Final Tree**

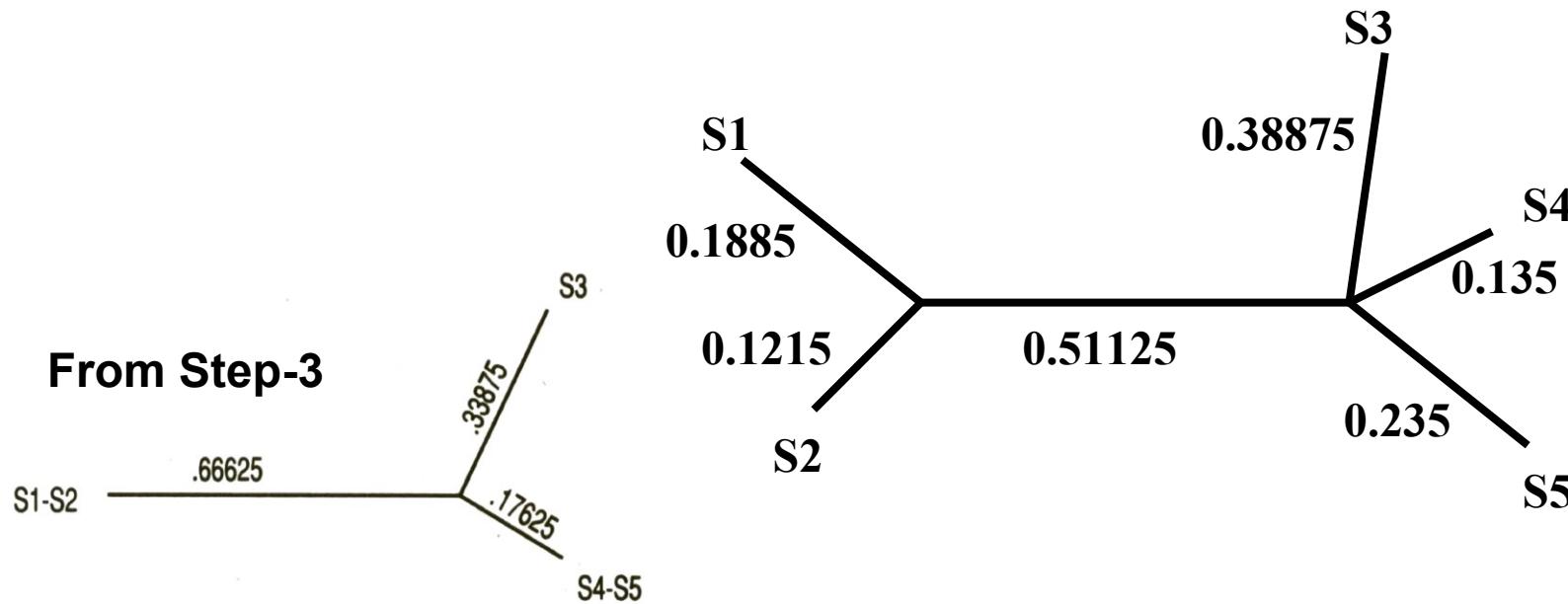
Final step is to compute the lengths **a** and **b**.

# Fitch-Margoliash Algorithm

Since  $S_1$  &  $S_2$  are on average  $(0.1885 + 0.1215)/2 = 0.155$  from the vertex joining them

$S_4$  &  $S_5$  are on average  $(0.135 + 0.235)/2 = 0.185$  from the vertex joining them, hence

$$a = 0.66625 - 0.155 = 0.51125,$$
$$b = 0.17625 - 0.185 = -0.00875.$$



# Fitch-Margoliash Algorithm

**FM algorithm and UPGMA both produce exactly the same topological tree when applied to a data set.**

**The reason being, when deciding which taxa or groups to join at each step, both methods consider exactly the same collapsed data table and both choose the pair corresponding to the smallest entry in the table.**

**Only the metric features of the resulting trees differ, undermining the hope that FM algorithm is much better than UPGMA**

# Fitch-Margoliash Algorithm

**To summarize,**

**FM produces a **better** metric tree, but topologically it **never** differs from UPGMA**

**FM does not assume molecular clock hypothesis**

**It produces an **unrooted** tree, while UPGMA gives a rooted tree**

# Rooting a Tree

**Finding a root is often desirable.**

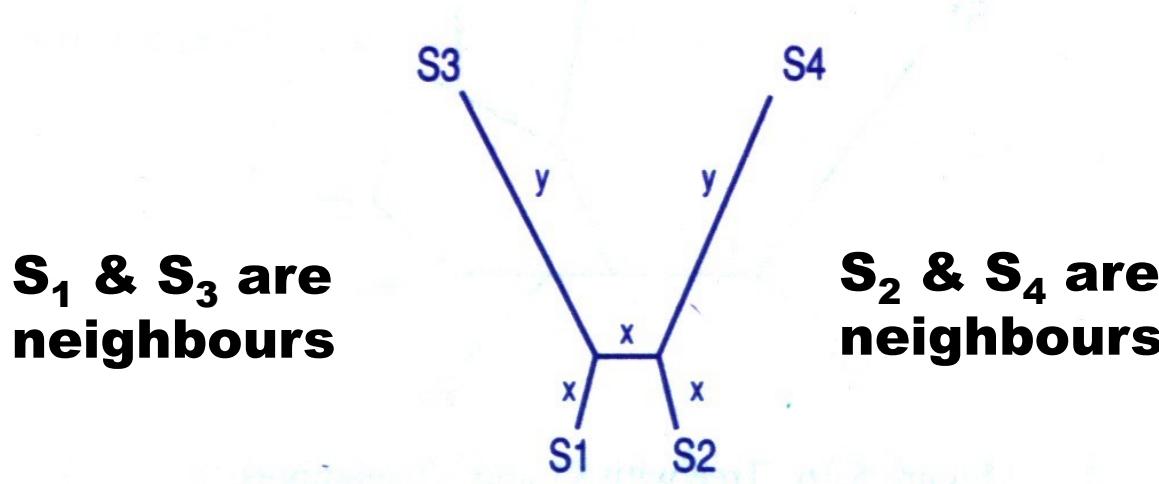
**When applying any phylogenetic tree method that produces an unrooted tree, an **additional taxon** can be included, which is chosen so that it is known to be more distantly related to each of the taxa of interest than they are to each other and is known as an **outgroup**.**

**The root is located where the edge to the outgroup joins the rest of the tree.**

# Neighbour Joining

**Both UPGMA & FM algorithm have a flaw.**

**Consider the metric tree with 4 taxa, where, x and y represent specific lengths, with  $x \ll y$ .**



**A 4-taxon metric tree with distant neighbours,  $x \ll y$**

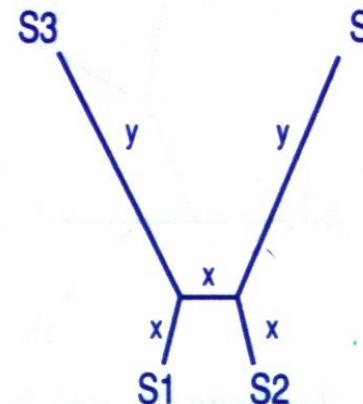
**No molecular clock operating**

# Neighbour Joining

If  $y > 2x$ , vertices  $S_1$  &  $S_2$  are closest by distance

	$S_1$	$S_2$	$S_3$	$S_4$
$S_1$		3x	$x+y$	$2x+y$
$S_2$			$2x+y$	$x+y$
$S_3$				$x+2y$

The very first joining step will be **incorrect** in FM or UPGMA, and once we join non-neighbours, we will not recover the true tree.



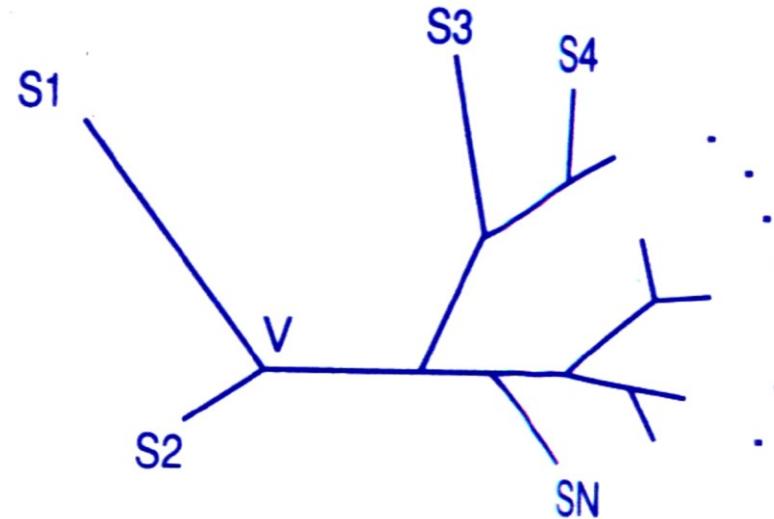
# Neighbour Joining

**Choosing the closest taxa can be misleading.**

**We need a more sophisticated criterion for choosing the taxa to join.**

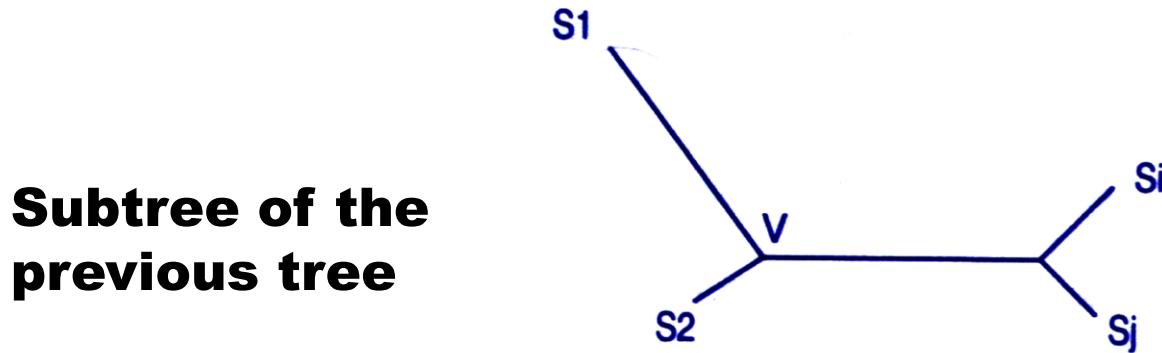
**Consider a tree in which taxa  $S_1$  and  $S_2$  are neighbours joined at vertex  $V$ , with  $V$  joined to the remaining taxa  $S_3, S_4, \dots, S_N$ .**

**Tree with  $S_1$  and  $S_2$  neighbours**



# Neighbour Joining

If the given data exactly fit this metric tree, then for every  $i, j = 3, 4, \dots, N$ , the tree would include a subtree as shown below:



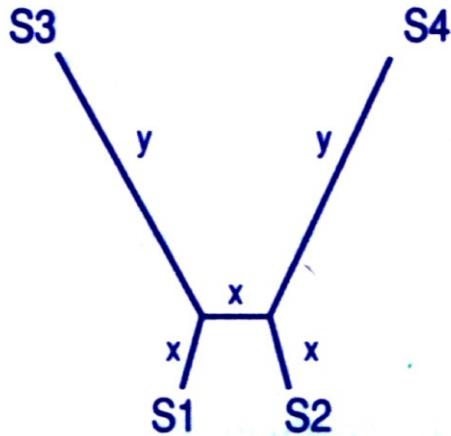
If  $S_1$  and  $S_2$  are neighbours, then for any choice of  $i, j$  between 3 and  $N$ :

$$d(S_1, S_2) + d(S_i, S_j) < d(S_1, S_i) + d(S_2, S_j)$$

- 4-point condition for neighbours, is the basis for Neighbour Joining method

# Neighbour Joining

Check this 4-pt condition for the tree:



$$d(S_1, S_3) + d(S_2, S_4) < d(S_1, S_2) + d(S_3, S_4)$$

$$(x+y) + (x+y) < 3x + (x+2y)$$

$$\Rightarrow 2x + 2y < 4x + 2y$$

This criterion holds true irrespective of whether  $x$  is  $<$ , or  $>$   $y$ .

# Neighbour Joining

**For fixed  $i$ , there are  $N - 3$  possible choices of  $j$  with  $3 \leq j \leq N$  and  $j \neq i$ . Adding up the 4-point inequalities for all  $j$ , we get**

$$(N - 3)d(S_1, S_2) + \sum_{\substack{j=3 \\ j \neq i}}^N d(S_i, S_j) < (N - 3)d(S_1, S_i) + \sum_{\substack{j=3 \\ j \neq i}}^N d(S_2, S_j)$$

**For  $N=4$ ,  $i=3$ , only 1 subtree possible:  $S_1-S_2-S_3-S_4$**

**For  $N=5$ ,  $i=3$ , 2 subtrees are possible:**

**$S_1-S_2-S_3-S_4$  ,**

**$S_1-S_2-S_3-S_5$**

# Neighbour Joining

**To simplify this relation,**

$$(N - 3)d(S_1, S_2) + \sum_{\substack{j=3 \\ j \neq i}}^N d(S_i, S_j) < (N - 3)d(S_1, S_i) + \sum_{\substack{j=3 \\ j \neq i}}^N d(S_2, S_j)$$

**Define total distance from taxon  $S_i$  to all other taxa as**

$$R_i = \sum_{j=1}^N d(S_i, S_j) \quad d(S_i, S_i) = 0$$

# Neighbour Joining

$$(N-3)d(S_1, S_2) + \sum_{\substack{j=3 \\ j \neq i}}^N d(S_i, S_j) < (N-3)d(S_1, S_i) + \sum_{\substack{j=3 \\ j \neq i}}^N d(S_2, S_j)$$

**Adding  $d(S_p, S_1) + d(S_p, S_2) + d(S_1, S_2)$  to each side of inequality, we obtain**

$$(N-2)d(S_1, S_2) + R_i < (N-2)d(S_1, S_i) + R_2$$

**Subtracting  $R_1 + R_2 + R_i$  from each side of inequality gives the more symmetric form**

$$(N-2)d(S_1, S_2) - R_1 - R_2 < (N-2)d(S_1, S_i) - R_1 - R_i$$

$$R_i = \sum_{j=1}^N d(S_i, S_j)$$

# Neighbour Joining

**Generalizing to any  $S_n$  &  $S_m$ , rather than to  $S_1$  &  $S_2$ ,**

$$M(S_n, S_m) = (N - 2)d(S_n, S_m) - R_n - R_m$$

**Then, if  $S_n$  and  $S_m$  are neighbours,**

$$M(S_n, S_m) < M(S_n, S_k) \quad \text{for all } k \neq m$$

**- criterion used for Neighbour Joining**

**Note:** This is different than  $d(S_1, S_2) < d(S_1, S_i)$  used in UPGMA and FM method

# Neighbour Joining

**Criterion used for Neighbour Joining:**

**From the distance data  $d(S_i, S_j)$ , compute a new table of values for  $M(S_i, S_j)$ .**

**Then, choose to join the pair of taxa with the smallest value of  $M(S_i, S_j)$ ,**

**i.e., if  $S_1$  and  $S_2$  are neighbours, their corresponding  $M$  value will be the smallest in the distance table**

# Outline of the NJ Method

**Step – 1:** Given distance data for  $N$  taxa, compute a new table of values of  $M$ .

Choose the smallest  $M$  value to determine which taxa to join.

**Step – 2:** If  $S_i$  &  $S_j$  are to be joined at a new vertex  $V$ , temporarily collapse all other taxa into a single group  $G$ , and determine lengths of the edges from  $S_i$  &  $S_j$  to  $V$  using the 3-point formulas, as in FM algorithm.

# Outline of the NJ Method

**Distances of  $S_i$  and  $S_j$  to the internal vertex  $V$  are given by**

$$d(S_i, V) = \frac{d(S_i, S_j)}{2} + \frac{R_i - R_j}{2(N - 2)}$$

$$d(S_j, V) = \frac{d(S_i, S_j)}{2} + \frac{R_j - R_i}{2(N - 2)}$$

**which can be written as**

$$d(S_j, V) = d(S_i, S_j) - d(S_i, V)$$

# Outline of the NJ Method

**Step – 3:** Determine distances from each taxa  $S_k$  in G to V by applying 3-point formulas to the distance data for the 3 taxa  $S_i$ ,  $S_j$  and  $S_k$ . Now include V in the table of distance data, and drop  $S_i$  and  $S_j$ .

$$d(S_k, V) = \frac{d(S_i, S_k) + d(S_j, S_k) - d(S_i, S_j)}{2}$$

**Step – 4:** Distance table now includes  $N - 1$  taxa. If there are only 3 taxa, use the 3-point formula to finish. Otherwise, go back to Step-1 and repeat.

# Example – NJ Method

**Consider the distance table to construct a tree using NJ algorithm:**

	S <sub>1</sub>	S <sub>2</sub>	S <sub>3</sub>	S <sub>4</sub>
S <sub>1</sub>		0.83	0.28	0.41
S <sub>2</sub>			0.72	0.97
S <sub>3</sub>				0.48

- (a) Compute R<sub>1</sub>, R<sub>2</sub>, R<sub>3</sub>, R<sub>4</sub>, and the table of M values for the four taxa: S<sub>1</sub>, S<sub>2</sub>, S<sub>3</sub>, & S<sub>4</sub>.**

$$R_i = \sum_{j=1}^N d(S_i, S_j)$$

$$M(S_n, S_m) = (N - 2)d(S_n, S_m) - R_n - R_m$$

# Example – NJ Method

$$\begin{aligned} R1 &= d(S1, S2) + d(S1, S3) + d(S1, S4) \\ &= 0.83 + 0.28 + 0.41 = \mathbf{1.52} \end{aligned}$$

$$\begin{aligned} R2 &= d(S2, S1) + d(S2, S3) + d(S2, S4) \\ &= 0.83 + 0.72 + 0.97 = \mathbf{2.52} \end{aligned}$$

$$\begin{aligned} R3 &= d(S3, S1) + d(S3, S2) + d(S3, S4) \\ &= 0.28 + 0.72 + 0.48 = \mathbf{1.48} \end{aligned}$$

$$\begin{aligned} R4 &= d(S4, S1) + d(S4, S2) + d(S4, S3) \\ &= 0.41 + 0.97 + 0.48 = \mathbf{1.86} \end{aligned}$$

# Example – NJ Method

Since,

$$M(S_n, S_m) = (N - 2)d(S_n, S_m) - R_n - R_m$$

$$M(S1, S2) = (4 - 2) \times 0.83 - 1.52 - 2.52 = - 2.38$$

$$M(S1, S3) = (4 - 2) \times 0.28 - 1.52 - 1.48 = - 2.44$$

$$M(S1, S4) = (4 - 2) \times 0.41 - 1.52 - 1.86 = - 2.56$$

$$M(S2, S3) = (4 - 2) \times 0.72 - 2.52 - 1.48 = - 2.56$$

$$M(S2, S4) = (4 - 2) \times 0.97 - 2.52 - 1.86 = - 2.44$$

$$M(S3, S4) = (4 - 2) \times 0.48 - 1.48 - 1.86 = - 2.38$$

# Example – NJ Method

**(b) We obtain a tie for the smallest value of M.**

**Consider any one of these smallest values, say,  
 $M(S_1, S_4) = -2.56$ , and join  $S_1$  &  $S_4$  first.**

**For the new vertex V where  $S_1$  &  $S_4$  join, compute  
 $d(S_1, V)$  &  $d(S_4, V)$  as given in Step-2, viz.,**

$$d(S_1, V) = \frac{d(S_1, S_4)}{2} + \frac{R_1 - R_4}{2(N-2)}$$

$$d(S_4, V) = d(S_1, S_4) - d(S_1, V)$$

# Example – NJ Method

## Step (b) contd.

$$d(S_1, V) = \frac{d(S_1, S_4)}{2} + \frac{R_1 - R_4}{2(N-2)}$$

$$= \frac{0.41}{2} + \frac{1.52 - 1.86}{2 \times (4 - 2)} = \frac{0.41}{2} - \frac{0.34}{4} = \frac{0.82 - 0.34}{4} = \frac{0.48}{4} = 0.12$$

$$d(S_4, V) = d(S_1, S_4) - d(S_1, V)$$

$$= 0.41 - 0.12 = 0.29$$

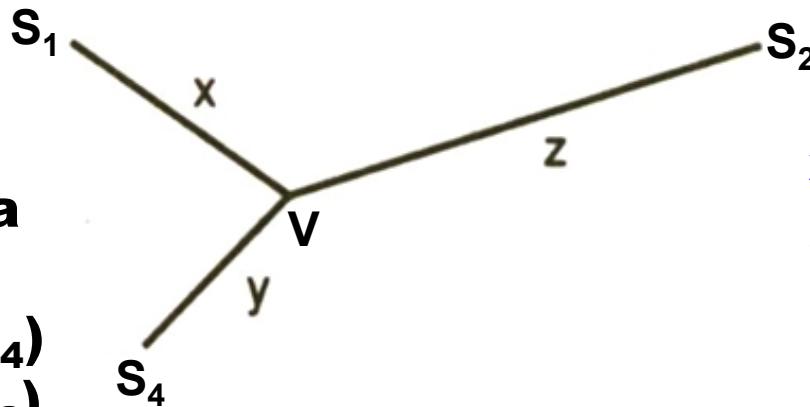
$$\mathbf{d(S_1, V) = 0.12, \quad d(S_4, V) = 0.29}$$

# Example – NJ Method

(c) Compute  $d(S_2, V)$  &  $d(S_3, V)$  using 3-point formula:

$$d(S_2, V) = \frac{d(S_1, S_2) + d(S_4, S_2) - d(S_1, S_4)}{2}$$

$$d(S_3, V) = \frac{d(S_1, S_3) + d(S_4, S_3) - d(S_1, S_4)}{2}$$



**Distance data  
defined as:**

$$x + y = d(S_1, S_4)$$

$$x + z = d(S_1, S_2)$$

$$y + z = d(S_4, S_2)$$

$$x = (d_{14} + d_{12} - d_{42})/2$$

$$y = (d_{14} + d_{42} - d_{12})/2$$

$$z = (d_{12} + d_{42} - d_{14})/2$$

**- 3-point formula for fitting taxa  
to a tree**

# Example – NJ Method

Step (c) contd.

$$d(S_2, V) = \frac{d(S_1, S_2) + d(S_4, S_2) - d(S_1, S_4)}{2}$$

$$= \frac{0.83 + 0.97 - 0.41}{2} = \frac{1.39}{2} = 0.695$$

$$d(S_3, V) = \frac{d(S_1, S_3) + d(S_4, S_3) - d(S_1, S_4)}{2}$$

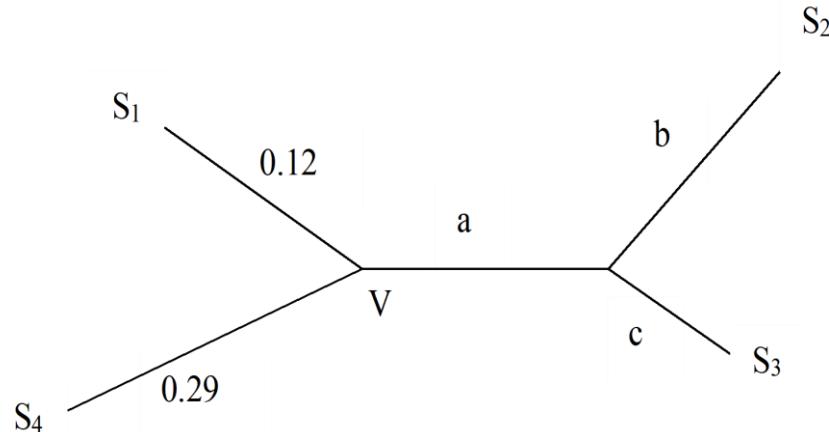
$$= \frac{0.28 + 0.48 - 0.41}{2} = \frac{0.35}{2} = 0.175$$

	V	S <sub>2</sub>	S <sub>3</sub>
V		0.695	0.175
S <sub>2</sub>			0.72

Note:  $d(S_2, S_3) = 0.72$  from the initial distance table

# Example – NJ Method

	V	$S_2$	$S_3$
V		0.695	0.175
$S_2$			0.72



**From the table above, we have**

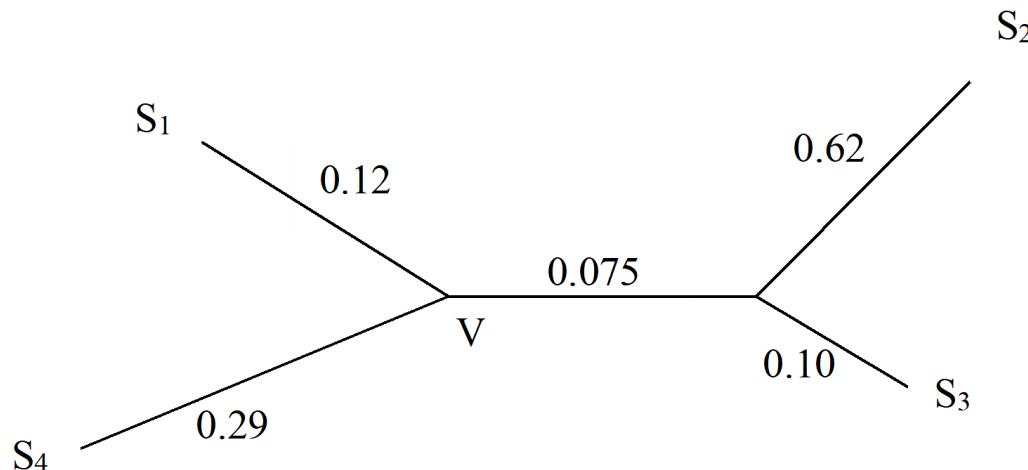
$$a + b = 0.695, \quad a + c = 0.175, \quad b + c = 0.72$$

**Solving these equations, we obtain**

$$a = 0.075, \quad b = 0.62, \quad c = 0.10$$

# Example – NJ Method

- (d) Because there are only 3 taxa left, we use the 3-point formulas to fit V, S<sub>2</sub> & S<sub>3</sub> to a tree.
- (e) Draw the final tree by attaching S<sub>1</sub> & S<sub>4</sub> to V with the distances from step (b).



	S <sub>1</sub>	S <sub>2</sub>	S <sub>3</sub>	S <sub>4</sub>
S <sub>1</sub>		0.83	0.28	0.41
S <sub>2</sub>			0.72	0.97
S <sub>3</sub>				0.48

# Summarize

Method	Tree	Molecular Clock
UPGMA	Rooted	Exists
FM	Unrooted	Does not Exist
NJ	Unrooted	Does not Exist

# Phylip Programs: Distance-based

- **dnadist** - computes distance among DNA sequences
- **protdist** - computes distance among protein sequences
- **fitch** - estimates branch length assuming additivity of branch lengths using Fitch-Margoliash method; molecular clock not assumed
- **kitsch** - same as fitch but assumes molecular clock
- **neighbor** - estimates phylogeny using neighbour-joining method

# Maximum Parsimony

## Criticism with distance methods

- reduces full DNA sequence data to a collection of pairwise distances between taxa

**Maximum Parsimony method is a different approach that uses the entire sequences**

- among all possible trees that might relate the taxa, it looks for the one that would require **fewest** possible mutations to have occurred. **Why?**

**To assess the no. of mutations, distances are not computed, how mutations occur at each separate site in the sequences is considered.**

# Maximum Parsimony

## Method:

- For a given tree, count the smallest number of mutations that would have been required if the sequences had arisen from a common ancestor according to that tree. This number is referred as the **parsimony score** of the tree.
- Compute parsimony score for all possible trees that might relate the given taxa.
- Choose the tree with **smallest parsimony score**
  - the most parsimonious tree is considered to be optimal for the given sequence data.

# Maximum Parsimony

e.g., suppose we look at a single site for 5-taxa:

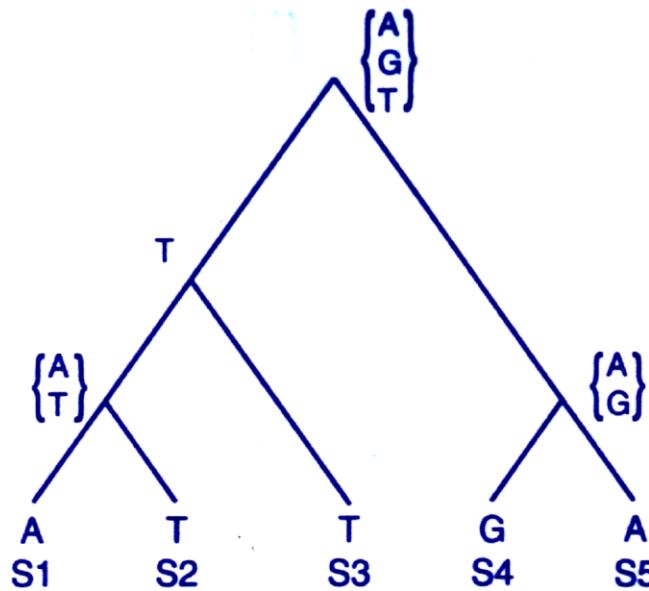
$S_1$ : A,

$S_2$ : T,

$S_3$ : T,

$S_4$ : G,

$S_5$ : A

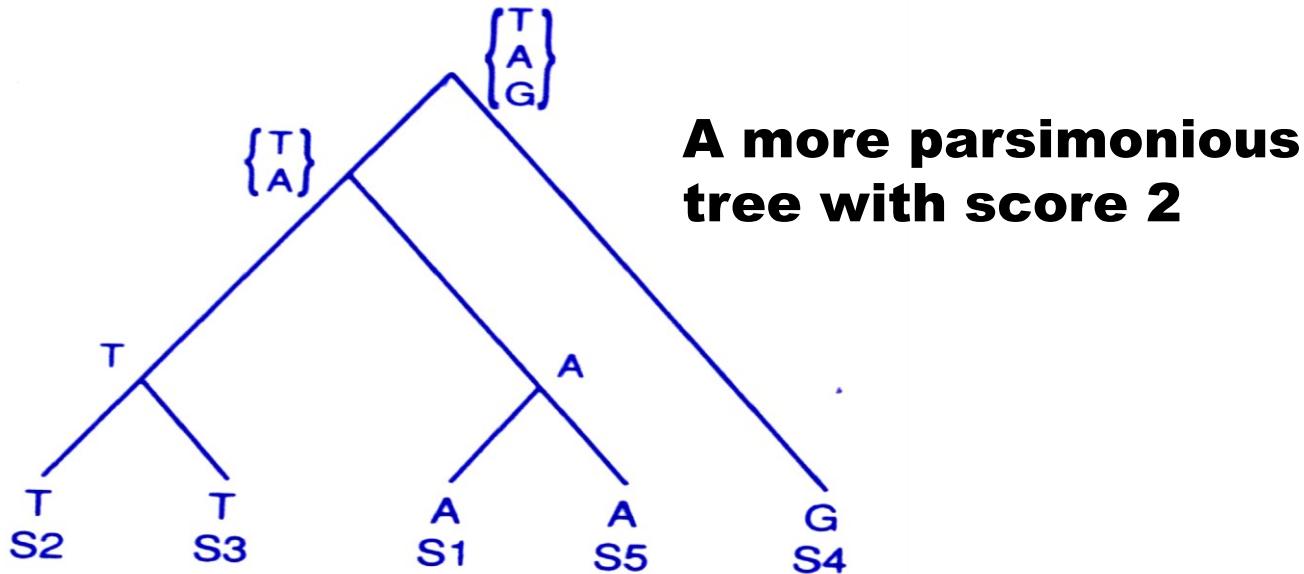


**No. of mutations = 3**  
**parsimony score = 3**

**Considering taxa related by this tree, trace backward up the tree to determine what base might have been at each vertex, assuming fewest possible mutations occurred.**

# Maximum Parsimony

Consider another tree relating the same 1-base sequences.



Labeling the internal vertices as before, we find this tree requires only two mutations. Thus, this tree is more parsimonious than the earlier one

# Maximum Parsimony

**To find the most parsimonious tree relating 5 taxa, we need to consider all 15 possible topologies of unrooted trees and compute the minimum number of mutations for each.**

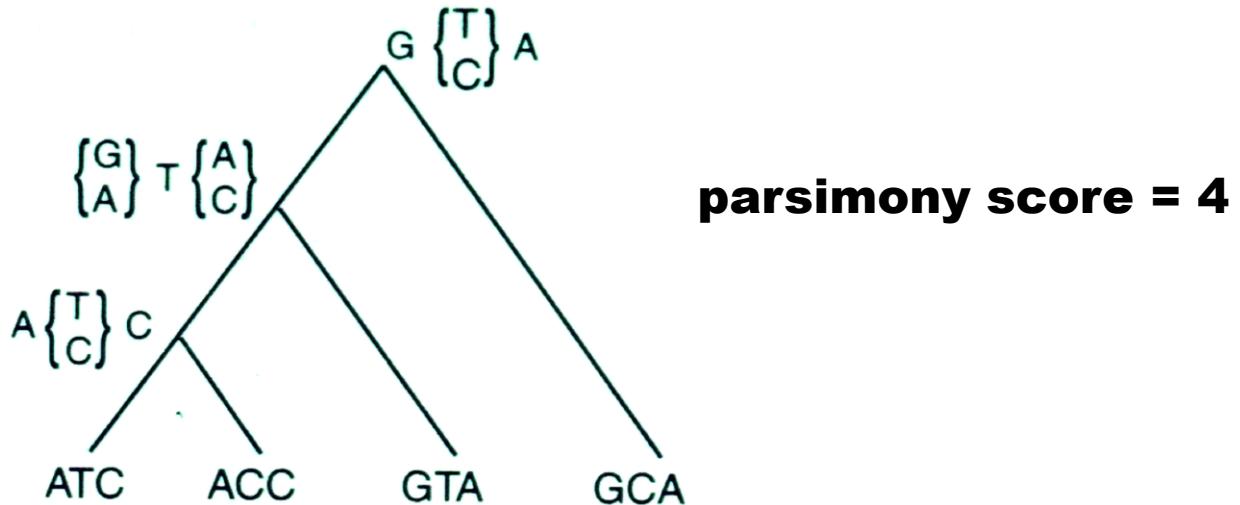
**For this example, there are 5 trees having the same parsimony score 2, the method reports **all 5 trees**, as all are equally good by selection criterion**

**When dealing with real sequence data, we need to count the no. of mutations required for a tree along **all sites** in the sequences.**

**- done in the same manner, treating each site in parallel.**

# Maximum Parsimony

Consider an example with 3 sites:



- comparing ATC and ACC, the mutation count is 1.
- At the vertex where the 3<sup>rd</sup> taxa joins, the mutation count increases to 3.
- At the root, we need a mutation in the 2<sup>nd</sup> site

# Maximum Parsimony

**As the no. of sites and the no. of taxa increase, no. of tree topologies that must be considered is huge**  
**- impractical for large sequences.**

**Some effort in using the parsimony method can be saved, if we make the observation that not all sites will affect the number of mutations needed for a tree.**

# Maximum Parsimony

- If all sequences have the **same base** at a site, then all trees will need 0 mutations for that site
  - these columns can be eliminated before applying the algorithm.
- When at a site all sequences have the same base (say A), except for **at most one** sequence each with the other bases (C, T, and G). In this case, regardless of the tree topology, if we put an A at every interior vertex, then we have the minimum possible no. of mutations
- An **informative site** is one at which at least **two** different bases occur at least **twice** each among the sequences being considered.

# Maximum Parsimony Example

Taxa	1	2	3	4	5	6	7	8	9
1	A	A	G	A	G	T	G	C	A
2	A	G	C	C	G	T	G	C	G
3	A	G	A	T	A	T	C	C	A
4	A	G	A	G	A	T	C	C	G

- **Four taxa giving 3 possible trees**
- **Sites 1, 6 & 8 not informative - do not favour a tree over another**
- **Sites 2, 3, & 4 not informative - to be informative sites should have same character in at least 2 taxa;**
- **Only informative sites analyzed: 5, 7, 9**

# Maximum Parsimony

**Maximum Parsimony method does not use any explicit model of DNA mutation, *viz.*, Jukes-Cantor model of molecular evolution.**

**Instead it carries an implicit assumption that mutation is rare, and the best explanation of evolutionary history is the one that requires the least mutation.**

# Parsimony Method: Discussion

- **Lake's method of invariance or evolutionary parsimony is another approach to identify long branches**
  - **Four sequences are considered at a time, and only transversions are scored as changes**
  - **All transversions are assumed to occur at the same rate**
- **dnainvar in Phylip computes Lake's and other phylogenetic invariants**

# Maximum Parsimony

## Problems with this method:

- **Not obvious that the method gives minimum possible mutations needed for the tree.**

**Also, one cannot assign bases to internal vertices in a way that requires fewer mutations**

**– as there can be assignments of bases to these vertices that are not consistent with this method, yet achieve the same minimum number of mutations.**

# Maximum Parsimony

- Parsimony score of a tree **does not** depend on the location of root.
  - while the counting procedure requires temporarily inserting a root, one is really judging the fitness of an unrooted tree.
- Because the method does not reliably construct the sequences at internal vertices, we have no way of knowing along which edges mutations occurred, i.e., we cannot assign a precise length to an edge by using the number of mutations occurring along it.

# Phylip Programs: Parsimony

- **dnapars:** treats gaps as 5<sup>th</sup> state
- **dnapenny:** uses branch and bound method
- **dnacomp:** based on compatibility criteria; finds tree that supports largest number of sites
- **dnamove:** performs parsimony and compatibility interactively
- **protpars:** based on no. of mutations to change a codon for aa1 to codon for aa2 for non-synonymous changes only

# Maximum Likelihood

- Maximum likelihood is similar to maximum parsimony, in that analysis is performed for each column of the alignment, all possible trees are considered, and trees with fewest changes are usually more likely
- It allows corrections for variations in the mutation rates by considering explicit evolutionary models
- Method can be used to explore relationships among more diverse sequences

# Maximum likelihood estimation

**So what's the the concept of likelihood?**

**If the probability of an event  $X$  dependent on model parameters  $p$  is written as  $P(X | p)$**

**then the likelihood of the parameters given the data is**

$$L(p | X)$$

**For most models, we find that certain data are more probable than other data.**

**Aim of maximum likelihood estimation is to find the parameter value(s) that makes the observed data most likely**

# Maximum Likelihood method

- **Specify a particular model of molecular evolution (such as Jukes-Cantor, Kimura, etc.).**
  - **parameters of this model give us the rate of mutation observed in these sequences.**
- **Consider a specific tree for relating our taxa.**
- **Compute the probability that the DNA sequence in our data could have been produced.**
  - **this is the likelihood of the tree, given our data.**
- **Repeat this process for all other trees and compute likelihood value for each.**
  - **choose the tree with the greatest likelihood as the tree best fitting the data.**
- **Observed data - mutations observed in the sequences**

# Maximum Likelihood method

- **ML assumes a model of evolution, which defines the probability/rate with which nucleotides mutate**
- **Probability of each tree is product of mutation rates in each branch**
- **Likelihoods given by each column are multiplied to give the likelihood of the tree**
- **Phylip programs `dnaml` and `dnamlk` (same as `dnaml` except that it assumes a molecular clock)**

# Maximum Likelihood method

## Problems with this method:

- **First, depends on choosing a specific model of evolution, and if that model does not describe the real process well, one could question the validity of the method.**
- **Second, as with parsimony, the method requires considering all possible trees, and so is computationally very intensive.**

# **Which method is the Best?**

**One of the difficulties of picking a method is that one can find good arguments for and against them all.**

**Cautious approach - always use a number of different methods on the data.**

**Rather than trusting a single method to give an accurate tree, check to see if different methods give roughly the same results.**

**They often do, and if they do not, it is worth investigating why they don't.**

# Bootstrapping

**Once a tree has been chosen by some method, it would be desirable to quantify how **confident** one is of it.**

**This is given by the statistical technique - **bootstrapping**.**

**In this procedure, the true data sequences are used to create a set of new **pseudo-replicate** sequences of the same length.**

**Bases at a particular site in the new sequences are chosen to be the bases appearing in a randomly chosen site in the original sequences.**

# Bootstrapping

**A tree is constructed using the pseudo-replicate sequences**

- the procedure is repeated many times, giving a large collection of bootstrap trees.**

**If a high percentage of bootstrap trees are in agreement with the one produced using original data, then we may be more confident of it.**

**Based on the concept that the data accurately reflects the variation in the population; by repeated sampling of the data the effect of this variation on any statistic of interest can be understood**

# Bootstrapping

**An important caveat on using bootstrapping is that**

**- the technique only helps assess the effects on tree construction of variability **within** the sequences.**

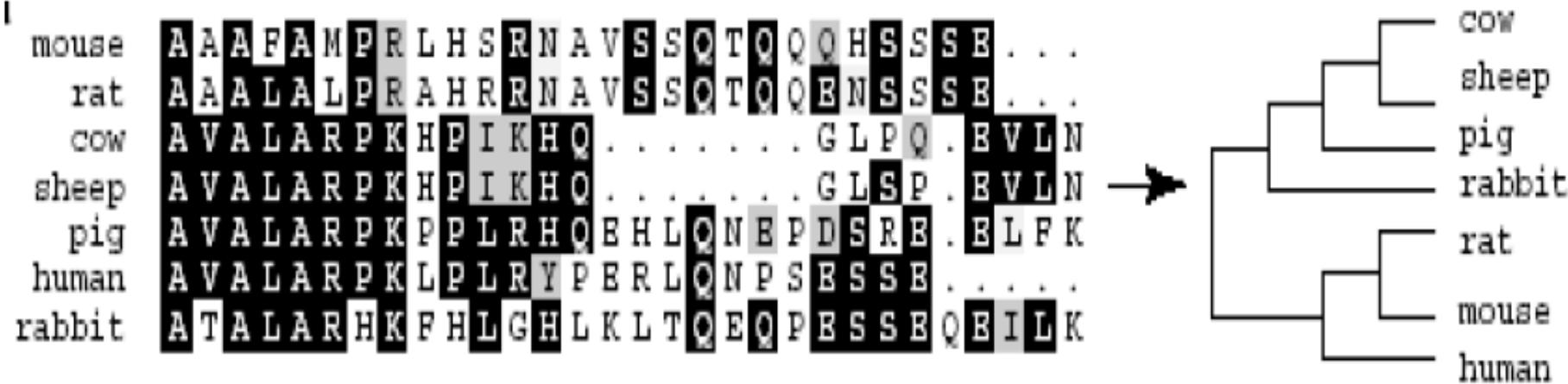
**Bootstrapping says nothing about the fundamental soundness of the method by which we choose a tree**

**– it only indicates how variability in the data affects the outcome of the method.**

# Bootstrapping

Initial alignment and tree:

*Each aligned site is considered independent*



In bootstrap analysis:

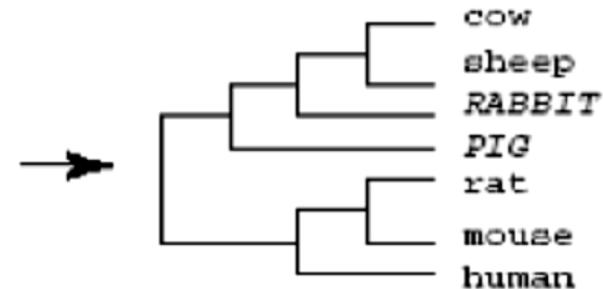
- All sites are considered independent as if freely available in a 'hat' to pick
- Available sites are picked up randomly to reconstruct a new alignment of the original size and a new phylogeny

# Bootstrapping

- The process is repeated many times to ascertain the strength of clustering. New “alignment” may contain several sites multiple times while some other sites may be absent (sampling with replacement)

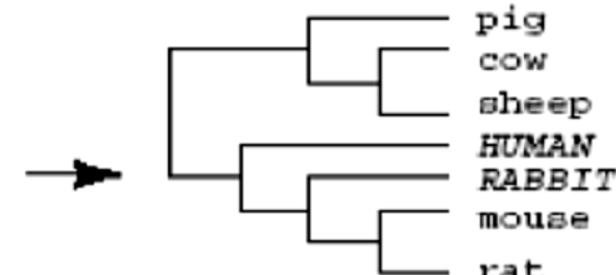
**Sequence 1:**

S	.	Q	S	S	A	T	L	P	O	F	.	M	Q	M	V	S	R	S	S	A	Q	R	H	H	S	H	S	A	A	
S	.	E	S	S	A	T	A	P	Q	L	.	L	Q	L	V	R	R	R	R	A	Q	R	H	N	S	H	S	A	A	
P	N	G	.	Q	Q	.	H	P	.	L	V	R	.	R	.	I	K	I	I	A	.	K	P	L	P	P	.	Q	Q	.
S	N	G	.	P	Q	.	H	P	.	L	V	R	.	R	.	I	K	I	I	A	.	K	P	L	S	P	.	Q	Q	.
R	K	D	L	B	Q	N	P	P	E	L	L	R	Q	R	E	L	R	L	L	A	E	R	P	S	R	P	L	Q	Q	
S	.	E	L	E	P	N	L	P	P	L	.	R	Q	R	E	L	R	L	L	A	P	R	P	S	S	P	L	P	P	
S	K	E	T	E	L	E	F	H	Q	L	I	R	Q	R	K	L	G	L	L	A	Q	G	H	S	S	H	T	L	L	



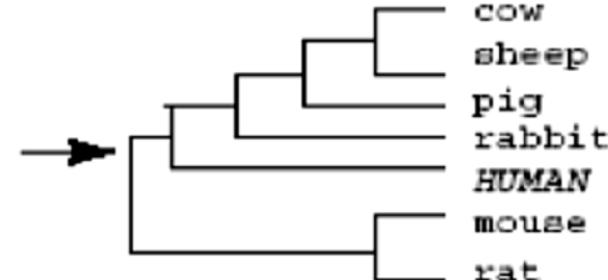
**Sequence 2:**

V	Q	Q	A	L	S	.	R	T	S	S	O	M	R	S	O	A	S	S	O	L	P	O	S	V	S	A	T	S	A	R	
V	E	Q	A	A	R	.	R	T	S	S	Q	L	R	S	Q	A	S	S	Q	A	P	Q	R	V	R	A	T	S	A	R	
.	G	.	A	H	I	N	K	.	Q	.	.	R	K	.	.	Q	P	.	.	H	P	.	I	.	I	A	.	Q	Q	K	.
.	G	.	A	H	I	N	K	.	P	.	.	R	K	.	.	Q	S	.	.	H	P	.	I	.	I	A	.	P	Q	K	.
E	D	E	A	P	L	K	K	N	E	H	P	R	K	L	P	Q	R	L	E	P	P	E	L	E	A	N	E	Q	R		
E	E	P	A	L	L	.	K	N	E	R	S	R	K	L	S	P	S	L	P	P	L	E	L	A	N	E	P	R			
K	E	Q	A	F	L	K	K	E	E	L	P	R	K	T	P	L	S	T	Q	F	H	Q	L	K	L	A	E	E	L	G	



**Sequence 3:**

A	.	S	M	Q	S	V	.	A	R	S	Q	H	V	Q	A	Q	R	H	P	R	F	S	T	A	T	R	A	.	M	
A	.	R	L	Q	S	V	.	A	R	S	Q	H	V	E	A	E	R	H	P	R	L	S	T	A	T	R	A	.	L	
Q	L	I	R	.	P	.	V	A	K	.	P	.	G	A	G	K	P	P	K	L	.	V	.	K	Q	N	R	.	Q	
Q	L	I	R	.	S	.	V	A	K	.	P	.	G	A	G	K	P	P	K	L	.	V	.	K	Q	N	R	.	Q	
Q	F	L	R	E	R	E	L	A	R	.	E	P	E	D	A	D	K	P	P	K	L	H	N	V	N	K	O	K	R	.
P	.	L	R	P	S	E	.	A	R	.	P	P	E	E	A	E	K	P	P	K	L	R	N	V	N	K	P	.	R	
L	L	L	R	Q	S	K	I	A	G	Q	Q	H	K	E	A	B	K	H	H	K	L	L	E	T	E	K	L	K	R	



# Bootstrapping

- **Phylogenies are compared to calculate Bootstrap values that signify the number of times a given branch/cluster occurred in the Multiple bootstrap trees**
- **Higher the value - higher the confidence of phylogenetic inference**
- **In general values < 50% provide very poor support**

# Comparison of Methods

Neighbor-joining	Maximum parsimony	Maximum likelihood
<b>Uses only pairwise distances</b>	<b>Uses only shared derived characters</b>	<b>Uses all data</b>
<b>Minimizes distance between nearest neighbors</b>	<b>Minimizes total distance</b>	<b>Maximizes tree likelihood given specific parameter values</b>
<b>Very fast</b>	<b>Slow</b>	<b>Very slow</b>
<b>Easily trapped in local optima</b>	<b>Assumptions fail when evolution is rapid</b>	<b>Highly dependent on assumed evolution model</b>
<b>Good for generating tentative tree, or choosing among multiple trees</b>	<b>Best option when tractable (&lt;30 taxa, homoplasy rare)</b>	<b>Good for very small data sets and for testing trees built using other methods</b>

# Summarize

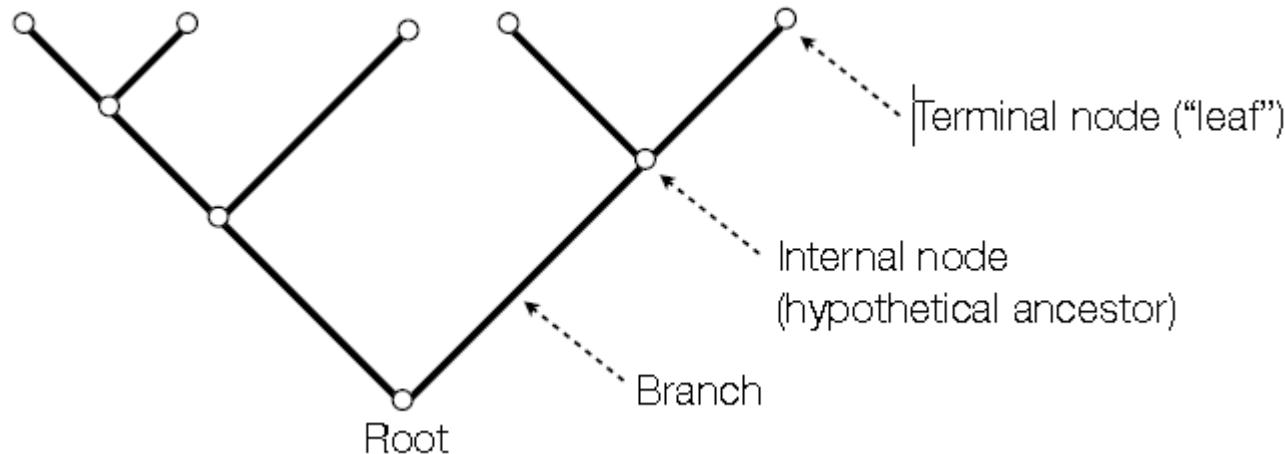
**There are many approaches to phylogenetic inference that are not sequence-based.**

**Evidence of all should be weighed before making too strong a statement about the phylogeny of the species under consideration.**

# Phylogenetic Trees

The sequences we want to relate could come from different **species**, **subspecies**, **populations**, or even **individuals**, each source of the DNA sequence is termed a ***taxon***.

An equivalent term in common use is ***operational taxonomic unit***, abbreviated as **OTU**.



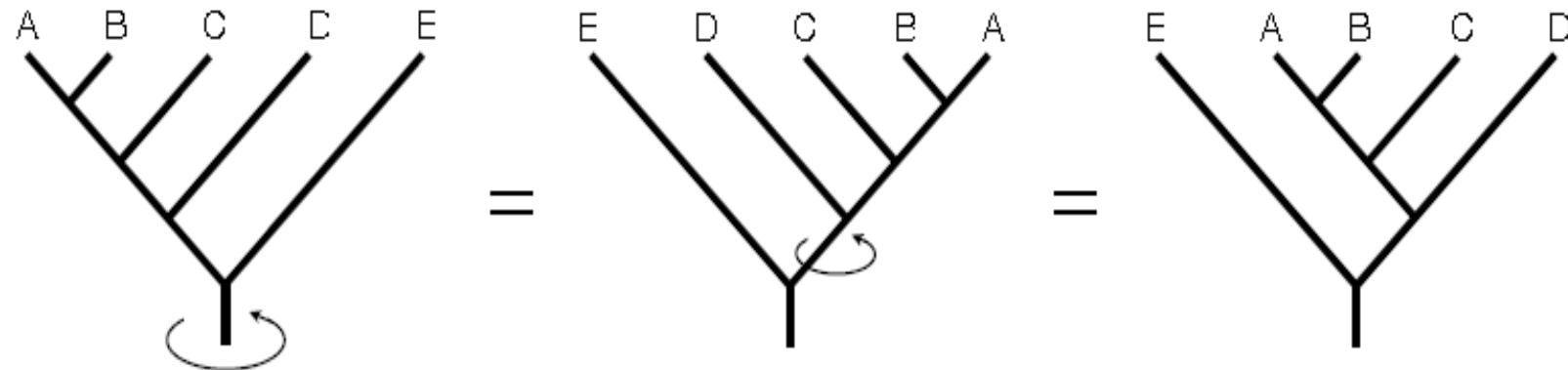
# Topological Trees

**Two trees are said to be topologically the same if we can **bend and stretch** the edges of either one to get the other.**

**We are not allowed to cut off an edge and reattach it elsewhere; doing that may give us a tree that is topologically distinct from the original one.**

# Topological Trees

Two **rooted trees** are topologically the same if one can be deformed into the other without moving the root. Edge lengths can be changed, but not the branching structure.

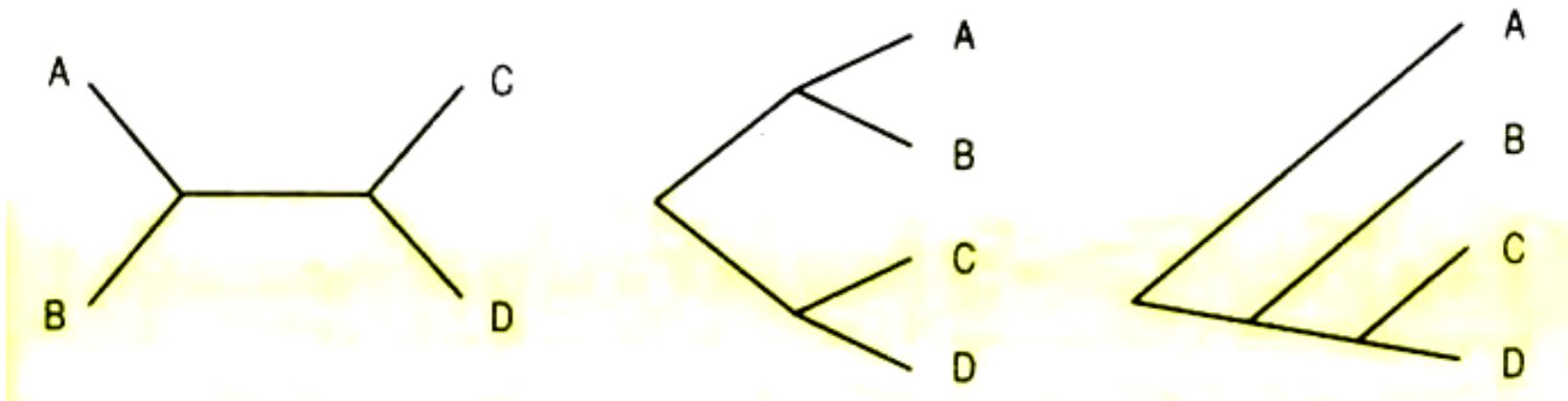


Three different representations of the same tree-topology

- A rooted tree has **directionality** (nodes can be ordered in terms of “earlier” or “later”).
- In rooted tree, **distance between two nodes is represented along the time-axis only (the 2<sup>nd</sup> axis just helps spread out the leafs)**

# Topological Trees

**Topologically same trees:**



**In unrooted trees there is no directionality: we do not know whether a node is earlier or later than another node.**

**Distance along branches directly represents node distance.**

# Topological Trees

**No. of rooted trees (for n species):**

$$(2n - 3)!/(2(n-2)[n - 2]!)$$

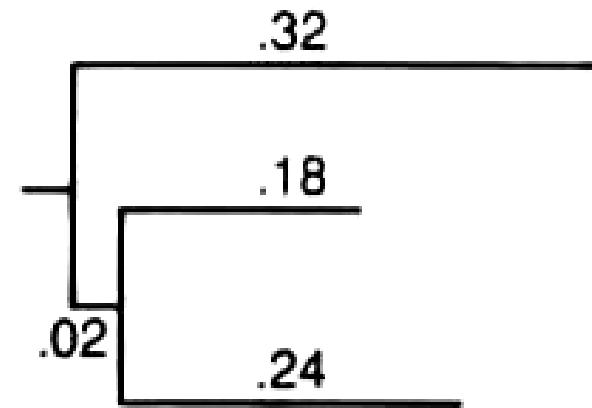
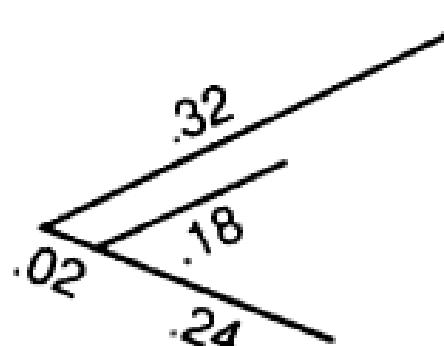
**No. of unrooted trees:**

$$(2n - 5)!/(2(n-3)[n - 3]!)$$

**For rooted trees, the no. of distinct trees grow faster: for 4 species - the no. is 15, for 5 species it is 105, ... , for 12 species the no. is more than 13 billion!**

**For unrooted trees the no. of possible trees is much less: for e.g., only 1 unrooted tree relating 3 taxa, 3 trees relating 4 taxa, 15 trees for 5 taxa, ..., for 12 species it is a little more than half a billion**

# Metric Trees



**Alternate depictions of the same metric tree**

# Problems with Tree Building

**Numerous problems are associated with tree-building using molecular data.**

**We might wonder the adequacy of a given mutation model as a description of the data.**

**A number of assumptions have been made in modeling the mutation process:**

- **Sites evolve independently of one another**
- **Sites evolve according to the same stochastic (Markov) model.**
- **The tree is rooted.**
- **The sequences are well-aligned.**

# References

- **Mathematical Models in Biology: An Introduction, E.S. Allman and J.A. Rhodes**
- **Bioinformatics Sequence & Genome Analysis, David W. Mount**
- **Biological Sequence Analysis, Probabilistic Models of Proteins and Nucleic Acids, R. Durbin, S.R. Eddy, A. Keoghs and G. Mitchison**