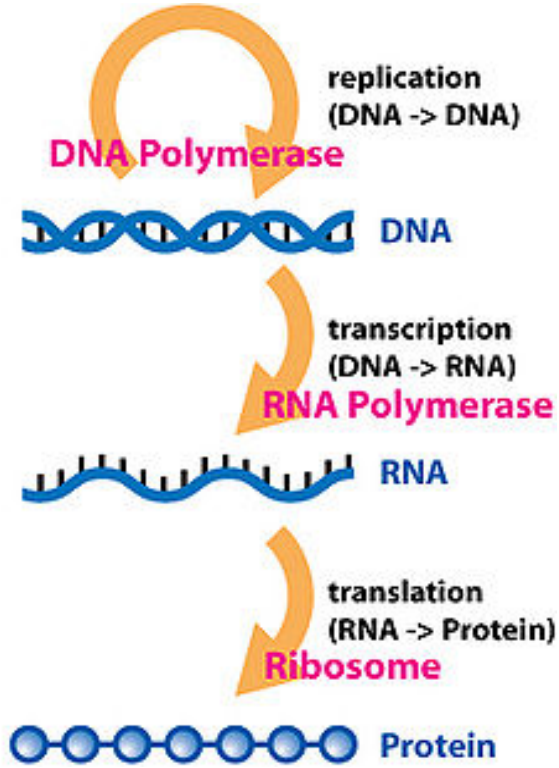
Introduction to Modeling and Algorithms in Life Sciences

Ananth Grama
Purdue University
http://www.cs.purdue.edu/homes/ayg

Acknowledgements

- To various sources, including Prof. Michael Raymer, Wiki sources (pictures), and other noted attributions.
- To the US National Science Foundation and the Center for Science of Information.

Central Dogma of Molecular Biology



http://en.wikipedia.org/wiki/Central_dogma_of_molecular_biology

Central Dogma of Molecular Biology

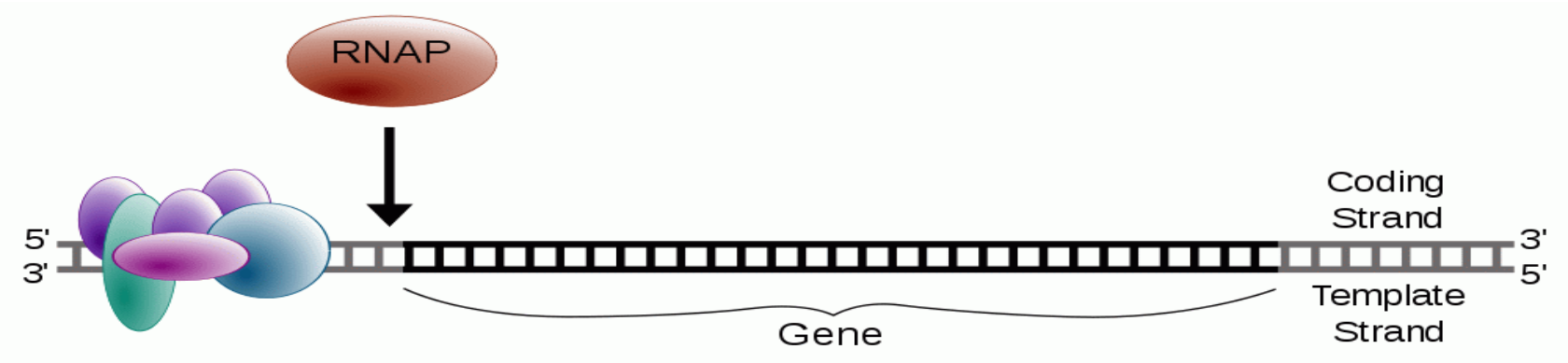
- Mostly valid with some exceptions:
  - Reverse Transcription: Retroviruses such as Feline Leukemia, HIV
  - RNA Replication: RNA to RNA transfer in viruses
  - Direct Translation: DNA to Protein (typically in cell fragments)

Protein Synthesis

- Transcription: a DNA molecule is converted into a complementary strand of RNA
- This RNA is also called messenger RNA (mRNA) since it acts as an intermediary between DNA and the Ribosomes
- Ribosomes are parts of cell that synthesize proteins from mRNA

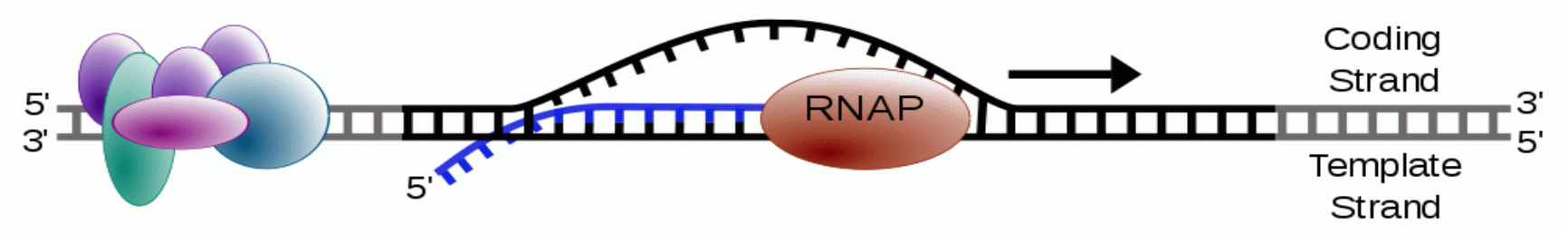
Transcription: Initiation

- Transcription is affected by an enzyme called RNA Polymerase (RNAP)
- RNAP binds to core promoters (in the presence of transcription factors) and pre-initiates the transcription process
- The essentials for pre-initiation include the core promoter sequence, transcription factors, DNA Helicase, RNA Polymerase, and activators/ repressors.



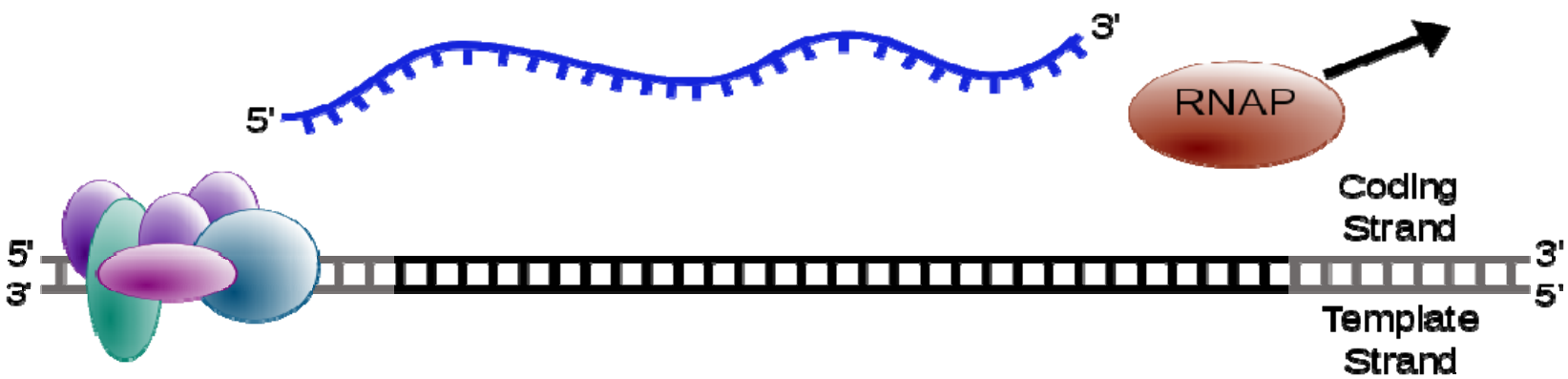
Transcription: Elongation

- One strand of DNA is used as a template for RNA synthesis
- RNAP traverses the template, using base-pair complementarity to synthesize RNA

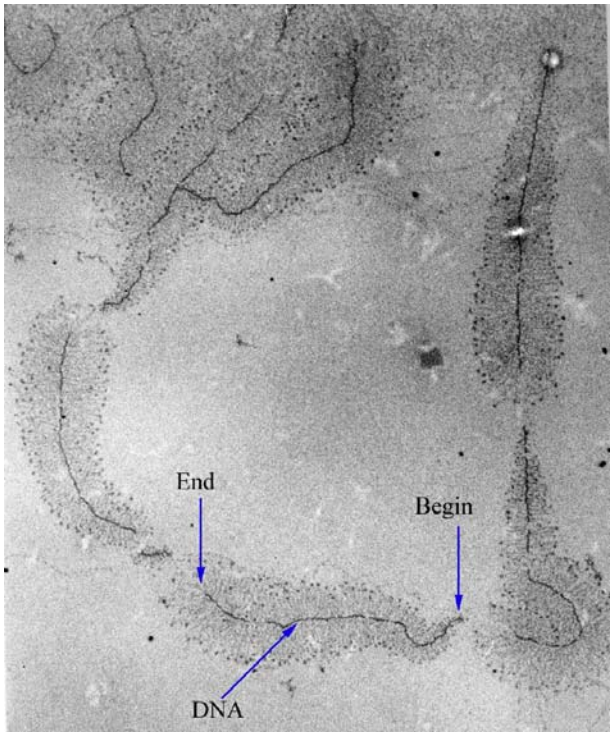


Transcription: Termination

- The transcript is cleaved (typically induced by a termination sequence on the template) followed by a template-independent addition of As at its new 3' end, in a process called polyadenylation.

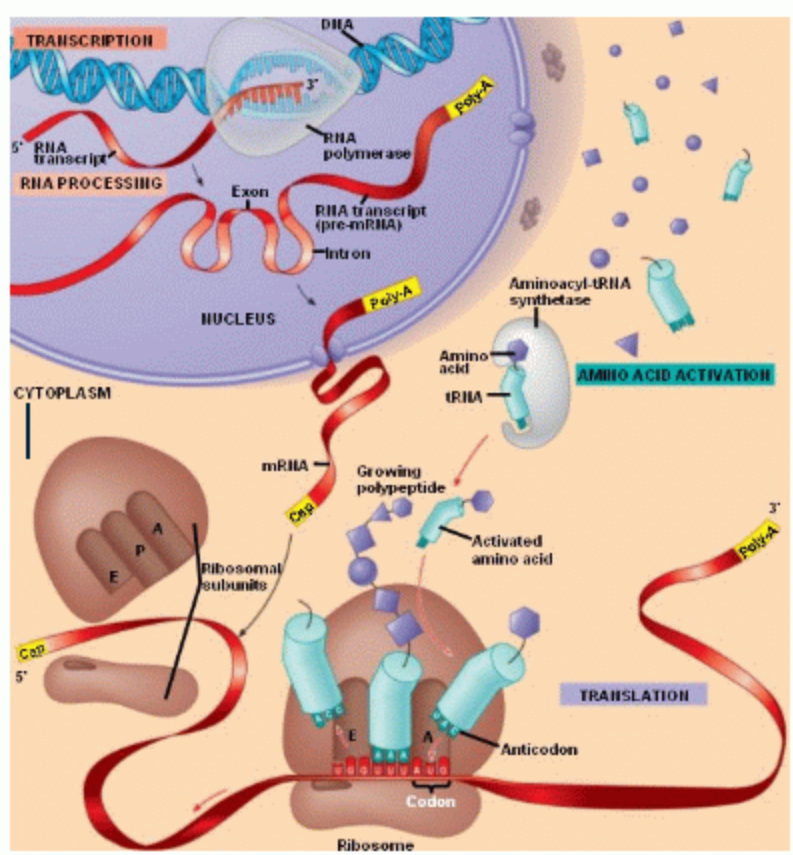


Transcription: Snapshot



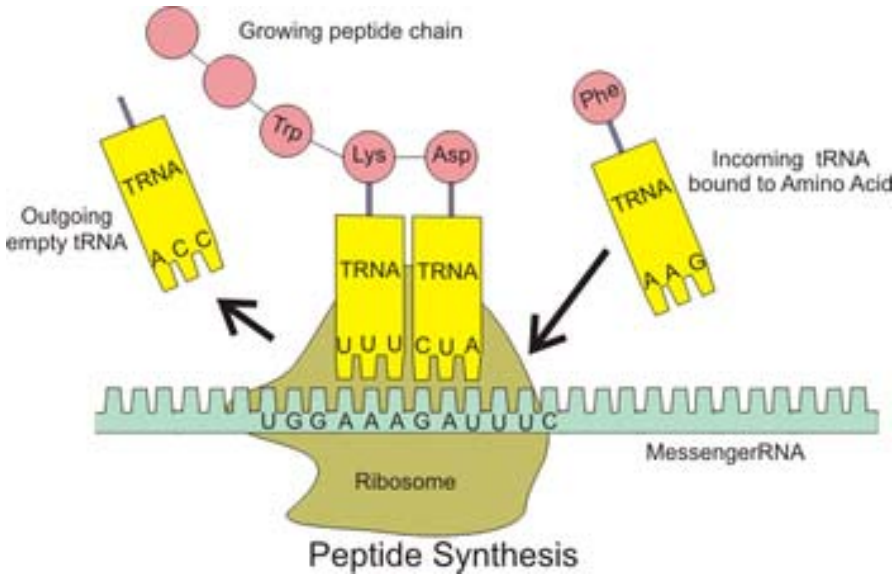
Micrograph of gene transcription of ribosomal RNA illustrating the growing primary transcripts. "Begin" indicates the 5' end of the coding strand of DNA, where new RNA synthesis begins; "end" indicates the 3' end, where the primary transcripts are almost complete. [Trepte, 2005]

Eukaryotic Transcription



Synthesizing Proteins: Translation

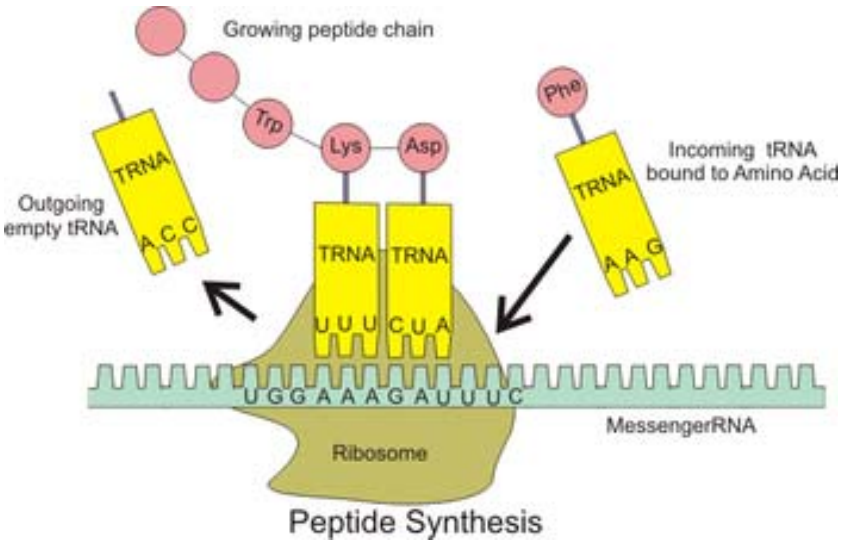
- mRNA is decoded by the Ribosome to produce specific proteins (polypeptide chains)
- Polypeptide chains fold to make active proteins
- The amino acids are attached to transfer RNA (tRNA) molecules, which enter one part of the ribosome and bind to the messenger RNA sequence.



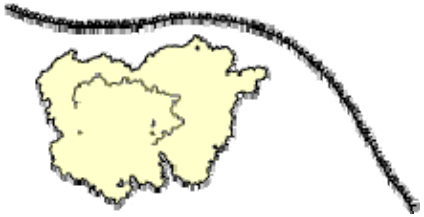
http://en.wikipedia.org/wiki/File:Peptide_syn.png

Translation

- Each combination of 3 nucleotides on mRNA is called a **codon.**
- Each codon specifies a **particular amino acid** that is to be placed in the polypeptide chain.
- Using the mRNA as a template, the ribosome traverses each codon of the mRNA, pairing it with the appropriate amino acid provided by a tRNA. Molecules of transfer RNA (tRNA) contain a complementary anticodon on one end and the appropriate amino acid on the other.

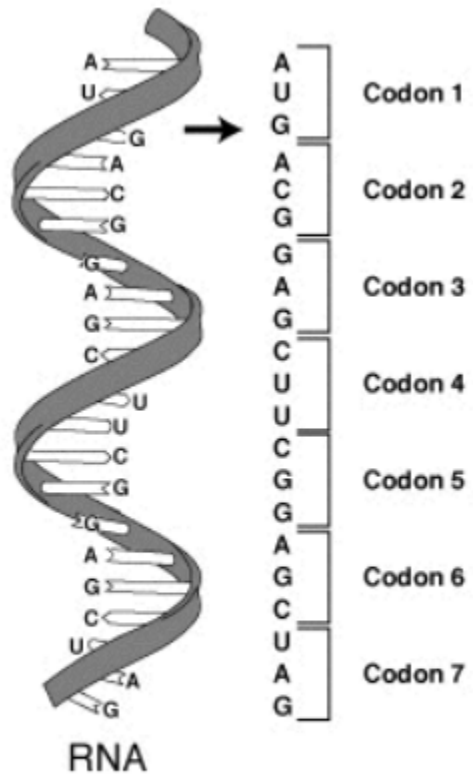


Translation



Bensaccount at en.wikipedia

Translation



Translation

- Since three mRNA nucleotides form a codon, and each nucleotide can be selected from among four bases, there are 64 possible combinations
- Of there, three are stop codons: UAA, UAG, UGA
- The codon AUG serves as a start codon, in addition to coding the amino acid methionine
- Since 61 codons (64 – 3) code 20 aminoacids, there is considerable redundancy in the code.

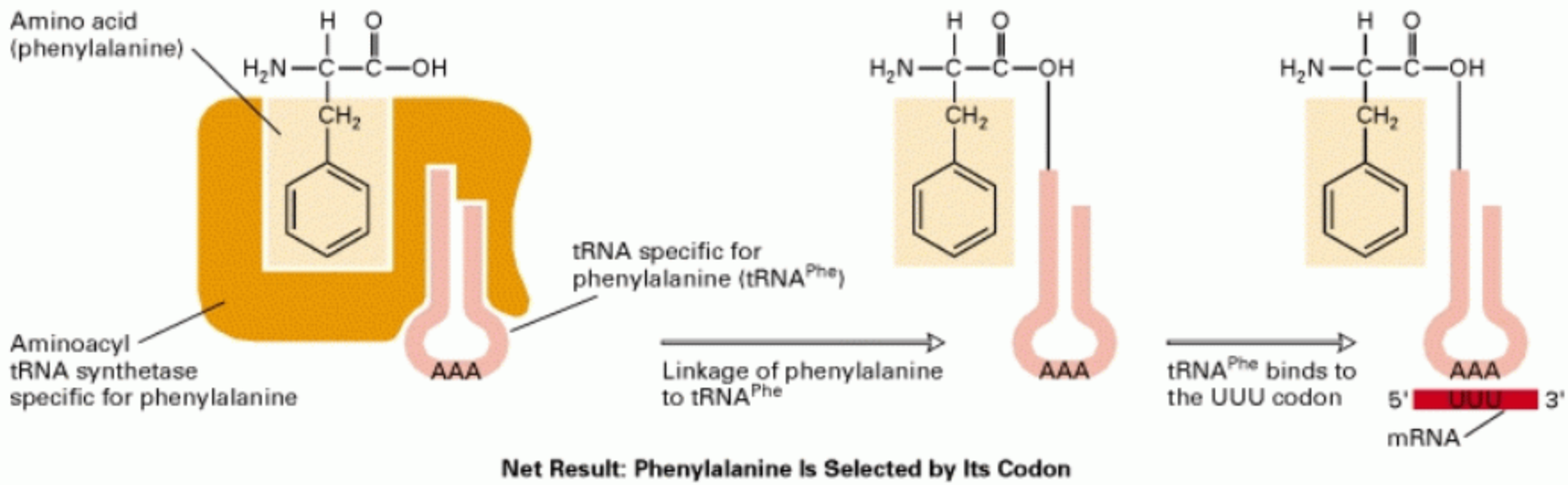
Translation

1st base

2nd base	U		C		A		G		3rd base
U	UUU	Phenylalanine	UCU	Serine	UAU	Tyrosine	UGU	Cysteine	U
	UUC	Phenylalanine	UCC	Serine	UAC	Tyrosine	UGC	Cysteine	C
	UUA	Leucine	UCA	Serine	UAA	Stop	UGA	Stop	A
	UUG	Leucine	UCG	Serine	UAG	Stop	UGG	Tryptophan	G
C	CUU	Leucine	CCU	Proline	CAU	Histidine	CGU	Arginine	U
	CUC	Leucine	CCC	Proline	CAC	Histidine	CGC	Arginine	C
	CUA	Leucine	CCA	Proline	CAA	Glutamine	CGA	Arginine	A
	CUG	Leucine	CCG	Proline	CAG	Glutamine	CGG	Arginine	G
A	AUU	Isoleucine	ACU	Threonine	AAU	Asparagine	AGU	Serine	U
	AUC	Isoleucine	ACC	Threonine	AAC	Asparagine	AGC	Serine	C
	AUA	Isoleucine	ACA	Threonine	AAA	Lysine	AGA	Arginine	A
	AUG	Methionine (Start)	ACG	Threonine	AAG	Lysine	AGG	Arginine	G
G	GUU	Valine	GCU	Alanine	GAU	Aspartic Acid	GGU	Glycine	U
	GUC	Valine	GCC	Alanine	GAC	Aspartic Acid	GGC	Glycine	C
	GUA	Valine	GCA	Alanine	GAA	Glutamic Acid	GGA	Glycine	A
	GUG	Valine	GCG	Alanine	GAG	Glutamic Acid	GGG	Glycine	G

Nonpolar, aliphatic Polar, uncharged Aromatic Positively charged Negatively charged

Translation



[Lodish, Burk, Zipurski, 2000]

Translation

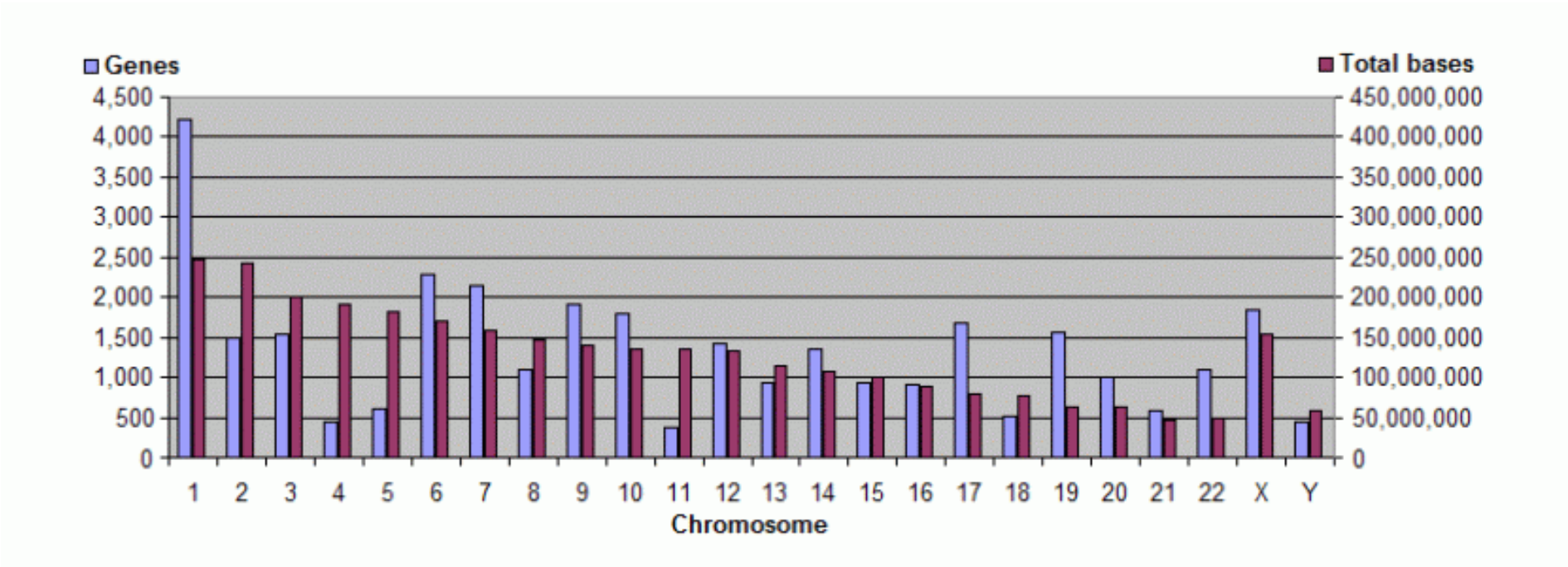
- Amino-acids are bonded through a covalent peptide bond
- The carboxyl group of one molecule reacts with the amino group of the other molecule, thereby releasing a molecule of water (H_2O)

Some Numbers

Human DNA has:

- 3 billion base pairs
- The length of DNA in a cell is 1m!
- This is packed into a nucleus of 3 – 10 microns
- Each chromosome (46 in all) is about 2 cm on average.

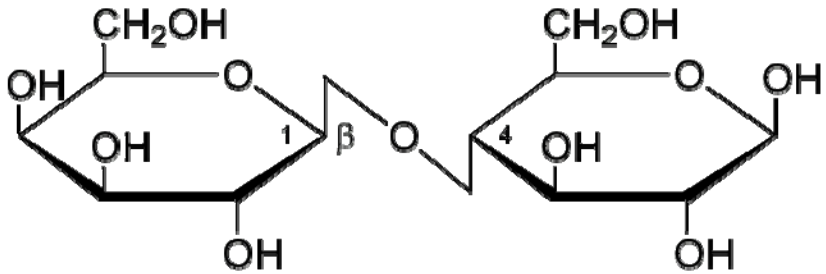
Some Numbers



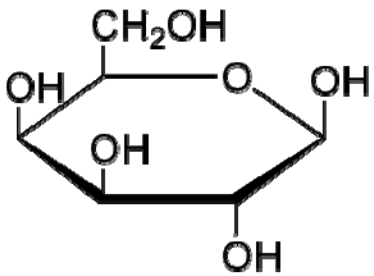
Putting it all Together

- Models for Lactose Intolerance
  - Roughly 80% of people (about 4B) have varying levels of lactose intolerance
  - These people are unable to break down lactose to smaller sugars
  - Lactose passes into the intestines, where it is broken down by the resident bacteria
  - This results in acute intestinal discomfort

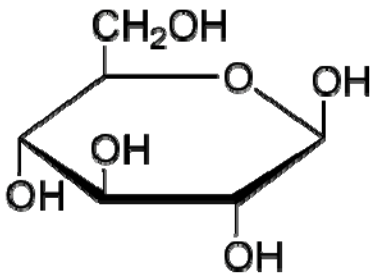
Models for Lactose Intolerance



Lactose



D-galactose



D-glucose

- beta-galactosidase (LacZ), an intracellular enzyme (protein) cleaves the disaccharide lactose into glucose and galactose
- These smaller molecules can be metabolized within the cell.

Models for Lactose Intolerance

- Beta-galactosidase is expressed by the lacZ gene
  - lacZ is part of the lac operon, comprised of lacZ, lacY, and lacA genes in E. Coli
  - lacY encodes #-galactoside permease (LacY), a membrane-bound transport protein that pumps lactose into the cell.
  - lacA encodes #-galactoside transacetylase (LacA), an enzyme that transfers an acetyl group from acetyl-CoA to beta-galactosides.
- So, should beta-galactosidase be expressed all the time? Would this be an effective use of limited cellular resources?

Models for Lactose Intolerance

- It was experimentally observed that when the bacteria is starved of lactose, very low levels of beta-galactosides are observed in the cell.
- Conversely, when the level of glucose in the cell is low, high levels of beta-galactosides are observed.
- This must imply other forms of control on the expression of genes!

Models for Lactose Intolerance

- Negative control: When levels of lactose are low, lacZ expression should be supressed.
- Negative control is affected by the lacI gene (a few hundred basepairs upstream from the lac operon).
- Recall that the ribosome is responsible for transcription.
- In E.Coli, this ribosome consists of a complex of five proteins.
- One of these proteins recognizes a promoter region for the lac operon, causing the ribosome to bind to the DNA and initiate transcrption.

Models for Lactose Intolerance

- Negative control is exerted by the lacI gene (few hundred bases upstream of the lac operon).
- lacI constantly (and in very low concentrations) expresses the lac repressor protein.
- lac repressor preferentially binds to the promoter site for the lac operon, preventing the RNAP from binding and transcribing the lac operon.
- In this way, no beta-galactoside is expressed.

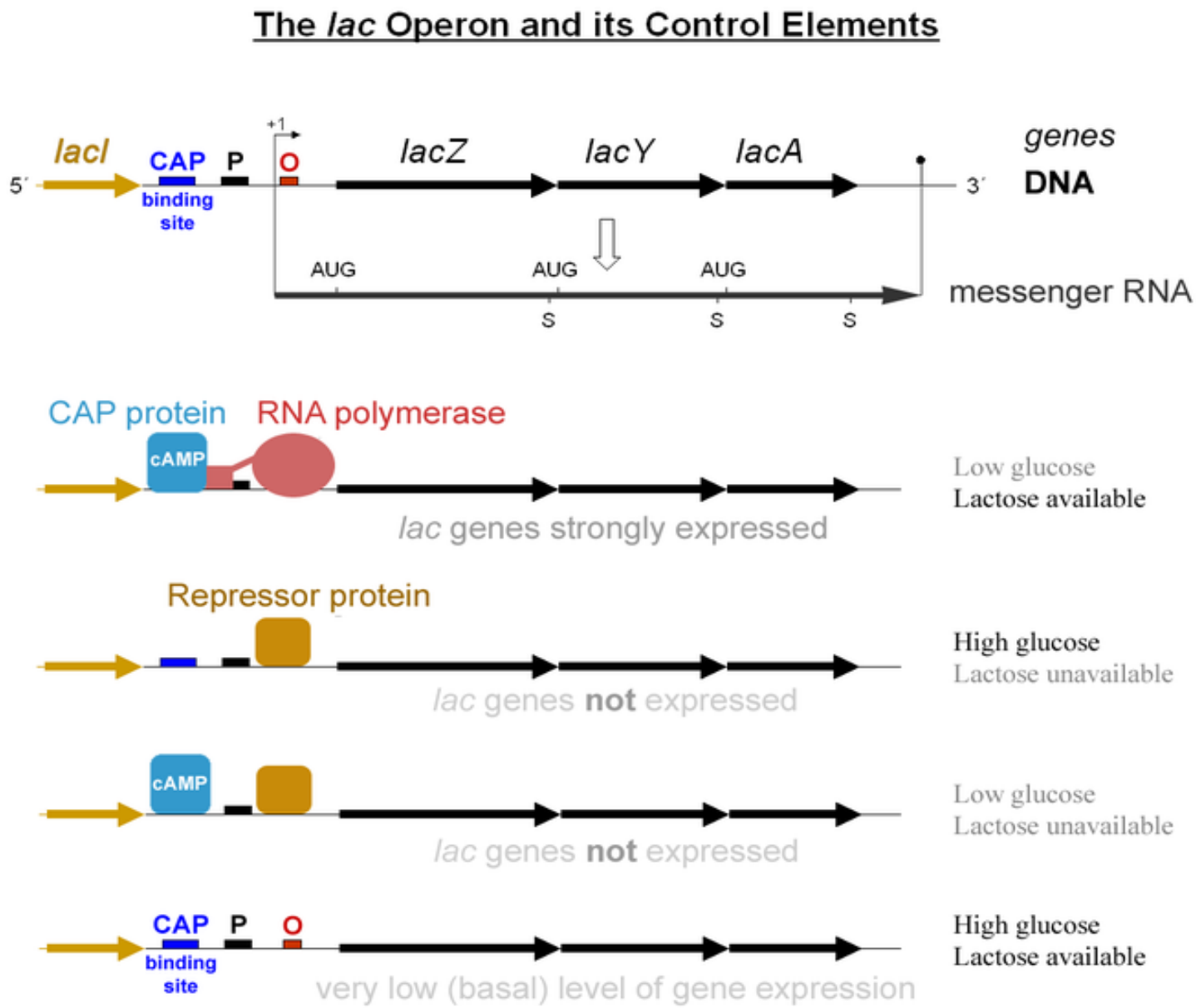
Models for Lactose Intolerance

- When lactose is present, it binds to the lac repressor protein, causing a change in its conformation
- This prevents it from binding to the promoter site for the lac operon
- This results in the RNAP binding and transcribing the lac operon!

Models for Lactose Intolerance

- Positive Feedback: when glucose levels in the cell are low, we want to generate significant amounts of beta-galactosidase, so as to be able to break down any amount of lactose present.
- Cyclic adenosine monophosphate (cAMP) is a signal molecule whose prevalence is inversely proportional to that of glucose.
- cAMP binds to the the Catabolite activator protein (CAP)
- This allows CAP to bind to the CAP binding site upstream of the promoter for the lac operon.
- This in turn facilitates the RNAP to bind and transcribe the operon.

Models for Lactose Intolerance



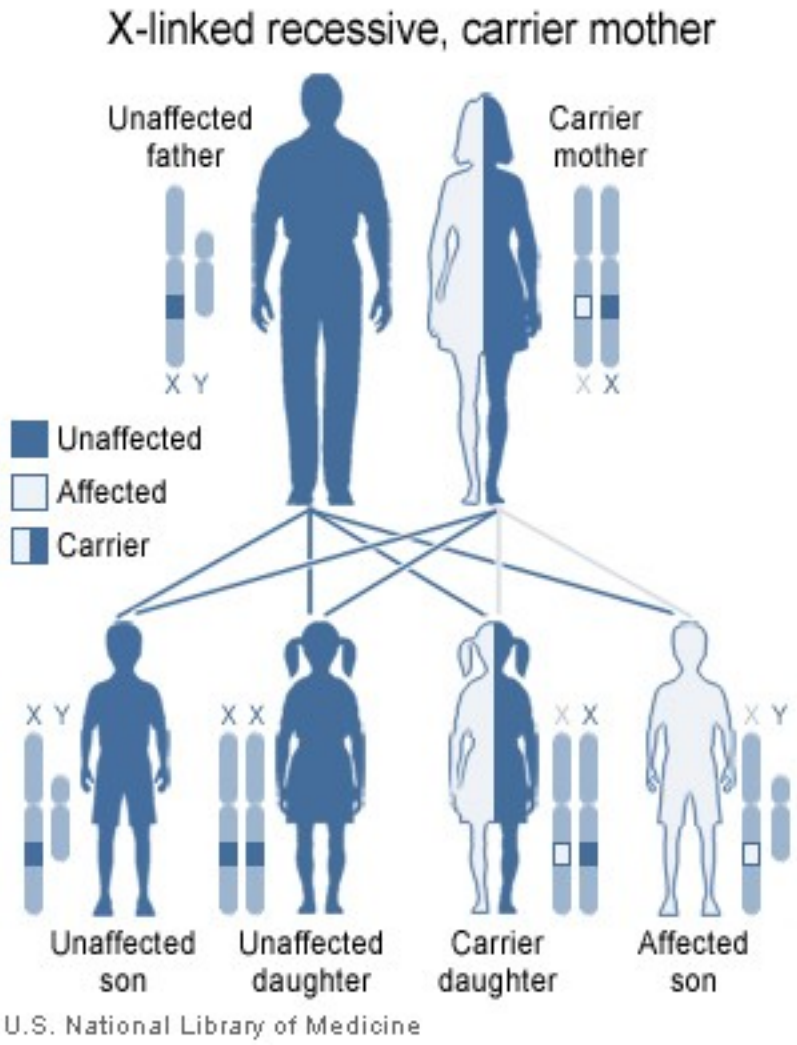
Models for Lactose Intolerance

- So what causes lactose intolerance anyway?
  - The lack of beta-galactoside in cells.
- What does one do about it?
  - Take lactase enzymes (beta-galactosides) before lactose!

One more example: Haemophilia

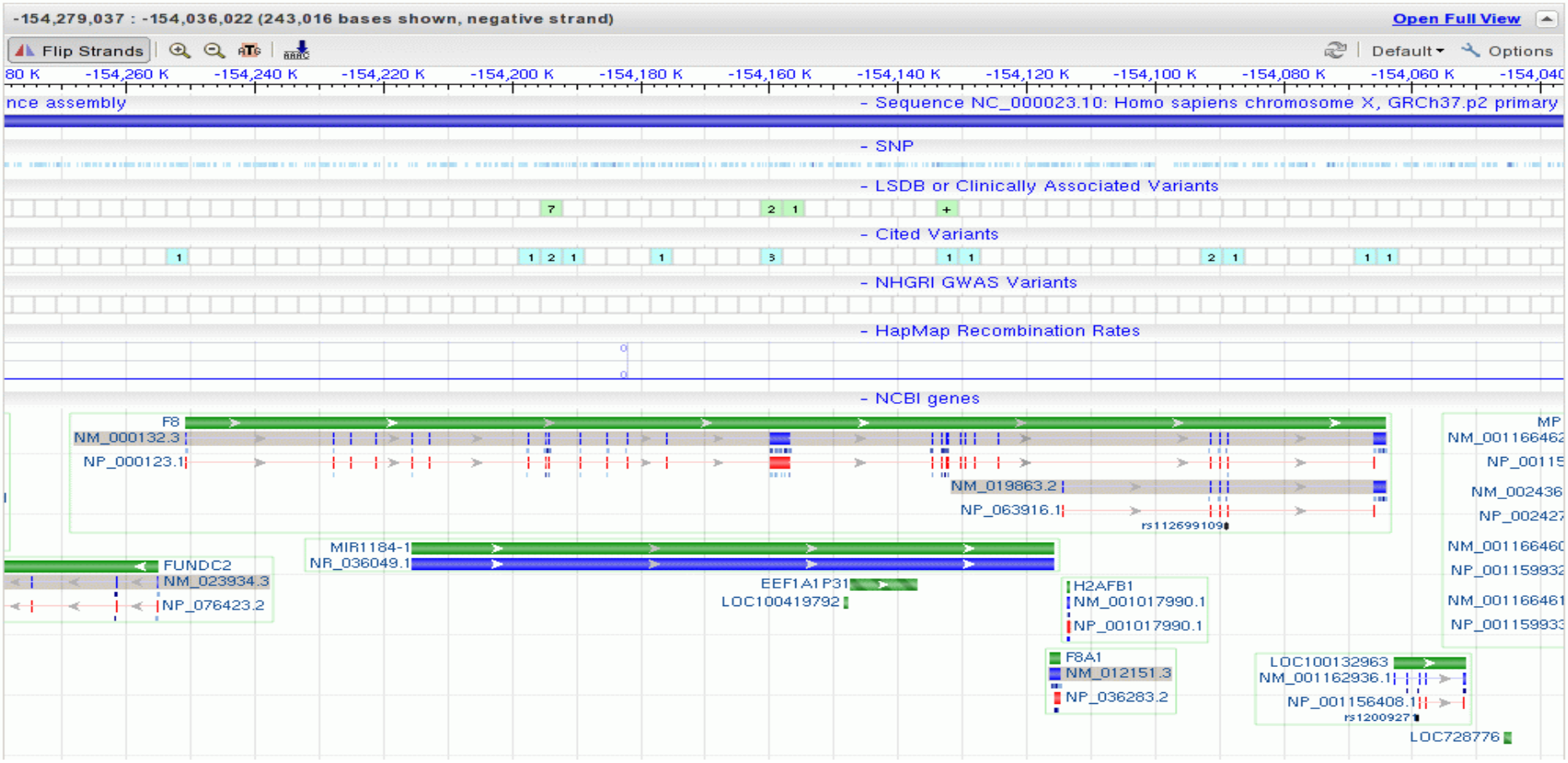
- Genetic disorder that impairs blood coagulation
- Haemophilia A is the most common form, effecting 1 in 5000 to 10000 male births
- Haemophilia A is a recessive X-linked genetic disorder involving a lack of functional clotting Factor VIII and represents 80% of haemophilia cases.
- In females, since there are two X chromosomes, a defect on one of the chromosomes may not manifest itself in the form of a disease.

Haemophilia-A



Haemophilia-A

- Haemophilia-A is caused by clotting factor VIII deficiency. Factor VIII is encoded by the F8 gene.



Haemophilia-A

- Through a sequence of complexes and co-factors, Factor VIII results in the generation of fibrin, which causes the clot.
- Factor VIII, concentrated from plasma is given to haemophiliacs to restore haemostasis.

What we have learnt thus far?

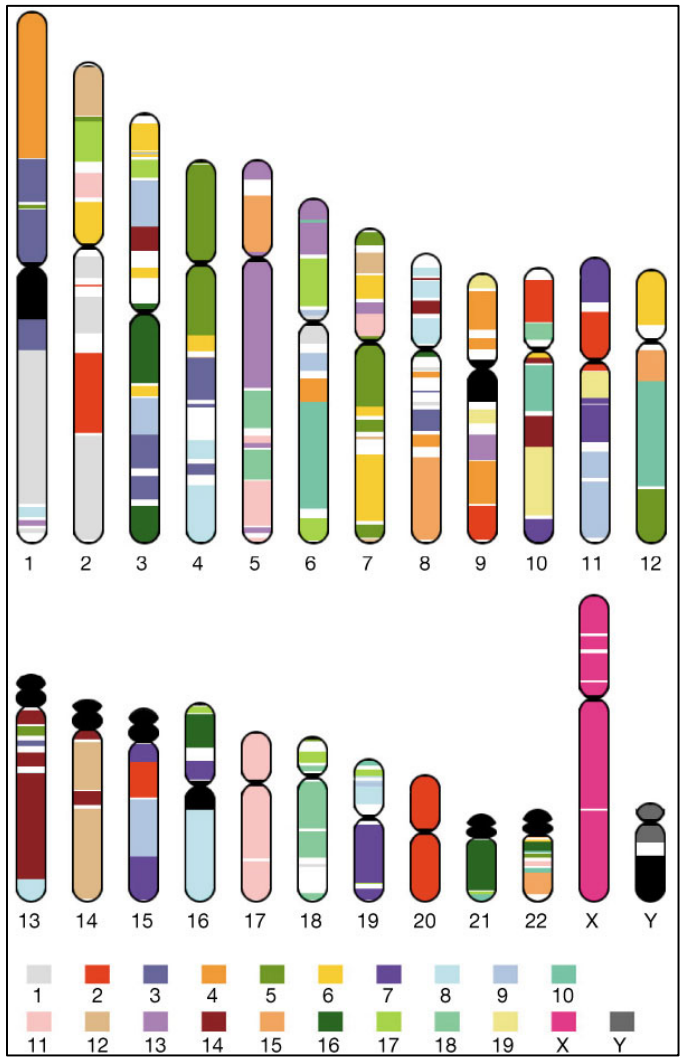
- Central dogma of biology
- The role of sequences (DNA, RNA, proteins)
- The ability to translate across DNA, RNA, and proteins
- The role of genes and proteins in disorders/ functions.

Analyzing Sequences

Sequences: An Evolutionary Perspective

- Evolution occurs through a set of modifications to the DNA
- These modifications include point mutations, insertions, deletions, and rearrangements
- Seemingly diverse species (say mice and humans) share significant similarity (80-90%) in their genes
- The locations of genes may themselves be scrambled

Chromosomal Rearrangements



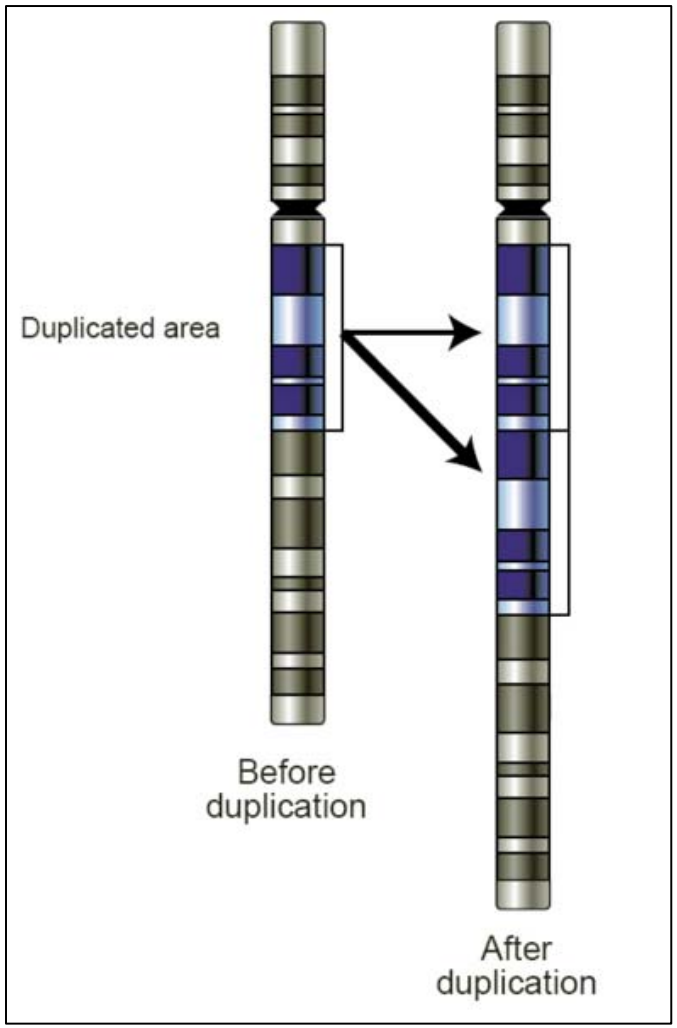
Mouse genome mappings to human genome.

Mouse Genome

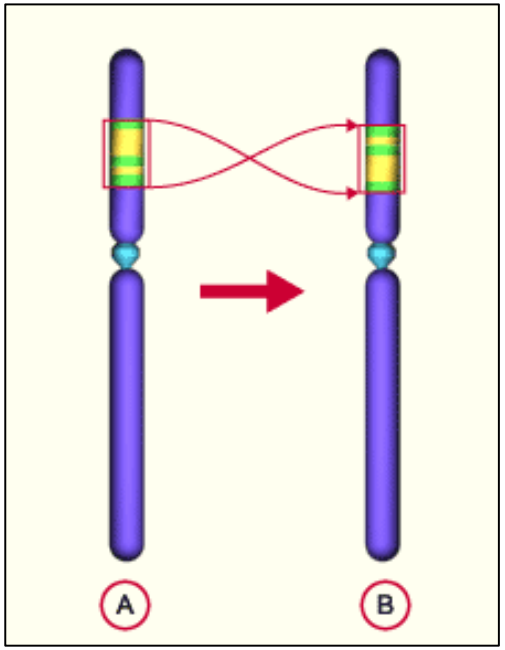
- Mouse genome 2.5 Gb vs human 2.9 Gb
- Can identify regions of synteny between mouse and human for 90% of genome.
- Both genomes have ~30,000 genes
- 99% of mouse genes have a human homolog (and vice versa)
- Some genes appear to have evolved more quickly than random chance (immunity and reproduction).

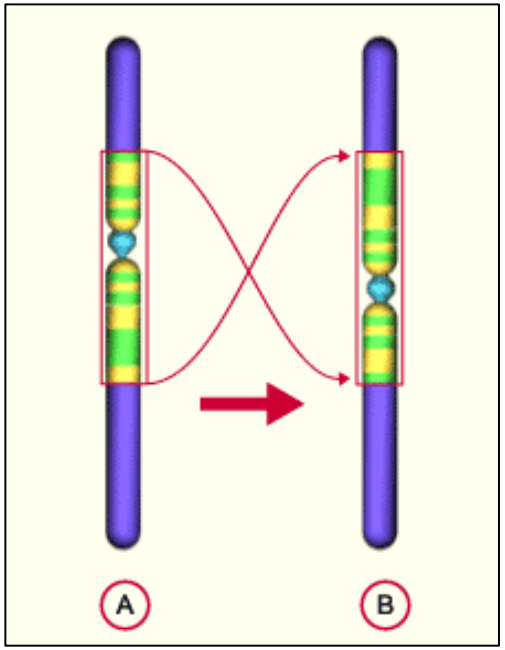
Gene Duplication

- Gene duplication has important evolutionary implications
- Duplicated genes are not subject to evolutionary pressures
- Therefore they can accumulate mutations faster (and consequently lead to specialization)



Inversions

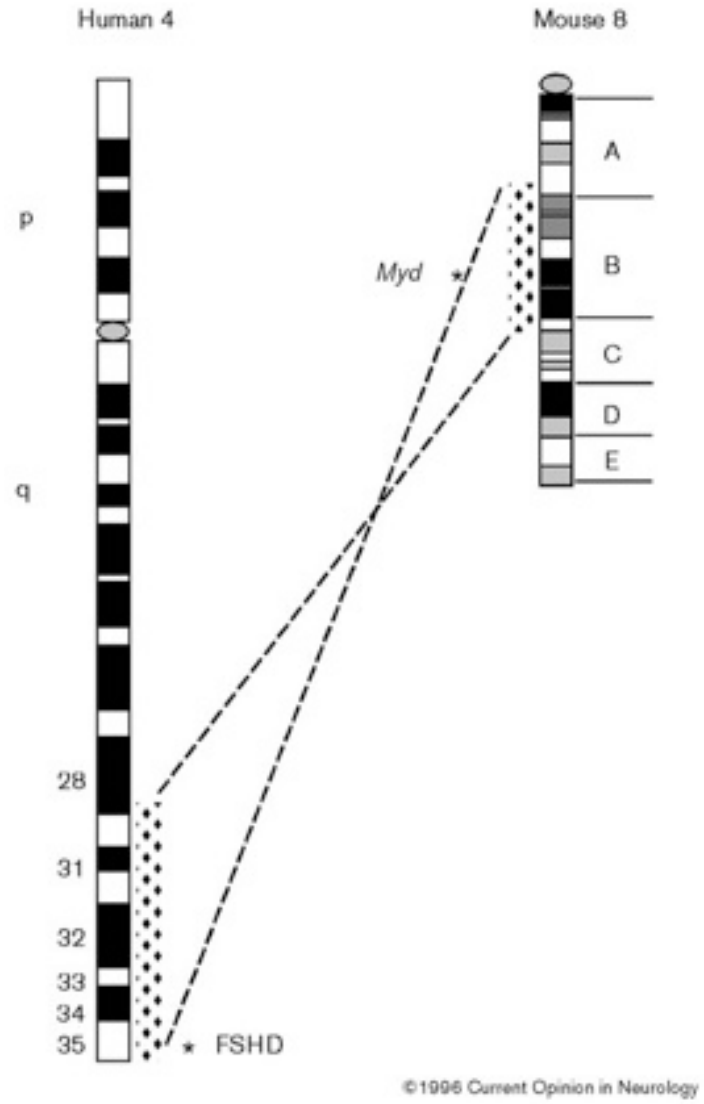




Para and pericentric inversions

Transposition

A group of conserved genes appears in a transposed fashion at a different location



Comparing Sequences

- Define distance between two sequences as the number of mutations that would result in the second string, starting from the first

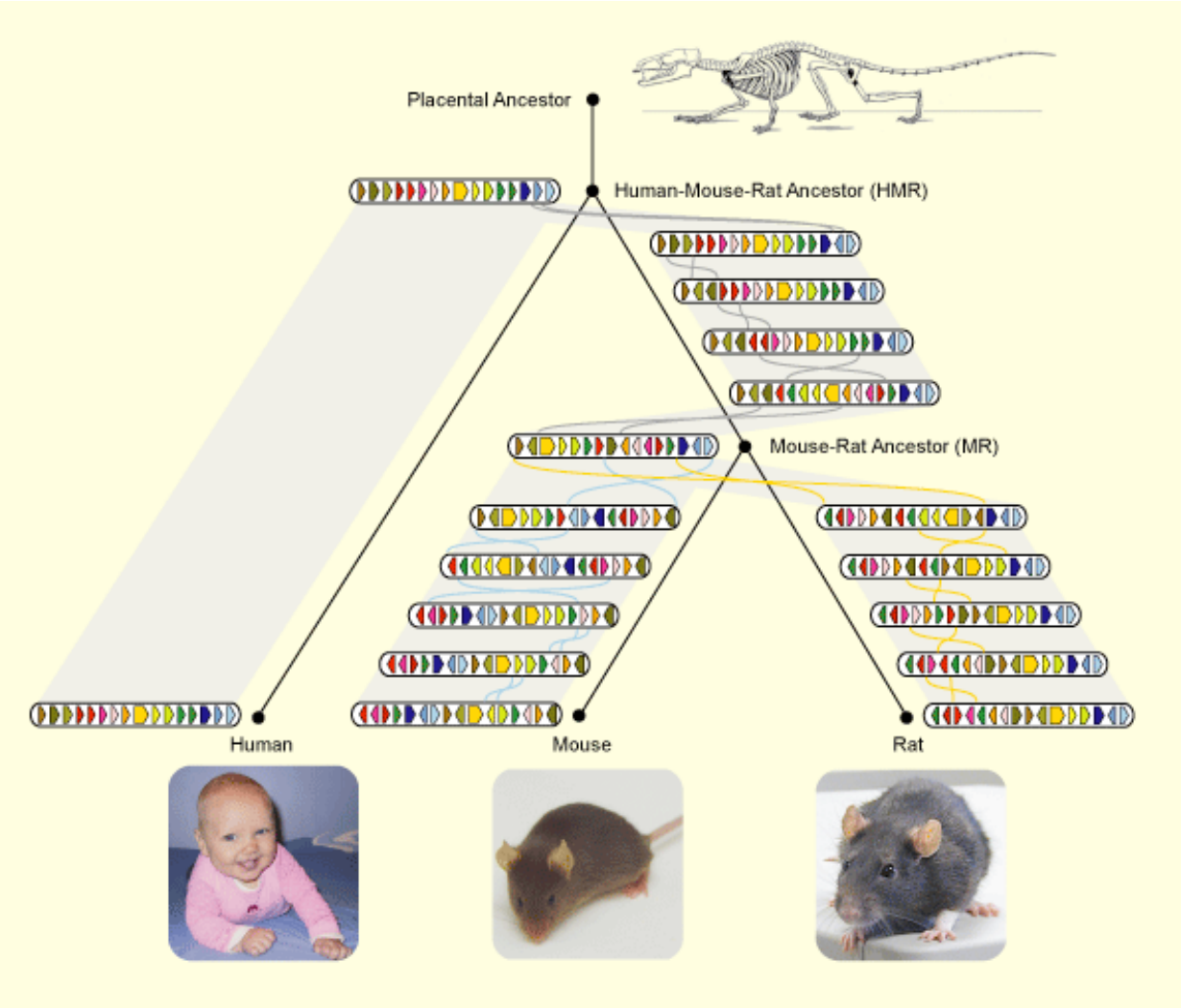
ACGGCGTGCTTTAGAACATAG

A**A**GGCGTGCTTTAGAACATAG

AAGGCGTGC**G**TTAGAACATAG

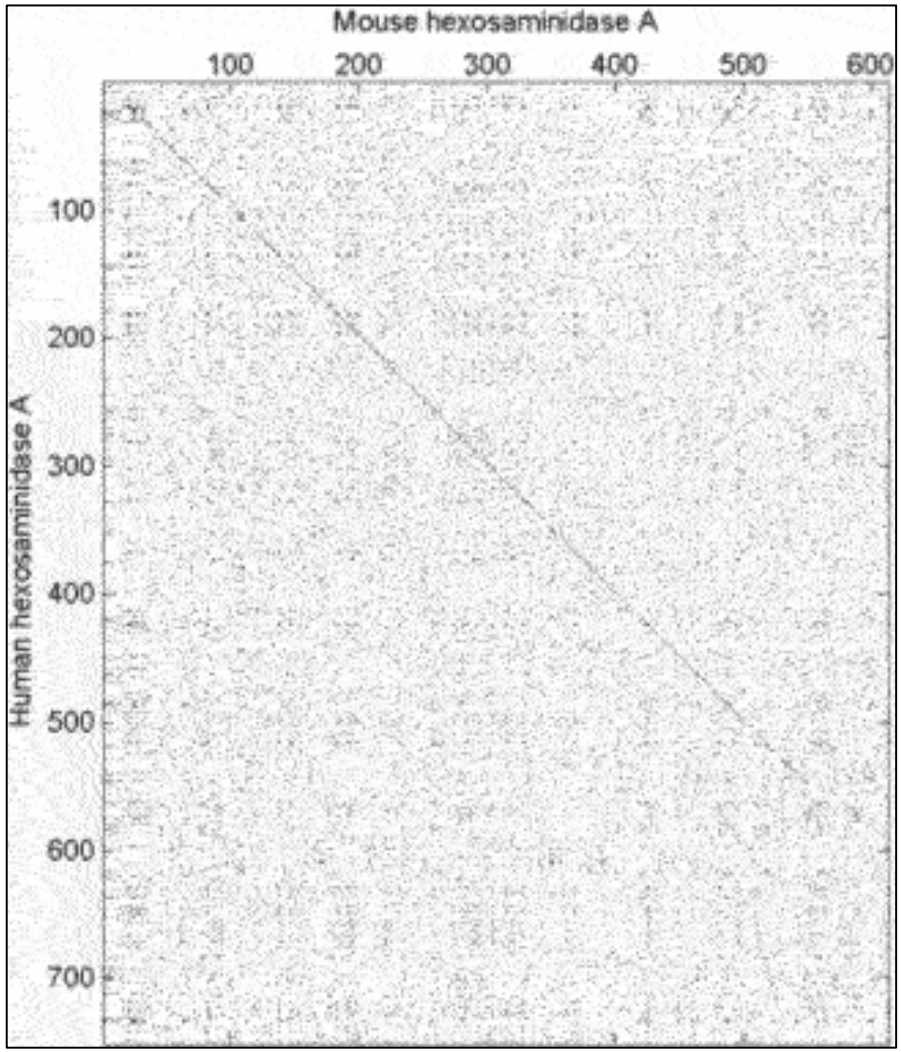
ACGGCGTGCGT**A**AG**G**ACA**A**TAG

Evolution and Edit Distances



Plotting Genome Rearrangements

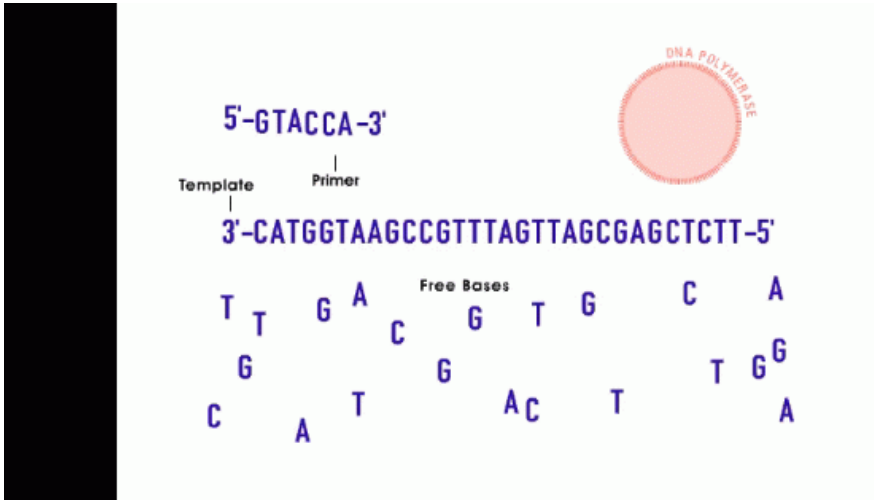
Diagonals imply direct alignment
Reverse diagonals imply inverse alignment

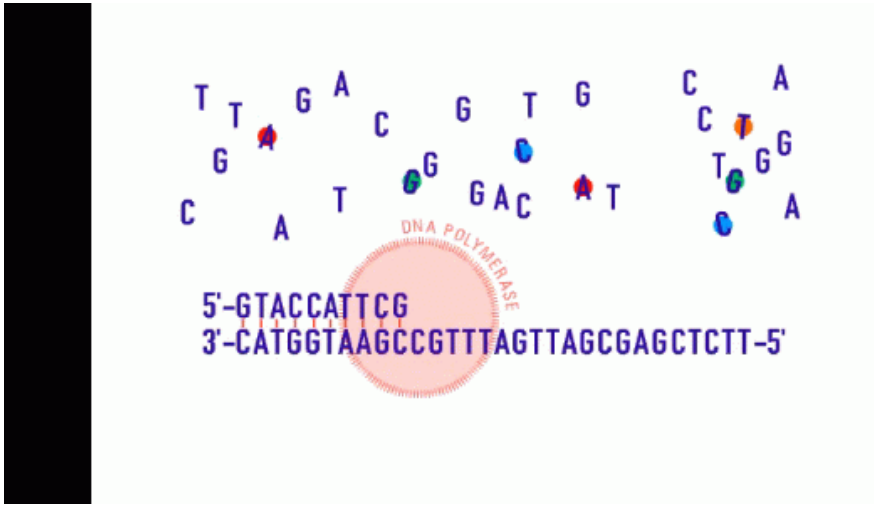


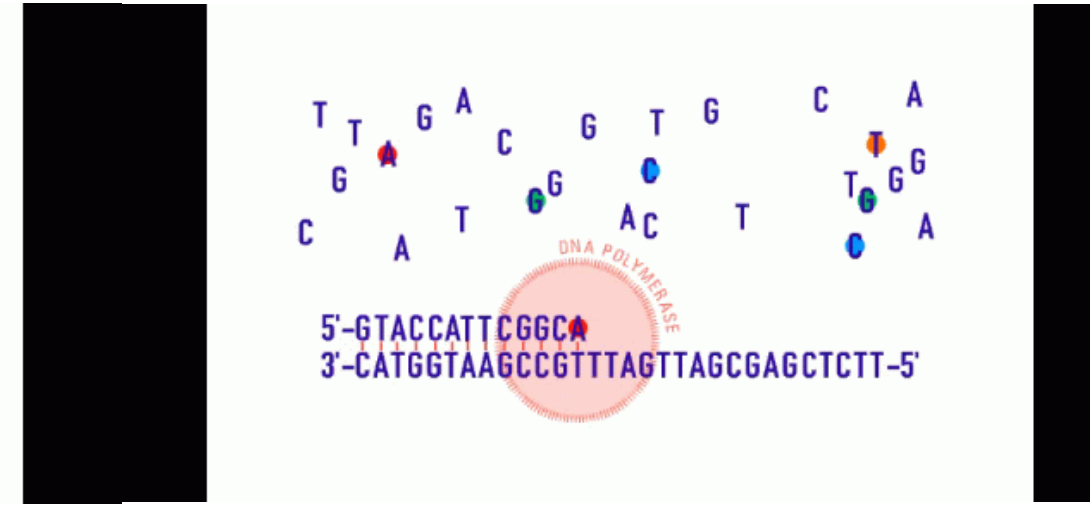
Genomic Sequences

- Sanger Sequencing
- Next-Generation Sequencing
  - Illumina Solexa
  - Helicos
  - Solid
  - Roche/454

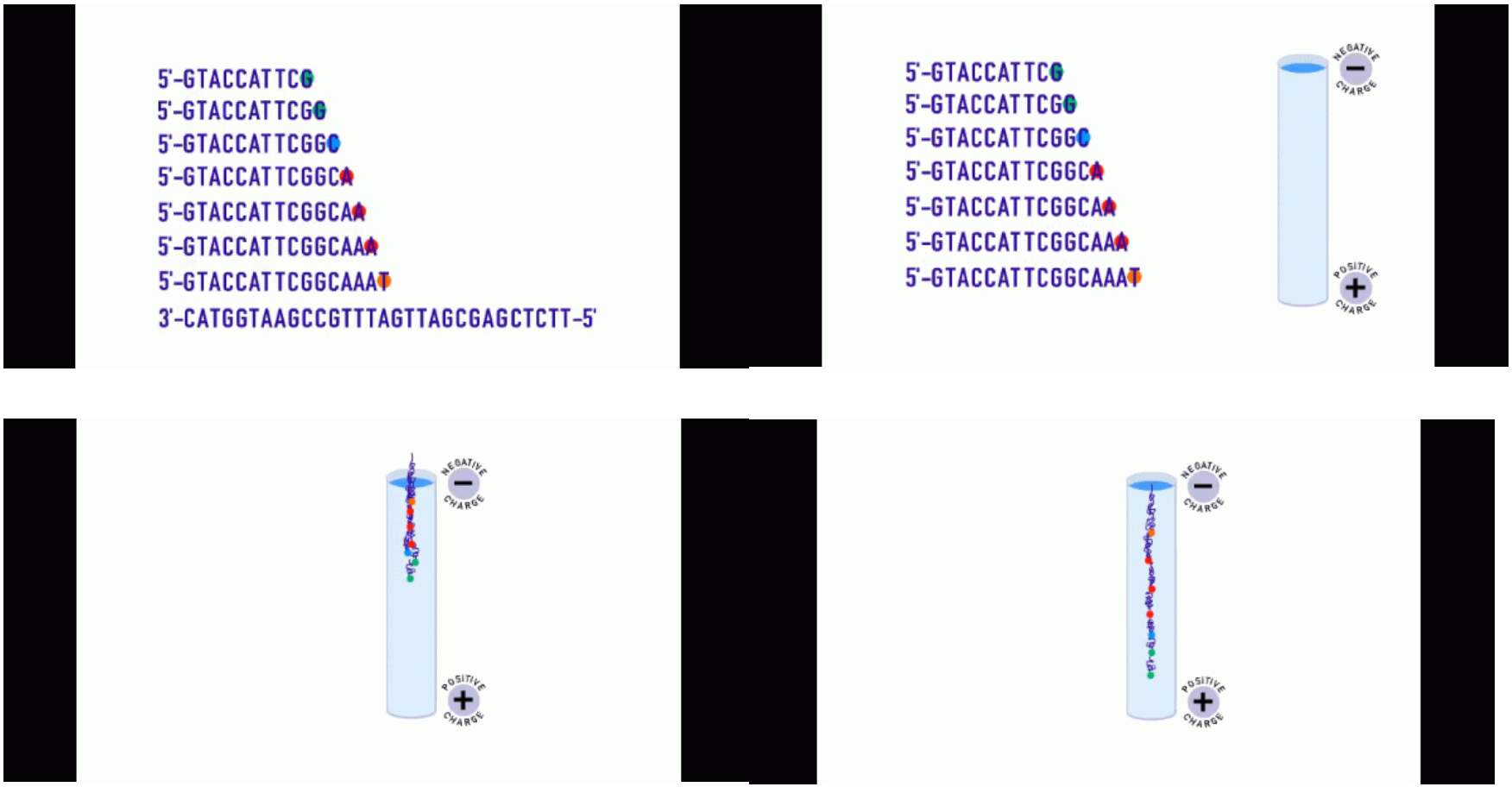
Sanger Sequencing



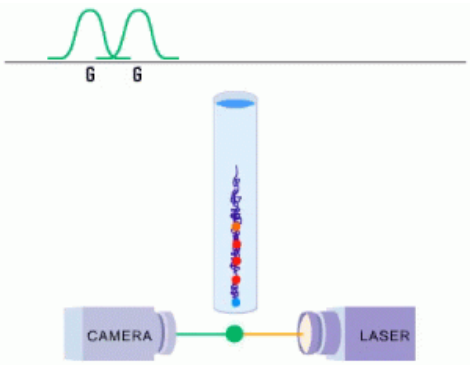


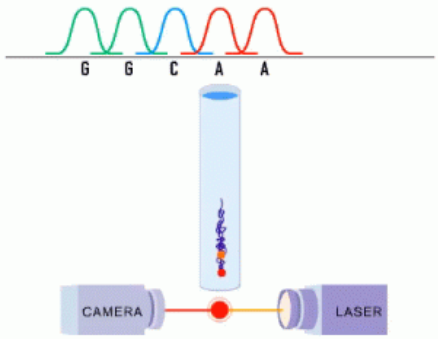


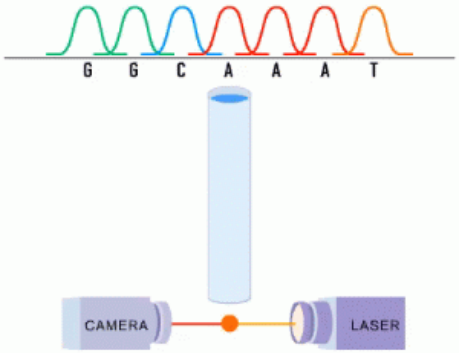
Sanger Sequencing

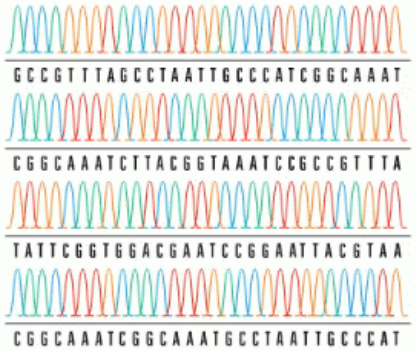


Sanger Sequencing

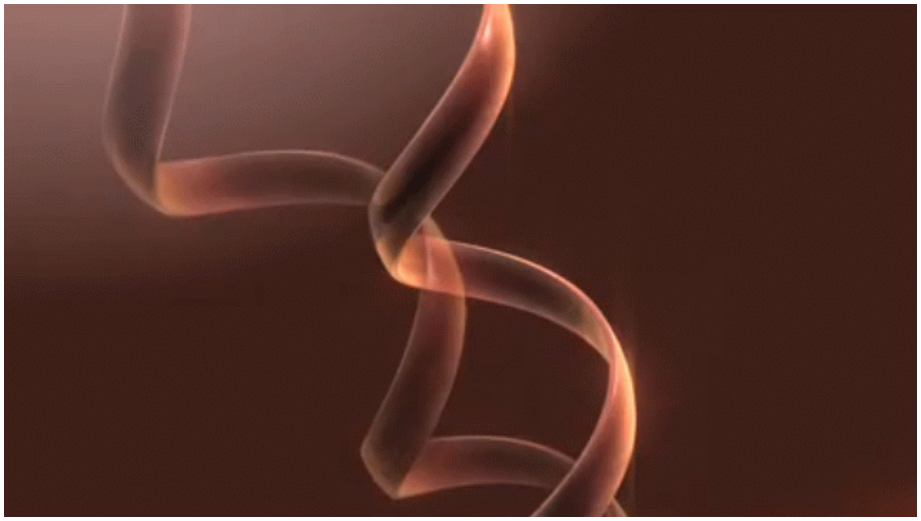


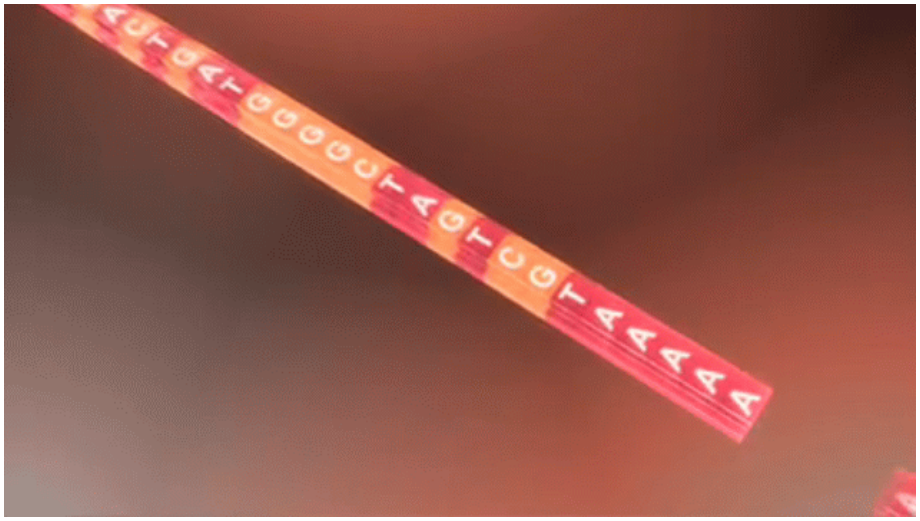




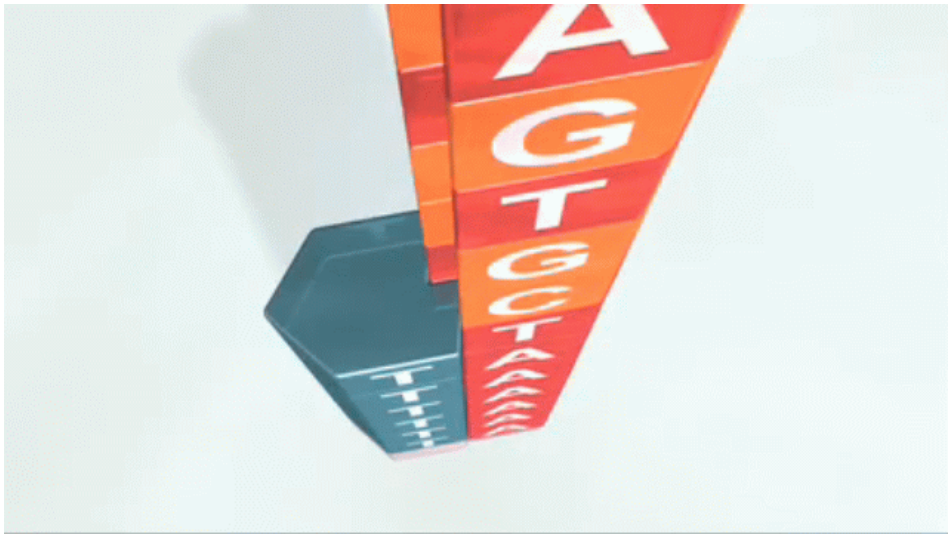


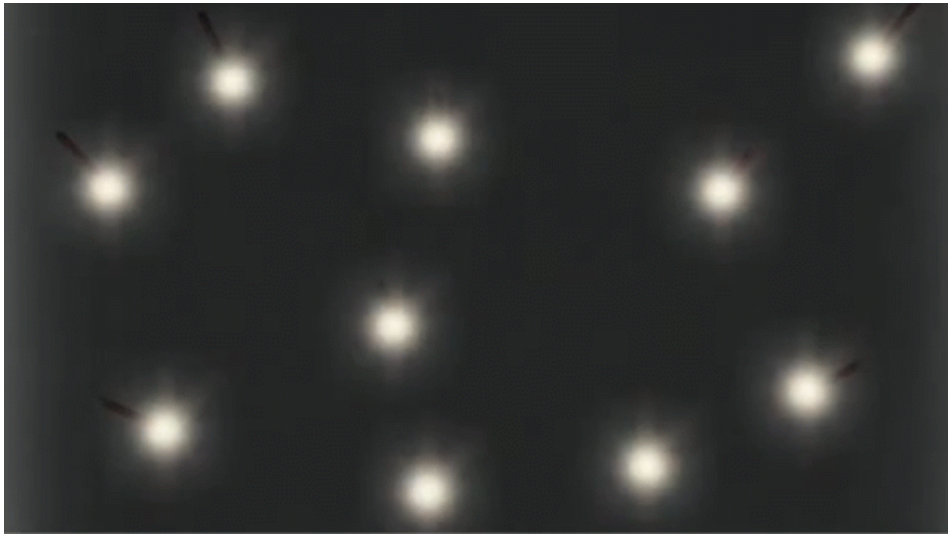
HeliScope Single Molecule Sequencer

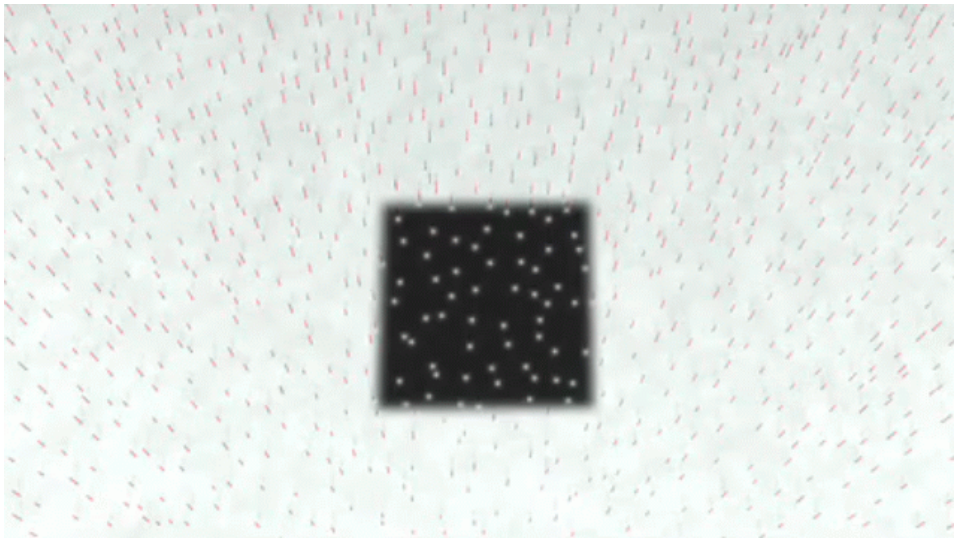




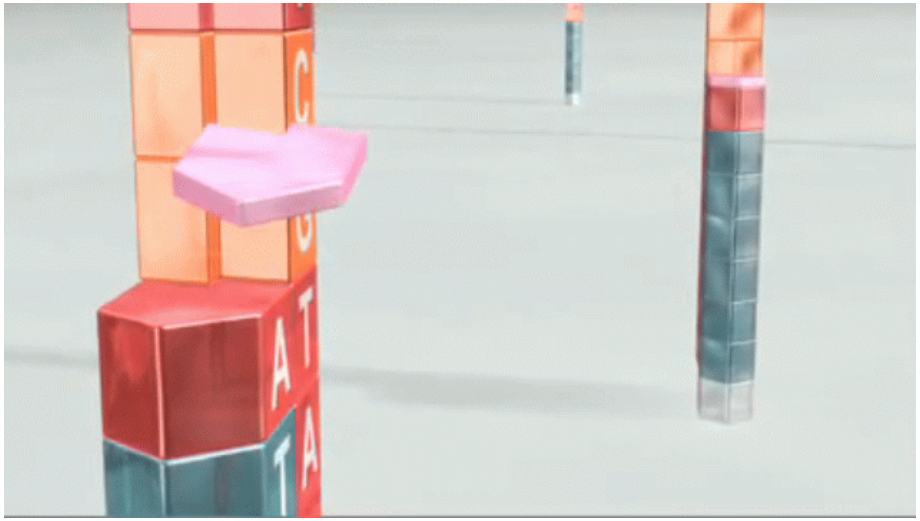
HeliScope Single Molecule Sequencer



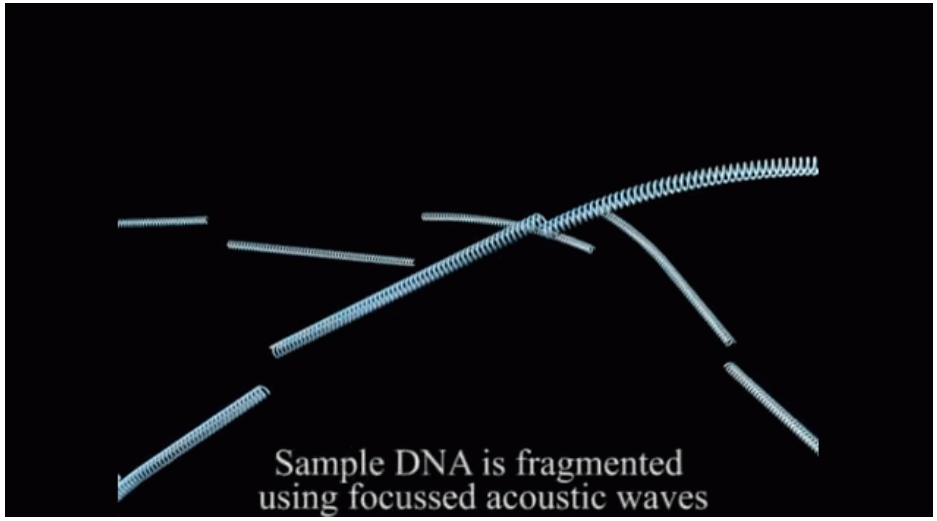


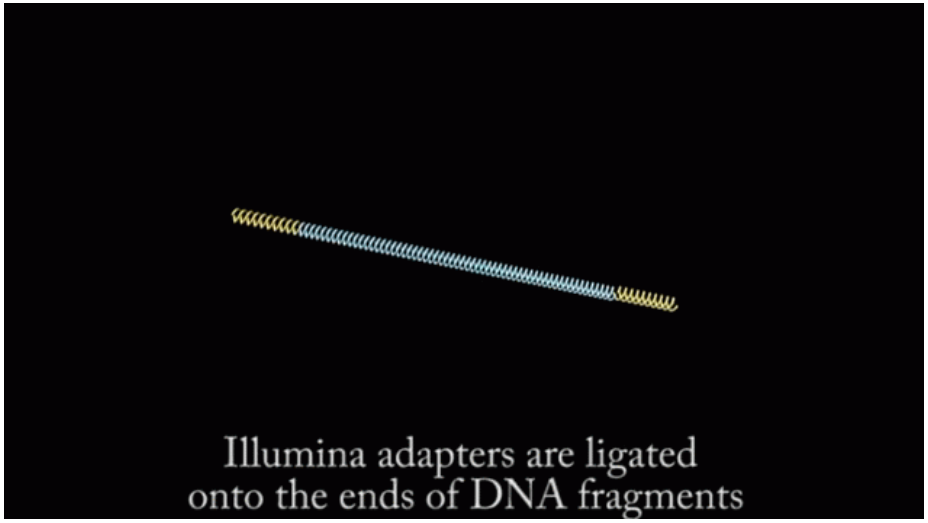


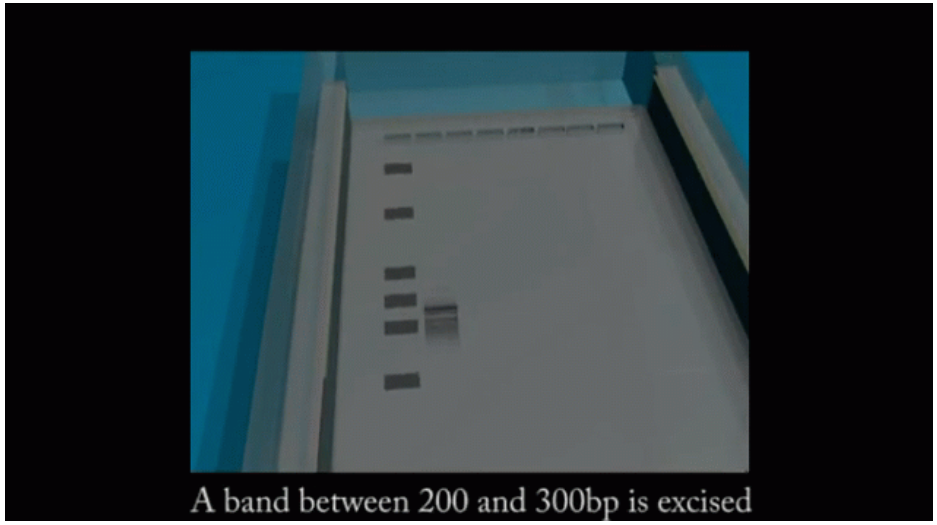
HeliScope Single Molecule Sequencer

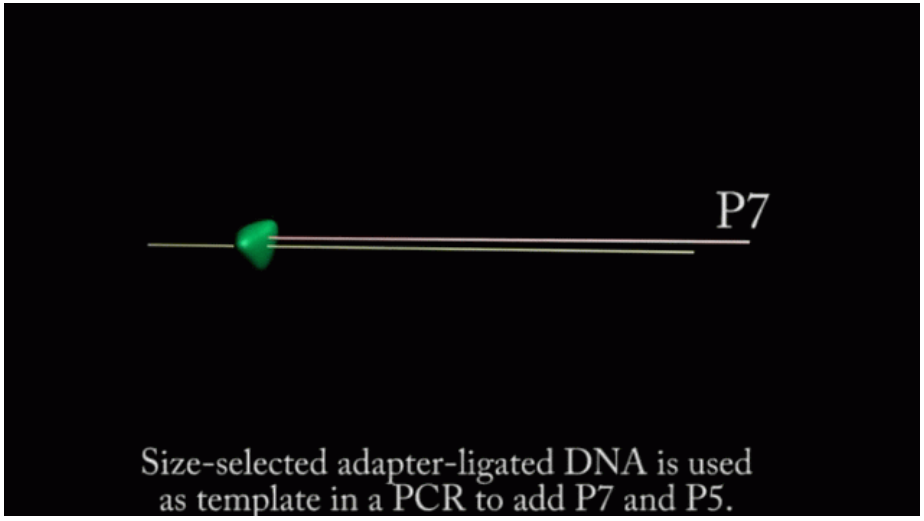


Illumina Solexa

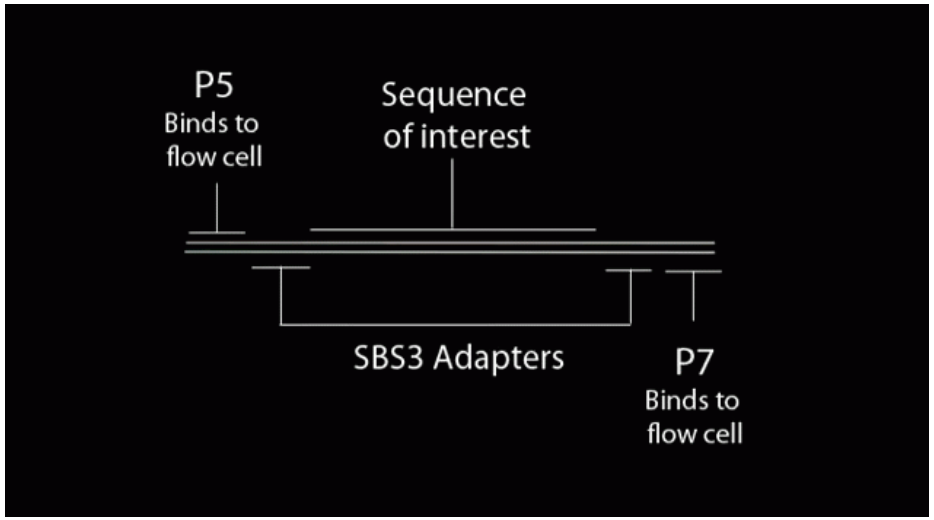


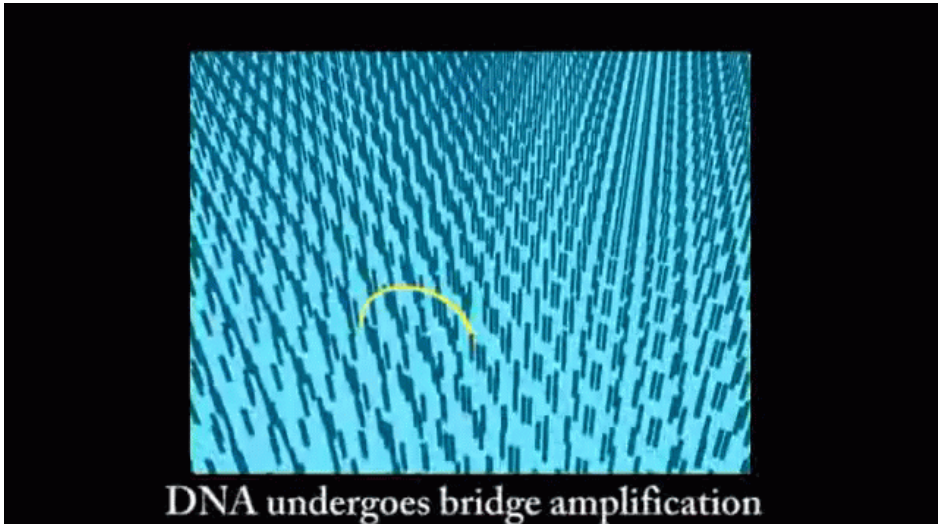


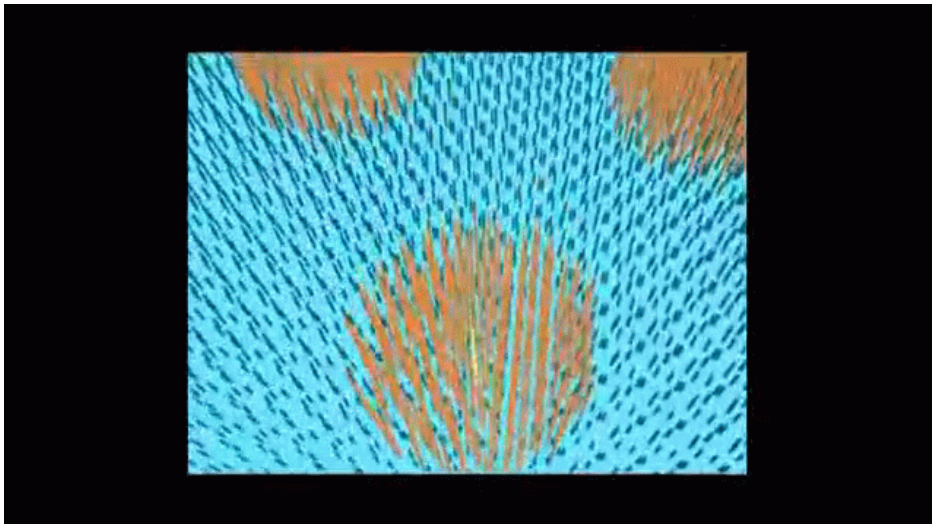


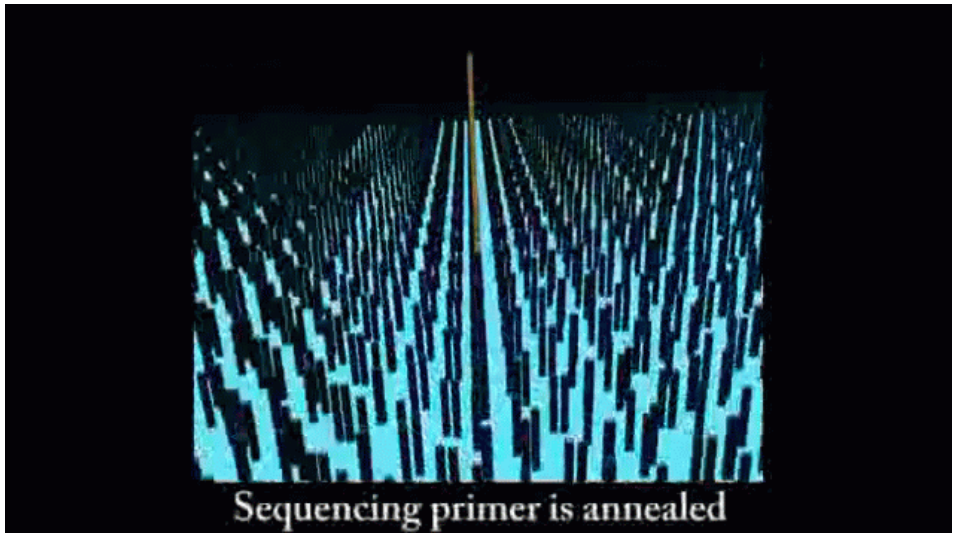


Illumina Solexa

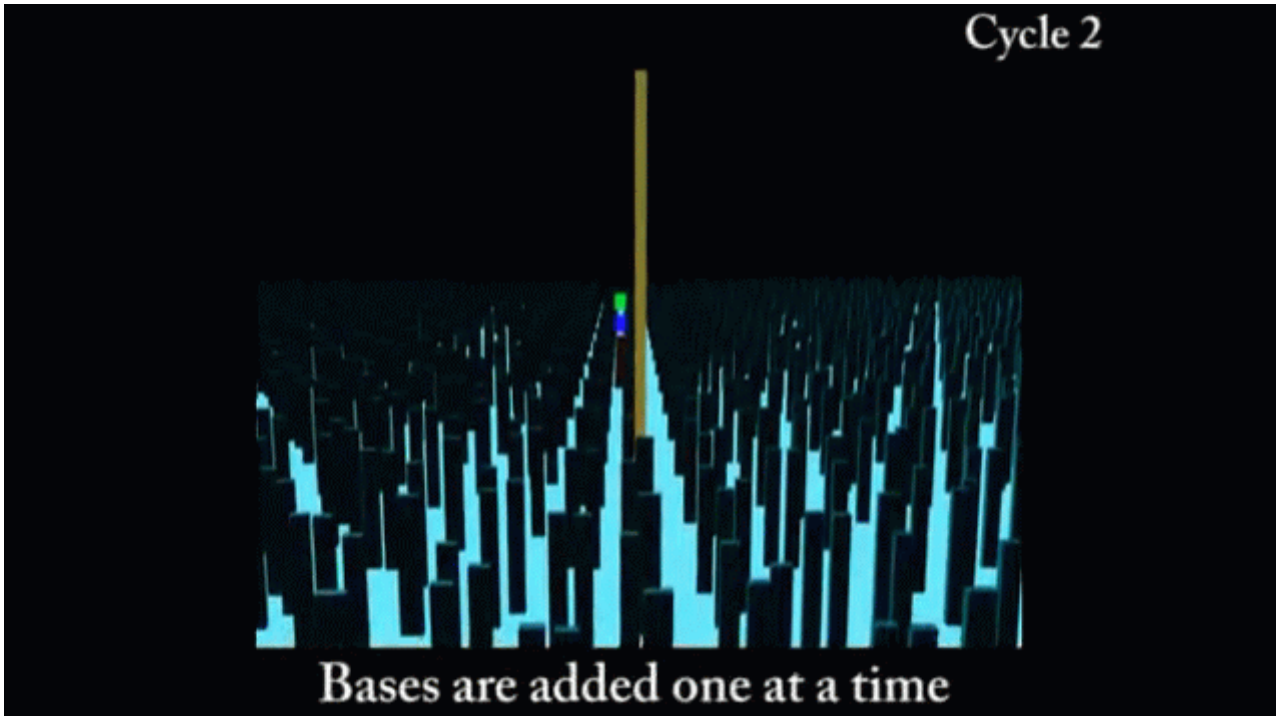








Illumina Solexa



Dealing with Reads

- From fluorescence to nucleotides (Phread)
- Error correction
- Mapping to reference genomes
- Assembly

Error Correction

- NGS reads range from 50 – 300 bps (constantly changing)
- Error rates range from 1 – 3%
- Errors are not uniformly distributed over the read
- Correcting errors is a critical step before mapping/ assembly

Error Correction

- Needed coverage on the genome
- k-mer based error correction
- Suffix Trees

Short Read Alignment

- Given a reference and a set of reads, report at least one “good” local alignment for each read if one exists
  - Approximate answer to: where in genome did read originate?

- What is “good”? For now, we concentrate on:
  - Fewer mismatches is better
  - Failing to align a low-quality base is better than failing to align a high-quality base

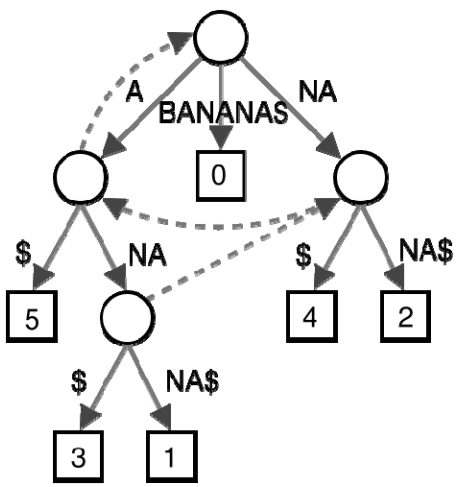
```
...TGATCATA...   better than   TGATCATA...
   GATCAA                      GAGAAT

...TGATATTA...                 TGATcaTA...
   GATcat                      GTACAT
```

Pop et al.

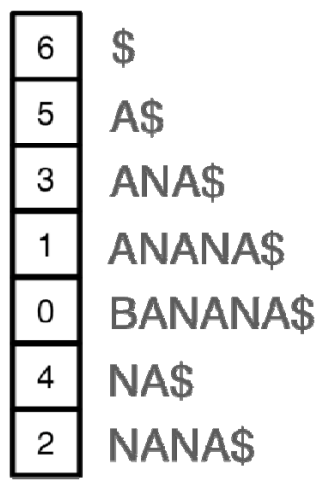
Indexing

- Genomes and reads are too large for direct approaches like dynamic programming

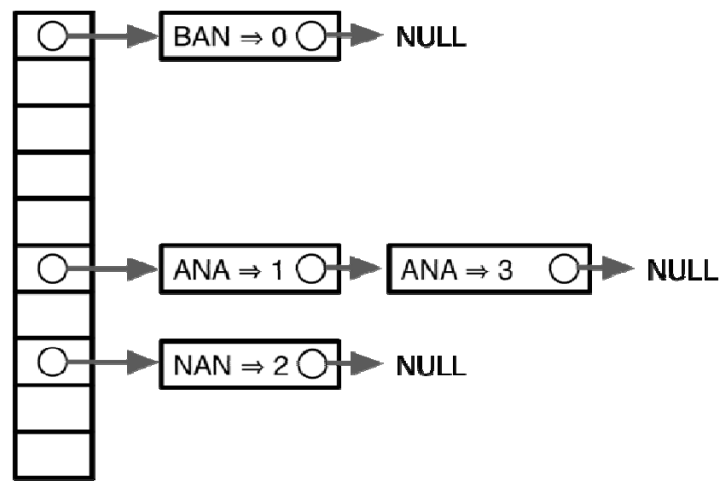
- *Indexing* is required



Suffix tree



Suffix array

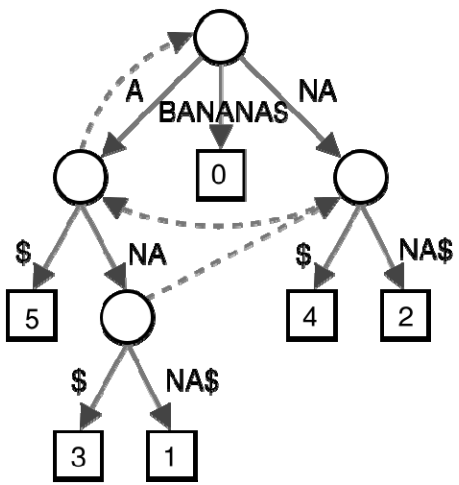


Many variants, incl. spaced seeds

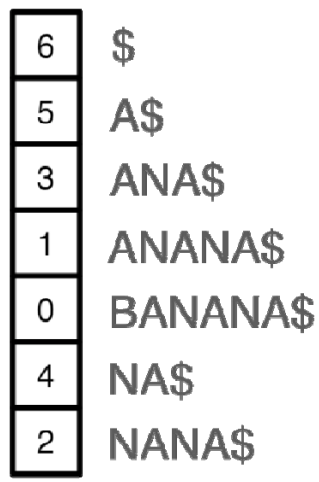
- Choice of index is key to performance

Indexing

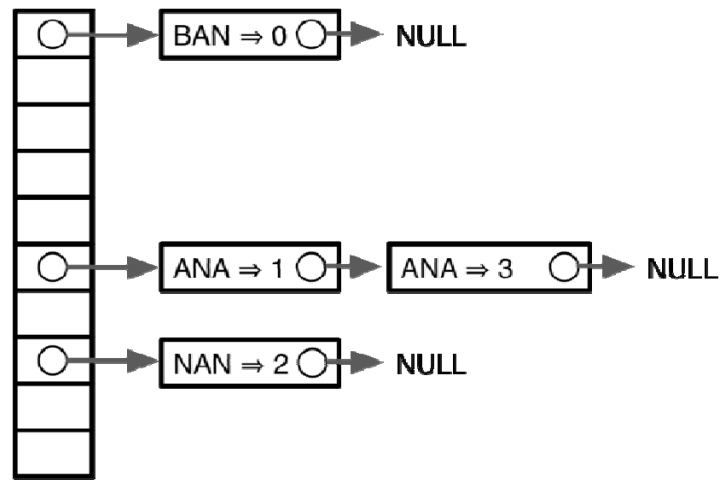
- Genome indices can be big. For human:



> 35 GBs



> 12 GBs

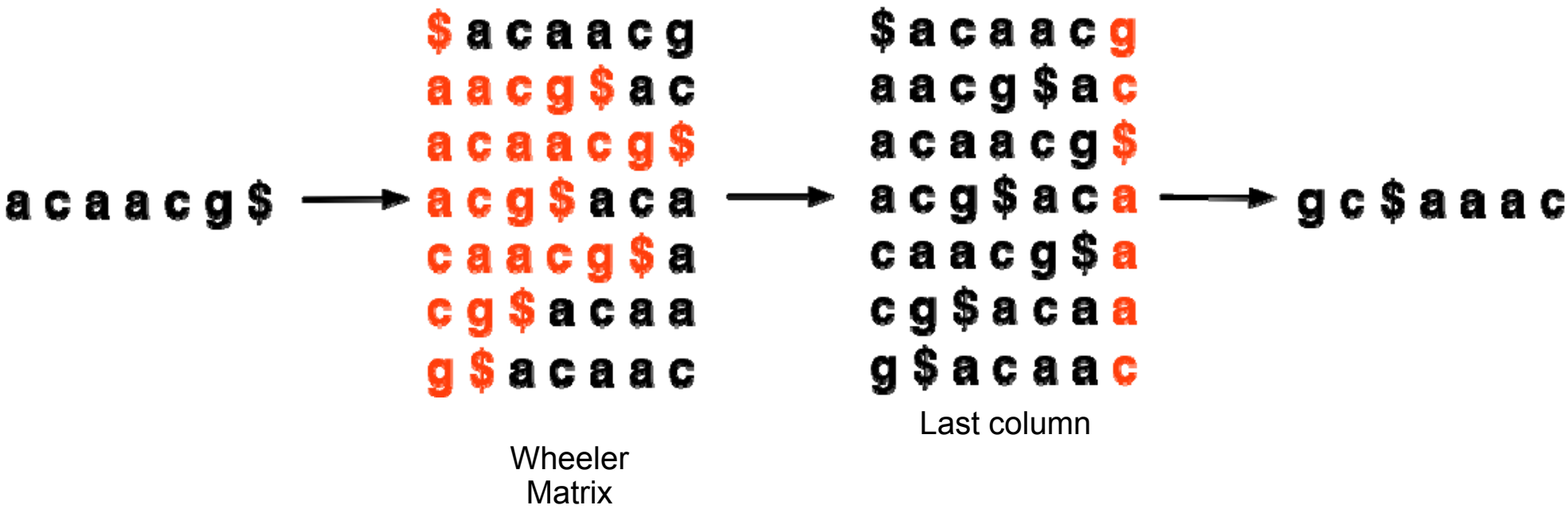


> 12 GBs

- Large indices necessitate painful compromises
  1. Require big-memory machine
  2. Use secondary storage
  3. Build new index each run
  4. Subindex and do multiple passes

Burrows-Wheeler Transform

- Reversible permutation used originally in compression

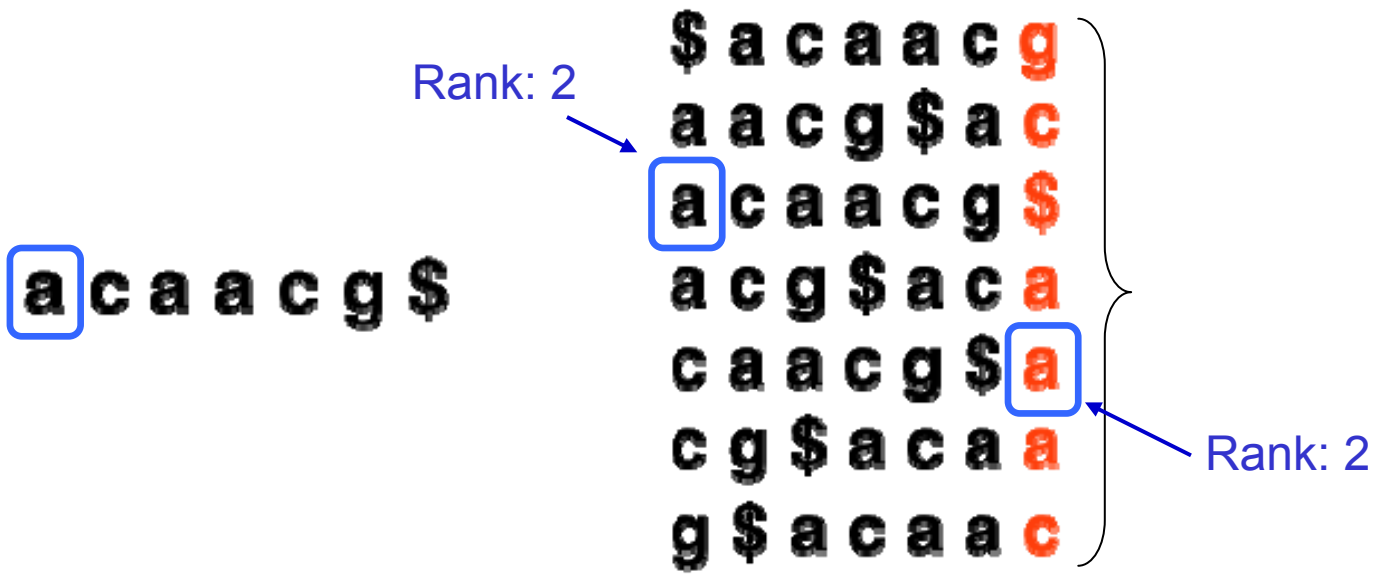


- Once BWT(T) is built, *all else shown here is discarded*
  - Matrix will be shown for illustration only

Burrows M, Wheeler DJ: A block sorting lossless data compression algorithm. *Digital Equipment Corporation, Palo Alto, CA* 1994, Technical Report 124; 1994

Burrows-Wheeler Transform

- Property that makes BWT(T) reversible is “LF Mapping”
  - i^{th} occurrence of a character in **L**ast column is same *text* occurrence as the ith occurrence in **F**irst column



Burrows Wheeler Matrix

Burrows-Wheeler Transform

- To recreate T from BWT(T), repeatedly apply rule:
  - **T** = **BWT**[**LF**(i)] + **T**; i = **LF**(i)
  - Where **LF**(i) maps row i to row whose first character corresponds to i's last per LF Mapping

- Could be called "unpermute" or "walk-left" algorithm

FM Index

- Ferragina & Manzini propose “FM Index” based on BWT

- Observed:
  - LF Mapping also allows *exact matching* within T
  - **LF**(i) can be made fast with *checkpointing*
  - …and more (see FOCS paper)

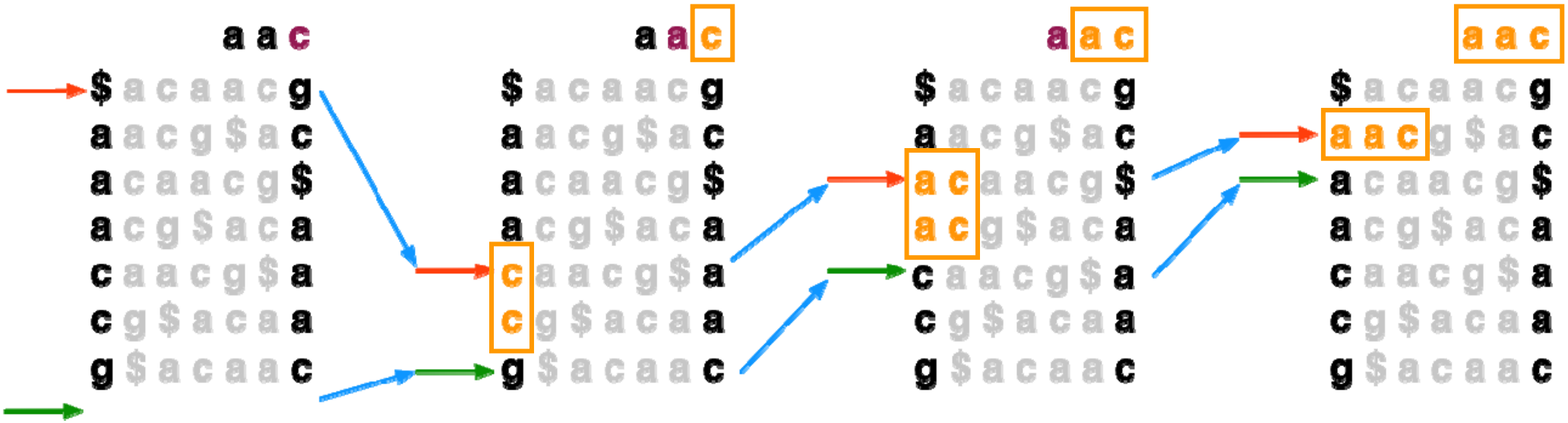
Ferragina P, Manzini G: Opportunistic data structures with applications. *FOCS. IEEE Computer Society; 2000.*

Ferragina P, Manzini G: An experimental study of an opportunistic index. *SIAM symposium on Discrete algorithms*. Washington, D.C.; 2001.

Exact Matching with FM Index

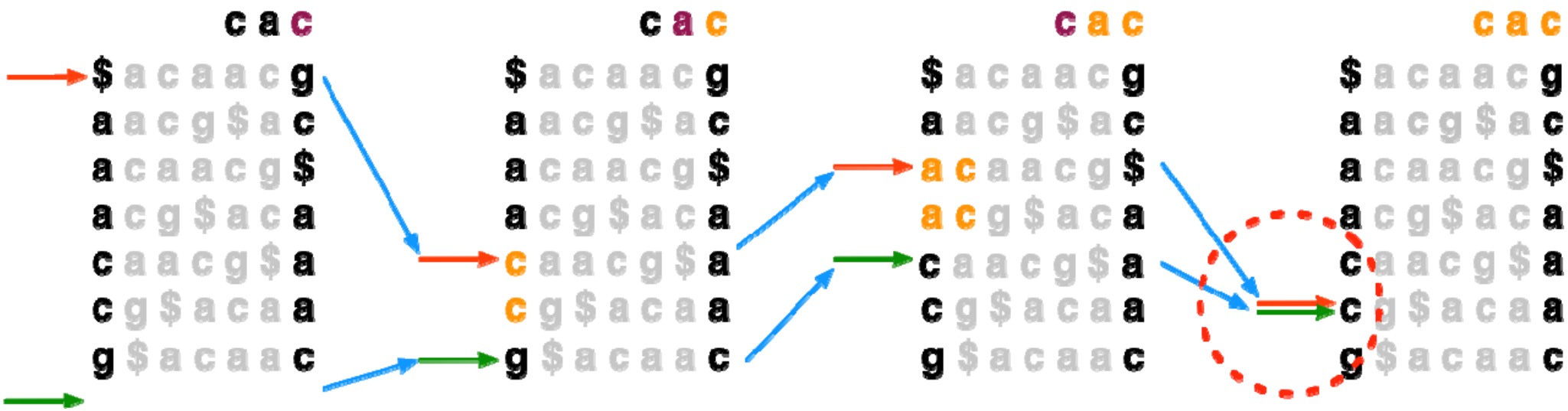
- To match Q in T using BWT(T), repeatedly apply rule:
  - **top** = **LF**(**top**, **qc**); **bot** = **LF**(**bot**, **qc**)
  - Where **qc** is the next character in Q (right-to-left) and **LF**(i, **qc**) maps row i to the row whose first character corresponds to i's last character *as if it were **qc***

Exact Matching with FM Index



- In progressive rounds, **top** & **bot** delimit the range of rows beginning with progressively longer suffixes of Q

Exact Matching with FM Index



- If range becomes empty (**top** = **bot**) the query suffix (and therefore the query) does not occur in the text

Backtracking

- Consider an attempt to find Q = “agc” in T = “acaacg”:

- Instead of giving up, try to “backtrack” to a previous position and try a different base

Sequencing

Find maximal overlaps between fragments:

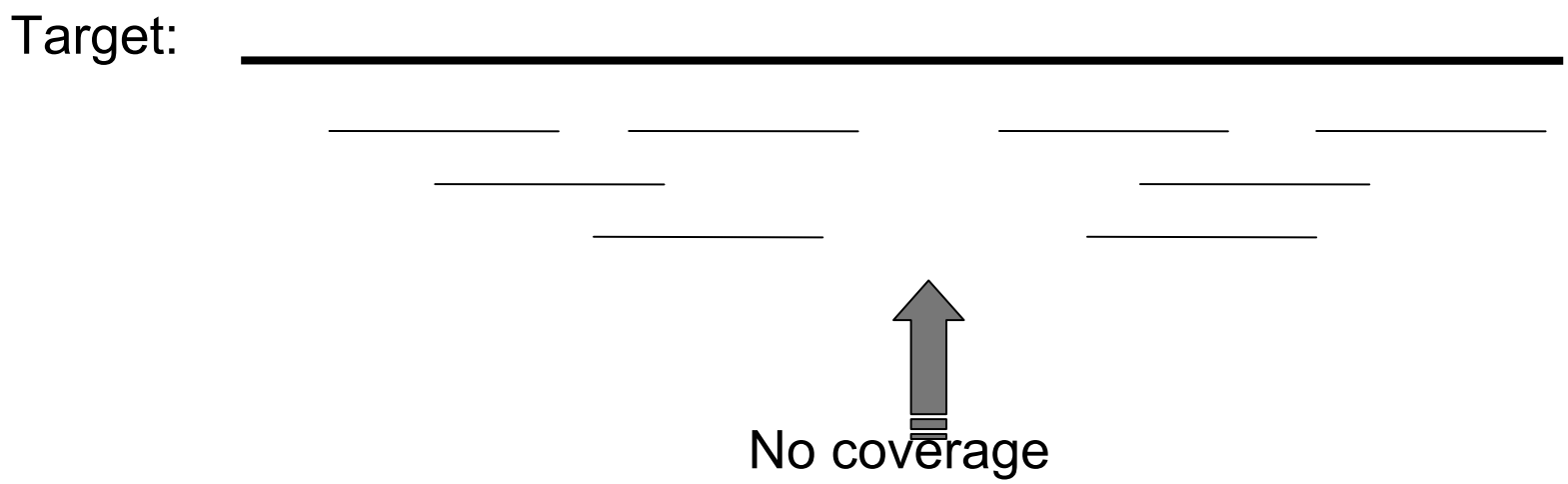
```
ACCGT
CGTGC
TTAC
TACCGT
```

```
--ACCGT--
----CGTGC
TTAC-----
-TACCGT—
TTACCGTGC
```

Consensus sequence determined by vote

Quality Metrics

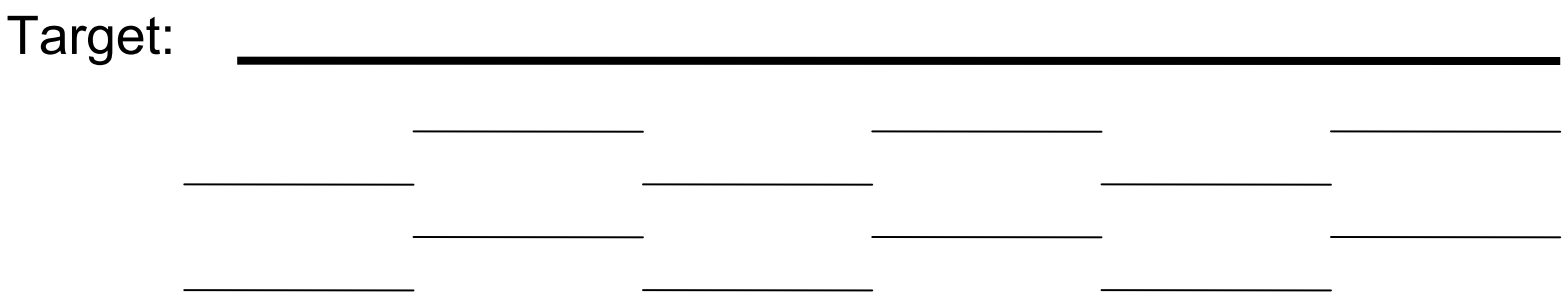
The *coverage* at position *i* of the target or consensus sequence is the number of fragments that overlap that position



Two *contigs*

Quality Metrics

Linkage – the degree of overlap between fragments



Perfect coverage, poor average linkage
poor minimum linkage

Real World Complications

Base call errors

Chimeric fragments, contamination (e.g. from the vector)

```
--ACCGT--
-----CGTGC
TTAC-----
-TGCCGT-
TTACCGTGC
```

Base Call Error

```
--ACC-GT--
----CAGTGC
TTAC------
-TACC-GT-
TTACC-GTGC
```

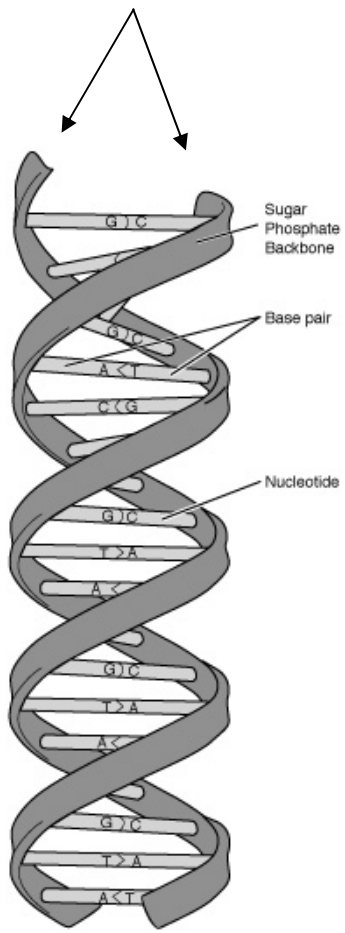
Insertion Error

```
--ACCGT--
-----CGTGC
TTAC-----
-TAC-GT-
TTACCGTGC
```

Deletion Error

Unknown Orientation

A fragment can come from either strand




```
CACGT   →   CACGT
ACGT    →   -ACGT
ACTACG  ←   --CGTAGT
GTACT   ←   -----AGTAC
ACTGA   →   --------ACTGA
CTGA    →   ---------CTGA
```

Sequence Alignment Models

Shortest common superstring

Input: A collection, $\mathscr{F}$, of strings (fragments)

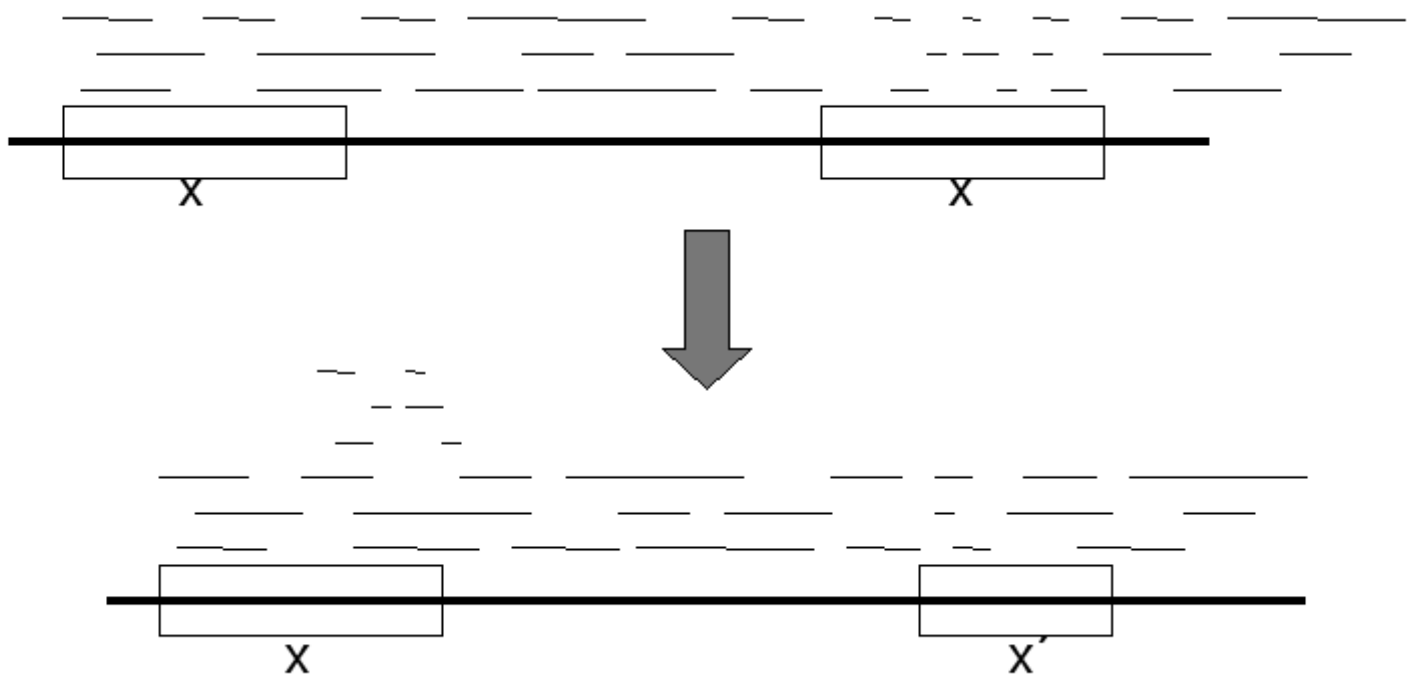
Output: A shortest possible string S such that for every $f \in \mathscr{F}$, S is a superstring of f.

Example:

$\mathscr{F}$ = {ACT, CTA, AGT}

S = ACTAGT

Problems with the SCS model



- Directionality of fragments must be known
- No consideration of *coverage*
- Some simple consideration of *linkage*
- No consideration of base call errors

Reconstruction

Deals with errors and unknown orientation

Definitions

f is an approximate substring of S at error level ε when $d_s(f, S) \leq \varepsilon \times |f|$

d_s = substring edit distance:

Match = 0
Mismatch = 1
Gap = 1

Reconstruction

Input: A collection, $\mathcal{F}$, of strings, and a tolerance level, ε

Output: Shortest possible string, S, such that for every $f \in \mathcal{F}$:

$$\min \left[d_s(f, S) + d_s(f, S) \leq \varepsilon |f| \right]$$

Reconstruction Example

Input: $\mathscr{F}$ = {ATCAT, GTCG, CGAG, TACCA}

$\varepsilon = 0.25$

Output:

```
ATGAT
------CGAC
-CGAG
----TACCA
ACGATACGAC
```

ATCAT

GTCG

d_s(CGAG, ACGATACGAC) = 1
= 0.25 × 4

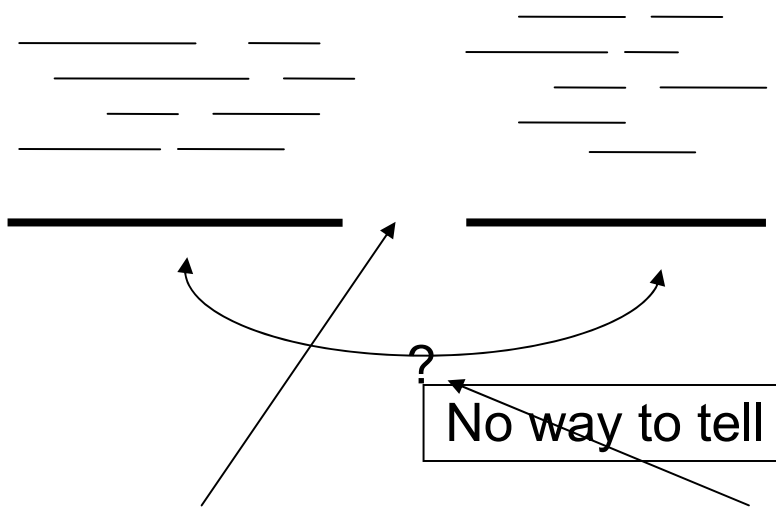
So this output is OK for $\varepsilon = 0.25$

Limitations of Reconstruction

- Models errors and unknown orientation
- Doesn’t handle repeats
- Doesn’t model coverage
- Only handles linkage in a very simple way
- Always produces a single contig

Contigs

Sometimes you just can't put all of the fragments together into one *contiguous* sequence:



No way to tell the order of these two *contigs.*

No way to tell how much sequence is missing between them.

Multicontig

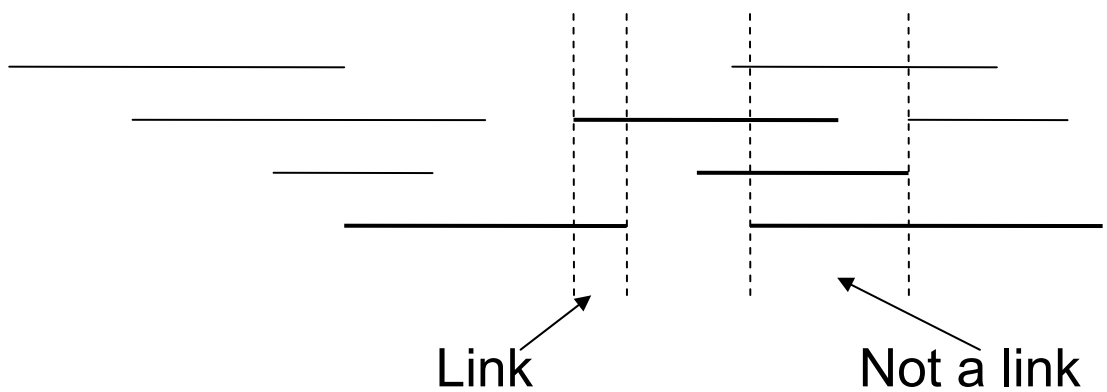
Definitions

A layout, $\mathcal{L}$, is a multiple alignment of the fragments

Columns numbered from 1 to $|\mathcal{L}|$

Endpoints of a fragment: $l(f)$ and $r(f)$

An overlap is a *link* where no other fragment completely covers the overlap



Multicontig

More definitions

The size of a link is the number of overlapping positions

```
ACGTATAG|CATGA
   GTA  |CATGATCA
ACGTATAG|    GATCA
```

A link of size 5

The *weakest link* is the smallest link in the layout

A *t-contig* has a weakest link of size *t*

A collection, $\mathscr{F}$, *admits* a *t*-contig if a *t*-contig can be constructed from the fragments in $\mathscr{F}$

Perfect Multicontig

Input: $\mathscr{F}$, and t

Output: a minimum number of collections, C_i, such that every C_i admits a t-contig

Let $\mathscr{F}$ = {GTAC, TAATG, TGTAA}

t = 3

```
--TAATG
TGTAA--

GTAC
```

t = 1

```
TGTAA-----
--TAATG---
------GTAC
```

Handling errors in Multicontig

- The *image* of a fragment is the portion of the consensus sequence, *S*, corresponding to the fragment in the layout
- *S* is an *ε-consensus* for a collection of fragments when the edit distance from each fragment, *f*, and its image is at most $\varepsilon \times |f|$

```
   TATAGCATCAT
CGTC     CATGATCA
ACGGATAG     GTCCA
ACGTATAGCATGATCA
```

An *ε-consensus* for $\varepsilon = 0.4$

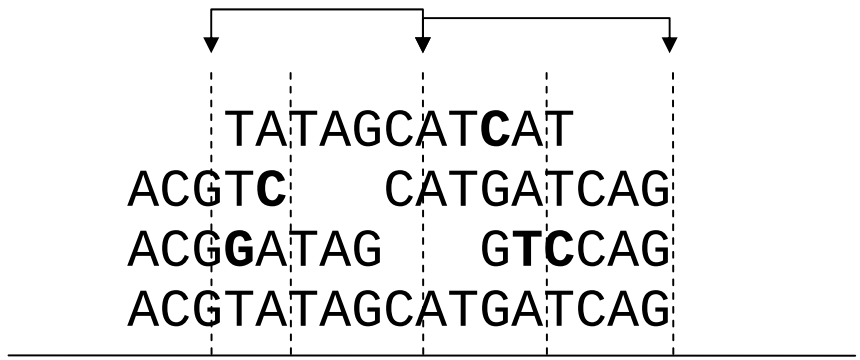
Definition of Multicontig

Input: A collection, $\mathscr{F}$, of strings, an integer $t \geq 0$, and an error tolerance ε between 0 and 1

Output: A partition of $\mathscr{F}$ into the minimum number of collections C_i such that every C_i admits a t-contig with an ε-consensus

Example of Multicontig

Let $\varepsilon = 0.4$, $t = 3$



Assembly Algorithms

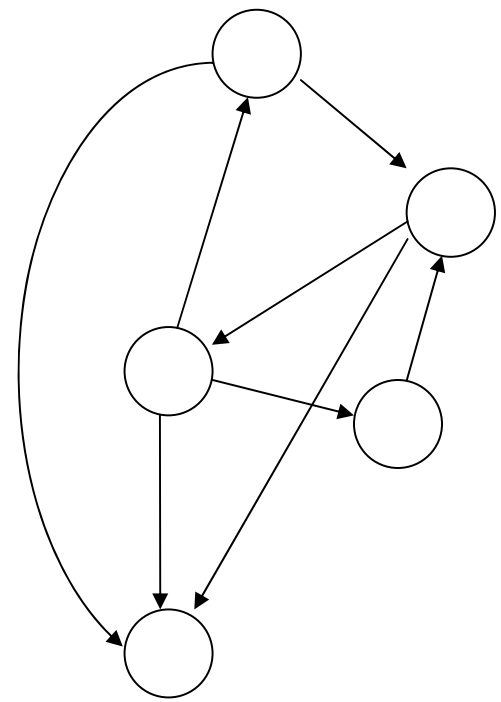
Most of the algorithms to solve the fragment assembly problem are based on a <u>graph</u> model

A graph, *G*, is a collection of edges, *e*, and vertices, *v*.

Directed or undirected

Weighted or unweighted

We will discuss representations and other issues shortly…

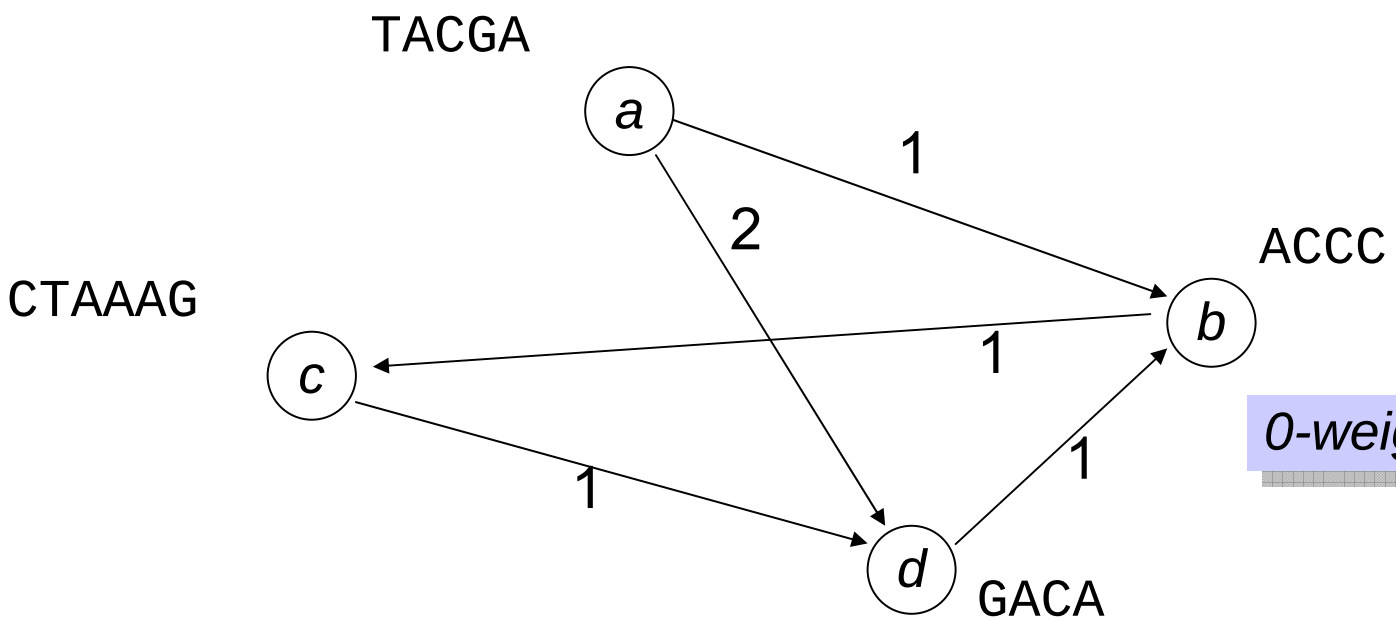


A directed, unweighted graph

The Maximum Overlap Graph

Overlap multigraph

Each directed edge, (*u,v*) is weighted with the length of the maximal overlap between a suffix of *u* and a prefix of *v*

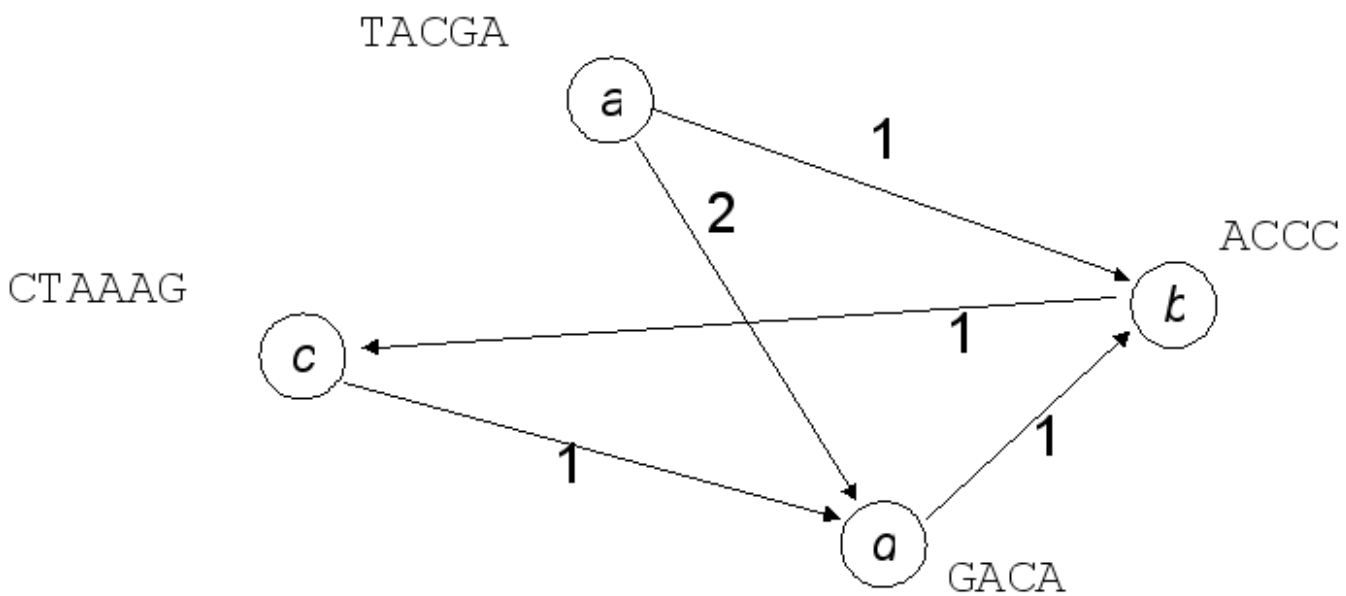


0-weight edges omitted!

Paths and Layouts

The path *dbc* leads to the alignment:

```
GACA--------
---ACCC-----
------CTAAAG
```



Superstrings

Every path that covers every node is a superstring

Zero weight edges result in alignments like:

```
GACA--------
----GCCC-----
  --------TTAAAG
```

Higher weights produce more overlap, and thus shorter strings

The *shortest common superstring* is the highest weight path that covers every node

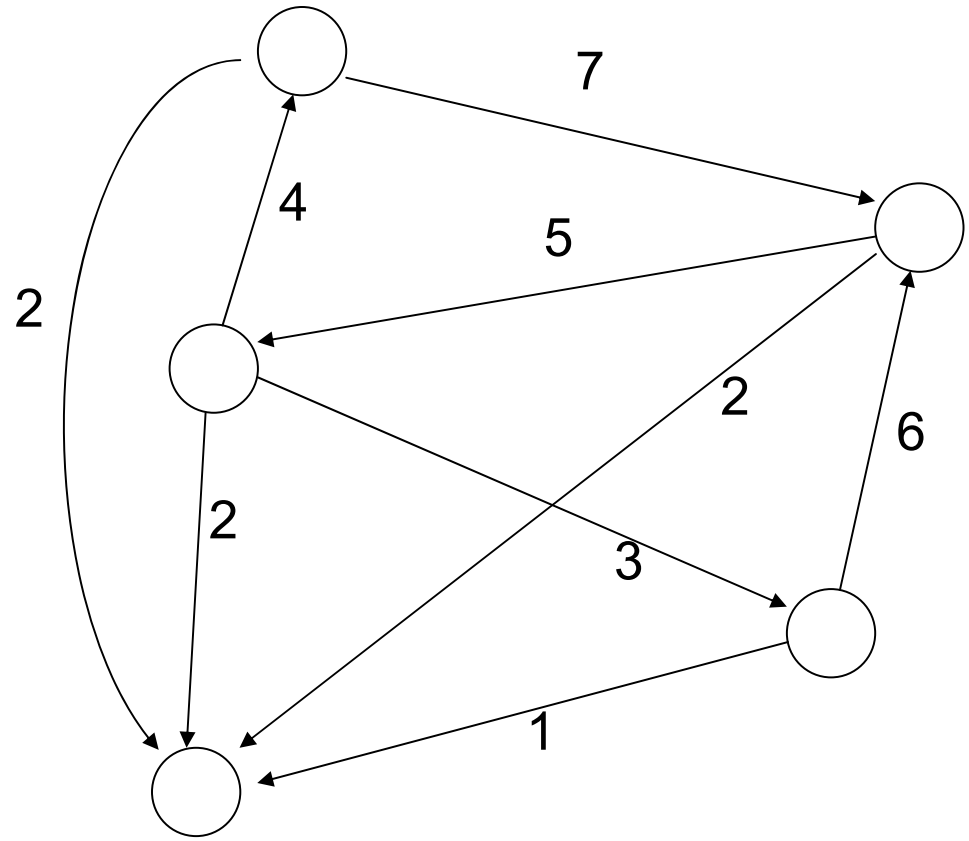
Graph formulation of SCS

Input: A weighted, directed graph

Output: The highest-weight path that touches every node of the graph

NP Hard, Use Greedy Approximation

Greedy Example



So we have sequences now!

- Find genes in sequences.
- Query: `AGTACGTATCGTATAGCGTAA`

What does it do?

- Find similar gene in other species with known function and reason from it
- Align sequences with known genes
- Find the gene with the “best” match

Sequence Alignment

- Point mutations can be easily handled:
  ACGTCTGAT**A**CGCC**G**TAT**A**GTCTATCT
  ACGTCTGAT**T**CGCC**C**TAT**C**GTCTATCT

- Insertions and deletions (InDels) are harder!

  ACG**TC**T**GAT**A**CGCCG**TAT**AGTCTATCT**
  CTG**AT**T**CGC**A**TCGTC**TAT**CT**

  ACGTCTGAT**A**CGCCGTAT**A**GTCTATCT
  ----CTGAT**T**CGC---AT**C**GTCTATCT

Sequence Alignment: Scoring

Match score: +1

Mismatch score: +0

Gap penalty: –1

```
ACGTCTGATACGCCGTATAGTCTATCT
     |||||  |||    ||  |||||||||
----CTGATTCGC---ATCGTCTATCT
```

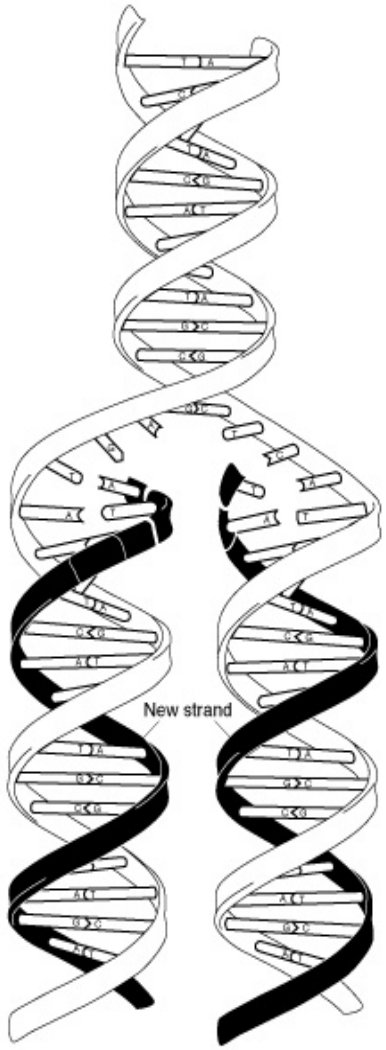
Matches: 18 × (+1)

Mismatches: 2 × 0

Gaps: 7 × (– 1)

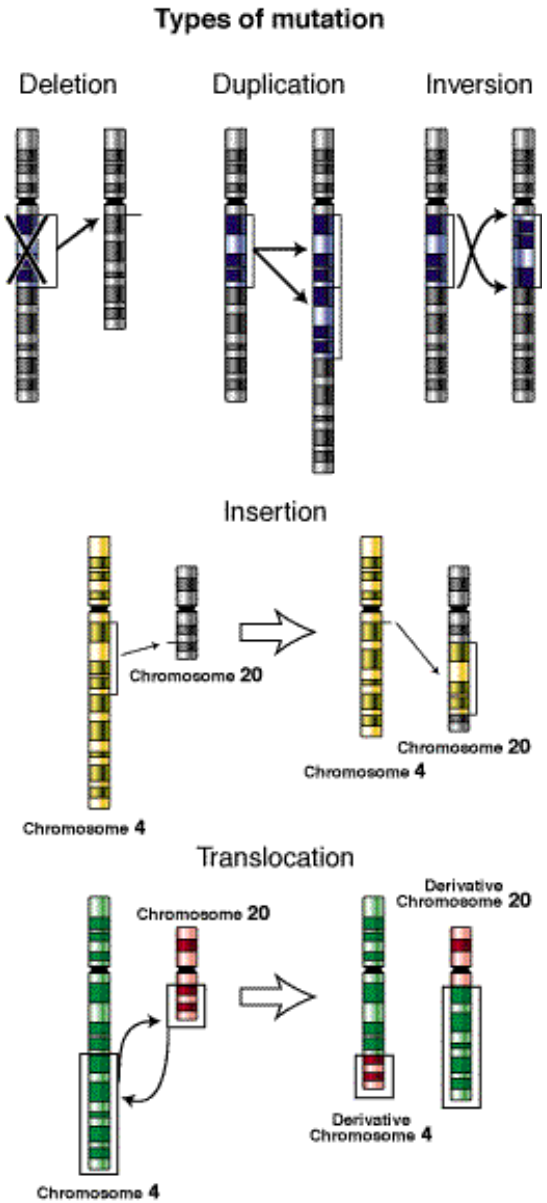
Sequence Alignment: Scoring

- Prior to cell division, all the genetic instructions must be “copied” so that each new cell will have a complete set
- DNA polymerase is the enzyme that copies DNA
  - Synthesizes in the 5' to 3' direction



Sequence Alignment: Scoring

- Environmental factors
  -
  -
- Mistakes in replication or repair
  -
  -
  -
  -



Deletions in Sequences

- Codon deletion:
  ACG ATA GCG TAT GTA TAG CCG…
  - Effect depends on the protein, position, etc.
  - Almost always deleterious
  - Sometimes lethal
- Frame shift mutation:
  ACG ATA GCG TAT GTA TAG CCG…
  ACG ATA GCG ATG TAT AGC CG?…
  - Almost always lethal

Insertions/ Deletions

- It is very difficult to determine whether an *InDel* is an insertion in one gene, or a deletion in another, unless ancestry is known:

ACGTCTGAT**ACG**CCGTATCGTCTATCT
ACGTCTGAT---CCGTATCGTCTATCT

Insertions/ Deletions

- We want to find alignments that are *evolutionarily likely.*
- Which of the following alignments is more likely?

```
ACGTCTGATACGCCGTATAGTCTATCT
ACGTCTGAT-------ATAGTCTATCT

ACGTCTGATACGCCGTATAGTCTATCT
AC-T-TGA--CG-CGT-TA-TCTATCT
```

- Initiating a gap must cost more than extending an existing gap! (why?)

Alignments

- Match/mismatch score: +1/+0
- Origination/length penalty: –2/–1

```
ACGTCTGATACGCCGTATAGTCTATCT
     |||||  |||    ||  |||||||||
----CTGATTCGC---ATCGTCTATCT
```

- Matches: 18 × (+1)
- Mismatches: 2 × 0
- Origination: 2 × (–2)
- Length: 7 × (–1)

Optimal Alignments

- Finding optimal alignment hard:
  ACGTCTGATACGCCGTATAGTCTATCT
  CTGAT- - - -TCG—CATCGTC- - -T-ATCT
- C(27,7) gap positions = ~888,000 possibilities
- Dynamic programming: The Smith Waterman algorithm

Optimal Alignments

An Example:

```
ACTCG
  ACAGTAG
```

- Match: +1
- Mismatch: 0
- Gap: –1

Dynamic Programming

- Each sequence along one axis
- Mismatch penalty multiples in first row/column 0 in [0,0]

		A	C	T	C	G
	0	-1	-2	-3	-4	-5
A	-1	1				
C	-2					
A	-3					
G	-4					
T	-5					
A	-6					
G	-7					

Dynamic Programming

- Vertical/Horiz. move: Score + (simple) gap penalty
- Diagonal move: Score + match/mismatch score
- Take the **MAX** of the three possibilities

		A	C	T	C	G
	0	-1	-2	-3	-4	-5
A	-1	1				
C	-2					
A	-3					
G	-4					
T	-5					
A	-6					
G	-7					

Dynamic Programming

		a	c	t	c	g
	0	-1	-2	-3	-4	-5
a	-1	1	0	-1	-2	-3
c	-2	0	2	1	0	-1
a	-3	-1	1	2	1	0
g	-4	-2	0	1	2	2
t	-5	-3	-1	1	1	2
a	-6	-4	-2	0	1	1
g	-7	-5	-3	-1	0	2

Optimal Alignment

- Trace back from the maximum value to the origin.

		a	c	t	c	g
	0	-1	-2	-3	-4	-5
a	-1	1	0	-1	-2	-3
c	-2	0	2	1	0	-1
a	-3	-1	1	2	1	0
g	-4	-2	0	1	2	2
t	-5	-3	-1	1	1	2
a	-6	-4	-2	0	1	1
g	-7	-5	-3	-1	0	2

Paths Correspond to Alignments

- ↑ = GAP in top sequence
- ← = GAP in left sequence
- ↖ = ALIGN both positions
- Path from the previous table: ↖↖↑↑↖↖↖
- Corresponding alignment (start at the end):

```
AC--TCG
ACAGTAG
```

Score = +2

Semi-Global Alignments

- Suppose we are aligning:
  GCG
  GGCG
- Which one is biologically relevant?
  G-CG -GCG
  GGCG GGCG
- Semi-global alignment allows gaps at the ends for free.

Semi-global alignment

- Semi-global alignment allows gaps at the ends for free.

		g	c	g
	0	0	0	0
g	0	1	0	1
g	0	1	1	1
c	0	0	2	1
g	0	1	1	3

-
-

Local alignment

- Global alignment – score entire alignment
- Semi-global alignments – allow unscored gaps at the beginning or end of either sequence
- Local alignment – find the best matching subsequence
- CG**ATG**
  AA**ATG**GA
- This is achieved through a 4th alternative at each position in the table: zero.

Local alignment

- Mismatch = –1 this time

		c	g	a	t	g
	0	-1	-2	-3	-4	-5
a	-1	0	0	0	0	0
a	-2	0	0	1	0	0
a	-3	0	0	1	0	0
t	-4	0	0	0	2	1
g	-5	0	1	0	1	3
g	-6	0	1	0	0	2
a	-7	0	0	2	1	1

CG**ATG**

AA**ATG**GA

Optimal Sub-alignments

- Consider the alignment:

```
ACGTCTGATACGCCGTATAGTCTATCT
      ||||| |||    || |||||||
----CTGATTCGC---ATCGTCTATCT
```

- Is it true that the alignment in the boxed region *must be optimal?*

A Greedy Strategy

- Consider this pair of sequences
  GAGC
  CAGC

GAP = –1

Match = +1

Mismatch = –2

- Greedy Approach:

```
G   or   G   or   -
C        -        G
```

- Leads to

```
GAGC---     Better:   GACG
---CAGC               CACG
```

Divide and Conquer

- Suppose we are aligning:

  ACTCG

  ACAGTAG

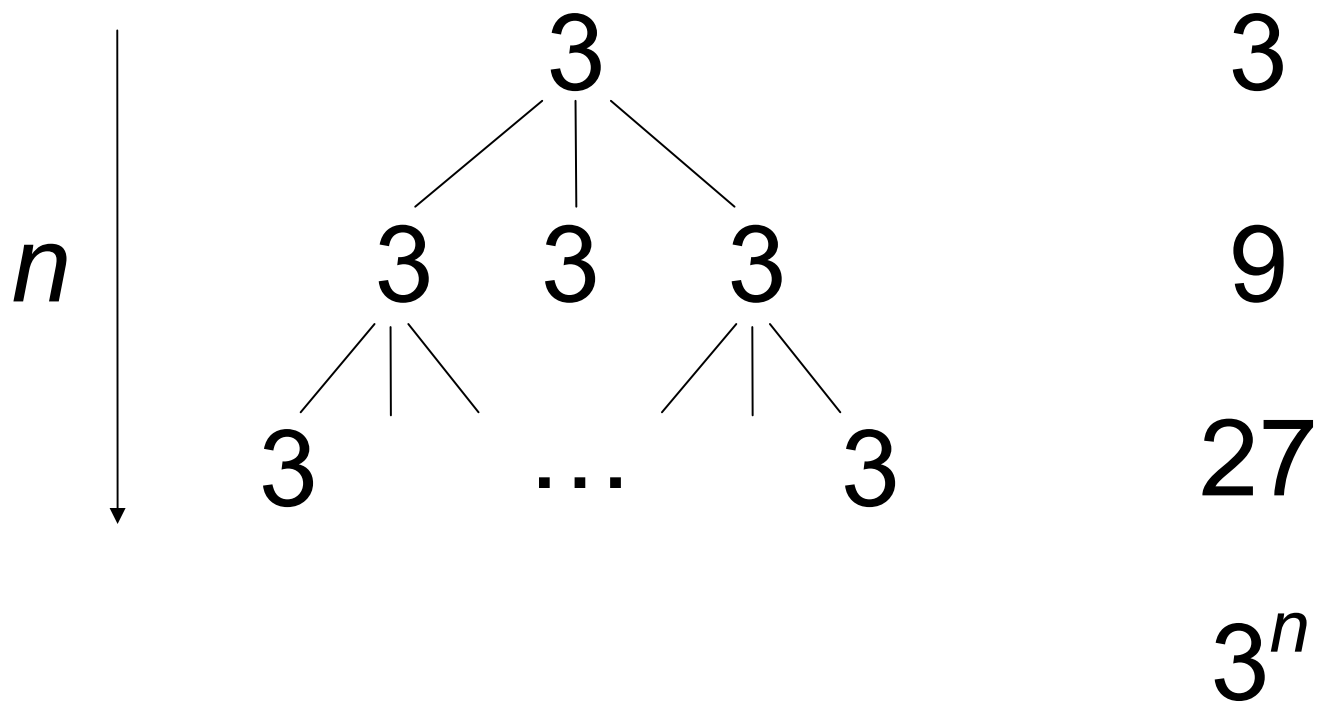
- First position choices:

A A	+1	CTCG CAGTAG
A –	–1	CTCG ACAGTAG
– A	–1	ACTCG CAGTAG

Complexity of RecurseAlign

- What is the recurrence equation for the time needed by RecurseAlign?

$$T(n) = 3T(n-1) + 3$$



Dynamic Programming

		a	c	t	c	g
	0	-1	-2	-3	-4	-5
a	-1	1	0	-1	-2	-3
c	-2	0	2	1	0	-1
a	-3	-1	1	2	1	0
g	-4	-2	0	1	2	2
t	-5	-3	-1	1	1	2
a	-6	-4	-2	0	1	1
g	-7	-5	-3	-1	0	2

- This is possible for any problem that exhibits *optimal substructure (Bellman's principle of optimality)*

Space Complexity

- Note that we can throw away the previous rows of the table as we fill it in:

This row is based only on this one

		a	c	t	c	g
	0	-1	-2	-3	-4	-5
a	-1	1	0	-1	-2	-3
c	-2	0	2	1	0	-1
a	-3	-1	1	2	1	0
g	-4	-2	0	1	2	2
t	-5	-3	-1	1	1	2
a	-6	-4	-2	0	1	1
g	-7	-5	-3	-1	0	2

Space Complexity

- ***Each row*** of the table contains the scores for aligning a ***prefix*** of the left-hand sequence with ***all prefixes*** of the top sequence:

Scores for aligning **`aca`** with all prefixes of **`actcg`**

		a	c	t	c	g
	0	-1	-2	-3	-4	-5
a	-1	1	0	-1	-2	-3
c	-2	0	2	1	0	-1
a	-3	-1	1	2	1	0
g	-4	-2	0	1	2	2
t	-5	-3	-1	1	1	2
a	-6	-4	-2	0	1	1
g	-7	-5	-3	-1	0	2

Space Complexity

- ***Each row*** of the table contains the scores for aligning a ***prefix*** of the left-hand sequence with ***all prefixes*** of the top sequence:

Scores for aligning **`aca`** with all prefixes of **`actcg`**

		a	c	t	c	g
	0	-1	-2	-3	-4	-5
a	-1	1	0	-1	-2	-3
c	-2	0	2	1	0	-1
a	-3	-1	1	2	1	0
g	-4	-2	0	1	2	2
t	-5	-3	-1	1	1	2
a	-6	-4	-2	0	1	1
g	-7	-5	-3	-1	0	2

So Where Does *i* Line Up?

- Find out where *i* aligns to the bottom sequence
  - Needs *two vectors* of scores

```
                    i
                    ↓
s:  ACGCTATGCTCATAG

t:  CGACGCTCATCG
```

- Assuming *i* lines up with a **character**:
  alignscore = align(ACGCTAT, prefix(*t*)) + score(G, char from *t*)
  + align(CTCATAG, suffix(*t*))
- Which character is best?
  - Can quickly find out the score for aligning ACGCTAT with **every prefix** of *t*.

So where does *i* line up?

- But, *i* may also line up with a gap

i

s: ACGCTAT**G**CTCATAG

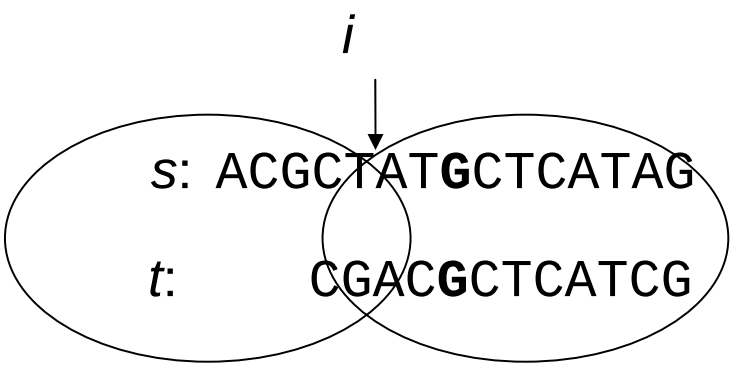
t: CGACGCTCATCG

- Assuming *i* lines up with a **gap**:

alignscore = align(ACGCTAT, prefix(*t*)) + gapscore
+ align(CTCATAG, suffix(*t*))

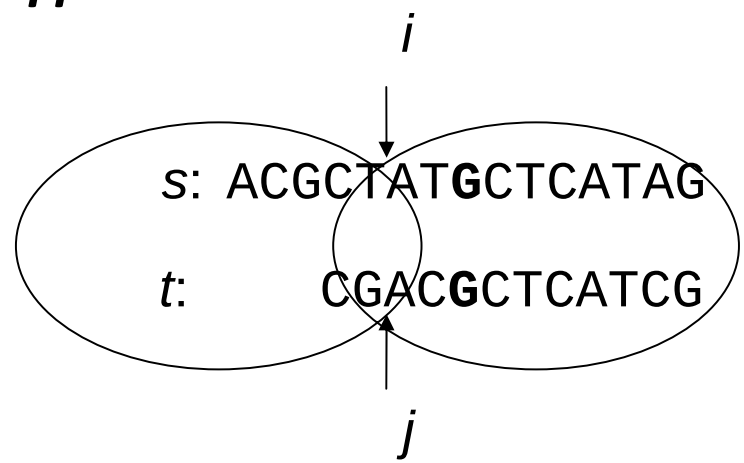
Recursive Call

- Fix the best position for *I*
- Call *align* recursively for the prefixes and suffixes:



Time Complexity

- Let len(s) = m and len(t) = n
- **Space: 2*m***
- **Time:**
  - Each call to build similarity vector = $m'n'$
  - First call + recursive call:



$$T(m,n) \le \frac{mn}{2} + \frac{mn}{2} + T\left(\left\lceil \frac{m}{2} \right\rceil, j\right) + T\left(\left\lceil \frac{m}{2} \right\rceil, n-j\right)$$

$$mn + mj + m(n-j)$$

¿ $2mn$

General Gap Penalties

- Suppose we are no longer using simple gap penalties:
  - Origination = –2
  - Length = –1
- Consider the last position of the alignment for `ACGTA` with `ACG`
- We can't determine the score for

G	or	–
–		G

- unless we know the previous positions!

Scoring Blocks

- Now we must score a *block* at a time

A	A	C	---	A	TATCCG	A	C	T	AC
A	C	T	ACC	T	------	C	G	C	--

- A *block* is a pair of characters, or a <u>maximal</u> group of gaps paired with characters
- To score a position, we need to either start a new block or add it to a previous block

Alignment Algorithm

- Three tables
  - *a* – scores for alignments ending in char-char blocks
  - *b* – scores for alignments ending in gaps in the top sequence (*s*)
  - *c* – scores for alignments ending in gaps in the left sequence (*t*)
- Scores no longer depend on only three positions, because we can put *any number* of gaps into the last block

The Recurrences

$$a[i,j] = p(i,j) + \max \begin{cases} a[i-1,j-1] \\ b[i-1,j-1] \\ c[i-1,j-1] \end{cases}$$

$$b[i,j] = \max \begin{cases} a[i,j-k] + w(k), & \text{for } 1 \le k \le j \\ c[i,j-k] + w(k), & \text{for } 1 \le k \le j \end{cases}$$

$$c[i,j] = \max \begin{cases} a[i-k,j] + w(k), & \text{for } 1 \le k \le i \\ b[i-k,j] + w(k), & \text{for } 1 \le k \le i \end{cases}$$

The Optimal Alignment

- The optimal alignment is found by looking at the maximal value in the lower right of all three arrays
- The algorithm runs in $O(n^3)$ time
  - Uses $O(n^2)$ space

Searching in Sequence Databases: BLAST

Database Searching

- How can we find a particular short sequence in a database of sequences (or one HUGE sequence)?
- Problem is identical to local sequence alignment, but on a much larger scale.
- We must also have some idea of the *significance* of a database hit.
  - Databases <u>always</u> return some kind of hit, how much attention should be paid to the result?

BLAST

- BLAST: Basic Local Alignment Search Tool
- An approximation of the Dynamic Programming algorithm
- Sacrifices some search sensitivity for speed

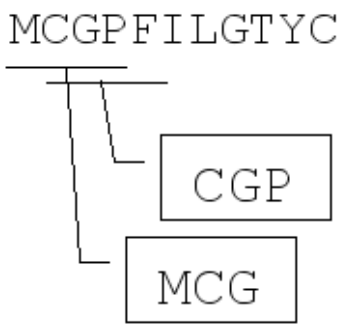
Scoring Matrices

- DNA
  - Identity
  - Transition/Transversion
- Proteins
  - PAM
  - BLOSUM

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

The BLAST algorithm

- Break the search sequence into *words*
  - $W = 3$ for proteins, $W = 12$ for DNA



MCG, CGP, GPF, PFI, FIL,
ILG, LGT, GTY, TYC

- Include in the search all words that score above a certain value (T) for any search word

MCG	CGP	
MCT	MGP	…
MCN	CTP	
…	…	

This list can be computed in linear time

BLAST Algorithm

- Search for the words in the database
  - Word locations can be precomputed and indexed
  - Searching for a short string in a long string
    - Regular expression matching: FSA
- HSP (High Scoring Pair) = A match between a query word and the database
- Find a “hit”: Two non-overlapping HSP’s on a diagonal within distance A
- Extend the hit until the score falls below a threshold value, X

The BLAST Search Algorithm

query word (W = 3)

```
Query:   GSVEDTTGSQSLAALLNKCKTPQGQRLVNQWIKQPLMDKNRIEERLNLVEAFVEDAELRQTLQEDL
```

neighborhood words

PQG	18
PEG	15
PRG	14
PKG	14
PNG	13
PDG	13
PHG	13
PMG	13
PSG	13
PQA	12
PQN	12
etc...	

neighborhood score threshold (T = 13)

```
Query:   325 SLAALLNKCKTPQGQRLVNQWIKQPLMDKNRIEERLNLVEA 365
             +LA++L+   TP G R++ +W+  P+ D   + ER   + A
Sbjct:   290 TLASVLDCTVTPMGSRMLKRWLHMPVRDTRVLLERQQTIGA 330
```

High-scoring Segment Pair (HSP)

Results from BLAST

NCBI CD-Search - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Search Favorites Media

Address http://www.ncbi.nlm.nih.gov/Structure/cdd/wrpsb.cgi?RID=1066068742-2377-951781.BLASTQ3 Go

---

- gnl|CDD|5811, LOAD_USPA, USPA, An ATP binding domain seen as a stand alone in USPA.

```
            CD-Length = 135 residues, 100.0% aligned
            Score = 90.5 bits (224), Expect = 8e-20

Query:  6    KKILYPTDFSETAEIALKHVKAFKTLKAEEVILLHVIDEREIKKRDIFSLLLGVAGLNKS  65
Sbjct:  1    KKILVAIDGSPESEKALRWAVDLAKRRGAELILLHVIPPSVSTAASPALDL---------  51

Query:  66   VEEFENELKNKLTEEAKNKMENIKKELEDVGFKVKDIIVVGIPHEEIVKIAEDEGVDIII  125
Sbjct:  52   ---------ALLEEALKLLEEALELLEEAGVKIDVEVEEGSPAEAILELAEESNADLIV  102

Query:  126  MGSHGKTNLKEILLGSVTENVIKKSNKPVLVVK  158
Sbjct:  103  VGSRGRGGLRRLLLGSVSEKVLRKAPCPVLVVR  135
```

- gnl|CDD|10459, COG0589, UspA, Universal stress protein UspA and related nucleotide-binding proteins [Signal transduction mechanisms]

```
            CD-Length = 154 residues, 100.0% aligned
            Score = 84.2 bits (207), Expect = 6e-18

Query:  1    MSVMYKKILYPTDF-SETAEIALKHVKAFKTLKAEEVILLHVIDEREIKKRDIFSLLLGV  59
Sbjct:  1    MPAMYKKILVAVDVGSEAAEKALEEAVALAKRLGAPLILLVVIDPLEPT----------A  50

Query:  60   AGLNKSVEEFENELKNKLTEEAKNKMENIKKELEDVG-FKVKDIIVVGIPH-EEIVKIAE  117
Sbjct:  51   LVSVALADAPIPLSEEELEEEAEELLAEAKALAEAAGVPVVETEVVEGSPSAEEILELAE  110

Query:  118  DEGVDIIIMGSHGKTNLKEILLGSVTENVIKKSNKPVLVVKRKN  161
Sbjct:  111  EEDADLIVVGSRGRSGLSRLLLGSVAEKVLRHAPCPVLVVRSEG  154
```

Internet

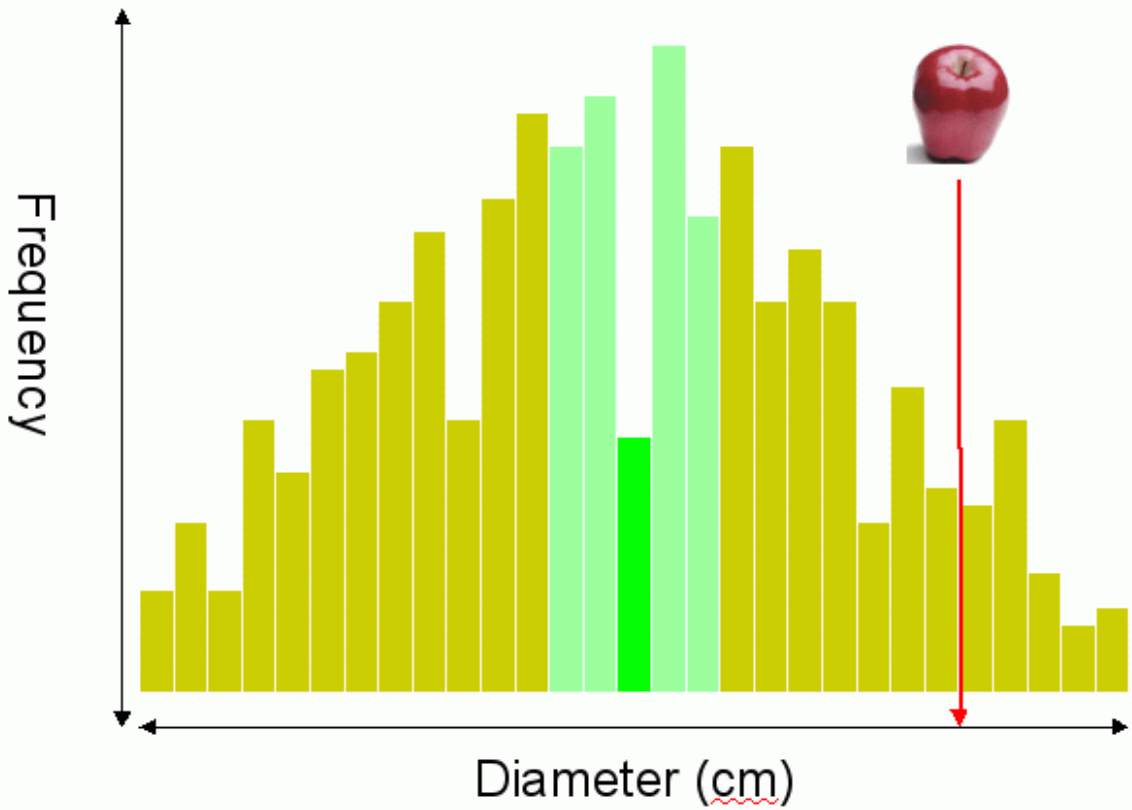
Search Significance Scores

- A search will *always* return some hits.

- How can we determine how “unusual” a particular alignment score is?
  - ORF’s
    - Assumptions

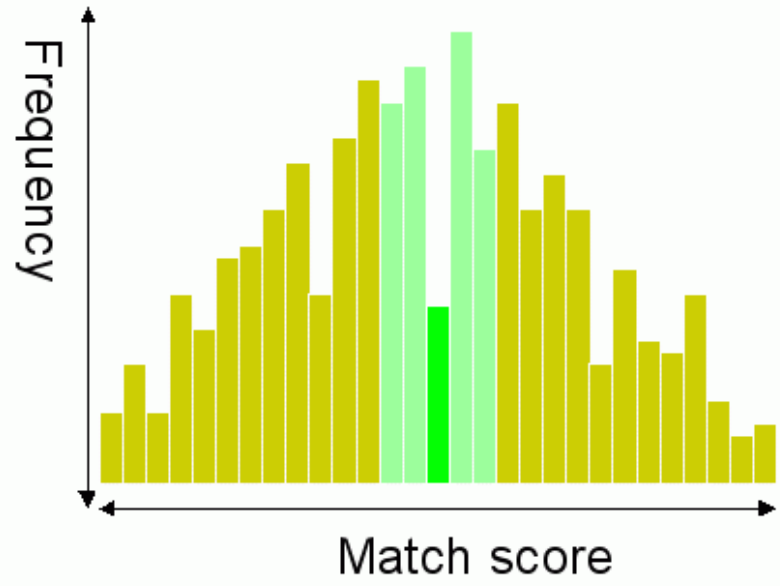
Significance from a Distribution

- I have an apple of diameter 5”. Is that unusual?



Is a Match Significant?

- Match scores for aligning my sequence with ***random sequences***.
- Depends on:
  - Scoring system
  - Database
  - Sequence to search for
    - Length
    - Composition



- How do we determine the ***random sequences?***

Generating “random” sequences

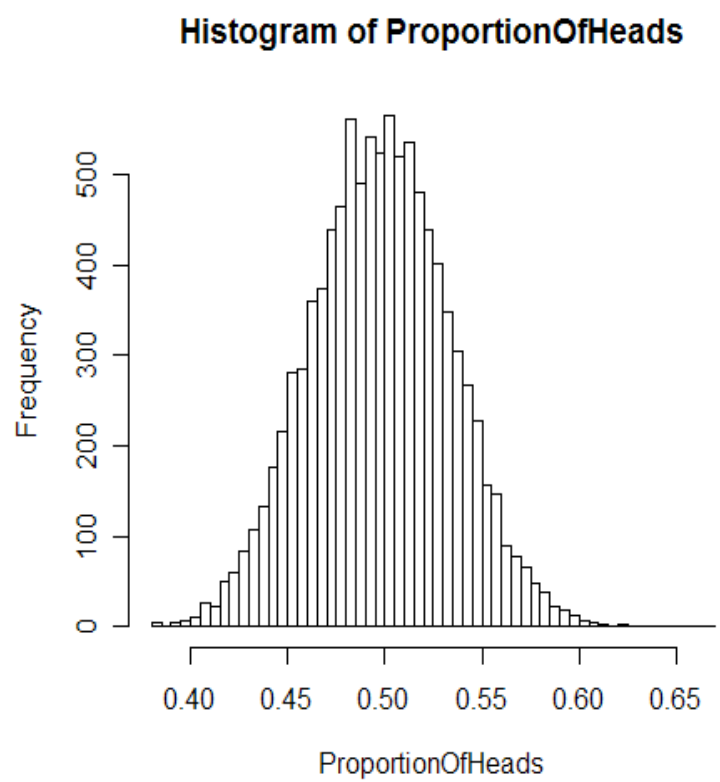
- Random uniform model:

  $$P(G) = P(A) = P(C) = P(T) = 0.25$$

  - Doesn’t reflect nature
- Use sequences from a database
  - Might have genuine homology
    - We want unrelated sequences
- Random shuffling of sequences
  - Preserves composition
  - Removes true homology

Sums of Distributions

- The <u>mean</u> of *n* random (i.i.d.) events tends towards a Normal/ Gaussian.
  - Example: Throw *n* dice and compute the mean.
  - Distribution of means:



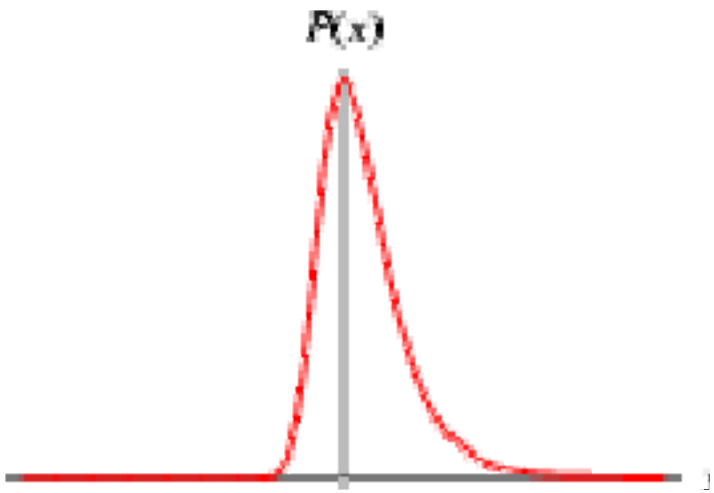
The Extreme Value Distribution

- This means that if we get the match scores for our sequence with *n* other sequences, the <u>mean</u> would follow a Gaussian distribution.

- The **maximum** of *n* (i.i.d.) random events tends towards the **extreme value distribution** as *n* grows large.

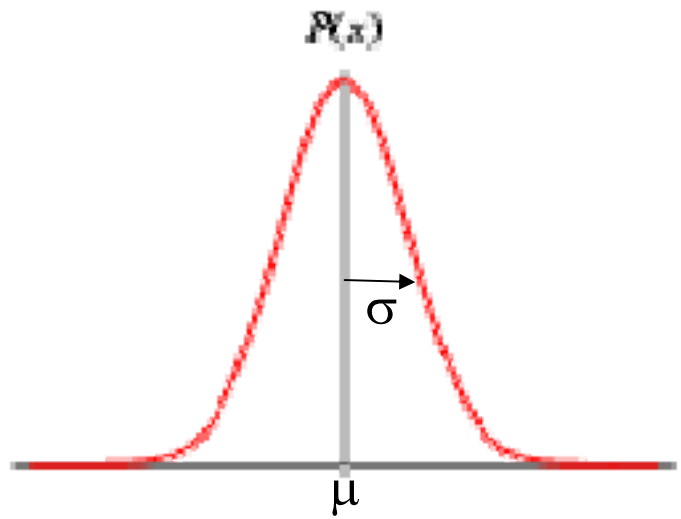
Gaussian/ Extreme Value Distributions

Extreme Value:



$$f(x) = \frac{1}{\beta} e^{-\frac{x-\mu}{\beta}} e^{-e^{-\frac{x-\mu}{\beta}}}$$

Gaussian:



$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2}$$

Computing P-values

- If we can estimate β and μ, then we can determine, for a given match score *x*, the probability that a random match with score *x* or greater would have occurred in the database.
- For sequence matches, a scoring system and database can be parameterized by two parameters, *K* and λ, related to β and μ.
  - It would be nice if we could compare hit significance without regard to the database and scoring system used!

Bit Scores

- Expected number of hits with score $\geq S$:

  $$E = Kmn\, e^{-\lambda s}$$

  - Where *m* and *n* are the sequence lengths
- Normalize the raw score using:

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

- Obtains a "bit score" S', with a *standard set of units.*
- The new E-value is: $E = mn\, 2^{-S'}$

P values and E values

- Blast reports *E*-values
- $E = 5$, $E = 10$ versus $P = 0.993$ and $P = 0.99995$
- When $E < 0.01$ *P*-values and *E*-values are nearly identical

BLAST Parameters

- Lowering the neighborhood word threshold (T) allows more distantly related sequences to be found, at the expense of increased noise in the results set.
- Raising the segment extension cutoff (X) returns longer extensions for each hit.
- Changing the minimum *E*-value changes the threshold for reporting a hit.

Aligning Protein Sequences

Sequence Alignments Revisited

- Scoring nucleotide sequence alignments was easier
  - Match score
  - Possibly different scores for transitions and transversions
- For amino acids, there are many more possible substitutions
- How do we score which substitutions are highly penalized and which are moderately penalized?
  - Physical and chemical characteristics
  - Empirical methods

Scoring Mismatches

- Physical and chemical characteristics
  - V $\rightarrow$ I – Both small, both hydrophobic, conservative substitution, small penalty
  - V $\rightarrow$ K – Small $\rightarrow$ large, hydrophobic $\rightarrow$ charged, large penalty
  - *Requires some expert knowledge and judgement*
- Empirical methods
  - How often does the substitution V $\rightarrow$ I occur in proteins that are known to be related?
    - Scoring matrices: PAM and BLOSUM

PAM Matrices

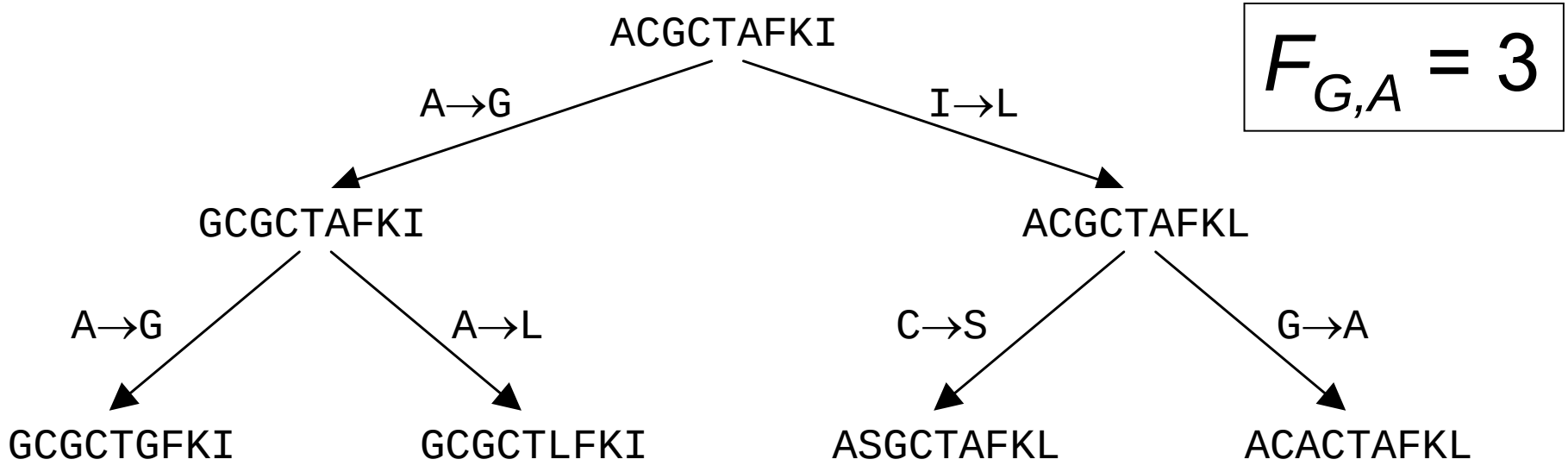
- PAM = "Point Accepted Mutation" interested only in mutations that have been "accepted" by natural selection
- Starts with a multiple sequence alignment of very similar (>85% identity) proteins. Assumed to be homologous
- Compute the *relative mutability*, m_i, of each amino acid
  - e.g. m_A = how many times was alanine substituted with anything else?

Relative Mutability

- ACGCTAFKI
  GCGCTAFKI
  ACGCTAFKL
  GCGCTGFKI
  GCGCTLFKI
  ASGCTAFKL
  ACACTAFKL
- Across *all pairs* of sequences, there are 28 A $\rightarrow$ X substitutions
- There are 10 ALA residues, so $m_A = 2.8$

Pam Matrices

- Construct a phylogenetic tree for the sequences in the alignment



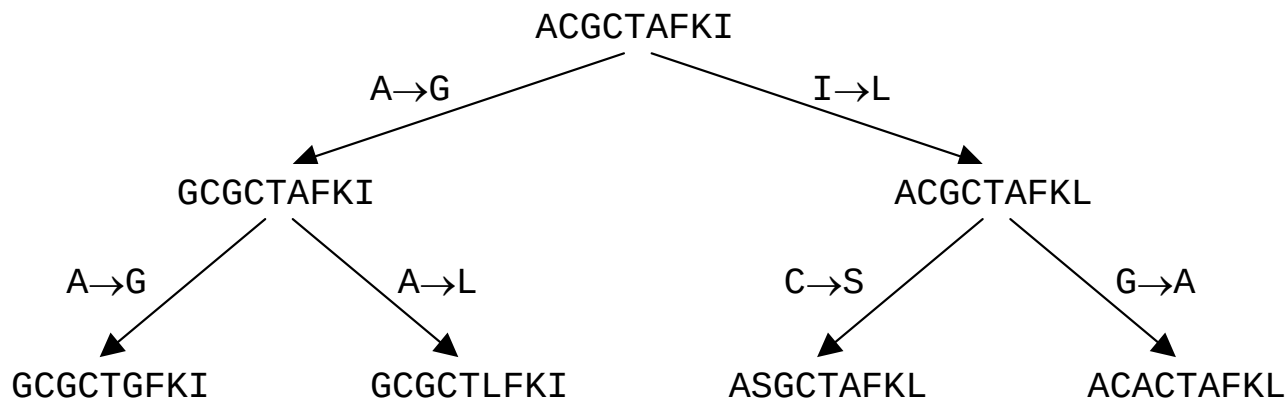
$$F_{G,A} = 3$$

- Calculate substitution frequences $F_{X,X}$
- Substitutions may have occurred either way, so A $\rightarrow$ G also counts as G $\rightarrow$ A.

Mutation Probabilities

- $M_{i,j}$ represents the score of J $\rightarrow$ I substitution.

$$M_{ij} = \frac{m_j F_{ij}}{\sum_i F_{ij}}$$



- $$M_{G,A} = \frac{2.7 \times 3}{4} = 2.025$$

The PAM matrix

- The entries, $R_{i,j}$ are the $M_{i,j}$ values divided by the frequency of occurrence, f_i, of residue i.
- f_G = 10 GLY / 63 residues = 0.1587
- $R_{G,A} = \log(2.025/0.1587) = \log(12.760) = 1.106$
- The log is taken so that we can add, rather than multiply entries to get compound probabilities.
- *Log-odds* matrix
- Diagonal entries are 1– m_j

Interpretation of PAM matrices

- PAM-1 – one substitution per 100 residues (a PAM unit of time)
- Multiply them together to get PAM-100, etc.
- "Suppose I start with a given polypeptide sequence M at time t, and observe the evolutionary changes in the sequence until 1% of all amino acid residues have undergone substitutions at time $t+n$. Let the new sequence at time $t+n$ be called M'. What is the probability that a residue of type j in M will be replaced by i in M'?"

PAM Matrix Considerations

- If $M_{i,j}$ is very small, we may not have a large enough sample to estimate the real probability. When we multiply the PAM matrices many times, the error is magnified.
- PAM-1 – similar sequences, PAM-1000 very dissimilar sequences

BLOSUM Matrix

- Starts by clustering proteins by similarity
- Avoids problems with small probabilities by using averages over clusters
- Numbering works opposite
  - BLOSUM-62 is appropriate for sequences of about 62% identity, while BLOSUM-80 is appropriate for **more** similar sequences.

Multiple Sequence Alignment

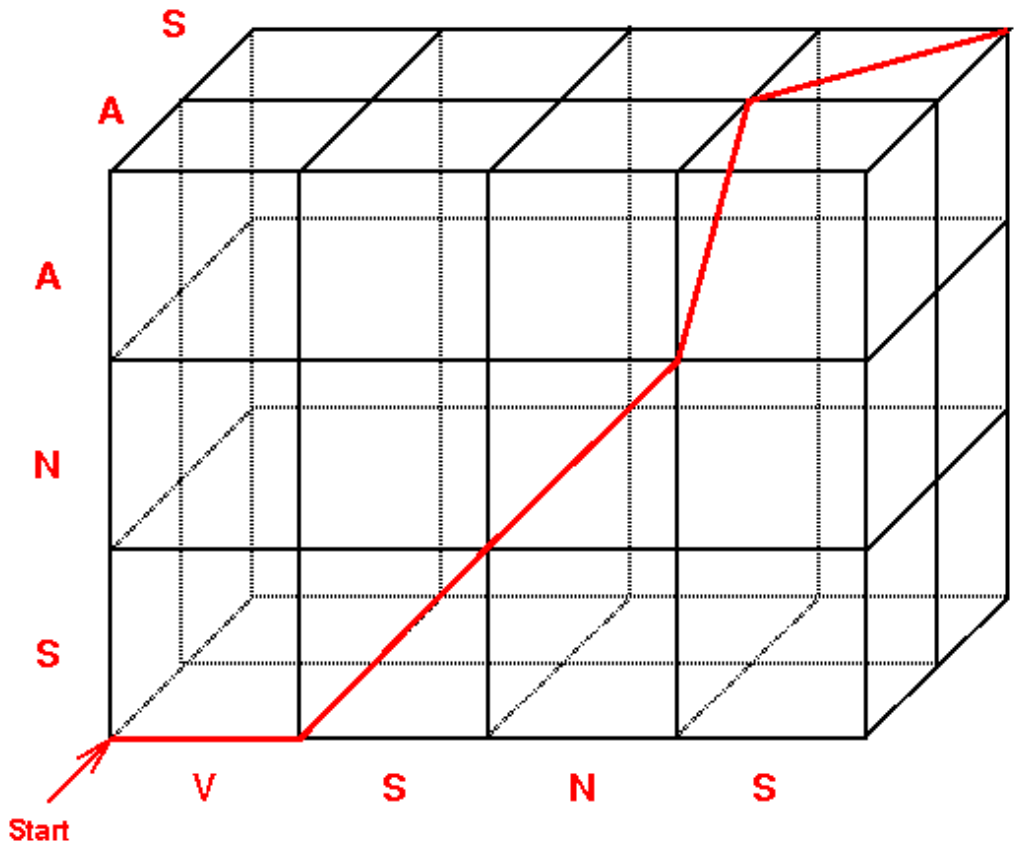
Multiple Alignment

```
                                 *   .     :     .         .   *       : : :     .
Q5E940_BOVIN -----------MPREDRATWKSNYFLKIIQLLDDYPKCFIVGADNVGSKQMQQIRMSLRGK-AVVLMGKNTMMRKAIRGHLENN--PALE    76
  RLA0_HUMAN -----------MPREDRATWKSNYFLKIIQLLDDYPKCFIVGADNVGSKQMQQIRMSLRGK-AVVLMGKNTMMRKAIRGHLENN--PALE    76
  RLA0_MOUSE -----------MPREDRATWKSNYFLKIIQLLDDYPKCFIVGADNVGSKQMQQIRMSLRGK-AVVLMGKNTMMRKAIRGHLENN--PALE    76
    RLA0_RAT -----------MPREDRATWKSNYFLKIIQLLDDYPKCFIVGADNVGSKQMQQIRMSLRGK-AVVLMGKNTMMRKAIRGHLENN--PALE    76
  RLA0_CHICK -----------MPREDRATWKSNYFMKIIQLLDDYPKCFVVGADNVGSKQMQQIRMSLRGK-AVVLMGKNTMMRKAIRGHLENN--PALE    76
  RLA0_RANSY -----------MPREDRATWKSNYFLKIIQLLDDYPKCFIVGADNVGSKQMQQIRMSLRGK-AVVLMGKNTMMRKAIRGHLENN--SALE    76
Q7ZUG3_BRARE -----------MPREDRATWKSNYFLKIIQLLDDYPKCFIVGADNVGSKQMQTIRLSLRGK-AVVLMGKNTMMRKAIRGHLENN--PALE    76
  RLA0_ICTPU -----------MPREDRATWKSNYFLKIIQLLNDYPKCFIVGADNVGSKQMQTIRLSLRGK-AIVLMGKNTMMRKAIRGHLENN--PALE    76
  RLA0_DROME -----------MVRENKAAWKAQYFIKVVELFDEFPKCFIVGADNVGSKQMQNIRTSLRGL-AVVLMGKNTMMRKAIRGHLENN--PQLE    76
  RLA0_DICDI -----------MSGAG-SKRKKLFIEKATKLFTTYDKMIVAEADFVGSSQLQKIRKSIRGI-GAVLMGKKTMIRKVIRDLADSK--PELD    75
Q54LP0_DICDI -----------MSGAG-SKRKNVFIEKATKLFTTYDKMIVAEADFVGSSQLQKIRKSIRGI-GAVLMGKKTMIRKVIRDLADSK--PELD    75
  RLA0_PLAF8 -----------MAKLSKQQKKQMYIEKLSSLIQQYSKILIVHVDNVGSNQMASVRKSLRGK-ATILMGKNTRIRTALKKNLQAV--PQIE    76
  RLA0_SULAC -----MIGLAVTTTKKIAKWKVDEVAELTEKLKTHKTIIIANIEGFPADKLHEIRKKLRGK-ADIKVTKNNLFNIALKNAG-----YDTK    79
  RLA0_SULTO ----MRIMAVITQERKIAKWKIEEVKELEQKLREYHTIIIANIEGFPADKLHDIRKKMRGM-AEIKVTKNTLFGIAAKNAG-----LDVS    80
  RLA0_SULSO ----MKRLALALKQRKVASWKLEEVKELTELIKNSNTILIGNLEGFPADKLHEIRKKLRGK-ATIKVTKNTLFKIAAKNAG-----IDIE    80
  RLA0_AERPE MSVVSLVGQMYKREKPIPEWKTLMLRELEELFSKHRVVLFADLTGTPTFVVQRVRKKLWKK-YPMMVAKKRIILRAMKAAGLE---LDDN    86
  RLA0_PYRAE -MMLAIGKRRYVRTRQYPARKVKIVSEATELLQKYPYVFLFDLHGLSSRILHEYRYRLRRY-GVIKIIKPTLFKIAFTKVYGG---IPAE    85
  RLA0_METAC ------MAEERHHTEHIPQWKKDEIENIKELIQSHKVFGMVGIEGILATKMQKIRRDLKDV-AVLKVSRNTLTERALNQLG-----ETIP    78
  RLA0_METMA ------MAEERHHTEHIPQWKKDEIENIKELIQSHKVFGMVRIEGILATKIQKIRRDLKDV-AVLKVSRNTLTERALNQLG-----ESIP    78
  RLA0_ARCFU ------MAAVRGS---PPEYKVRAVEEIKRMISSKPVVAIVSFRNVPAGQMQKIRREFRGK-AEIKVVKNTLLERALDALG-----GDYL    75
  RLA0_METKA MAVKAKGQPPSGYEPKVAEWKRREVKELKELMDEYENVGLVDLEGIPAPQLQEIRAKLRERDTIIRMSRNTLMRIALEEKLDER--PELE    88
  RLA0_METTH -------------MAHVAEWKKKEVQELHDLIKGYEVVGIANLADIPARQLQKMRQTLRDS-ALIRMSKKTLISLALEKAGREL--ENVD    74
  RLA0_METTL -------MITAESEHKIAPWKIEEVNKLKELLKNGQIVALVDMMEVPARQLQEIRDKIR-GTMTLKMSRNTLIERAIKEVAEETGNPEFA    82
  RLA0_METVA -------MIDAKSEHKIAPWKIEEVNALKELLKSANVIALIDMMEVPAVQLQEIRDKIR-DQMTLKMSRNTLIKRAVEEVAEETGNPEFA    82
  RLA0_METJA --------METKVKAHVAPWKIEEVKTLKGLIKSKPVVAIVDMMDVPAPQLQEIRDKIR-DKVKLRMSRNTLIIRALKEAAEELNNPKLA    81
  RLA0_PYRAB -------------MAHVAEWKKKEVEELANLIKSYPVIALVDVSSMPAYPLSQMRRLIRENGGLLRVSRNTLIELAIKKAAQELGKPELE    77
  RLA0_PYRHO -------------MAHVAEWKKKEVEELAKLIKSYPVIALVDVSSMPAYPLSQMRRLIRENGGLLRVSRNTLIELAIKKAAKELGKPELE    77
  RLA0_PYRFU -------------MAHVAEWKKKEVEELANLIKSYPVVALVDVSSMPAYPLSQMRRLIRENNGLLRVSRNTLIELAIKKVAQELGKPELE    77
  RLA0_PYRKO -------------MAHVAEWKKKEVEELANIIKSYPVIALVDVAGVPAYPLSKMRDKLR-GKALLRVSRNTLIELAIKRAAQELGQPELE    76
  RLA0_HALMA -----MSAESERKTETIPEWKQEEVDAIVEMIESYESVGVVNIAGIPSRQLQDMRRDLHGT-AELRVSRNTLLERALDDVD-----DGLE    79
  RLA0_HALVO -----MSESEVRQTEVIPQWKREEVDELVDFIESYESVGVVGVAGIPSRQLQSMRRELHGS-AAVRMSRNTLVNRALDEVN-----DGFE    79
  RLA0_HALSA -----MSAEEQRTTEEVPEWKRQEVAELVDLLETYDSVGVVNVTGIPSKQLQDMRRGLHGQ-AALRMSRNTLLVRALEEAG-----DGLD    79
  RLA0_THEAC -------------MKEVSQQKKELVNEITQRIKASRSVAIVDTAGIRTRQIQDIRGKNRGK-INLKVIKKTLLFKALENLGD----EKLS    72
  RLA0_THEVO -------------MRKINPKKKEIVSELAQDITKSKAVAIVDIKGVRTRQMQDIRAKNRDK-VKIKVVKKTLLFKALDSIND----EKLT    72
  RLA0_PICTO -------------MTEPAQWKIDFVKNLENEINSRKVAAIVSIKGLRNNEFQKIRNSIRDK-ARIKVSRARLLRLAIENTGK----NNIV    72
       ruler 1.......10........20........30........40........50........60........70........80........90
```

Multiple Alignment

- The alignment of two sequences is relatively straightforward.
- Can we generalize our Dynamic Programming approach to multiple sequences?

Multiple Alignment



V	S	N	_	S
_	S	N	A	_
_	_	_	A	S

Turns out the complexity in exponential in the number of sequences.

Optimal Multiple alignment

- What is a suitable cost measure?
- Sum of scores of pairwise alignments?
- Most of the available multiple alignment methods use a progressive approach that makes pairwise alignments, averages them into a consensus (actually a profile), then adds new sequences one at a time to the aligned set.

Optimal Multiple alignment

- There can be various rules for building the consensus: simple majority rules, plurality by a specific fraction.
- This is an approximate method! (why?)

Multiple alignment: ClustalW

- CLUSTAL is the most popular multiple alignment program
- Gap penalties can be adjusted based on specific amino acid residues, regions of hydrophobicity, proximity to other gaps, or secondary structure.
- It can re-align just selected sequences or selected regions in an existing alignment
- It can compute phylogenetic trees from a set of aligned sequences.

Editing Multiple Alignments

- There are a variety of tools that can be used to modify and display a multiple alignment.
- An editor can also be used to make modifications by hand to improve biologically significant regions in a multiple alignment created by an alignment program.
- Examples of such editors include MACAW, SeqVu, and GeneDoc.

Multiple Alignments

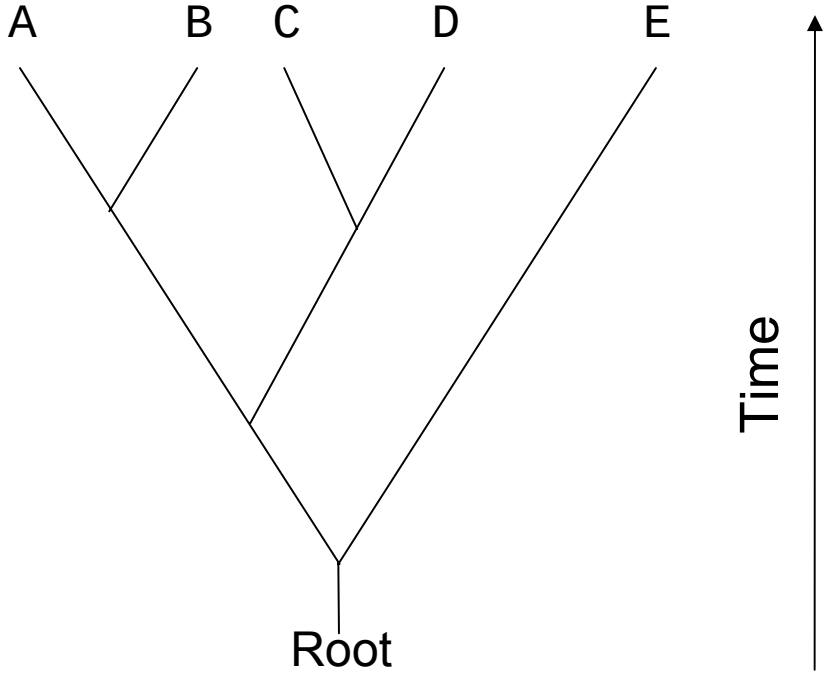
- Multiple Alignments are starting points for calculating phylogenetic trees
- Motifs and Profiles are calculated from multiple alignments

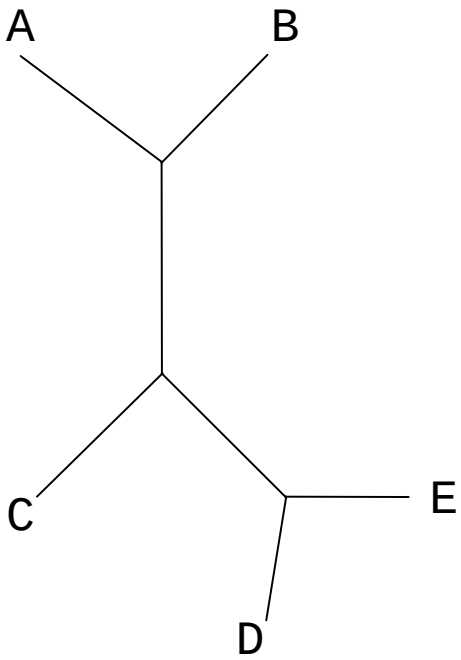
Phylogenetics

Phylogenetic Trees

Hypothesis about the relationship between organisms

Can be rooted or unrooted





Tree proliferation

$$N_R = \frac{(2n-3)!}{2^{n-2}(n-2)!}$$

$$N_U = \frac{(2n-5)!}{2^{n-3}(n-3)!}$$

Species	Number of Rooted Trees	Number of Unrooted Trees
2	1	1
3	3	1
4	15	3
5	105	15
6	34,459,425	2,027,025
7	213,458,046,767,875	7,905,853,580,625
Species	**Number of Rooted Trees**	**Number of Unrooted Trees**

Molecular phylogenetics

Specific genomic sequence variations (alleles) are much more reliable than phenotypic characteristics

More than one gene should be considered

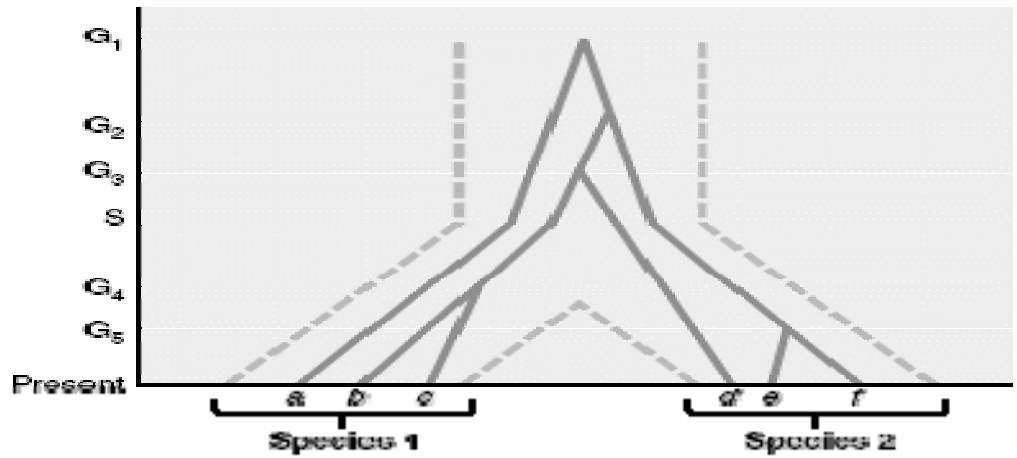


FIGURE 4.4 *Individuals may actually appear to be more closely related to members of a species other than their own when only one gene is considered. Gene divergence events (G_1 through G_5) often occur before as well as after speciation events (S). The evolutionary history of gene divergence resulting in the six alleles denoted a through f is shown in solid lines; speciation (i.e., population splitting) is shown by broken lines. Individual d would actually appear to be more closely related to individuals in species 1 if only this locus were considered even though it is a member of species 2.*

Distance matrix methods

```
                10         20         30         40         50
A:    GTGCTGCACGG  CTCAGTATA  GCATTTACCC  TTCCATCTTC  AGATCCTGAA
B:    ACGCTGCACGG  CTCAGTGCG  GTGCTTACCC  TCCCATCTTC  AGATCCTGAA
C:    GTGCTGCACGG  CTCGGCGCA  GCATTTACCC  TCCCATCTTC  AGATCCTATC
D:    GTATCACACGA  CTCAGCGCA  GCATTTGCCC  TCCCGTCTTC  AGATCCTAAA
E:    GTATCACATAG  CTCAGCGCA  GCATTTGCCC  TCCCGTCTTC  AGATCTAAAA
```

FIGURE 4.5 *A five-way alignment of homologous DNA sequences.*

Species	**A**	**B**	**C**	**D**
B	9	–	–	–
C	8	11	–	–
D	12	15	10	–
Species	**A**	**B**	**C**	**D**

UPGMA

Similar to average-link clustering

Merge the closest two groups

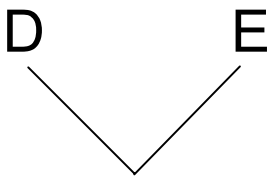
Replace the distances for the new, merged group with the average of the distance for the previous two groups

Repeat until all species are joined

UPGMA Step 1

Species	A	B	C	D
B	9	–	–	–
C	8	11	–	–
D	12	15	10	–
Species	**A**	**B**	**C**	**D**

Merge D & E

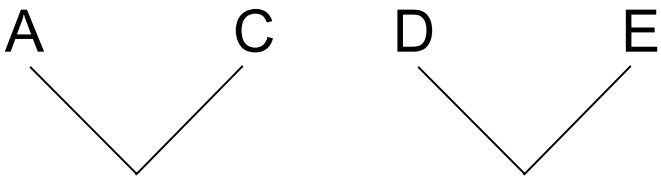


Species	A	B	C
B	9	–	–
C	8	11	–
Species	**A**	**B**	**C**

UPGMA Step 2

Species	A	B	C
B	9	–	–
C	8	11	–
Species	**A**	**B**	**C**

Merge A & C



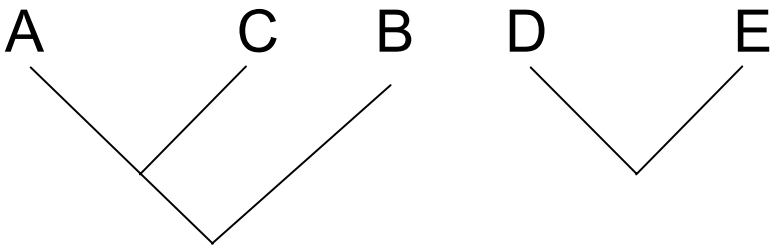
Species	B	AC
AC	10	–
Species	**B**	**AC**

UPGMA Steps 3 & 4

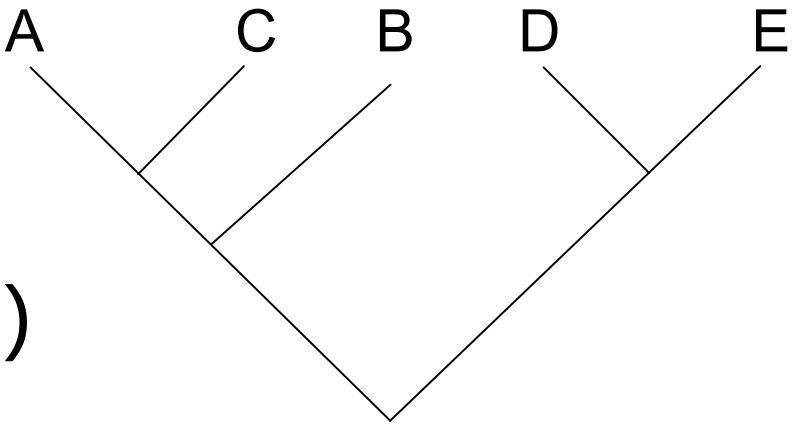
Species	B	AC
AC	10	–

Species	B	AC

Merge B & AC



Merge ABC & DE



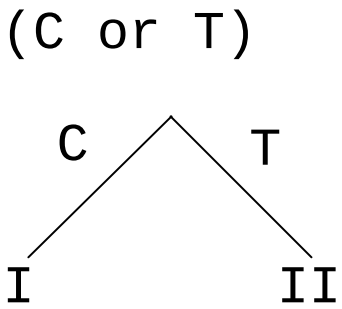

```
(((A,C)B)(D,E)
)
```

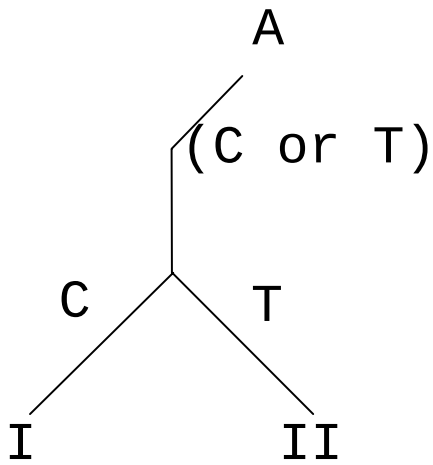
Parsimony approaches

Belong to the broader class of <u>character based</u> methods of phylogenetics

Emphasize simpler, and thus more likely evolutionary pathways

```
I:  G**C**GGACG
II: G**T**GGACG
```





Parsimony methods

Enumerate all possible trees

Note the number of substitutions events invoked by each possible tree

Can be weighted by transition/transversion probabilities, etc.

Select the most parsimonious

Branch and Bound methods

Key problem – number of possible trees grows enormous as the number of species gets large

Branch and bound – a technique that allows large numbers of candidate trees to be rapidly disregarded

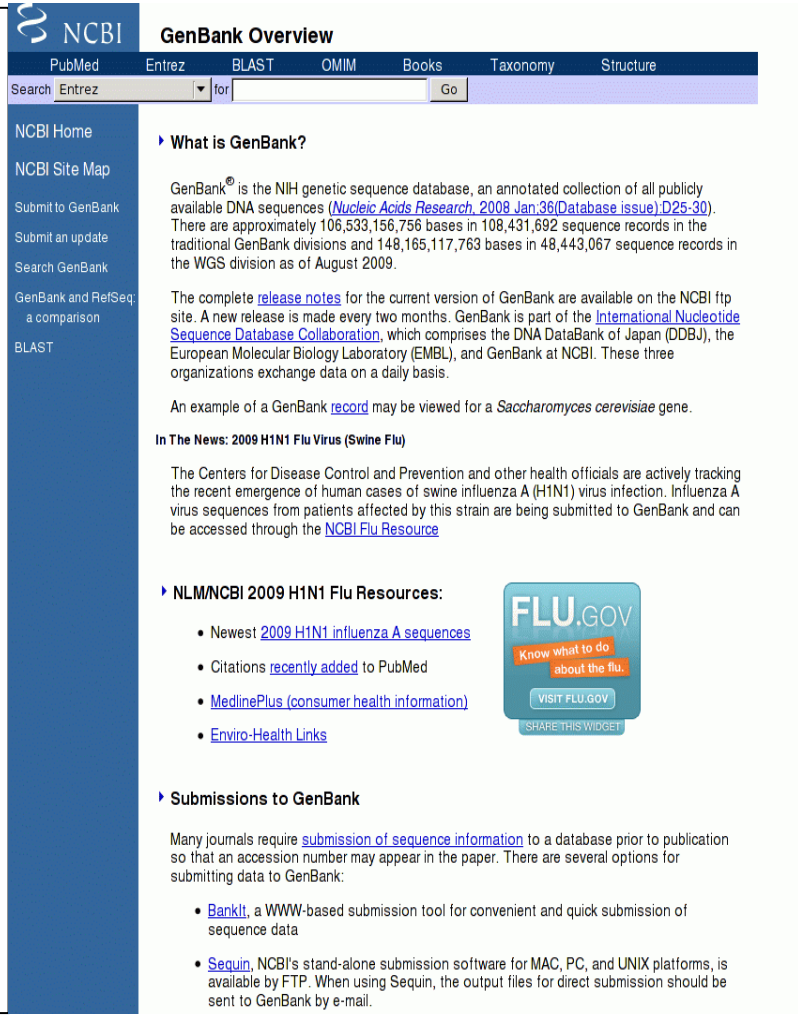
Requires a “good guess” at the cost of the best tree

Parsimony – Branch and Bound

- Use the UPGMA tree for an initial best estimate of the minimum cost (most parsimonious) tree
- Use branch and bound to explore all feasible trees
- Replace the best estimate as better trees are found
- Choose the most parsimonious

Online Resources

- Genbank: http://www.ncbi.nlm.nih.gov/genbank/
- NIH genetic sequence database, an annotated collection of all publicly available sequences



NCBI

GenBank Overview

PubMed Entrez BLAST OMIM Books Taxonomy Structure

Search Entrez for Go

NCBI Home
NCBI Site Map
Submit to GenBank
Submit an update
Search GenBank
GenBank and RefSeq: a comparison
BLAST

What is GenBank?

GenBank® is the NIH genetic sequence database, an annotated collection of all publicly available DNA sequences (*Nucleic Acids Research*. 2008 Jan;36(Database issue):D25-30). There are approximately 106,533,156,756 bases in 108,431,692 sequence records in the traditional GenBank divisions and 148,165,117,763 bases in 48,443,067 sequence records in the WGS division as of August 2009.

The complete release notes for the current version of GenBank are available on the NCBI ftp site. A new release is made every two months. GenBank is part of the International Nucleotide Sequence Database Collaboration, which comprises the DNA DataBank of Japan (DDBJ), the European Molecular Biology Laboratory (EMBL), and GenBank at NCBI. These three organizations exchange data on a daily basis.

An example of a GenBank record may be viewed for a *Saccharomyces cerevisiae* gene.

In The News: 2009 H1N1 Flu Virus (Swine Flu)

The Centers for Disease Control and Prevention and other health officials are actively tracking the recent emergence of human cases of swine influenza A (H1N1) virus infection. Influenza A virus sequences from patients affected by this strain are being submitted to GenBank and can be accessed through the NCBI Flu Resource

NLM/NCBI 2009 H1N1 Flu Resources:

- Newest 2009 H1N1 influenza A sequences
- Citations recently added to PubMed
- MedlinePlus (consumer health information)
- Enviro-Health Links

FLU.GOV
Know what to do about the flu.
VISIT FLU.GOV
SHARE THIS WIDGET

Submissions to GenBank

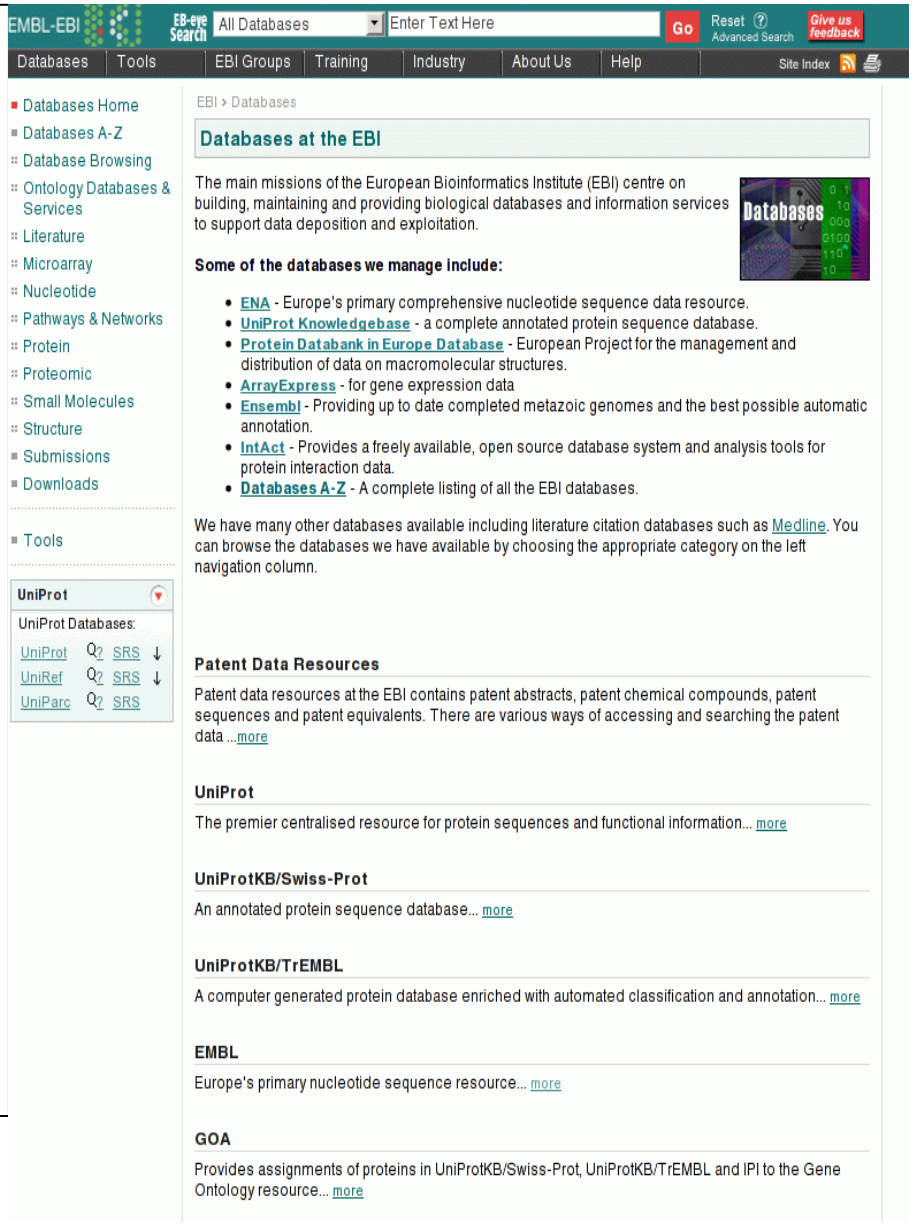
Many journals require submission of sequence information to a database prior to publication so that an accession number may appear in the paper. There are several options for submitting data to GenBank:

- BankIt, a WWW-based submission tool for convenient and quick submission of sequence data
- Sequin, NCBI's stand-alone submission software for MAC, PC, and UNIX platforms, is available by FTP. When using Sequin, the output files for direct submission should be sent to GenBank by e-mail.

Online Resources: EBI

European Bioinformatics Institute

- http://www.ebi.ac.uk/Databases/
- ENA - Europe's primary comprehensive nucleotide sequence data resource.
- UniProt Knowledgebase - a complete annotated protein sequence database.
- Protein Databank in Europe Database - European Project for



Online Resources: DDBJ

DDBJ: DNA Data Bank of Japan

- `http://www.ddbj.nig.ac.jp/`
- Data exchange with EMBL/EBI, GenBank on a daily basis.
- Data across these databases is virtually identical (modulo curation practices)
- Virtually all sequence data in Japan is submitted through DDBJ.

Online Resources: SwissProt/UniProt

- `http://ca.expasy.org/sprot/sprot-top.html`
- Curated protein sequence databases
- UniProt/TrEMBL: annotated supplement to UniProt of EMBL nucleotides

Online Resources: Structure Databases

- PDB: http://www.rcsb.org/pdb/home/home.do Experimental structures of proteins, nucleic acids and assemblies
- NDB: http://ndbserver.rutgers.edu/ Nucleic acid structures
- SCOP: http://scop.mrc-lmb.cam.ac.uk/scop/ Structural classification of proteins
- Cambridge Structure Database: http://www.ccdc.cam.ac.uk/ structure, visualization and analysis of organic molecules and metal-organic structures

Online Resources

- Motifs in protein structure and/or function
  PROSITE http://ca.expasy.org/prosite/
- Function
  EC Enzyme database http://ca.expasy.org/enzyme/
- Integrated databases
  WIT
  Entrez http://www.ncbi.nlm.nih.gov/sites/gquery

Online Resources

Fetching the sequences

- BLAST Search http://blast.ncbi.nlm.nih.gov/Blast.cgi
- Genbank Database Query Form at NCBI
- Entrez at NCBI
- Batch downloads via Batch Entrez
- NCSA Biology Workbench