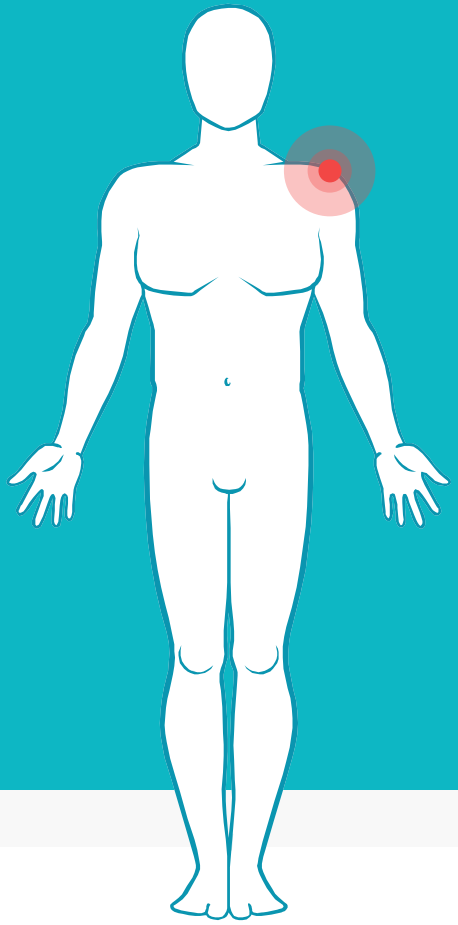
Gene Duplication and Read Mapping

Week 7

Department of CSE, DIU



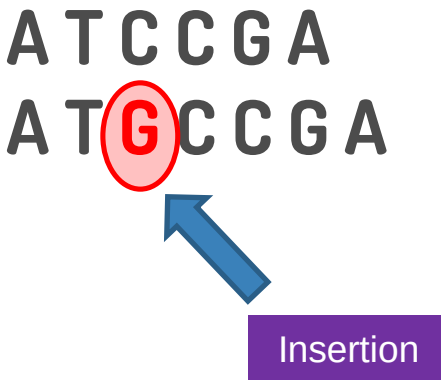
CONTENTS

1. DNA Mutation

What and how mutation occurs, common forms

Mutation

DNA Mutation refers to sudden, random changes in DNA sequences which leads to different phenotypic expressions.



Common Mutation Types

Substitution

AATTCGCA

AATGCGCA

Deletion

AATTCGCA

AATCGCA

Inversion

AATCGCA → AACGGCA

AGCATCG → ACTATCG

Duplication

AATCGCA

AATCATCGCA

Insertion

AATCGCA

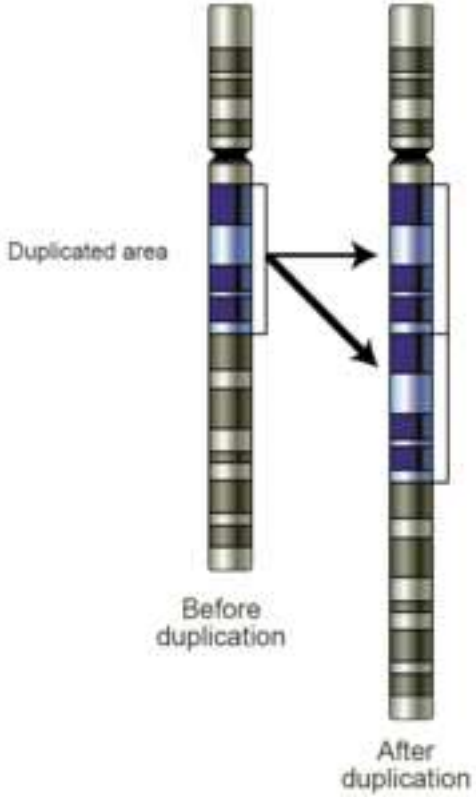
AATTCGCA

2. Gene Duplication

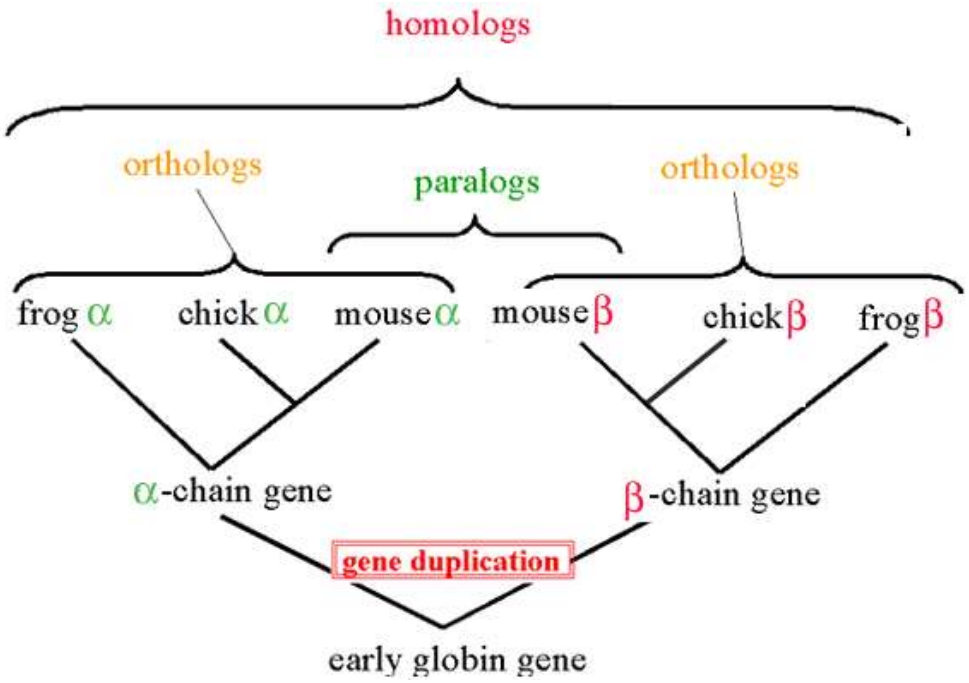
Duplication of Genes, Homolog, Ortholog, Paralogs

Gene Duplication

Gene duplication (or chromosomal duplication or gene amplification) is a major mechanism through which new genetic material is generated during molecular evolution. It can be defined as any duplication of a region of DNA that contains a gene.



Homolog, Ortholog, Paralog and Speciation



- Homolog – A gene related to a second gene by descent from a common ancestral DNA sequence
- Ortholog – Orthologs are genes in different species that evolved from a common ancestral gene by speciation*
- Paralog – Paralogs are genes related by duplication within a genome
- Speciation* – Speciation is the origin of a new species capable of making a living in a new way from the species from which it arose

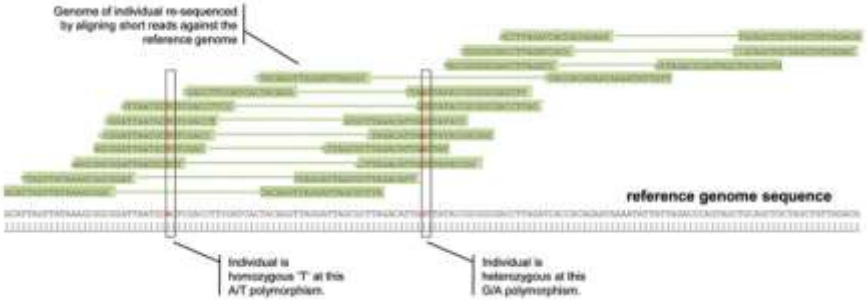
3. Read Mapping

Short Read Mapping, Genome Indexing

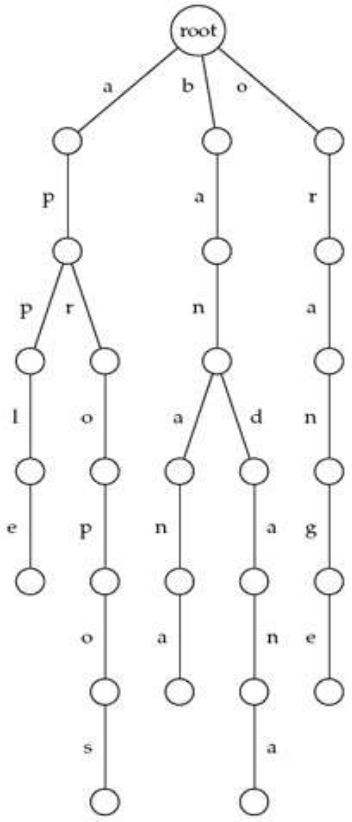
Read Mapping

Mapping refers to the process of aligning short reads to and finding the starting position in a reference sequence (typically Genome).

Short read generally are reads with a length of 30-350 base pairs.



Genome Indexing (Keyword Tree)



- Stores a set of keywords in a rooted labeled tree.
- Each edge is labeled with a letter from an alphabet.
- Any two edges coming out of the same vertex have distinct labels.
- Every keyword stored can be spelled on a path from root to some leaf.
- Furthermore, every path from root to leaf gives a keyword.

Keywords

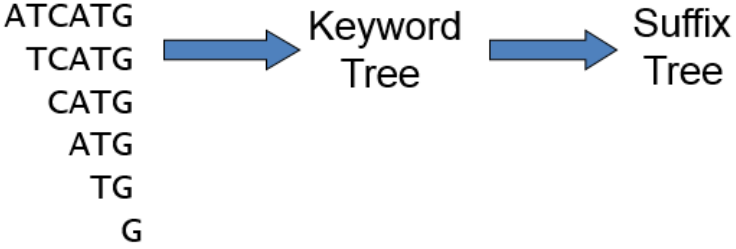
- Apple
- Apropos
- Banana
- Bandana
- Orange

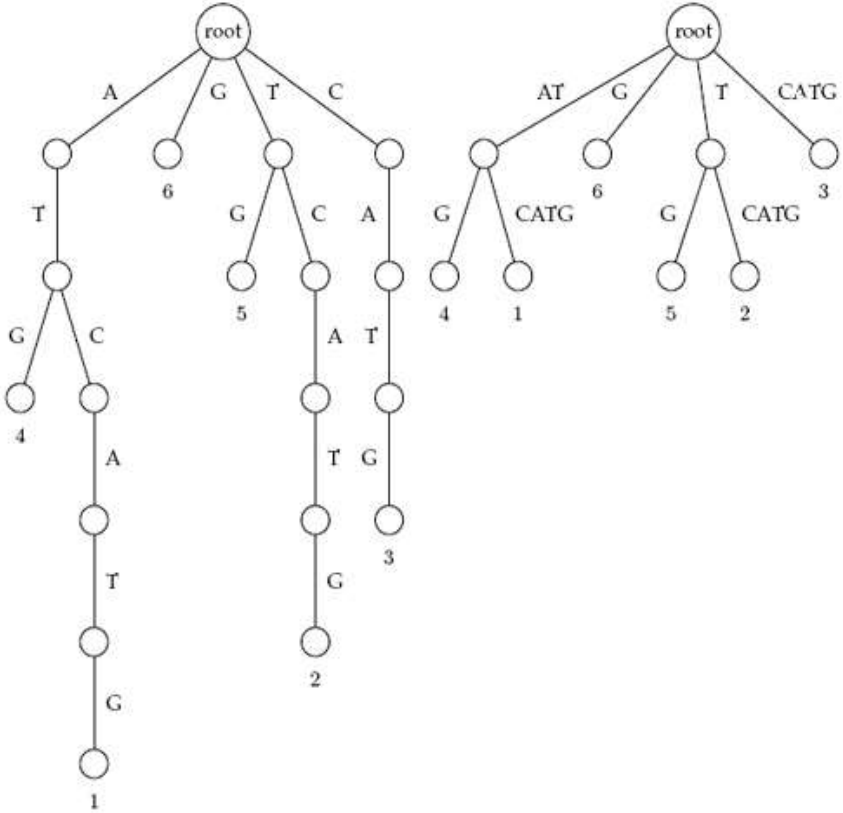
Genome Indexing (Suffix Tree)

- **Similar to Keyword Tree**
- Suffixes of the text are keywords
- Edges that form paths are collapsed
- Each edge is labeled with a substring of the text
- All internal edges have at least two outgoing edges.
- Leaves are labeled by the index of the pattern.

Suffix tree of <u>ATCATG</u>

ATCATG
TCATG
CATG
ATG
TG
G





Genome Indexing (Suffix Array)

1 ATCATG$

2 TCATG$

3 CATG$

4 ATG$

5 TG$

6 G$

7 $

Sort the suffixes lexicographically →

7 $

1 ATCATG$

4 ATG$

3 CATG$

6 G$

2 TCATG$

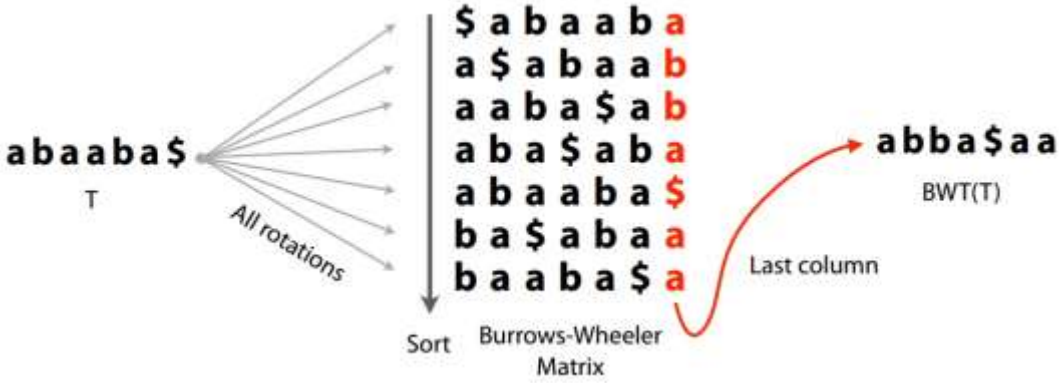
5 TG$

- **More space efficient than suffix tree**
- Suffix tree index for human genome is about 47 GB
- Lexicographically sort all the suffixes
- Store the starting indices of the suffixes along with the original string

Generate Suffix Array of <u>ATCATG</u>

Genome Indexing (Burrows Wheeler Transform)

- Given Sequence – **abaaba**
- Add **$** as ending notation – **abaaba$**
- By Shifting each alphabet to the right once, generate all the rotations
- Lexicographically Sort all the rotations
- The very last column will be denoted as BWT (T)



Genome Indexing (Burrows Wheeler Transform)

$ a b a a b a
a $ a b a a b
a a b a $ a b
a b a $ a b a
a b a a b a $
b a $ a b a a
b a a b a $ a

BWM(T)

6	$
5	a $
2	a a b a $
3	a b a $
0	a b a a b a $
4	b a $
1	b a a b a $

SA(T)

- Given Sequence – **abaaba**
- Add **$** as ending notation – **abaaba$**
- Lexicographically sorted all rotations will generate BWT Matrix which will be denoted as BWM (T)
- Suffix Array generated from all the rotations will be called SA (T)
- BWM can be derived from any given BWT (T)

Genome Indexing (Burrows Wheeler Transform)

LF (Last to First) Mapping

- Generate Burrows Wheeler Matrix for a given sequence
- Assign numbers to distinguish same characters
- Assign the numbers in a ascending manner for each character

F						L
$	a_3	b_1	a_1	a_2	b_0	a_0
a_0	$	a_3	b_1	a_1	a_2	b_0
a_1	a_2	b_0	a_3	$	a_3	b_1
a_2	b_0	a_0	$	a_3	b_1	a_1
a_3	b_1	a_1	a_2	b_0	a_0	$
b_0	a_0	$	a_3	b_1	a_1	a_2
b_1	a_1	a_2	b_0	a_0	$	a_3

Ascending rank

Genome Indexing (Burrows Wheeler Transform)

	F	L
Start →	$	a_0
	a_0	b_0
	a_1	b_1 ←
	a_2	a_1
	a_3	$
	b_0	a_2
row 6 →	b_1	a_3

<u>Find out the row starting with b1 using LF Mapping</u>

1. Start from the row containing **$** in the First Column
2. Find out what's in Last Column of that row (here its $\mathbf{a_0}$)
3. Compare it with query **(b_1)**

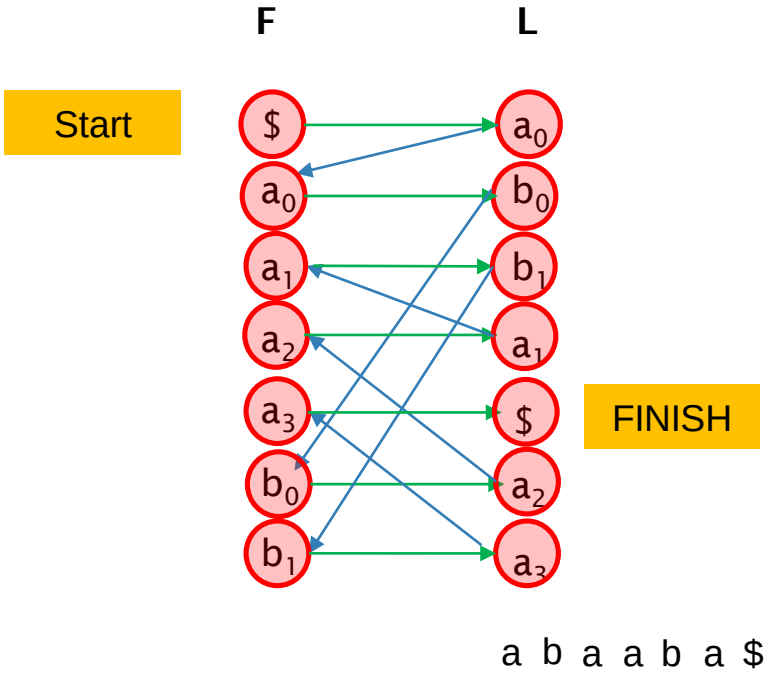
1. If MATCH, then
   - Find **b1** in First Column
   - Print row number
   - Terminate

1. If No MATCH, then
   - Find the row with that element in the First column
   - Go to Step 2 and Repeat

Genome Indexing (Burrows Wheeler Transform)

Find Original Gene using LF Mapping if BWT (T) is Given

1. Original Gene = **abaaba** (Not Given)
2. Given BWT (T) = **abba$aa**
3. Store it as Last Column
4. Draw the First Column by sorting the elements of Last Column Lexicographically
5. Assign numbers to distinguish characters in an ascending manner
6. Start LF Mapping from Starting Element ($)
7. For each element found in the **LAST** column, write it from right to left



Whales and Dolphins

Their ancestors had back legs once, they could walk

Birds came from Dinosaurs

And they both descended from Reptiles

Humans have tails

While they are inside the womb! It dissolves eventually.

Bacterium

All livings beings can be traced back to a bacterium