AMRITA VISHWA VIDYAPEETHAM
DEEMED TO BE UNIVERSITY

22BIO211: Intelligence of Biological Systems - 2

Exploding Newspaper Problem AND The String Reconstruction Problem

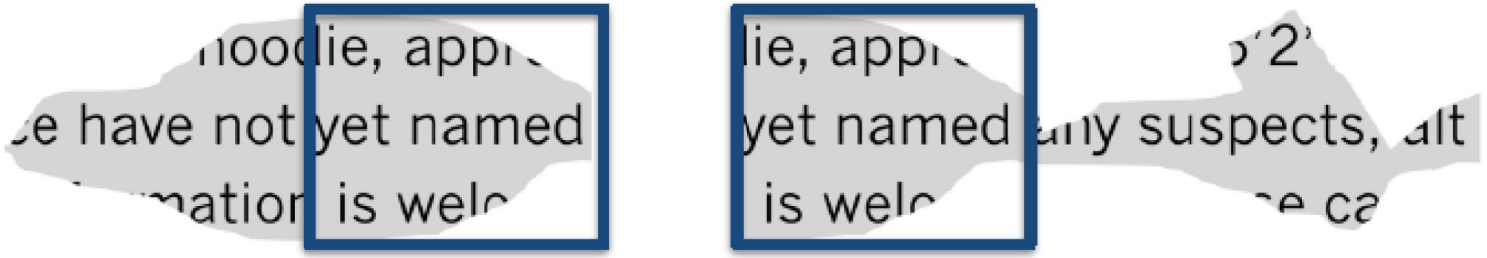
Dr. Manjusha Nair M
Amrita School of Computing, Amritapuri

Email : manjushanair@am.amrita.edu
Contact No: 9447745519

Exploding News Paper Problem

Exploding News Paper Problem

An Overlapping Puzzle



Genome Sequencing and Assembling – An Analogy

Multiple Copies of a Genome

CTGATGATGGACTACGCTACTACTGCTAGCTGTATTACGATCAGCTACCACATCGTAGCTACGATGCATTAGCAAGCTATCGGATCAGCTACCACATCGTAGC
CTGATGATGGACTACGCTACTACTGCTAGCTGTATTACGATCAGCTACCACATCGTAGCTACGATGCATTAGCAAGCTATCGGATCAGCTACCACATCGTAGC
CTGATGATGGACTACGCTACTACTGCTAGCTGTATTACGATCAGCTACCACATCGTAGCTACGATGCATTAGCAAGCTATCGGATCAGCTACCACATCGTAGC
CTGATGATGGACTACGCTACTACTGCTAGCTGTATTACGATCAGCTACCACATCGTAGCTACGATGCATTAGCAAGCTATCGGATCAGCTACCACATCGTAGC

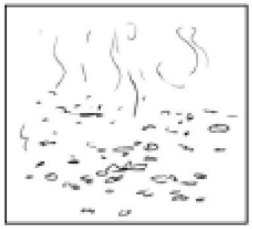
Breaking the Genomes at Random Positions

Genome Sequencing and Assembling – An Analogy

Generating "Reads"

CTGATGA TGGACTACGCTAC TACTGCTAG CTGTATTACG ATCAGCTACCACA TCGTAGCTACG ATGCATTAGCAA GCTATCGGA TCAGCTACCA CATCGTAGC
CTGATGATG GACTACGCT ACTACTGCTA GCTGTATTACG ATCAGCTACC ACATCGTAGCT ACGATGCATTA GCAAGCTATC GGATCAGCTAC CACATCGTAGC
CTGATGATGG ACTACGCTAC TACTGCTAGCT GTATTACGATC AGCTACCAC ATCGTAGCTACG ATGCATTAGCA AGCTATCGG A TCAGCTACCA CATCGTAGC
CTGATGATGGACT ACGCTACTACT GCTAGCTGTAT TACGATCAGC TACCACATCGT AGCTACGATGCA TTAGCAAGCT ATCGGATCA GCTACCACATC GTAGC

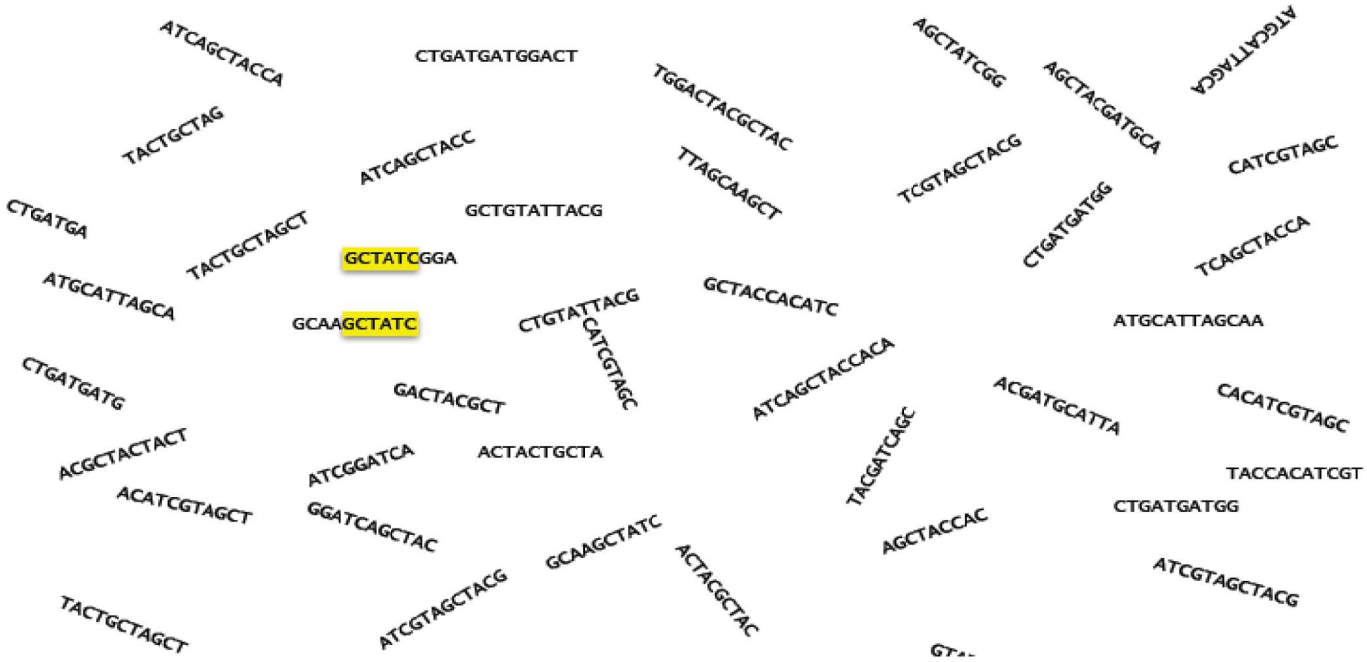
"Burning" Some Reads



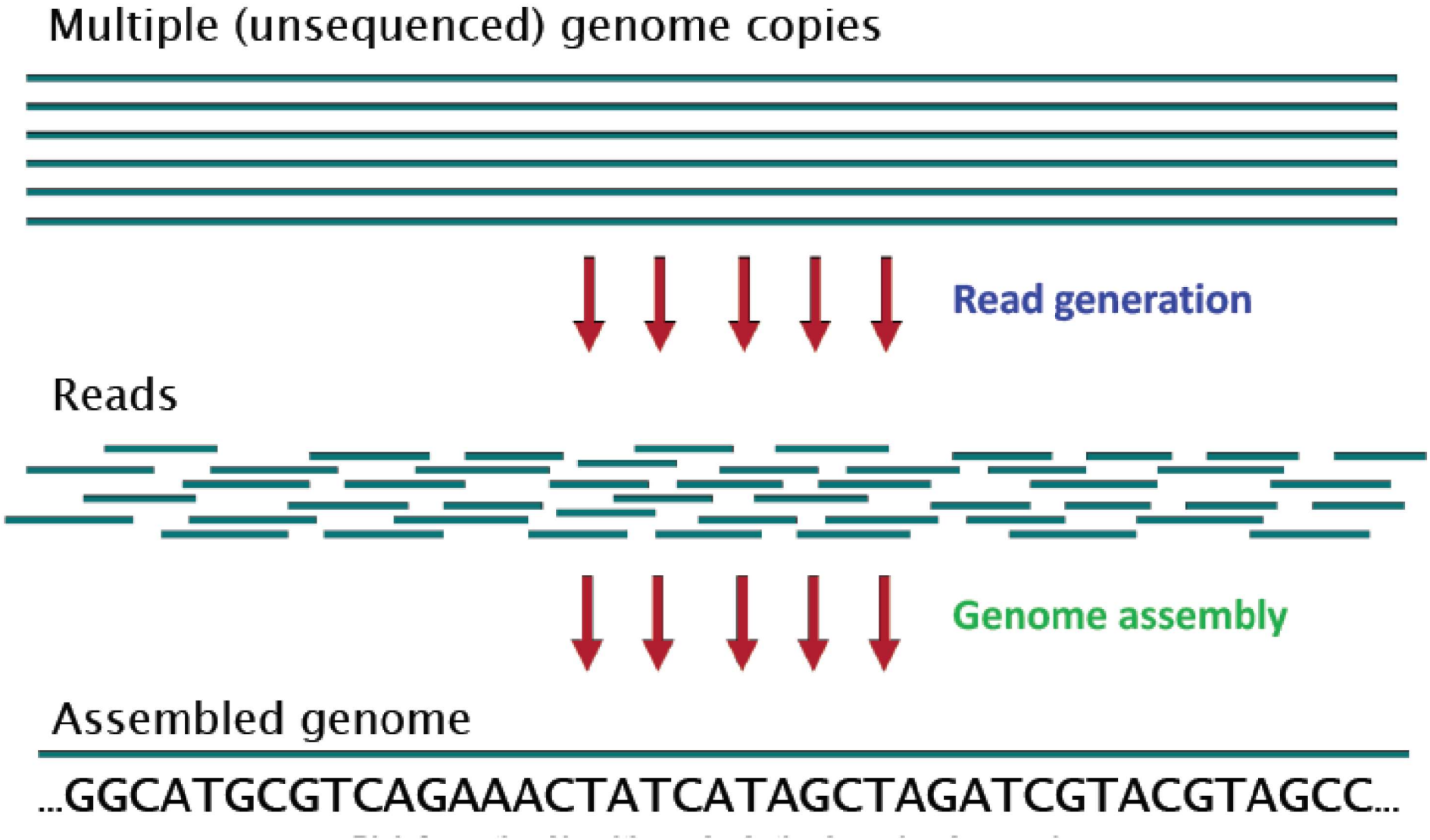
CTGATGA TGGACTACGCTAC **TACTGCTAG CTGTATTACG** ATCAGCTACCACA **TCGTAGCTACG ATGCATTAGCAA GCTATCGGA TCAGCTACCA CATCGTAGC**
CTGATGATG GACTACGCT ACTACTGCTA GCTGTATTACG **ATCAGCTACC ACATCGTAGCT** ACGATGCATTA **GCAAGCTATC GGATCAGCTAC CACATCGTAGC**
CTGATGATGG ACTACGCTAC **TACTGCTAGCT GTATTACGATC AGCTACCAC ATCGTAGCTACG ATGCATTAGCA AGCTATCGG A** TCAGCTACCA **CATCGTAGC**
CTGATGATGGACT ACGCTACTACT GCTAGCTGTAT **TACGATCAGC TACCACATCGT AGCTACGATGCA** TTAGCAAGCT **ATCGGATCA GCTACCACATC GTAGC**

Genome Sequencing and Assembling – An Analogy

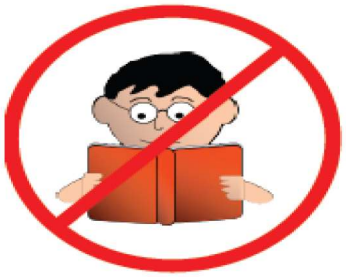
No Idea What Position Every Read Comes From



From Experimental to Computational Challenges



What makes genome sequencing difficult?



- Short DNA fragments/ reads
  - Cannot read an entire genome one nucleotide at a time from beginning to end

- Overlapping reads
  - The genome assembly is not the same as a jigsaw puzzles, but is a giant **overlap puzzle!**

Is Genome Assembling Same as Exploding Newspaper Problem?

- DNA is double-stranded
  - Do we need to use a strand or its reverse complement?
- Sequencing errors
- Part of the genome not covered by any reads
- **Genome Assembling is more difficult than Newspaper Problem**
  - Existence of reverse complement.
  - Sequencing Errors.
  - Lost sequences

Towards a computational Problem

- Reads generated by modern sequencers often have the same length.
  - Reads are all k-mers for some value of k.
- **Assume an ideal but unrealistic situation**
- Assumptions
  - All reads come from the **same strand.**
  - Reads have **no errors.**
  - Reads exhibit **perfect coverage.**
- So, every k-mer substring of the genome is generated as a read.

What is a K-mer Composition?

- Given a string Text, its k-mer composition **$\mathbf{COMPOSITION_k(Text)}$** is the collection of all k-mer substrings of Text (including repeated k-mers).

$\mathbf{COMPOSITION_3(TATGGGGTGC)}$ =
{ATG, GGG, GGG, GGT, GTG, TAT, TGC, TGG}.

$Composition_3$(TAATGCCATGGGATGTT) =

TAA
AAT
ATG
TGC
GCC
CCA
CAT
ATG
TGG
GGG
GGA
GAT
ATG
TGT
GTT

Listed k-mers in **lexicographic order**

String Composition Problem:

String Composition Problem:
Generate the k-mer composition of a string.

Input: A string *Text* and an integer k.
Output: $\text{COMPOSITION}_k(Text)$, where the k-mers are arranged in lexicographic order.

```
def stringComposition(Text,k):
    kmerArray=[ ]
        for i in range(len(Text - k)):
            kmer = Text(i:i+k)
            kerArray.append(kmer)
        kmerArray.sort()
return kmerArray
```

Input :
5,CAATCCAAC
Output:
AATCC
ATCCA
CAATC
CCAAC
TCCAA

String Reconstruction Problem

- What is Genome Sequencing Problem?

Genome Sequencing Problem. Reconstruct a genome from reads.

- **Input.** A collection of strings *Reads*.
- **Output.** A string *Genome* reconstructed from *Reads*.

String Reconstruction Problem:
Reconstruct a string from its k-mer composition.

Input: An integer k and a collection *Patterns* of k-mers.
Output: A string *Text* with k-mer composition equal to *Patterns* (if such a string exists).

The Naïve Approach: String Reconstruction : An Example

***K*-mer composition (Reads):** AAT ATG GTT TAA TGT

Approach : "connect" a pair of k-mers if they overlap in k-1 symbols

Starting String : The string should start with TAA – Why?

Steps :

TAA
AAT

TAA
AAT
ATG

TAA
AAT
ATG
TGT

TAA
AAT
ATG
TGT
GTT

Reconstructed String : TAATGTT

Another Example

Step: 1

AAT ATG ATG ATG CAT CCA GAT GCC GGA GGG GTT TAA TGC TGG TGT

Step: 2

AAT ATG ATG ATG CAT CCA GAT GCC GGA GGG GTT TGC TGG TGT

TAA

Step: 3

ATG ATG ATG CAT CCA GAT GCC GGA GGG GTT TGC TGG TGT

TAA
AAT

Another Example

Step: 4

Step: 5

What is next?

ATG ATG CAT CCA GAT GCC GGA GGG TGC TGG

TAA
AAT
ATG
TGT
GTT

No 3-mers in the composition start with TT!

One Solution - Backtracking

TAA
AAT
ATG
TGC

String Reconstructed :

TAA
AAT
ATG
TGC
GCC
CCA
CAT
ATG
TGG
GGA
GAT
ATG
TGT
GTT

TAATGCCATGGATGTT

We have only used fourteen of the fifteen 3-mers in the composition (we omitted GGG).

String Reconstruction Problem: In Python

- String Reconstruction from its genome path
- Genome path
  - last *k - 1* symbols of $Pattern_i$ are equal to the first *k - 1* symbols of $Pattern_{i+1}$ for *i* from 1 to n-1.

```
def stringReconstruction(kmers):
    string=kmer[0]
        for i in range(1,len(kmers)):
              string += kmers[i][-1]
   return string
```

Sample Input :

```
ACCGA
CCGAA
CGAAG
GAAGC
AAGCT
```

Sample Output:

```
ACCGAAGCT
```

Summary

- Genome Assembling is more difficult than Exploding Newspaper Problem
- Assumptions
- Computational Approaches
  - String composition problem : Example and Python Script
  - String reconstruction problem :Example and Python Script
- String Reconstruction Problem
  - Example 1
  - Example 2