DNA Sequencing and Data Analysis

Prof Noam Shomron

Amit Levon

Lecture 7, June 5, 2025

DNA Sequencing and Data Analysis

The De-novo Shotgun Assembly Problem
RNA-seq

Thursday 18:30 to 21:00

C.L03

nshomron@gmail.com

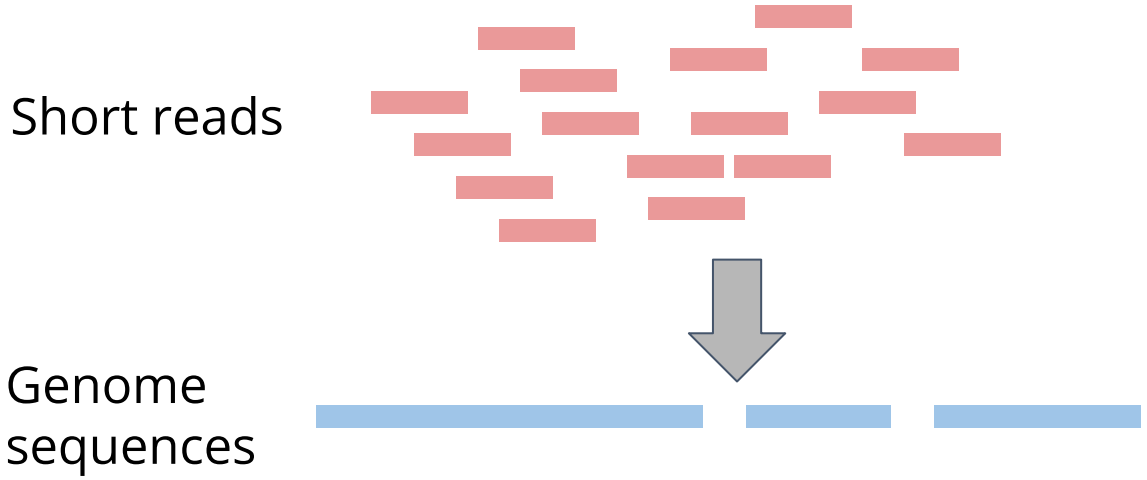
amit.levon@post.runi.ac.il

What is De Novo Genome Assembly?

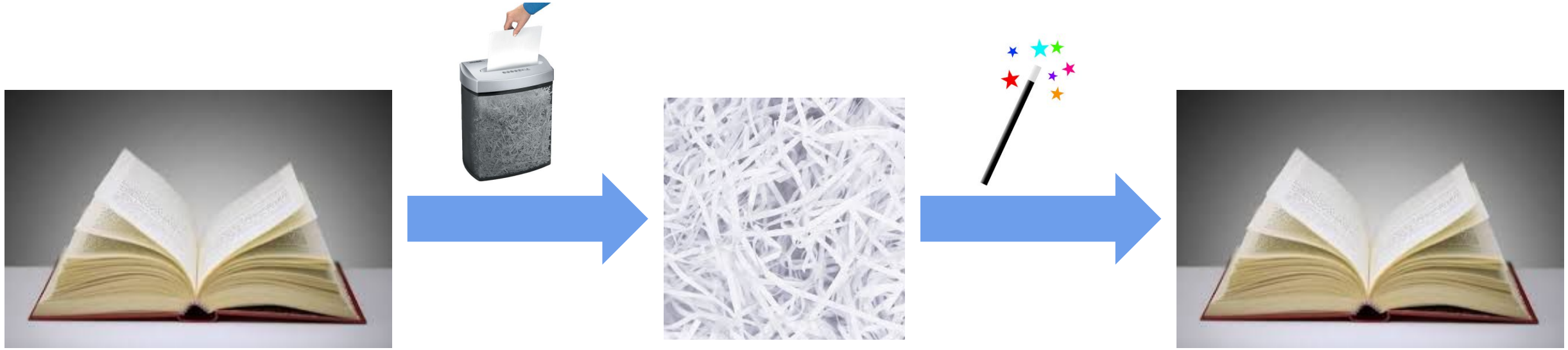
Genome assembly - constructing long genomic sequences from shorter ones

De novo = “from scratch”

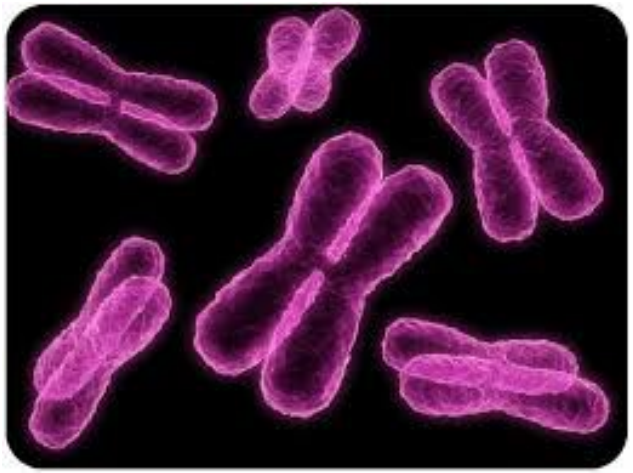
In NGS context - short reads ⟶ whole genome, without any external reference



The Assembly Problem

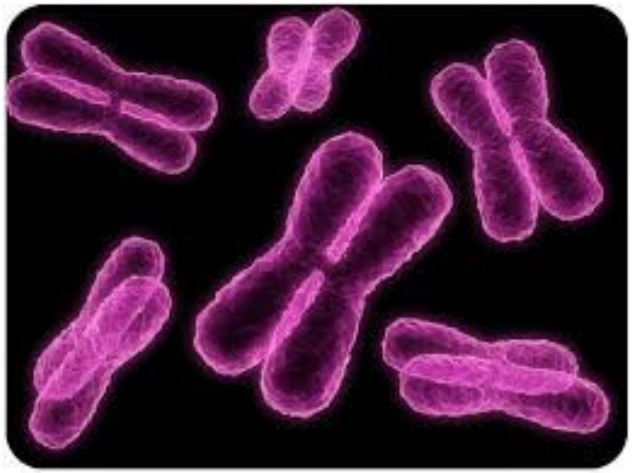


The Assembly Problem



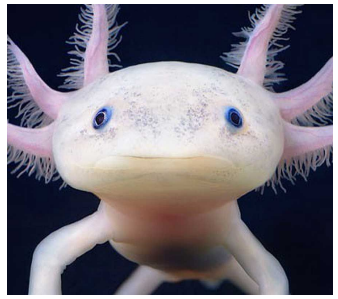
Shotgun sequencing

De novo assembly



Why Do We Need De Novo Assembly?

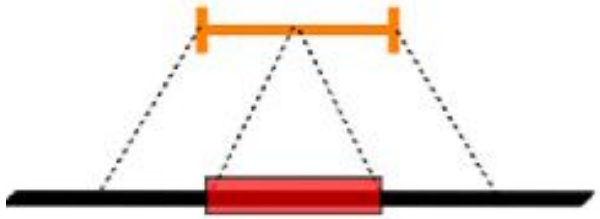
Completely new organism

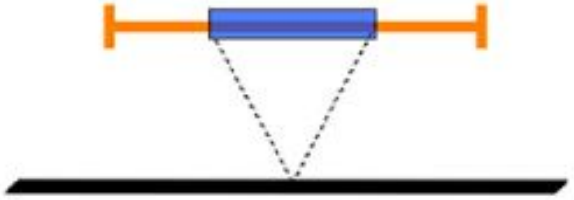


Existing reference is too different from what we are interested in

Cancer genomics

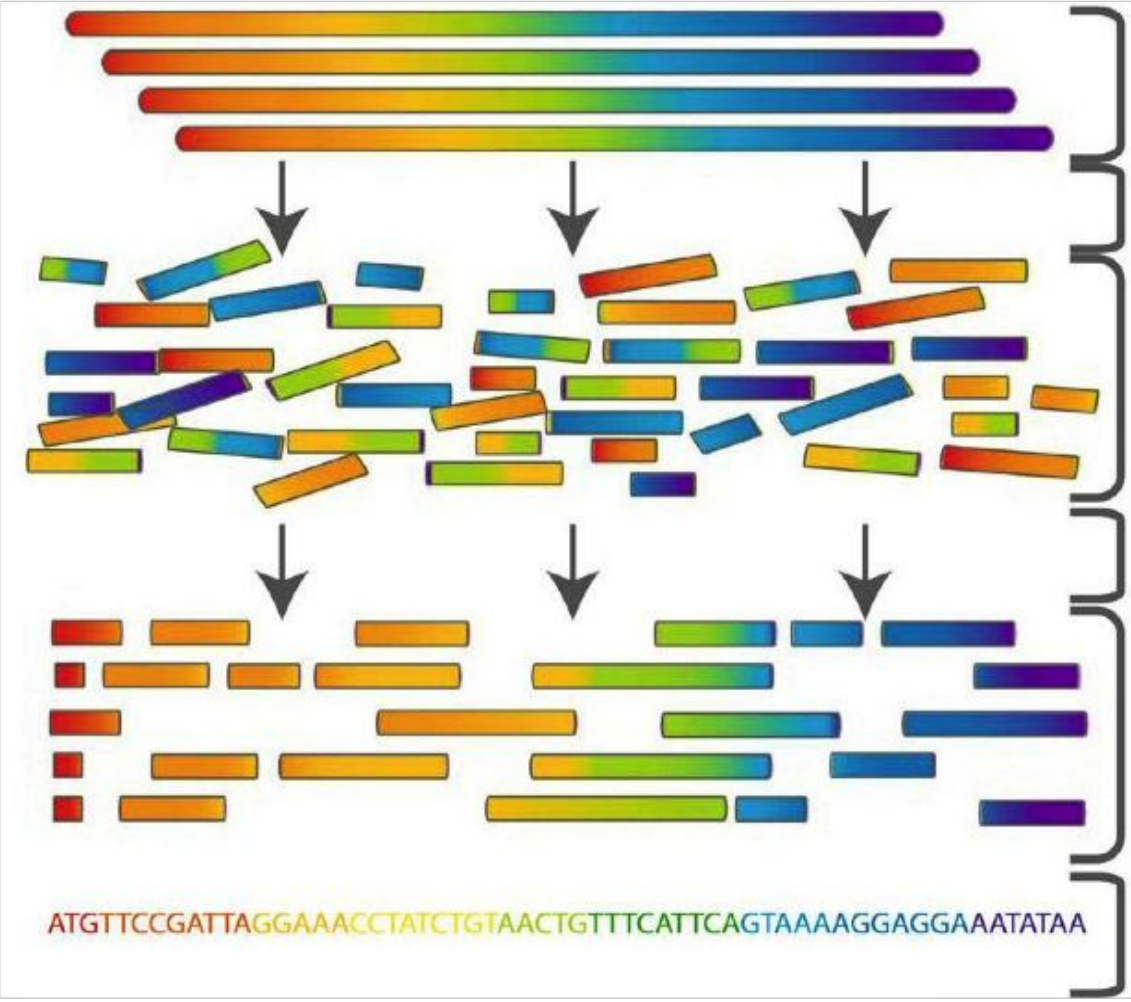
Identify large structural variation





Detect novel sequences not present in the reference

Genome Assembly by Reads Overlap



Genomic DNA

Fragmentation + Sequencing

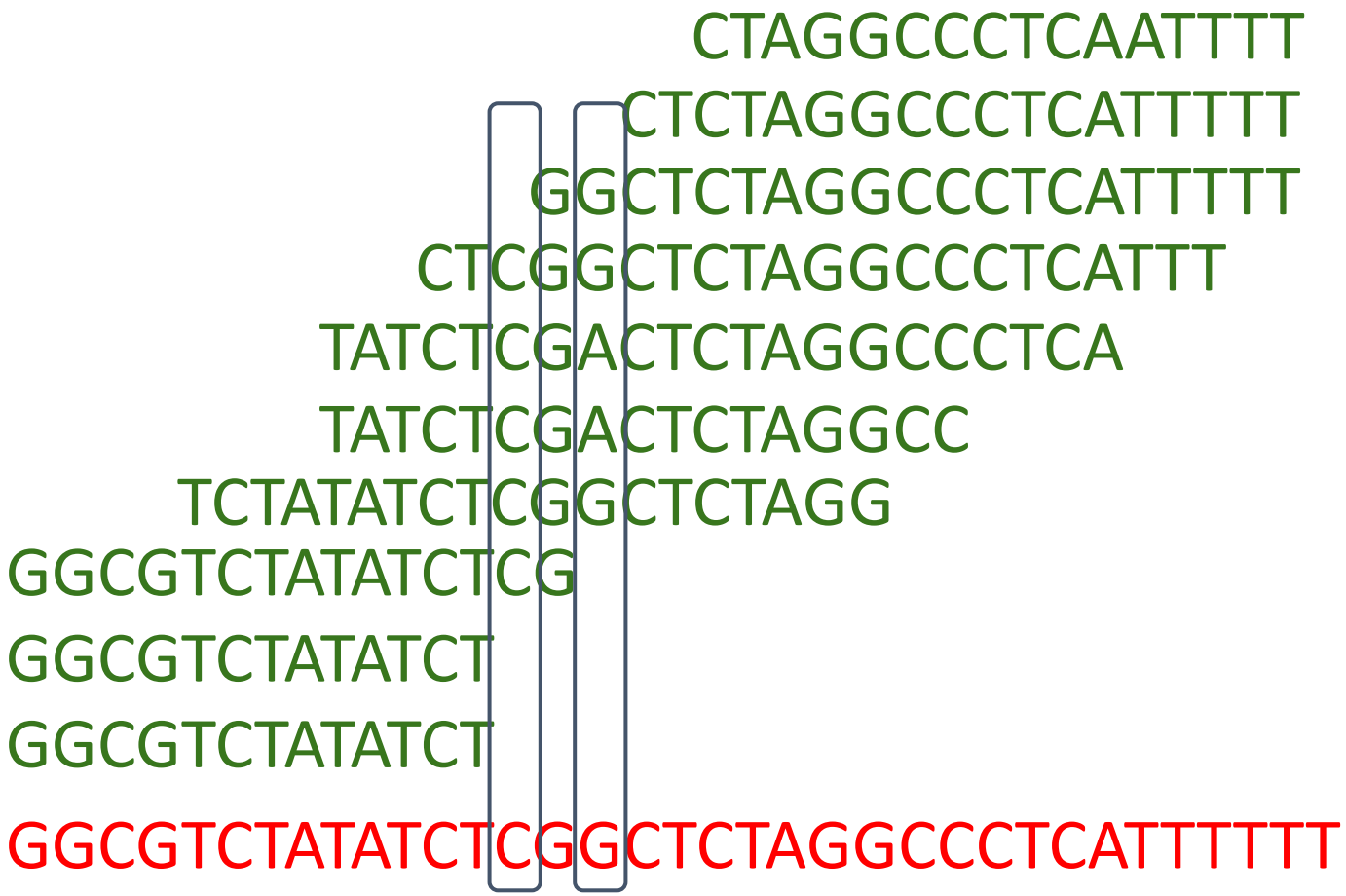
Sequence reads

Assembly

Connection between reads found

Consensus sequence

Coverage Definition



Coverage = 5

Average Coverage

CTAGGCCCTCAATTTT
CTCTAGGCCCTCATTTTT
GGCTCTAGGCCCTCATTTTT
CTCGGCTCTAGGCCCTCATTT
TATCTCGACTCTAGGCCCTCA
TATCTCGACTCTAGGCC
TCTATATCTCGGCTCTAGG
GGCGTCTATATCTCG
GGCGTCTATATCT
GGCGTCTATATCT
GGCGTCTATATCTCGGCTCTAGGCCCTCATTTTTT

177 bases

35 bases

Average Coverage = 177/35 ~= 5-fold (5x)

Suffix Prefix Matching

TCTATATCTCGGCTCTAGG

TATCTCGACTCTAGGCC

Suffix Prefix Matching

TCTATATCTCGGCTCTAGG

TATCTCGACTCTAGGCC

Suffix Prefix Matching

TCTATATCTCGGCTCTAGG

GGCGTCTATATCTCGGCTCTAGGCCCTCATTTTTT

TATCTCGACTCTAGGCC

If a suffix of read A is similar to a prefix of read then A and B might overlap in the genome

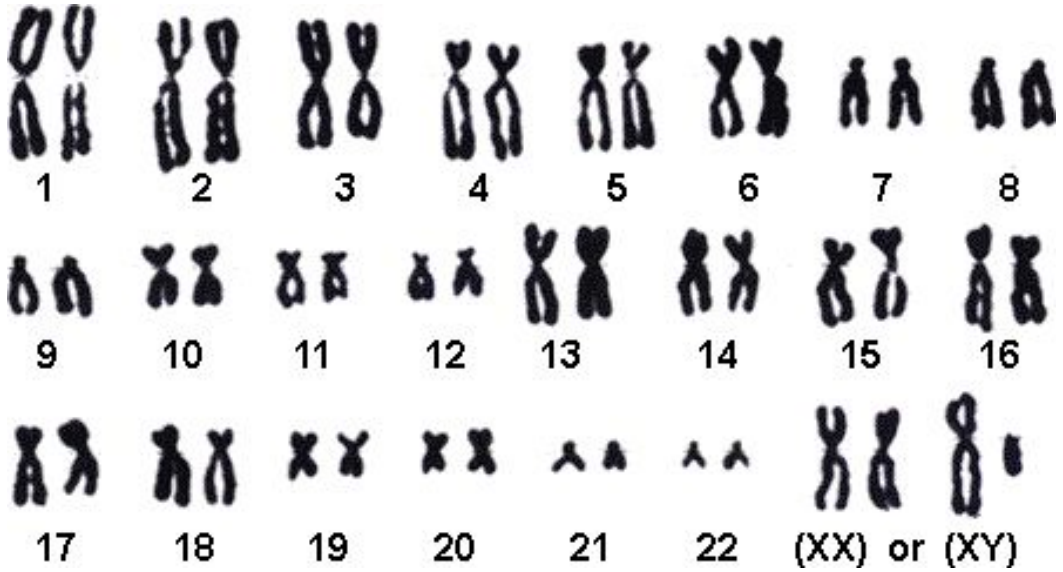
Suffix Prefix Differences

TCTATATCTCGGCTCTAGG

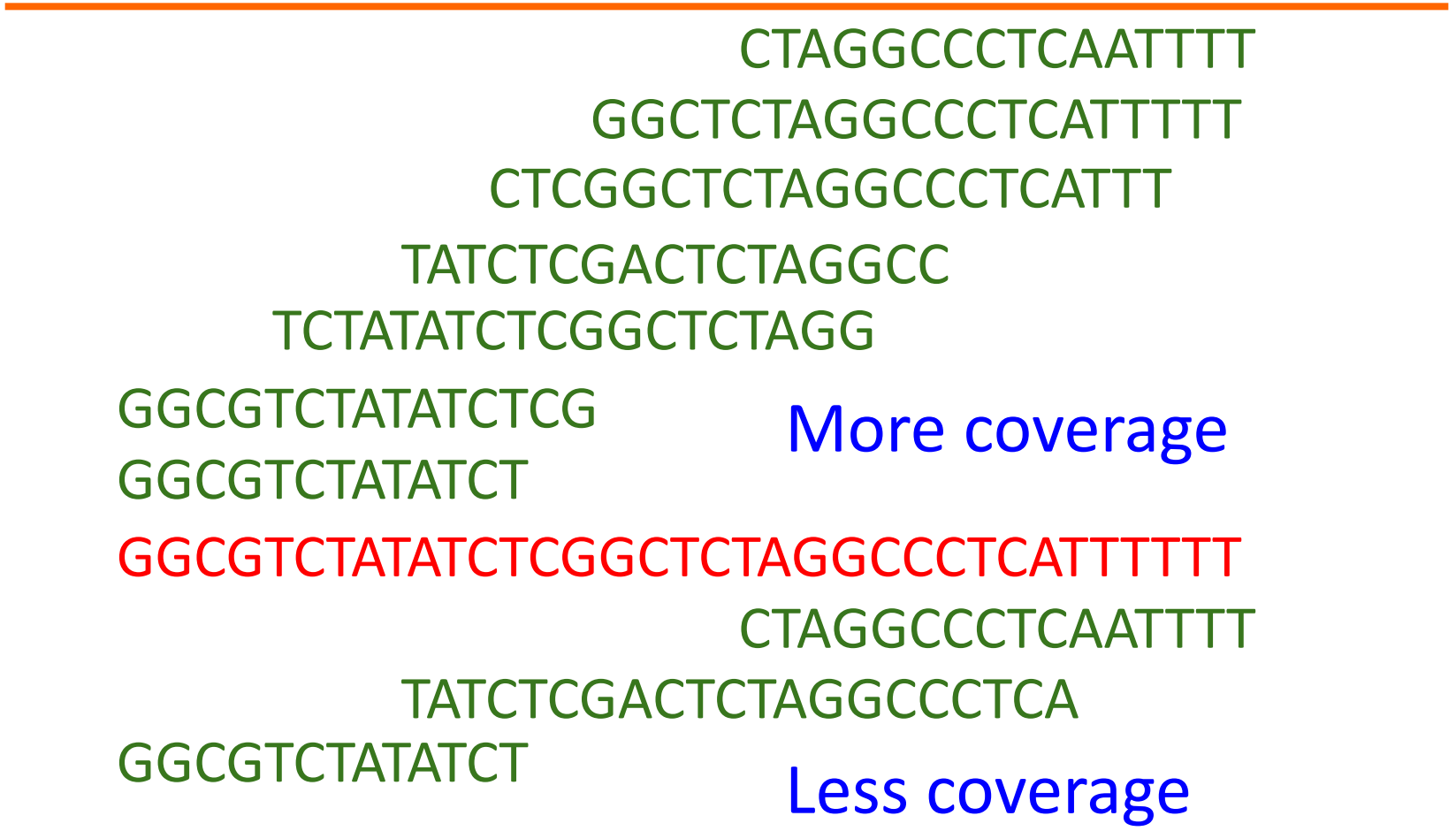
TATCTCGACTCTAGGCC

Why the differences?

1. Sequencing errors
2. Polyploidy



High and Low Coverage



More coverage leads to more and longer overlaps

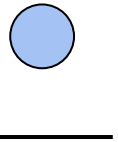
Overlap Consensus

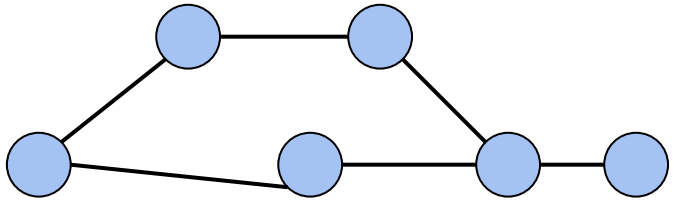
What is the best representation to the set of sequences?

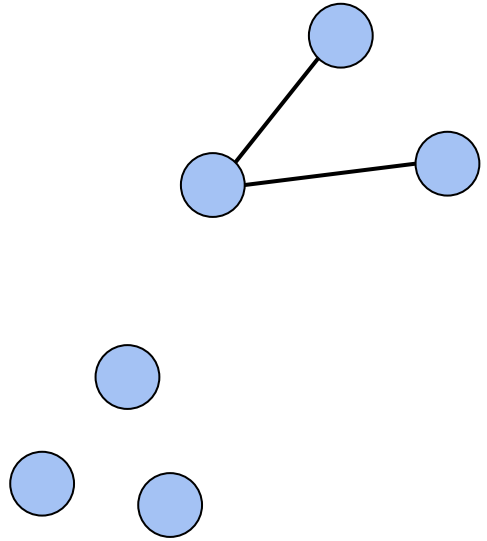
Graph

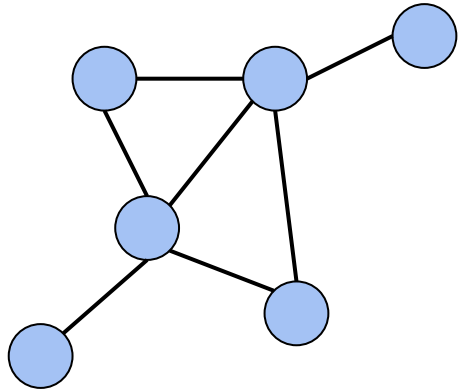
A graph is a set of:

- Nodes (vertices)
- Edges - connecting two nodes



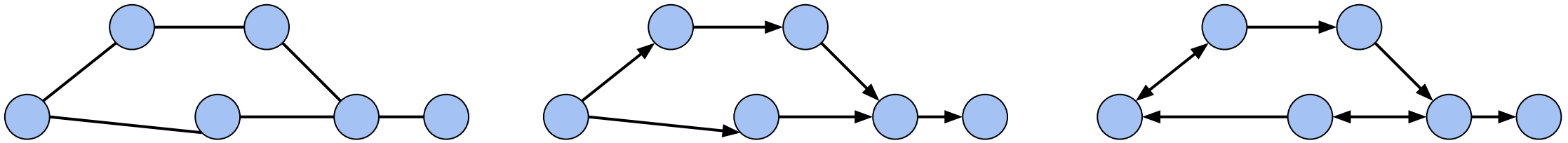




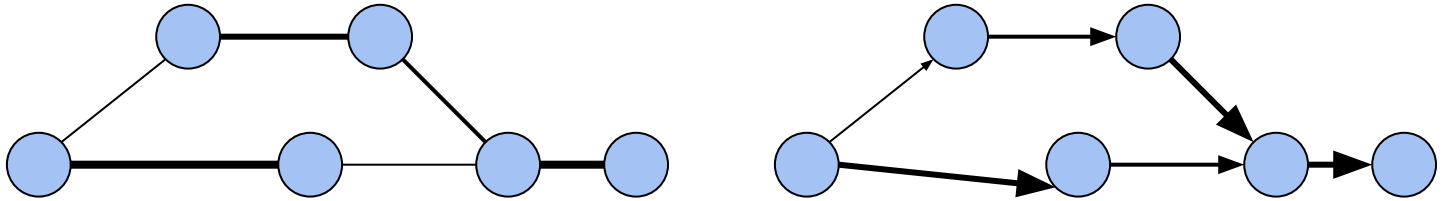


Directed and Weighted Graphs

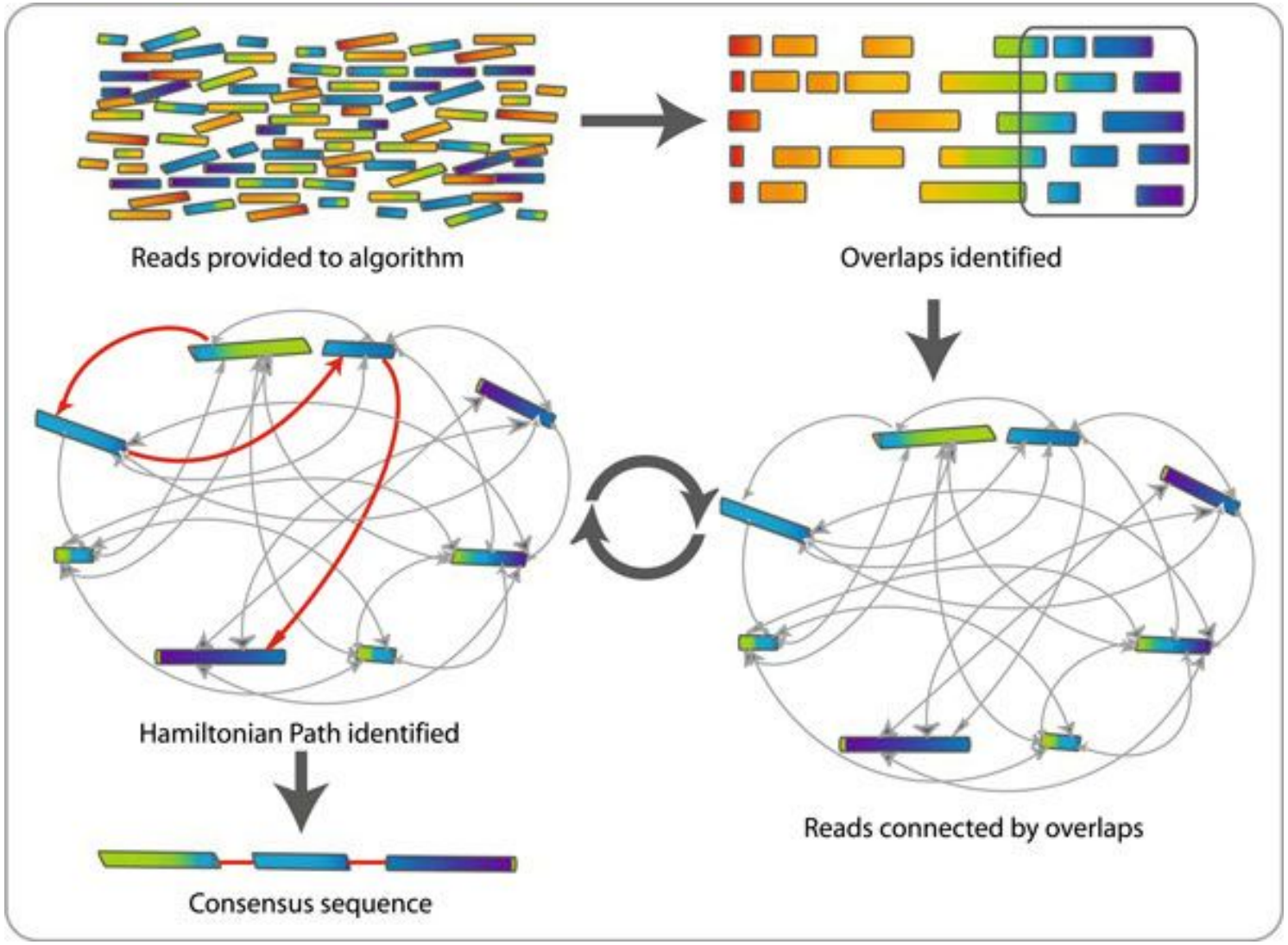
Graphs can be **directed** or **non-directed**



We can assign **weights** to edges



Overlap–layout–consensus genome assembly algorithm



Overlap Consensus Graph

Nodes: all 6-mers in GTACGTACGAT

Edges: overlaps of length > 3

GTACGT

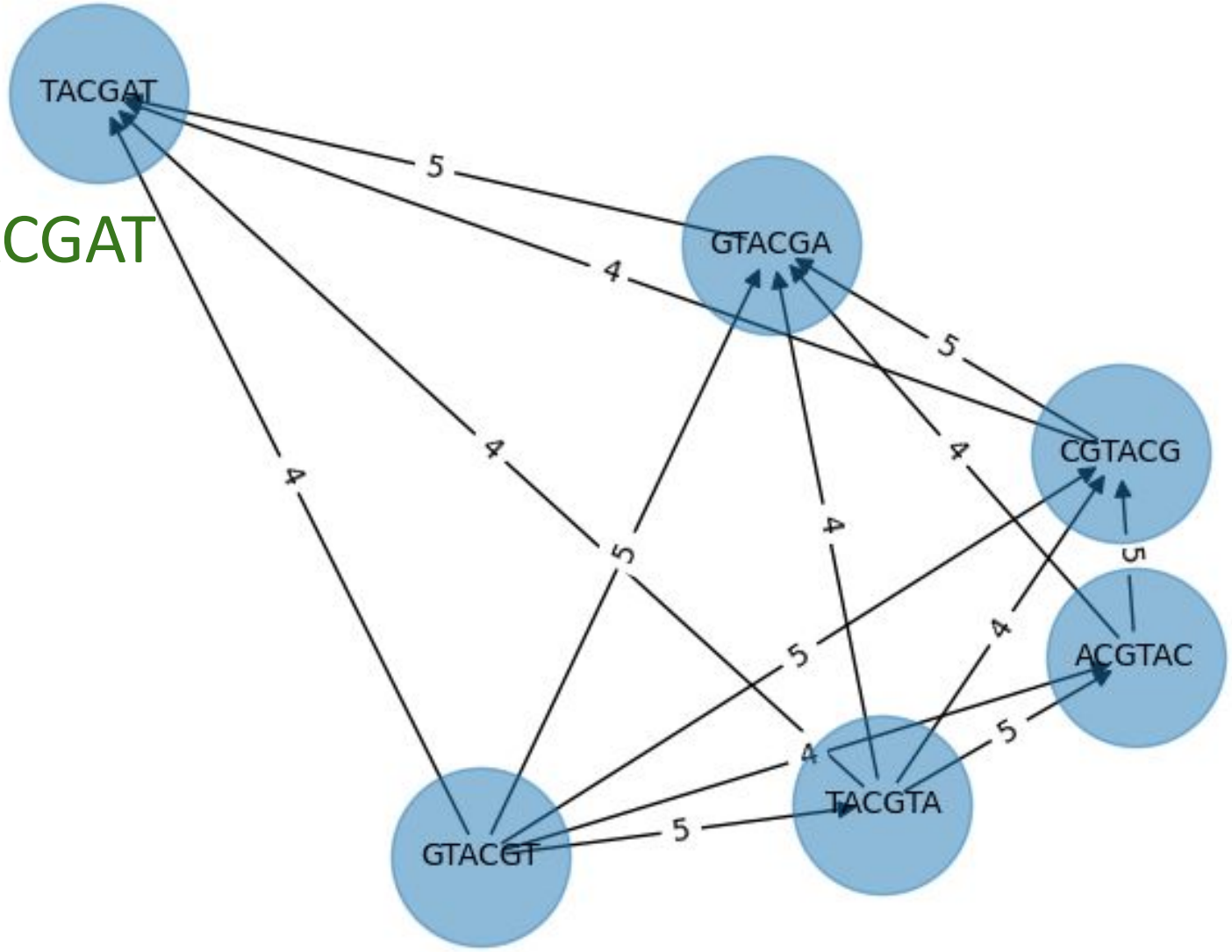
TACGTA

ACGTAC

CGTACG

GTACGA

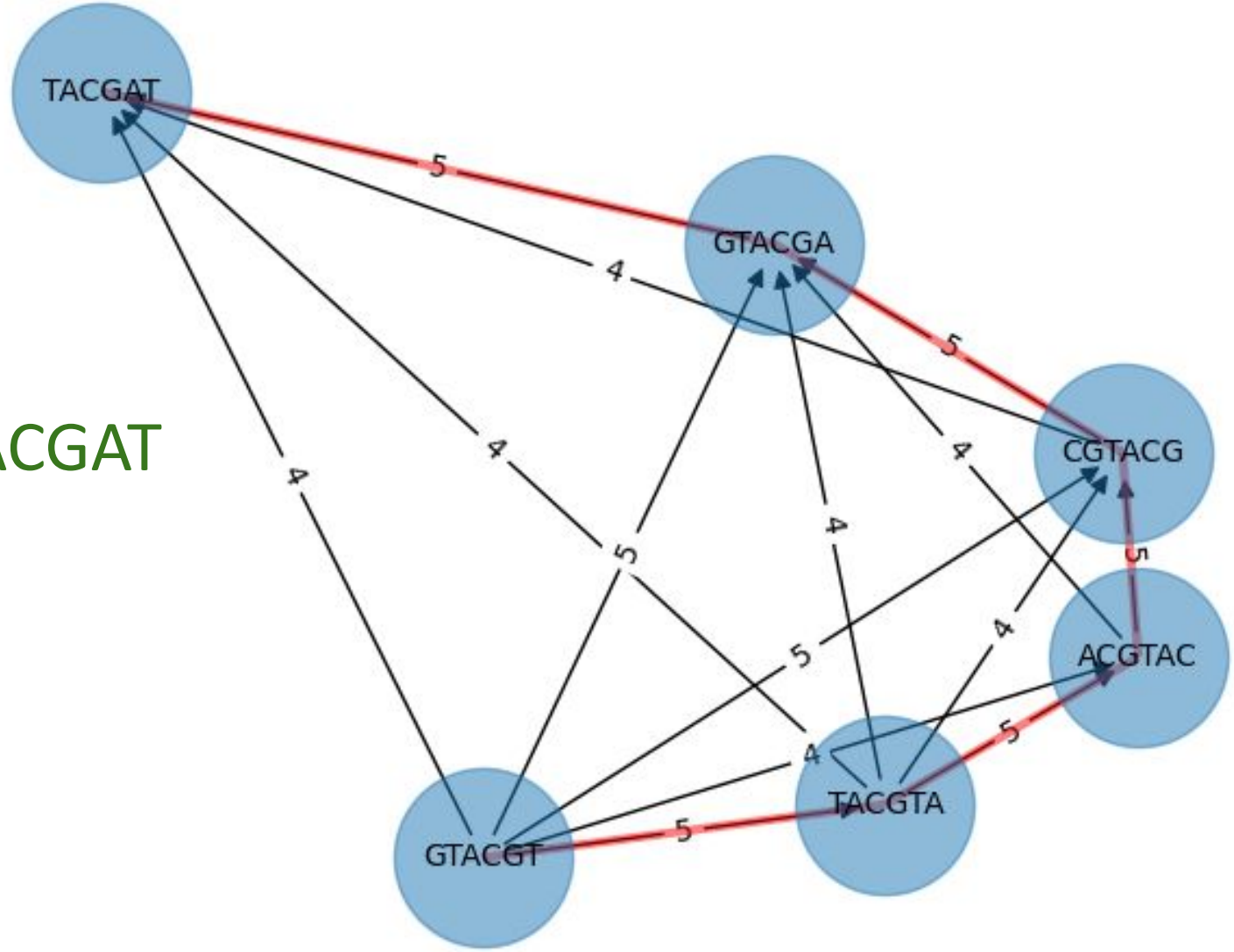
TACGAT



Overlap Consensus Graph

Nodes: all 6-mers in GTACGTACGAT

Edges: overlaps of length > 3



Shortest Common Superstring

The Shortest Common Superstring problem (SCS) aim to find the shortest possible string that contains every string in a given set as substrings

Example: BAA AAB BBA ABA ABB BBB AAA BAB

Concatenation: BAAAABBBAABAABBBBBBAAABAB

24

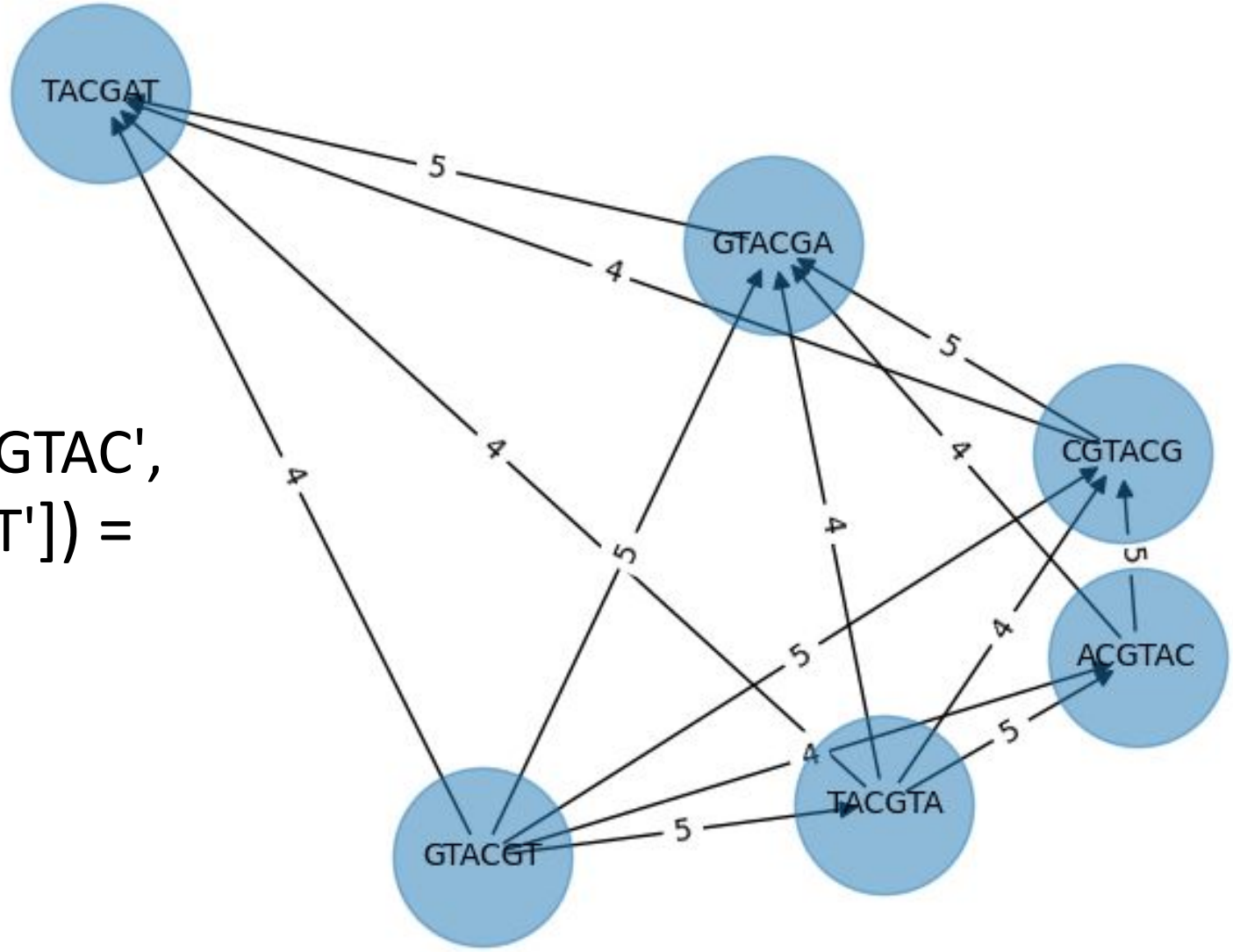
AAA
AAB
ABB
BBB
BBA
BAB
ABA
BAA

SCS: AAABBBABAA

10

Shortest Common Superstring

SCS(['GTACGT', 'TACGTA', 'ACGTAC', 'CGTACG', 'GTACGA', 'TACGAT']) = GTACGTACGAT



Shortest Common Superstring

Brute Force

Order 1: AAA AAB ABA ABB BAA BAB BBA BBB

AAABABBAABABBABBB Superstring 1

Order 2: AAA AAB ABA BAB ABB BBB BAA BBA

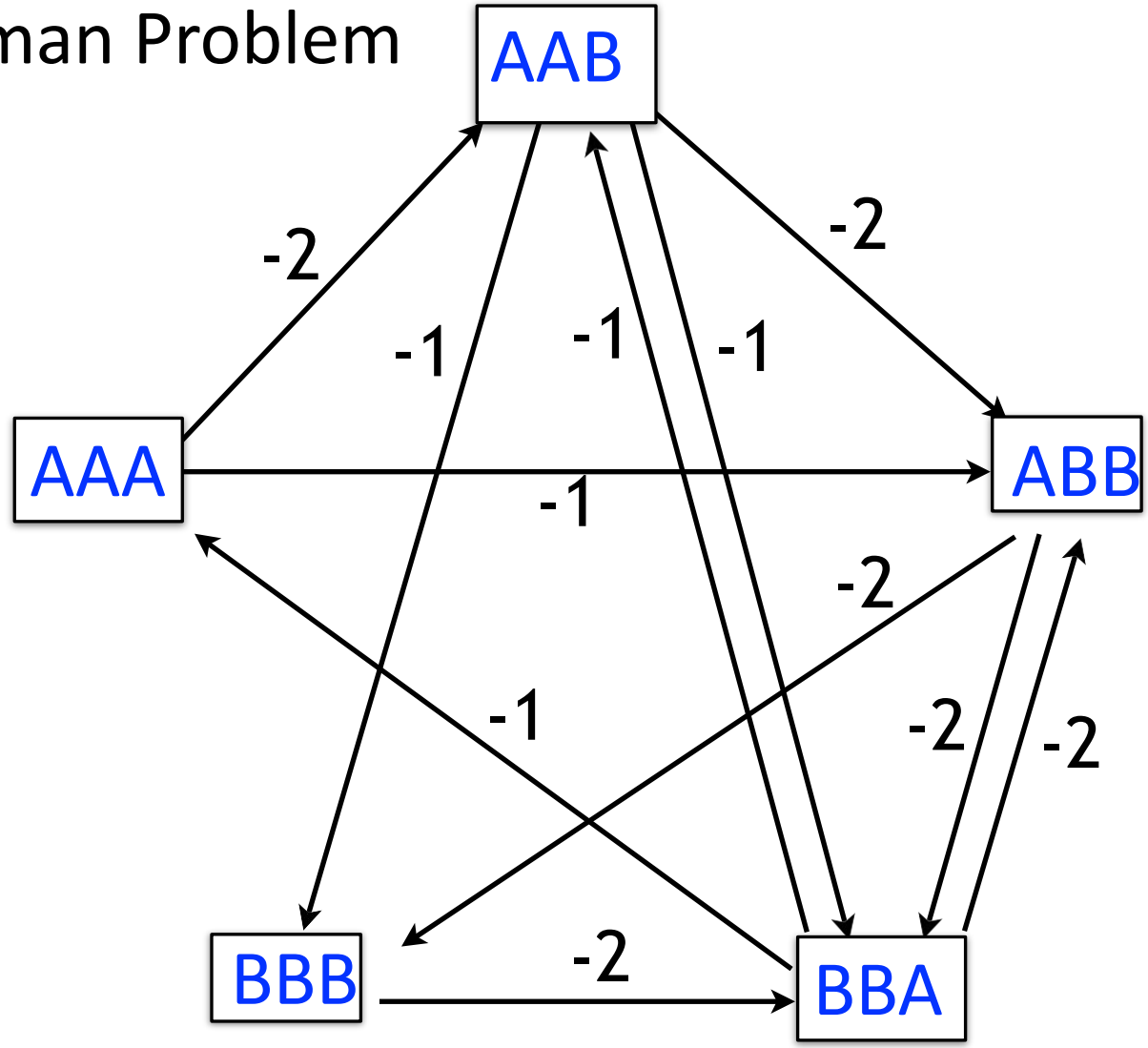
AAABABBBAABBA Superstring 2

O(n!)

Shortest Common Superstring

Traveling Salesman Problem

Modified overlap graph where each edge has cost = - (length of overlap)
SCS corresponds to a path that visits every node once, minimizing total cost along path



Shortest Common Superstring

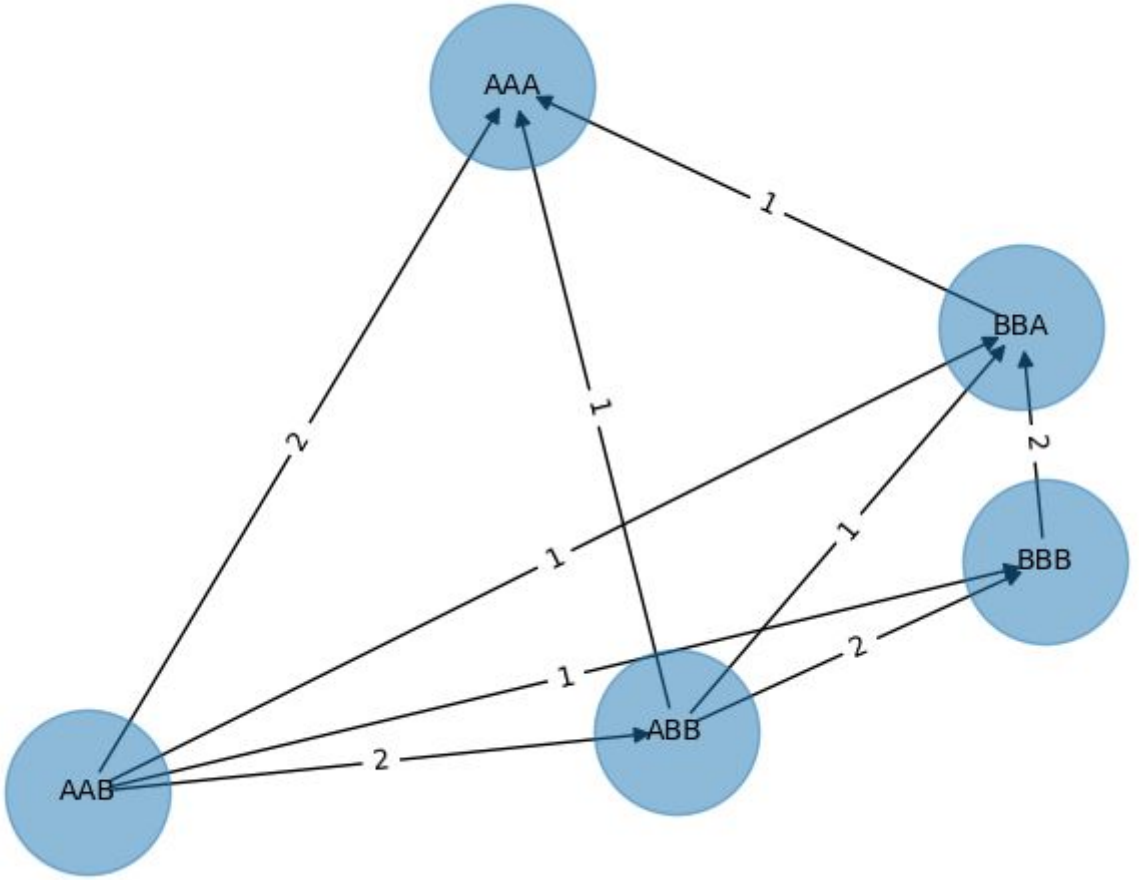
NP-complete

No efficient solution algorithm has been found

Shortest Common Superstring

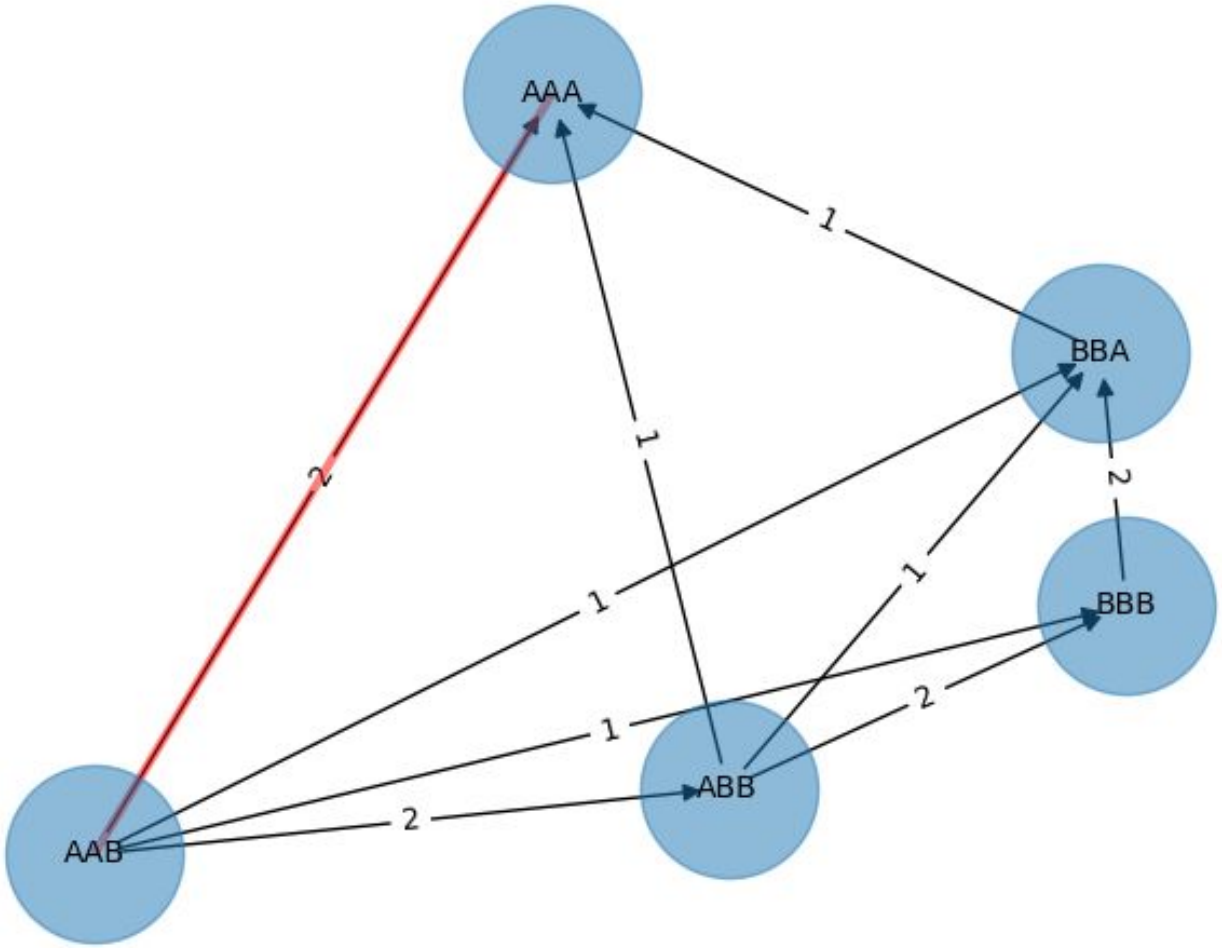
Greedy

Example: BAA AAB BBA ABA ABB BBB AAA BAB



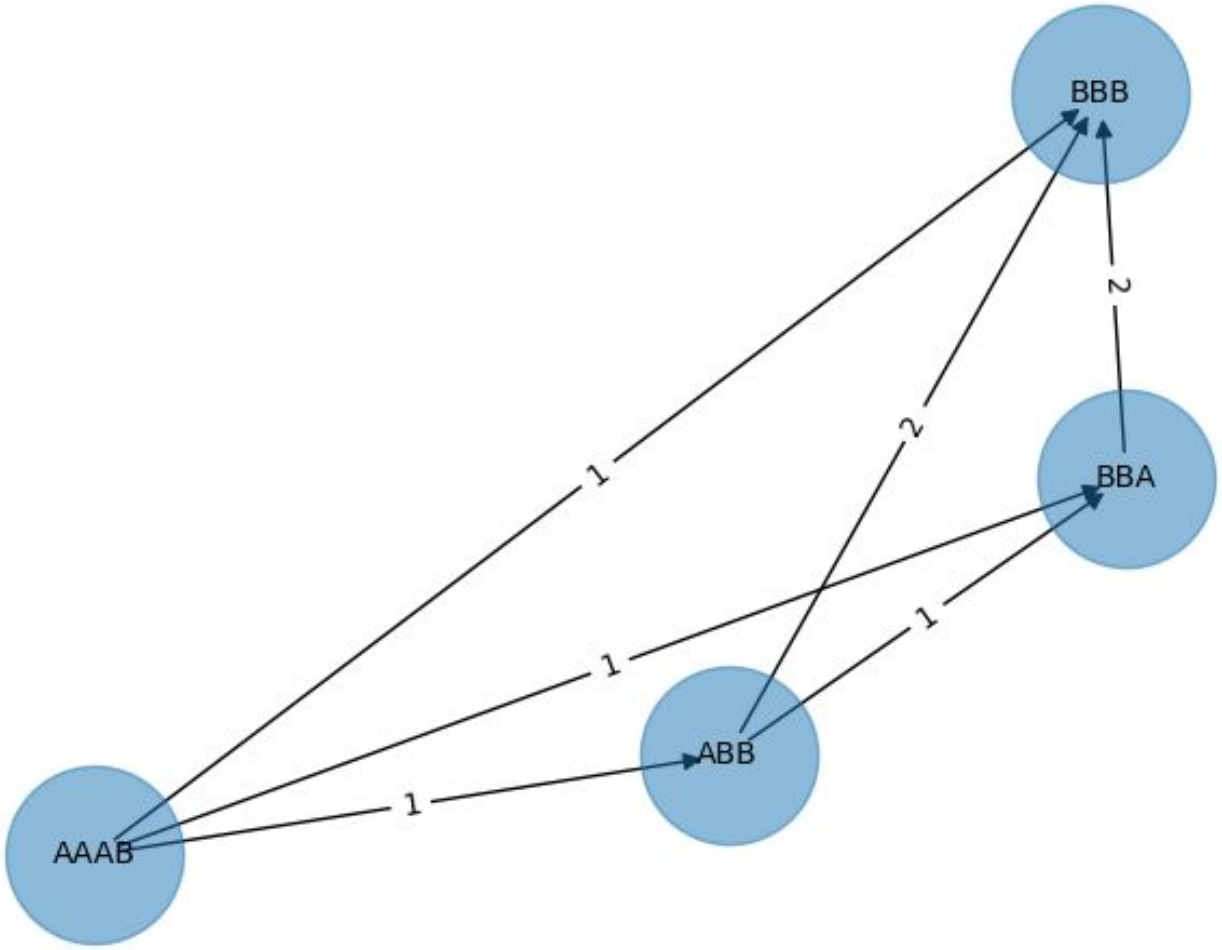
Shortest Common Superstring

Greedy



Shortest Common Superstring

Greedy



Shortest Common Superstring

Greedy

Superstring, length 7

Alternative Shuffling

Superstring, length 9

Greedy answer isn’t necessarily optimal

Shortest Common Superstring

Greedy

Greedy algorithm is not guaranteed to choose overlaps yielding SCS

But greedy algorithm is a good approximation; i.e. the superstring yielded by the greedy algorithm won't be more than ~2.5 times longer than true SCS

Gusfield, Dan. "Algorithms on Strings, Trees, and Sequences - Computer Science and Computational Biology." (1997).

Shortest Common Superstring

Greedy

a_*long*_*long*_*long*_time *l=6*

ng_lon _long_ a_long long_l ong_ti ong_lo long_t g_long g_time ng_tim

5 ng_time ng_lon _long_ a_long long_l ong_ti ong_lo long_t g_long

5 ng_time g_long_ ng_lon a_long long_l ong_ti ong_lo long_t

5 ng_time long_ti g_long_ ng_lon a_long long_l ong_lo

5 ng_time ong_lon long_ti g_long_ a_long long_l

5 ong_lon long_time g_long_ a_long long_l

5 long_lon long_time g_long_ a_long

5 long_lon g_long_time a_long

5 long_long_time a_long

4 a_long_long_time

a_*long*_*long*_time

Missing a _long

Shortest Common Superstring

Repeats often foil assembly. They certainly foil SCS, with its “shortest” criterion!

Reads might be too short to “resolve” repetitive sequences. This is why sequencing vendors try to increase read length.

Algorithms that don’t pay attention to repeats (like our greedy SCS algorithm) might *collapse* them

a_long_long_long_time

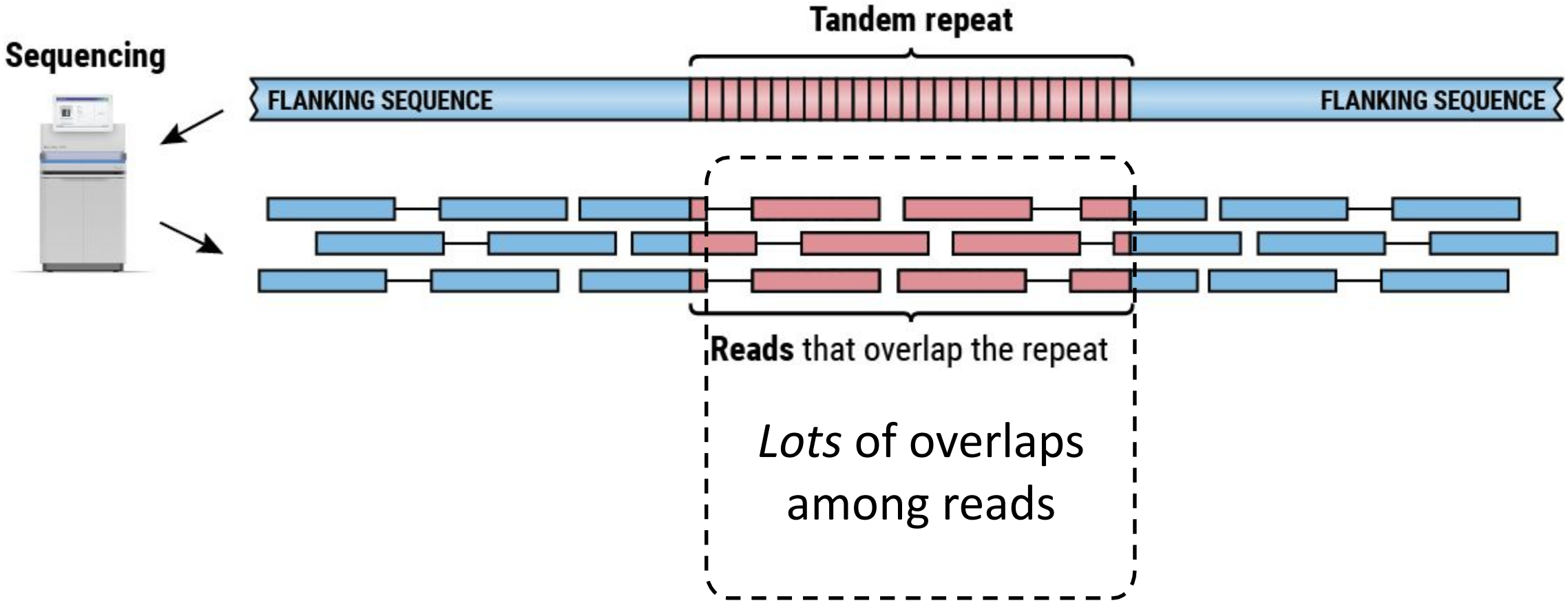
collapse

a_long_long_time

The human genome is ~ 50% repetitive!

Shortest Common Superstring

Repeats



Genome Assembly by Reads Overlap - Challenges

Do we believe short overlaps?

How do we handle sequencing errors?

Computationally ineffective for large data sets

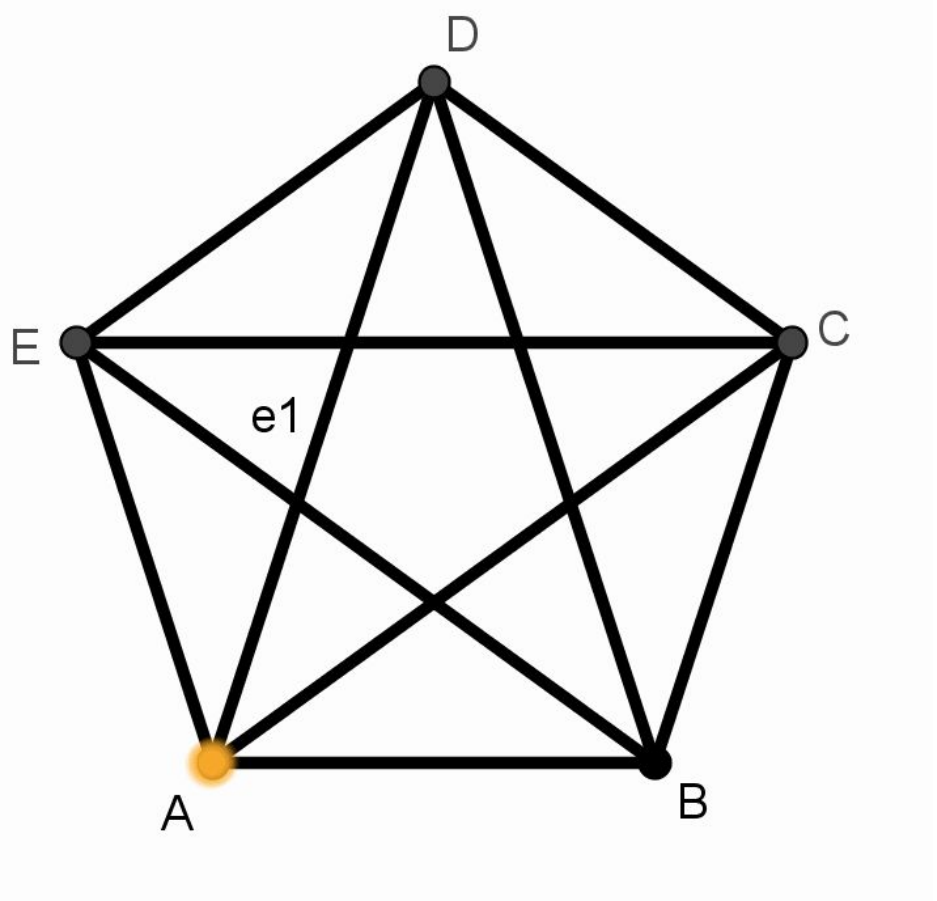
Bottom line: not good enough for large and complex genomes (e.g. human)

We need something smarter!

The Eulerian path

A path in a graph that visits every **edge** exactly once

- Must visit **all** edges
- Can’t visit an edge twice
- Can visit a node more than once



K-mers

AGATCCAGCGAGGTCGCTATCCGTTAATTG

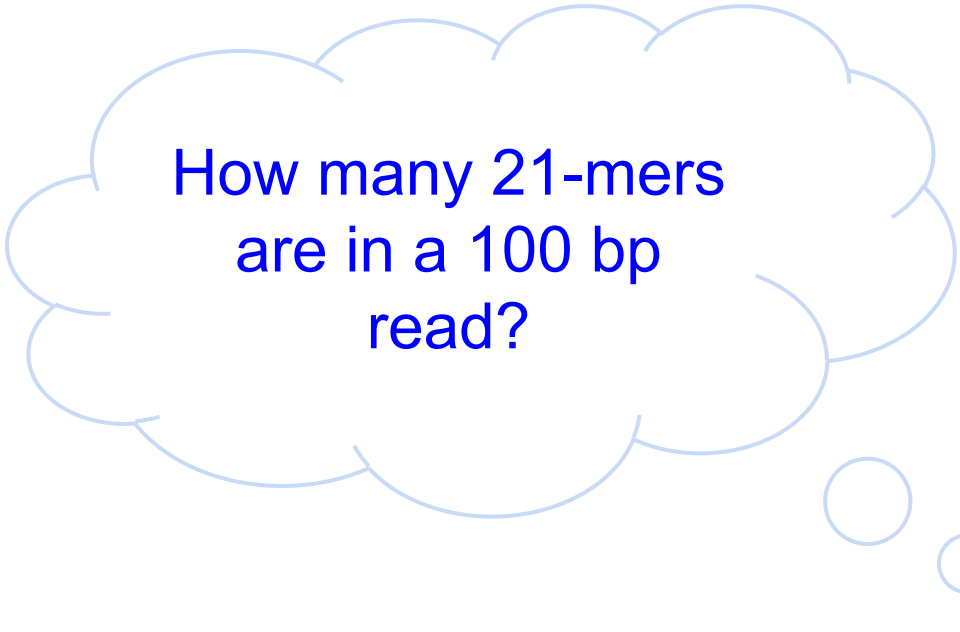
5-mers

AGATC
GATCC
ATCCA
…
AATTG

K-mers

AGATCCAGCGAGGTCGCTATCCGTTAATTG

5-mers
AGATC
GATCC
ATCCA
…
AATTG



7-mers
AGATCCA
GATCCAG
ATCCAGC
...
TTAATTG

De Bruijn Graph

Genome (G=30): AGATCCAGCGAGGTCGCTATCCGTTAATTG

Reads (L=10): AGATCCAGCG

AGCGAGGTCG

GCTATCCGTT

CCGTTAATTG

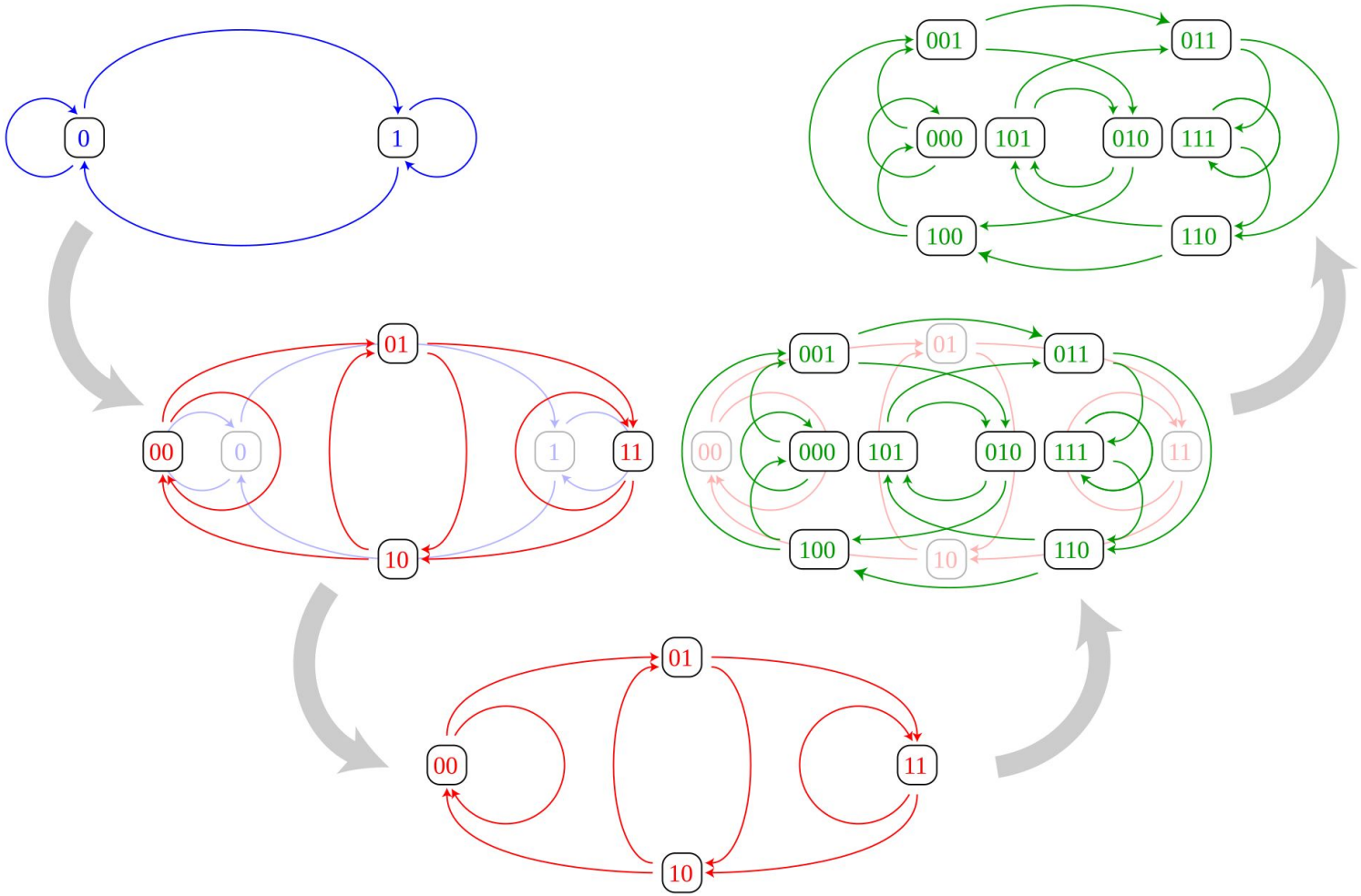
Break into k-mers (k=4):

AGATCCAGCG → AGAT GATC ATCC TCCA CCAG CAGC AGCG

AGCGAGGTCG → AGCG GCGA CGAG GAGG AGGT GGTC GTCG

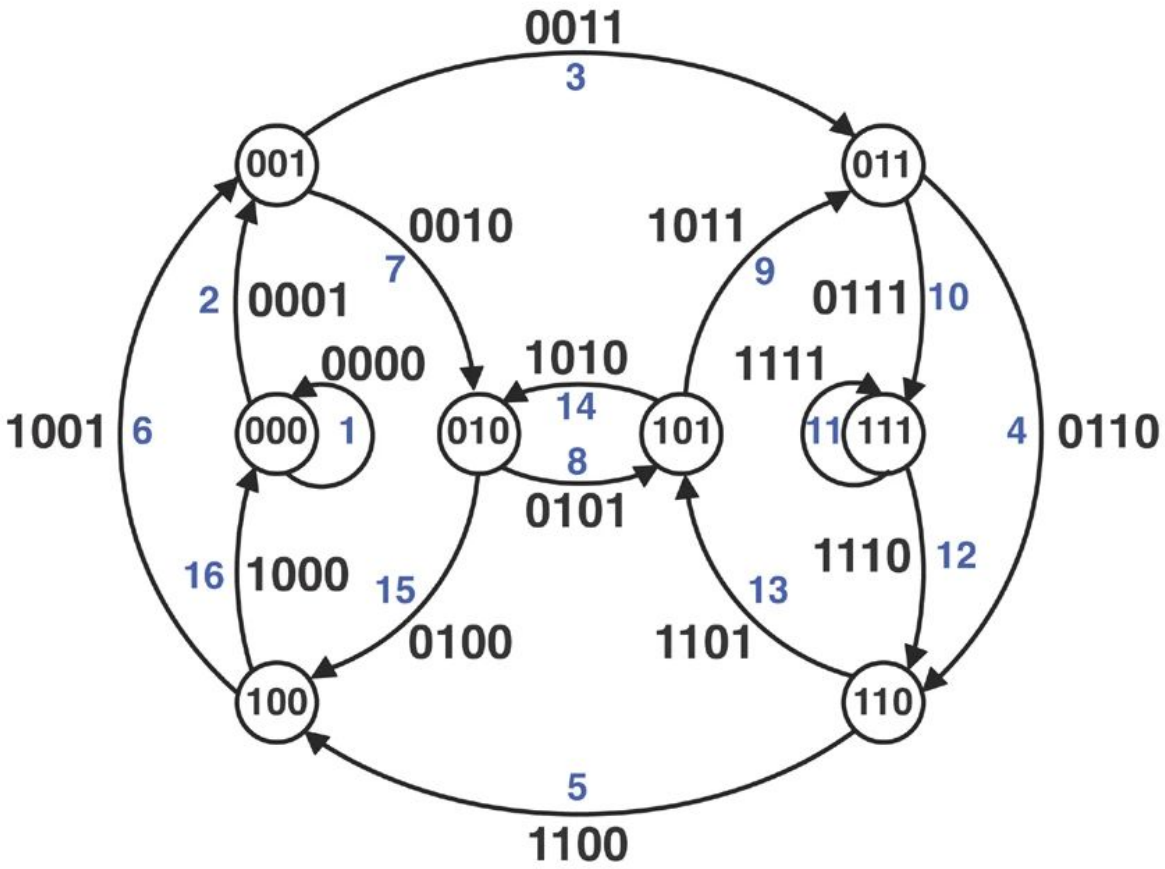
…

De Bruijn Graph



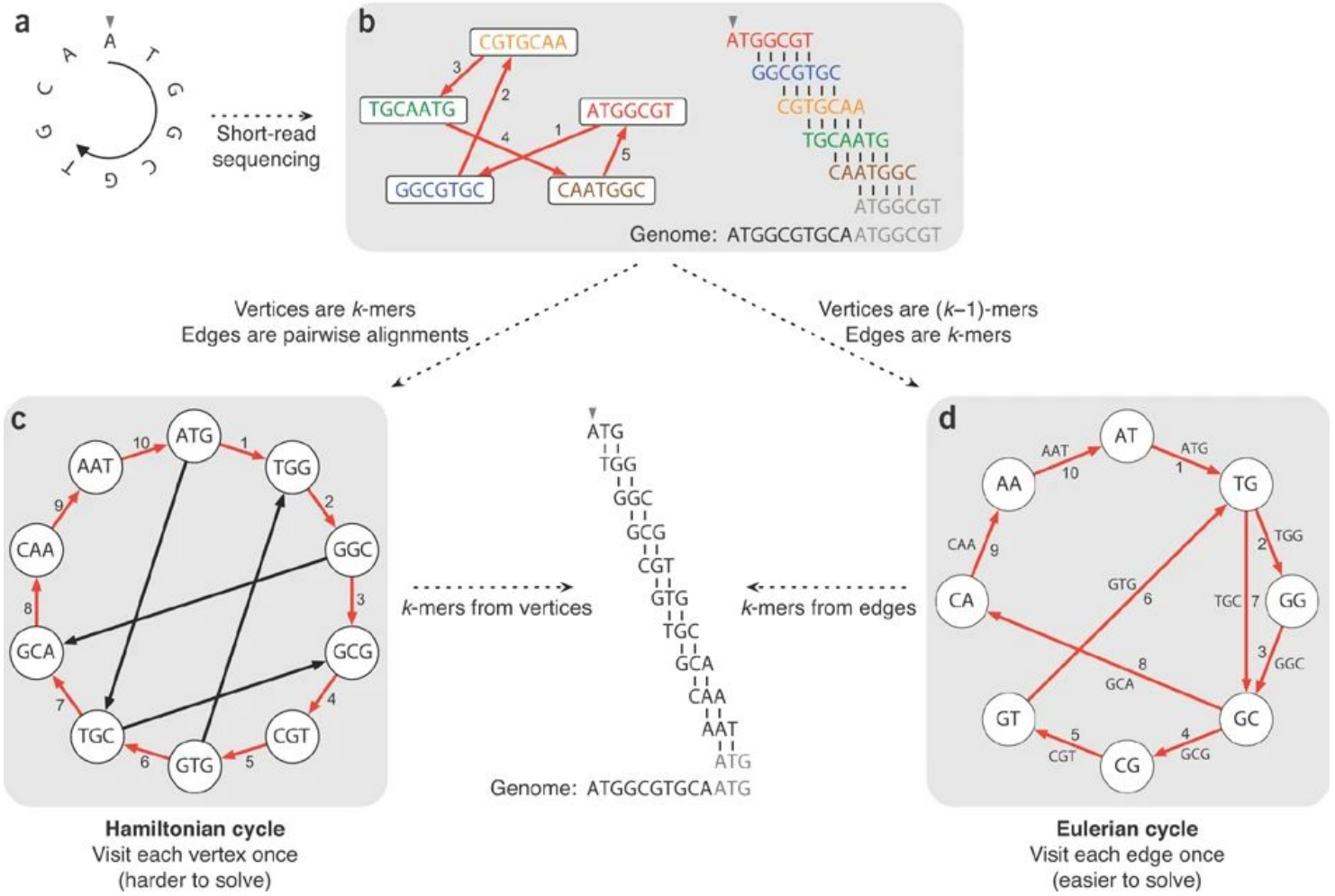
https://en.wikipedia.org/wiki/De_Bruijn_graph#

De Bruijn Graph



https://www.nature.com/articles/nbt.2023

De Bruijn Graph



https://www.nature.com/articles/nbt.2023

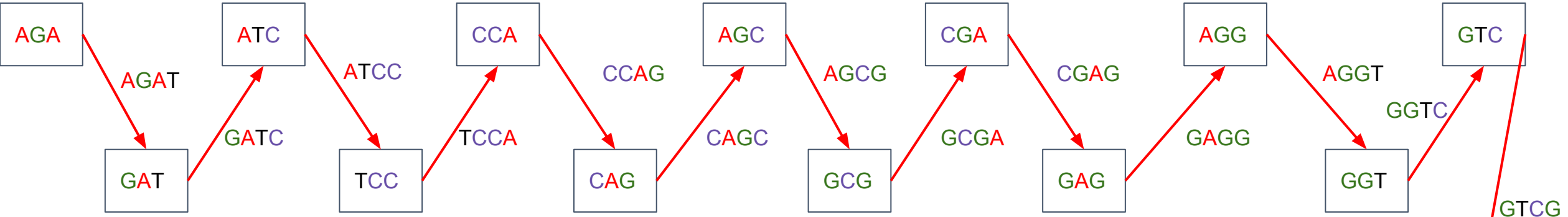
Create graph nodes - each **unique prefix and suffix** of length k-1 of k-mers

AGA ATC CCA AGC CGA AGG GTC

GAT TCC CAG GCG GAG GGT

Add directed edge between node x and node y if k-mer exists with prefix x and suffix y

TCG



Find an Eulerian path (visit every **edge** exactly once) - this is the assembly!

De Bruijn Graph

A procedure for making a De Bruijn graph for a genome

Assume perfect sequencing where each length-k substring is sequenced exactly once with no errors

Pick a substring length k: 5

Start with each read:

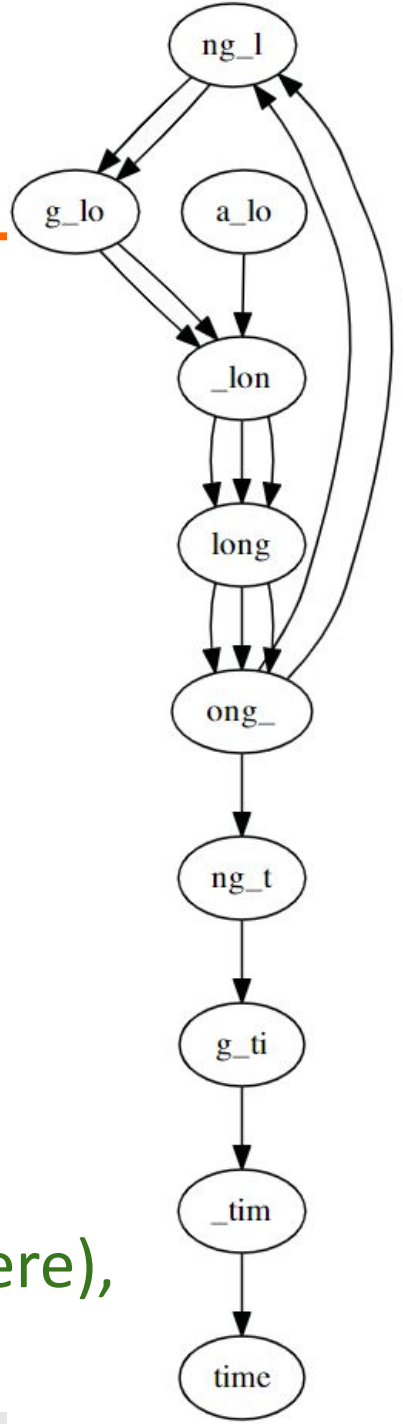
a_long_long_long_time

long_

long ong_

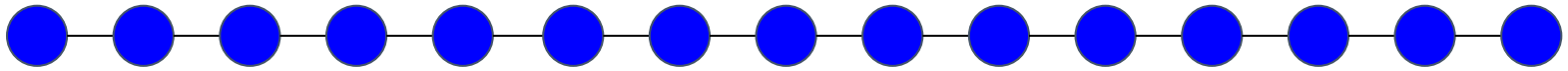
Take each k mer and split into left and right k-1 mers

Add k-1 mers as nodes to De Bruijn graph (if not already there), add edge from left k-1 mer to right k-1 mer

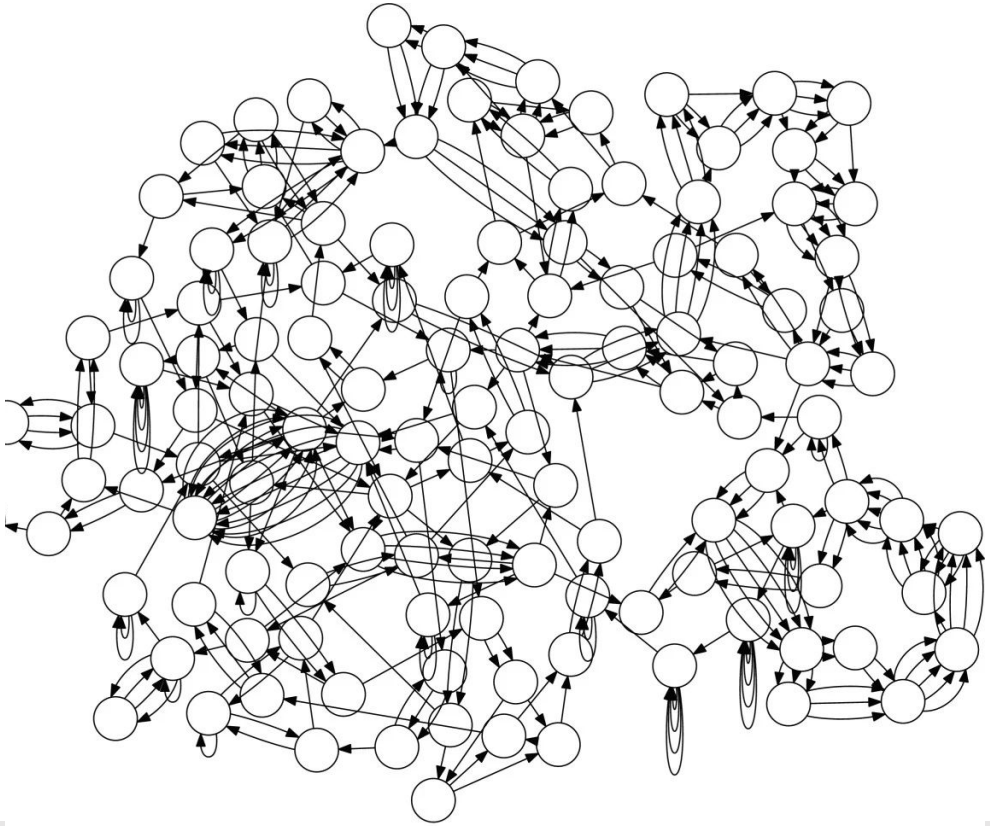


De Bruijn Graph

Ideally, we want our graph to look like this:



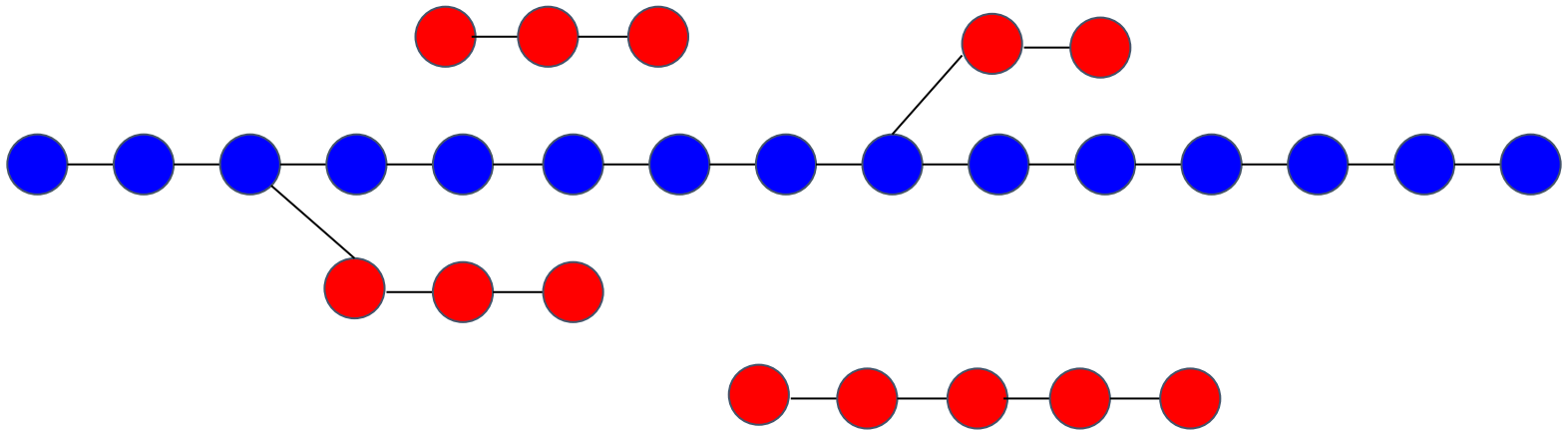
But in practice:



De Bruijn Graph

Sequencing errors

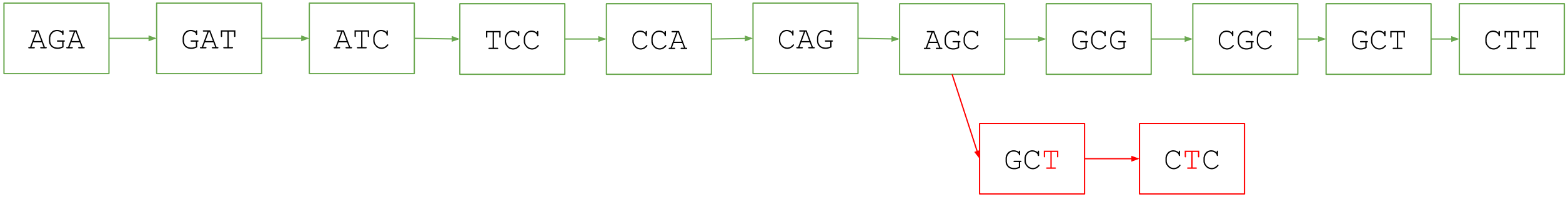
- Side branches
- Disconnected bits



De Bruijn Graph

Sequencing errors

```
AGATCCAGCG      → AGAT GATC ATCC TCCA CCAG CAGC AGCG
 GATCCAGCTC     → GATC ATCC TCCA CCAG CAGC AGCT GCTC
   TCCAGCGCTT   → TCCA CCAG CAGC AGCG GCGC CGCT GCTT
```



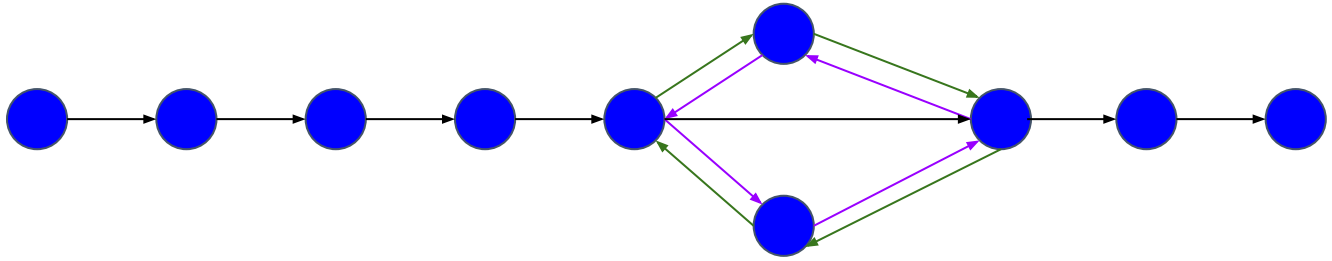
De Bruijn Graph

Genomic Repeats

- Tandem duplication

- Interspersed duplications

- Might create ambiguity - multiple possible eulerian paths



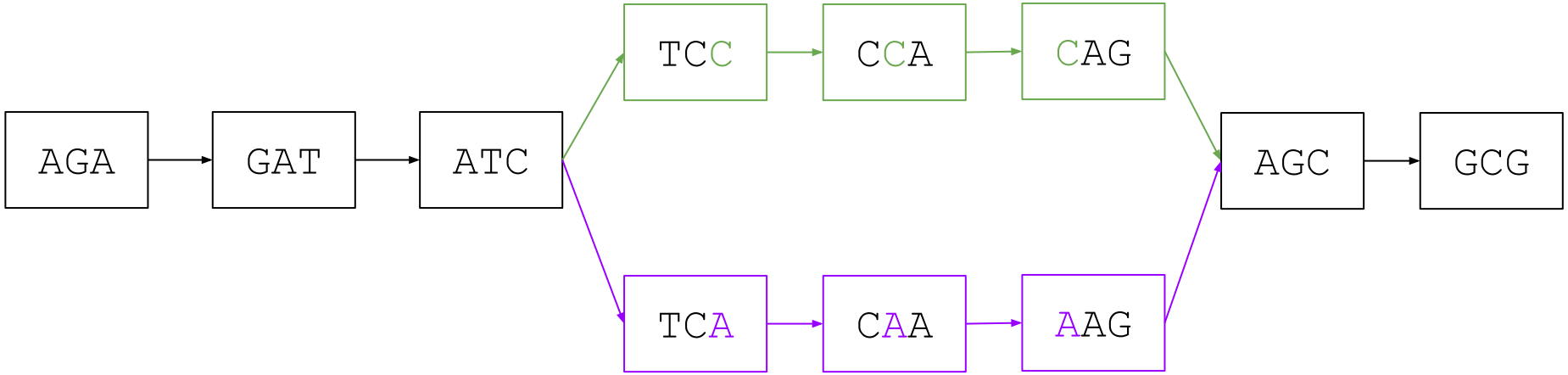
De Bruijn Graph

Heterozygosity

- Creates “bubbles” in the graph

Maternal `AGATCCAGCG → AGAT GATC ATCC TCCA CCAG CAGC AGCG`

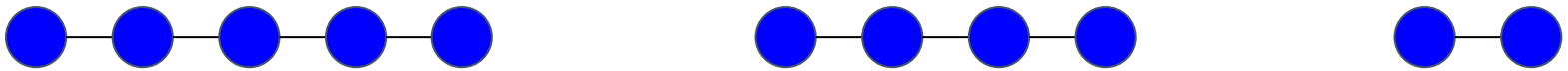
Paternal `AGATCAAGCG → AGAT GATC ATCA TCAA CAAG AAGC AGCG`



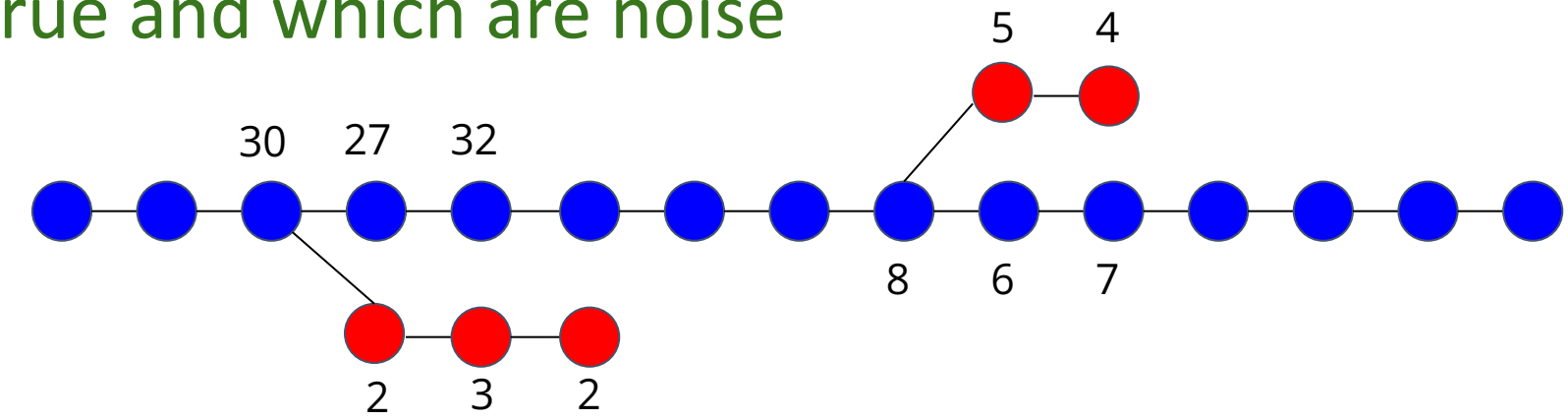
De Bruijn Graph

Uneven sequencing depth

Very low depth (e.g. low complexity regions) can fragment the graph:



Uneven depth can make it hard to determine which branches are true and which are noise



Error Correction

Extract all k-mers from all reads

Count how many times each k-mer was observed

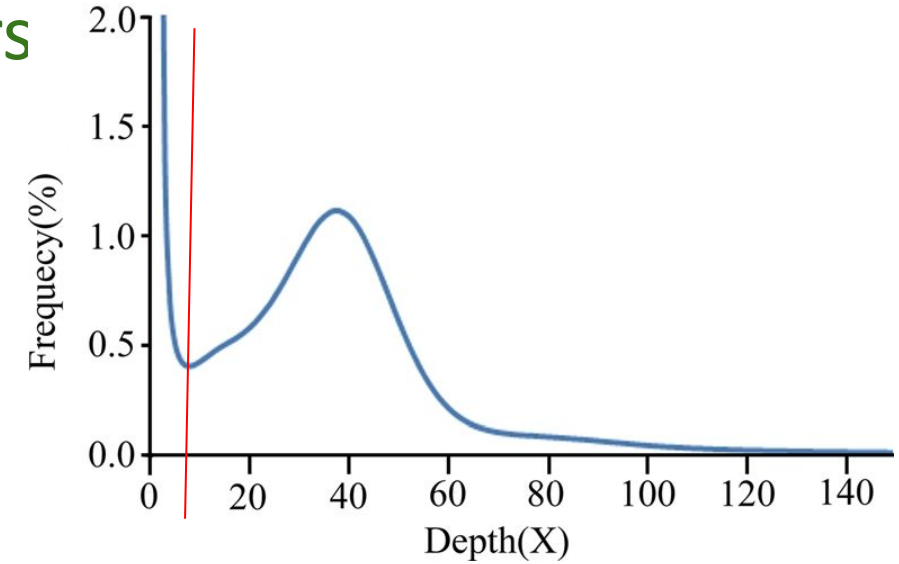
Label rare k-mers as error k-mers

Find reads from which error k-mers came

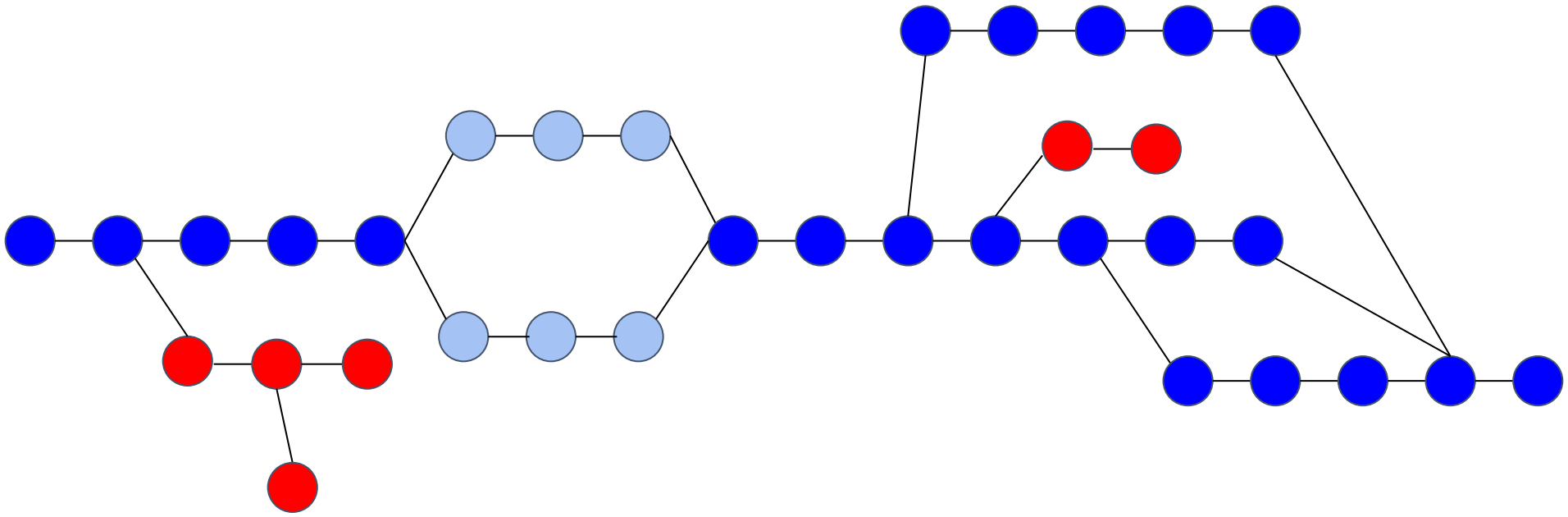
Discard or correct the reads with error k-mers

Count(GGATAGGCACCAGTTAT) = 30

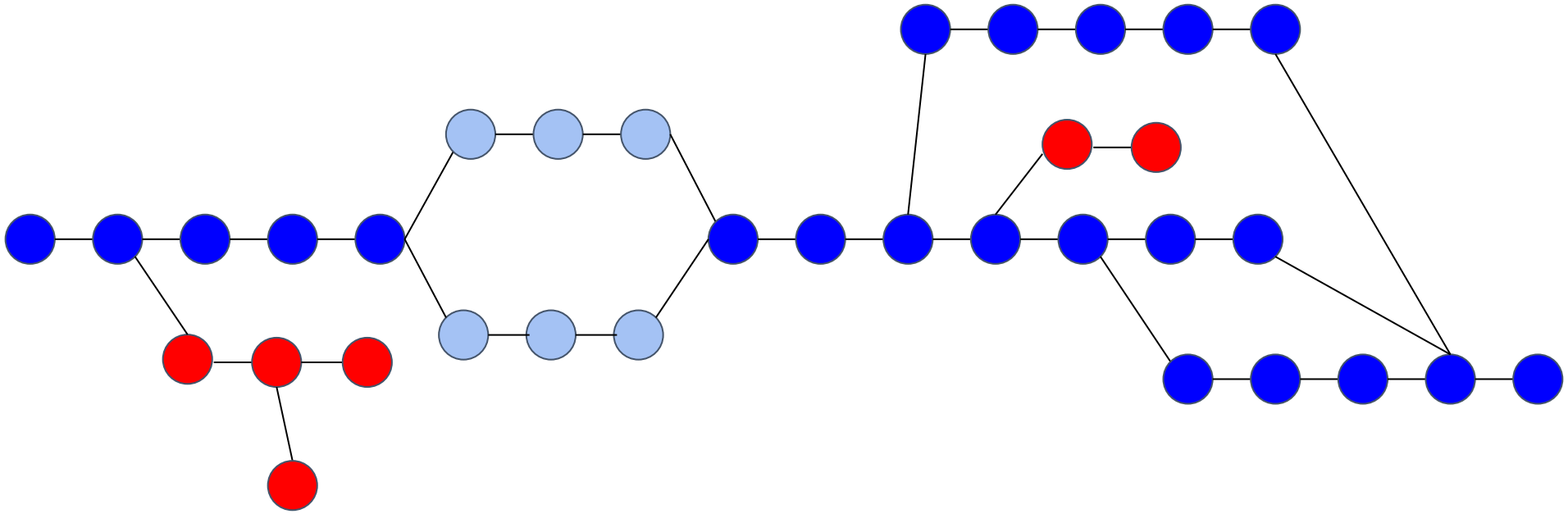
Count(GGATAGGTACCAGTTAT) = 1



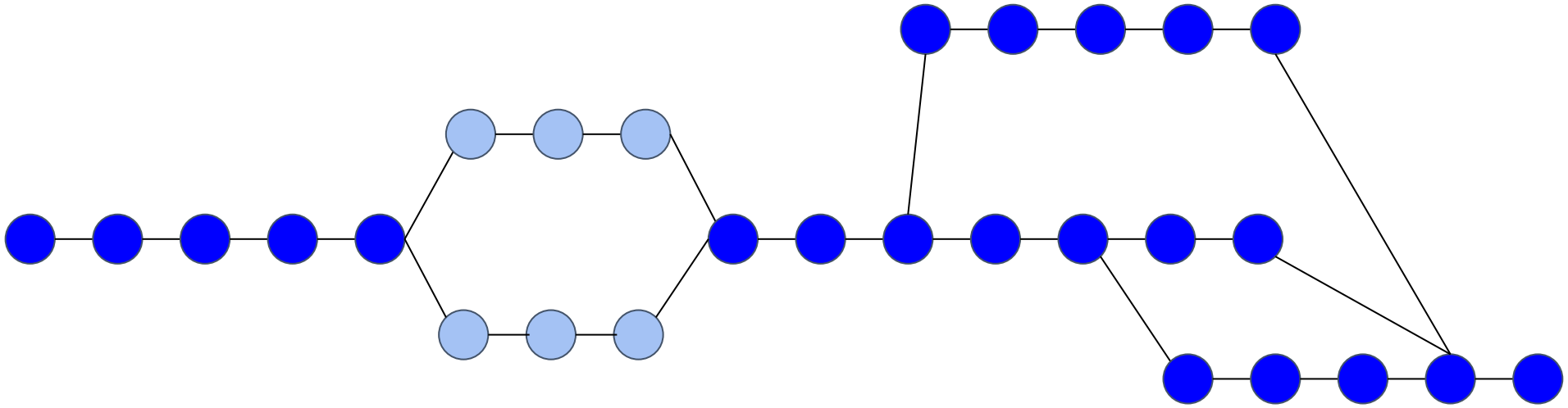
Build De Bruijn Graph



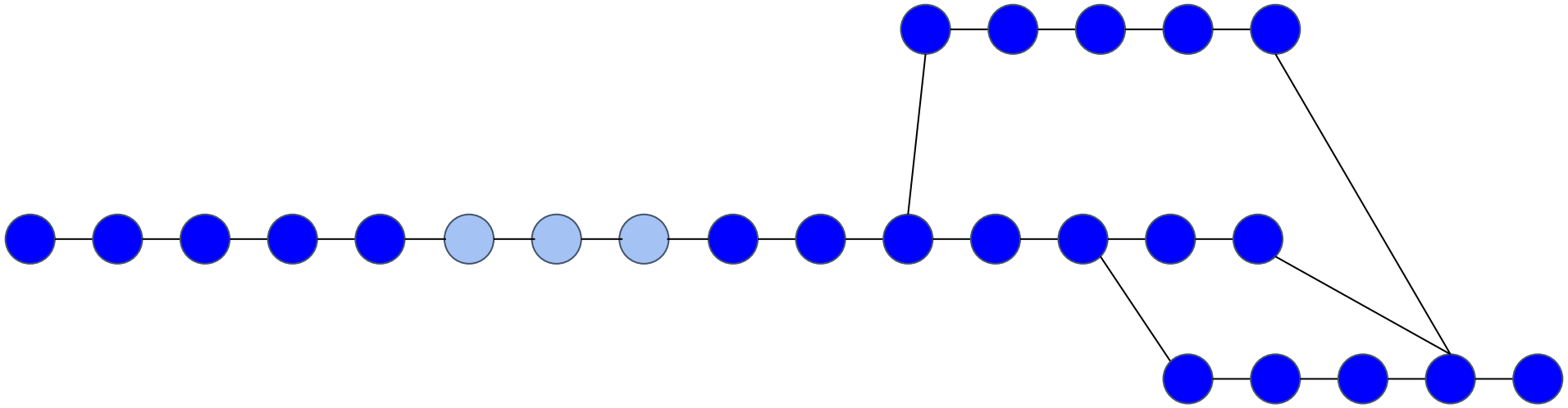
Prune Error Branches



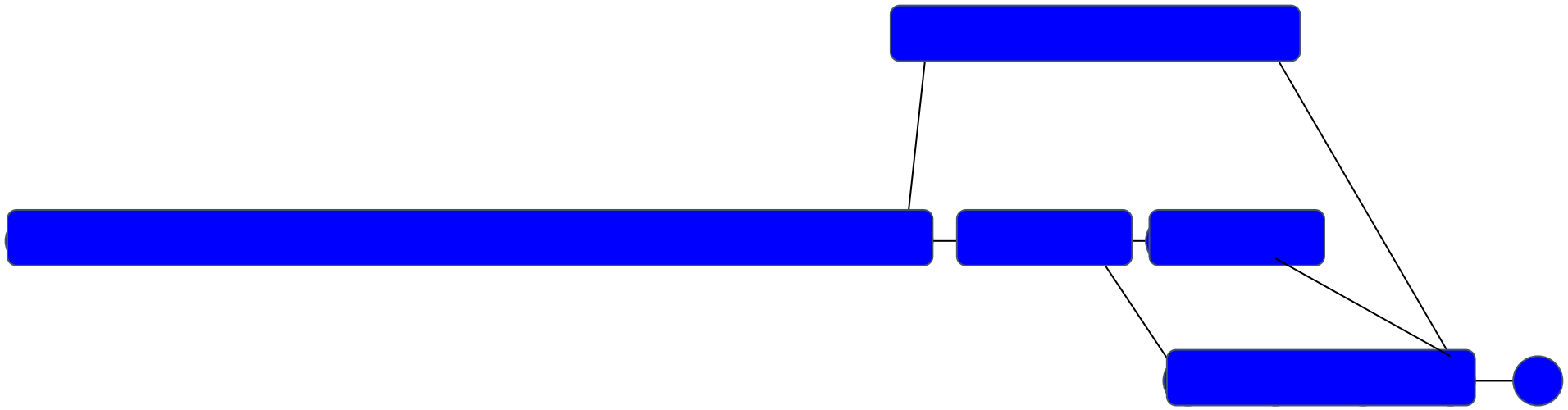
Resolve Bubbles



Resolve Bubbles



Create Contigs

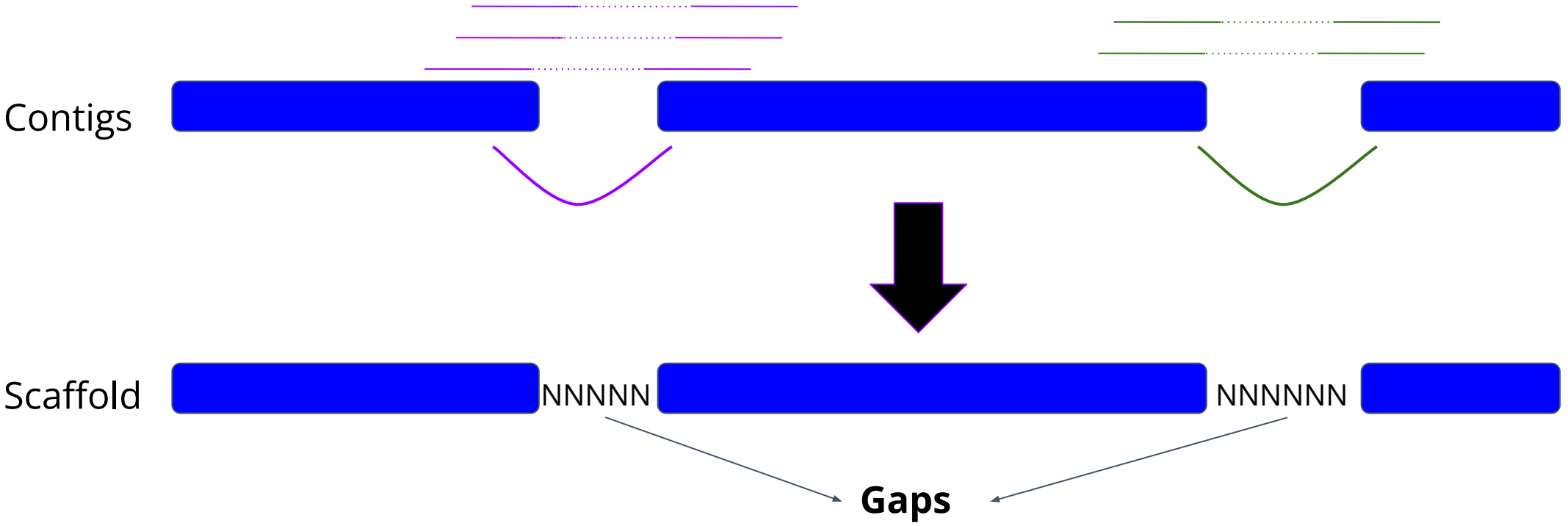


Scaffolding

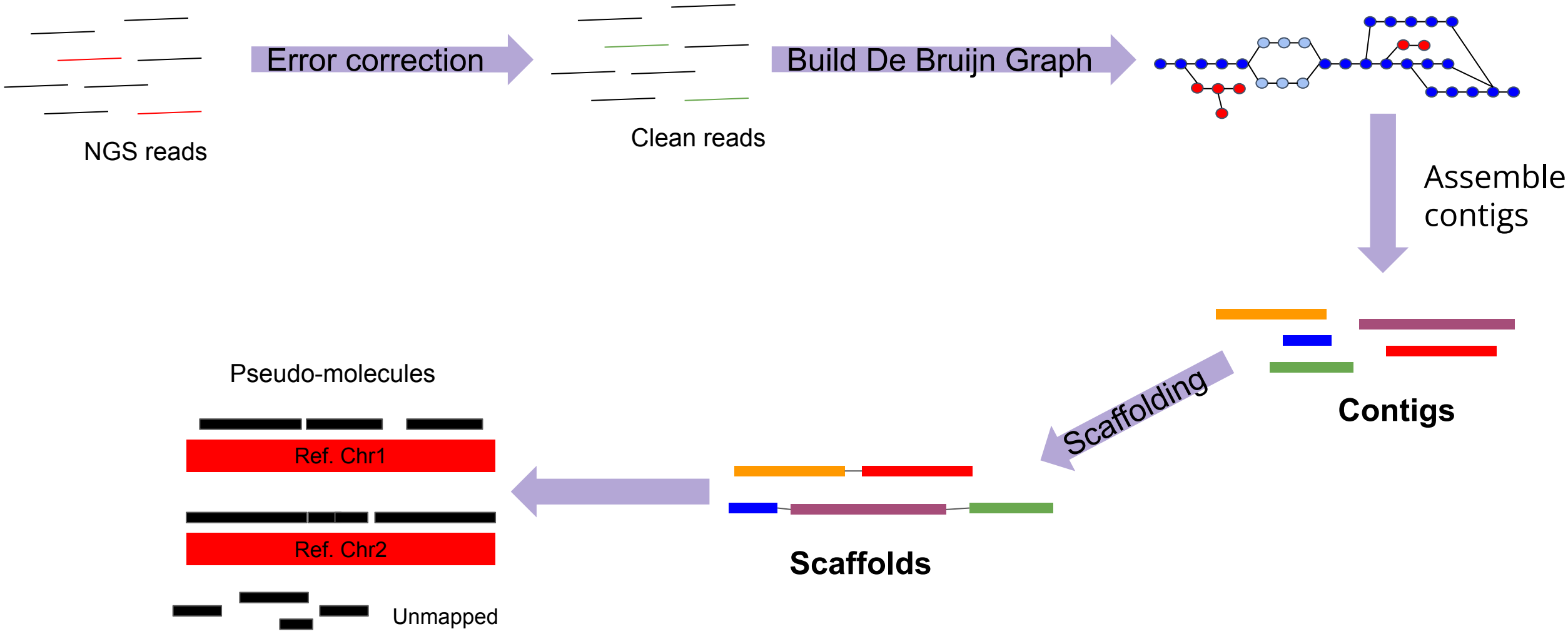
Mate Pair Sequencing

Use paired end information

Look for evidence of two contigs linked together



Workflow


NGS reads
Error correction
Clean reads
Build De Bruijn Graph
Assemble contigs
Contigs
Scaffolding
Scaffolds
Pseudo-molecules
Ref. Chr1
Ref. Chr2
Unmapped

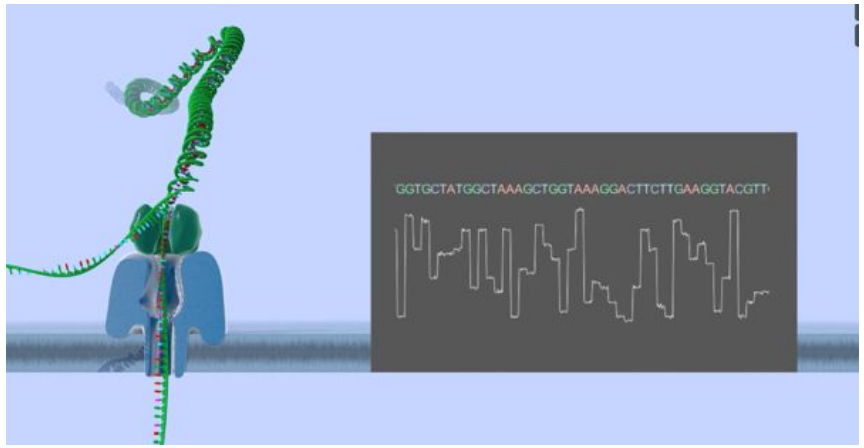
De Bruijn Graph

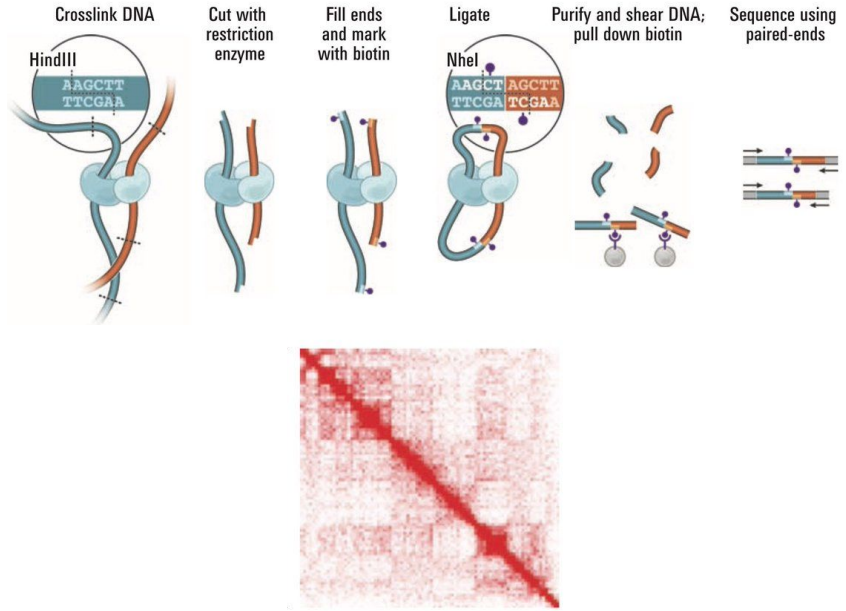
Currently the most popular assembly method

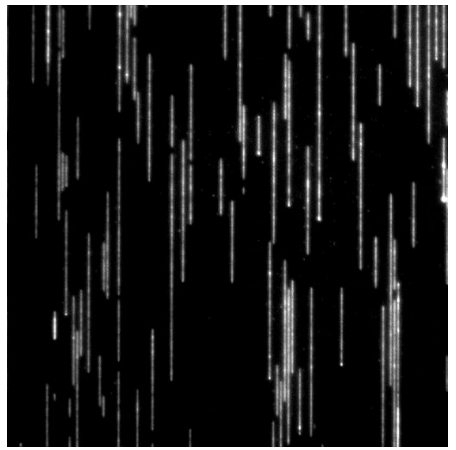
Many software tools use variants of the method

- SPAdes
- SOAPdenovo
- AbySS
- MEGAHIT

Emerging Technologies for Genome Assemblies







Long reads **Hi-C** **Optical mapping**

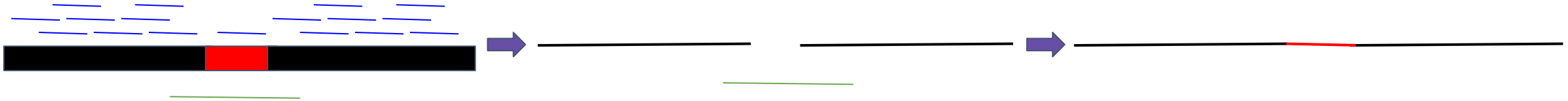
Long and Linked Reads in Genome Assembly

Many modern assemblers can work with 3rd generation reads

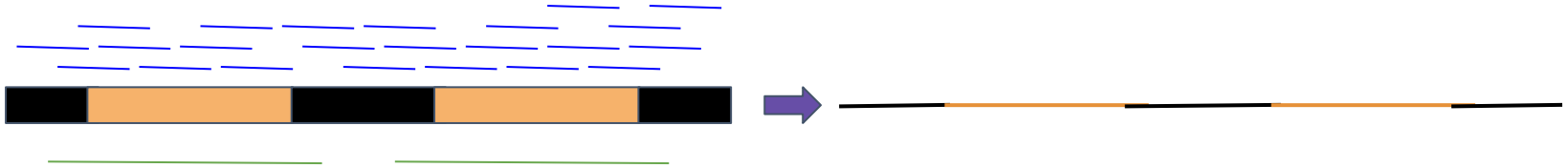
- Falcon - PacBio reads
- Canu, SPAdes - PacBio and ONT reads
- Supernova - 10X reads

Most assemblers take a “hybrid” approach - long + short reads

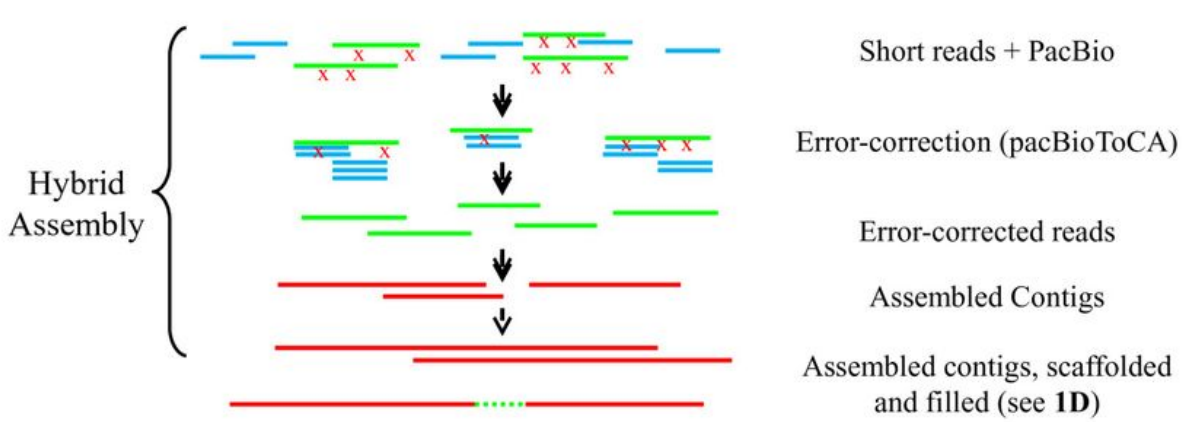
Long/linked reads can help link contigs by bridging over difficult regions

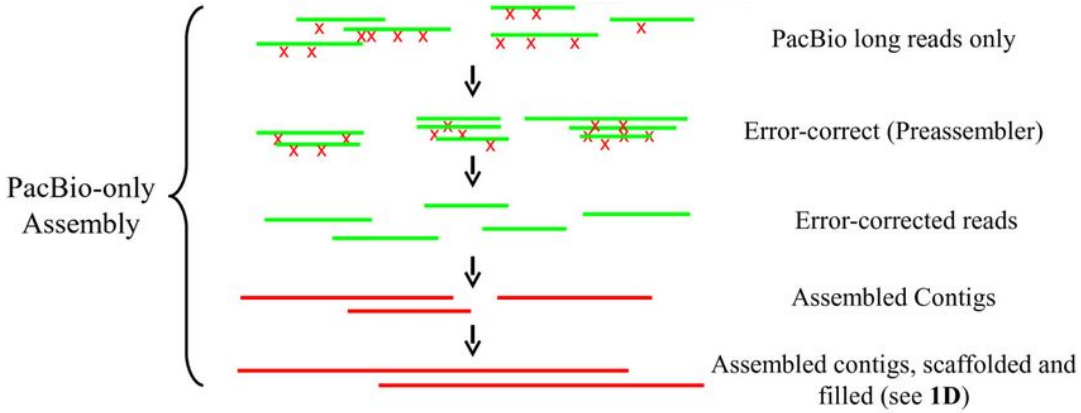


Long reads can help solve long repeats



Different Assembly Strategies





Haplotype Phasing

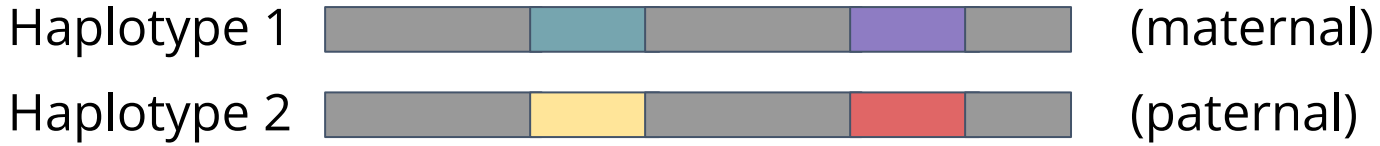
Many interesting eukaryote genomes are diploid or polyploid

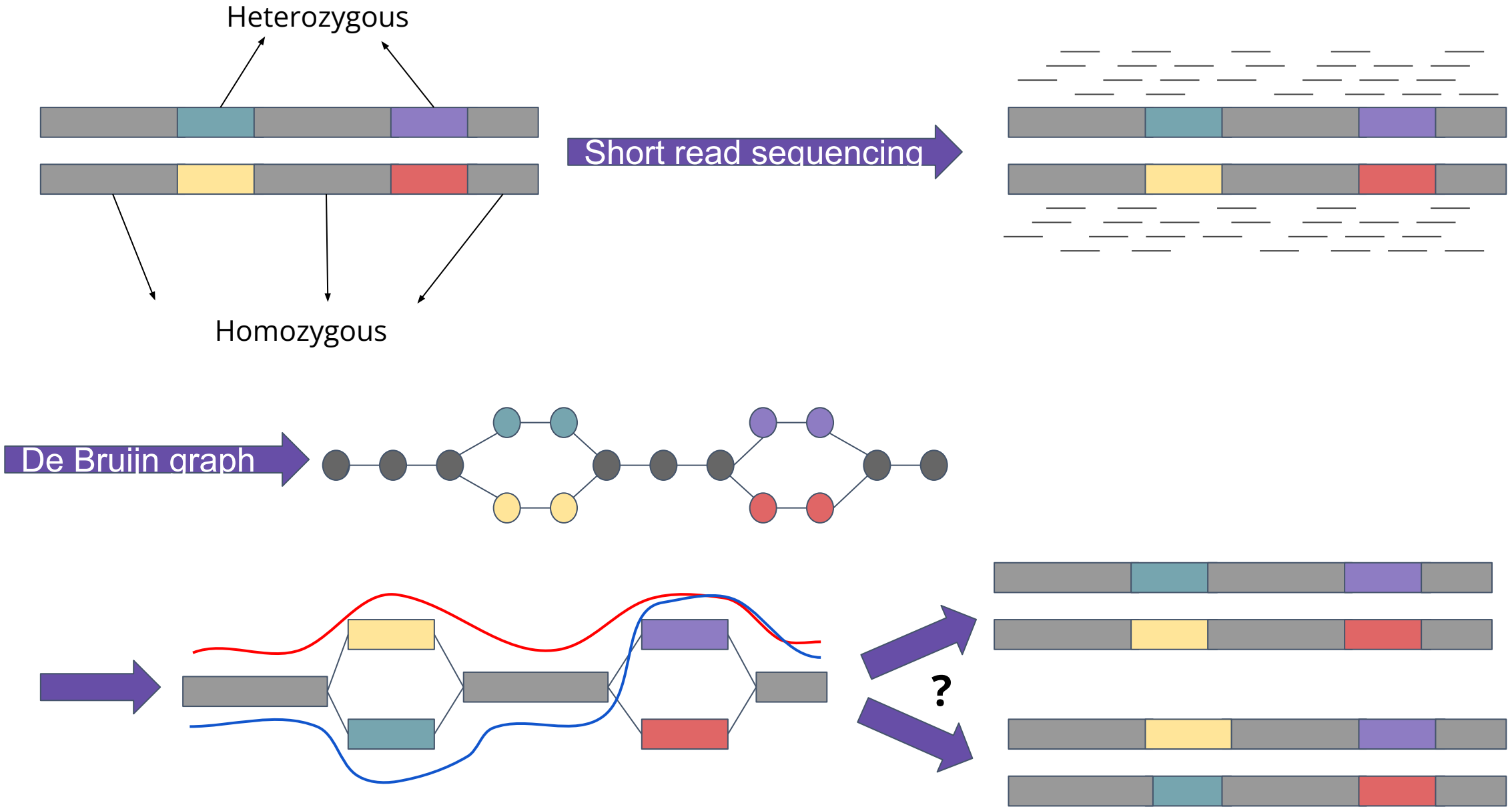
Still, most assemblies are haploid

Heterozygosity is “squished” into consensus sequences

A **haplotype** is a group of alleles arising from the same molecule

Splitting an assembly into haplotypes is called **phasing**




Heterozygous
Homozygous
Short read sequencing
De Bruijn graph
?

DNA Sequencing and Data Analysis

RNA-Seq

Lesson Goals

Understand some common RNA-seq uses and protocols

Be familiar with the basic workflow of gene expression data analysis

- Specifically differential gene expression analysis

Know how to map RNA-seq reads to a reference genome using STAR

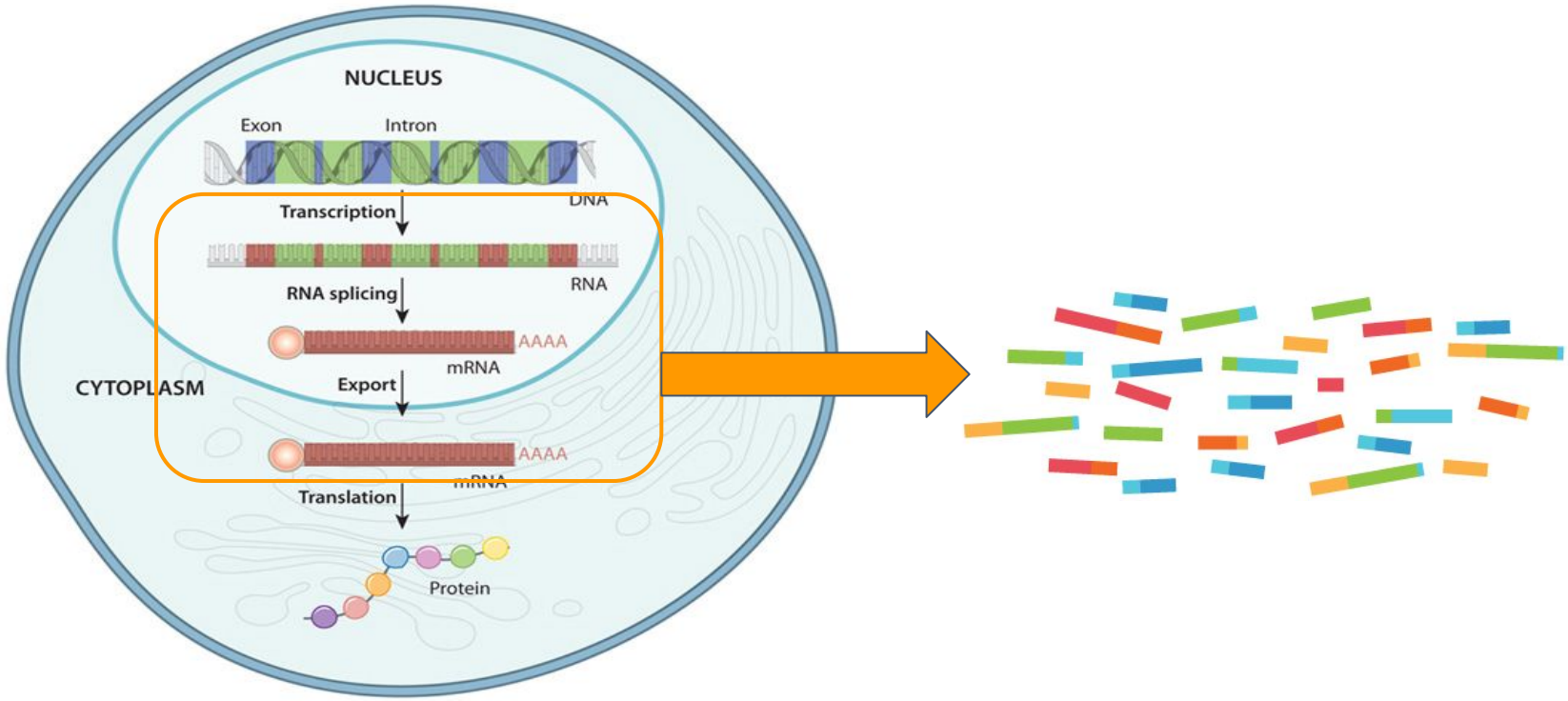
Perform Differential Gene Expression (DGE) analysis

What is RNA-Sequencing?

Sequencing of RNA using NGS technology

Allows assessment of presence and quantity or RNA in a sample

Take a snapshot of expression in a sample



Why Study the Transcriptome?

The full range of messenger RNA molecules expressed by an organism

- Indication of cell physiology
- Dynamic - responds to the environment
  - Changes over time
  - Responds to external stimuli
  - Controls cellular processes
- Reduced representation of the genome
  - Smaller = cheaper
  - Only the “functional” parts of the genome

Types of Expression Analysis

Expression quantification

Differential gene expression

Assemble whole transcriptomes

Detect new transcripts

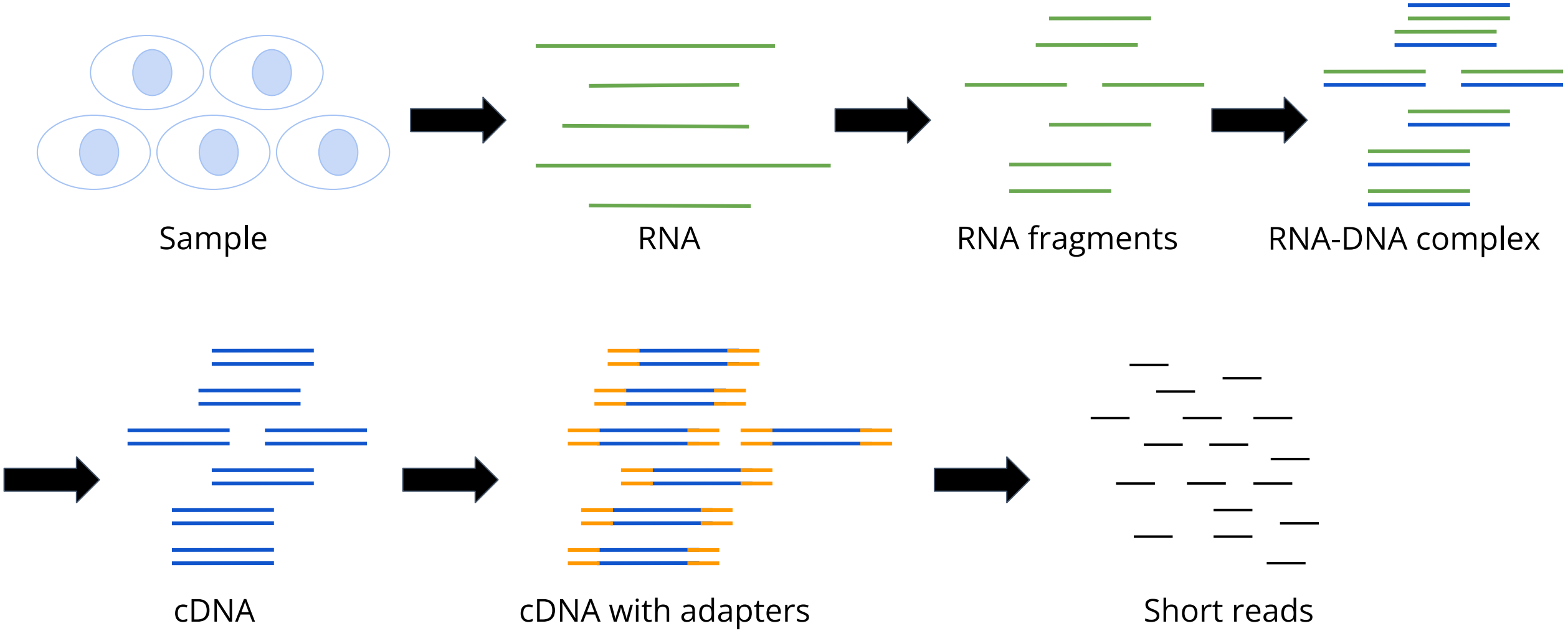
Detect splicing variants

Detect allele-specific expression

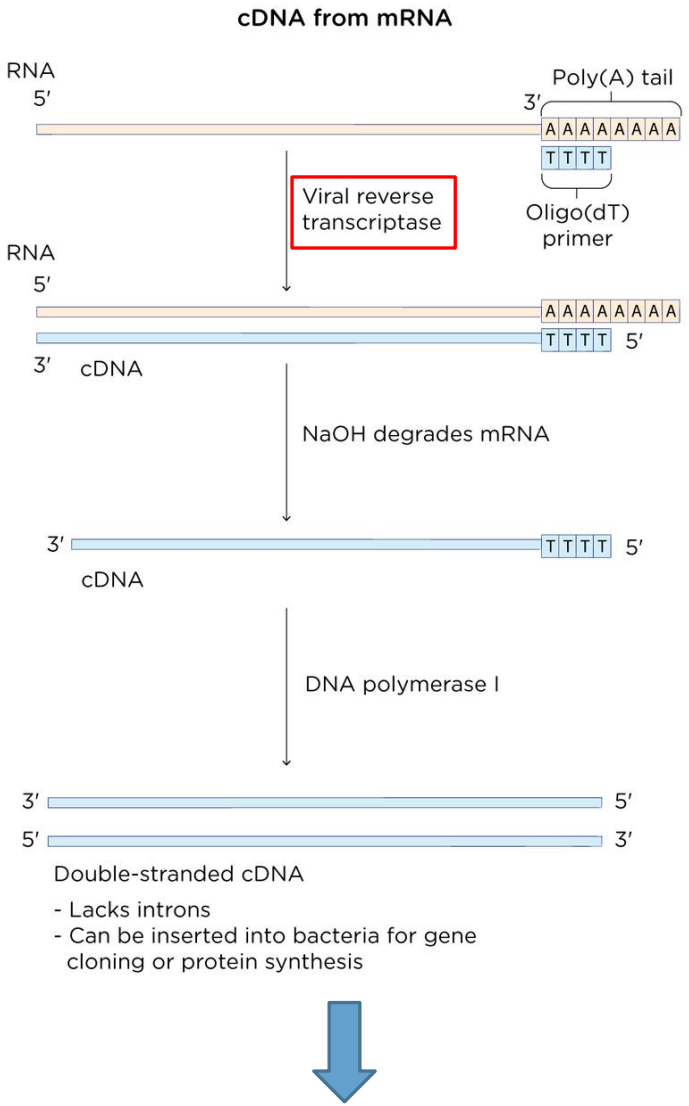
Gene co-expression

Single-cell analysis

RNA-Seq Basic Protocol



cDNA



And continue as before!

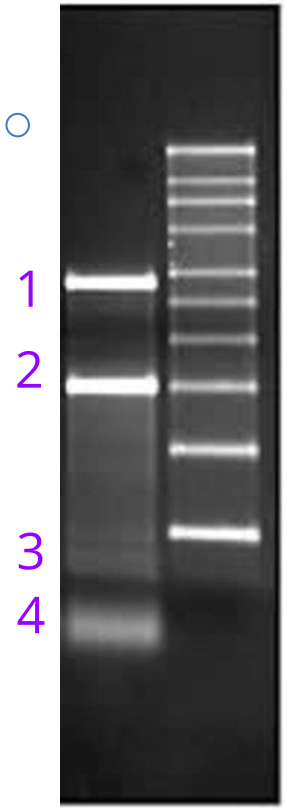
RNA is unstable and can easily degrade, while cDNA is stable and easier to work with!

cDNA (complementary DNA) Libraries:

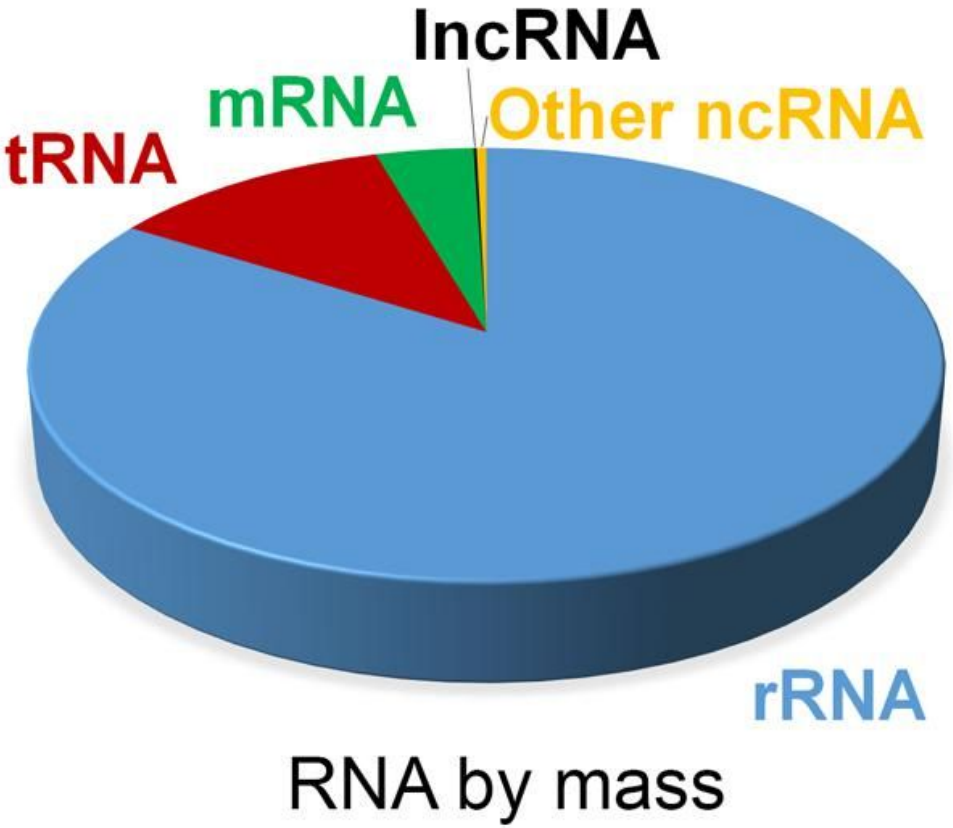
- Collections of only the coding regions of an organism's genes.
- The genes being expressed at a certain time point.
- Help in studying gene expression, understanding disease mechanisms, and can be used in drug discovery.
- Help to study alternative splicing, mutations, and gene fusion events in cancer studies.
- Insertion into a vector, and transformation into bacteria for amplification.

Total RNA

Which band contains the mRNA?



Total RNA (Mammalian)

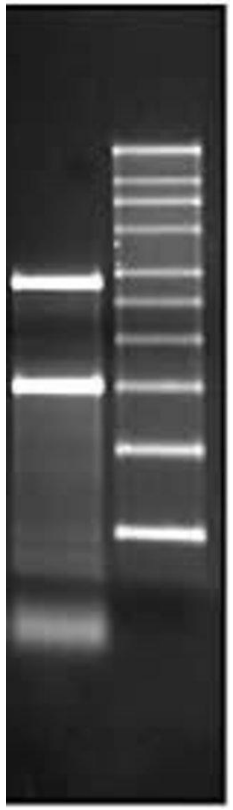


rRNA large subunit

rRNA small subunit

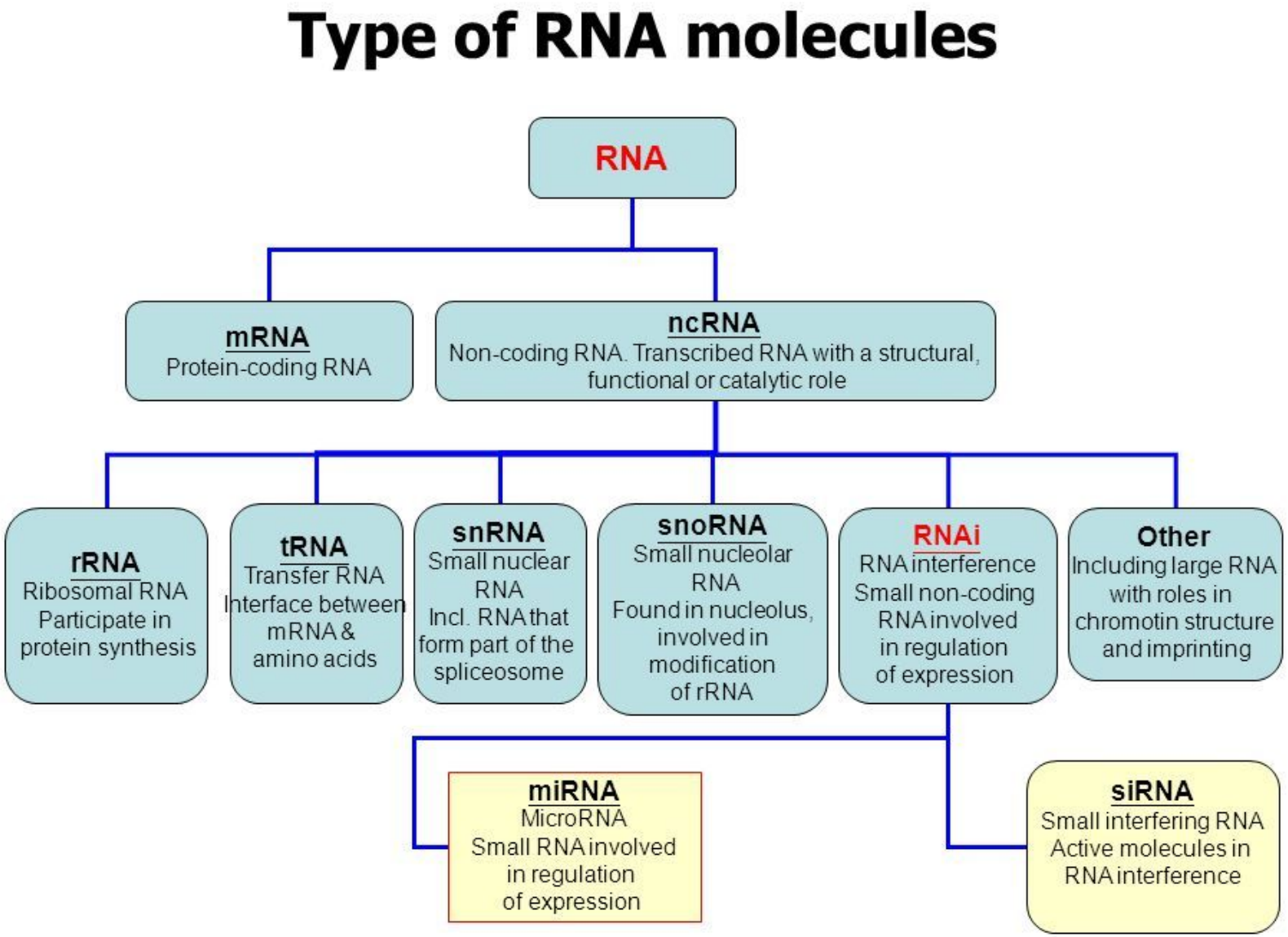
mRNA

tRNA



Palazzo, Alexander F., and Eliza S. Lee. "Non-coding RNA: what is functional and what is junk?." *Frontiers in genetics* 6 (2015): 2.

RNA


Type of RNA molecules
RNA
mRNA
Protein-coding RNA
ncRNA
Non-coding RNA. Transcribed RNA with a structural, functional or catalytic role
rRNA
Ribosomal RNA
Participate in protein synthesis
tRNA
Transfer RNA
Interface between mRNA & amino acids
snRNA
Small nuclear RNA
Incl. RNA that form part of the spliceosome
snoRNA
Small nucleolar RNA
Found in nucleolus, involved in modification of rRNA
RNAi
RNA interference
Small non-coding RNA involved in regulation of expression
Other
Including large RNA with roles in chromotin structure and imprinting
miRNA
MicroRNA
Small RNA involved in regulation of expression
siRNA
Small interfering RNA
Active molecules in RNA interference

Enrichment for Mature RNA

Poly-A selection method

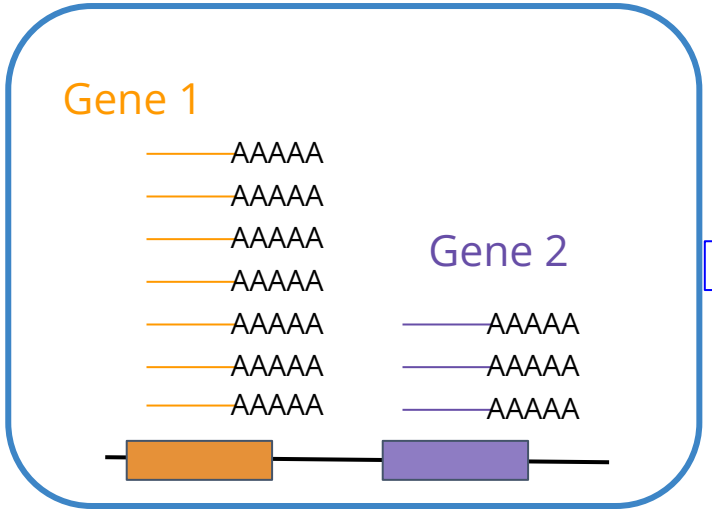
- use a poly-dT baits to bind mRNAs and discard the rest
- Removes rRNA, tRNA and others
- Enriches for mature mRNA (containing poly-A tail)
- Not all mRNAs have poly-A tails
  - Histones mRNA in Metazoans
  - Mitochondrial mRNA

rRNA depletion method

- Use baits designed specifically for rRNA
- Does not remove tRNA
- Only option in bacteria (no poly-A tail)
- Only option when extracting small RNAs

Expression Levels

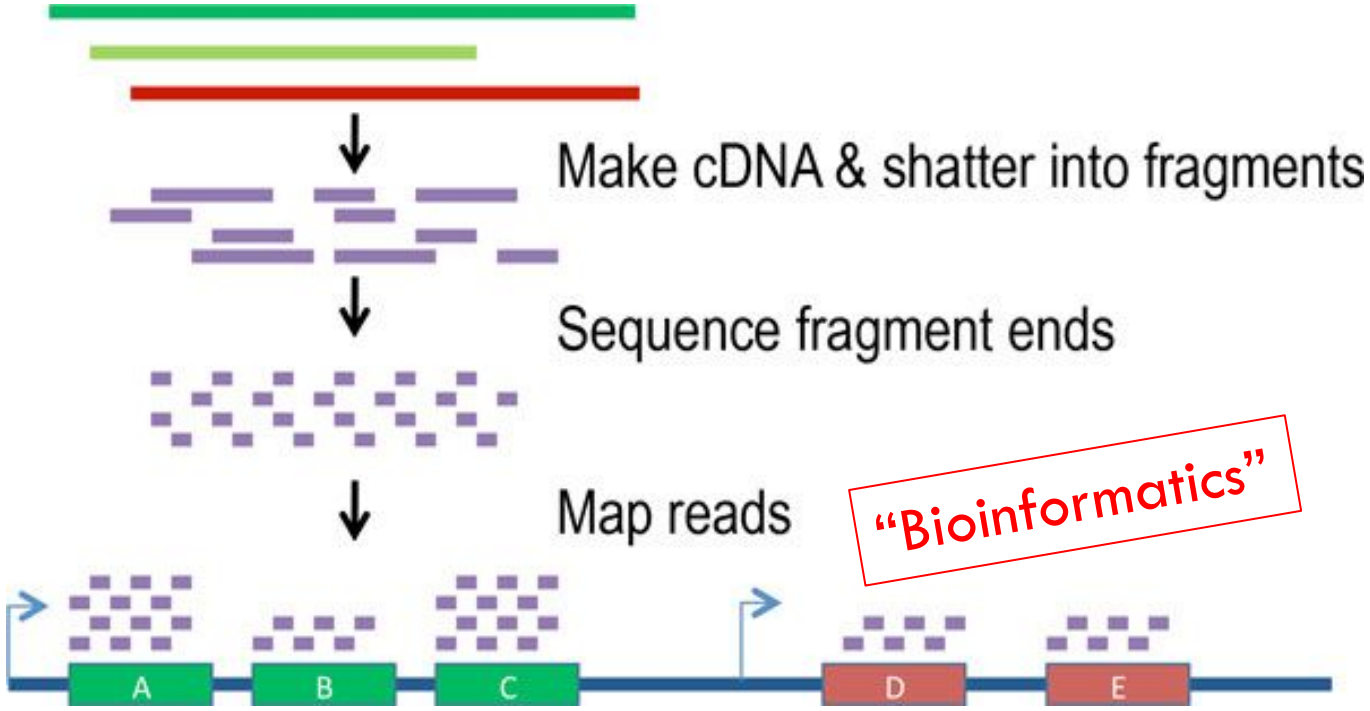
Higher expression → more transcripts → more RNA-seq reads



Gene	Reads
Gene 1	7
Gene 2	3

Summary

Reveal the presence and quantity of RNA in a biological sample at a given moment (mostly mRNA)

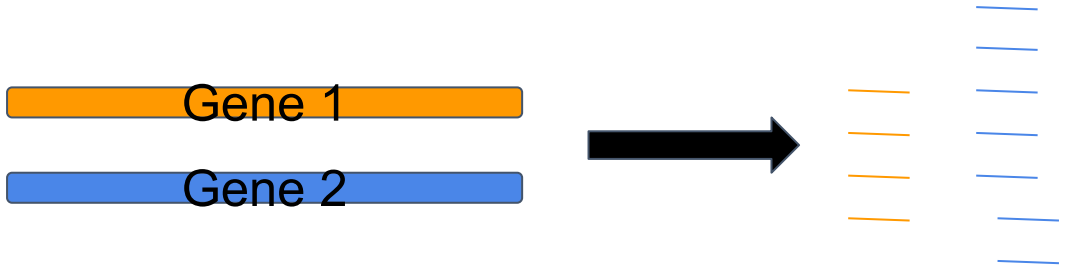


https://www.youtube.com/watch?v=tlf6wYJrwKY

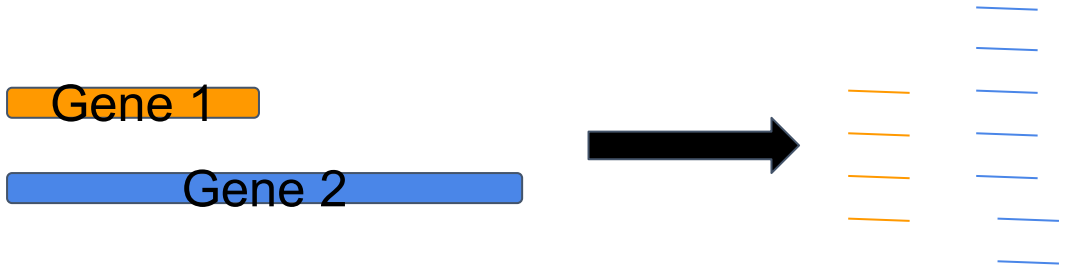
Are Read Counts a Good Measure?

Variable length of transcript of interest

Experiment 1

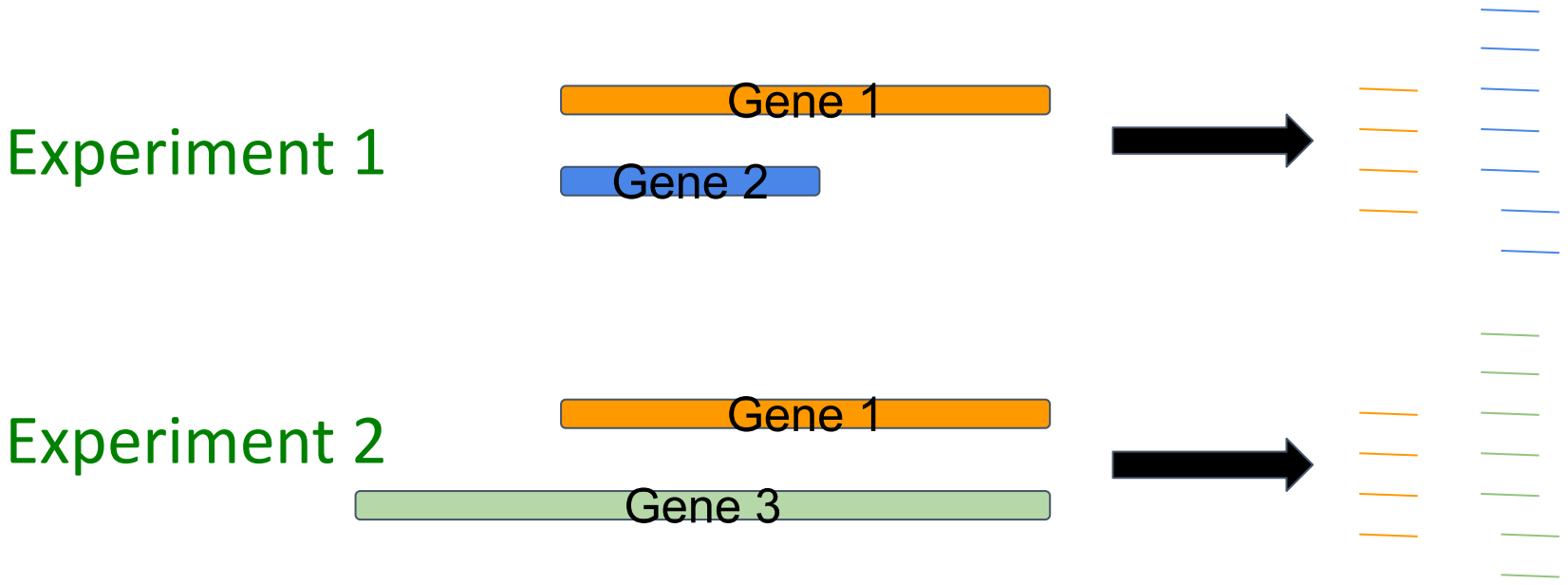


Experiment 2



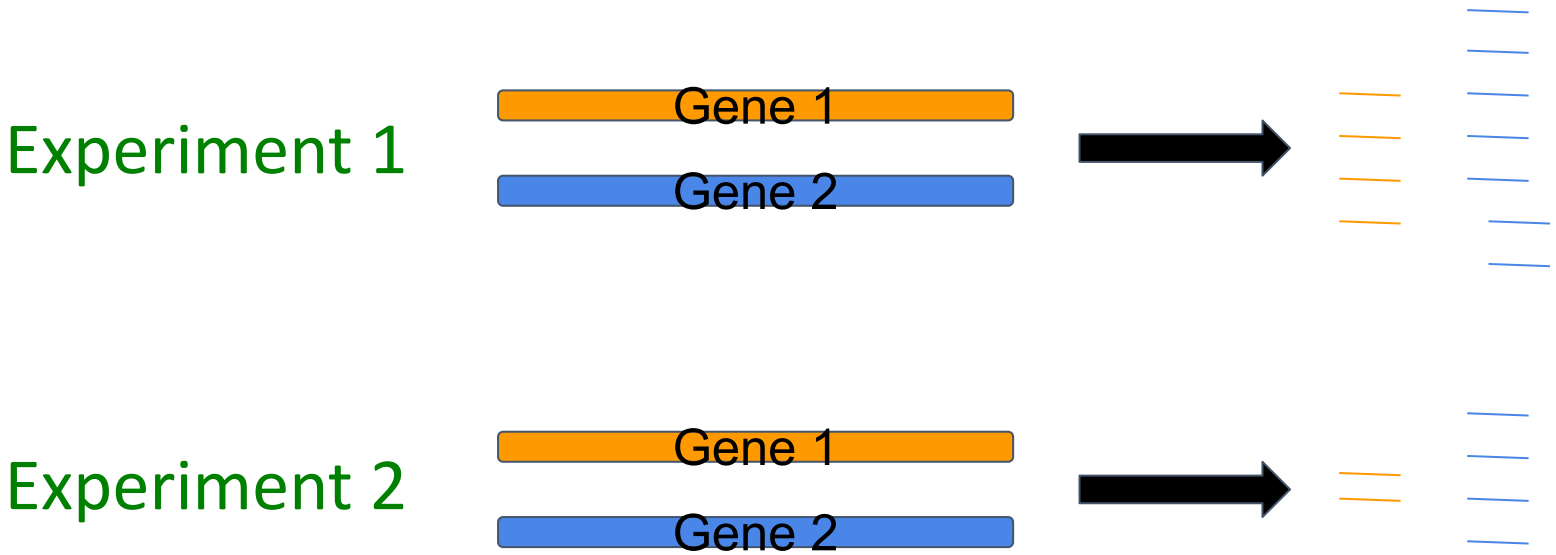
Are Read Counts a Good Measure?

Variable length of other transcripts



Are Read Counts a Good Measure?

Variable number of reads between experiments



Read Count Normalization

Normalize for transcript length

RPK - **reads per kilobase** (of transcript)

$$RPK_i = 10^3 \cdot \frac{n_i}{l_i}$$

Normalize for sequencing depth

$$RPKM_i = 10^9 \cdot \frac{n_i}{l_i \cdot \sum_j n_j}$$

RPKM - **reads per kilobase** (of transcript) **per million** (reads)

Read Count Normalization

Experiment 1 - total reads: 100,000

Gene	Transcript length	Reads	RPK	RPKM
Gene1	500	10	20	200
Gene2	1000	20	20	200

Experiment 2 - total reads: 1,000,000

Gene	Transcript length	Reads	RPK	RPKM
Gene1	500	10	20	20
Gene2	1000	50	50	50

Differential Gene Expression (DGE)

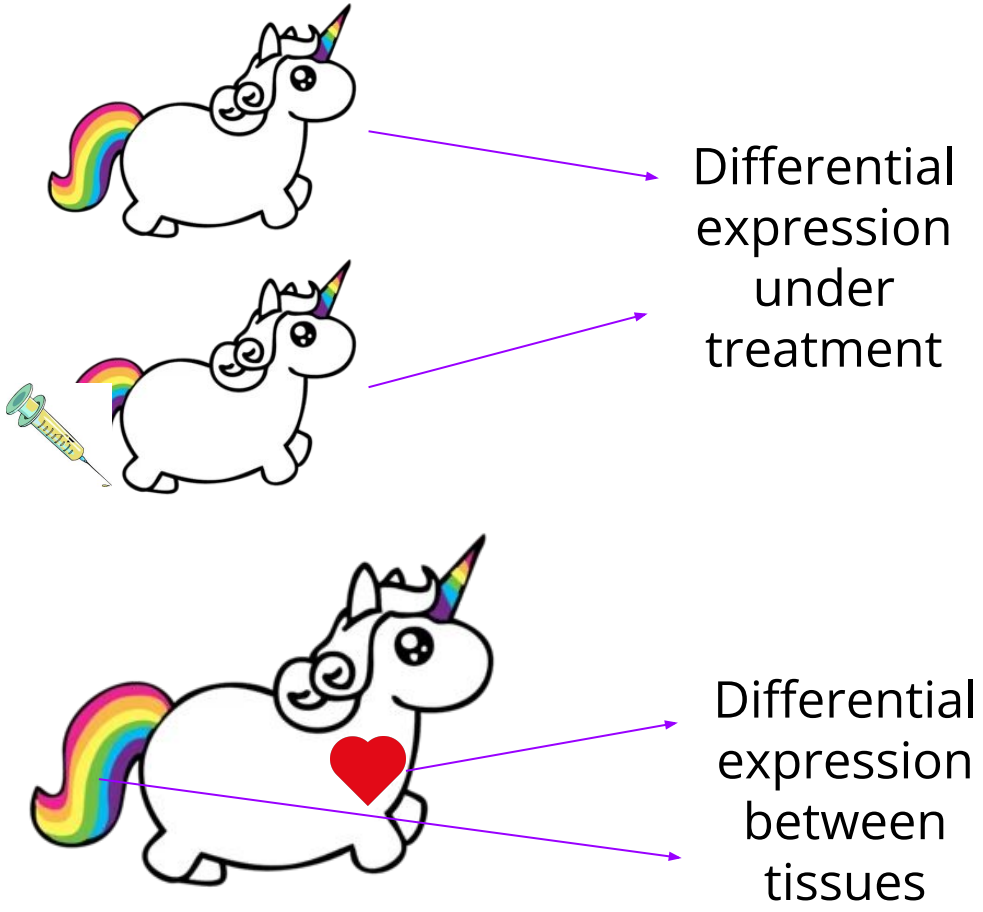
Compare gene expression levels across all genes between two (or more) samples

Samples of the same cell type

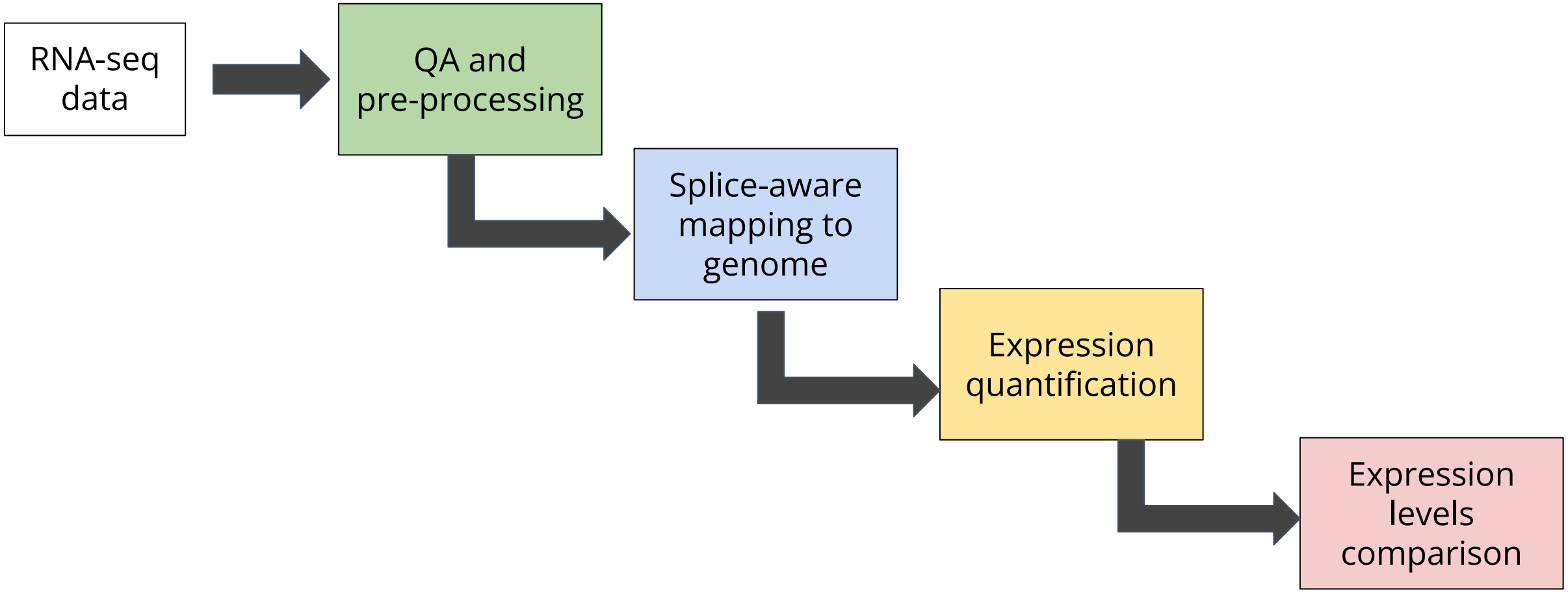
- Different physiological conditions
- Different environmental conditions
  - Growth medium
  - Treatment

Samples of different cell types

- Different tissues
- WT vs. mutant
- Different strains
- Normal vs. tumor

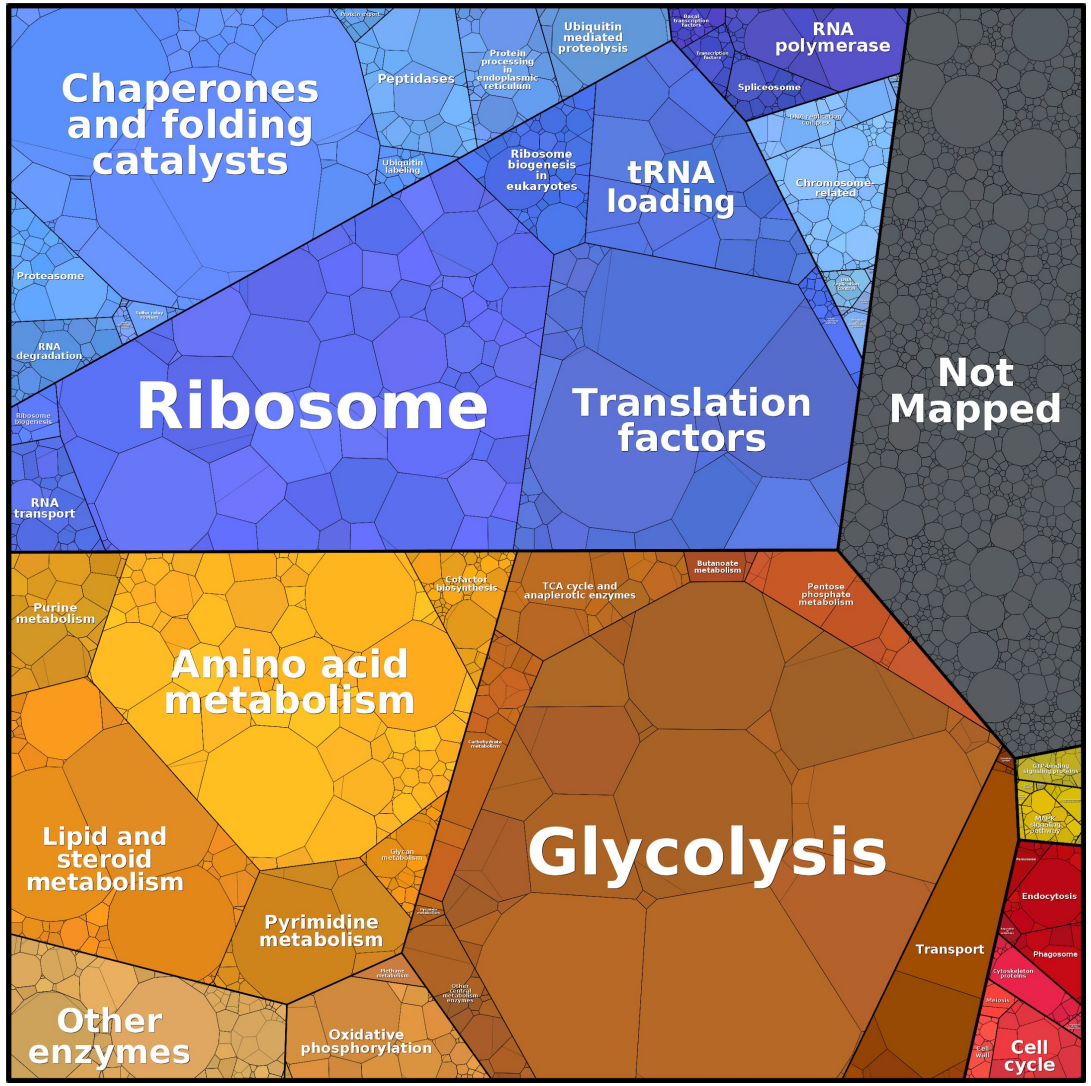


DGE Workflow



Exploring Expression Levels Data

Expression levels differ in orders of magnitude between genes

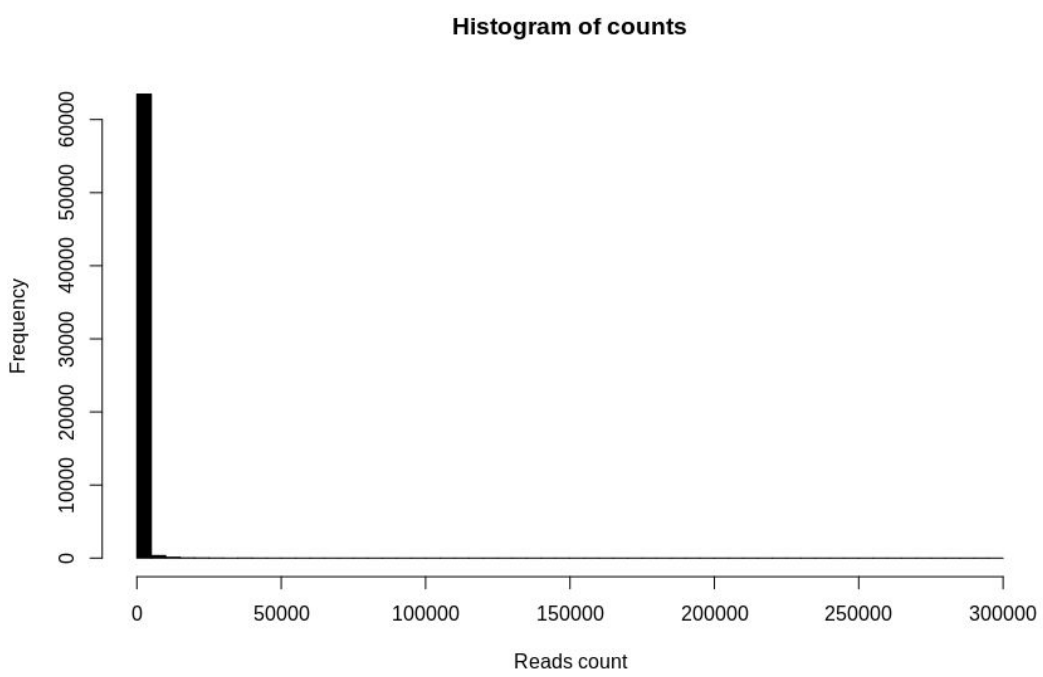


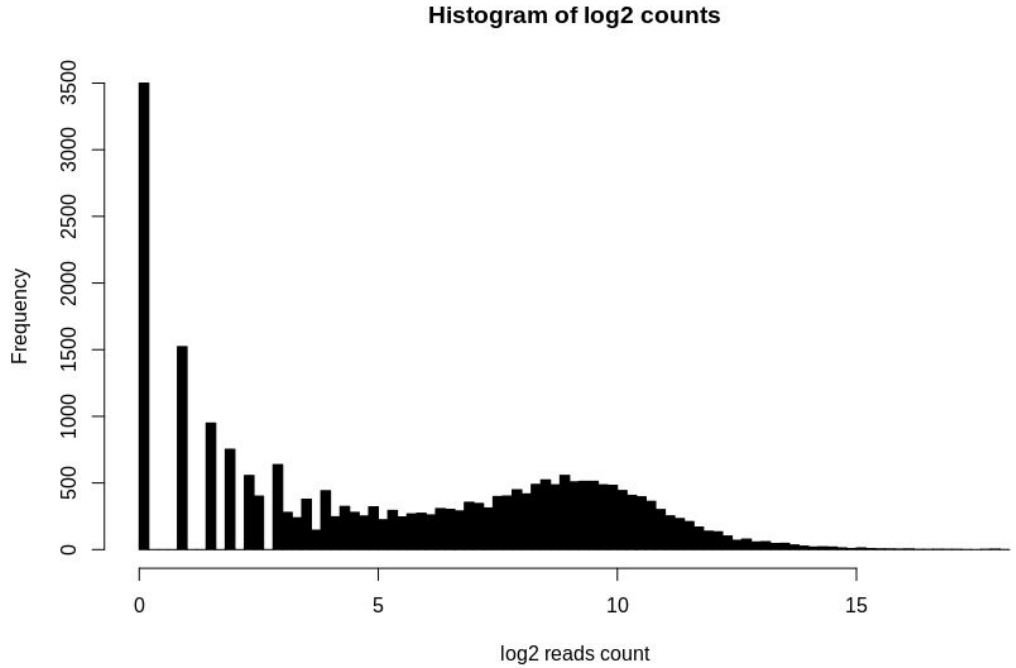
Liebermeister W., Noor E., Flamholz A., Davidi D., Bernhardt J., and Milo R. (2014), Visual account of protein investment in cellular functions. PNAS 111 (23), 8488-8493.

Log Transformation

We usually apply a $\log_2$ transformation to read counts

Makes it easier to explore the data





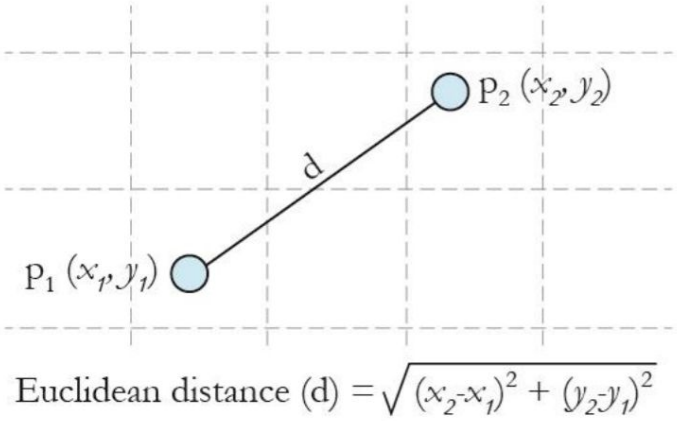
QA Expression Quantification

Which samples are overall similar/different from one another?

Does it match our expectations, given the experimental design?

We can use Euclidean distances:

In 2 dimensions

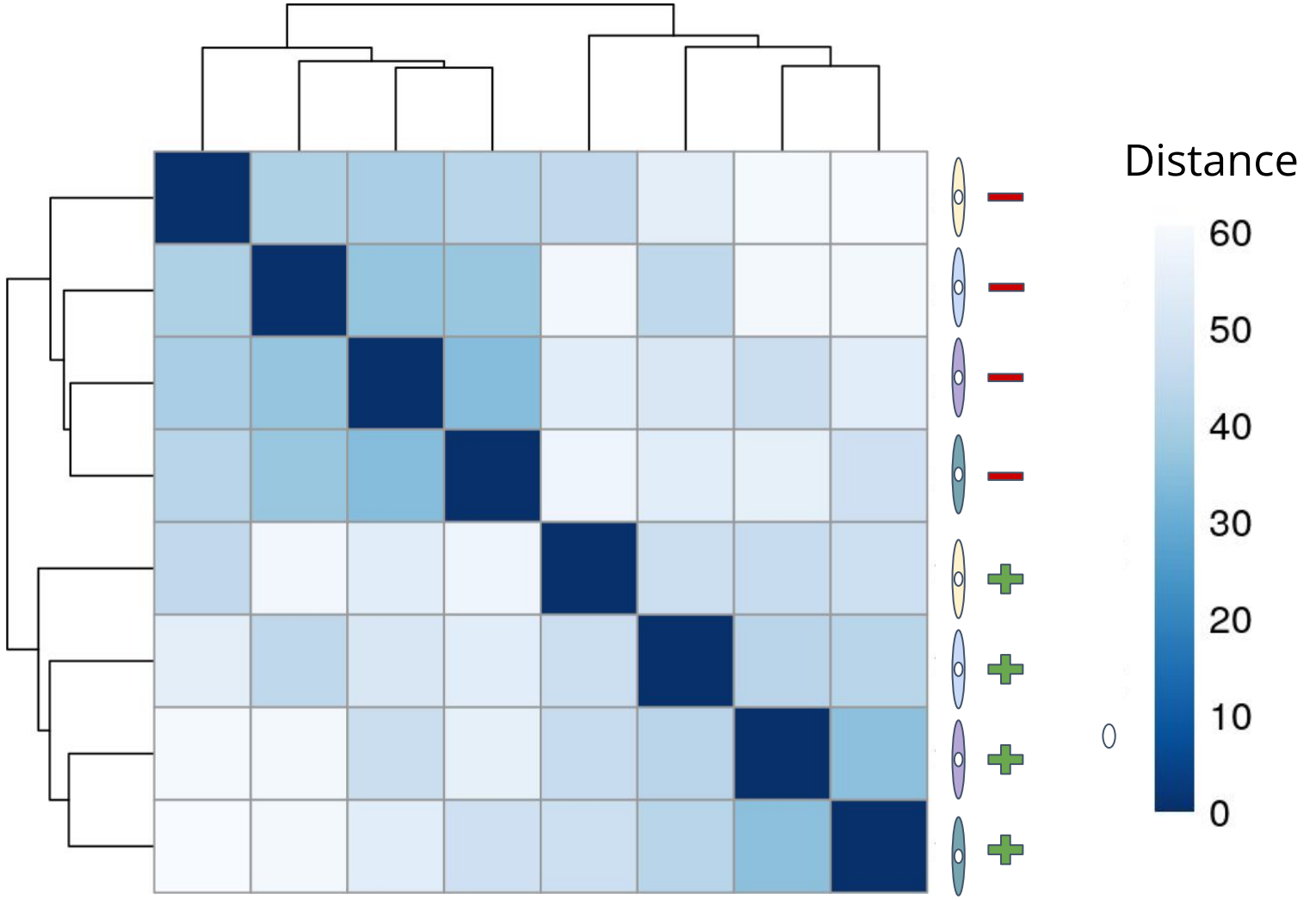


In n dimensions

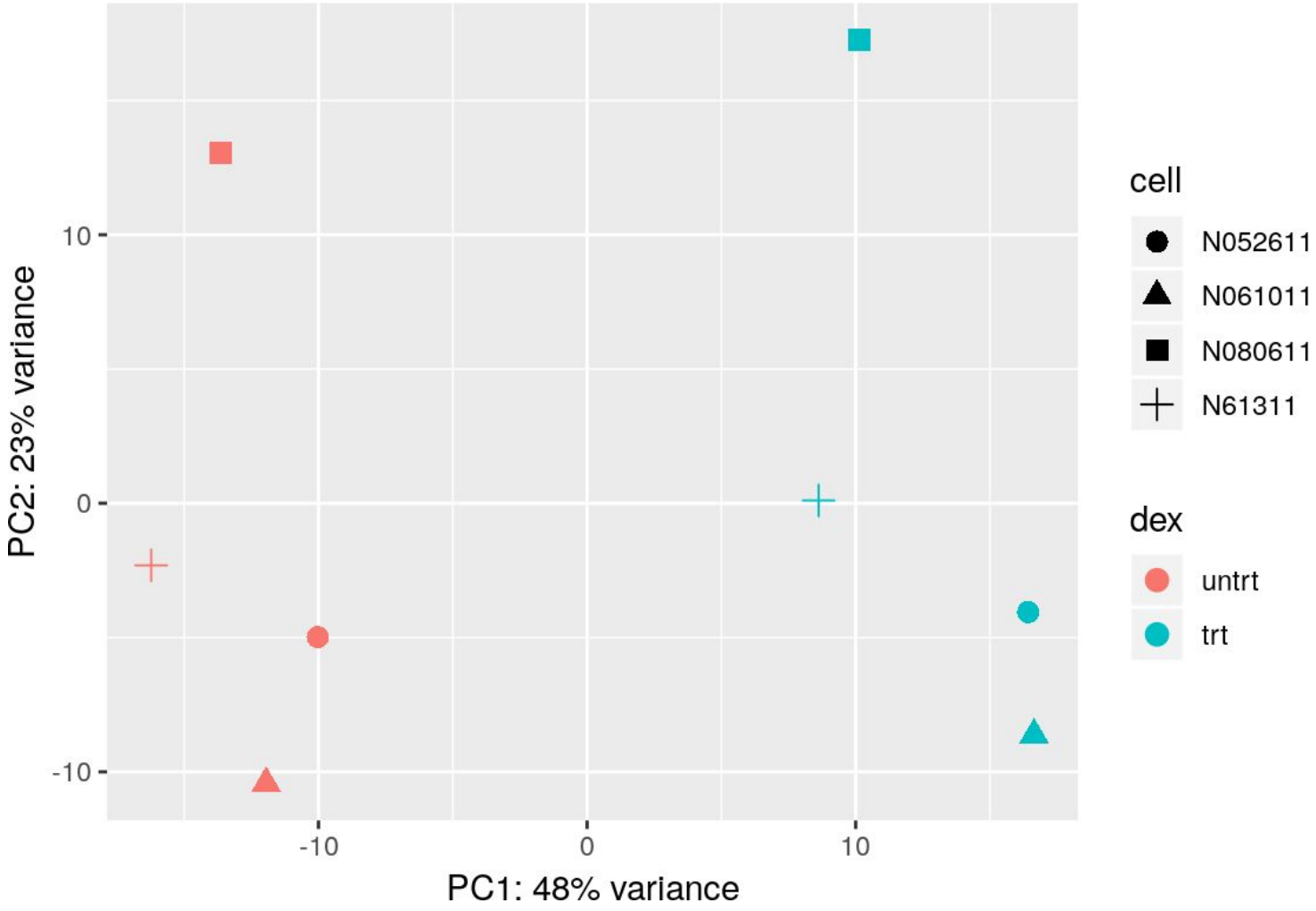
	sample 1	sample2
gene1	m_{11}	m_{12}
gene2	m_{21}	m_{22}
...	...	...
gene n	m_{n1}	m_{n2}

$$d = \sqrt{(m_{12} - m_{11})^2 + (m_{22} - m_{21})^2 + \ldots + (m_{n2} - m_{n1})^2}$$

Hierarchical Clustering



PCA


PC2: 23% variance
10
0
-10
-10
0
10
PC1: 48% variance
cell
N052611
N061011
N080611
N61311
dex
untrt
trt

Filtering Counts Data

It is useful to remove genes with very few reads from the analysis

- Slow down the analysis
- Reduce detection power for other genes

We won't be able to detect DE anyway

We can choose a count cutoff

Or we can remove the X^{th} percentile

In the Himes et. al data:

~64k transcripts ⟶ Require >= 10 reads ⟶ ~20k transcripts

Differential Expression Analysis

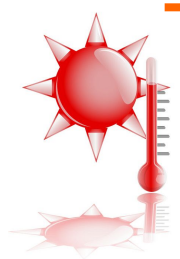
Goal: detect genes that significantly differ in their expression levels between samples

Input:

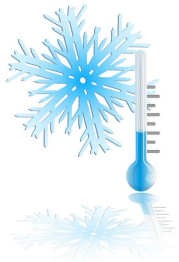
- Normalized, filtered expression quantification matrix
- Description of the experimental design

Output: per gene - estimated difference between samples and **significance level**

Batch Effects



Day 1



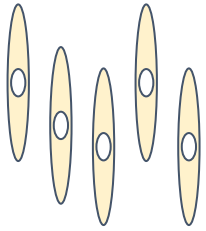
Day 2

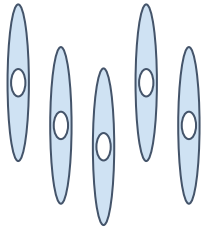
N61311

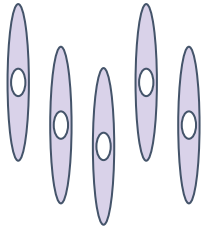
N052611

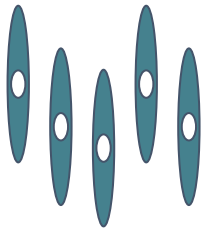
N080611

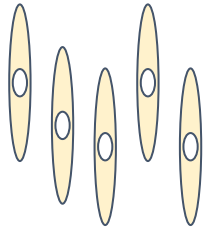
N061011

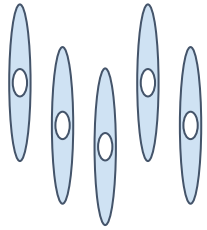


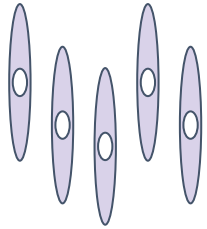


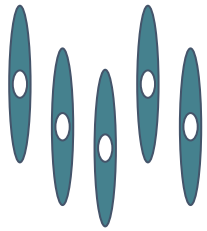


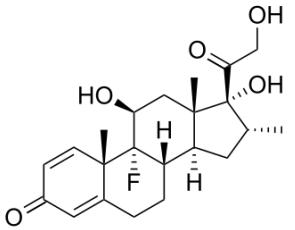


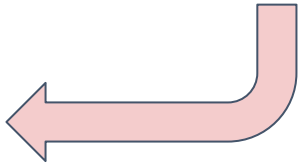












Batch Effects

Introduced during sample handling and preparation

- Technical factors
- External factors

Minimize by:

- Use the same protocol for all samples
- Prepare all samples together

Not always possible

We must test for batch effects when performing DE statistical tests

Batch Effects

Sample	Cell line	Dex	Batch
SRR1039508	N61311	Untreated	1
SRR1039509	N61311	Treated	1
SRR1039512	N052611	Untreated	1
SRR1039513	N052611	Treated	1
SRR1039516	N080611	Untreated	2
SRR1039517	N080611	Treated	2
SRR1039520	N061011	Untreated	2
SRR1039521	N061011	Treated	2

Fold Change

The main measure used in DGE analysis is **fold change** - a.k.a ratio

Ratios are highly non-symmetric $$R = \frac{Count_{sample1}}{Count_{sample2}}$$

Therefore we use log scaling - **log2 fold change (L2FC)**

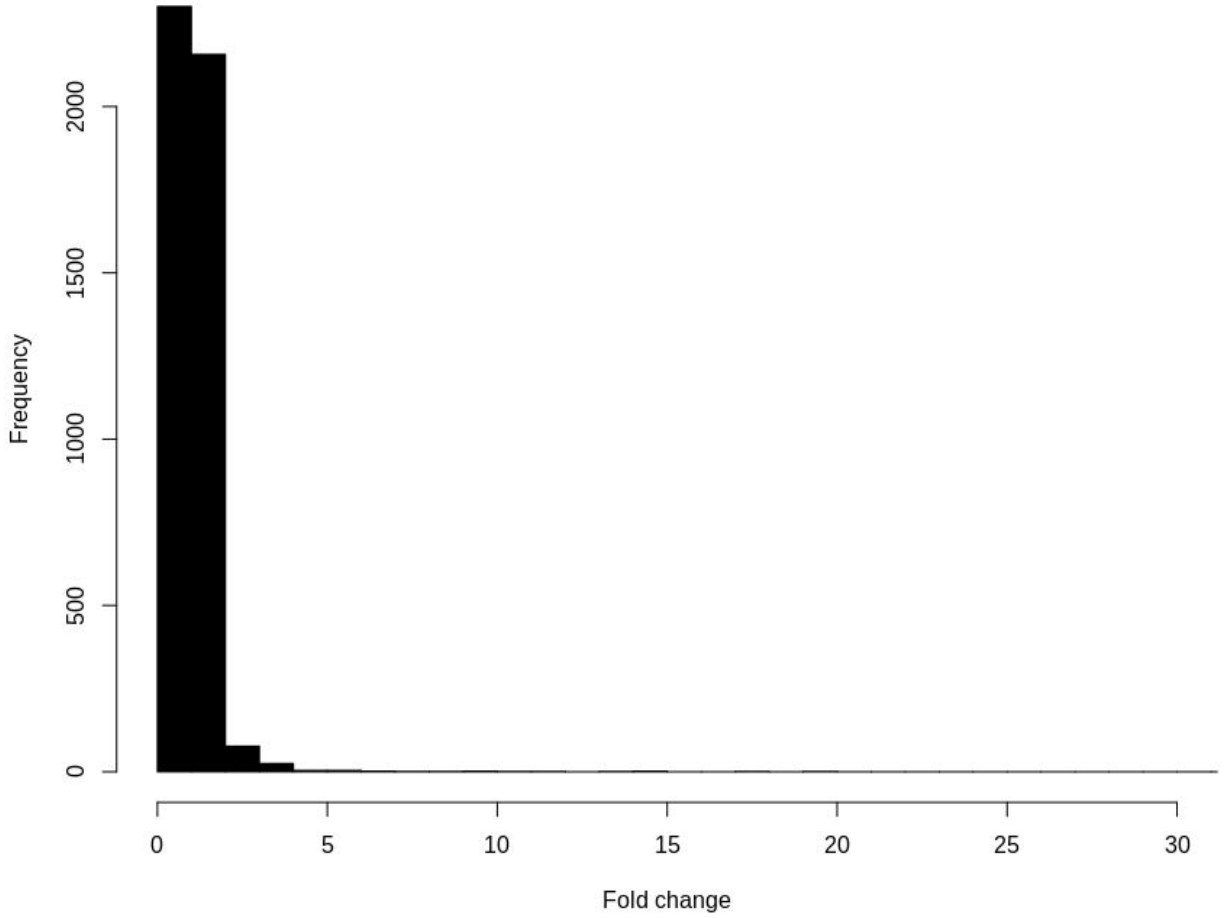
$$L2FC = \log_2(\frac{Count_{sample1}}{Count_{sample2}}) = \log_2 Count_{sample1} - \log_2 Count_{sample2}$$

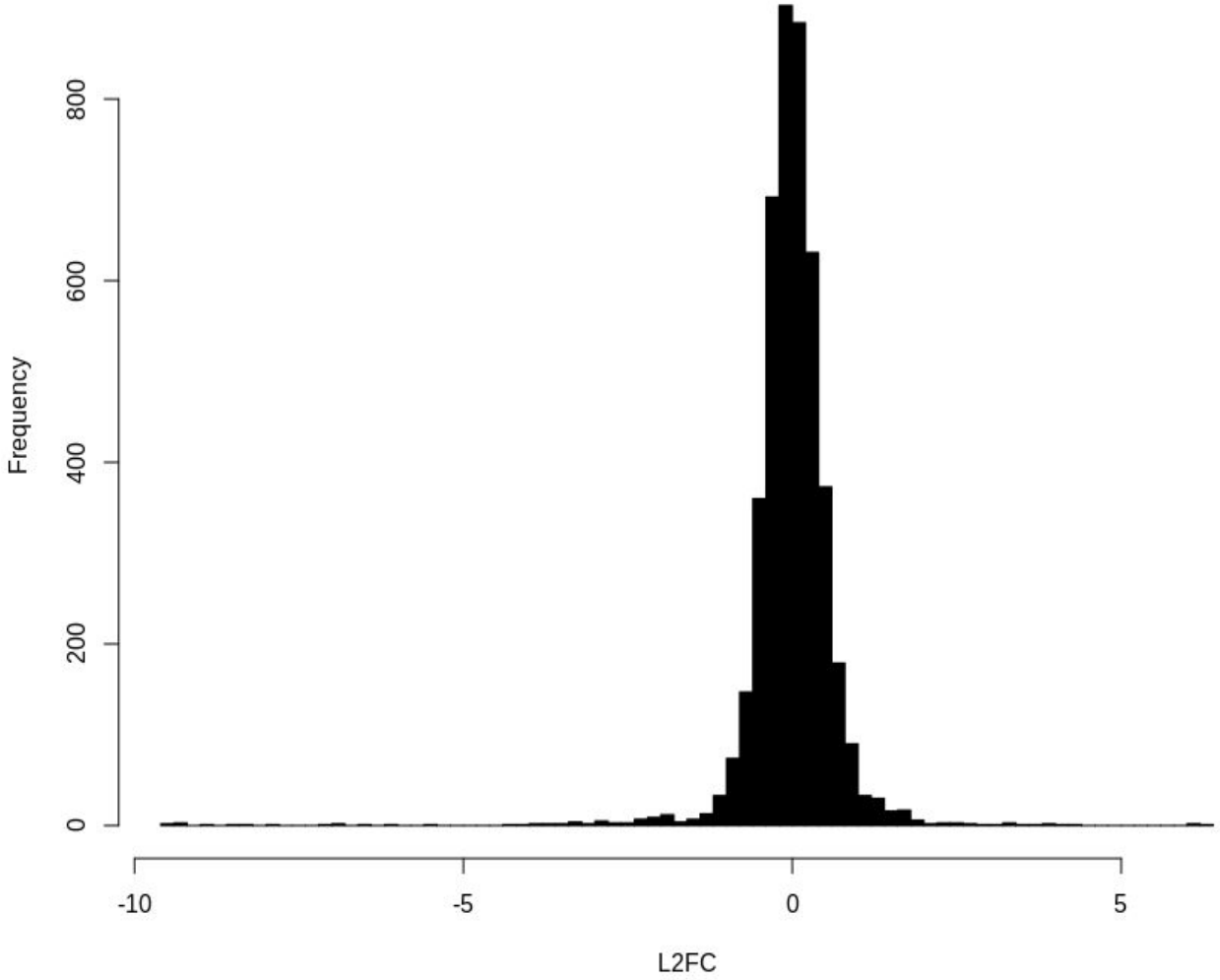
Fold Change

	N61311 Dex	N61311 Dex	N05261 1 Dex	N08061 1 Dex	N61311 Unt	N61311 Unt	N05261 1 Unt	N08061 1 Unt	Mean Dex	Mean Unt	FC	L2FC
Gene 1	34	512	66	121	25	344	297	76	183.25	185.50	0.99	-0.02
Gene 2	1112	985	1003	898	214	128	188	203	999.50	183.25	5.45	2.45
Gene 3	6	9	4	6	12	9	15	16	6.25	13.00	0.48	-1.06

L2FC = 0 : no difference
L2FC > 0 : sample1 expression > sample 2 expression
L2FC < 0 : sample1 expression < sample 2 expression

Fold Change


Frequency
0
500
1000
1500
2000
0
5
10
15
20
25
30
Fold change


Frequency
0
200
400
600
800
-10
-5
0
5
L2FC

Log2 Fold Changes

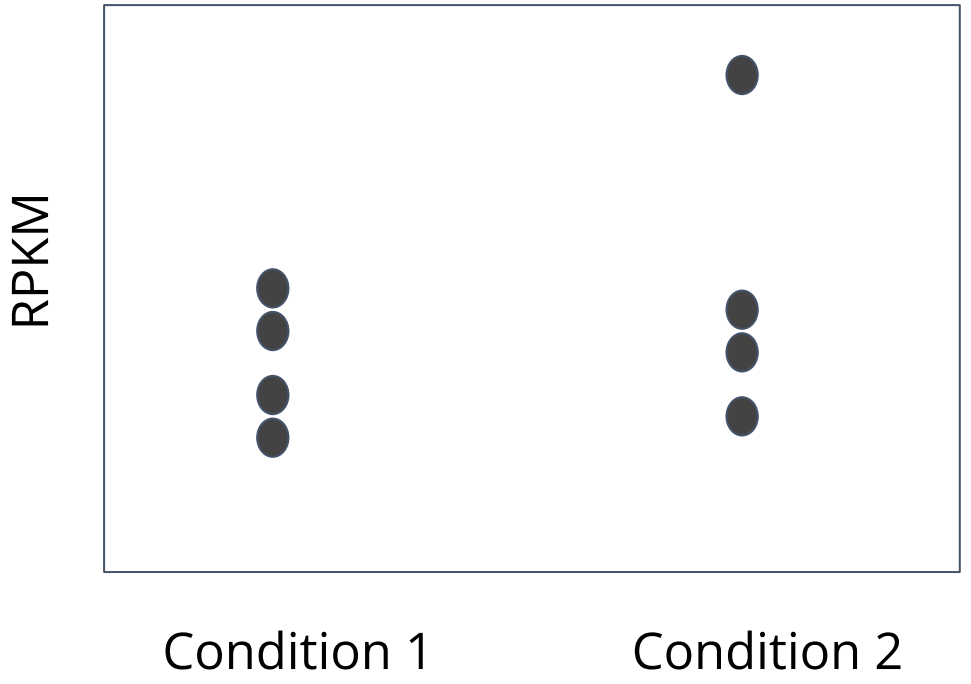
Can we tell just by looking at L2FC values?

Maybe it’s just random noise?

Replicates can help

Gene X - L2FC = 0.5

Gene Y - L2FC = 0.4



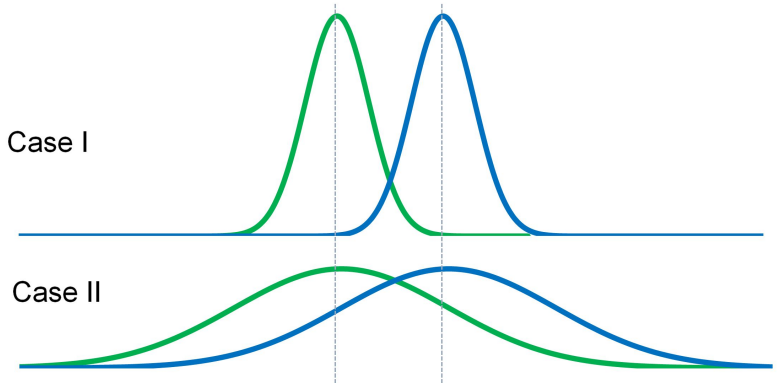
Hypothesis Testing

H_0: there is no difference in expression levels between samples
H_1: expression levels differ between samples

We try and reject H_0 with an appropriate statistical test, e.g.:

- Parametric tests: *t*-test, ANOVA
- Non-parametric tests: Mann-Whitney U test
- Other modeling methods: linear models, GLM

The result is a significance score - **per gene p-value**

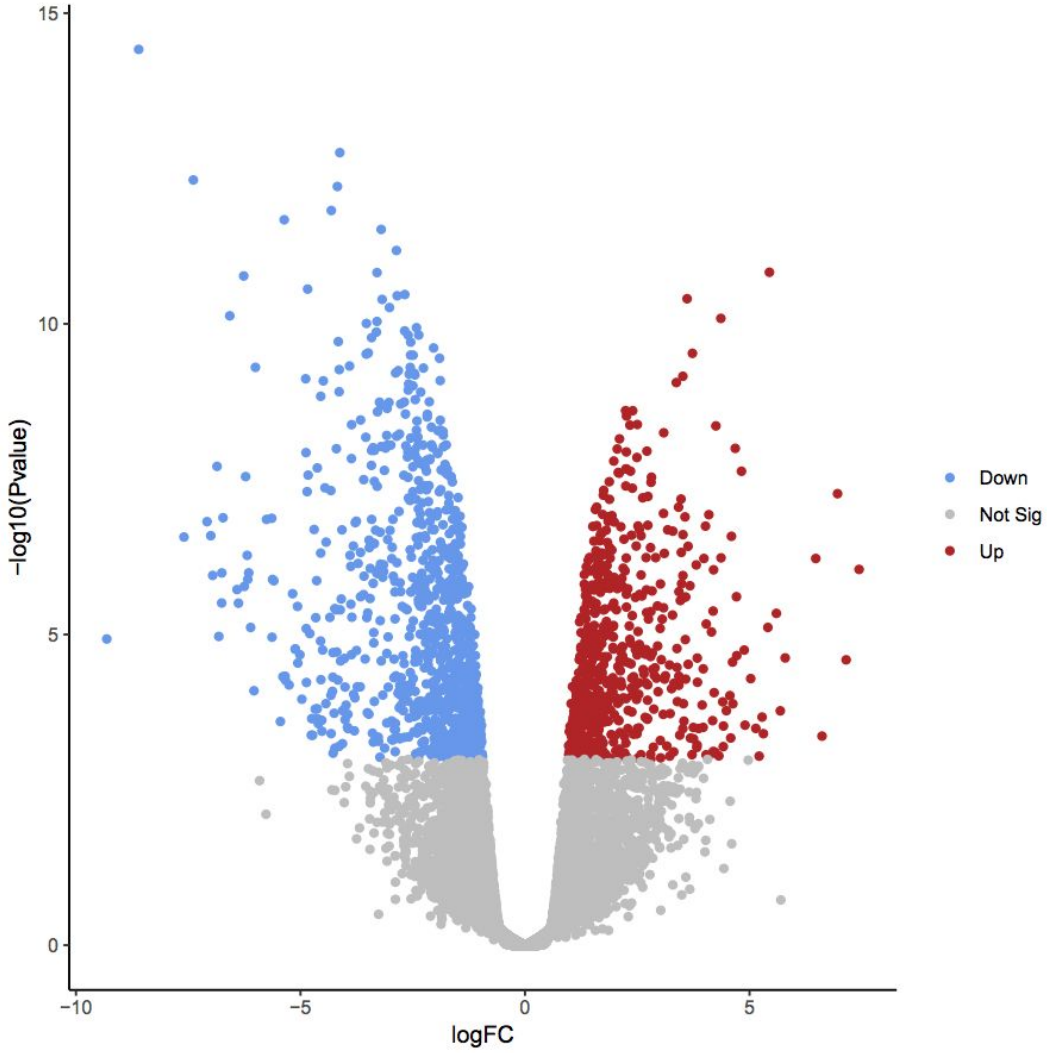


Volcano Plots

For each gene, we must consider both L2FC and p-value

To get a global view - use a **volcano plot**

We can choose a p-value cutoff, e.g. 0.05 or 0.01



Cutoff

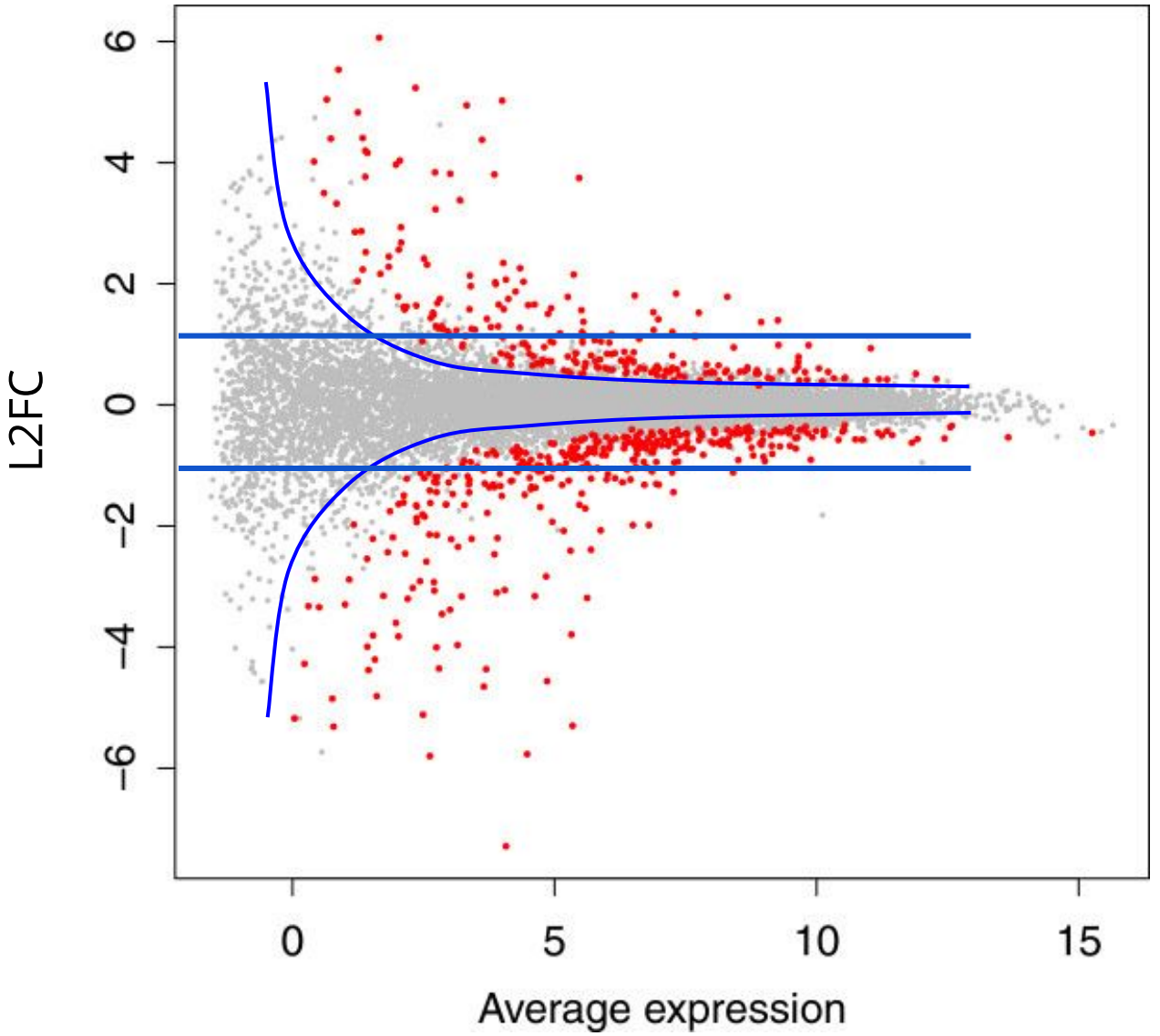
We can choose an arbitrary cutoff, e.g. 1

Lowly-expressed genes usually display higher variability in expression

This can be seen in a **MA-plot**

Can be used as a sanity-check

Or to choose dynamic L2FC cutoffs



Correcting for Multiple Testing

Recall the meaning of a p-value…

Since we are performing multiple tests, we must correct (adjust) p-values

The simple and stringent way - Bonferroni correction

$$p_{adj} = p \times [\#\ of\ genes]$$

The common way - False Discovery Rate (FDR - BH procedure):

1. Order p-values from smallest to largest

   $p_1, p_2, \ldots, p_k, \ldots p_m$

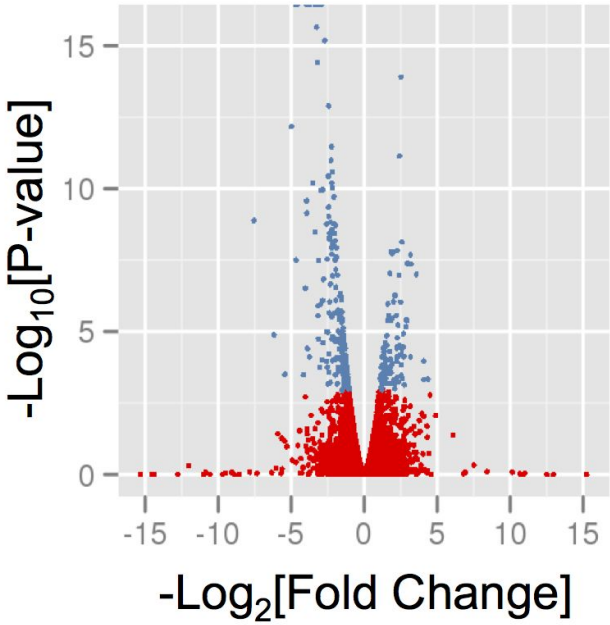
1. $$p_{adj}^{k} = \frac{p^k \times [\#\ of\ genes]}{k}$$

Himes et al. - DGE Analysis Results

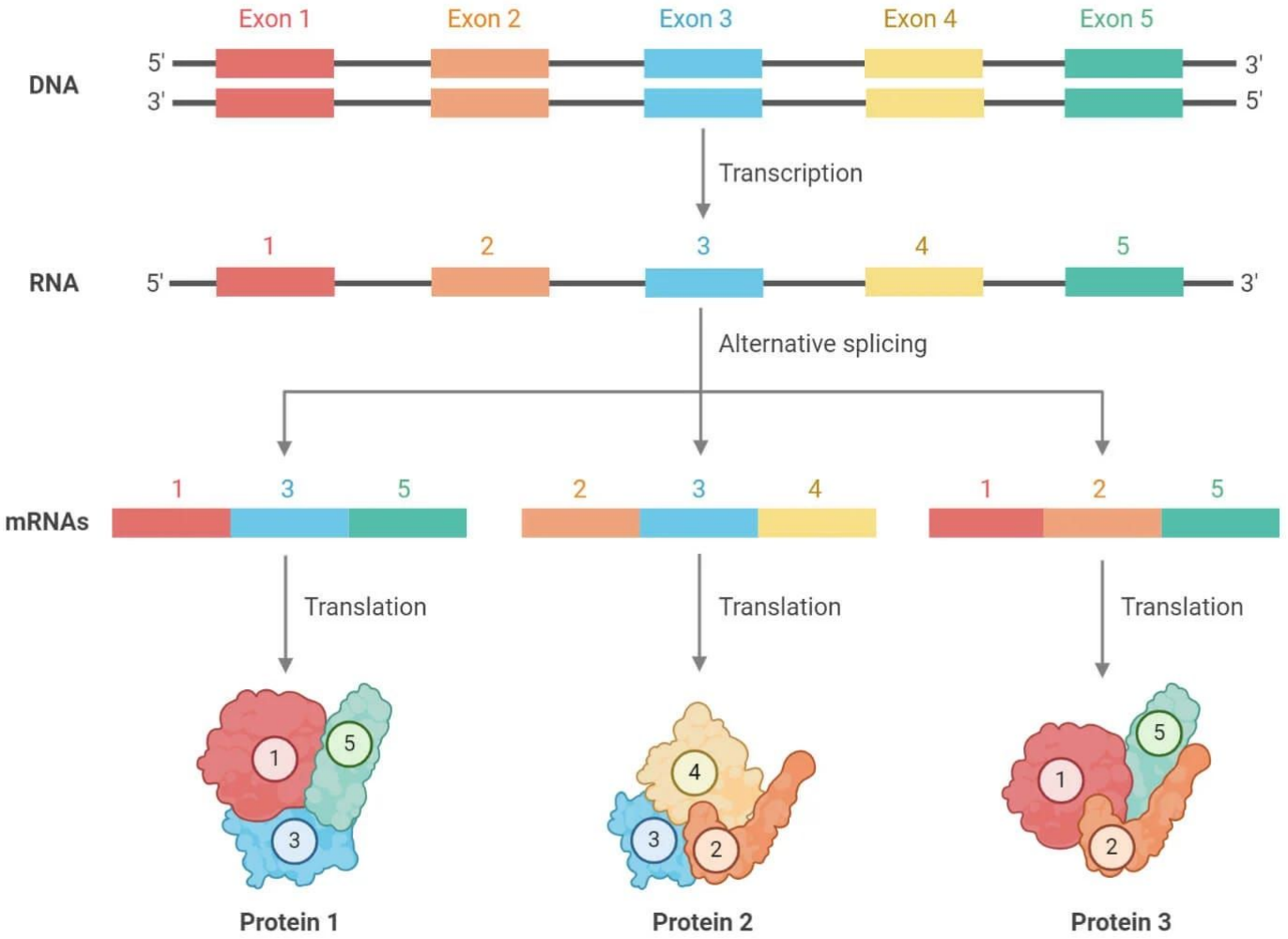
A total of 316 DE genes between treated and untreated samples

Top 5:

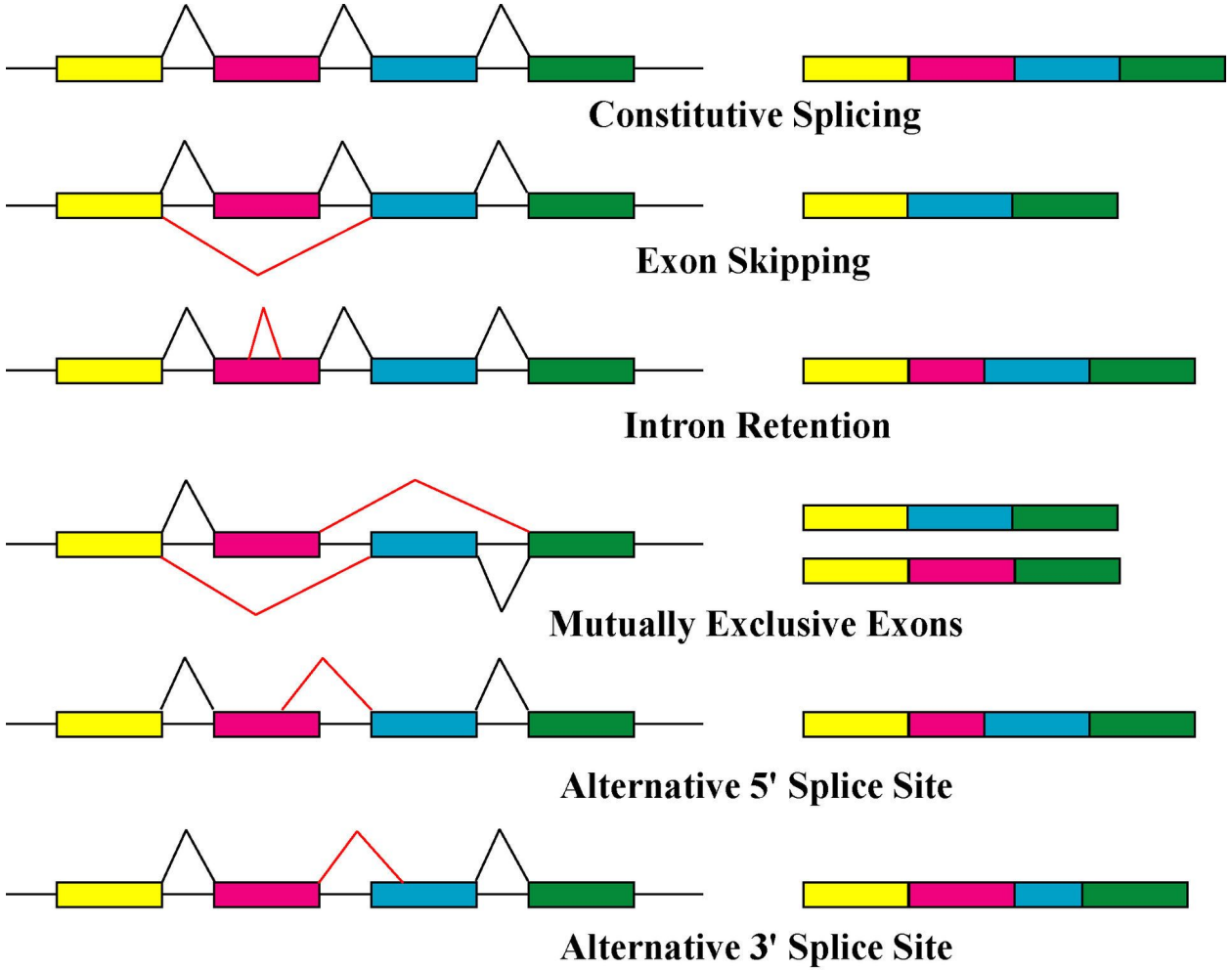
Gene	Dex RPKM	Untreated RPKM	Ln[Fold Change]	Test Statistic	Adj. P-Value
C7	38.41	3.76	-3.35	8.74	0
CCDC69	47.39	6.24	-2.92	8.61	0
DUSP1	144.96	18.26	-2.99	8.99	0
FKBP5	53.05	3.43	-3.95	10.52	0
GPX3	613.37	45.18	-3.76	9.19	0



RNA-Seq and Long Reads



RNA-Seq and Long Reads



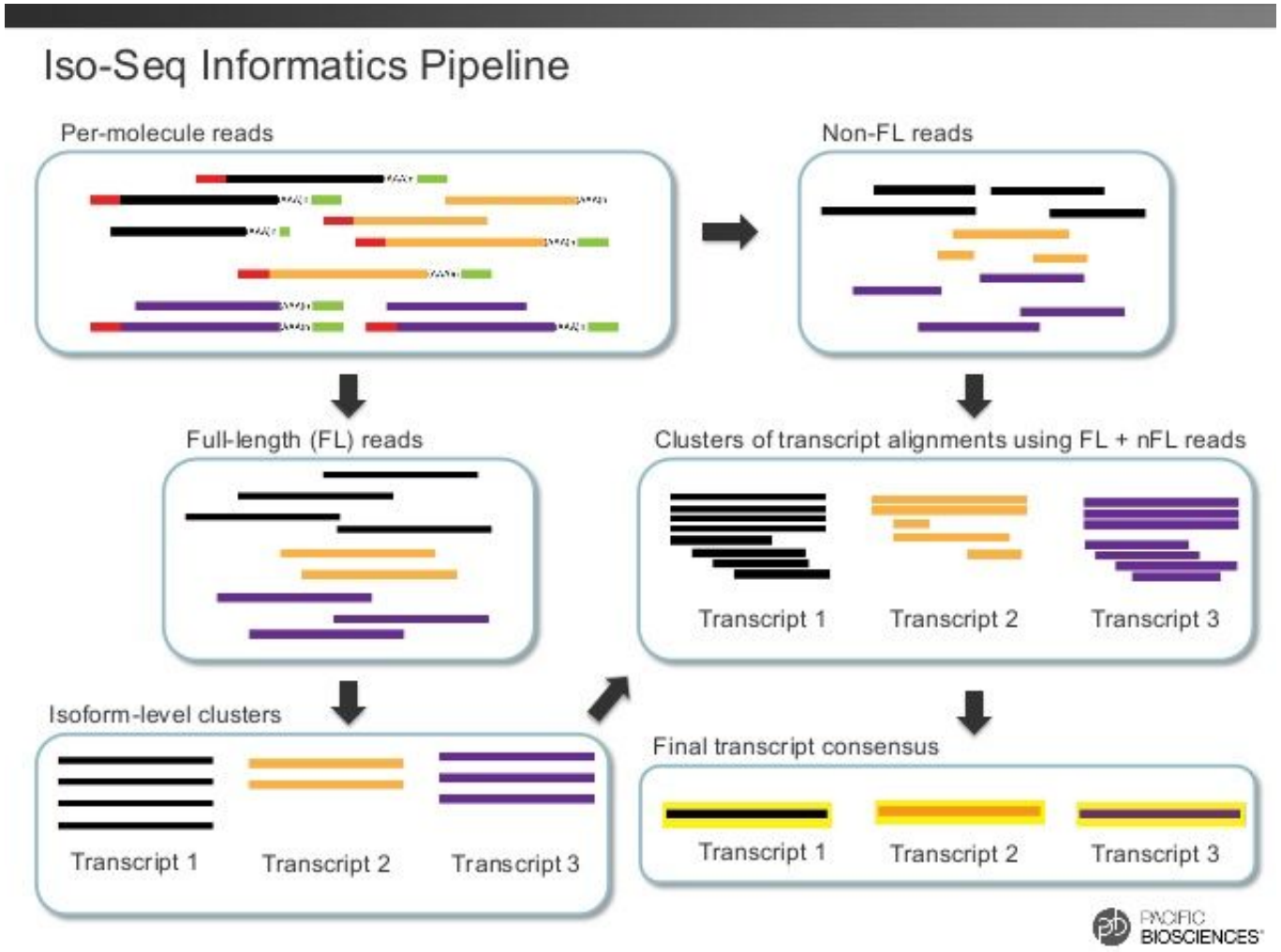
RNA-Seq and Long Reads

Read length is usually larger than mRNA size

Full-length transcripts

No transcript assembly is needed

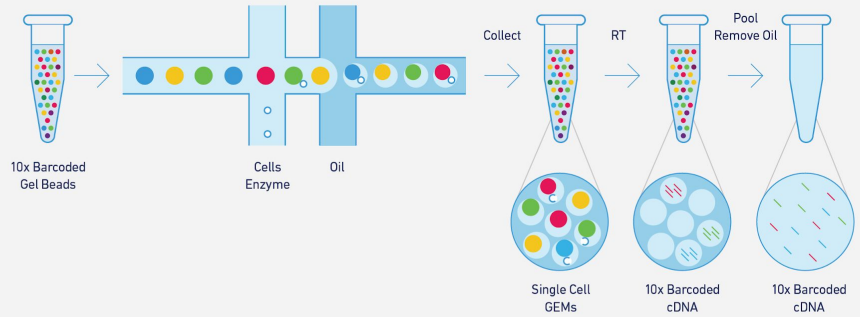
Easier to detect and quantify isoforms



10X for Single Cell RNA-Seq

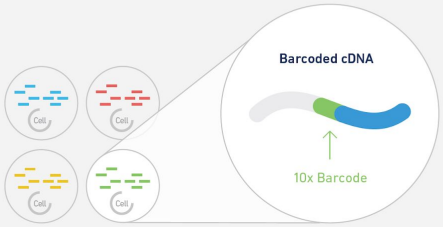
GemCode™ Technology for Single Cell Partitioning

Utilize an efficient droplet-based system to encapsulate up to 100-80,000+ cells in a single 10-minute run.



Single Cell Digital Gene Expression

Enable digital quantification of transcripts in every cell, for single cell digital gene expression analysis.



mRNA is transcribed by a reverse transcriptase that creates barcoded cDNA.

A barcode identifies transcripts originating from a single cell, which are then counted.