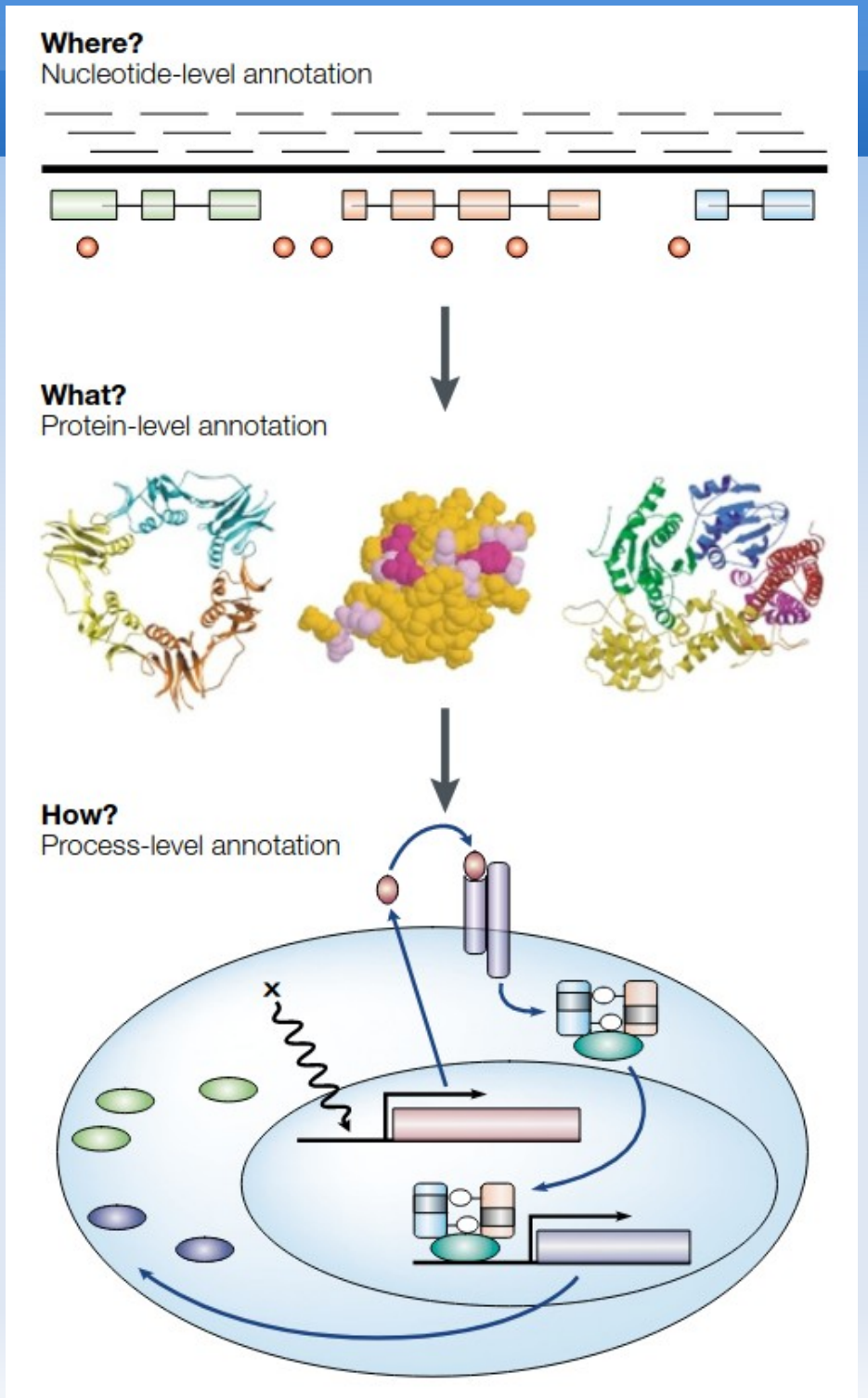
Topic 9.2: Genomes

Sequence Mapping, Assembly, and Annotation



Genome Sequencing

In theory, we could take a single cell and sequence every chromosome end to end. Not really done because:

- takes too long for large chromosomes
- things can go wrong:
  - DNA can break
  - Sequencers make mistakes – sequencing errors

Instead, what works now is:

- many cells, extract the DNA for that individual
- we break the DNA into pieces (called DNA ”fragments”)
- sequencers are able to ”read” small (e.g. 300 bp) fragments reliably
- we get a few 100s of millions of reads from our sequencer.

Mapping

If we already have a sample genome, e.g. the ”human” genome, we can map sequence reads for an individual to a ”reference genome”

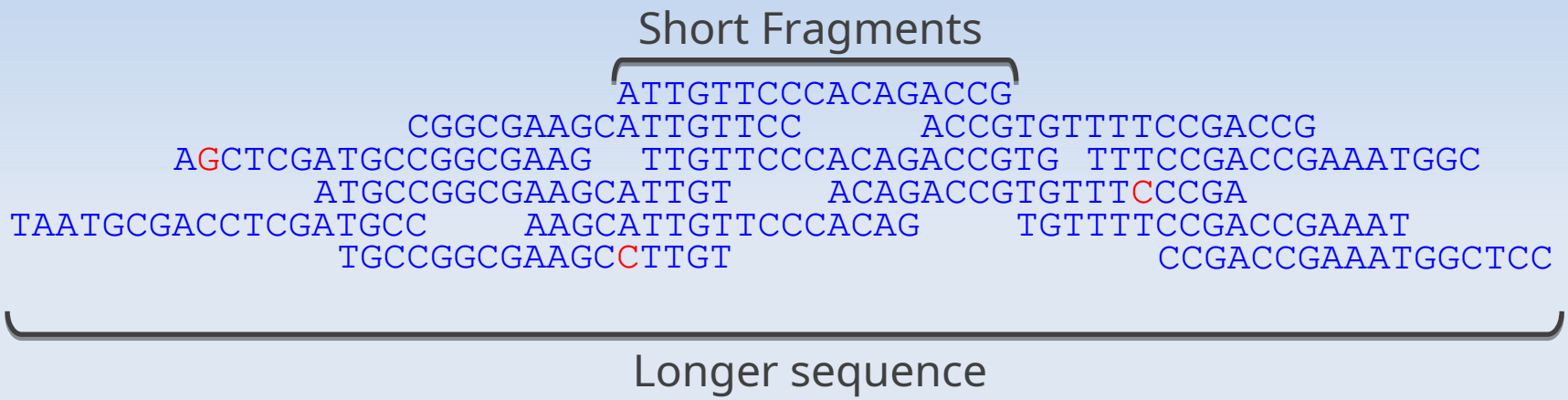
Where does a ”read” go on a 3 billion base-pair genome? This is the ”Mapping” problem: align a sequence read to a known reference genome

A "map" is the best location in the reference sequence for a new read

The ”mapping” problem: find the best alignment in the reference genome for each read

Genome Assembly

The process of assembling fragments into a long sequence



Consensus:

TAATGCGACCTCGATGCCGGCGAAGCATTGTTCCCACAGACCGTGTTTTCCGACCGAAATGGCTCC

Coverage: # of reads underlying the consensus

Average coverage: Total bases / consensus length

13 reads * 17 bases per read = **221 bases**

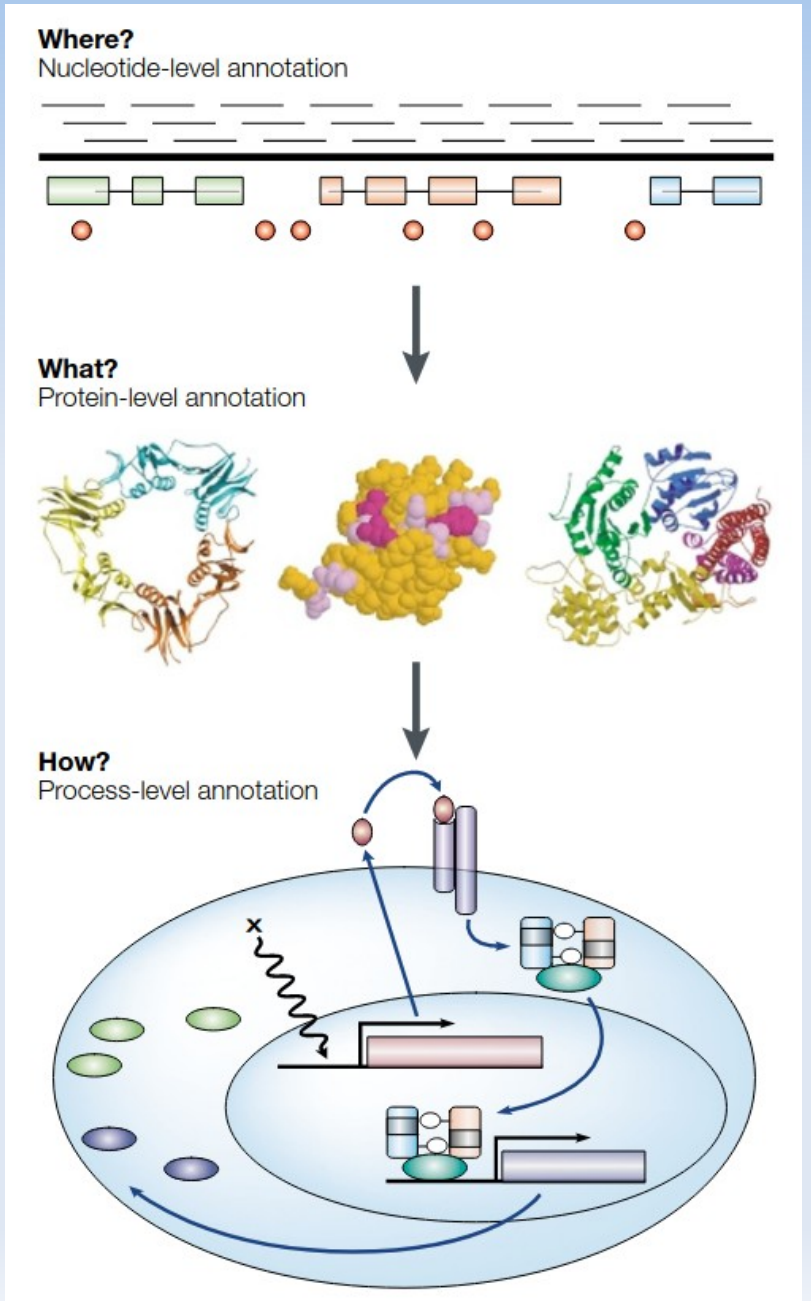
221 bases / 66 base consensus = **3.35-fold coverage**

Genome Annotation

The process of describing what it all means.

Annotation intersects with gene ontology – a unified study of gene attributes across species

→ can infer biology



Mapping

Already have a genome

Got new sequence information: reads

”Mapping”: for each read, find the best alignment

Mapping steps

1) Run a sample through a sequencer, get back 1000s to 1000,000,000s of sequences
2) Get rid of ”low-quality” sequences like

   sequencing problems – low Q scores, ambiguous reads, etc.

   adapter sequences, not part of organism
3) Map remaining data to a reference – e.g. Smith-Waterman local alignment works
4) Analyze results

FASTQ file format

Reads are stored as FASTQ files

FASTQ is a text format for storing sequence information and associated quality scores

Example:

@ followed by a sequence identifier and optional description

Raw sequence - letters

```
@SEQ_ID
GATTTGGGGTTCAAAGCAGTATCGATCAAATAGTAAATCCATTTGTTCAACTCACAGTTT
+
!''*((((***+))%%%++)(%%%%).1***-+*''))**55CCF>>>>>>CCCCCCC65
```

+ followed by an optional sequence identifier and description

Quality scores, one for each seq letter from 0x21 or ! in ASCII … 0x7e or ~ in ASCII
Quality values (low to high) are:

```
!"#$%&'()*+,-./0123456789:;<=>?@
ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`
abcdefghijklmnopqrstuvwxyz{|}~
```

Strategies for Mapping

Traditional Mapping

- Burrows Wheeler Transform – BWA, Bowtie
- Hashing

Split-read Mapping – pindel, scissors

Graph Alignment – glia

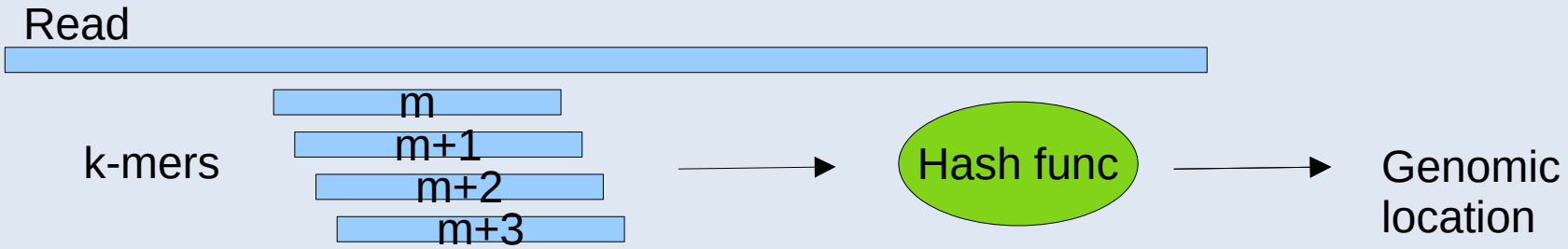
Hashing approach to Mapping

1) Find all k-mers in the reference genome

2) Store the genomic locations in a hash table

A hash table (dictionary) of locations is a fast way to search the entire reference genome!

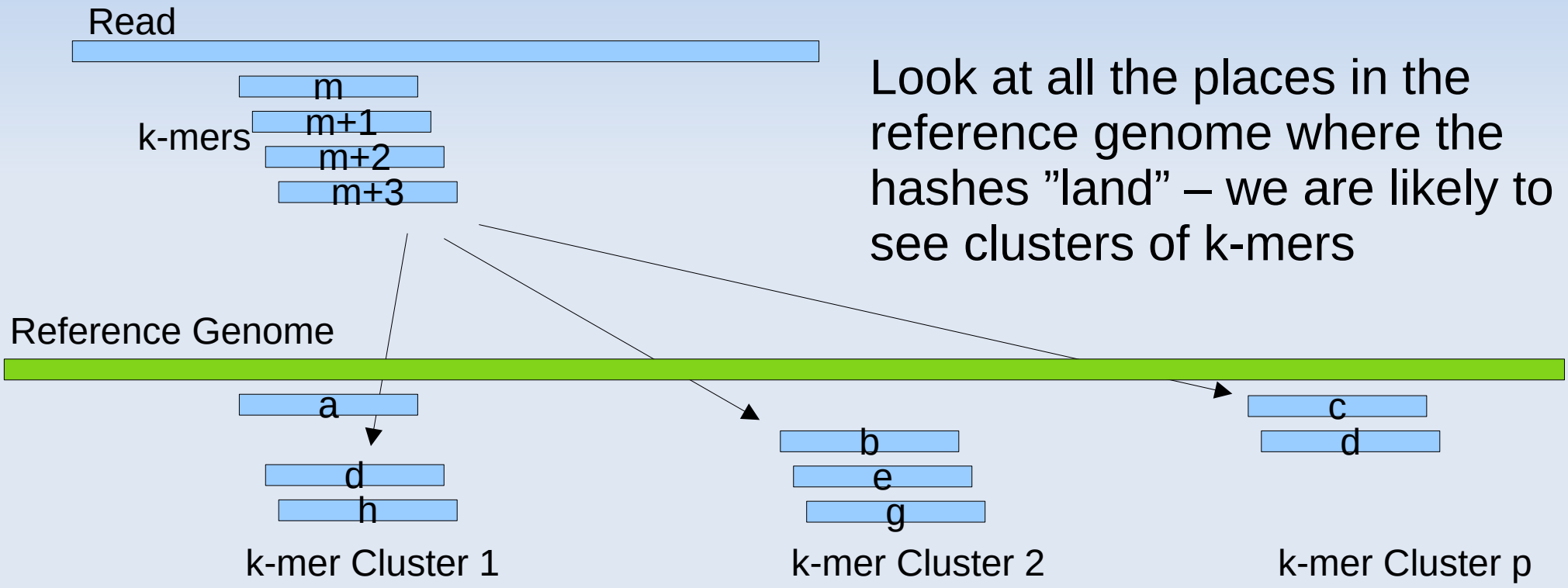
3) For each read:



generate all its k-mers and use the hash table to find matching genome location → find where the read goes in the genome!

Hashing approach to Mapping

Need to take into account: possible sequencing errors, true differences between the reference genome and the genome being sequenced.

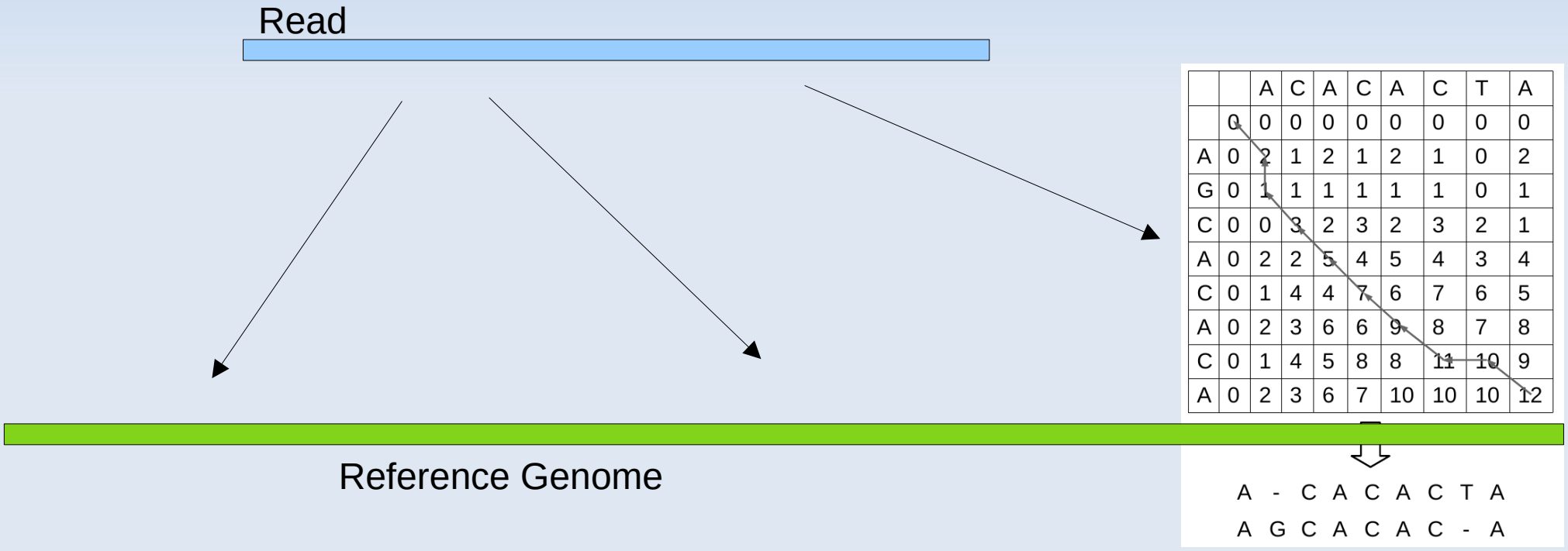


Clusters of k-mers are likely to occur all over the genome. Smaller (e.g. 1 or 2 k-mer) clusters are likely to be ”noise” and, therefore, not candidates for aligning reads.

Mapping: Aligning reads

How can we test the alignment of a read to potential genomic locations?

Local alignment, BLAST, Smith-Waterman will work!



		A	C	A	C	A	C	T	A
	0	0	0	0	0	0	0	0	0
A	0	2	1	2	1	2	1	0	2
G	0	1	1	1	1	1	1	0	1
C	0	0	3	2	3	2	3	2	1
A	0	2	2	5	4	5	4	3	4
C	0	1	4	4	7	6	7	6	5
A	0	2	3	6	6	9	8	7	8
C	0	1	4	5	8	8	11	10	9
A	0	2	3	6	7	10	10	10	12

Mapping data

Sequences aligned to a reference (genome) sequence can be stored using:

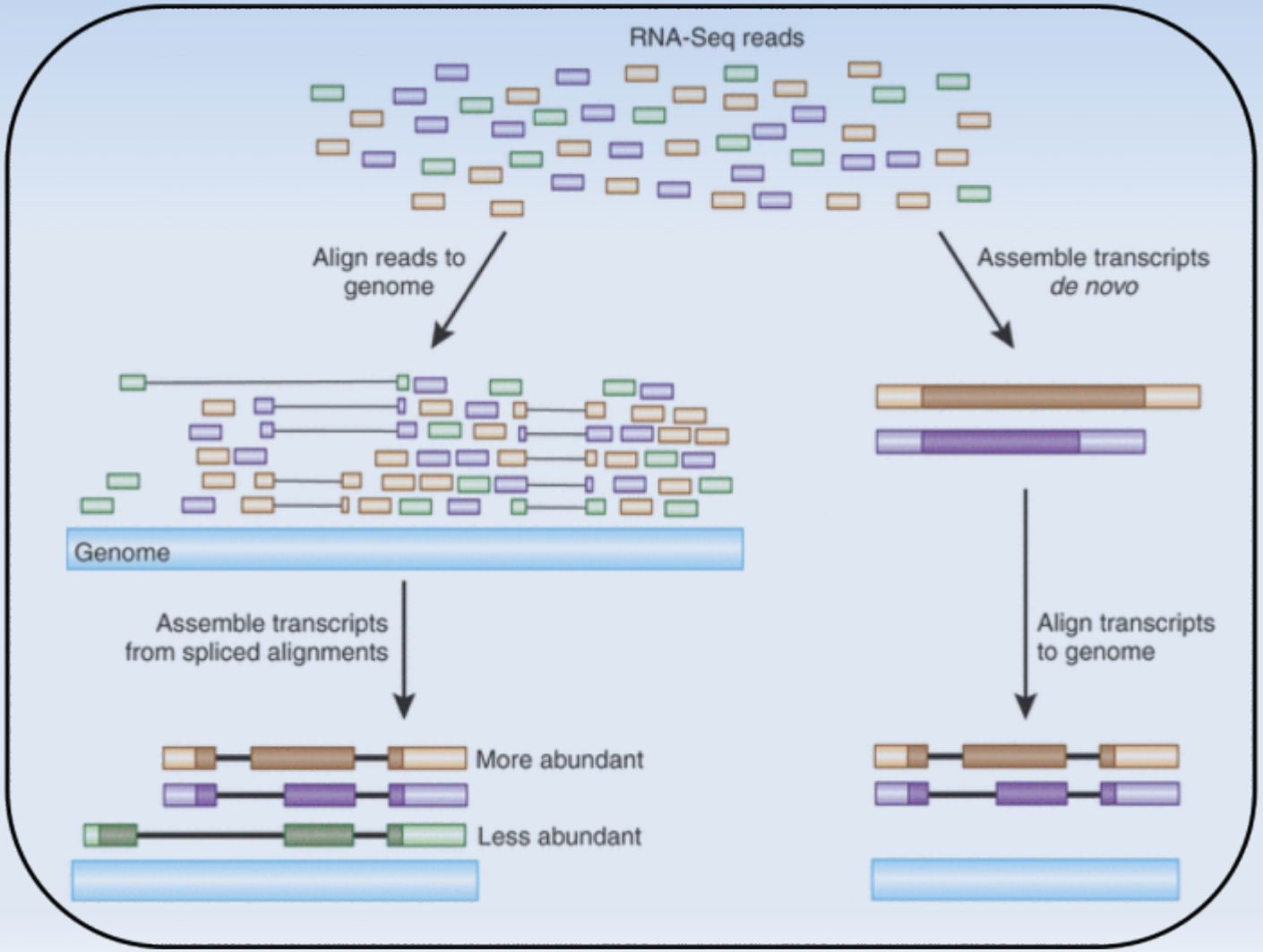
- MAQ map
- Sequence Alignment Map (SAM)
- Binary Alignment Map (BAM)
- Compressed Reference-oriented Alignment Map (CRAM)

These file formats can easily go into the 10s of GB.

Author Carl Zimmer's genome was a 70 GB BAM file

Mapping: mRNA reads to genome

We can sequence cDNA obtained from mRNA (RNA-Seq) and align these reads to genomes as well:



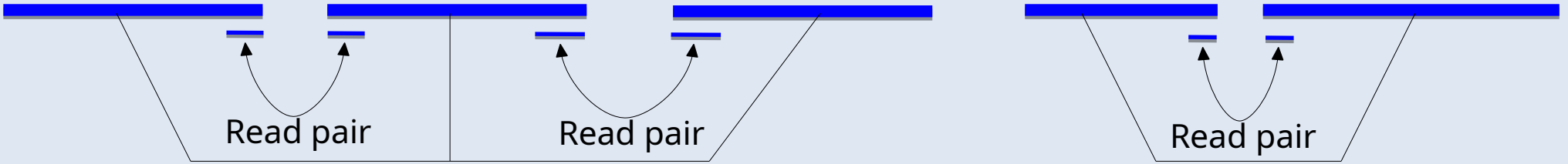
Potential to get information about splice variants

Assembly

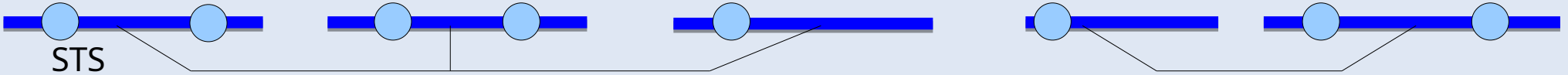
Many cells → many copies of (polyploid) genome → mechanical/enzymatic fragments → sequence reads

Sequence Contigs - contiguous segments completely covered by reads

Scaffolds: like contigs but with missing parts – use read pairs and PCR + sequencing to fill in



Mapped Scaffolds



Genome Map

Assembly

A pair of reads may have mismatched bases

Mismatch may be due to:

- a sequencing error in one read

- if diploid, a difference between alleles in inherited copies

We can assemble a diploid genome but that will require a different assembly technique

Usually, ploidy is ignored. Human reference genome is haploid

Assembly as a CS problem

Shortest common supersequence problem

```
TAATGCGACCTCGATGCCGGCGAAGCATTGTTCCCACAGACCGTGTTTTCCGACCGAAATGGCTCC

                          ATTGTTCCCACAGACCG
                 CGGCGAAGCATTGTTCC     ACCGTGTTTTCCGACCG
       AGCTCGATGCCGGCGAAG  TTGTTCCCACAGACCGTG TTTCCGACCGAAATGGC
             ATGCCGGCGAAGCATTGT    ACAGACCGTGTTTCCCGA
TAATGCGACCTCGATGCC    AAGCATTGTTCCCACAG    TGTTTTCCGACCGAAAT
               TGCCGGCGAAGCCTTGT                    CCGACCGAAATGGCTCC
```

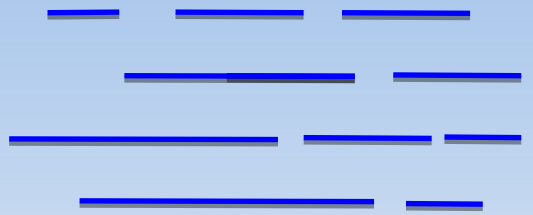
We have a number of sequences (reads) and we want to find the shortest string (assembled genome) that has all the sequences as subsequences.

”NP-hard” → expensive to compute.

Can convert a SSP to a Traveling Salesperson Problem!

Two approaches to assembly

Sequence Reads



String Graph Assemby uses
Overlap→ Layout→Consensus.
Used for Celera human genome

Good for long reads

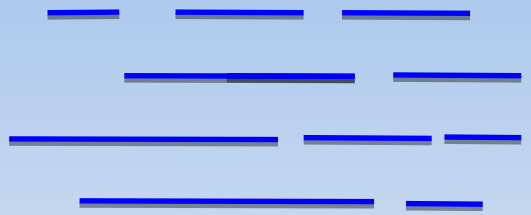
Eulerian path
de Bruijn Graph Assemby:
Construct k-mer graph from reads

Trace path in graph to assemble

Contig

String Graph Assembly

Sequence Reads



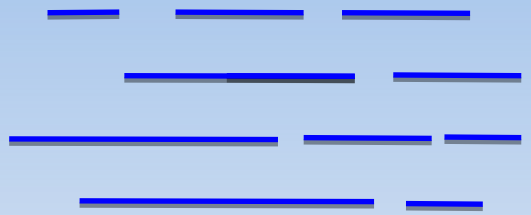
String Graph Assemby uses
Overlap→ Layout→Consensus
good for long reads

1) Compare each read vs. all other reads; find ~50 bp overlaps
2) Build Overlap Graph
3) Bundle stretches of the overlap into contigs
4) Pick most likely genome sequence

Contig

String Graph Assembly: Find overlaps

Sequence Reads



For each pair of reads: R1: GGCCTAGAG
R2: CTAGAGCCT

1) take the last, say, 3 nucleotides – GAG of R1
2) Look for it in R2

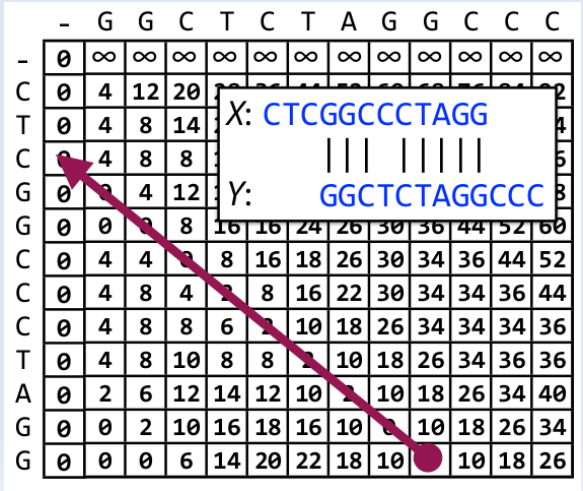
R1: GGCCTA**GAG**
R2: CTA**GAG**CCT

Found it!

3) Extend to the left

R1: GGC**CTAGAG**
R2: **CTAGAG**CCT

To allow mismatches and gaps, use Dynamic Programming!



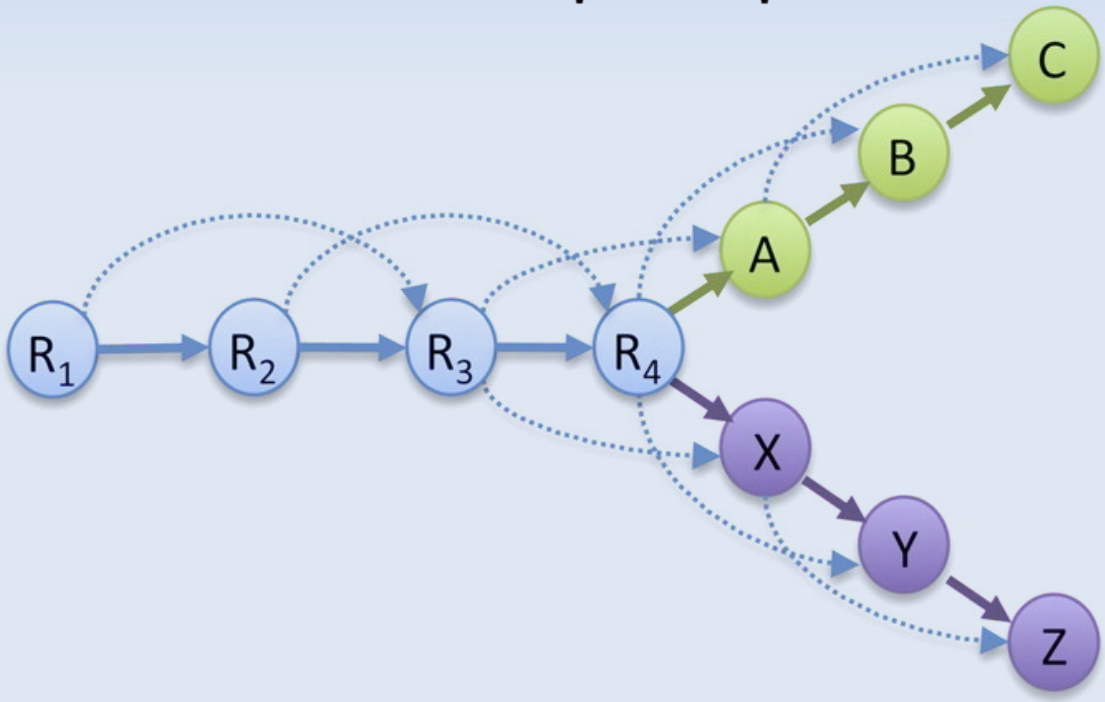
String Graph Assembly: Overlap Graph

An Overlap Graph links the various read fragments

A Read Layout

```
R1: GACCTACA
R2:    ACCTACAA
R3:      CCTACAAG
R4:       CTACAAGT
A:          TACAAGTT
B:           ACAAGTTA
C:            CAAGTTAG
X:          TACAAGTC
Y:           ACAAGTCC
Z:            CAAGTCCG
```

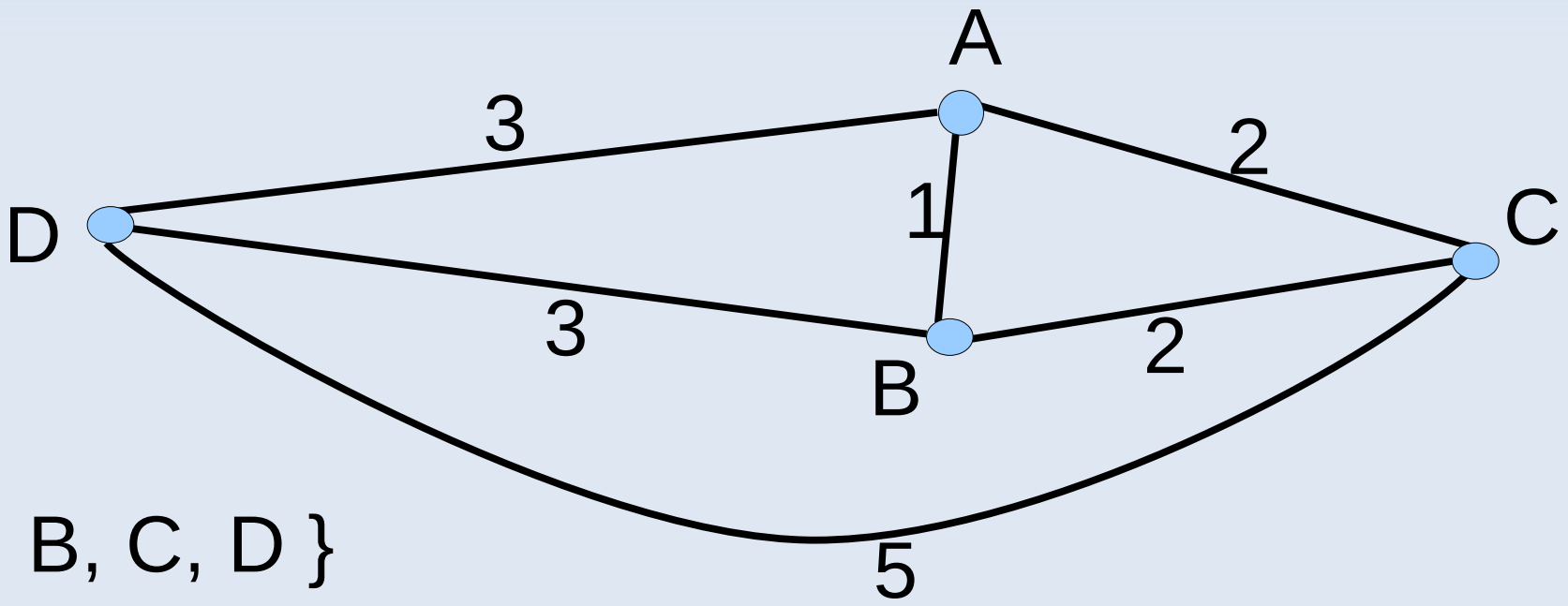
B Overlap Graph



Two reads with significant overlap share an edge

Graphs

A graph is a set of nodes and edges (between nodes)



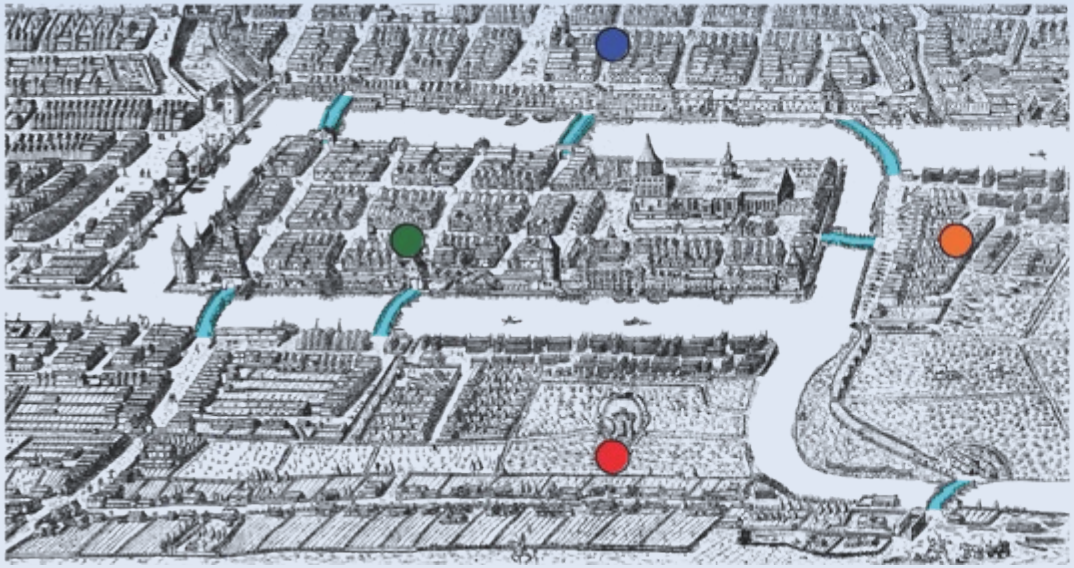
N = { A, B, C, D }

E = {(A,B), (B,C), (A,C), (A,D), (B,D), (D,C)}

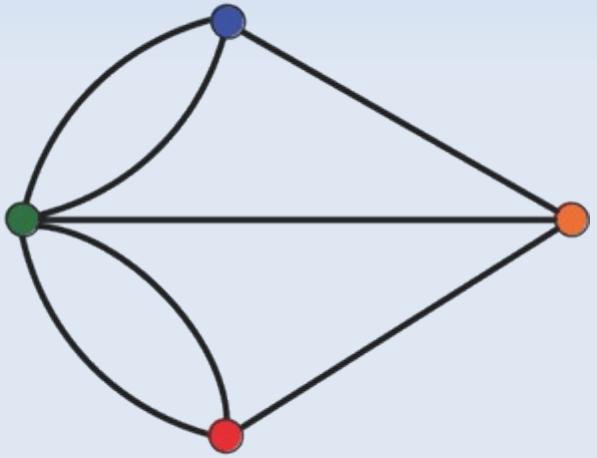
Seven Bridges of Königsberg

City on two sides of a river with two islands, 7 connecting 7 bridges. Challenge: tour the city crossing each bridge just once.

a



b

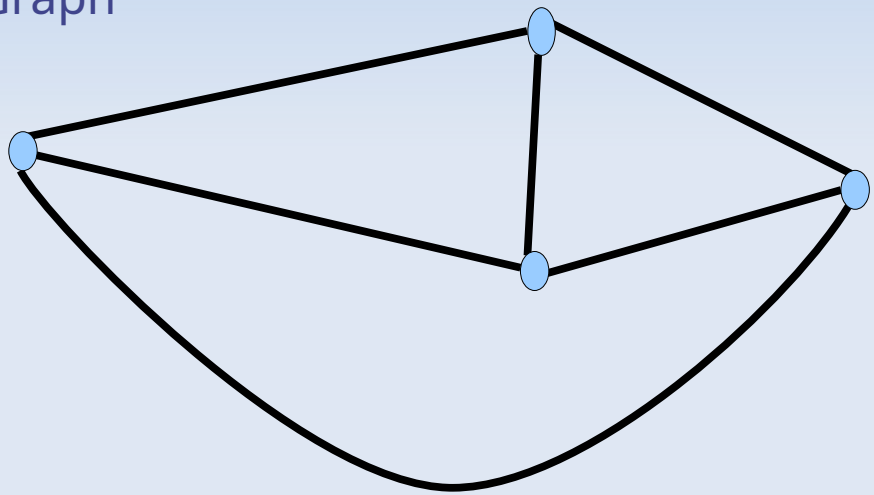


Resolved by the mathematician Euler who noted: nodes have odd # of edges → kicked off the area of Graph Theory

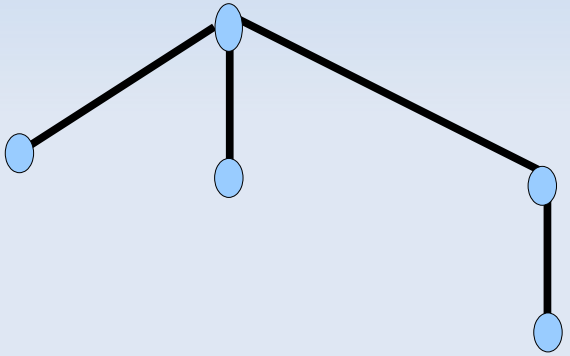
Graphs vs. Trees

Trees are special cases of Graphs

Graph



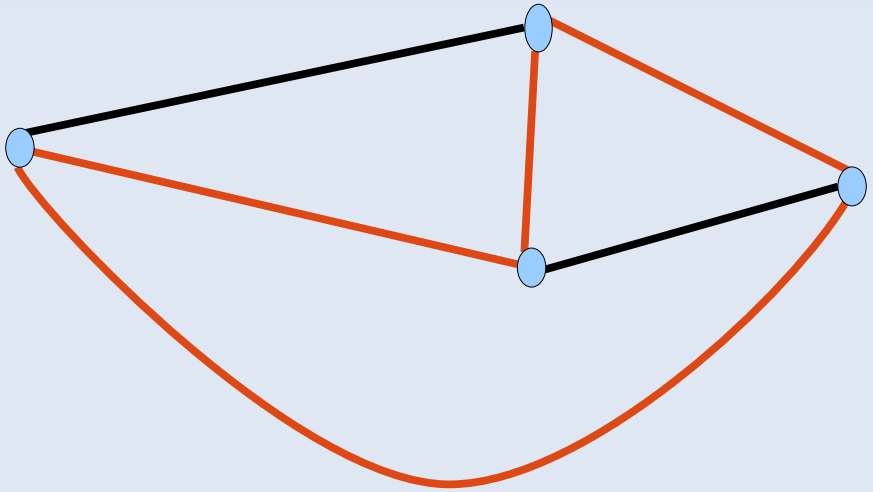
Tree



Key difference: No hierarchy in graphs

Hamiltonian Cycle

A Hamiltonian cycle of a graph is a loop that visits each node just once and returns to the start.



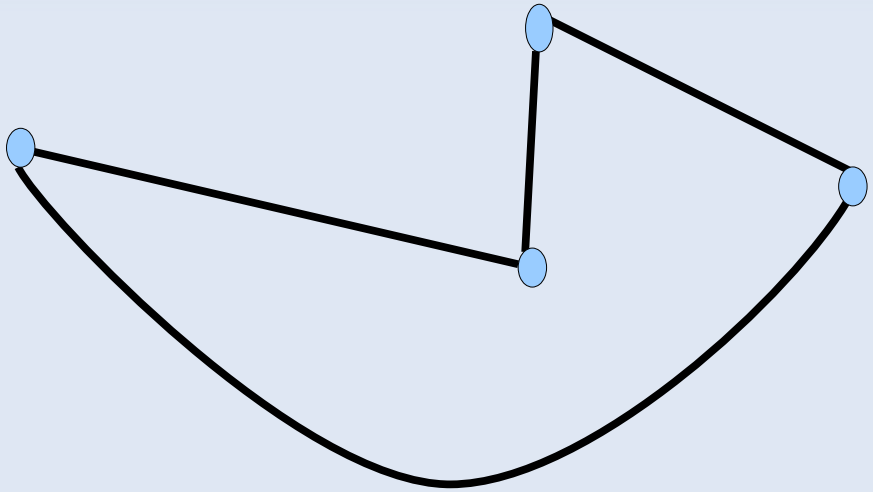
Hamiltonian cycle of Graph

NP-complete!

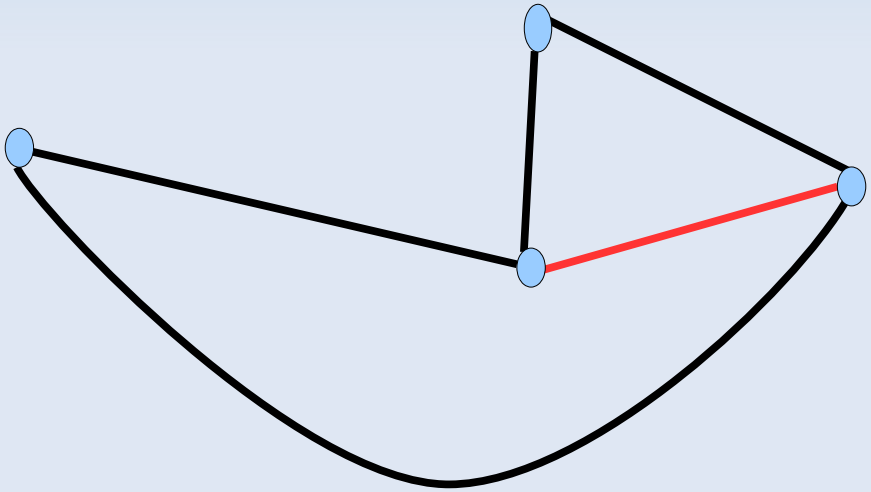
Traveling Salesman: find the best Hamiltonian cycle.

Eulerian Cycle

A Eulerian cycle of a graph is a loop that visits each edge just once.



Eulerian cycle of Graph



Non-Eulerian Graph
No Eulerian cycle

Seven Bridges of Konigsberg → Is there a Eulerian cycle? Euler proved: No!

Consensus

Finding a consensus:

6x coverage
100% identity

Coverage: # of reads underlying the consensus

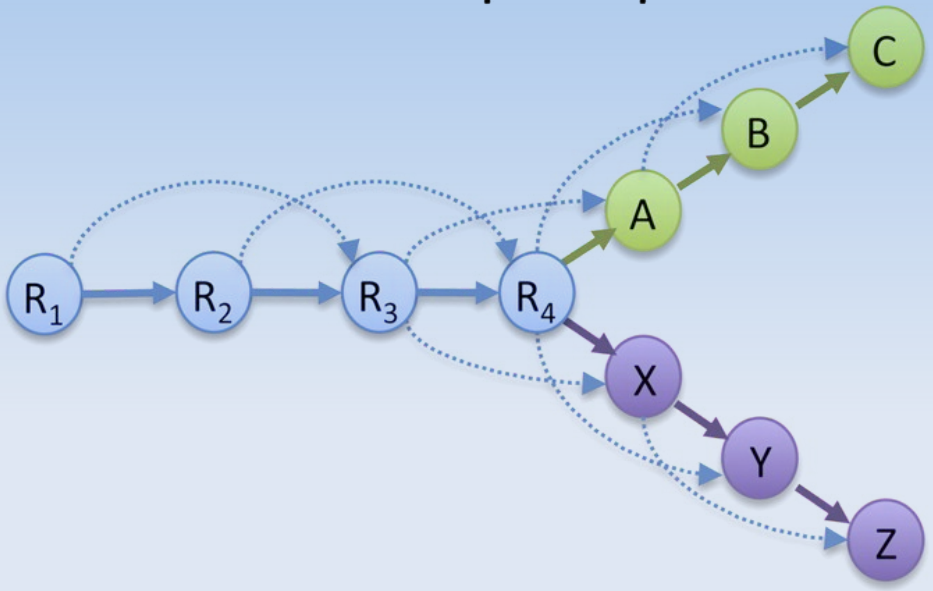
de Bruijn graph

Read overlaps

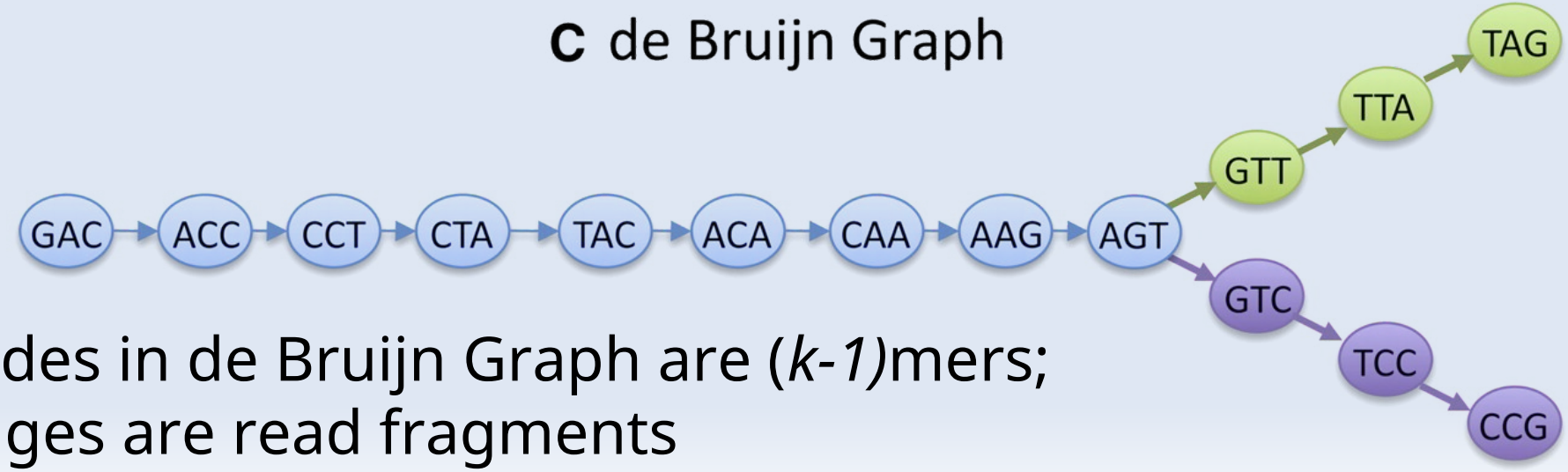
A Read Layout

```
R1: GACCTACA
R2:   ACCTACAA
R3:    CCTACAAG
R4:     CTACAAGT
A:       TACAAGTT
B:        ACAAGTTA
C:         CAAGTTAG
X:       TACAAGTC
Y:        ACAAGTCC
Z:         CAAGTCCG
```

B Overlap Graph



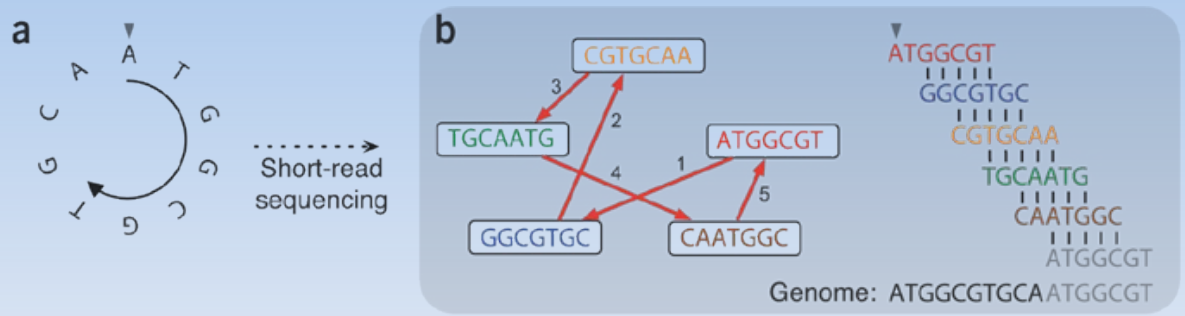
C de Bruijn Graph



Nodes in de Bruijn Graph are (*k-1)*mers; edges are read fragments

Graphs and Genome Assembly

(a) An example small circular genome.

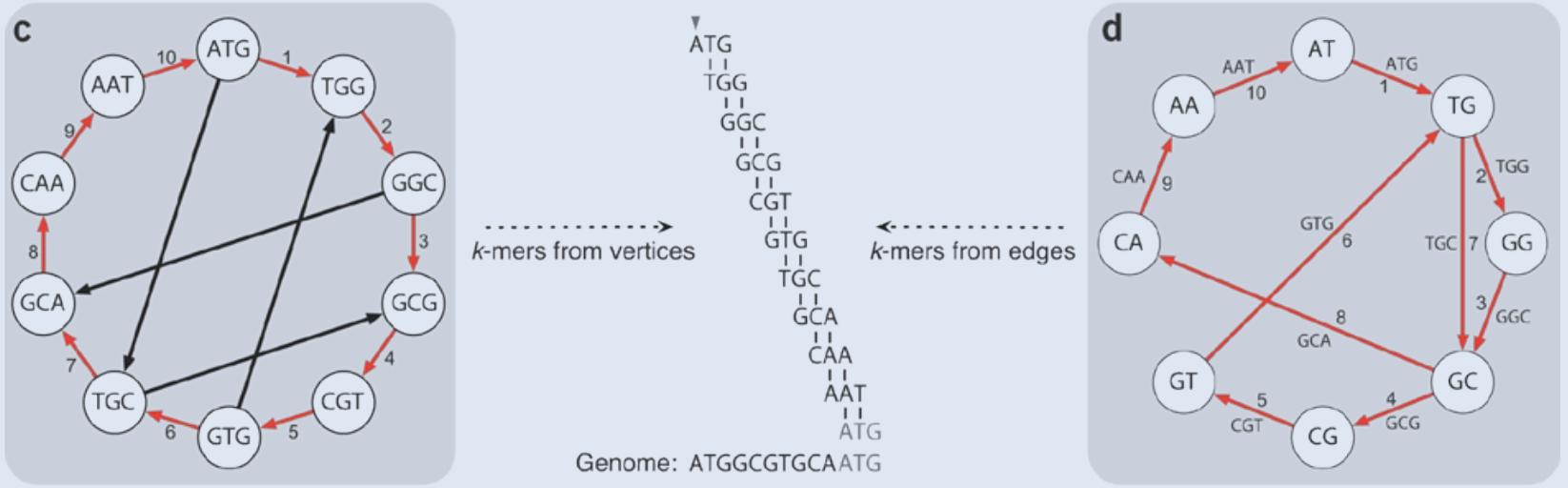


(b) Traditional sequencing algorithms: reads are nodes in a graph, edges are alignments between reads.
Genome assembly is like Traveling Salesperson (NP-hard) problem.
A Hamiltonian cycle is easier and reconstructs the genome.

Vertices are *k*-mers
Edges are pairwise alignments

Vertices are (*k*–1)-mers
Edges are *k*-mers

Typical assemblers use k-mers of sizes 60-70



Hamiltonian cycle
Visit each vertex once
(harder to solve)

Eulerian cycle
Visit each edge once
(easier to solve)

(c) Split reads into k-mers: successive k-mers shifted by one position. Hamiltonian cycle (red) reconstructs the genome. NP-complete.

(d) Modern short-read assembly algorithms construct a de Bruijn graph of (k-1)-mers. An **Eulerian** cycle reconstructs the genome sequence.

Data in Genome Browsers

Along with genomic sequence information, genome browsers usually also have ”tracks” like:

- Mapping and Sequencing
- Genes (RefSeq, Ensembl) gene (mRNA and protein) and Gene prediction
- Phenotype and Literature
- mRNA and Expressed sequence tags
- GENCODE track with non-coding functional elements as found by projects like ENCoDE
- Single Nucleotide Polymorphism track

GENCODE Project

The GENCODE project aims to identify all gene features in human/mouse genomes.

One goal is to include all alternative splice variants:
- protein coding loci
- noncoding loci
- pseudogenes

using computational methods, experimental verification and manual curation.
Annotation pipeline aligns RNA-seq, cDNA, and protein data to genomes.

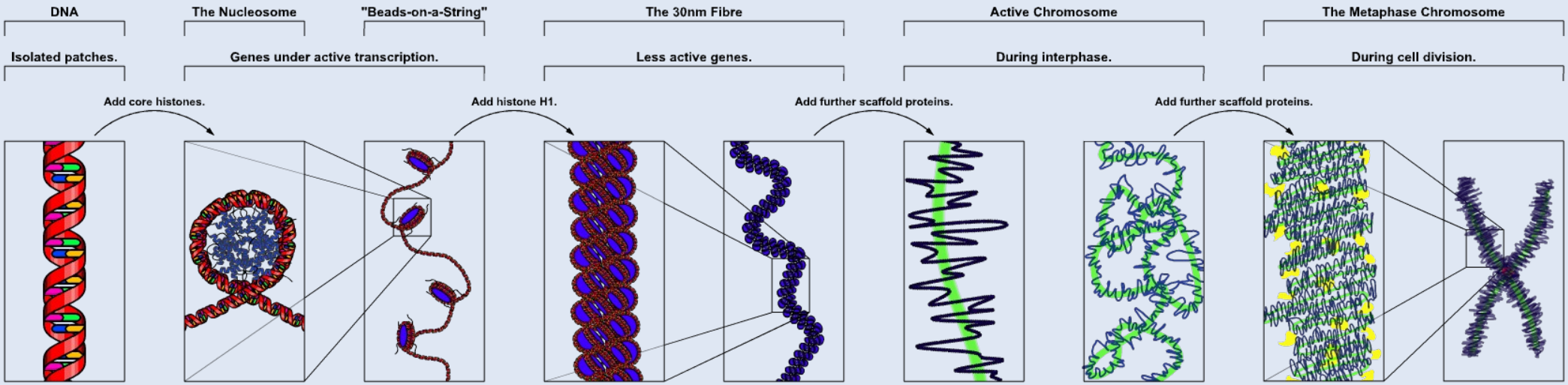
Started as an addition to the **ENC**yclopedia **o**f **D**NA **E**lements project - see the Encode portal.

ENCoDE Project Phase II

ENCyclopedia **o**f **D**NA **E**lements: Identify all functional elements in human genome

Up to version 5 now:

- 80% of human genome participates in biochemical RNA or chromatin (30 nm) process



- 95% of genome lies within 8 kb of a DNA-protein interaction → challenges idea of "junk" DNA.

Genome Annotation

- Uses software to locate and identify all the protein coding, rRNA, and tRNA genes within a genomic sequence
- An area of Genome Annotation: ”gene prediction”

  Goal is to find patterns that indicate areas of the genome that may represent genes.
- Prokaryotic genomes are simple
- Our knowledge of sequence signals in eukaryotic genomic DNA is limited: current gene prediction methods are approximate.

See Genomic and transcriptomic annotations in ENCODE v. 5

Gene Prediction

- Remember from Gene Sequences lecture (Lecture 2): a dsDNA may have 6 reading frames – 6 ways to divide it up into codon triplets
- Prokaryotic genomic DNA sequence: one of the 6 is considered the "open reading frame": mRNA from transcription will be translated in only one way from the start (AUG) to a stop codon (UGA, UAG, UAA). Can use genomic DNA to predict ORFs.
- Eukaryotic genomes have exons and introns. mRNA with introns spliced out is the basis for finding the open reading frame → Cannot use genomic DNA to find ORF.
  Need to know splice sites in DNA to predict ORFs

Gene Prediction

Two main approaches:

- Intrinsic or ab initio gene prediction

  Searching "by signal" or "by content"

- Extrinsic or evidence based gene finders

  Homology-based

  Using BLASTX

  Using experimental mRNA and RNA-Seq data

Ab Initio Prokayotic gene prediction

Typically, a prokaryotic gene starts with an ATG (translation initiation site, TIS)

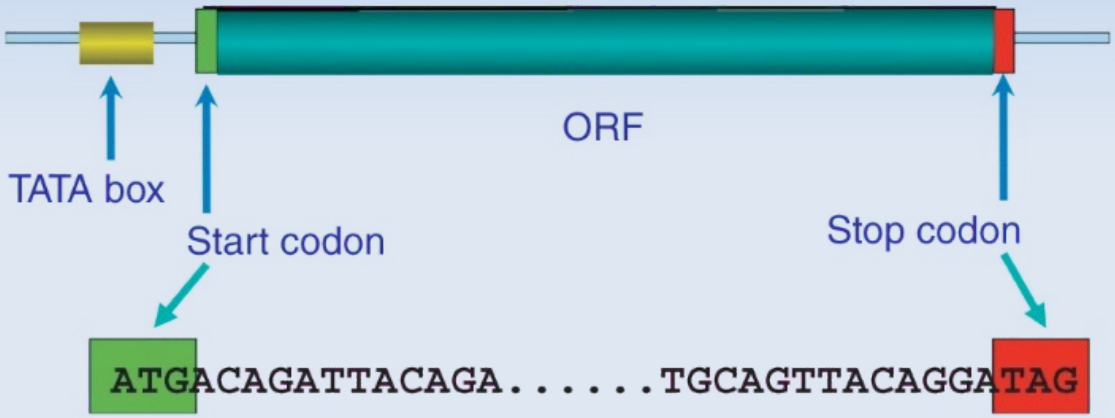


Figure 5.1 A simplified depiction of a prokaryotic gene or open reading frame (ORF) including the start codon (or translation initiation site), the stop codon (TAG), and the TATA or Pribnow box.

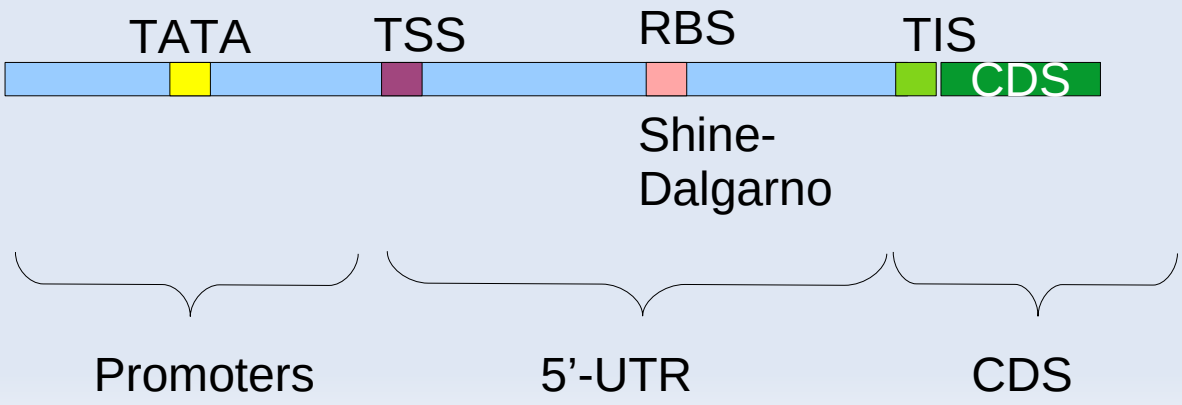
and ends with one of 3 stop codons in the reading frame: TAG, TAA, or TGA

Prokayotic gene 5’-UTR

Upstream of the TIS (5’-UTR):

- 8-10 bases: ribosome binding site (RBS) – AGGAGG Shine-Dalgarno consensus sequence
- Transcription (from DNA to RNA) start site (TSS) - RNA polymerase binds
- 10 bases upstream of TSS: TATA box (archea) or TATAAT Pribnow box (bacteria) consensus sites

In practice, rather than actually searching for consensus sites gene signals are identified using Positional Weight Matrices (PWM) or Position-Specific Scoring Matrices (PSSM)



Consensus sequence

A consensus sequence like Shine-Dalgarno or the Pribnow box is different from things like the ATG start codon sequence or the stop codons

We obtain a consensus sequence using experimental evidence indicating that the seq is a DNA binding site. Often, there is some **variability** in the bases and so the **most probable** bases are listed in the ”consensus sequence”

Slight base variations are tolerated – still work

Conserved sequences are found using multiple sequence alignment and calculating position-specific scoring matrices

Eukaryotic Gene Prediction

- Eukaryotic genes have ~100 times lower protein-coding density

Human genome is:
1.1% exons
24% introns
75% intergenic

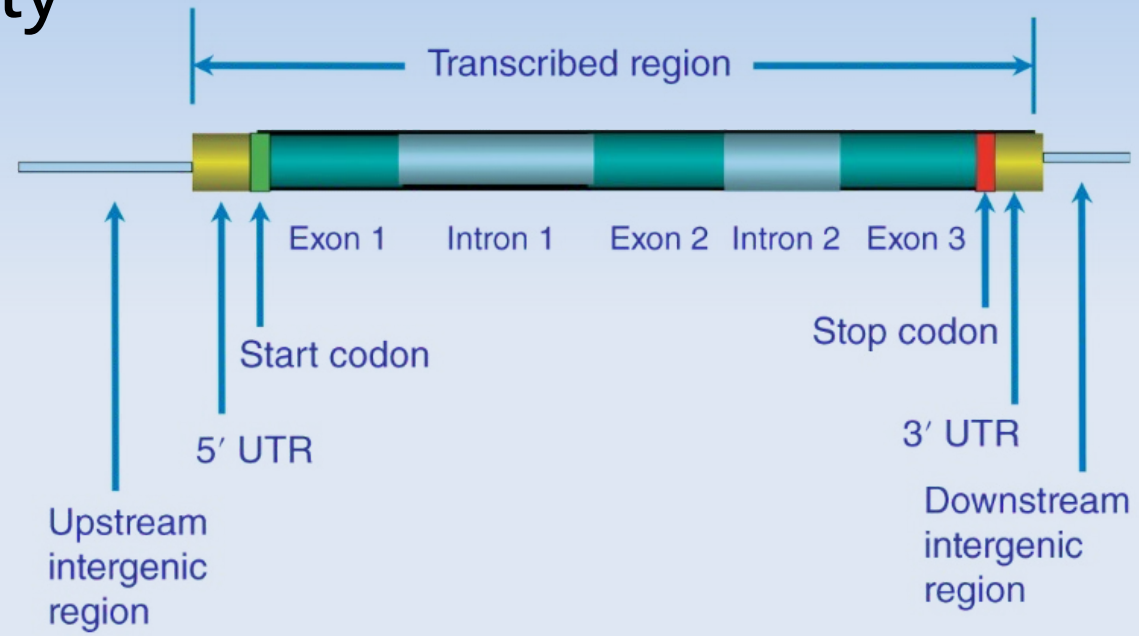


Figure 5.2 A simplified depiction of a eukaryotic gene illustrating the multi-intron/exon structure, the location of the start and stop codons, the untranslated regions (UTRs), and the intergenic regions that surround the transcribed gene.

- Exons & introns within the CDS – junctions (cut sites) are recognized by small ribonuclear proteins – mimicked in computational gene prediction

Predicting exon locations

Some eukaryotic genes have a promoter region called a GC box – GGGCGG consensus sequence – binding site for Zinc finger proteins

CAT box signals binding site for RNA transcription factor – GGCCAATCT consensus sequence

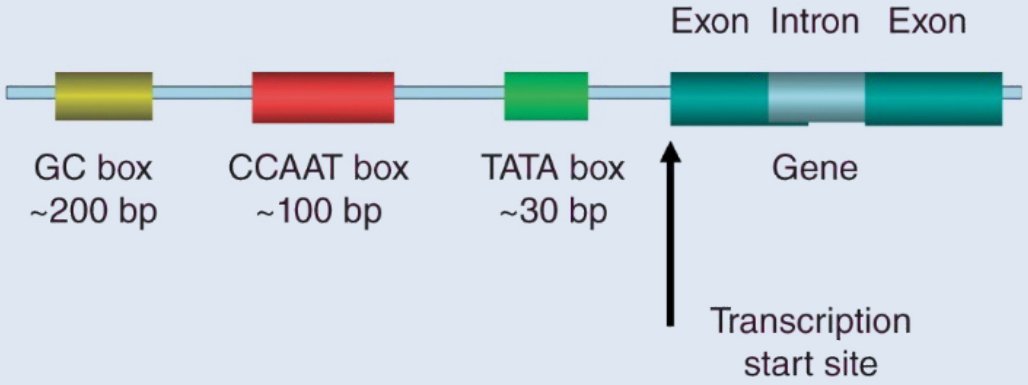


Figure 5.3 A schematic illustration of the upstream regions of a eukaryotic gene with the GC box located ⊠ 200 bp upstream, the CCAAT box located ⊠ 100 bp upstream, and the TATA box located ⊠ 30 bp upstream of the transcription start site.

Best studied eukaryotic promoter is TATA box found in ~30% of genes. TATA(A/T)(A/T) consensus site

Predicting exon locations

4 DNA signals define exons:

- TIS – Kozak consensus site CC*R*CC**ATG**G
- 5’ donor RNA splice site: GG-cut-GT-intron
- 3’ acceptor splice site: intron-NCAG-cut-G
- Stop codon

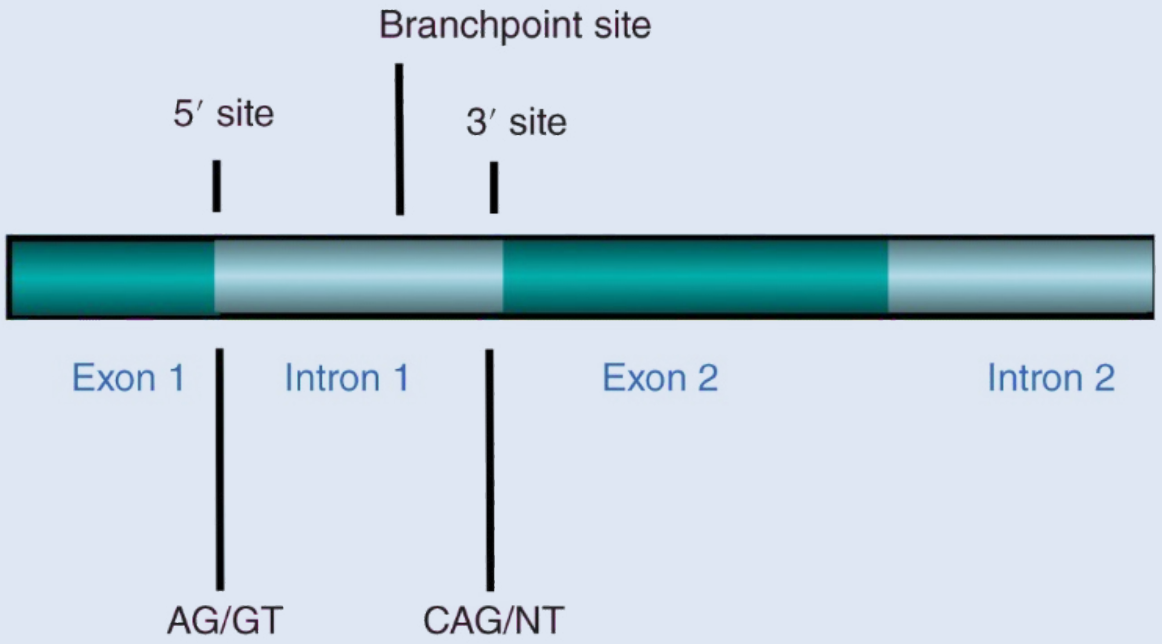
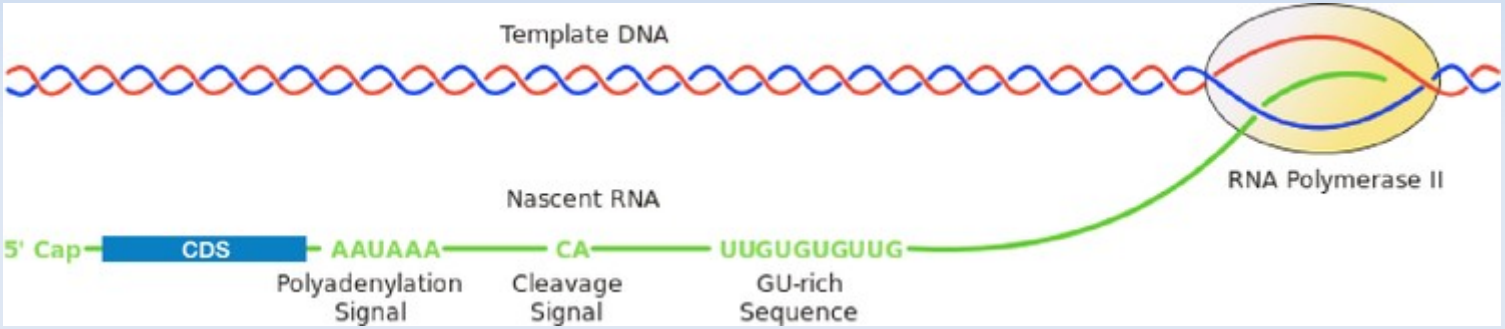


Figure 5.4 A schematic illustration of the splice site regions around exons and introns including the 5′ and 3′ splice sites and their consensus sequences.

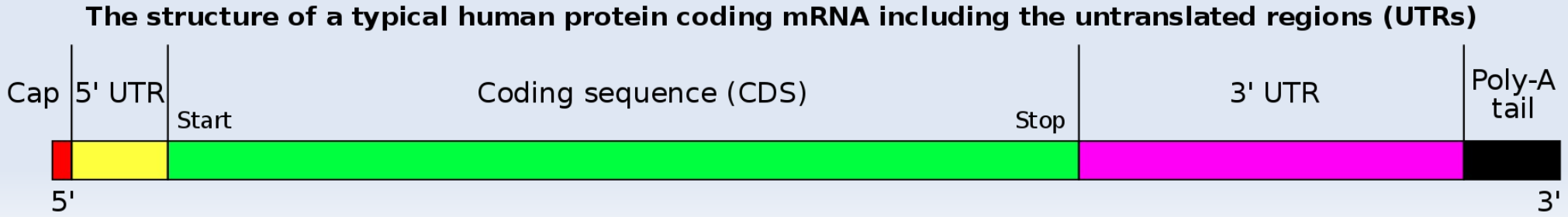
Polyadenylation

At the end of gene transcription:

- the 3’ end of the pre-mRNA is cleaved at the polyadenylation signal (transcribed from DNA): AAUAAA



- Enzyme adds poly-A (string of adenines) tail to 3’ end of mRNA: AAUAAA-20nt-polyA



Annotation: Bacterial prophage seqs

PHASTER - for identification and annotation of prophage sequences (a phage genome incorporated) within bacterial plasmids and genomes

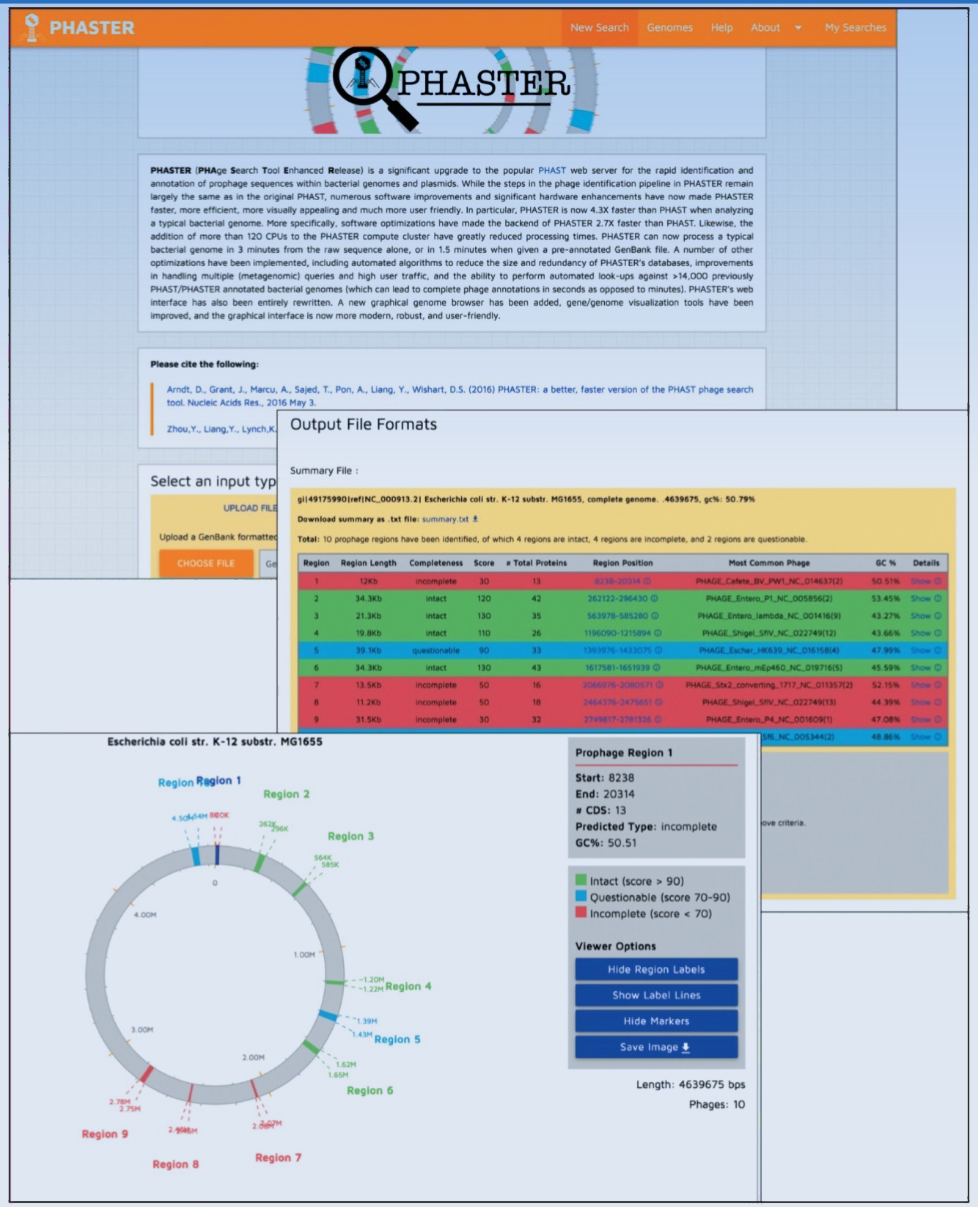


Figure 5.8 A screenshot montage of the PHASTER web server showing the website homepage along with examples output.

Annotations

The *Shigella flexneri* bacteria, NCTC1, has a single 4.5 Mbp chromosome that has been sequenced and assembled more than 12,000 times.

NCTC1 is remarkable because it was isolated in France during WWI. Experiments on this 100 year-old sample indicates that it is resistant many modern antibiotics in use today!

Note: some bacteria fight other bacteria in the wild using antibiotics and evolve resistance to these antibiotics.