



BIO-463

Genomics and bioinformatics

Lecture 2.1: Genome assembly algorithms

Professors: Jacques Rougemont, Anne-Florence Bitbol, Raphaëlle Luisier

EPFL

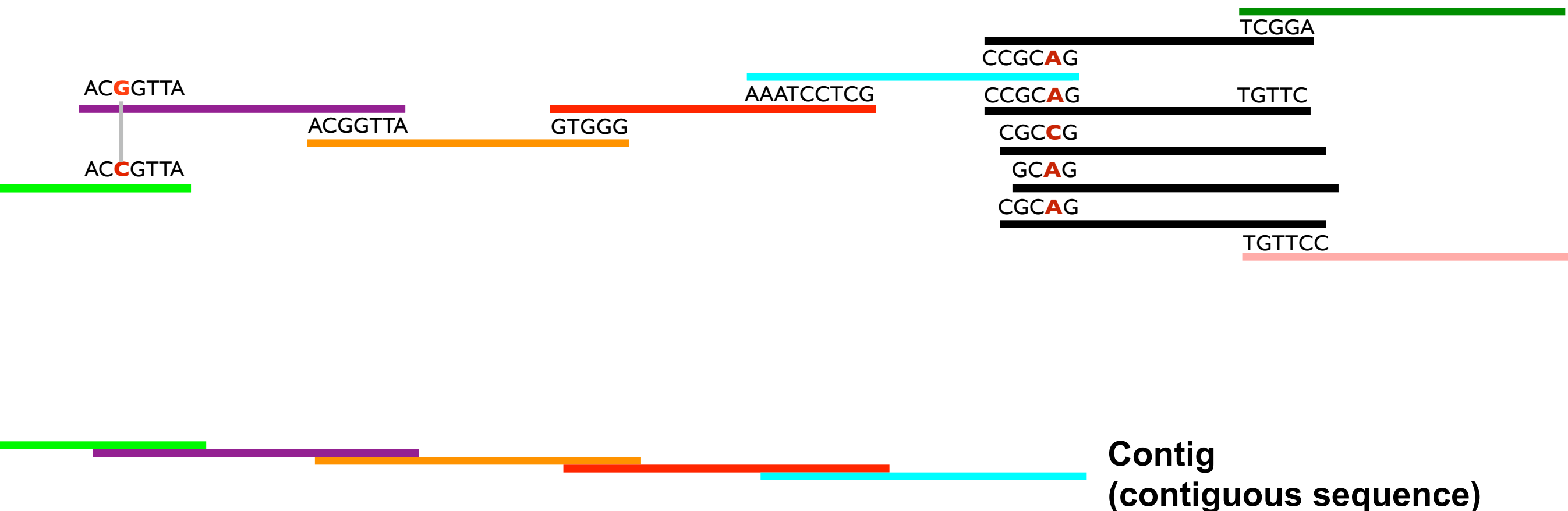
Fragments assembly

General Procedure:

- Overlap → Layout → Consensus

Difficulties:

- Computing overlap with sequencing errors (1-3%) and unknown orientation



Assembling a genome

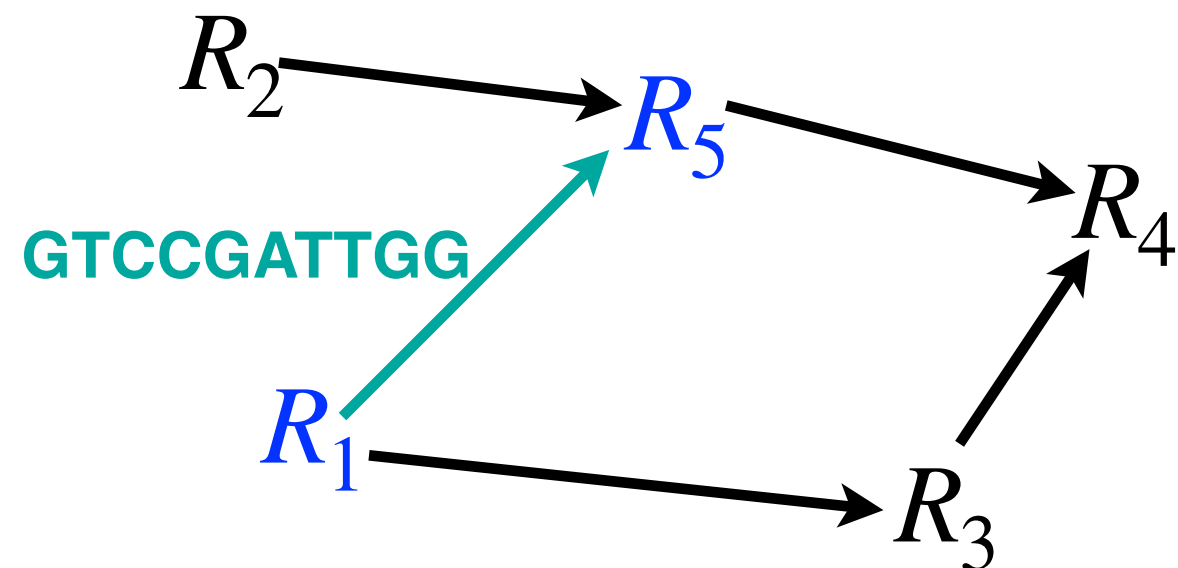
We start with a simpler problem:

sequencing provides N reads R_1, \dots, R_N all of length L ,
when 2 reads overlap, it is always by ℓ nucleotides

$R_1 = \text{ACGTGTCCGATTGG}$
 $R_5 = \text{GTCCGATTGGTGTA}$

$L = 14, \ell = 10$

overlap graph:
vertices = reads
edges = overlaps

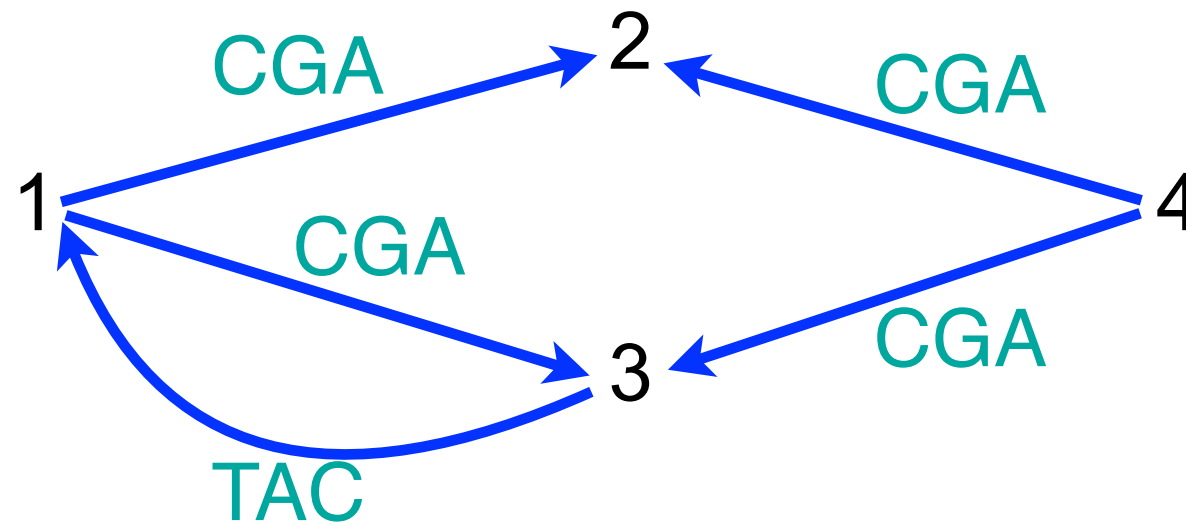


Overlap graph

Contig is a Hamiltonian path

1.TACCGA
2.CGATCG
3.CGATAC
4.ATTCGA

$$N = 4, L = 6, \ell = 3$$



contig:

4 → 3 → 1 → 2

ATTCGA

CGATAC

TACCGA

CGATCG

ATTCGATACCGATCG

Overlap graph

Hamiltonian paths are hard to find

Definition: A **Hamiltonian path** in a graph is a path visiting every vertex once and only once

Finding a Hamiltonian path is a NP-complete problem:
there is no good algorithm to solve this problem

Definition: An **Eulerian path** in a graph is a path visiting every edge once and only once. If the path closes on itself it is called an **Eulerian cycle**.

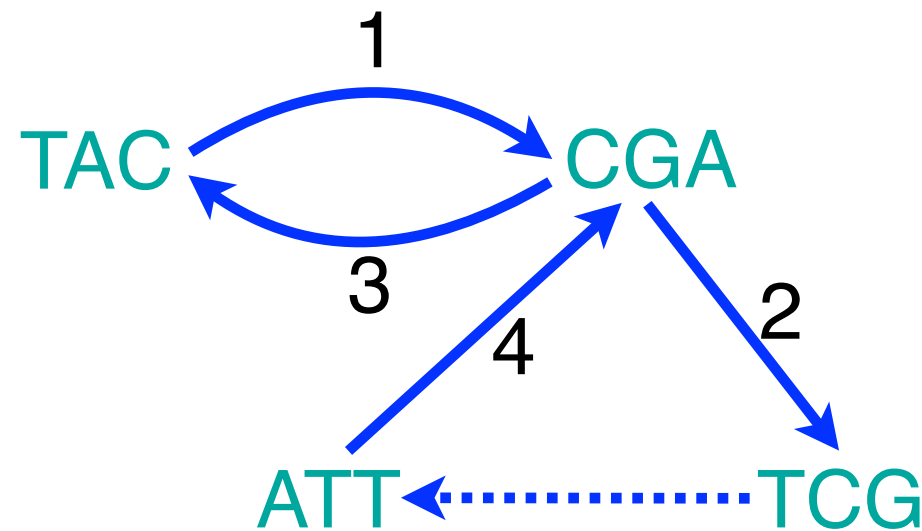
Theorem: There exists an Eulerian cycle in a graph if and only if the graph is balanced: for each vertex v :

$$\text{indegree}(v) = \text{outdegree}(v)$$

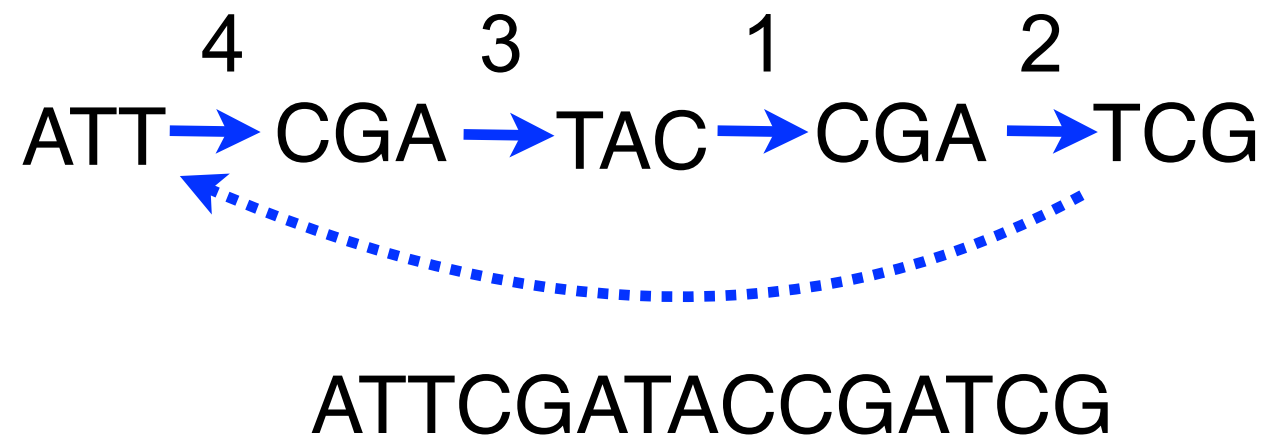
Dual graph

- 1.TACCGA
- 2.CGATCG
- 3.CGATAC
- 4.ATTCGA

vertices are overlaps:
TAC, CGA, TCG, ATT
edges are reads



contig is an Eulerian path:



Euler assembler

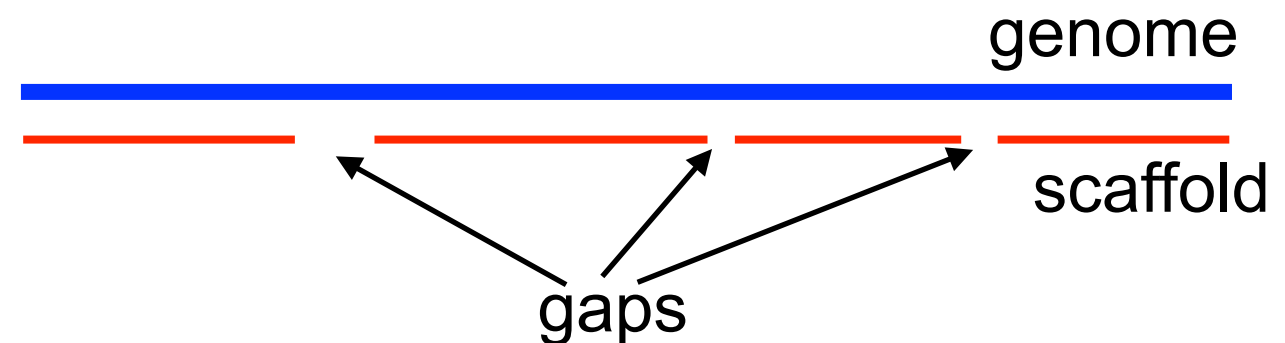
Problem:

- Reads have variable length (sometimes)
- Reads have sequencing errors
- Reads have random orientation
- Overlap size is variable and unknown
- Graph is not balanced and is highly redundant

Strategy:

- Construct a de Bruijn graph
- Heuristically simplify graph
- Extract many quasi-eulerian paths

⇒ many disjoint contigs



de Bruijn graph

Reads:

1. ATTCGAT
2. CGATCG
3. CGATACCGA

overlap parameter:

$$\ell = 4$$

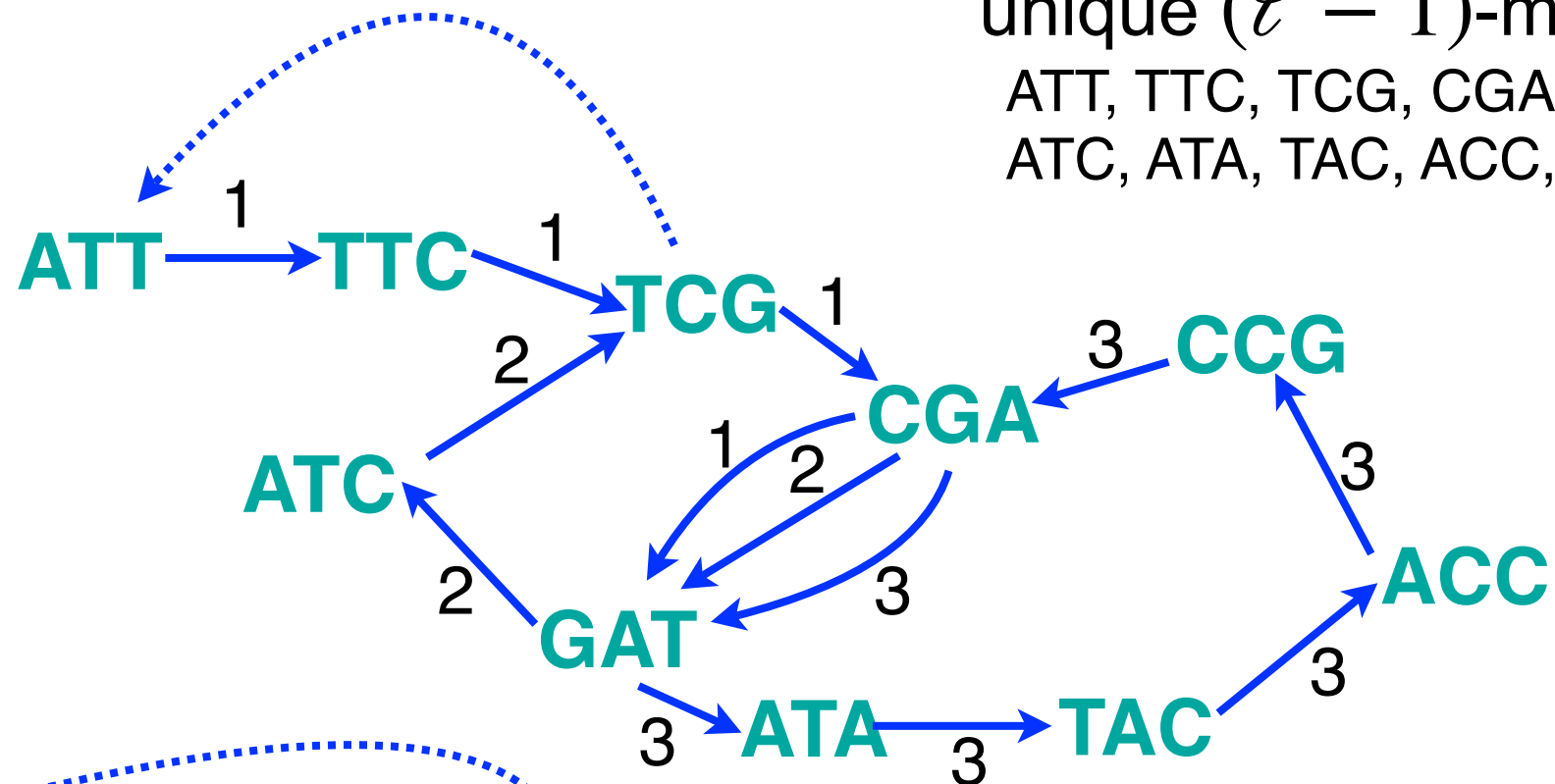
all ℓ -mers:

ATTC, TTCG, TCGA, CGAT,
CGAT, GATC, ATCG,
CGAT, GATA, ATAC, TACC, ACCG, CCGA

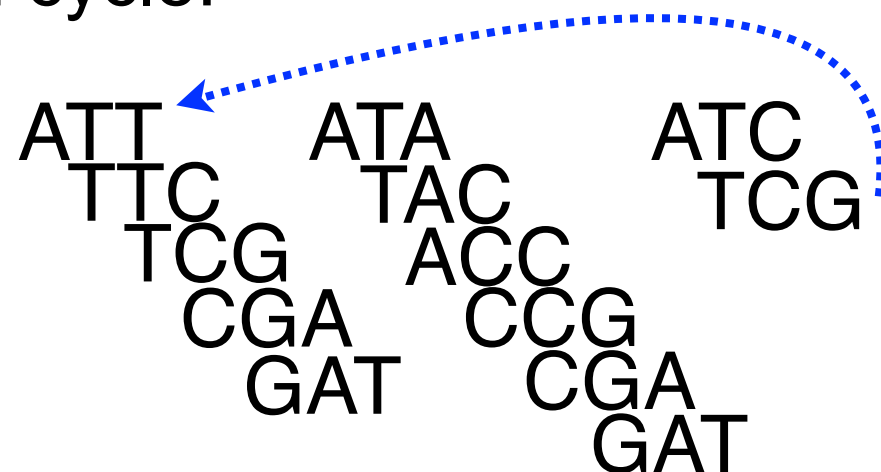
unique $(\ell - 1)$ -mers:

ATT, TTC, TCG, CGA, GAT,
ATC, ATA, TAC, ACC, CCG

Dual graph:



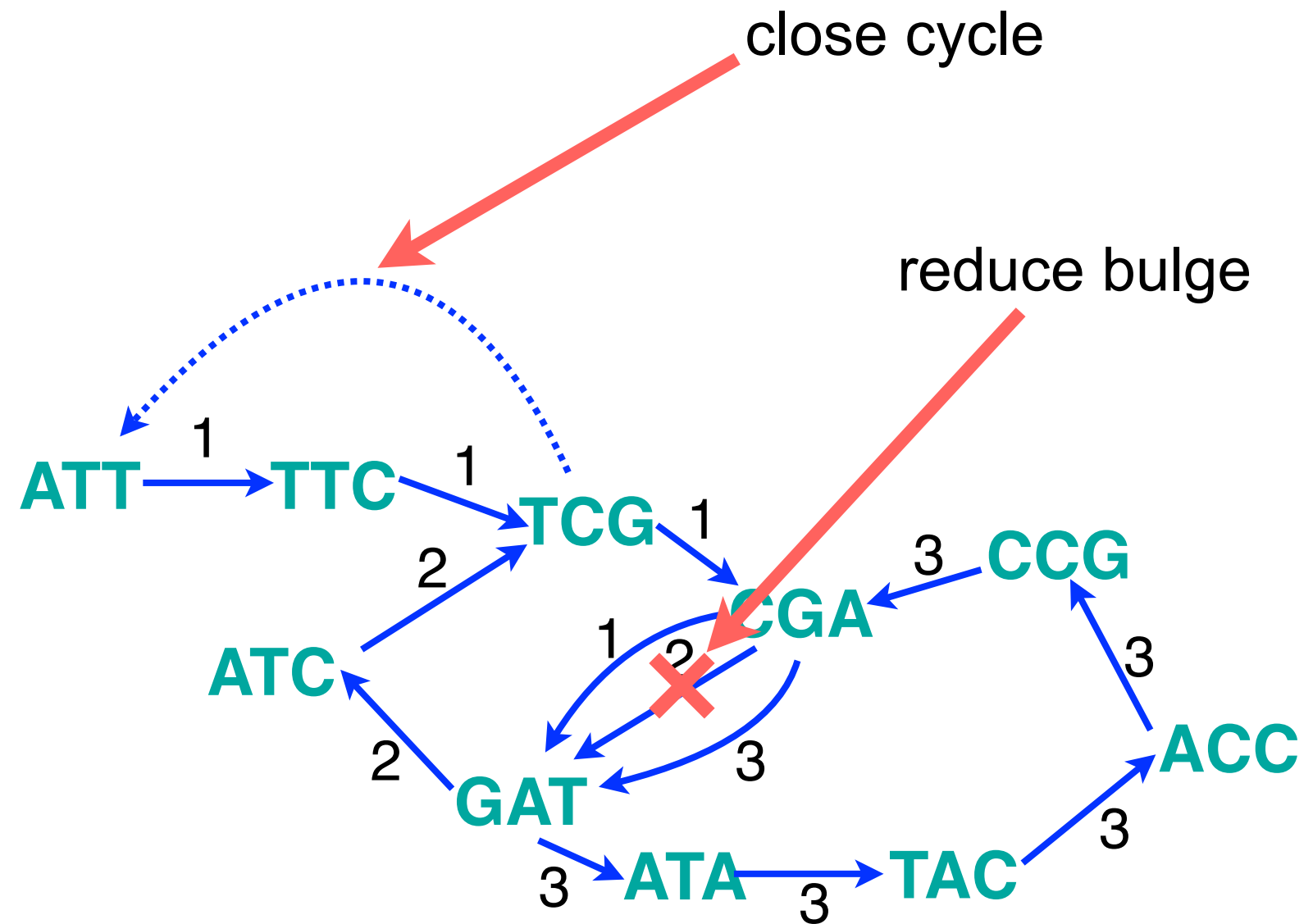
quasi-Eulerian cycle:



ATTCGATACCGATCG

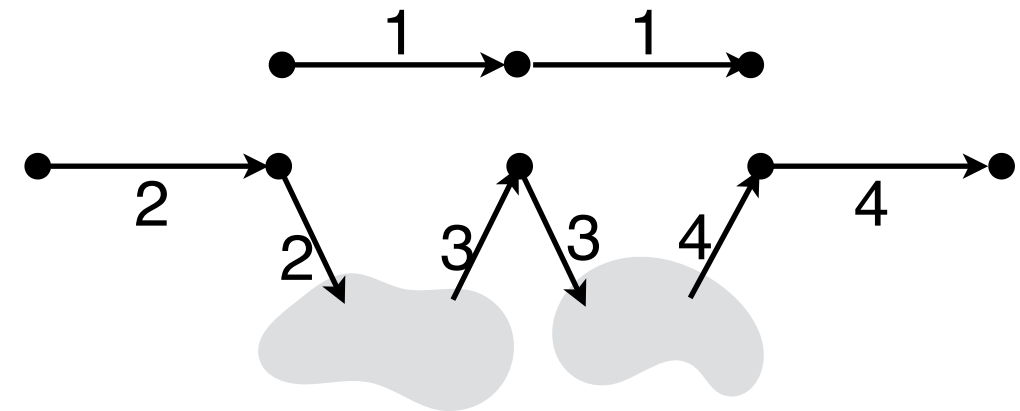
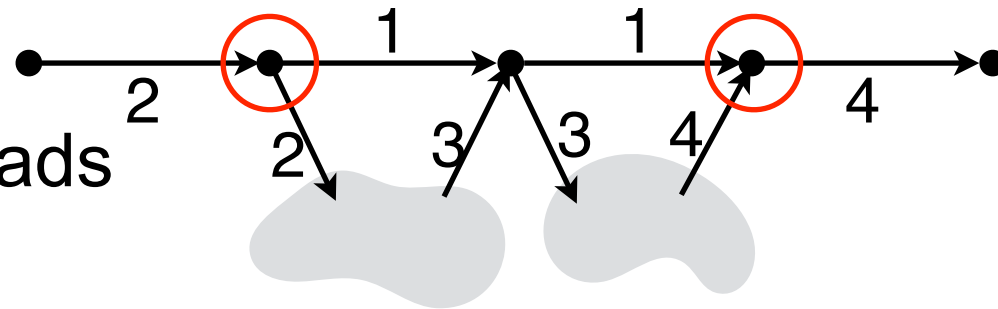
de Bruijn graph

Dual graph:

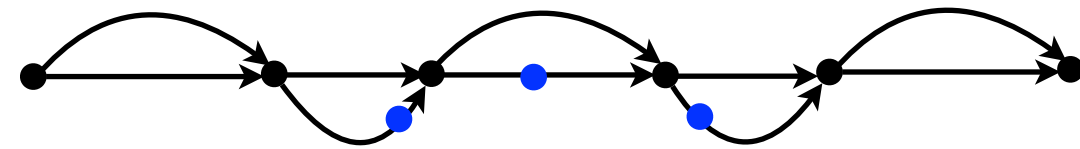
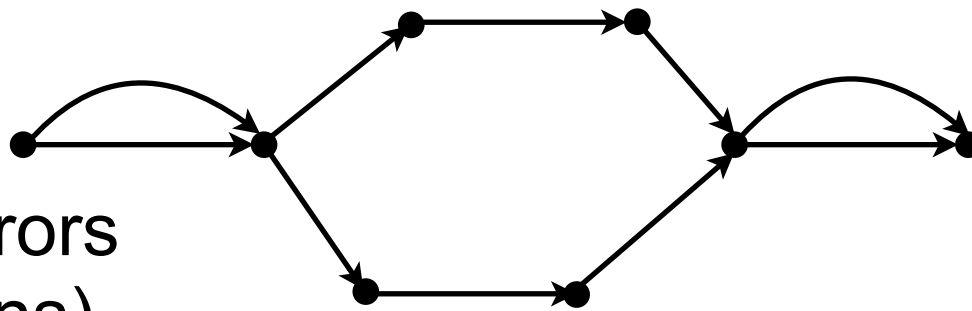


Heuristics: reduce graph imbalance

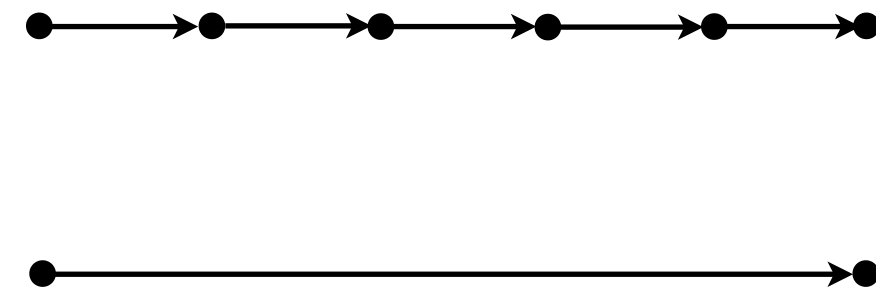
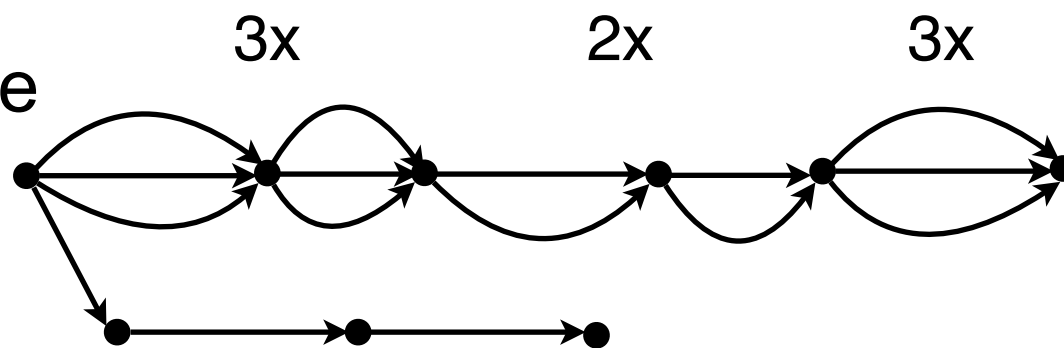
split
incompatible reads



check for
sequencing errors
(try all mutations)



correct for variable
coverage,
dead ends, etc



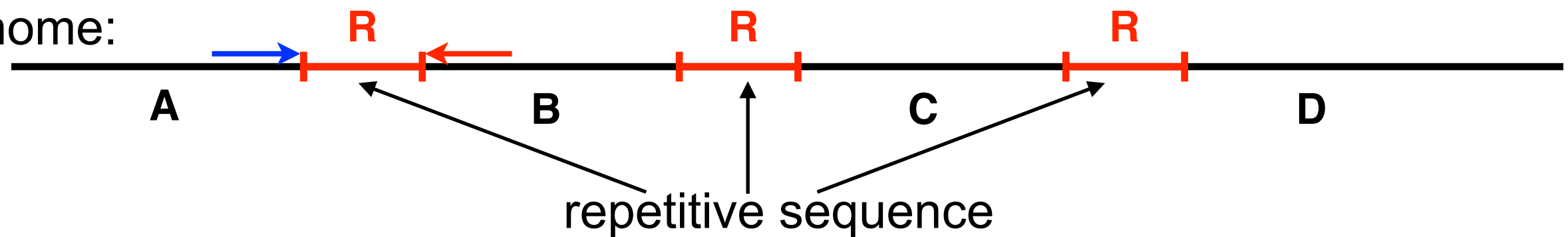
Paired-end sequencing

2 reads from same DNA fragment,
from both ends

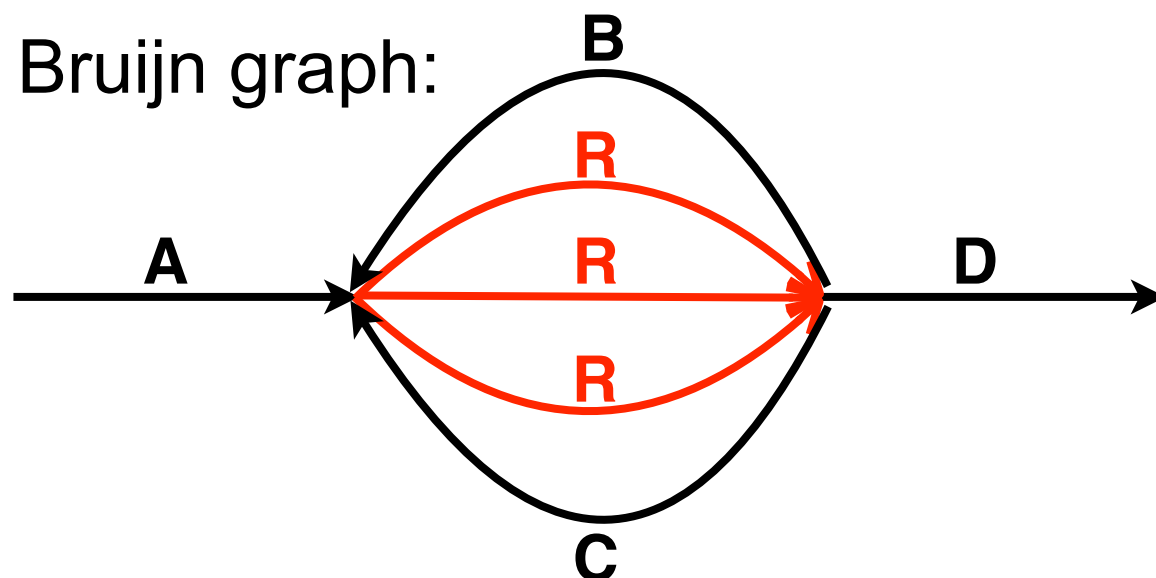


Fragment size known: ~ 10kb
Read length: 1kb

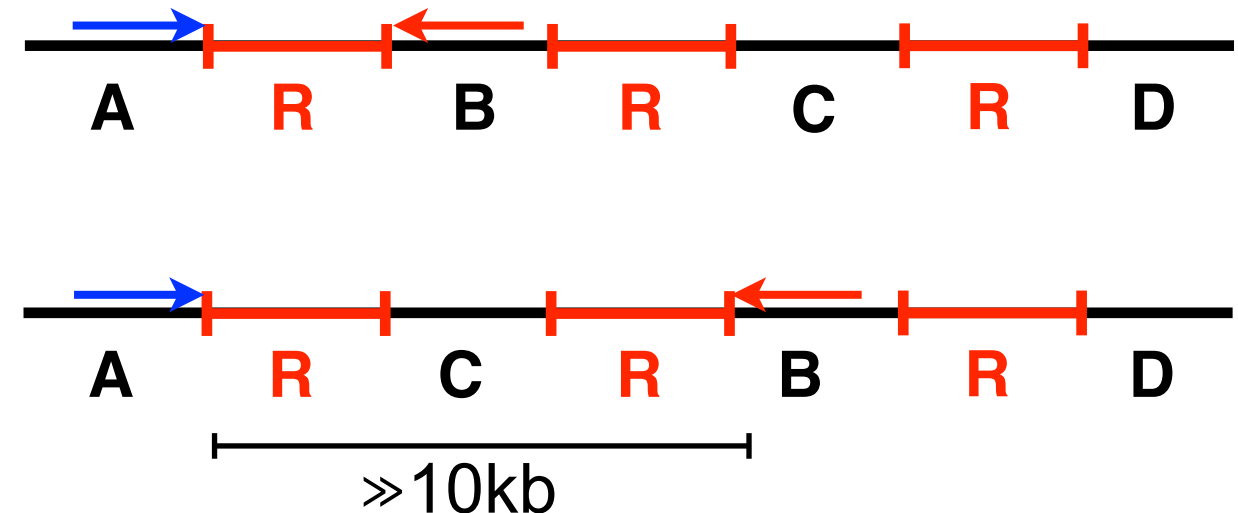
Genome:



de Bruijn graph:



2 possible scaffolds:





BIO-463

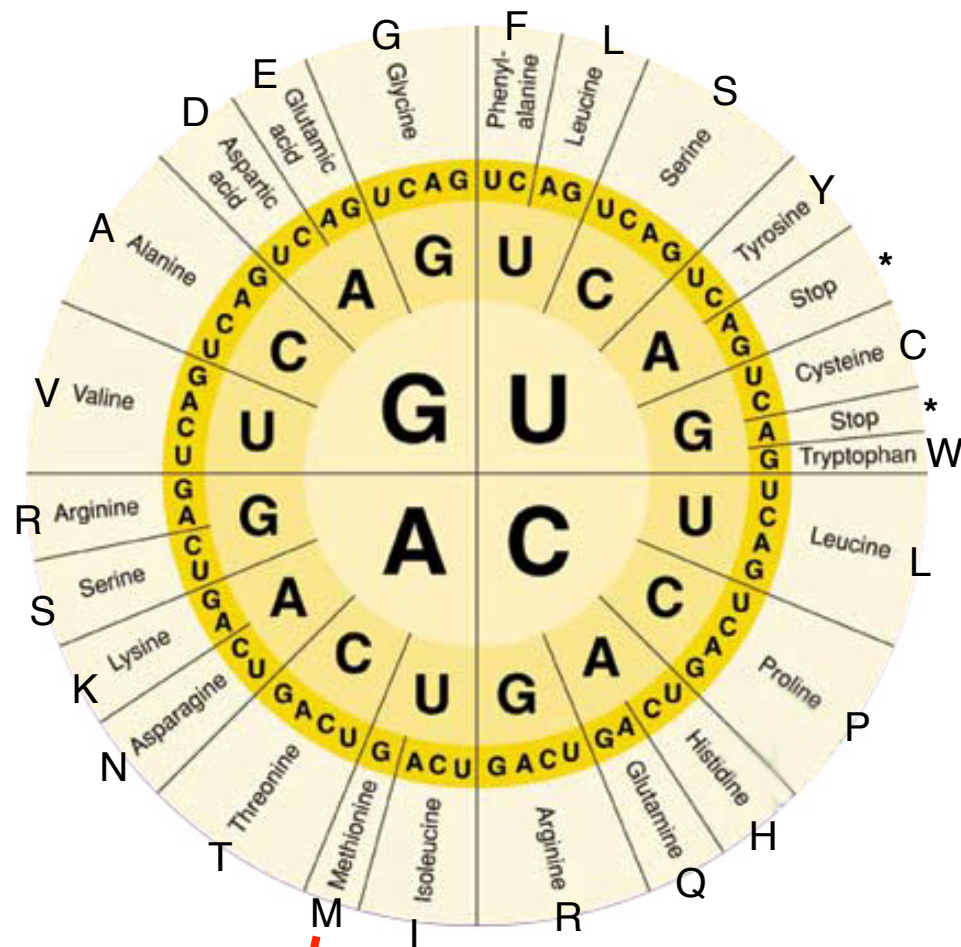
Genomics and bioinformatics

Lecture 2.2: Sequence alignments

Professors: Jacques Rougemont, Anne-Florence Bitbol, Raphaëlle Luisier

EPFL

Open Reading Frames (ORFs)



Methionine (M) = AUG = Start

6-frame translation

atgatcgacgcctcctcagcaagctga

M I D A S S A S *

* S T P P Q Q A

D R R L L S K L

tcagcttgctgaggaggcgatcat

S A C * G G V D H

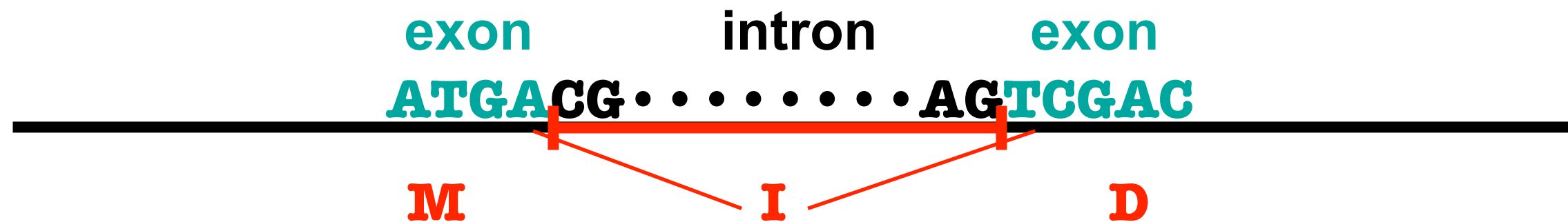
Q L A E E A S I

S L L R R R R S

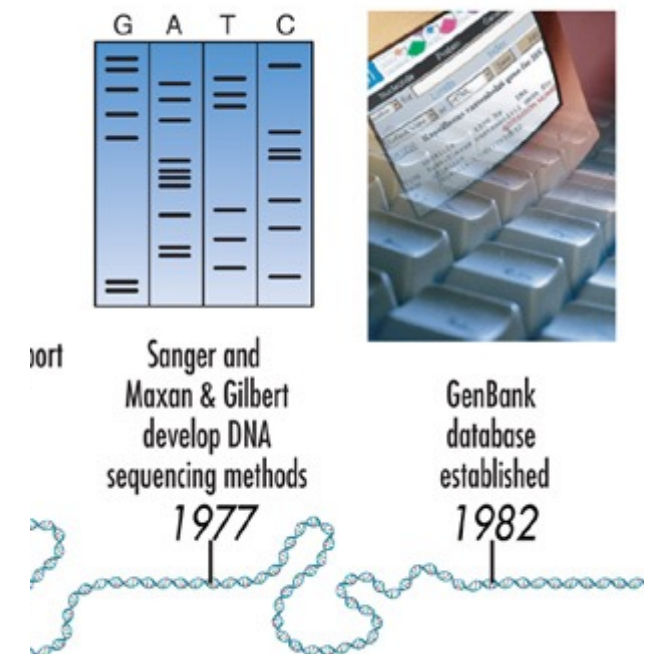
An ORF is anything between M and *

The Genetic Code

- On a bacterial genome, practically all proteins can be identified by direct translation (maybe ignoring short ORFs)
- In eukaryotes, genes have introns and alternative splicing



- We will use comparison to known transcripts to identify gene structures in the genome
- There are large databases of RNA sequences (full transcripts or fragments)



ALIGN calculates a global alignment of two sequences
version 2.2u
Please cite: Myers and Miller, CABIOS (1989) 4:11-17
chr|NC_000068|NC_000068.6 Chromosome 2; [Mus musculus] gi:1 1494 nt
vs.
Shark_HoxD12 504 nt
using matrix file: DNA, gap open/ext: -12/-4
25.2% identity in 1494 nt overlap; Global score: -3279

	190	200	210	220	230	240
chr NC	GAGCACAGCCGAGGCCCTTTGTTGGAGATGTGTGAGCGCAGTCTCTACAGAGCTGGCTAT					
Shark_						:::::
						30
	250	260	270	280	290	300
chr NC	GTGGGCTCGCTTCTGAATTTACAGTCACCGGACTCTTTCTACTTTTCCAACCTGAGAGCC					
Shark_	GTCTGGCTCCCTGTTAAATTTTACCAGCCCGGAGCCCTTCTACTTCCCCAACCTGCGTCCG					
	40	50	60	70	80	90
	310	320	330	340	350	360
chr NC	AATGGCAGCCAGTTGGCCGCGCTTCCCCCATCTCATAACCCTCGCAGCGCGCTGCCCTGG					
Shark_	AATGG					
	370	380	390	400	410	420
chr NC	GCTACTACGCCCCGCTCATGCACCCCTGCGCAGCCTGCCACCGCCTCTGCCTTTGGAGGC					
Shark_	GGCTCAA		CTGGCA		ACTCTGTC	
	100		110			
	430	440	450	460	470	480
chr NC	TTCTCTCAGCCTTACTTGACCGGCTCTGGGCCAATTGGCCTGCAGTCTCCAGGCGCCAAG					
Shark_	GCCAGCACTCTCC					
			120			
	490	500	510	520	530	540
chr NC	GACGGACCCGAAGACCAGGTCAAGTTCTATACGCCTGATGCGCCACCGCATCTGAGGAA					
Shark_	TATAC		CCGCAGG		GAGGTG	
	130		140			

chr NC	CGCAGCCGGACTAGGCCGCCCTTCGCCCCCGAGTCTAGTCTGGTTCATTTCGGCTCTCAAA					
Shark_	TGC	TCGCTCCCGTG				
	150		160			
	610	620	630	640	650	660
chr NC	GGCACCAAGTATGACTACGCGGGTGTGGGCCGGACCGCTCCAGGCTCTGCGACCCTGCTC					
Shark_	GACTTCGAG		TCCATGC		GCATCGCCGCCG	
		170		180		
	670	680	690	700	710	720
chr NC	CAGGGGGCCCCCTGTGCCTCCAGCTTCAAGGAAGACACCAAAGGCCCGCTCAACTTGAAC					
Shark_	CAGAG	CCGCGCCTTCAGCGGCTA	CTCTCA			
	190	200	210			
	730	740	750	760	770	780
chr NC	ATGGCAGTGCAAGTGGCCGGGGTGGCCTCTTGCCTGCGATCTTCACTGCCCCGACGGTAAA					
Shark_	GTCCT		ATCT		CAGCA	ACT
		220		230		
	790	800	810	820	830	840
chr NC	CAGTGCCCATGCTCCCCAAGCCAGTTTAGGCAGGGACGGGAGGTGGGGTGTCAGGGACA					
Shark_	CAGTCTCCAT	CAGCATCAA	TAGGCACGGA	TCAG		
	240	250	260			
	850	860	870	880	890	900
chr NC	GTTGGACAGGGAGGAGACCCGCCAGCAGTGGTGAACGTCTGTGGGGCGGGCAGTTGATCT					
Shark_	ACAAGG	CAGCAG				
	270	280				
	910	920	930	940	950	960
chr NC	GAGCGAGCTGACATGGGTCGGGGCTCTGTTGCAGGCCTGCCGTGGGGGGCGGCCCGGGG					
Shark_	CCGGC					
	970	980	990	1000	1010	1020
chr NC	AGGGCCCGCAAGAAGAGGAAACCCTACACAAAGCAGCAGATTGCGGAGCTGGAGAACGAA					
Shark_	GAAGAG	CCTA	ACAAA			

Sequence alignments

Definition: An alignment of the two sequences X (length n) and Y (length m) is a sequence of operations:

match M , **delete** D , and **insert** I such that

$$\#M + \#D = n \quad \#M + \#I = m$$

CACCGCATC-TG
DDMMMMMMIDM
--CCGCAGGA-G

CACCGCATC-TG
MDMDMMMDMIMM
C-C-GCA-GGAG

- Is one alignment a better choice than the other?
- Are these alignments significant or not?

Sequence alignments

CACCGCATC-TG
DDMMMMMMIDM
--CCGCAGGA-G

CACCGCATC-TG
MDMDMMMDMIMM
C-C-GCA-GGAG

... (167960 possibilities)

How many different alignments exist? $\binom{n+m}{m}$

$m \backslash n$	10	100	200
1	11	101	201
50	$8 \cdot 10^{10}$	10^{40}	10^{53}
100	$5 \cdot 10^{13}$	10^{59}	10^{81}

Scoring an alignment

We calculate a quality score for each alignment based on a scoring matrix

$M =$

	M	A	C	G	T	-	D
A		2	-1	-1	-1	$-\gamma$	
C		-1	2	-1	-1	$-\gamma$	
G		-1	-1	2	-1	$-\gamma$	
T		-1	-1	-1	2	$-\gamma$	
-		$-\gamma$	$-\gamma$	$-\gamma$	$-\gamma$		$-\infty$
I							

Annotations: mismatch (A-C), gap penalty (A-G), match (G-T), mismatch (T-C), gap penalty (T-G), gap penalty (-A), gap penalty (-C), gap penalty (-G), gap penalty (-T), impossible (-D).

sequence+gaps

character no k

$$S(X', Y' | M) = \sum_{k=1}^L M(X'_k, Y'_k)$$

X' CACCGCATC-TG
 Y' DDMMMMMMIDM
 --CCGCAGGA-G

$$-\gamma - \gamma + 2 + 2 + 2 + 2 + 2 - 1 - 1 - \gamma - \gamma + 2 = 10 - 4\gamma$$

Scoring matrix must be:

- symmetric
- diagonal > 0 , off-diagonal < 0
- $M(-, -)$ impossible: $-\infty$

We always use:

- All diagonal element equal
- All gaps equal

Scoring an alignment

We calculate a quality score for each alignment based on a scoring matrix

$$M = \begin{matrix} & \begin{matrix} A & C & G & T & - \end{matrix} \\ \begin{matrix} A \\ C \\ G \\ T \\ - \end{matrix} & \begin{pmatrix} 2 & -1 & -1 & -1 & -\gamma \\ -1 & 2 & -1 & -1 & -\gamma \\ -1 & -1 & 2 & -1 & -\gamma \\ -1 & -1 & -1 & 2 & -\gamma \\ -\gamma & -\gamma & -\gamma & -\gamma & -\infty \end{pmatrix} \end{matrix}$$

$$S_{\text{affine}}(X', Y' | M, \delta) = \sum_{k=1}^L M(X'_k, Y'_k) - G\delta$$

number of gap opening

affine gap penalty

gap openings

X' CACCGCATC-TG
 Y' DDMMMMMMIDM
 --CCGCAGGA-G

$$-\gamma - \gamma + 2 + 2 + 2 + 2 + 2 - 1 - 1 - \gamma - \gamma + 2 - 3\delta = 10 - 4\gamma - 3\delta$$

Examples

Which of these alignments is better depends on choice of scoring matrix

CACCGCATCTG
--CCGCAGGAG
+++

CACCGCATCTG---
--CCGCA---GGAG

$$\gamma = 2 : \quad -4 - 3 + 12 = 5 \quad > \quad -16 + 12 = -4$$

$$\gamma = 0 : \quad -3 + 12 = 9 \quad < \quad 12$$

$$\gamma = 0, \delta = 2 : \quad -3 + 12 - 2 = 7 \quad > \quad 12 - 6 = 6$$

$$M = \begin{matrix} & \begin{matrix} A & C & G & T & - \end{matrix} \\ \begin{matrix} A \\ C \\ G \\ T \\ - \end{matrix} & \left(\begin{array}{ccccc} 2 & -1 & -1 & -1 & -\gamma \\ -1 & 2 & -1 & -1 & -\gamma \\ -1 & -1 & 2 & -1 & -\gamma \\ -1 & -1 & -1 & 2 & -\gamma \\ -\gamma & -\gamma & -\gamma & -\gamma & -\infty \end{array} \right) \end{matrix}$$

Global alignment: The Needleman-Wunsch algorithm

- Can we find the best scoring alignment (given M) without searching through all possibilities?
- **Dynamic programming:** a class of algorithms that work by recursively extending the solution of a sub-problem.
- We find the alignment of sequences of length (n, m) by extending the alignments of lengths $(n - 1, m - 1)$, $(n, m - 1)$, $(n - 1, m)$

$$F_{i,j} = \max \begin{cases} F_{i-1,j-1} + M(Y_i, X_j) \\ F_{i,j-1} + M(-, X_j) \\ F_{i-1,j} + M(Y_i, -) \end{cases}$$

recursive formula

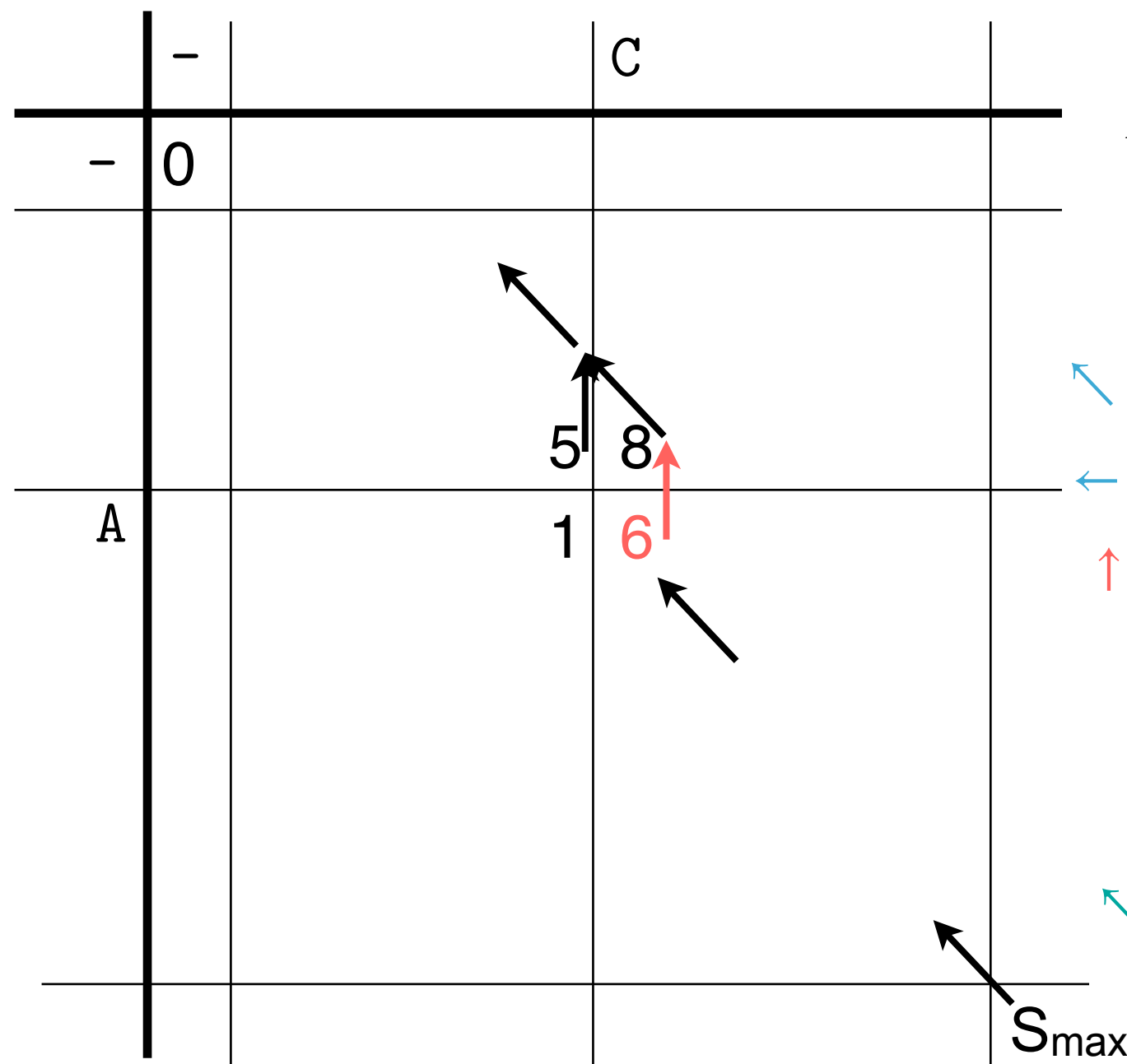
scoring table

	-	C	A	...	G
-	0	$-\gamma$	-2γ	...	$-n\gamma$
C	$-\gamma$	F_{11}	F_{12}	...	F_{1n}
C	-2γ	F_{21}	F_{22}	...	F_{2n}
⋮	⋮	⋮	⋮	⋮	⋮
G	$-m\gamma$	F_{m1}	F_{m2}	...	F_{mn}

Global alignment: The Needleman-Wunsch algorithm

$$F_{i,j} = \max \begin{cases} F_{i-1,j-1} + M(Y_i, X_j) \\ F_{i,j-1} + M(-, X_j) \\ F_{i-1,j} + M(Y_i, -) \end{cases}$$

- Optimal score is at bottom-right
- Backtracking follows optimal alignment



$$M = \begin{matrix} & \begin{matrix} A & C & G & T & - \end{matrix} \\ \begin{matrix} A \\ C \\ G \\ T \\ - \end{matrix} & \begin{pmatrix} 2 & -1 & -1 & -1 & -2 \\ -1 & 2 & -1 & -1 & -2 \\ -1 & -1 & 2 & -1 & -2 \\ -1 & -1 & -1 & 2 & -2 \\ -2 & -2 & -2 & -2 & -\infty \end{pmatrix} \end{matrix}$$

$$\nwarrow 5 + M(A, C) = 5 - 1 = 4$$

$$\leftarrow 1 + M(-, C) = 1 - 2 = -1$$

$$\uparrow 8 + M(A, -) = 8 - 2 = 6$$

$$\nwarrow = M \quad \uparrow = I \quad \leftarrow = D$$

Global alignment: The Needleman-Wunsch algorithm

$$M = \begin{matrix} & \begin{matrix} A & C & G & T & - \end{matrix} \\ \begin{matrix} A \\ C \\ G \\ T \\ - \end{matrix} & \begin{pmatrix} 2 & -1 & -1 & -1 & -2 \\ -1 & 2 & -1 & -1 & -2 \\ -1 & -1 & 2 & -1 & -2 \\ -1 & -1 & -1 & 2 & -2 \\ -2 & -2 & -2 & -2 & -\infty \end{pmatrix} \end{matrix}$$

	-	C	A	C	C	G
-	0	-2	-4	-6	-8	-10
C	-2	2	0	-2	-4	-6
C	-4	0	1	2	0	-2
C	-6	-2	-1	3	4	2
G	-8	-4	-3	1	2	6

Cost of the algorithm (in time and memory):

$$\mathcal{O}((m+1)(n+1)) = 6 \times 5 = 30$$

compare to number of possible alignments:

$$\binom{m+n}{m} = \binom{9}{4} = 126$$

CACCG
MDMMM
C-CCG

$\nwarrow = M \quad \uparrow = I \quad \leftarrow = D$

Local alignment: The Smith-Waterman algorithm

$$F_{i,j} = \max \begin{cases} 0 \\ F_{i-1,j-1} + M(Y_i, X_j) \\ F_{i,j-1} + M(-, X_j) \\ F_{i-1,j} + M(Y_i, -) \end{cases}$$

- Optimal score is highest anywhere in table
- Backtrack from a high score until you reach a 0

Possible local alignments:

CCG
MMM
CCG
score: 6

CC
MM
CC
score: 4

CAC
MMM
CCC
score: 3

	-	C	A	C	C	G
-	0	0	0	0	0	0
C	0	2	0	2	2	0
C	0	2	1	2	4	2
C	0	2	1	3	4	3
G	0	0	1	1	2	6

How to make a scoring matrix

- The scoring matrix contains "prior information" about what we consider a relevant alignment
- A standard interpretation of alignment scores is as log-likelihood ratios
- You can estimate them empirically

Negative set: random sequences

CGCA-CATG-TG
-CAGTAG-TAGT

Frequency of nucleotide pair = $q_C \cdot q_A$
= product of individual frequencies

$$L_{\text{random}}(X, Y) = \prod_i q_{x_i} q_{y_i}$$

$$\mathcal{L}(X, Y) = \log \left(\frac{L_{\text{model}}(X, Y)}{L_{\text{random}}(X, Y)} \right) = \sum_i (\log p_{x_i y_i} - \log q_{x_i} - \log q_{y_i}) = \sum_i M(x_i, y_i)$$

Positive set: curated pairs of homologous sequences

CGCATCATG-GT
-GCATG--CAAT

Count frequency of each nucleotide pair = p_{xy}

$$L_{\text{model}}(X, Y) = \prod_i p_{x_i y_i}$$

Empirical chemical similarity of amino-acids:
[wikipedia:Substitution_matrix](https://en.wikipedia.org/wiki/Substitution_matrix)

BLAST

Basic Local Alignment Software Tool

search local alignments of query ("gene") in a large database ("genome")

1. Remove low-complexity (repeat-like) regions from query
2. Cut query in small words (DNA: 11 bases, AA: 3 residues), look for exact matches in the database (pre-computed table)
3. Perform a Smith-Waterman alignment in the neighborhood of each hit to produce a high-scoring segment pair (HSP)

Ranking of HSP is performed by **E-value**, assuming an extreme value distribution:

$$E = Kmn e^{-\lambda S}$$

S-W score

sizes of query and database

Parameters K, λ have been empirically tuned.

Ranking will not change if you rescale all scores as $S' = \frac{\lambda S - \log K}{\log 2}$

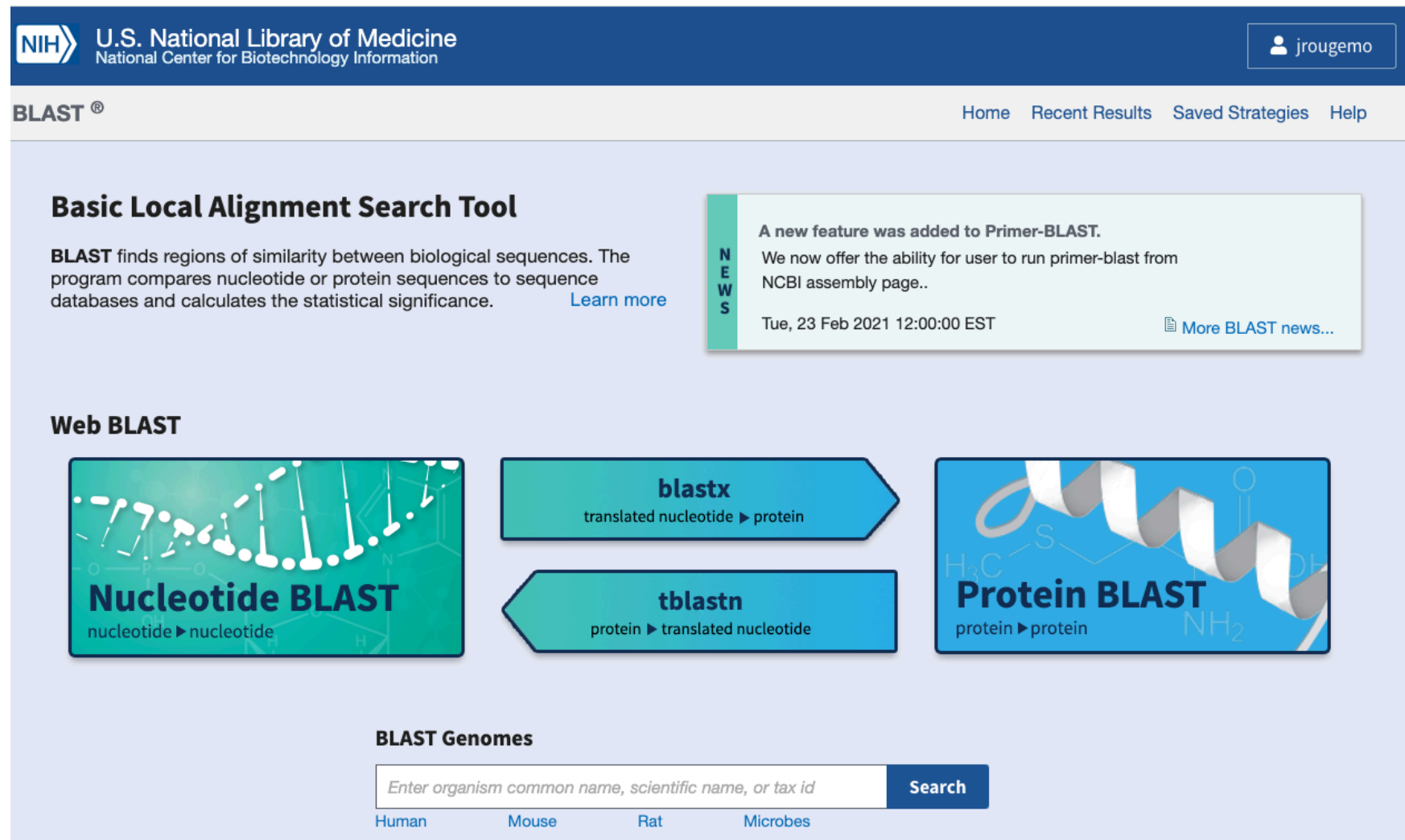
$$E = mn 2^{-S'}$$

"bit-score"

BLAST

Basic Local Alignment Software Tool

search local alignments of query ("gene") in a large database ("genome")



The screenshot shows the BLAST web interface from the U.S. National Library of Medicine. The header includes the NIH logo and the text "U.S. National Library of Medicine National Center for Biotechnology Information". A user profile "jrougemo" is visible in the top right. The main navigation bar contains links for "Home", "Recent Results", "Saved Strategies", and "Help". The "BLAST" logo is on the left. The main content area features the "Basic Local Alignment Search Tool" title and a description: "BLAST finds regions of similarity between biological sequences. The program compares nucleotide or protein sequences to sequence databases and calculates the statistical significance." A "Learn more" link is provided. A "Web BLAST" section offers three options: "Nucleotide BLAST" (nucleotide to nucleotide), "blastx" (translated nucleotide to protein), and "tblastn" (protein to translated nucleotide). "Protein BLAST" (protein to protein) is also shown. A "BLAST Genomes" section includes a search bar with the placeholder "Enter organism common name, scientific name, or tax id" and a "Search" button. Below the search bar are links for "Human", "Mouse", "Rat", and "Microbes". A "NEWS" sidebar on the right announces a new feature for Primer-BLAST, dated "Tue, 23 Feb 2021 12:00:00 EST", with a link to "More BLAST news..."

blast.ncbi.nlm.nih.gov

- **Database:** dog genome
- **Query:** human BRCA1

[< Edit Search](#)
[Save Search](#)
[Search Summary ▾](#)

Search Parameters	
Program	blastn
Word size	28
Expect value	0.05
Hitlist size	100
Match/Mismatch scores	1,-2
Gapcosts	0,2.5
Low Complexity Filter	Yes
Filter string	L;m;
Genetic Code	1

Database	
Posted date	Jan 8, 2021 2:33 PM
Number of letters	4,807,835,216
Number of sequences	3,644
Entrez query	Includes: Canis lupus familiaris (taxid:9615)

Karlin-Altschul statistics		
Lambda	1.33271	1.28
K	0.620991	0.46
H	1.12409	0.85

Results Statistics	
Length adjustment	33
Effective length of query	7055
Effective length of database	4807714964
Effective search space	33918429071020
Effective search space used	33918429071020

Job Title	ref[NM_007294.4]
RID	44HCYTK7013 <small>Search expires on 03-06 21:52 pm</small> Download All ▼
Program	BLASTN ? Citation ▼
Database	Genome (ROS_Cfam_1.0 reference, Annotation Release 106) See details ▼
Query ID	NM_007294.4
Description	Homo sapiens BRCA1 DNA repair associated (BRCA1), tr...
Molecule type	rna
Query Length	7088
Other reports	Distance tree of results MSA viewer ?

Filter Results

Organism only top 20 will appear

Type common name, binomial, taxid

[+ Add organism](#)

Percent Identity to **E value**

Descriptions	Graphic Summary	Alignments	Taxonomy
Alignment view	Pairwise	<input type="checkbox"/> CDS feature ?	Restore defaults

1 sequences selected [?](#)

[Download](#) ▼ [GenBank](#) [Graphics](#) Sort by: ▼

Canis lupus familiaris isolate SID07034 breed Labrador retriever chromosome 9, ROS_Cfam_1.0

Sequence ID: [NC_051813.1](#) Length: 62002293 Number of Matches: 7

Range 1: 20703495 to 20706941 [GenBank](#) [Graphics](#) ▼ [Next Match](#) ▲ [Previous Match](#)

Score	Expect	Identities	Gaps	Strand
3097 bits(1677)	0.0	2878/3465(83%)	54/3465(1%)	Plus/Plus
Query 783	GCTGCTTGTGAATTTTCTGAGACGGATGTAACAAATACTGAACATCATCAACCCAGTAAT	842		
Sbjct 20703495	GCTGCTTGTGAATTTTCTG-G--GGACATAACAAATATTGAACATCATCAATCCGGTAAT	20703551		
Query 843	AATGATTTGAACACCACTGAGAAGCGTGCAGCTGAGAGGCATCCAGAAAAGTATCAGGGT	902		
Sbjct 20703552	AAAGATTGACCACCACTGAGAAGCATGCAACTAAGAAGCATCCAGAAAAGTATCAGGGT	20703611		

Range 4: 20712451 to 20712622 [GenBank](#) [Graphics](#) ▼ [Next Match](#) ▲ [Previous Match](#) ▲ [First Match](#)

Score	Expect	Identities	Gaps	Strand
191 bits(103)	4e-45	149/172(87%)	0/172(0%)	Plus/Plus
Query 4296	CAGCAGAGGGATACCATGCAACATAACCTGATAAAGCTCCAGCAGGAAATGGCTGAACTA	4355		
Sbjct 20712451	CAGCAGAGAGATACCATGCAAGATAACCTGATAAAGCTCCAGCAGGAAATGGCTGAACTG	20712510		
Query 4356	GAAGCTGTGTTAGAACAGCATGGGAGCCAGCCTTCTAACAGCTACCCTTCCATCATAAGT	4415		
Sbjct 20712511	GAAGCTGTGTTAGAGCAGCATGAGAGCCAGCCCTCTAACAGCTCCCTTCCCTTATAGCA	20712570		
Query 4416	GACTCTTCTGCCCTTGAGGACCTGCGAAATCCAGAACAAAGCACATCAGAAA	4467		
Sbjct 20712571	GATTCTTGTTCGCCTGAGGATCTGCTGAATCCGGAACAAAACGCATCAGAAA	20712622		

Previous Match First Match		CTTGCATGTGGAGCCATGTGGCACAATACTCATGCCAGCTCATTA	962
		CTTGCATGTGGAGCCATGTGGCACAATACTCATGCCAGCTCATTA	20703671
strand			
plus/Plus			
GAAATGGCTGAACTA	4355	GCAGTTTATTACTCACTAAAGACAGAATGAATGTAGAAAAGGCTGAA	1022
GAAATGGCTGAACTG	20712510	GCAGTTTATTACTCACTAAACACAGAATGAATGTAGAAAAGGCTGAA	20703731
CAAAACAGCCTGGCTTAGCAAGGAGCCAACATAACAGATGGGCTGGA		CAAAACAGCCTGGCTTAGCAAGGAGCCAACATAACAGATGGGCTGGA	1082
CAAAACAGCCTGGCTTAGCAAGGAGCCAACAGAGCAGATGGGCTGAA		CAAAACAGCCTGGCTTAGCAAGGAGCCAACAGAGCAGATGGGCTGAA	20703791
CCTTCCATCATAAGT	4415	CTAATGATAGGCGGACTCCCAGCACAGAAAAAGGTAGATCTGAAT	1142
CCTTCCCTTATAGCA	20712570	CTAATGATAGGCGGACTCCCAGCACAGAAAAAGGTAGATCTGAAT	20703851
TCAGAAA	4467	CTGAGAGAGAAAAGAACTGAATAAACAGAAACCTCCACACTCTGATAGT	1202
TCAGAAA	20712622	CTGAGAGAGAAAAGAACTGAATAAACAGAAACCTCCACACTCTGATAGT	20703911
Query	1203	CCTAGAGATACTGAAGATGTTTCCTTGATAAACTAAATAGCAGCATTCAGAAAAGTTAAT	1262
Sbjct	20703912	CCTAGAGATTCCCAAGATGTTTCCTTGATAAACTGAATAGTAGCATACGGAAAAGTTAAT	20703971

UCSC BLAT = "BLAST-like alignment tool"

UCSC BLAT Search Genome

Genome: ☐ Search all
D. melanogaster

Query type: Release 6 + ISO1 MT/dm6) ▼

BLAT's guess ▼

☐ All Results (no minimum matches)

Paste in a query sequence to find its location in the the genome. Multiple sequences may be searched if separated by line sequence name.

File Upload: Rather than pasting a sequence, you can choose to upload a text file containing the sequence.

Upload sequence: No file chosen

Only DNA sequences of 25,000 or fewer bases and protein or translated sequence of 10000 or fewer letters will be processed. The total limit for multiple sequence submissions is 50,000 bases or 25,000 letters.

genome.ucsc.edu