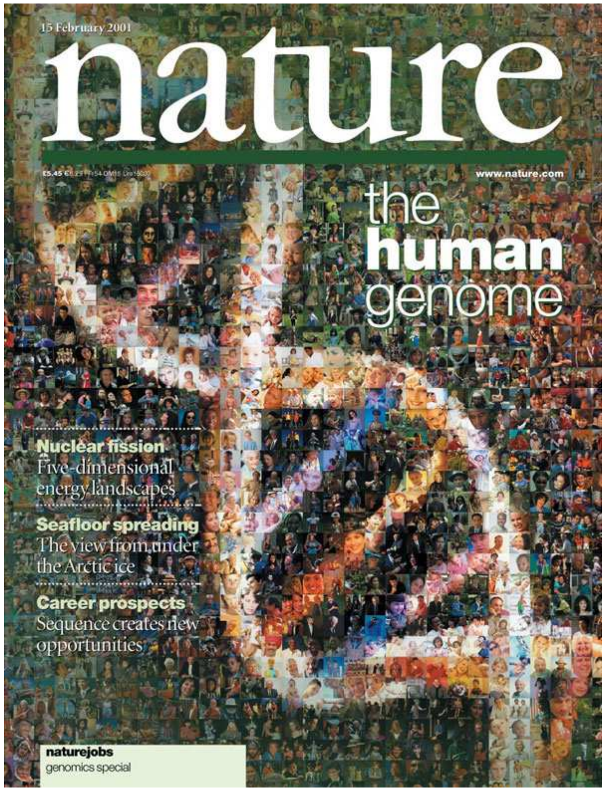
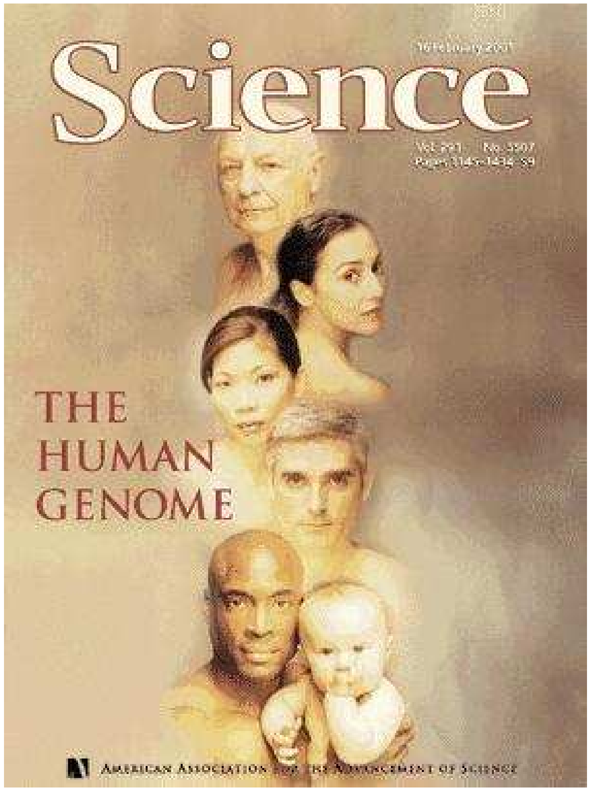
Genome Sequencing and Assembly
(Sekvenovanie a zostavovanie genómov)

Tomáš Vinař
23.9.2021





DNA Sequencing Overview

1. Chromosomes are cut randomly into smaller fragments
   (e.g. using **sonication**)

2. Each fragment is copied multiple times
   (e.g. through PCR, bacterial cloning, . . .)

3. Ends of fragments are sequenced by one of the sequencing technologies
   $\Rightarrow$ many short strings called **reads**

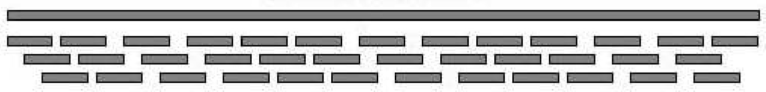
4. Short strings are **computationally assembled** back into chromosomes

Overview of Sequencing Technologies

Technology	Read length	Errors	Output per day	Cost per MB
1st generation				
Sanger	up to 1000bp	$< 1\%$	3 MB	$4000
2nd (next) generation (cca 2004)				
Illumina	250bp	$< 0.1\%$	150 GB	$0.03
3rd generation (emerging)				
PacBio	cca 14kbp	10%	700 GB	$0.02
PacBio HiFi	cca 15kbp	$< 1\%$	70 GB	$0.20
Oxford Nanopore	really long	up to 10%	50 GB	$0.02

Bioinformatics Problem: Sequence Assembly (zostavenie genómu)

- **Input:** short DNA fragments (reads)
- **Goal:** reconstruct the sequenced genome
  — using sequence identity in overlapping reads
- Important factors:
  - **Size of the genome**
  - **Length of individual reads**
  - **Coverage** — how many times on average is the genome covered?



Simple but Unrealistic Formulation

Shortest common superstring problem.

We are given several strings $S_1, \ldots S_k$ (sequenced reads), find the shortest string S containing each S_i as a (contiguous) substring

Motivation: use overlaps between reads as much as possible

Example:

Input: GCCAAC,CCTGCC,ACCTTC

Output: CCTGCCAACCTTC (reads connected in order S_2, S_1, S_3)

Shortest Common Superstring

- **NP hard problem**
  no known polynomial-time algorithm can find optimal answer for each input
- **Simple heuristics:** repeatedly find two reads with longest overlap and connect them to a single read
- Example: CATATAT, TATATA, ATATATC
  Optimum: CATATATATC, length 10
  Heuristics: CATATATCTATATA, length 14
- This heuristics is an **approximation algorithm:**
  It finds a string which is at most $3.5\times$ longer than optimal superstring
- Conjecture: it is in fact a 2-approximation algorithm
- There is a different 2.5-approximation algorithm

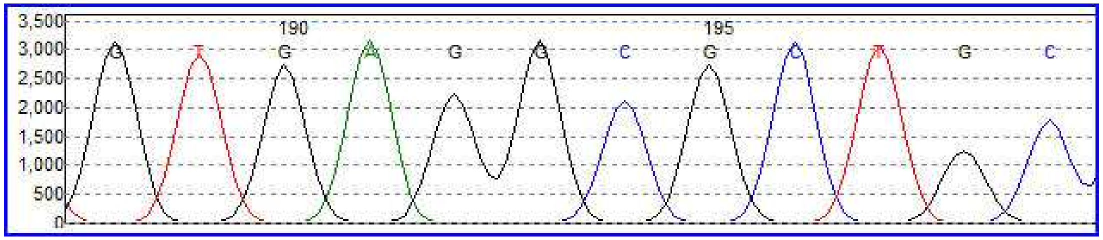
Shortest Common Superstring: Unaccounted Factors

- Sequencing errors
- Polymorphism
- Two strands (reads come in two different orientations)
- Contamination (e.g. by DNA from bacteria used for cloning), chimeric reads
- Multiple chromosomes, incomplete genome coverage
- Sequence repeats
  cca 50% of human genome is repetitive DNA
  Example: 10xTTAATA, 10xATATTA, 3xTTAGCT
  TTAATATTAGCT?
  TTAATATTAATATTAATATTAATATTAGCT?
  TTAATATTA + ATATTAGCT?

Unaccounted factors: base quality

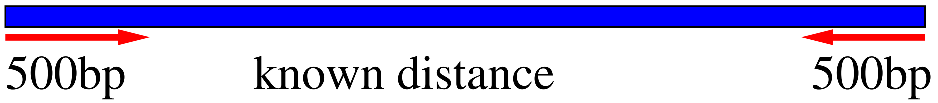
- Reads typically accompanied by **base qualities**
  How likely is this base correct?
- Base with quality $q \Rightarrow$ probability of error $10^{-q/10}$
  i.e. base with $q > 40$ is correct for 99.99%

Example of Sanger sequencing result (trace):



Shortest Common Superstring: Simplifying Factors

Additional information: pair-end reads



plasmid 2–10 kB
cosmid 40 kB

Simplification: we do not need to connect everything to one string, we connect only parts bridged by multiple reads.

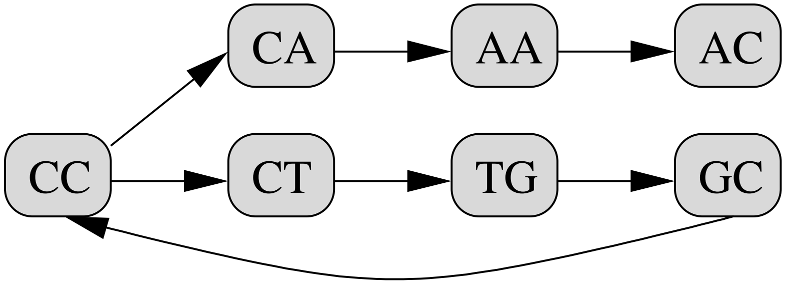
Conservative approach: sacrifice completeness for accuracy

Shortest Common Superstring: Summary

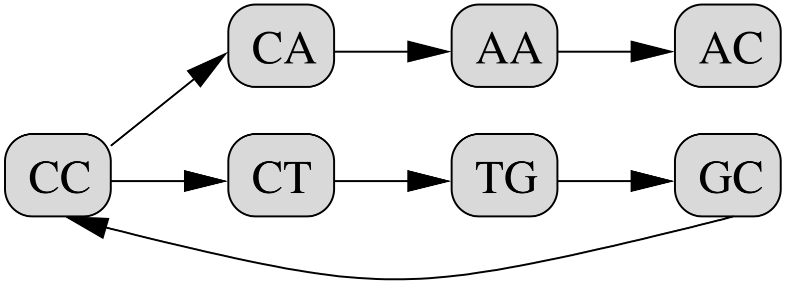
- Unrealistic formulation and difficult problem
- Perhaps theoretical problem can yield some insights into real application?
- Overlap-Layout-Consensus approach motivated by greedy algorithms (join fragments with large overlaps)

Assembling Short Reads: de Bruijn Graphs

- Split reads to overlapping windows of length k
- **de Bruijn graph** of dimension k is a **directed graph**:
  - **vertices:** substrings of length k from all reads
  - **directed edges:** connect k-mers consecutive in at least one of the reads (overlapping by $k-1$ bases)
- **Example:** $k=2$, reads: CCTGCC, GCCAAC



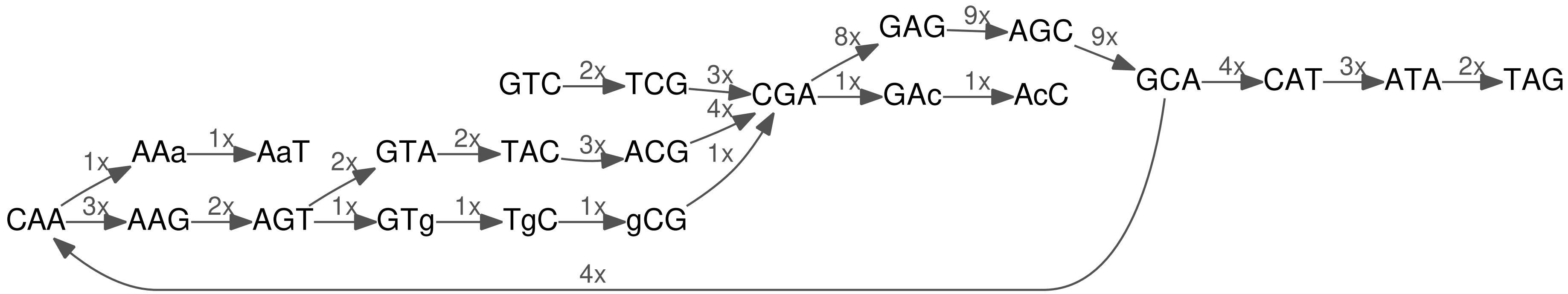
How to use de Bruijn graph for assembly?



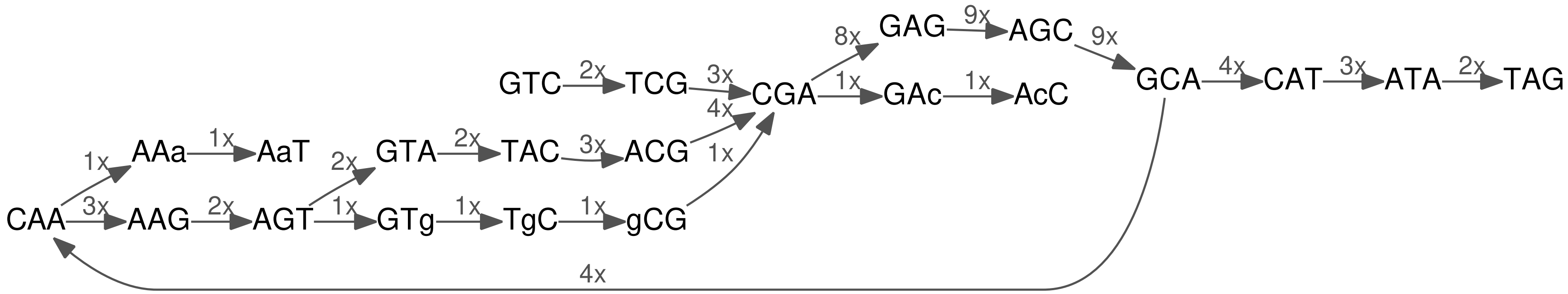
- If there was only a single chromosome and there were no ambiguous k-mers, the correct assembly would correspond to a **Eulerian path:** a path in the graph which uses each edge exactly once
- We can easily test if such a path exists and to find it in $O(m+n)$
- In general, assembly will correspond to a set of **walks in the de Bruijn graph** covering most edges

Example: reads and their de Bruijn graph

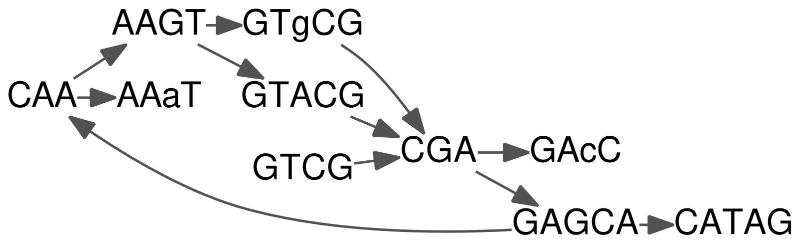
```
GTCGAGCAAGTACGAGCATAG
 TCGAGCA      AGCATAG
    AGCAAaT   AGCATAG
GTCGAcC  GTACGAG
GTCGAGC   TACGAGC
  CGAGCAA  ACGAGCA
        AGTgCGA
      CAAGTAC
     GCAAGTA GAGCAT
   GAGCAAG   GAGCATA
          TACGAGC
```



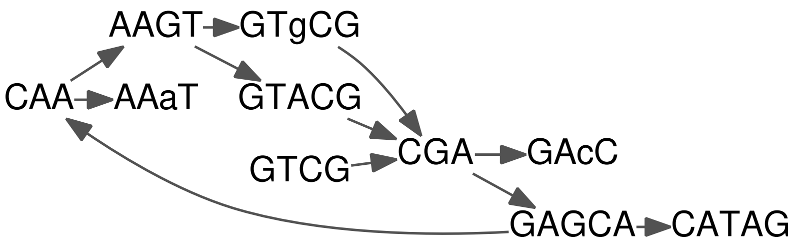
Example: simplifying de Bruijn graph



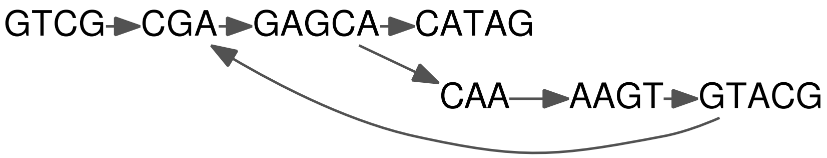
Unique paths are contracted to a single vertex



Example: removing errors from de Bruijn graph

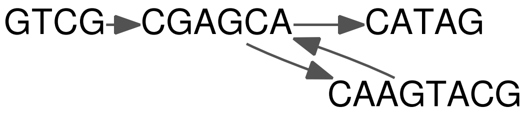


Remove tips and bubbles with low coverage



Contract unique paths again ⇒ four **contigs**

(originally $\texttt{GT\underline{CGAGCA}AGTA\underline{CGAGCA}TAG}$)



Typical Results of Assembly

- Many **short contigs** that can be further combined to **longer scaffolds** by using pair-end read information
- Some portions cannot be resolved due to **long repetitive sequences**

Example: Human chromosome 14, 88 Mbp, $70\times$ coverage (source: GAGE)

Method	Contigs	Errors	N50 corr
Velvet (basic de Bruijn)	>45000	4910	2.1 kbp
Velvet (with scaffolding)	3565	9156	27 kbp
AllPaths-LG	225	45	4.7 Mbp

N50: contigs with this length or longer contain 50% of the genome
here N50 after error correction is shown

Summary

- Sequencing is a complicated process in which bioinformatics plays an important role
- Illumina technology offers extremely low price but only short reads
- Problem of genome assembly, shortest common superstring
- de Bruijn graphs: a practical solution for short reads
- Assembled sequence may contain errors, gaps, multiple contigs
- Next lecture: How to deal with 3rd generation reads?
- Genome coverage and read size are determining factors in how fragmented assembly will be:
  - for Sanger reads: typically $7 - 10\times$ coverage
  - for NGS reads: typically $40 - 70\times$ coverage
  - for 3rd generation: $30\times$ coverage

Genome Sequencing Milestones

1976	MS2 (RNA virus) 40 kB
1988	Human genome sequencing project (15 years)
1995	bacterium H. influenzae 2 MB, shotgun (TIGR)
1996	S. cerevisiae 10 MB, BAC-by-BAC (Belgium, UK)
1998	C. elegans 100 MB, BAC-by-BAC (Wellcome Trust)
1998	Celera: human genome in three years!
2000	D. melanogaster 180 MB, shotgun (Celera, Berkeley)
2001	2x human genome 3 GB (NIH, Celera)
after 2001	mouse, rat, chicken, chimpanzee, dog,. . .
2007	Genomes of Watson and Venter (454)
2012	1000 human genomes
soon	10k vertebrate genomes, sequencing as a diagnostic tool
2021	3.5 million SARS-CoV-2 genomes