

The wonderful world of long-read genome assembly

Rayan Chikhi

Institut Pasteur & CNRS

ERGA workshop 2022

Hello

- PI, Bioinformatics algorithms lab @ Institut Pasteur
- CV: PhD@ENS Rennes, Postdoc@PSU, CNRS

Research:

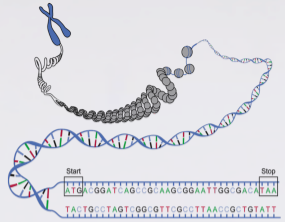
- *de novo* assembly
- k-mer methods
- metagenomics
- large-scale bioinfo



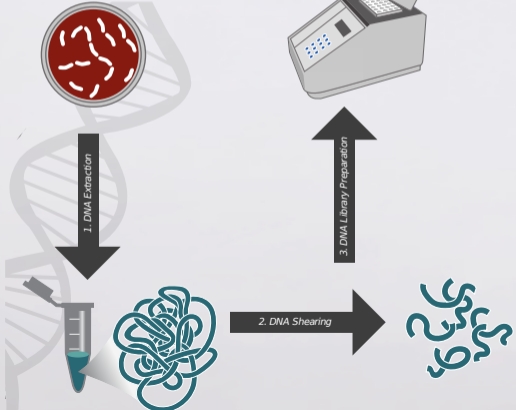
@RayanChikhi on Twitter 

<http://rayan.chikhi.name>

DNA Sequencing



Bacterial Culture



4. DNA Library Sequencing

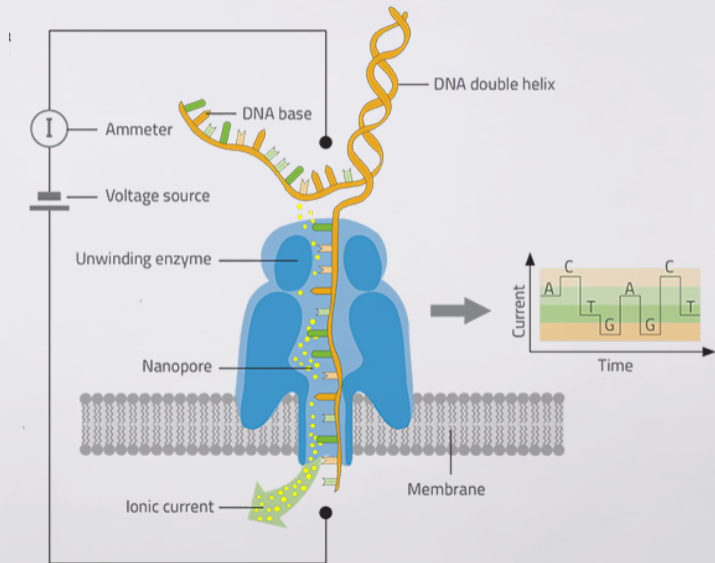


5. DNA Sequence Analysis



DNA Reads

Long-read, portable DNA sequencing (Oxford Nanopore)



Genome assembly

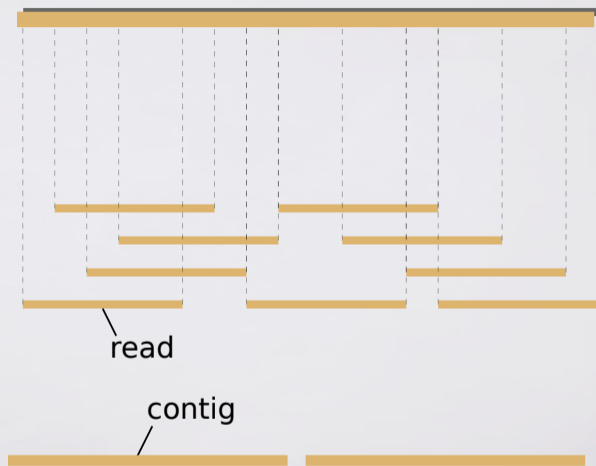
genome
(unknown)

sequenced reads:

overlapping sub-sequences, covering the genome redundantly

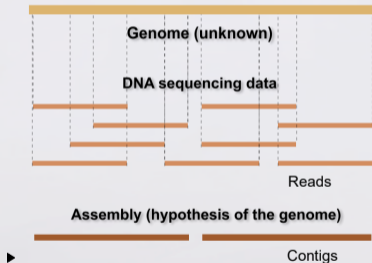


assembly
hypothesis of the genome



44 years of genome assembly

- ▶ **1977:** First complete genome assembled (phi X 174)
- ▶ **2003:** Human Genome Project completed
- ▶ **2014:** First \$1,000 genome
- ▶ **2021:** Truly completed (Telomere-2-Telomere)



▶ (Staden 1979) *“With modern fast sequencing techniques and suitable computer programs it is now possible to sequence whole genomes without the need of restriction maps.”*



The many applications of assembly

- **Reconstruct** genomes
- transcriptomes
- metagenomes
- genes
- Phylogeny of species
- Evolution of genes
- Find novel **insertions**
- **SNPs** in non-model organisms
- **cell-free DNA** struct. variants
- Pangenomics
- ...
- *Any major one I'm missing?*

Algorithmic pre-history

```
Contig Editor: -1 SRR030257.787415/2
Cons 2 Qual 0 Insert Edit Modes >> Cutoffs Undo Next Search Commands >> Settings >> Quit Help >>
161080 161090 161100 161110 161120 161130 161140 161150
+17 NC_012967 gggTaccagaacatggcgggcaaacaggaacgccgggttcacgcgcatatcgTTatggatacgcgtatcggTattcttCag
+18 _cer_sxa_0_ gggTaccagaacatggcgggcaaacaggaacgccgggt
+19 _cer_sxa_60_ gggTaccagaacatggcgggcaaacaggaacgccggg
+20 _cer_sxa_123_ gggTaccagaacatggcgggcaaacaggaacgccgg
+3663 _cer_sxa_190_ gggTaccagaacatggcgggcaaacaggaacgcc
+3664 _cer_sxa_260_ gggTaccagaacatggcgggcaaacaggaacgcc
-33401 SRR030257.8883 ACCAGAACATGGCGGCAAACAGGAACGCCGGGTGCA
-33402 SRR030257.2989 CCAGAACATGGCGGCAAACAGGAACGCCGGGTGCA
+33403 SRR030257.1128 CAGAACATGGCGGCAAACAGGAACGCCGGGTGCACG
+33404 SRR030257.7073 AGAACATGGCGGCAAACAGGAACGCCGGGTGCACGC
-33405 SRR030257.1204 AGAACATGGCGGCAAACAGGAACGCCGGGTGCACGC
-33406 SRR030257.2602 GAACATGGCGGCAAACAGGAACGCCGGGTGCACGCG
-33407 SRR030257.2767 GAACATGGCGGCAAACAGGAACGCCGGGTGCACGCG
-33408 SRR030257.1413 AACATGGCGGCAAACAGGAACGCCGGGTGCACGCGC
+33409 SRR030257.1755 ACATGGCGGCAAACAGGAACGCCGGGTGCACGCGCA
+33410 SRR030257.1463 ACATGGCGGCAAACAGGAACGCCGGGTGCACGCGCA
+33411 SRR030257.1623 CATGGCGGCAAACAGGAACGCCGGGTGCACGCGCAT
+33412 SRR030257.6602 TGGCGGCAAACAGGAACGCCGGGTGG
-33413 SRR030257.1821 GGGCGGCAAACAGGAACGCCGGGTGCACGC
-33414 SRR030257.2932 GGGCGGCAAACAGGAACGCCGGG
+33415 SRR030257.5986 GGC GGCAAACAGGAACGCCGGGTGG
-33416 SRR030257.3729 GGC GGCAAACAGGAACGCCGGGTGCACGCGCATATC
+33417 SRR030257.3423 GCGGCAAACAGGAACGCCGGGTGCACGCGCATATCG
+33418 SRR030257.2482 GCGGCAAACAGGAACGCCGGGTGCACGCGCATATCG
-33419 SRR030257.1401 CGGCAAACAGGAACGCCGGGTGCACGCGCATATCG
-33420 SRR030257.3565 CGGCAAACAGGAACGCCGGGTGCACGCGCATATCGT
+33421 SRR030257.1125 CGGCAAACAGGAACGCCGGGTGCACGCGCATATCGT
-33422 SRR030257.3529 GGCAAACAGGAACGCCGGGTGCACGCGCATATCGTT
> CONSENSUS -*- GGGTACCAGAACATGGCGGCAAACAGGAACGCCGGGTTCACGCGCATATCGTTATGGATACGCGTATCGGTATTCTTCAG
Tag type:SVEC Direction:- Comment:'''
```


Algorithmic pre-history



1. Assembly using strings

- ▶ Shortest Common Superstring (Kececioğlu, Myers 1993)
- ▶ Greedy algorithms (CAP3 from Huang, Madan 1999)

Algorithmic pre-history



1. Assembly using strings

- ▶ Shortest Common Superstring (Kececioğlu, Myers 1993)
- ▶ Greedy algorithms (CAP3 from Huang, Madan 1999)

2. Assembly using graphs: string graphs and de Bruijn Graphs (both from DIMACS'94)

A History of DNA Sequence Assembly, G. Myers, 2016

dBGs widely used across genomics (SPAdes: 13,000 citations; Trinity: 12,000 citations)

Algorithmic pre-history



1. Assembly using strings

- ▶ Shortest Common Superstring (Kececioğlu, Myers 1993)
- ▶ Greedy algorithms (CAP3 from Huang, Madan 1999)

2. Assembly using graphs: string graphs and de Bruijn Graphs (both from DIMACS'94)

A History of DNA Sequence Assembly, G. Myers, 2016

dBGs widely used across genomics (SPAdes: 13,000 citations; Trinity: 12,000 citations)

Modern genome assembly: graphs

1. Construct a graph
2. Nodes are reads (or k -mers)
3. Edges are overlaps



Theory will say..

[Nagarajan 09]

4. Return a path of *minimal length* that traverses **each node at least once**.

Assembly theory is somewhat unhelpful

Genome assembly is **linear-time** solvable.

[Pevzner *et al*, 2001]

Genome assembly is **NP-hard**.

[Medvedev, Brudno 2007]

If all **repeats** are **longer** than reads,
Genome assembly is **polynomial**. (!)

[Nagarajan, Pop 2009]

If all repeats are either **shorter** than reads, or are **spanned** by reads,
Genome assembly is **polynomial**, *and* with a **unique** solution.

[Nagarajan, Pop 2009]
[Bresler, Bresler, Tse 2013]



Yet, in practice..

- **Illumina** data: none of the theories apply
- Because graph is often disconnected
- So, can't frame the problem as finding a single path
- Contigs = all the unambiguous paths

- **Long reads**: “theory meets practice” appears possible

[Kamath *et al*, 2017]

Either way:

- \geq **100s of GB** of input data (for eukaryotes)
- **days/months** of CPU time
human: 1 CPU-month, 200 GB RAM (w/ dbg2)

Modeling biological problems in computer science: a case study in genome assembly

Paul Medvedev 

Briefings in Bioinformatics, bby003, <https://doi.org/10.1093/bib/bby003>

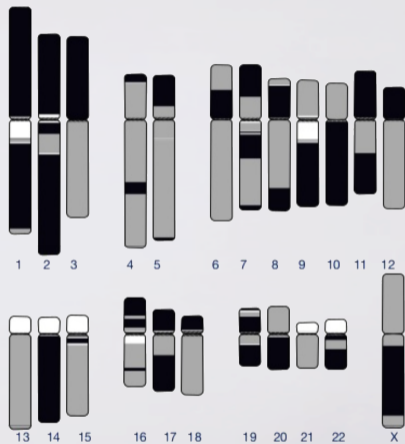
Published: 30 January 2018 **Article history** ▼

The human genome is challenging to assemble



ref28 NG50 contig 0.5 Mbp

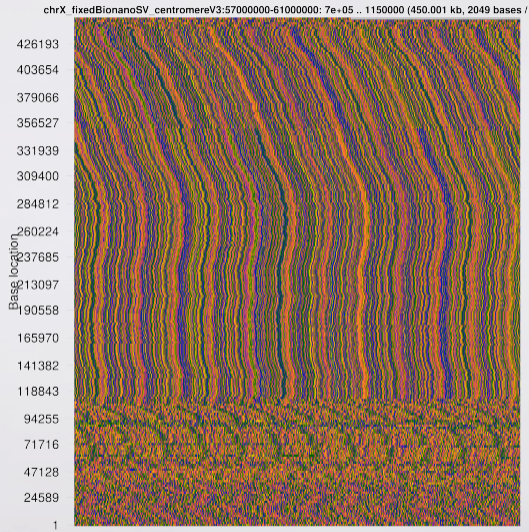
Human genome
in 2001
<-
in 2019
->



CHM13 NG50 contig 75 Mbp (70x PacBio + 35x UL ONT)

credit: A. Phillippy

Why? centromeres, but not only



credit: D. Eccles, https://twitter.com/gringene_bio/status/1102121757828210688

Consensus caveats

Suppose a diploid organism with 2 haplotypes:

```
..AGCCTGAGTTC..  
..AGCATGATTTC..
```

Assembly usually results in a single consensus:

```
..AGCCTGATTTC..
```

Haplotype separation at chromosome scale is recent.

Letter | [Open Access](#) | Published: 07 December 2020

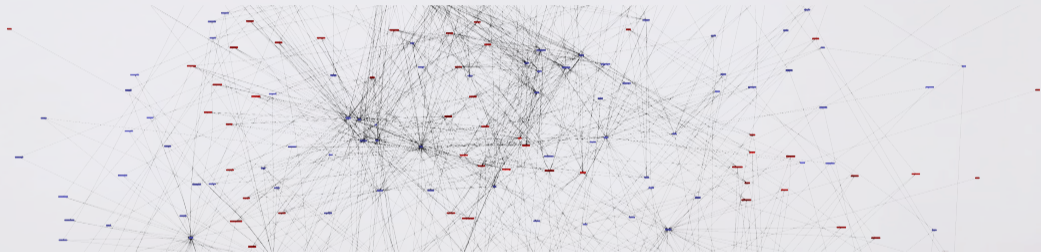
Chromosome-scale, haplotype-resolved assembly of human genomes

Letter | [Open Access](#) | Published: 07 December 2020

Fully phased human genome assembly without parental data using single-cell strand sequencing and long reads

Genome assembly software is complex

- Coding:
 - ▶ PhD
 - ▶ or a team of engineers (1-2 years)
- Several not-so-independent components



- Heuristics everywhere

*“A genome assembly is like a good sausage,
it is best to ignore how it was made”*

(apocryphal) S. Gnerre, ALLPATHS assembler

Genome assembly landscape, in 2022

- **Short reads:** don't, except for Hi-C
- **Long reads:** Oxford Nanopore, **PacBio HiFi**, PacBio CLR

Genome assembly landscape, in 2022

- **Short reads:** don't, except for Hi-C
- **Long reads:** Oxford Nanopore, **PacBio HiFi**, PacBio CLR



(Img: M. Watson)

Genome assembly landscape, in 2022

- **Short reads:** don't, except for Hi-C
- **Long reads:** Oxford Nanopore, **PacBio HiFi**, PacBio CLR



(Img: M. Watson)

But: short-read **methods** are making a come-back with long reads. de Bruijn graphs in Flye, rust-mdbg, LJA, Verkko & *k*-mer assembly validation.

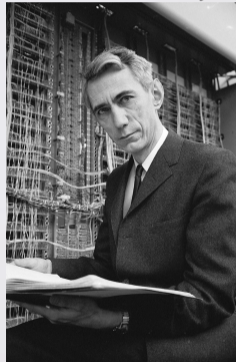
k -mers

k -mer: any sequence of length k

N.G. de Bruijn (1946),
de Bruijn sequences ¹



C. Shannon (1948),
information theory ²



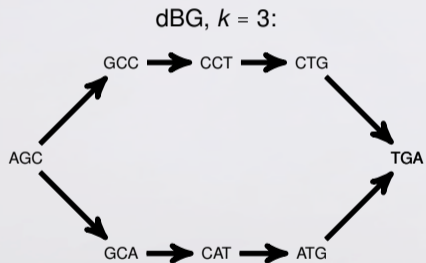
¹construct shortest sentence containing all k -mers exactly once

²predict future data given past data, where past = last seen k -mer

de Bruijn graph

A **de Bruijn** graph for a fixed integer k :

1. **Nodes** = all k -mers (substrings of length k) in the reads
2. **Edges** = all exact overlaps of length exactly $(k - 1)$



Of those reads:

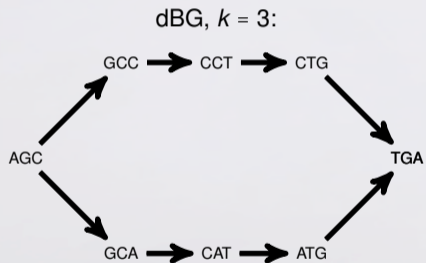
AGCCTGA

AGCATGA

de Bruijn graph

A **de Bruijn** graph for a fixed integer k :

1. **Nodes** = all k -mers (substrings of length k) in the reads
2. **Edges** = all exact overlaps of length exactly $(k - 1)$



Of those reads:

AGC**CT**GA

AGC**AT**GA

dBG of *E. coli* reads, $k=71$:

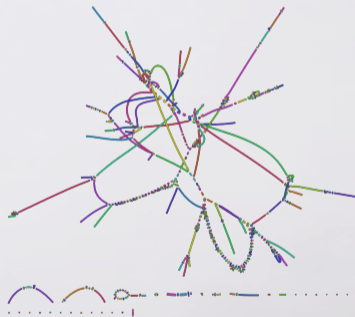


Fig: Bandage

Assembly graph visualization: Bandage and Bandage-NG

Bandage - /media/ryan/Data/Bandage_demo/O7_NW1_metagenome/NW1_LastGraph

File Tools View Help

De Bruijn graph information

Nodes: 51,639
Edges: 65,832
Total length: 18,712,634

Graph drawing

Scope: Entire graph
Style: Single Double
Draw graph

Graph display

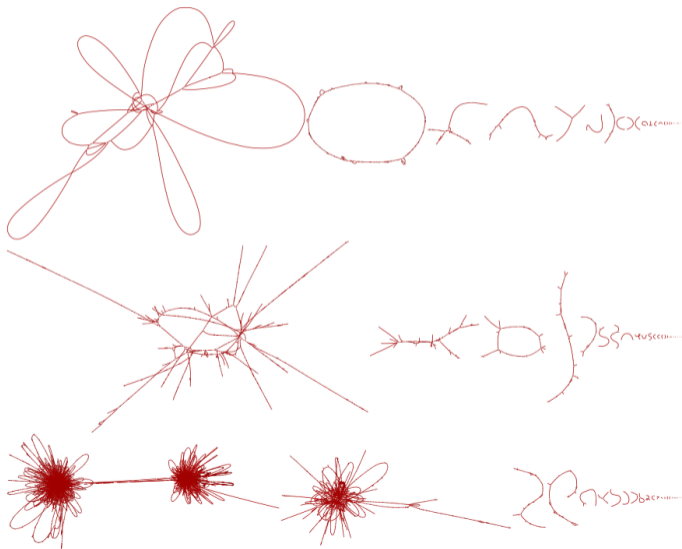
Zoom: 2.6%
Uniform colour

Node labels

Custom Number
 Length Coverage
Font Text outline

BLAST

Create/view BLAST search



Find nodes

Node(s):
Find node(s)







The screenshot shows the top portion of a Science journal article page. At the top left is the Science logo. Below it is a breadcrumb trail: HOME > SCIENCE > VOL. 376, NO. 6588 > THE COMPLETE SEQUENCE OF A HUMAN GENOME. A lock icon is followed by the text 'SPECIAL ISSUE RESEARCH ARTICLE | HUMAN GENOMICS'. The main title 'The complete sequence of a human genome' is prominently displayed. Below the title, the authors are listed: SERGEY NURK, SERGEY KOREN, ARANG RHIE, MIKKO RAUTIAINEN, ANDREY V. BZIKADZE, ALLA MIKHEENKO, MITCHELL R. VOLLGER, NICOLAS ALTEMOSE, LEV URALSKY, and ADAM M. PHILLIPPY, with a '+91 authors' button and a link to 'Authors Info & Affiliations'. At the bottom, the journal information is provided: SCIENCE · 31 Mar 2022 · Vol 376, Issue 6588 · pp. 44-53 · DOI: 10.1126/science.abj6987.




Science

HOME > SCIENCE > VOL. 376, NO. 6588 > THE COMPLETE SEQUENCE OF A HUMAN GENOME

🔒 | SPECIAL ISSUE RESEARCH ARTICLE | HUMAN GENOMICS

The complete sequence of a human genome

SERGEY NURK , SERGEY KOREN , ARANG RHIE , MIKKO RAUTIAINEN , ANDREY V. BZIKADZE , ALLA MIKHEENKO, MITCHELL R. VOLLGER ,

NICOLAS ALTEMOSE , LEV URALSKY , [...] ADAM M. PHILLIPPY  [+91 authors](#) [Authors Info & Affiliations](#)

SCIENCE · 31 Mar 2022 · Vol 376, Issue 6588 · pp. 44-53 · DOI: 10.1126/science.abj6987

- First gap-less human genome
- 154 Mb contig N50 (retiring that metric)
- Added \approx 200 Mbp compared to GRCh38 (centromeres mostly)
- CHM13: haploid, no Y chromosome
- 30x HiFi (HiCanu) + 120x ONT ultralong

Han1

A second complete human genome. (* not peer reviewed)

The first gapless, reference-quality, fully annotated genome from a Southern Han Chinese individual
(KH. Chao, A. Zimin, M. Pertea, S. Salzberg)

Only 4 authors!

How?:

- 39x HiFi (hifiasm), 35x ONT ultralong (Flye, discarded)
- scaffolded against CHM13 (with MaSuRCA)
- Semi-manual gap-closing
- JASPER for HiFi polishing

Automatic near-T2T tools

- **Verkko**

Telomere-to-telomere assembly of diploid chromosomes

HiFi + UL + Hi-C, Strand-seq, trio

- **LJA**

HiFi only

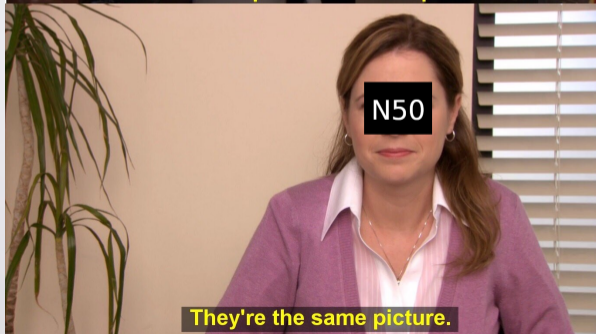
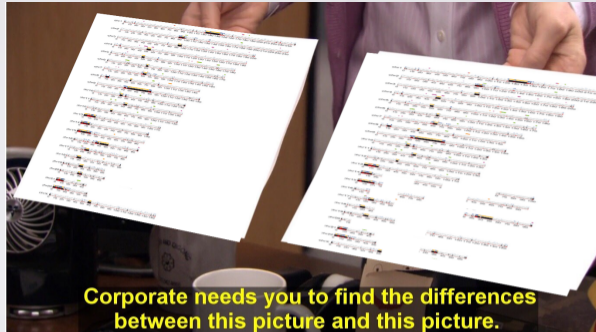
- **Hifiasm**

HiFi + Hi-C, trio

Relevance of assembly metrics

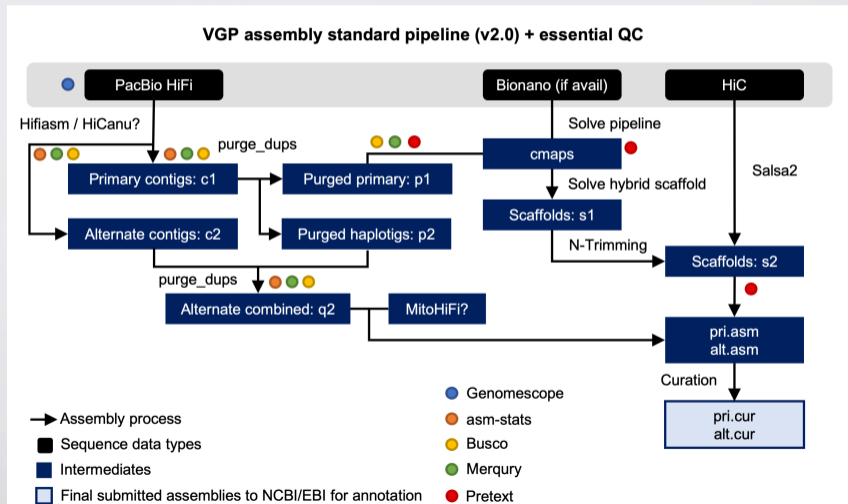
- N50
- Completeness / BUSCO

Is N50 relevant in the T2T era? less and less. BUSCO? Very important for spotting duplications



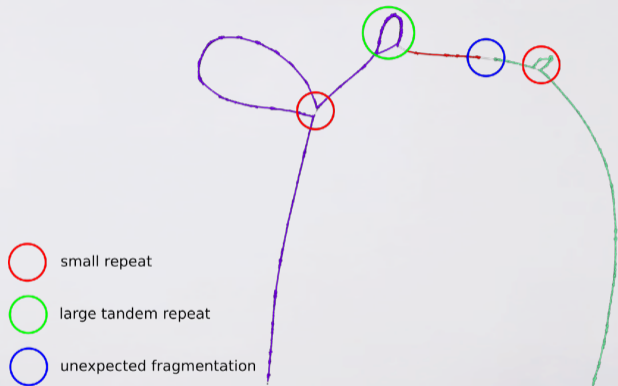
VGP, ERGA, DToL, AfricaBP

- Sequence as much as possible
- Assemble it
- Challenges ahead (repetitiveness, ploidy, ...samples collection)



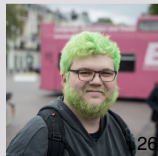
Is bacterial genome assembly solved?

In 2019, 30% of PacBio bacterial assemblies were fragmented



P. Marijon *et al*, Bioinformatics 2019

<https://gitlab.inria.fr/pmarijon/knot>



Viral assembly

Serratus:

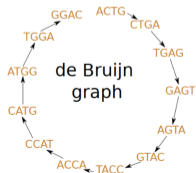
- 5M RNA-seqs aligned (10 PB)
50k assemblies, 28,000 vCPUs on AWS in a weekend
- Discovery of a new coronavirus species
- 10x expansion of RNA viruses species
- It remains challenging to assemble viruses from metagenomes
- *Edgar et al, 2022*



2) Fast (meta)genome assembly with accurate long reads



PacBio HiFi reads
(~1% error rate)



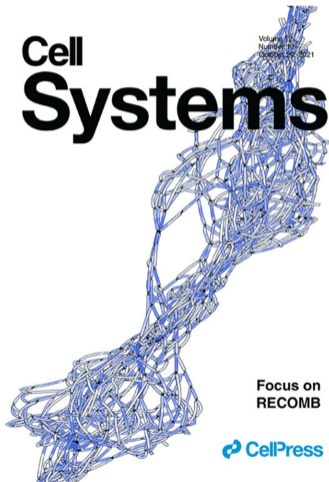
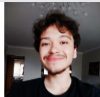
- Minimizer-space de Bruijn graphs
- Human genome assembled in 10 minutes / 10 GB RAM
- Pangenome of 661k bacteria



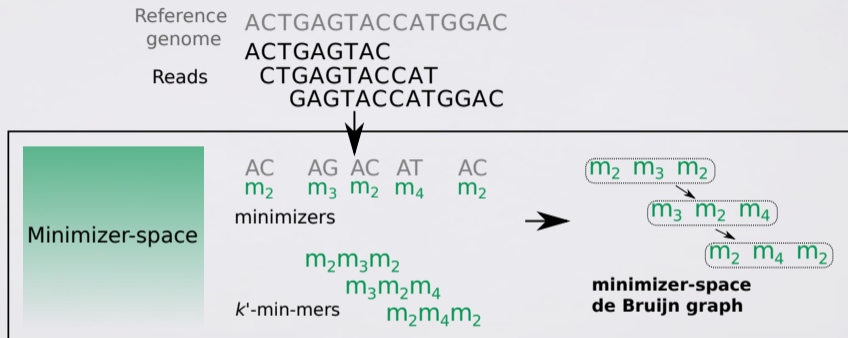
“Tricks”: change of alphabet (ACGT->minimizers)
discards 99% of the bases until last step

Collaborators:

B. Ekim,
B. Berger
(MIT)



Minimizer-space de Bruijn graph



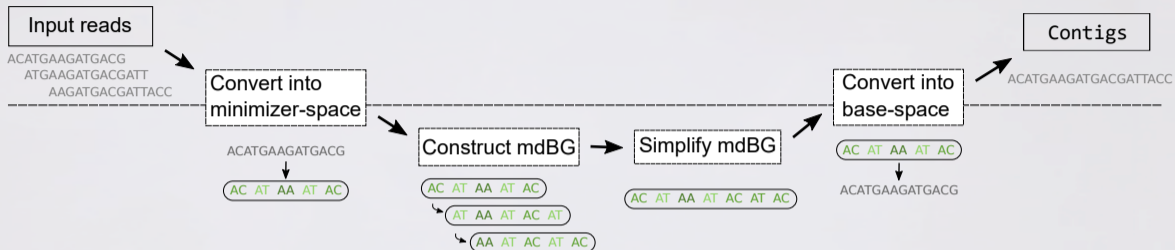
A **minimizer-space de Bruijn graph** is a **de Bruijn graph** over the **minimizer alphabet**.

Nodes = k-min-mers,

Edges = exact overlaps between k-1 minimizers

Applied to whole-genome *de novo* assembly

From accurate HiFi (< 1% error-rate) reads



Whole human PacBio HiFi (HG002) 50x coverage:

Tool name	Peregrine	hifiasm	rust-mdbg
Wall-clock time	14h8m	58h41m	10m23s
Memory usage	188 GB	195 GB	10 GB
# contigs	8109	431	805
NG50 (Mbp)	18.2	88.0	16.1
Genome fraction	97.0%	94.2%	95.5%

Conclusion



Take-aways:

- Genome assembly, nowadays done with long reads
- One of the two fundamental “sequence bioinformatics” problem along with alignment
- Data structures & algorithms play a huge role in methods
- Also, large amounts of Python/C++/Rust for making tools

An open problem:

- Highest-quality assembly from single-sample long reads only

Thank you! Any questions?

Acknowledgements for this talk material: Pierre Marijon, Jean-Stéphane Varré, Adam Phillippy, Antoine Limasset, Camille Marchet, Brian Bushnell, Sergey Nurk, Marco Previtali, Paul Medvedev, Shaun Jackman, Guillaume Rizk, Ryan Wick, David Eccles, Mick Watson

Sequence Bioinformatics @ Institut Pasteur



Y. Dufresne, R. Vicedomini, T. Lemane, C. Duitama, L. Blassel, F. Andrace

Funding: EC H2020, ANR Inception, ANR Prairie, ANR Transipedia, ANR SeqDigger



Lex Nederbragt

@lexnederbragt

En réponse à [@ctitusbrown](#)

“Finding your way in life is like finding the genome in a De Bruijn graph: it is very easy to find **a** path, very hard to find **the** path”.