Reviews in Quantitative Biology

Applications of kmer analysis in quantitative biology

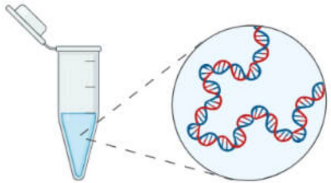
Sina Majidian

November 18th, 2022

Outline

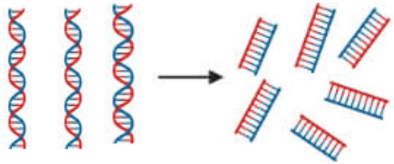
- **Introduction**
  - DNA sequencing
  - Kmer definition
  - Minimizer definition
- **Applications**
  - Metagenomics
  - Genome assembly
  - Kmer counting
- **Discussion**

DNA Sequencing



DNA extraction

GGCGTCTATATCTCGGCTCTAGGCCCTCATTTTTT



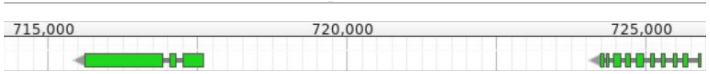
Amplification

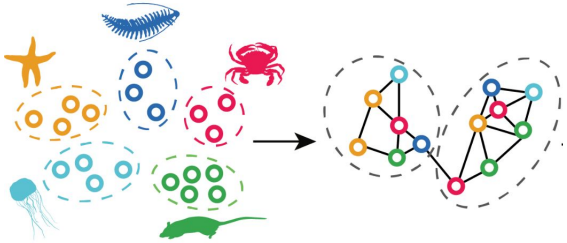
GGCGTCTATATCTCGGCTCTAGGCCCTCATTTTTT
GGCGTCTATATCTCGGCTCTAGGCCCTCATTTTTT
GGCGTCTATATCTCGGCTCTAGGCCCTCATTTTTT
GGCGTCTATATCTCGGCTCTAGGCCCTCATTTTTT

Sequencing

GGCGTCTA TATCTCGG CTCTAGGCCCTC ATTTTTT
GGC GTCTATAT CTCGGCTCTAGGCCCTCA TTTTTT
GGCGTC TATATCT CGGCTCTAGGCCCT CATTTTTT
GGCGTCTAT ATCTCGGCTCTAG GCCCTCA TTTTTT

Genomics pipelines





1) De novo genome assembly

2) Gene annotation

3) Orthology inference

4) Read alignment

5) Variant calling & genotyping

6) Metagenomics

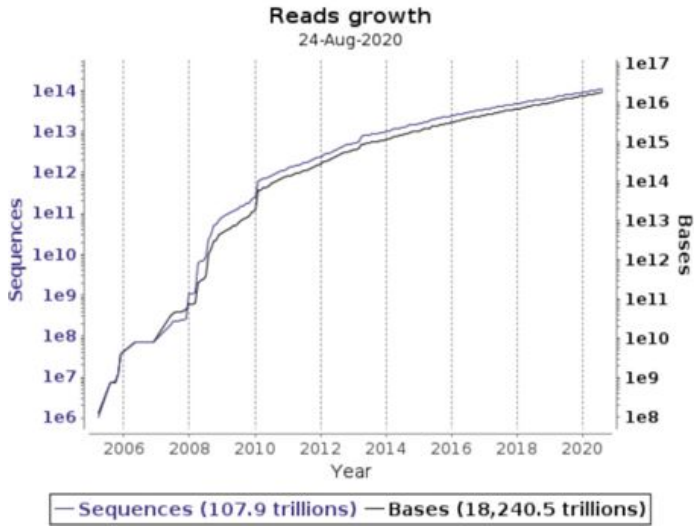
Genome assembly: importance of scalable methods

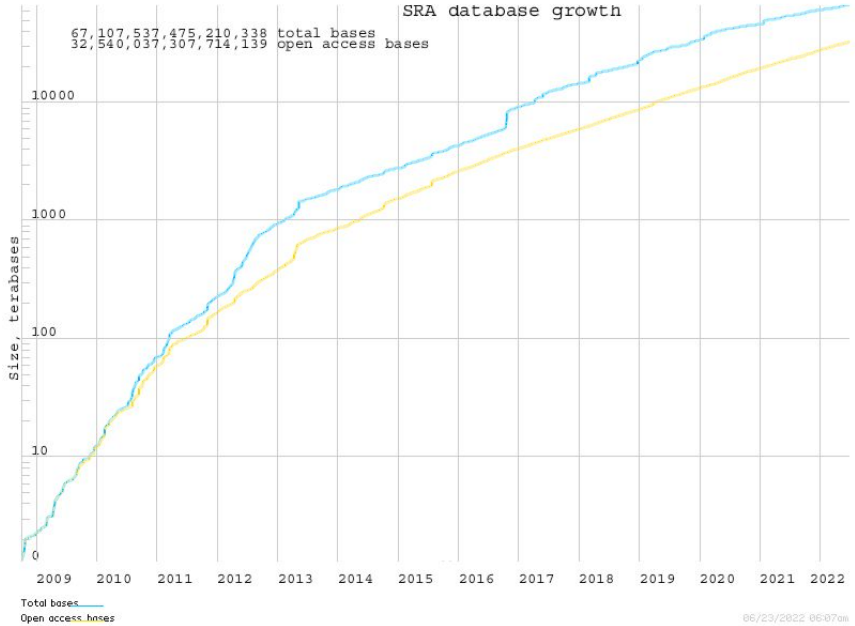
	D. mel 100× real HiFi reads				Human real HiFi reads		
Tool	Peregrine	HiCanu	Hifiasm	Rust-mdbg	Peregrine	Hifiasm	Rust-mdbg
Time	40 min 11s	7 h 43min	5 h 17min	**1 min 9 s**	14 h 8 min	58 h 41min	**10 min 23 s**
Memory	12 GB	12 GB	21 GB	**1.5 GB**	188 GB	195 GB	**10 GB**

We need efficient methods in terms of **memory**, **speed** and **storage**.

Minimizer-space de Bruijn graphs: Whole-genome assembly of long reads in minutes on a personal computer, Cell Systems, 2021.

Fast growing genomics data

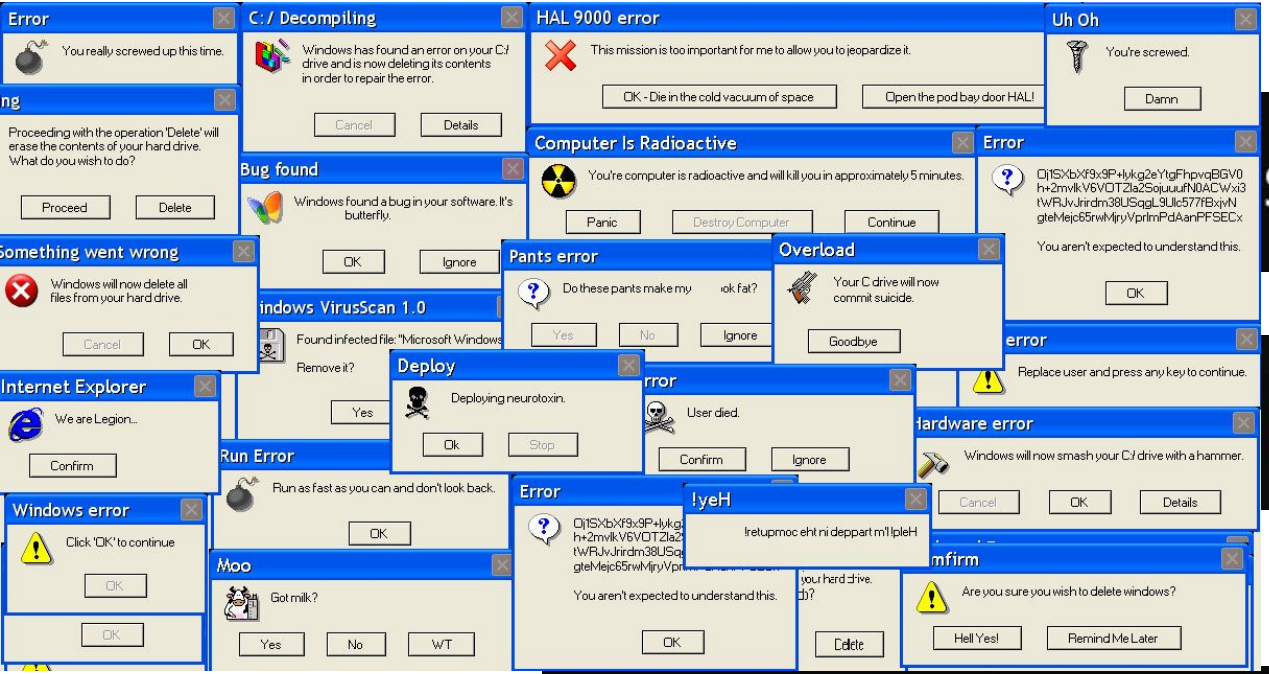




The European Nucleotide Archive in 2020, Nucleic Acids Research

ncbi.nlm.nih.gov/sra/docs/sragrowth/

No Space Left on Device



Segmentation fault

Segmentation fault

Core dumped

ERROR ERROR ERROR

Too many open files

Basics on K-mer & Minimizers

K-mer definition

- a contiguous substring of length k.

- alphabet= {A,C,G,T}
- string (sequence)= GATTACA

- Example: k=2, 2mers

- GA, AT, TT, TA, AC, CA
- Number of 2-mers = 6

GATTACA

GATTACA

GATTACA

GATTACA

GATTACA

GATTACA

https://www.youtube.com/c/RobEdwardsVideos/videos
http://www.langmead-lab.org/teaching-materials/

K-mer stats

For the same sequence GATTACA (L=7)

k=3: GAT, ATT, TTA, TAC, ACA (5 kmers)

k=4: GATT, ATTA, TTAC, TACA (4 kmers)

GATTACA

GATTACA

GATTACA

GATTACA

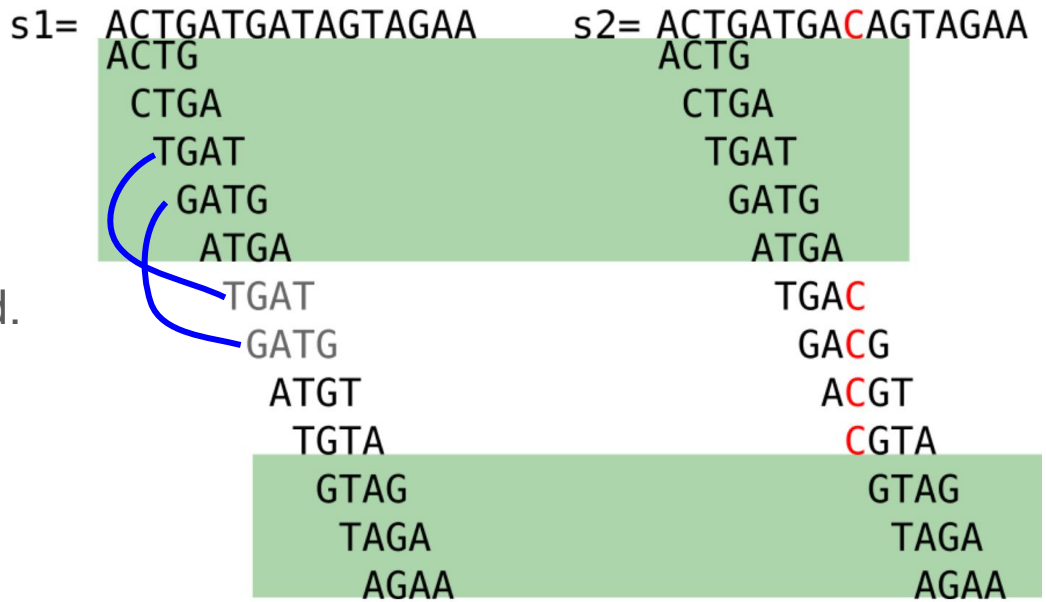
GATTACA

- How many 3-mers do exist for a sequence of length L=100?
  - Number of 3-mers = 100-3+1 = 98 (L-k+1)
- How many different 3-mers do exist?
  - 4^3=64 (4^k)

Sequence comparison

Two similar sequences with one base error:

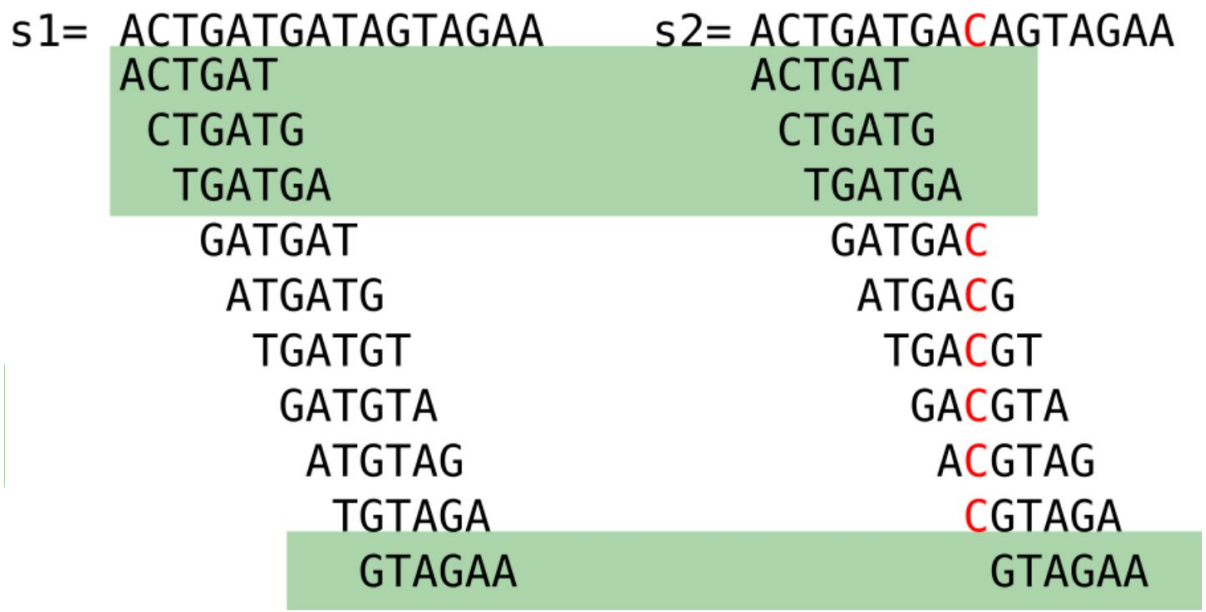
- Many shared 4-mers (green)
- Neighbors of C/T are not shared.
- Repeated kmers



kamimrcht.github.io/webpage/strobemers.html

Impact of k on specificity

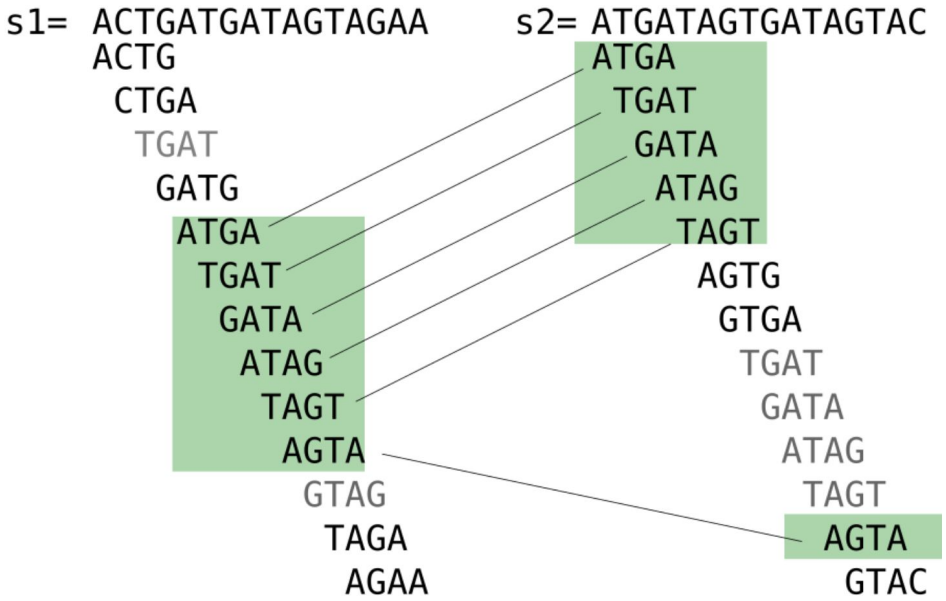
By increasing k, we can keep the differences.



Impact of k on specificity

- Two not so similar sequences:
- Still a lot of shared 4-mers

- Number 5-mers= 4^5=1024
- Every 1k, one shared kmer by chance. (if distributed uniformly)



K-mer stats

- Billions (10^9) of k-mers exist in a metagenomic datasets,
- resulting in computational challenges

Dataset	k	Total	Distinct
Cow rumen	28	7.39×10^9	5.09×10^9
	55	4.69×10^9	3.73×10^9
Marine	28	7.26×10^9	3.73×10^9
	55	4.57×10^9	3.04×10^9

- Each k-mer contains little information compared to the nearest one.
- Idea: sampling (also referred to as “sketching”)

GATTACA

GATTACA

GATTACA

GATTACA

GATTACA

Table from “Compact and evenly distributed k-mer binning for genomic sequences.” Bioinformatics, 2021

Minimizer

- The smallest kmer in a window.
- Aphabetical order: AA <AT
- Window of size 5, 3-mers.

1st window GATTA : ATT

2nd window ATTAC : ATT

3rd window TTACA : ACA

Minimizers for GATTACA are ATT, ACA.

(2 fingerprints instead of 5 kmers)

Similar sequences ~ shared minimizers.

GATTA

GATTA

GATTA

ATTAC

ATTAC

ATTAC

TTACA

TTACA

TTACA

GATTACA

GATTACA 1st window

GATTACA 2nd window

GATTACA 3rd window

Reducing storage requirements for biological sequence comparison. Bioinformatics, 2004.

Different ordering

1. Alphabetical order: AA <TT

2. Defining a function on a string

A=1, C=2, G=3, T=4

TAAT = 4114
CTTT = 2444

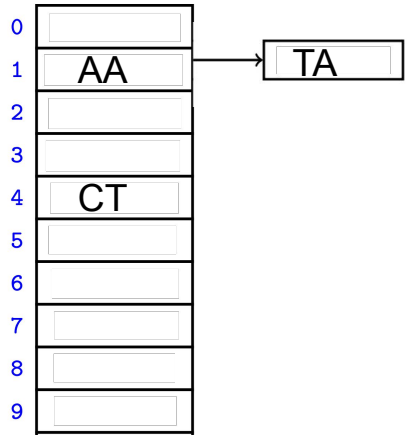
TAAT > CTTT

3. Hash function
h(x) = x mod 11
“remainder of a division by 10”

AA=11 -> {AA:1}

CT: 24 =2*10+ 4

TA: 41 =4*10+ 1

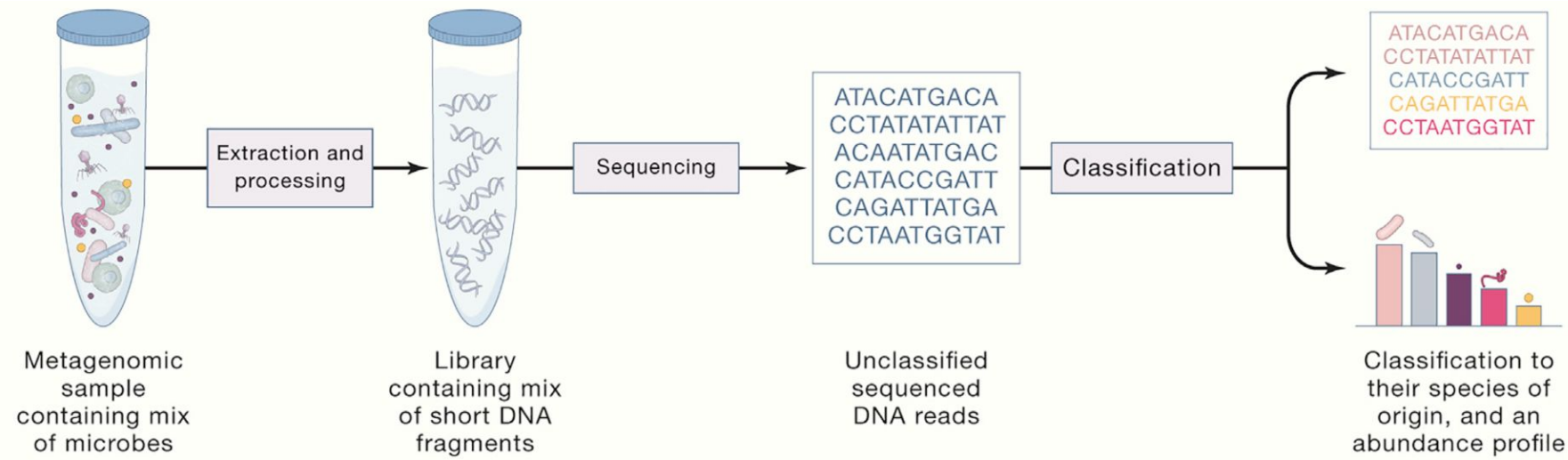


Applications

1- Metagenomics

Metagenomics

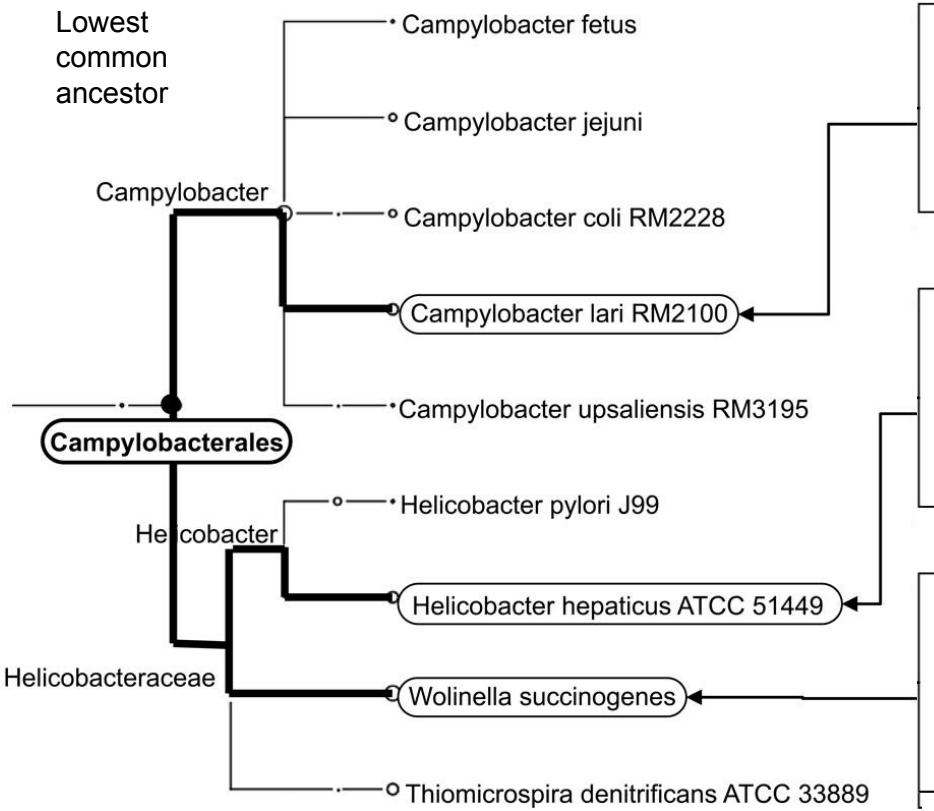
Metagenomics: the study of genomic sequences obtained directly from an environment (gut microbiome, wastewater surveillance, ..)



Benchmarking Metagenomics Tools for Taxonomic Classification, Cell 2019

MEGAN: exploring taxonomical content

BLASTX read1 against NCBI-NR




```
>gi|57241447|ref|ZP_00369393.1| flagellar motor switch protein FliG
                [Campylobacter lari RM2100]
 Score = 33.9 bits (76), Expect = 1.8
 Identities = 13/26 (50%), Positives = 19/26 (73%)

Query: 79  LMFVFDDLATVEENGIREIINRADKK 2
           LMF FDD++ +  N IRE++  ADK+
Sbjct: 243 LMFTFDDISQLSTNAIREVLKAADKR 268
```

1st match for read1

```
>gi|32262158|gb|AAP77207.1| flagellar motor switch protein FliG
              [Helicobacter hepaticus ATCC 51449]
 Score = 33.5 bits (75), Expect = 2.4
 Identities = 13/26 (50%), Positives = 20/26 (76%)

Query: 79  LMFVFDDLATVEENGIREIINRADKK 2
           +MF F+D++ ++ N IREI+  ADKK
Sbjct: 244 MMFTFEDISKLDNNAIREILKIADKK 269
```

2nd match for read1

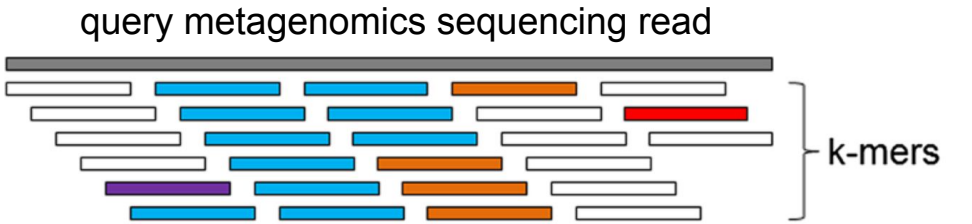
```
>gi|34484004|emb|CAE11000.1| FLAGELLAR MOTOR SWITCH PROTEIN FLIG
           [Wolinella succinogenes]
 Score = 32.7 bits (73), Expect = 4.1
 Identities = 13/26 (50%), Positives = 19/26 (73%)

Query: 79  LMFVFDDLATVEENGIREIINRADKK 2
           +MF F+D+  ++ N IREI+  ADKK
Sbjct: 242 MMFTFEDIEKLDNNAIREILKVADKK 267
```

3rd match for read1

MEGAN analysis of metagenomic data, Genome research, 2007 (3k citations)

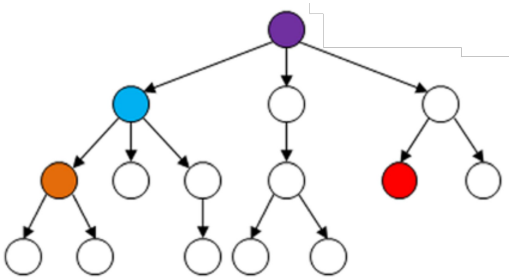
Kraken: assigning taxonomic labels

1. Extract k-mers from the input sequencing read (query)



2. Use precomputed database, extract the taxonomic ID of 31-mers

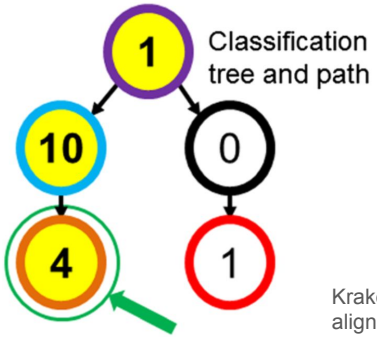
31-mers	Taxonomy ID
GACAAGGAGGCTATTGAGAAGAATATGGAAC	Prevotella intermedia
TAGCAGGCAGGTTCACCGGGGTGGCGGTTTT	Rothia mucilaginosa
ATATGATTTTGAGCGAAAAATGAATAAAATT	Streptococcus pneumoniae



3. Count # kmers assigned to each taxonomic level

4. Find the path with maximum sum

5. Report the leaf as the taxonomic label



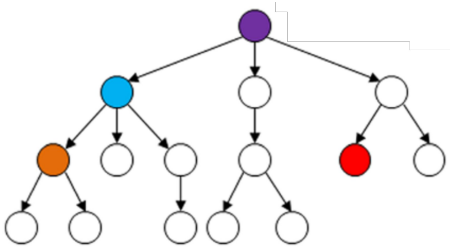
Kraken: ultrafast metagenomic sequence classification using exact alignments, Genome biology, 2014 (3k citations)

Kraken: kmer-LCA table

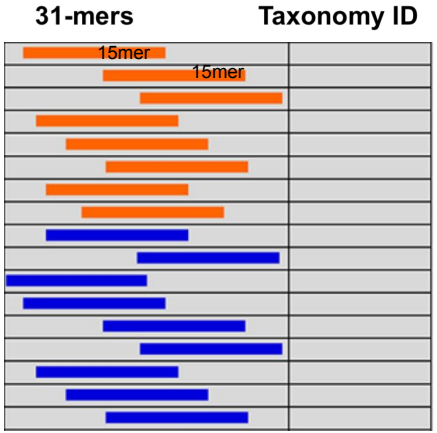
- Pre-computing the kmer-LCA database
- Extract all 31-mers from microbial genome in NCBI RefSeq

31-mers	Taxonomy ID
GACAAGGAGGCTATTGAGAAGAATATGGAAC	Prevotella intermedia
TAGCAGGCAGGTTCACCGGGGTGGCGGTTTT	Rothia mucilaginosa
ATATGATTTTGAGCGAAAAATGAATAAAATT	Streptococcus pneumoniae

31-mers



- To speed up kmer search in the table
- Use 15mer minimizer for grouping similar 31mer



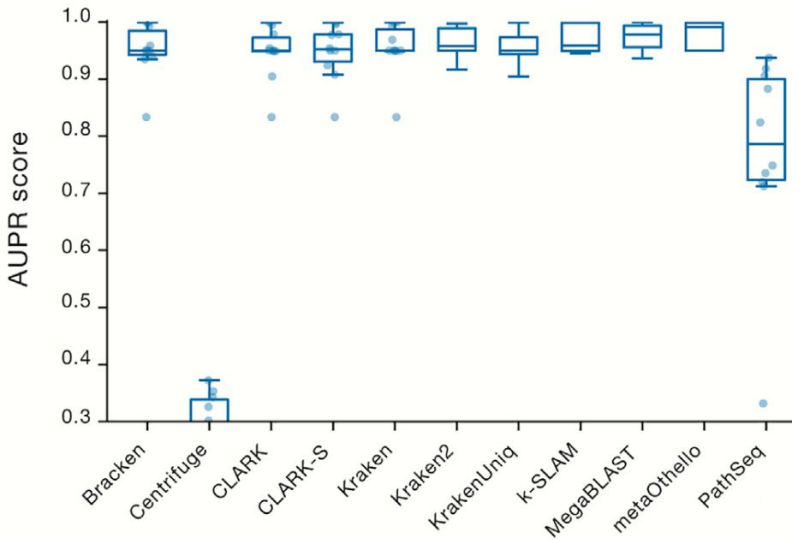
- Fill the table with 31-mers of shared minimizer near each other
- For a 31-mer query, start searching within its minimizer

Kraken: ultrafast metagenomic sequence classification using exact alignments, Genome biology, 2014 (3k citations)

Kraken2 results

Classification accuracy and speed comparison on a simulated dataset

Classifier	Memory Required	Time Required
Bracken	<1 Gb	<1 min
Centrifuge	20 Gb	7 min
CLARK	80 Gb	2 min
CLARK-S	170 Gb	40 min
Kraken	190 Gb	1 min
Kraken2	36 Gb	1 min
KrakenUniq	200 Gb	1 min
k-SLAM	130 Gb	2 h
MegaBLAST	61 Gb	4 h
metaOthello	30 Gb	1 min
PathSeq	140 Gb	5 min



Improved metagenomic analysis with Kraken 2, 2019.

Article | Open Access | Published: 11 July 2019

Strain-level metagenomic assignment and compositional estimation for long reads with MetaMaps

Alexander T. Dilthey, Chirag Jain, Sergey Koren & Adam M. Phillippy

Nature Communications **10**, Article number: 3066 (2019) | Cite this article

15k Accesses | **49** Citations | **68** Altmetric | Metrics

Research | Open Access | Published: 31 October 2022

Sketching and sampling approaches for fast and accurate long read classification

Arun Das & Michael C. Schatz

BMC Bioinformatics **23**, Article number: 452 (2022) | Cite this article

19 Accesses | **3** Altmetric | Metrics

Comparing minimizer with MinHash & ..

Short Report | Open Access | Published: 28 November 2019

Improved metagenomic analysis with Kraken 2

Derrick E. Wood, Jennifer Lu & Ben Langmead

Genome Biology **20**, Article number: 257 (2019) | Cite this article

52k Accesses | **1081** Citations | **91** Altmetric | Metrics

PeerJ

View 3 tweets

Related research

‹ BIOINFORMATICS AND GENOMICS

Deconvolute individual genomes from metagenome sequences through short read clustering

Research article | Bioinformatics | Microbiology | Data Science

Kexue Li *[1,2], Yakang Lu *[1,2], Li Deng [1,2,3], Lili Wang [1,2], Lizhen Shi [4], Zhong Wang [3,5,6]

Published April 8, 2020

Metagenome assembly

Journals & Magazines > IEEE/ACM Transactions on Comp... > Volume: 19 Issue: 1

K2Mem: Discovering Discriminative K-mers From Sequencing Data for Metagenomic Reads Classification

Publisher: IEEE | Cite This | PDF

Improved kraken2

Davide Storato ; Matteo Comin | All Authors

MetaProb 2: Metagenomic Reads Binning Based on Assembly Using Minimizers and K-Mers Statistics

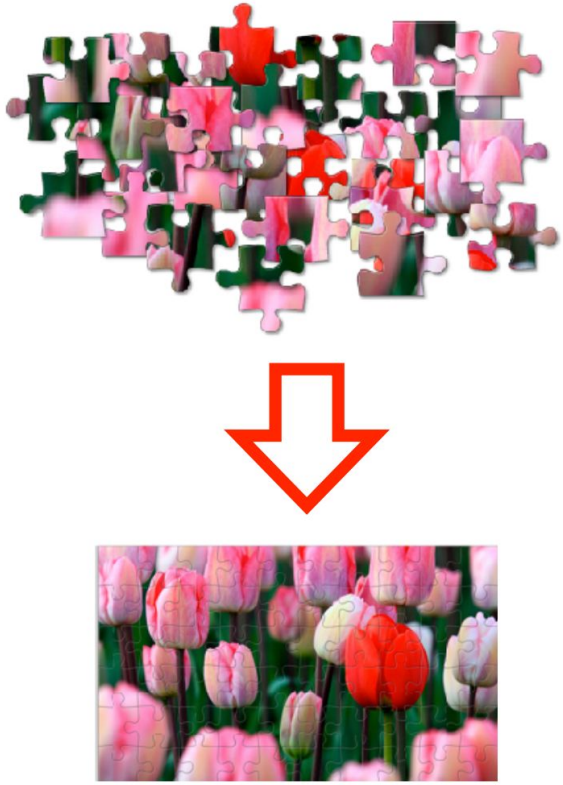
FRANCESCO ANDREACE, CINZIA PIZZI, and MATTEO COMIN[i]

Metagenome assembly

Applications

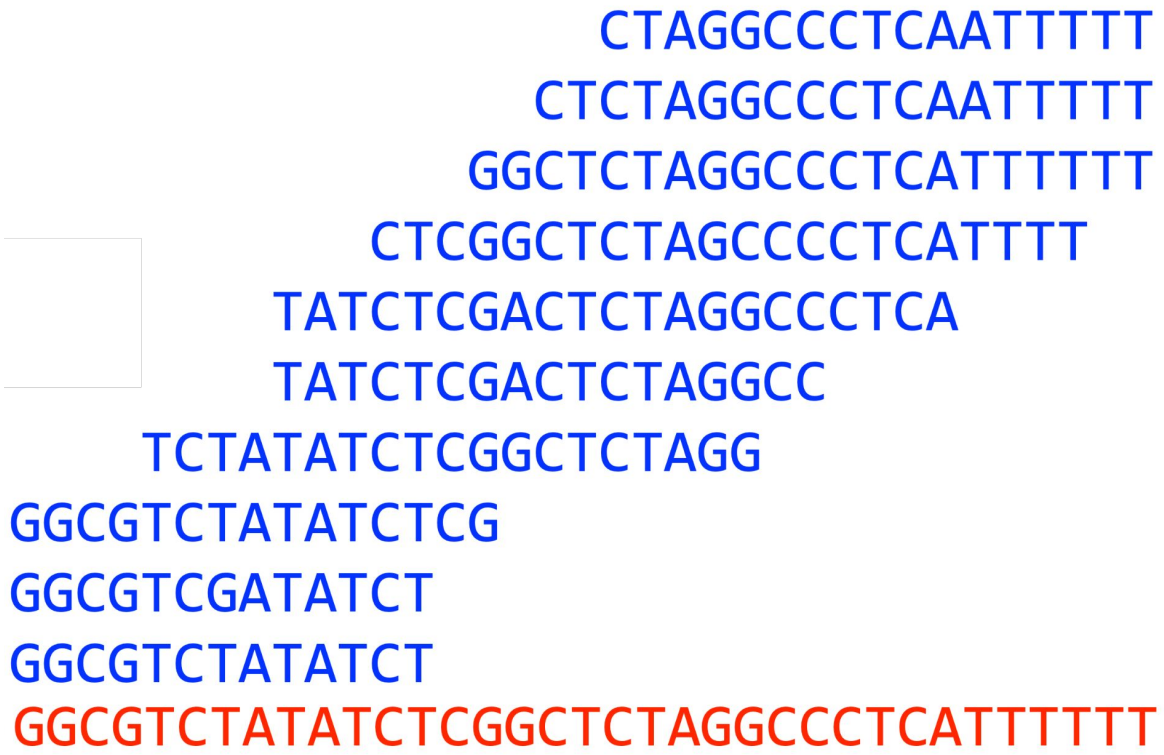
2- Genome assembly

De novo genome assembly



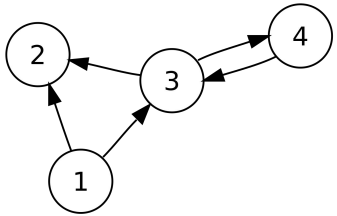
Sequencing reads

CTAGGCCCTCAATTTTT
GGCGTCTATATCT
CTCTAGGCCCTCAATTTTT
TCTATATCTCGGCTCTAGG
GGCTCTAGGCCCTCATTTTTT
CTCGGCTCTAGCCCCTCATTTT
TATCTCGACTCTAGGCCCTCA
GGCGTCGATATCT
TATCTCGACTCTAGGCC
GGCGTCTATATCTCG



- de Bruijn graphs
  - assemble the genome.
  - finding overlaps between kmers of reads

de Bruijn graphs



Graph: nodes, edges

- Nodes: kmers
- Edge: link between k-mers that overlap by k-1 bases.

- Example:
  - 3mers for the sequence **ACTG**
  - 3mers: ACT, CTG

- Merging two nodes result in a K+1 mer (=a part of the sequence).

dBG for few reads

ACTG

CTGC

TGCC

ACTG

ACTG

CTGC

CTGC

CTGC

TGCC

TGCC

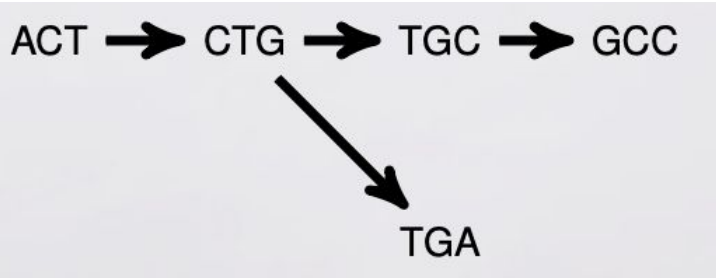
A sequencing error on de Bruijn graph

ACTG

CTGC

TGCC

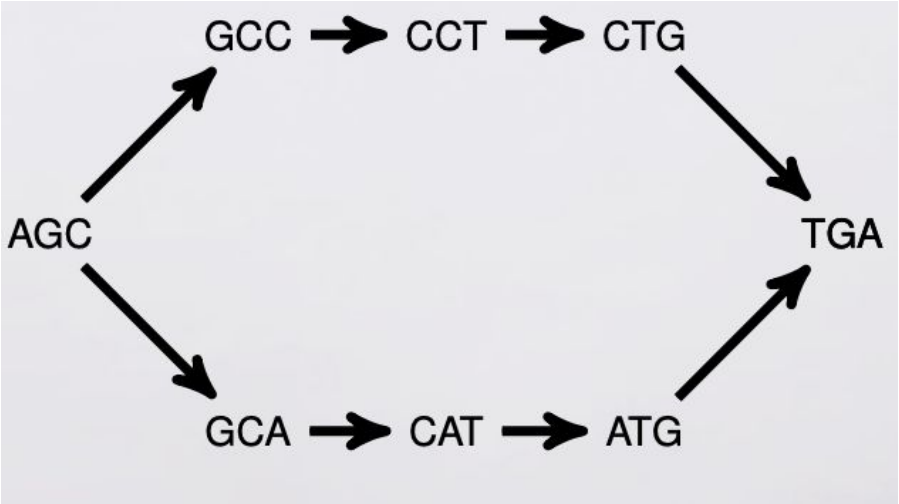
CTG**A**



SNPs in the graph

AGC**C**TGA

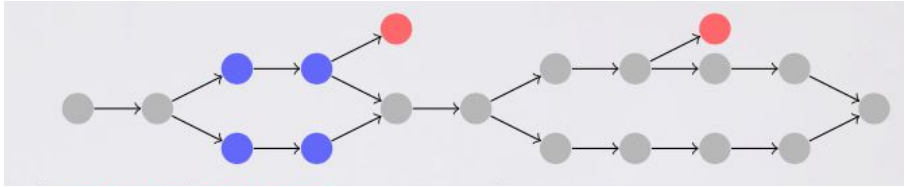
AGC**A**TGA

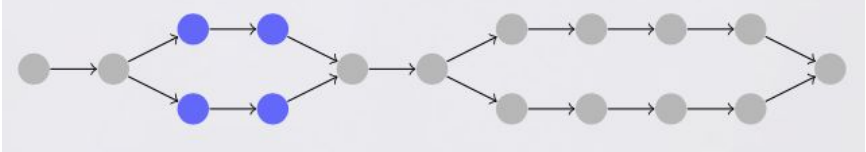


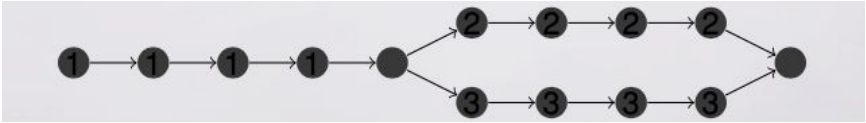
Short read assemblers

- Construct de Bruijn graph
- Remove likely sequencing errors

- Remove variations
- Return simple paths (contigs)
  - Combining all kmers in the path
- Extra steps: repeat-resolving, scaffolding



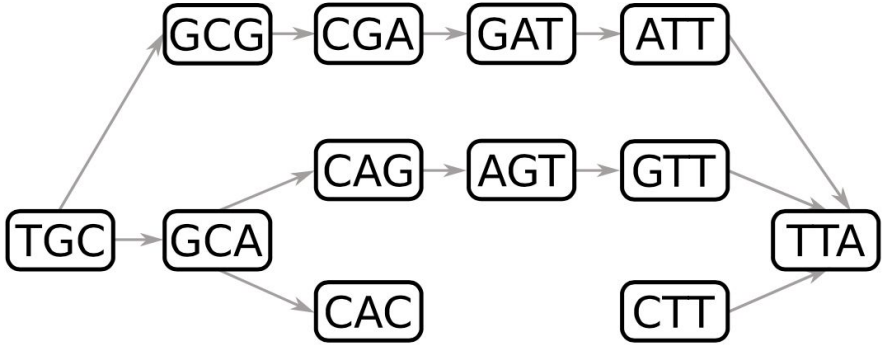


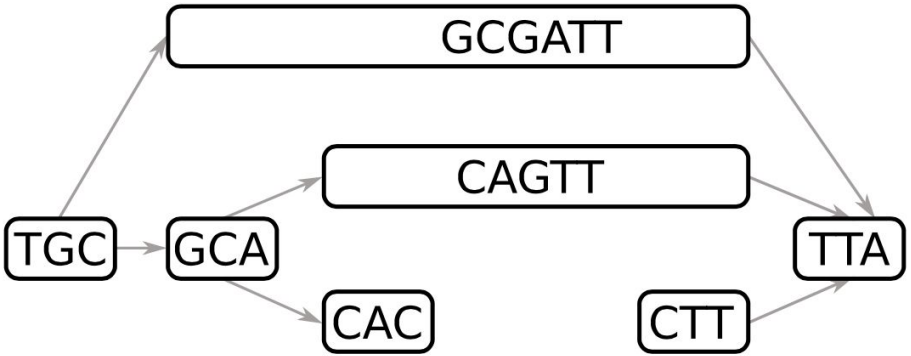


How to apply de Bruijn graphs to genome assembly, Nature biotechnology 2011.

Compacting the de Bruijn graph

1 TB seq data
700 GB kmers





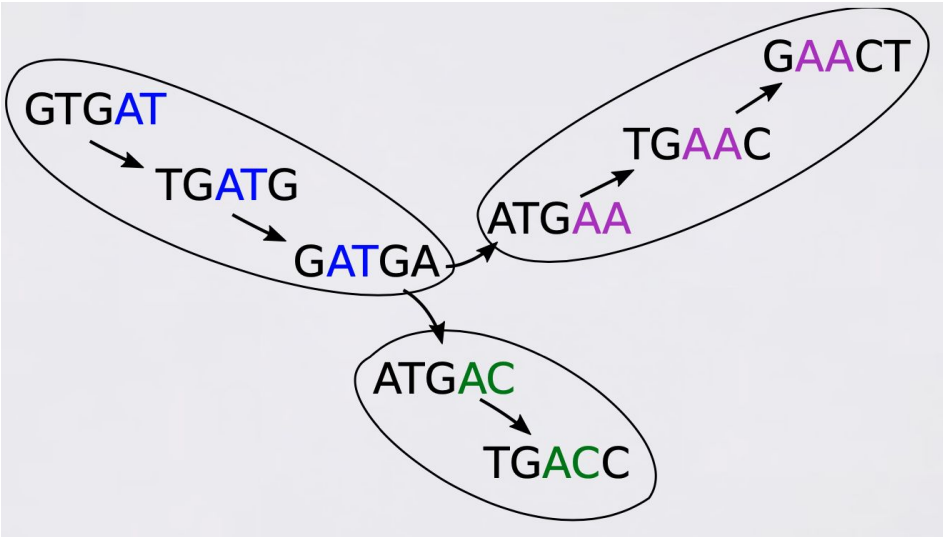
Bifrost: highly parallel construction and indexing of colored and compacted de Bruijn graphs, 2020.

Compacting in parallel

GTGATGA

ATGACC

ATGAACT



Grouping kmers based on their minimizer to work it parallel.

BCALM2 is a tool for compacting the de Bruijn graph.

Bioinformatics, 32, 2016, i201–i208
doi: 10.1093/bioinformatics/btw279
ISMB 2016

OXFORD

Compacting de Bruijn graphs from sequencing data quickly and in low memory

Holley and Melsted *Genome Biology* (2020) 21:249
https://doi.org/10.1186/s13059-020-02135-8

Genome Biology

SOFTWARE Open Access

Bifrost: highly parallel construction and indexing of colored and compacted de Bruijn graphs

Check for updates

Bioinformatics, 37(16), 2021, 2476–2478
doi: 10.1093/bioinformatics/btab004
Advance Access Publication Date: 21 January 2021
Applications Note

OXFORD

Sequence analysis

MBG: Minimizer-based sparse de Bruijn Graph construction

JOURNAL OF COMPUTATIONAL BIOLOGY
Volume 22, Number 5, 2015

Pp. 336–352
DOI: 10.1089/cmb.2014.0160

Research Articles

Too much math

On the Representation of De Bruijn Graphs

JOURNAL OF COMPUTATIONAL BIOLOGY
Volume 11, Number 4, 2004
© Mary Ann Liebert, Inc.
Pp. 734–752

A Preprocessor for Shotgun Assembly of Large Genomes

Ye et al. BMC Bioinformatics 2012, 13(Suppl 6):S1
http://www.biomedcentral.com/1471-2105/13/S6/S1

BMC Bioinformatics

PROCEEDINGS Open Access

Exploiting sparseness in *de novo* genome assembly

Bioinformatics, 36(12), 2020, 3885–3887
doi: 10.1093/bioinformatics/btaa253
Advance Access Publication Date: 20 April 2020
Applications Note

OXFORD

Genome analysis

ntJoin: Fast and lightweight assembly-guided scaffolding using minimizer graphs

ARTICLES
https://doi.org/10.1038/s41587-020-00747-w
nature biotechnology
Check for updates

OPEN

Efficient hybrid de novo assembly of human genomes with WENGAN

CellPress
OPEN ACCESS
Focus on RECOMB
Cell Systems

Article

Minimizer-space de Bruijn graphs: Whole-genome assembly of long reads in minutes on a personal computer

LJA: Assembling Long and Accurate Reads Using Multiplex de Bruijn Graphs

Anton Bankevich, Andrey Bzikadze, Mikhail Kolmogorov, Dmitry Antipov, Pavel A. Pevzner

doi: https://doi.org/10.1101/2020.12.10.420448

nature biotechnology
ARTICLES
https://doi.org/10.1038/s41587-022-01220-6
Check for updates

Multiplex de Bruijn graphs enable genome assembly from long, high-fidelity reads

Applications

3- Kmer counting

K-mer counting

- Given a string, counting the number of occurrences of every kmer.
- Many applications

  - A naive approach:
    - count_dict={}; count_dict[‘ATA’]=1; count_dict[‘ATC’]=2.
    - k-mers as keys and their counts as values.

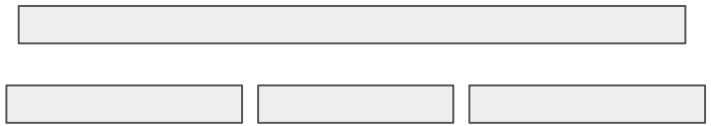
  - Efficient in terms of speed, space and memory
  - Parallelization: locking memory

JellyFish: based on Hash table

- k-mers are stored as keys and their counts are stored as values.
- Lock-free hash table allowing parallel insertion of k-mers and frequency updates.

CAG	62
GGA	33

Gerbil: with minimizer



Evaluation

Data

- Illumina reads
- H. sapiens
- 292 GByte

System

- memory: 64GB
- Hard disk 1TB
- CPUs 16
- Intel CPU E5–2698 2.30GHz

	Time (s)	RAM (GB)	Disk (GB)
Jellyfish (2.2.6)	>15 hours (system hang)		
DSK (2.2.0)	**7,722**	**12**	**133**
DSK (2.2.0; gzip)	9,240	11	134
KAnalyze (2.0.0)	Failed: "IO error writing segment file: no space left on device"		
KAnalyze (2.0.0; gzip)	Failed: "IO error writing segment file: no space left on device"		
KMC3	**3,725***	10	78
KMC3 (gzip)	1,964	11	79
Gerbil (1.0)	4,078	**6***	**66***
Gerbil (1.0; gzip)	2,849	6	66
KCMBT (1.0)	>23 hours		
MSPKmerCounter (0.1)	>15 hours (phase 2 failed: "OutOfMemoryError")		
aTurtle (0.3)	Aborted (core dumped)		

Minimizer -> not equal bins

A benchmark study of k-mer counting methods for high-throughput sequencing , 2018

Bioinformatics, 31(10), 2015, 1569–1576
doi: 10.1093/bioinformatics/btv022
Advance Access Publication Date: 20 January 2015
Original Paper

OXFORD

Sequence analysis

KMC 2: fast and resource-frugal *k*-mer counting

Sebastian Deorowicz[1,*], Marek Kokot[1], Szymon Grabowski[2] and Agnieszka Debudaj-Grabysz[1]

Bioinformatics Advances, 2022, 1–8
https://doi.org/10.1093/bioadv/vbac029
Advance Access Publication Date: 29 April 2022
Original Paper

OXFORD

Sequence Analysis

kmtricks: efficient and flexible construction of Bloom filters for large sequencing data collections

Téo Lemane [1], Paul Medvedev [2,3,4], Rayan Chikhi[5] and Pierre Peterlongo [1,*]

Data Set-Adaptive Minimizer Order Reduces Memory Usage in *k*-Mer Counting

DAN FLOMIN, DAVID PELLOW, and RON SHAMIR

Bioinformatics, 37(17), 2021, 2563–2569
doi: 10.1093/bioinformatics/btab156
Advance Access Publication Date: 8 March 2021
Original Paper

OXFORD

Sequence analysis

Compact and evenly distributed *k*-mer binning for genomic sequences

Johan Nyström-Persson[1,2,*], Gabriel Keeble-Gagnère[3] and Niamat Zawad[2]

Universal hitting set (not a priority)

Shibuya *et al.*
Algorithms for Molecular Biology (2022) 17:5
https://doi.org/10.1186/s13015-022-00212-0

Algorithms for Molecular Biology

RESEARCH Open Access

Space-efficient representation of genomic k-mer count tables

Yoshihiro Shibuya[1], Djamal Belazzougui[2] and Gregory Kucherov[1,3*]

Erbert *et al. Algorithms Mol Biol* (2017) 12:9
DOI 10.1186/s13015-017-0097-9

Algorithms for Molecular Biology

RESEARCH Open Access

Gerbil: a fast and memory-efficient *k*-mer counter with GPU-support

Marius Erbert, Steffen Rechner* and Matthias Müller-Hannemann

Interdisciplinary Sciences: Computational Life Sciences (2020) 12:99–108
https://doi.org/10.1007/s12539-019-00348-5

ORIGINAL RESEARCH ARTICLE

Review paper

Counting Kmers for Biological Sequences at Large Scale

Jianqiu Ge[1] · Jintao Meng[1] · Ning Guo[1] · Yanjie Wei[1] · Pavan Balaji[2] · Shengzhong Feng[1]

OXFORD (GIGA)n SCIENCE

doi: 10.1093/gigascience/giy125
Advance Access Publication Date: 22 October 2018
Review

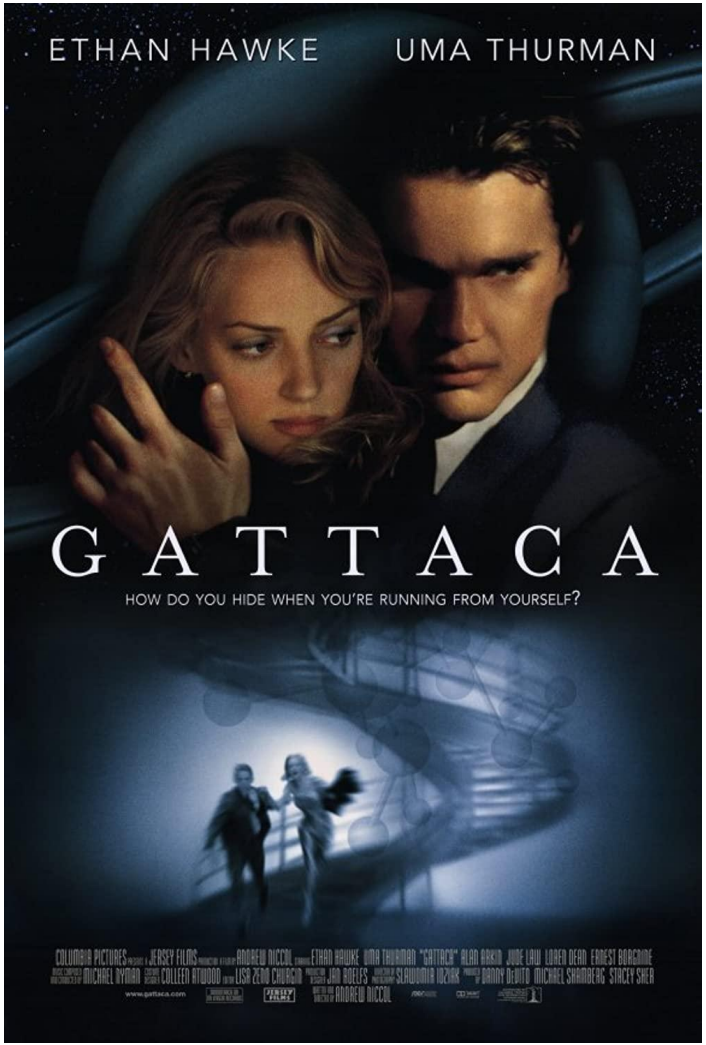
Review paper

REVIEW

A benchmark study of *k*-mer counting methods for high-throughput sequencing

Thank you!

QA?


ETHAN HAWKE
UMA THURMAN
GATTACA
HOW DO YOU HIDE WHEN YOU'RE RUNNING FROM YOURSELF?

Available reviews papers

Rowe *Genome Biology* (2019) 20:199
https://doi.org/10.1186/s13059-019-1809-x

Genome Biology

REVIEW Open Access

When the levee breaks: a practical guide to sketching algorithms for processing the flood of genomic data

Will P. M. Rowe[1,2]

Tutorial + basics

ACM Computing Surveys, Vol. 54, No. 1, Article 17. Publication date: March 2021.

Time order O(n)

Data Structures to Represent a Set of *k*-long DNA Sequences

RAYAN CHIKHI, Center of Bioinformatics and Biostatistics and Integrative Biology
JAN HOLUB, Department of Theoretical Computer Science, Czech Technical University in Prague
PAUL MEDVEDEV, Center for Computational Biology and Bioinformatics

Review

Data structures based on *k*-mers for querying large collections of sequencing data sets

Camille Marchet,[1] Christina Boucher,[2] Simon J. Puglisi,[3] Paul Medvedev.[4,5,6]
Mikaël Salson,[1] and Rayan Chikhi[7]

CSH PRESS GENOME RESEARCH

Useful, we can write similarly describing Biological applications.
Good supplementary describing basics

Annual Review of Biomedical Data Science

Sketching and Sublinear Data Structures in Genomics

Guillaume Marçais,[1,*] Brad Solomon,[2,*] Rob Patro,[3] and Carl Kingsford[1]

Two pages on Minimizer

A survey of mapping algorithms in the long-reads era

Intro on seeding, anchor, kmer

4- Sequence compression

SCIENTIFIC REPORTS
nature research

Corrected: Author Correction

FQSqueezer: *k*-mer-based compression of sequencing data

Sebastian Deorowicz

JOURNAL OF COMPUTATIONAL BIOLOGY
Volume 25, Number 7, 2018
© Mary Ann Liebert, Inc.
Pp. 825–836
DOI: 10.1089/cmb.2018.0068

Dynamic Alignment-Free and Reference-Free Read Compression

PLOS COMPUTATIONAL BIOLOGY

RESEARCH ARTICLE

Hamming-shifting graph of genomic short reads: Efficient construction and its application for compression

Yuansheng Liu, Jinyan Li*

Data Science Institute, University of Technology Sydney, Sydney, Australia

Bioinformatics, 35(12), 2019, 2066–2074
doi: 10.1093/bioinformatics/bty936
Advance Access Publication Date: 8 November 2018
Original Paper

OXFORD

Sequence analysis

Index suffix–prefix overlaps by (*w*, *k*)-minimizer to generate long contigs for reads compression

Bioinformatics, 31(9), 2015, 1389–1395
doi: 10.1093/bioinformatics/btu844
Advance Access Publication Date: 22 December 2014
Original Paper

OXFORD

Sequence analysis

Disk-based compression of data from genome sequencing

Szymon Grabowski[1], Sebastian Deorowicz[2,*] and Łukasz Roguski[3,4]

Bioinformatics, 34(16), 2018, 2748–2756
doi: 10.1093/bioinformatics/bty205
Advance Access Publication Date: 29 March 2018
Original Paper

OXFORD

Sequence analysis

FaStore: a space-saving solution for raw sequencing data

Łukasz Roguski[1,2], Idoia Ochoa[3], Mikel Hernaez[4] and Sebastian Deorowicz[5,*]

5- Sequence error correction

LaPierre *et al. BMC Bioinformatics* (2019) 20:552
https://doi.org/10.1186/s12859-019-3103-z

BMC Bioinformatics

SOFTWARE Open Access

De novo Nanopore read quality improvement using deep learning

Check for updates

nature communications

Article https://doi.org/10.1038/s41467-022-34381-8

VeChat: correcting errors in long reads using variation graphs

Simultaneous compression of multiple error-corrected short-read sets for faster data transmission and better de novo assemblies

nature COMMUNICATIONS

ARTICLE

Check for updates

https://doi.org/10.1038/s41467-020-20340-8 OPEN

Error correction enables use of Oxford Nanopore technology for reference-free transcriptome analysis

Kristoffer Sahlin[1] & Paul Medvedev[2,3,4]✉

Bioinformatics, 37(11), 2021, 1604–1606
doi: 10.1093/bioinformatics/btaa915
Advance Access Publication Date: 28 October 2020
Applications Note

OXFORD

Sequence analysis

Minirmd: accurate and fast duplicate removal tool for short reads via multiple minimizers

Yuansheng Liu[1], Xiaocai Zhang[2], Quan Zou[3] and Xiangxiang Zeng[1,*]

Not a priority

6- Variant calling & MSA

Dysgu: efficient structural variant calling using short or long reads

Kez Cleal* and Duncan M. Baird*

State-of-the-art structural variant calling: What went conceptually wrong and how to fix it?

Markus Schmidt[1] and Arne Kutzner[1,]*

Mapping-free variant calling using haplotype reconstruction from k-mer frequencies

Peter A. Audano*, Shashidhar Ravishankar and Fredrik O. Vannberg*

FAME: fast and memory efficient multiple sequences alignment tool through compatible chain of roots

Etminan Naznooshsadat[1], Parvinnia Elham[1,*] and Sharifi-Zarchi Ali[2]

7- Read alignment
(good reviews exist)

Bioinformatics, 36, 2020, i111–i118
doi: 10.1093/bioinformatics/btaa435
ISMB 2020

OXFORD

Weighted minimizer sampling improves long read mapping

Chirag Jain[1,*], Arang Rhie[1], Haowen Zhang[2], Claudia Chu[2], Brian P. Walenz[1], Sergey Koren[1] and Adam M. Phillippy[1]

nature methods
ARTICLES
https://doi.org/10.1038/s41592-022-01457-8
Check for updates

Long-read mapping to repetitive reference sequences using Winnowmap2

Chirag Jain[1,2], Arang Rhie[2], Nancy F. Hansen[3], Sergey Koren[2] and Adam M. Phillippy[2]

Rautiainen and Marschall *Genome Biology* (2020) 21:253
https://doi.org/10.1186/s13059-020-02157-2

Genome Biology

SOFTWARE Open Access

GraphAligner: rapid and versatile sequence-to-graph alignment

Mikko Rautiainen[1,2,3*] and Tobias Marschall[4*]

Alser *et al. Genome Biology* (2021) 22:249
https://doi.org/10.1186/s13059-021-02443-7

Genome Biology

REVIEW Open Access

Technology dictates algorithms: recent developments in read alignment

Mohammed Alser[1,2,3†], Jeremy Rotman[4†], Dhrithi Deshpande[5], Kodi Taraszka[4], Huwenbo Shi[6,7], Pelin Icer Baykal[8], Harry Taegyun Yang[4,9], Victor Xue[4], Sergey Knyazev[8], Benjamin D. Singer[10,11,12], Brunilda Balliu[13], David Koslicki[14,15,16], Pavel Skums[8], Alex Zelikovsky[8,17], Can Alkan[2,18], Onur Mutlu[1,2,3†] and Serghei Mangul[5*†]

A survey of mapping algorithms in the long-reads era

Kristoffer Sahlin
Department of Mathematics, Science for Life Laboratory, Stockholm University, 106 91, Stockholm, Sweden.

Focused on minimizer

Bioinformatics, 34(18), 2018, 3094–3100
doi: 10.1093/bioinformatics/bty191
Advance Access Publication Date: 10 May 2018
Original Paper

OXFORD

Sequence analysis

Minimap2: pairwise alignment for nucleotide sequences

Heng Li*

Department of Medical Population Genetics Program, Broad Institute, Cambridge, MA 02142, USA

Bioinformatics, 32(14), 2016, 2103–2110
doi: 10.1093/bioinformatics/btw152
Advance Access Publication Date: 19 March 2016
Original Paper

OXFORD

Sequence analysis

Minimap and miniasm: fast mapping and de novo assembly for noisy long sequences

Heng Li

Alternative data structures
(math+computer sci.)

PeerJ

Syncmers are more sensitive than minimizers for selecting conserved *k*-mers in biological sequences

Robert Edgar
None, Corte Madera, CA, USA

Masked Minimizers: Unifying sequence sketching methods

Effective sequence similarity detection with strobemers

Kristoffer Sahlin

A randomized parallel algorithm for efficiently finding near-optimal universal hitting sets

Sequence-specific minimizers via polar sets

Hongyu Zheng, Carl Kingsford and Guillaume Marçais*

A performant bridge between fixed-size and variable-size seeding

Arne Kutzner[1], Pok-Son Kim[2] and Markus Schmidt[1*]

Improved design and analysis of practical minimizers

Hongyu Zheng, Carl Kingsford and Guillaume Marçais*

The minimizer Jaccard estimator is biased and inconsistent*

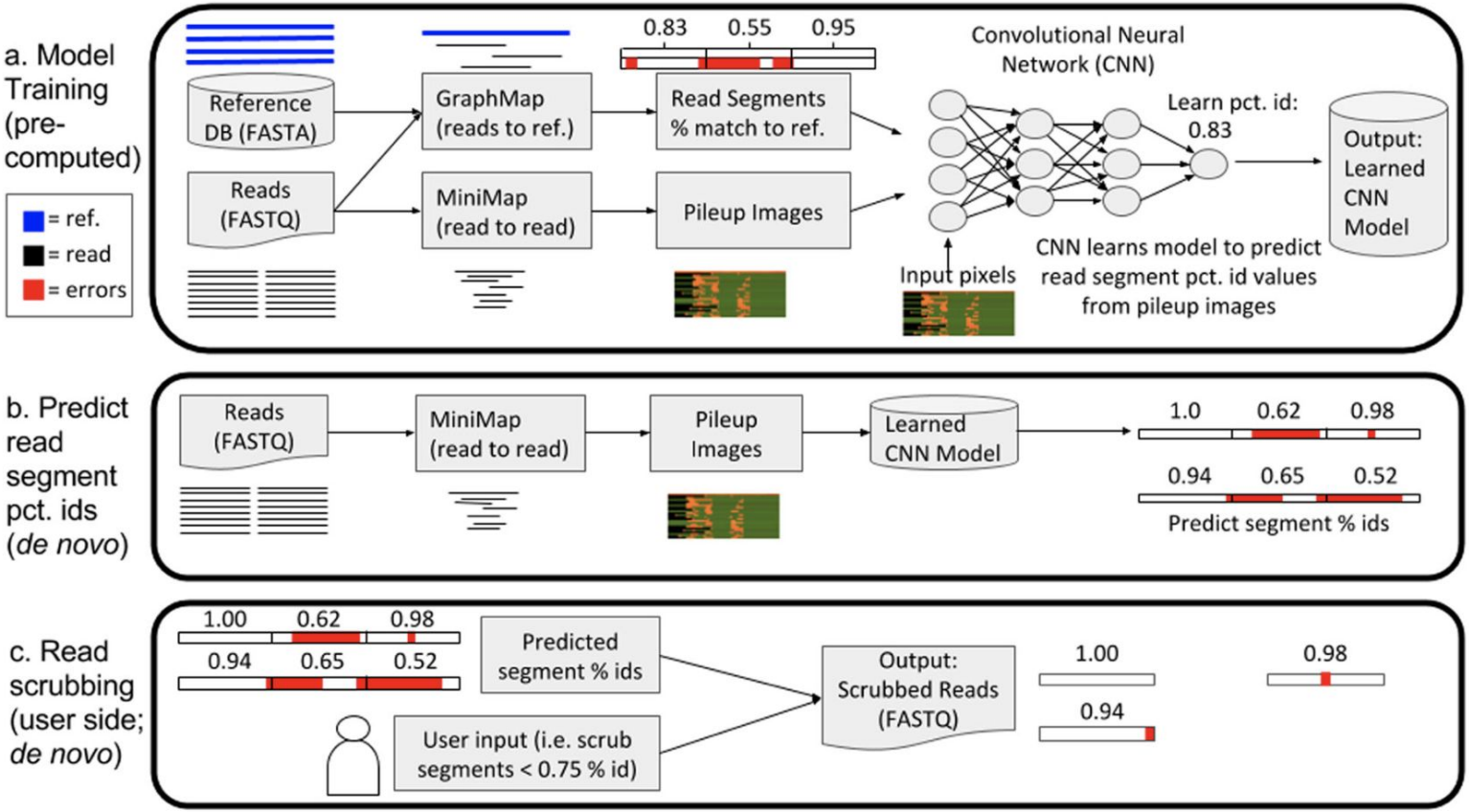
BLight: efficient exact associative structure for k-mers

Camille Marchet *, Mael Kerbiriou and Antoine Limasset *

Paper outline “**Applications of Minimizer in genomics**”

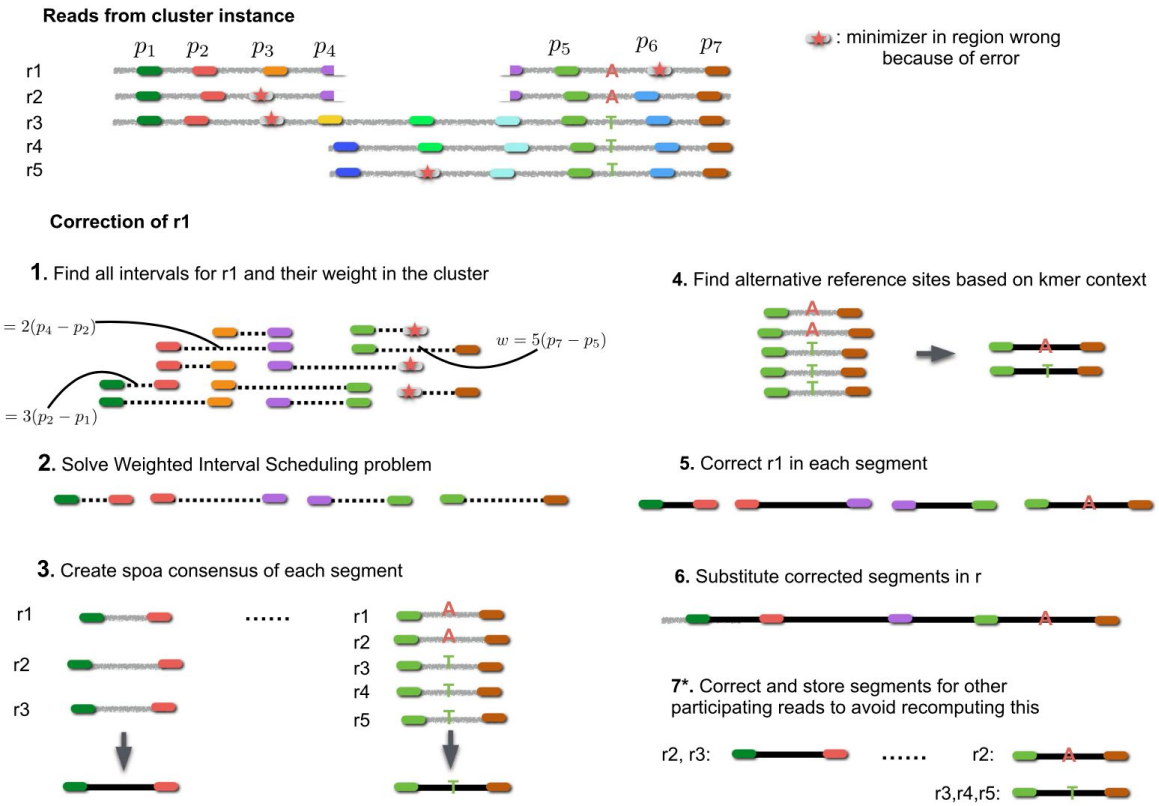
1. intro (importance, why now(biogenome project), definition)
2. metagenomics (7)
3. de Bruijn graph (4)
4. genome assembly (7)
5. kmer counting (5)
6. sequence compression (6)
7. sequence error correction (3)
8. variant calling (3)
9. Discussion and conclusion

Tables, figure.



De novo Nanopore read quality improvement using deep learning, BMC bioinformatics 2019

isONcorrect method



Error correction enables use of Oxford Nanopore technology for reference-free transcriptome analysis, Nature communication, 2021.

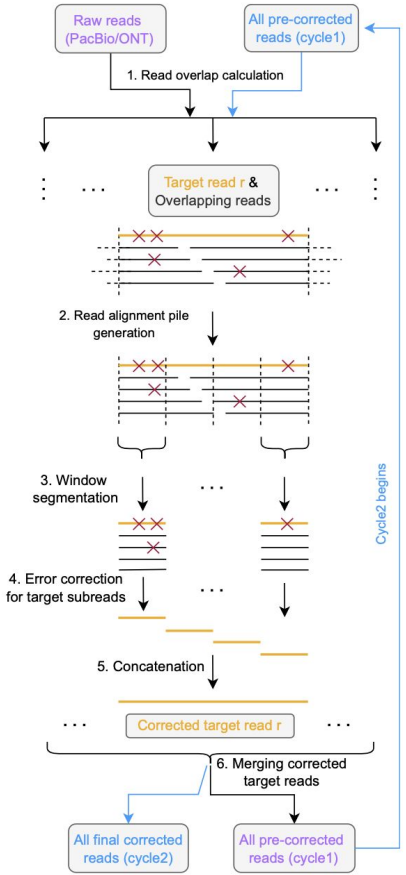


Fig. 1 | Workflow of VeChat. The input and output of cycle 1 and cycle 2 are labeled

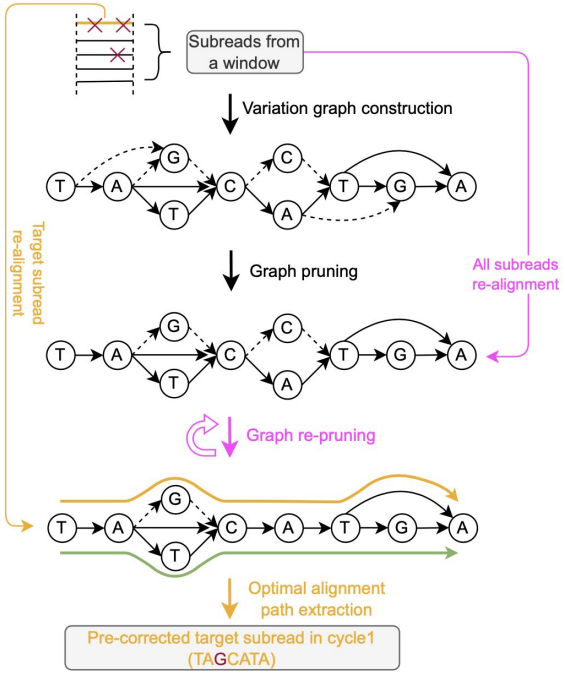
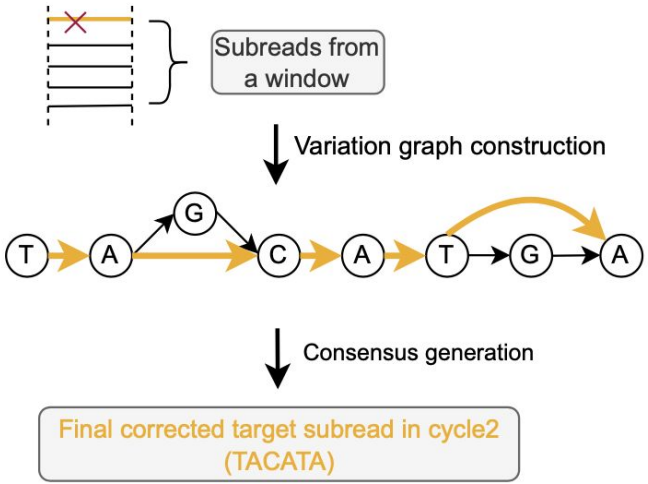


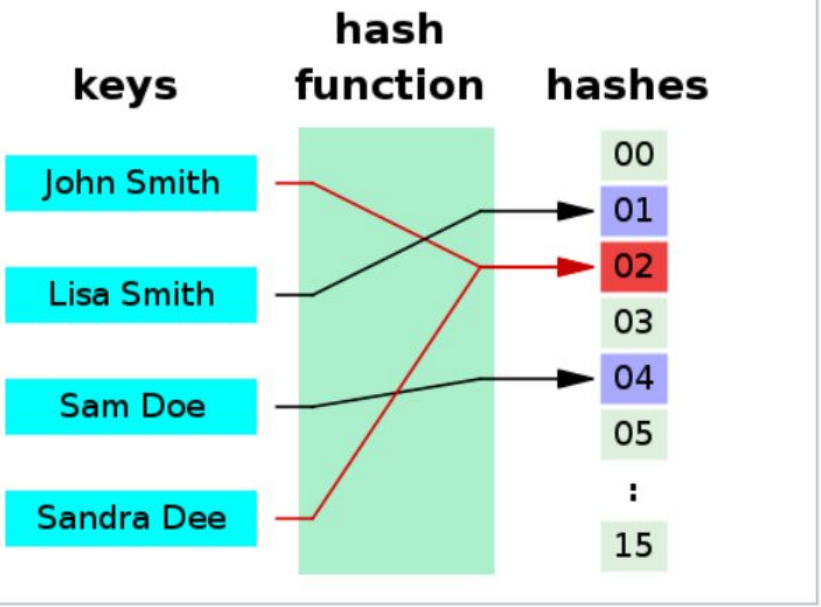
Fig. 2 | Error correction for one target subread in cycle 1. The error correction



3 | Error correction for one target subread in cycle 2. The bold orang

Hash function

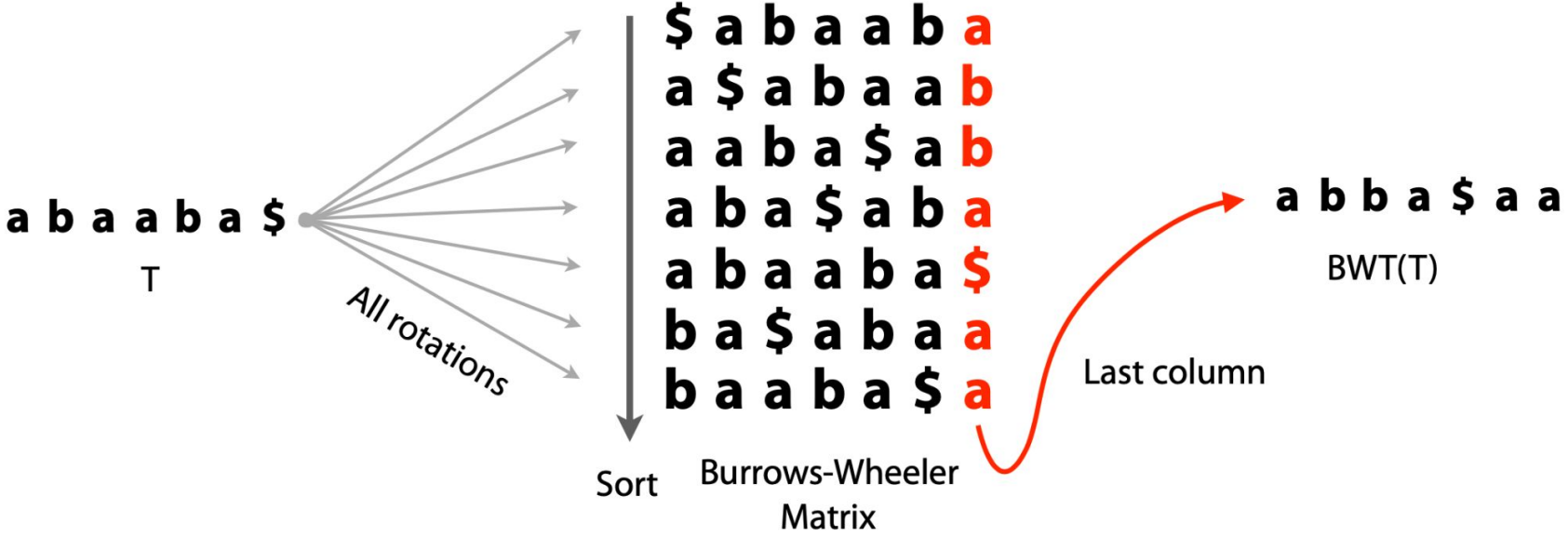
A hash function is any function that can be used to map data of arbitrary size to fixed-size values.



A hash function that maps names to integers from 0 to 15. There is a collision between keys "John Smith" and "Sandra Dee".

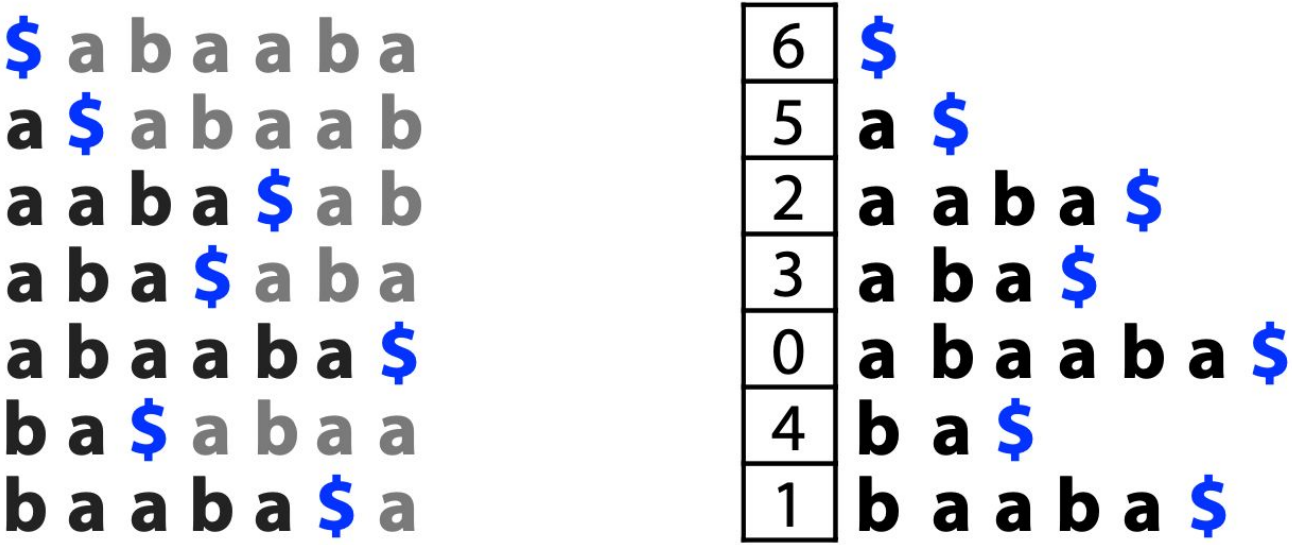
wikipedia

Burrows–Wheeler transform



langmead-lab.org/teaching-materials/

Suffix array



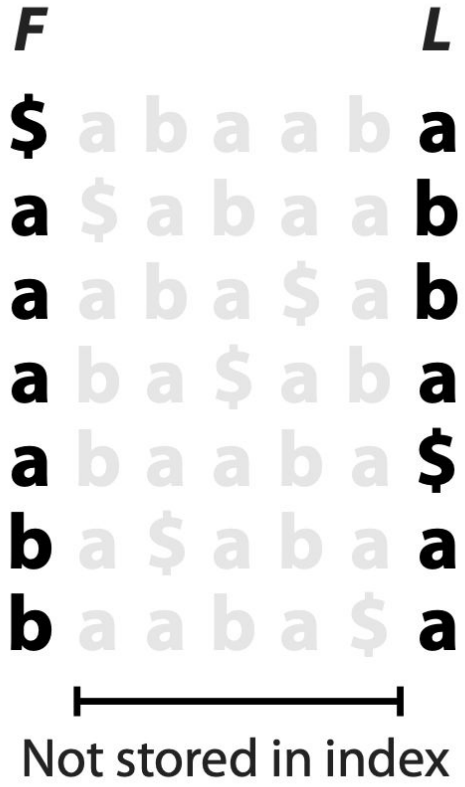
BWT

langmead-lab.org/teaching-materials/

FM index

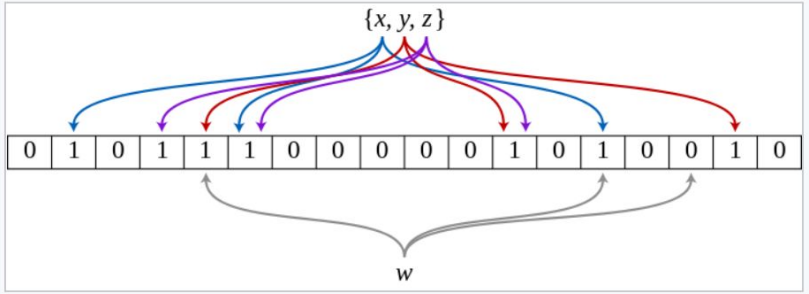
Using the BWT matrix

a last-to-first column mapping LF(i) from an index i to a
L[i]

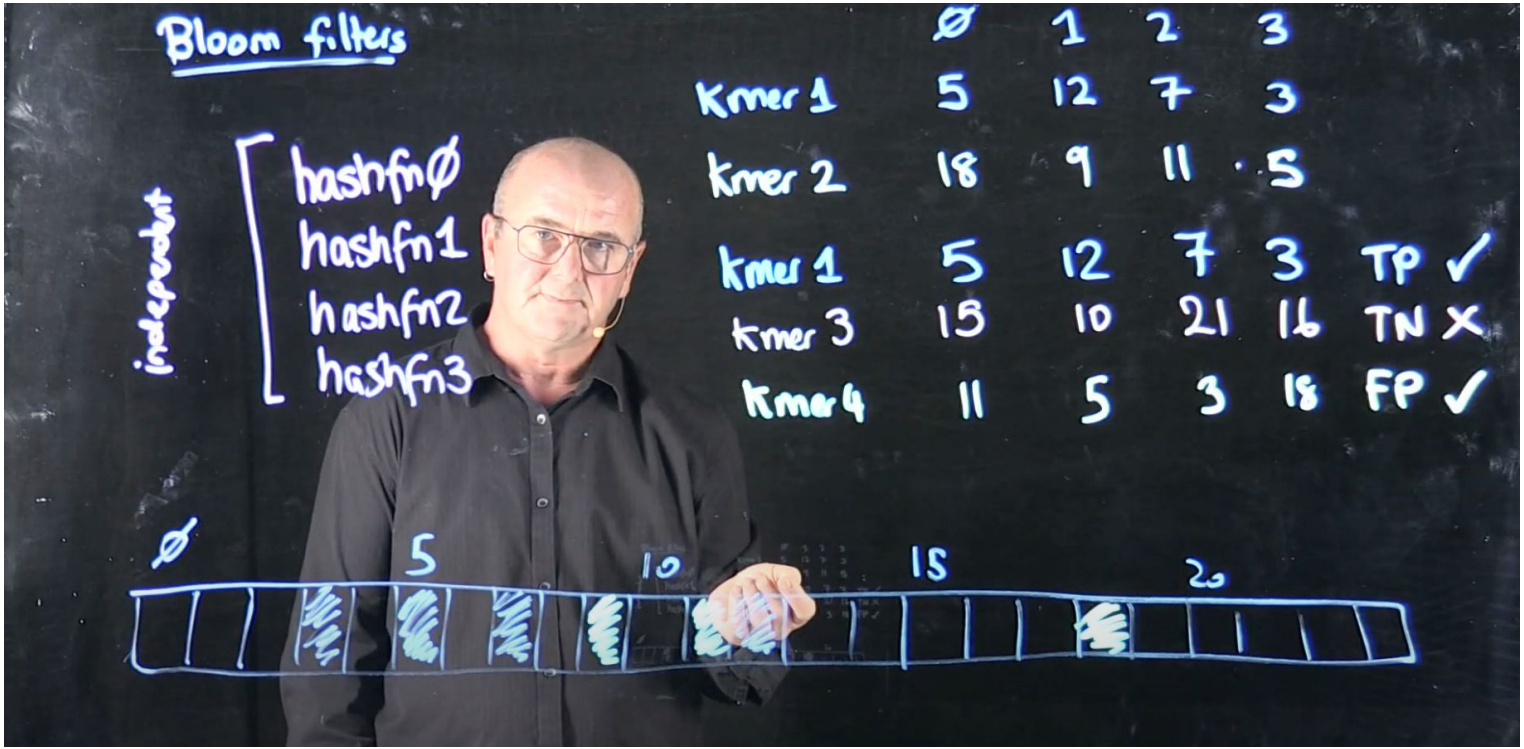


Bloom filter

- a space-efficient probabilistic data structure
- test whether an element is a member of a set
- False positive matches are possible, but false negatives are not
- An empty Bloom filter is a bit array of m bits, all set to 0.
- Using k different hash functions
- each function maps (hashes) an element to one of the m array positions.



An example of a Bloom filter, representing the set $\{x, y, z\}$. The colored arrows show the positions in the bit array that each set element is mapped to. The element w is not in the set $\{x, y, z\}$, because it hashes to one bit-array position containing 0. For this figure, $m = 18$ and $k = 3$.


Bloom filters
independent
hashfn0
hashfn1
hashfn2
hashfn3
Ø 1 2 3
Kmer 1 5 12 7 3
Kmer 2 18 9 11 5
kmer 1 5 12 7 3 TP ✓
kmer 3 15 10 21 16 TN X
kmer 4 11 5 3 18 FP ✓
Ø
5
10
15
20

MinHASH

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} \approx \frac{|S(A \cup B) \cap S(A) \cap S(B)|}{|S(A \cup B)|}$$

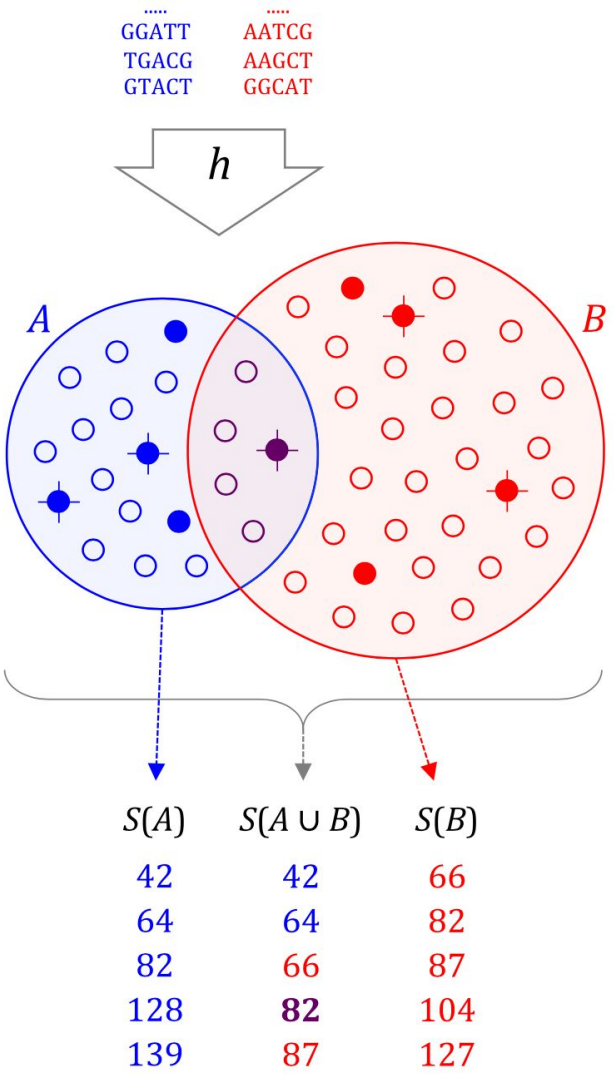
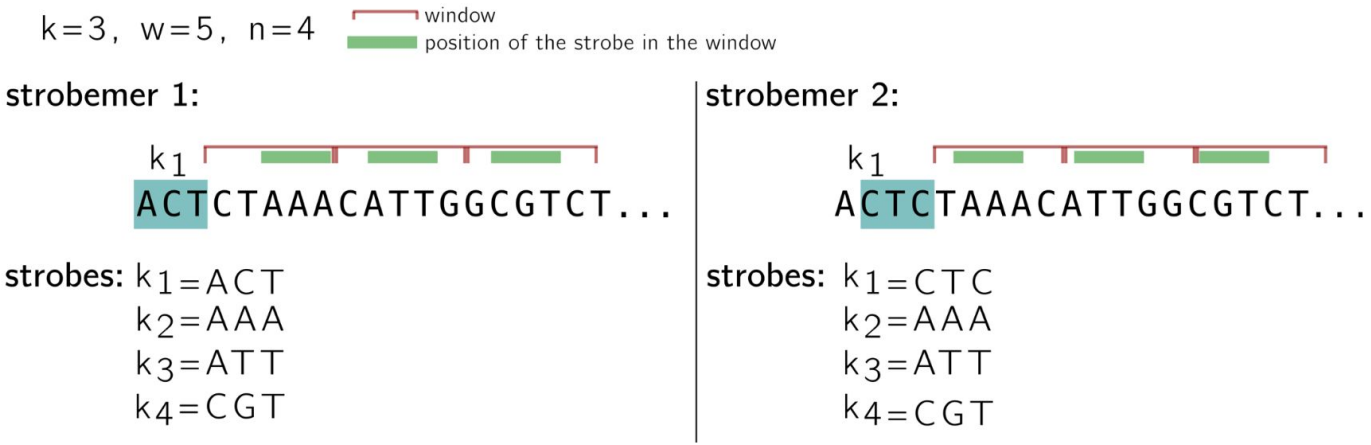
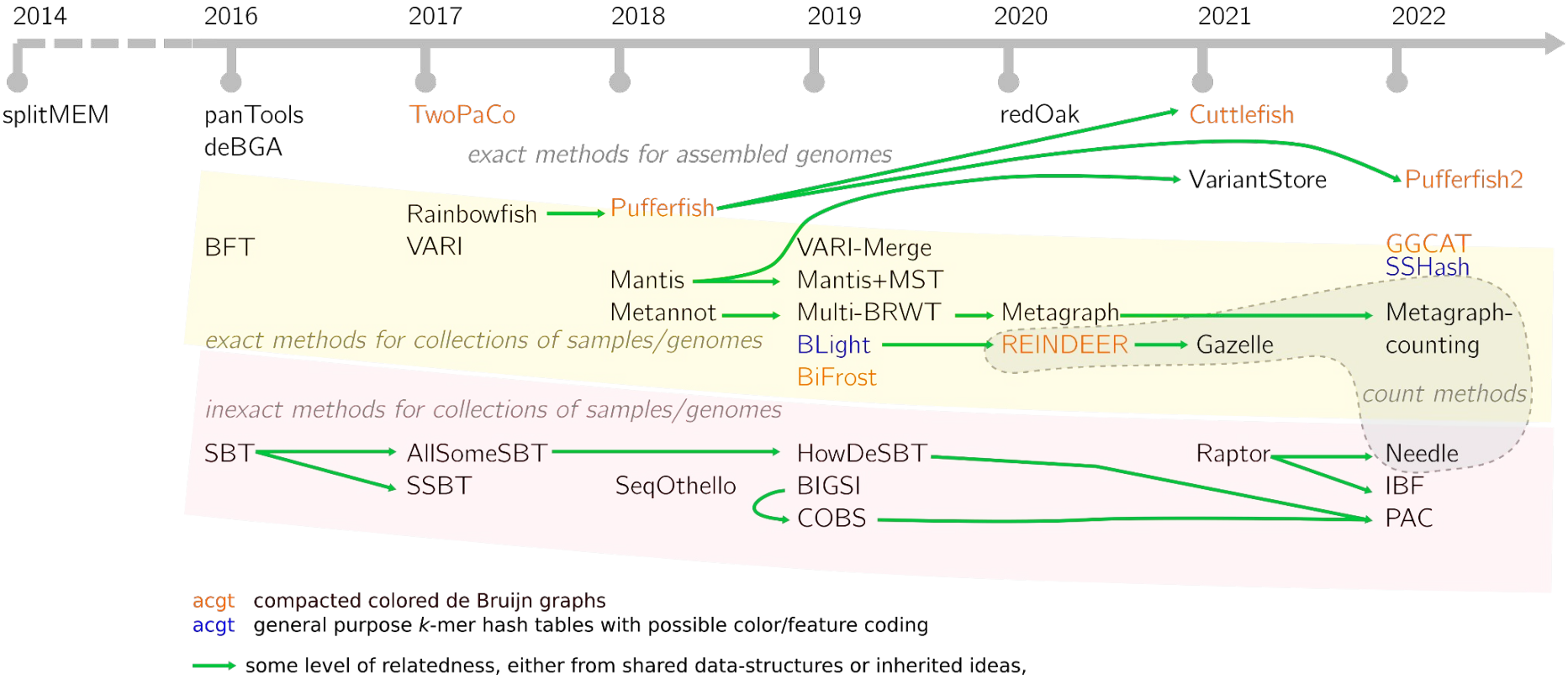


Fig. 1 Overview of the MinHash bottom sketch strategy for estimating the Jaccard index. First, the sequences of two datasets are decomposed into their constituent k-mers (*top*, *blue and red*) and each k-mer is passed through a hash function *h* to obtain a 32- or 64-bit hash, depending on the input k-mer size. The resulting hash sets, *A* and *B*, contain |*A*| and |*B*| distinct hashes each (*small circles*). The Jaccard index is simply the fraction of shared hashes (*purple*) out of all distinct hashes in *A* and *B*. This can be approximated by considering a much smaller random sample from the union of *A* and *B*. MinHash sketches *S*(*A*) and *S*(*B*) of size $s = 5$ are shown for *A* and *B*, comprising the five smallest hash values for each (*filled circles*). Merging *S*(*A*) and *S*(*B*) to recover the five smallest hash values overall for *A*∪*B* (*crossed circles*) yields *S*(*A*∪*B*). Because *S*(*A*∪*B*) is a random sample of *A*∪*B*, the fraction of elements in *S*(*A*∪*B*) that are shared by both *S*(*A*) and *S*(*B*) is an unbiased estimate of *J*(*A*,*B*)

StrobeMer


k=3, w=5, n=4
window
position of the strobe in the window
strobemer 1:
k1
ACTCTAAACATTGGCGTCT...
strobes: k1=ACT
k2=AAA
k3=ATT
k4=CGT
strobemer 2:
k1
ACTCTAAACATTGGCGTCT...
strobes: k1=CTC
k2=AAA
k3=ATT
k4=CGT



https://kamimrcht.github.io/webpage/sets_kmer_sets.html

Time and space

count information with each k-mer. Rather than present how this is c
structure that supports it, we have a separate section (Section 7.4) dec
data structures can be adapted to store count information.

3 BASIC APPROACHES

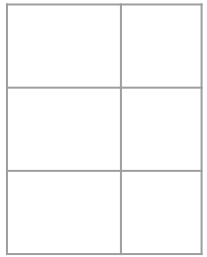
Perhaps the most basic static representation that is used in practice i
list of k-mers. The construction time is $O(nk)$ using any linear tim
the space needed to store the list is $\Theta(nk)$. A membership query is ex
time $O(k \log n)$. This representation is both space- and time-inefficien
approaches we will discuss (e.g., unitig-based approaches or BOSS). B
with very limited computer science background, making it still relev

Sorted lists can be partitioned to speed up queries. In this approach,
[2014], the k-mers are partitioned according to a minimizer functi
minimizer of a k-mer x is the smallest (according to some given p

Swap and compare kmer counting

Example:

workers W1 & W2.

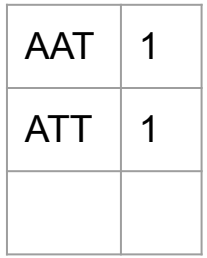


W1: read count of AAT (0)

W1: increment AAT 0->1

W2: read count of ATT (0)

W2: increment ATT 0->1



W1:

W2: read count AAT (1)

W1: read count AAT (1)

W2: increment AAT 1->2

W1: increment AAT 1->2

W2:

W1, No, I won't increment AAT as 1->2 does not hold

to ensure low random access memory (RAM) usage, as has been argued by Li et al. (2013).

In this study, we introduce a new approach for adapting the minimizer order to the target sequence data. The method, called Adaptive Order (AdaOrder), iteratively updates the minimizer order based on an estimate of the minimizer loads in the data to reduce the maximum load. We demonstrated its ability to lower the maximum load compared with all predefined orders, except the frequency-based order.

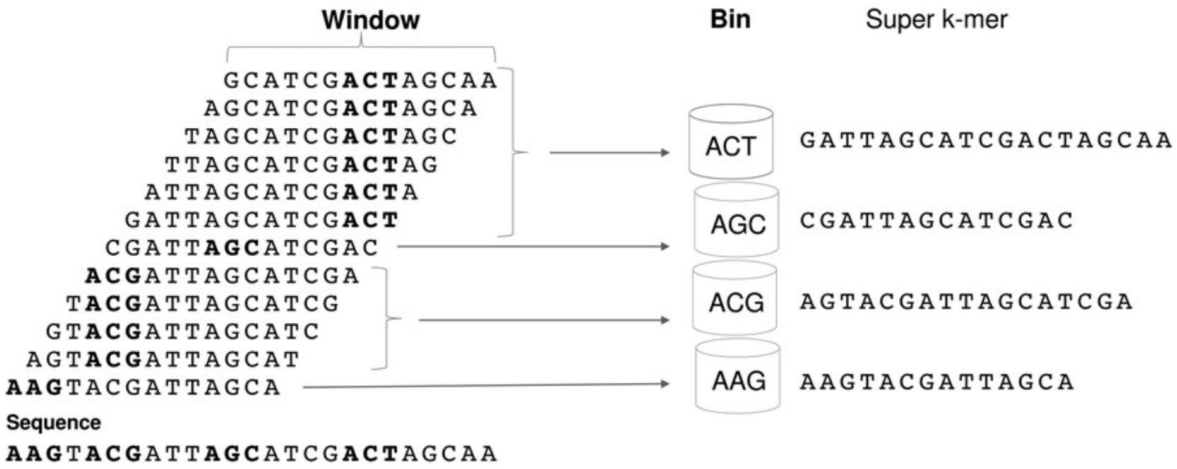


FIG. 1. Illustration of a minimizer scheme and binning. Here $k=12$ and $m=3$. The input sequence is broken into windows of length k, and in each window the m-long minimizer according to the lexicographic order (shown in bold) is selected. The k-mer is assigned to the partition whose label is the minimizer. Consecutive windows tend to select the same minimizer, and the concatenation of the consecutive windows forms the super-k-mer that is stored in the bin.

Sequence comparison with minimizer

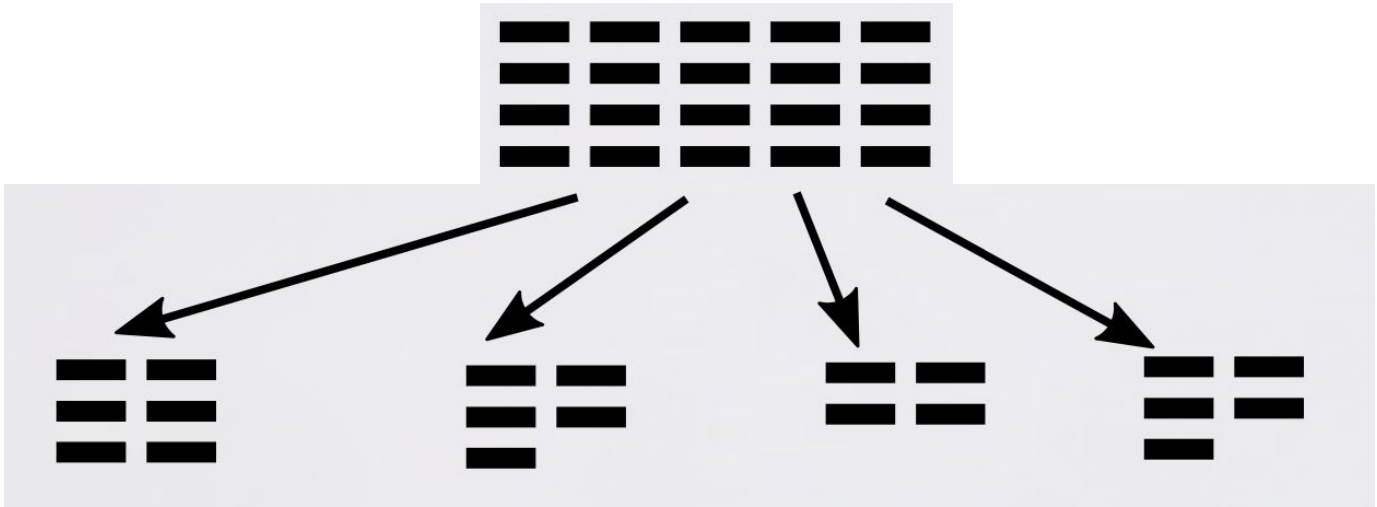
- Number of minimizer at most = L-w+1 but
- Similar sequences have high probability of having the same minimizers.

- Sequence of minimizers is also called a sketch of the original seq.

- With non-overlapping windows -> L/w minimizers for sequence of length L.
- For a random order (w not too large), a factor of 2/(w + 1).

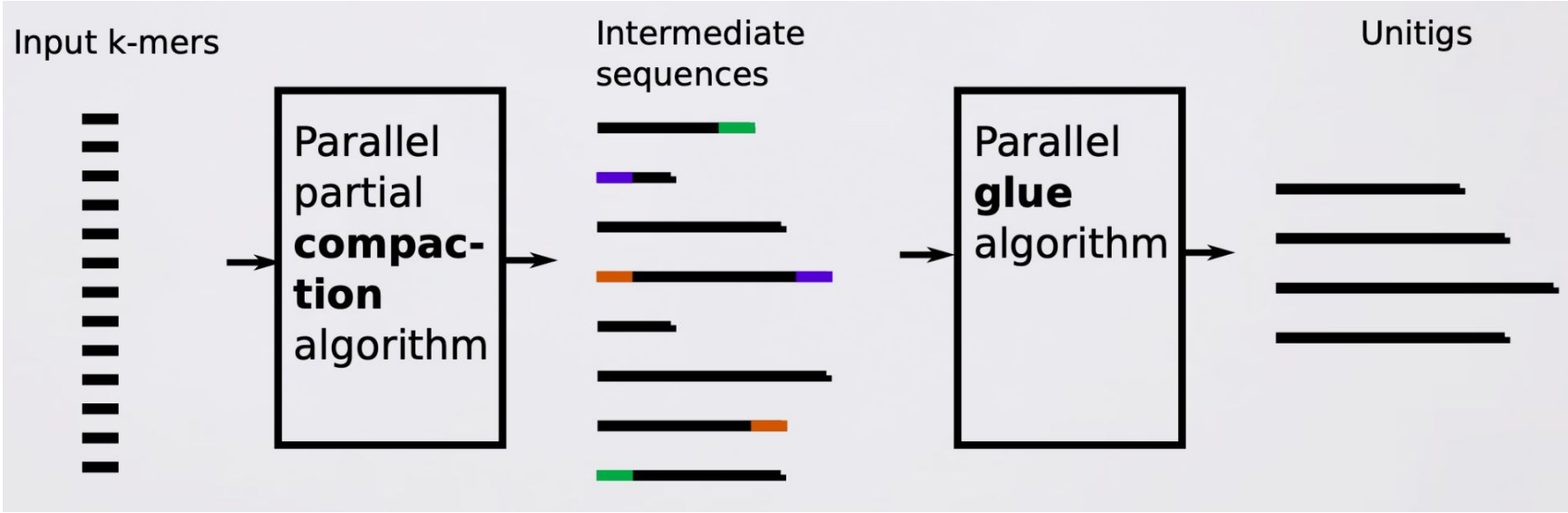
Memory and speed; minimizer

- partitioning input k-mers on disk
- based on minimizers

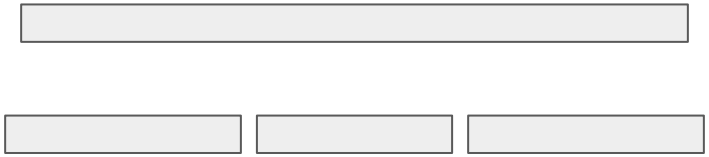


BCALM 2

A tool for compacting the de Bruijn graph



Parallel counting:
splitting the sequence



Issues: missing kmers, combining stats of each part.

Gerbil: with minimizer

- 4-mers
- 3-mer minimizer (bold)
- Finding (overlapping) part of sequence with the same minimizer on temp files

```
CAAGAACAGTG
CAAG                1. CAAGA
 AAGA
  AGAA              2. AGAA
   GAAC             3. GAACA
    AACA
     ACAG           4. ACAG
      CAGT          5. CAGTG
       AGTG
```

Gerbil: a fast and memory-efficient k-mer counter with GPU-support, 2017