Variation Graphs

Finlay Maguire

March 9, 2020

FCS, Dalhousie

Overview

- Characterising heterogeneous DNA/RNA samples:

Overview

- Characterising heterogeneous DNA/RNA samples:
  - Coloured de Bruijn Graphs (e.g. Cortex, Mykrobe)

Overview

- Characterising heterogeneous DNA/RNA samples:
    - Coloured de Bruijn Graphs (e.g. Cortex, Mykrobe)
    - Succinct data structures (e.g. Vari)

Overview

- Characterising heterogeneous DNA/RNA samples:
    - Coloured de Bruijn Graphs (e.g. Cortex, Mykrobe)
    - Succinct data structures (e.g. Vari)
- A taxonomy of graphs

Overview

- Characterising heterogeneous DNA/RNA samples:
  - Coloured de Bruijn Graphs (e.g. Cortex, Mykrobe)
  - Succinct data structures (e.g. Vari)
- A taxonomy of graphs
- Searching large databases using reference graphs:

Overview

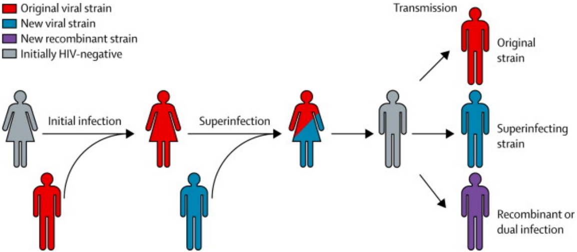
- Characterising heterogeneous DNA/RNA samples:
    - Coloured de Bruijn Graphs (e.g. Cortex, Mykrobe)
    - Succinct data structures (e.g. Vari)
- A taxonomy of graphs
- Searching large databases using reference graphs:
    - k-mer graph indexing (e.g. groot, BlastFrost)

Overview

- Characterising heterogeneous DNA/RNA samples:
  - Coloured de Bruijn Graphs (e.g. Cortex, Mykrobe)
  - Succinct data structures (e.g. Vari)
- A taxonomy of graphs
- Searching large databases using reference graphs:
  - k-mer graph indexing (e.g. groot, BlastFrost)
  - Burrows-Wheeler Transform extensions (e.g. Variation Graph toolkit GCSA)

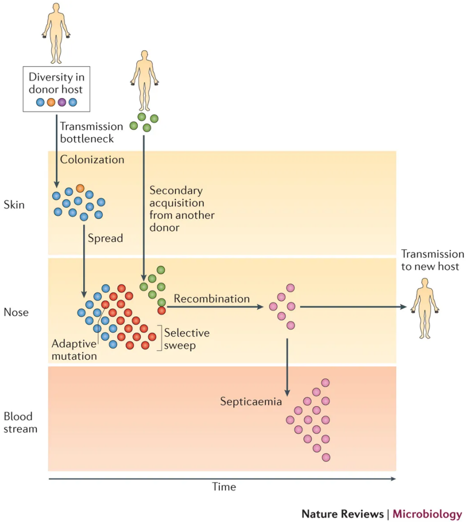
What is heterogeneity

Mixed-infections



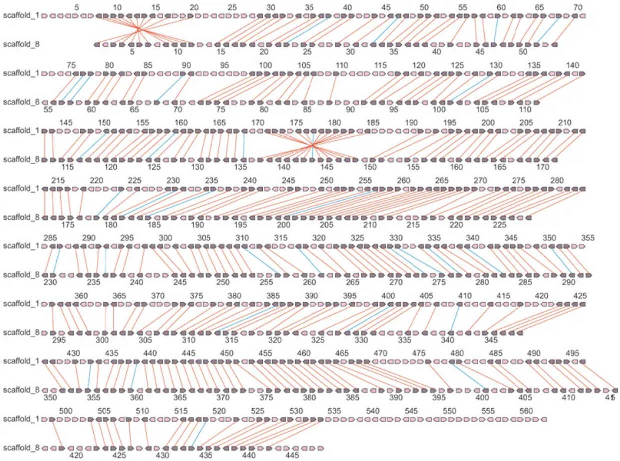
HIV ‘super-infection’ [Redd et al., 2013]

Within-host evolution



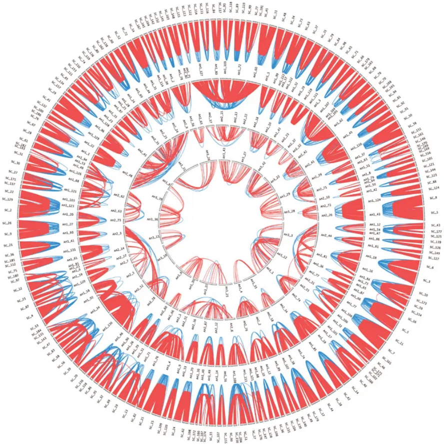
Within host evolution of *Staphylococcus aureus* [Didelot et al., 2016]

Polyploidy



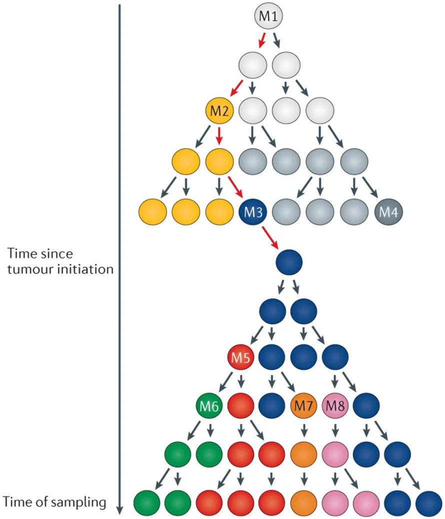
Polyploidy and whole genome duplication in *Paramecium* [Aury et al., 2006]

Polyploidy



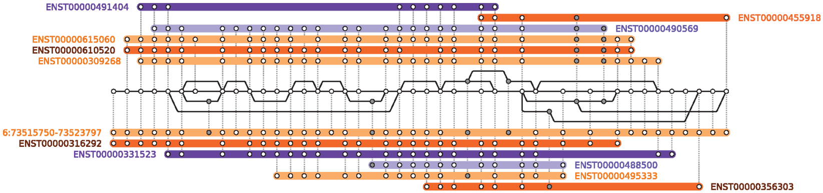
Polyploidy and whole genome duplication in *Paramecium* [Aury et al., 2006]

Cancer evolution



Tumour lineage tracking [Gawad et al., 2016]

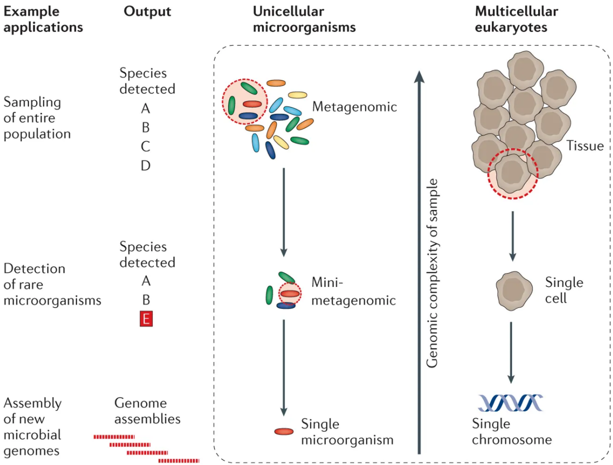
Transcriptomics



All transcripts of the EEF1A1 gene in Ensembl v80 [com, 2018]

Characterising heterogeneity

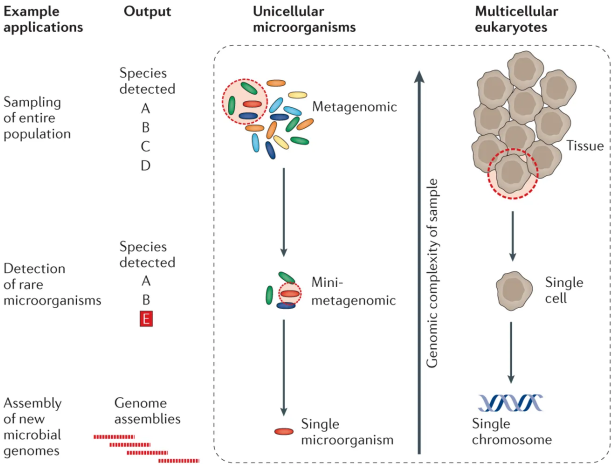
Single-cell methods



[Gawad et al., 2016]

- Often more like 'a few'-cell sequencing

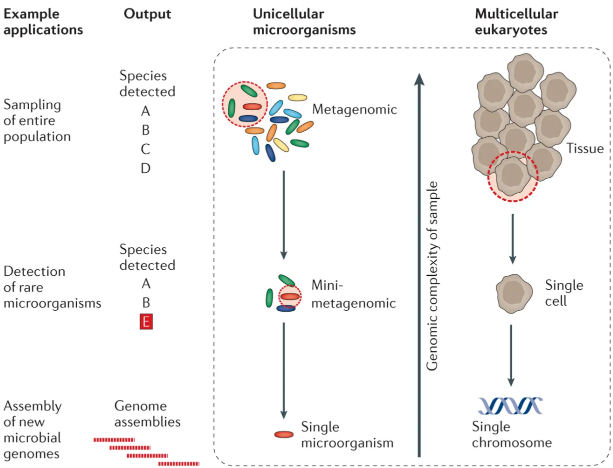
Single-cell methods



[Gawad et al., 2016]

- Often more like ‘a few’-cell sequencing
- Ploidy and viruses are still difficult

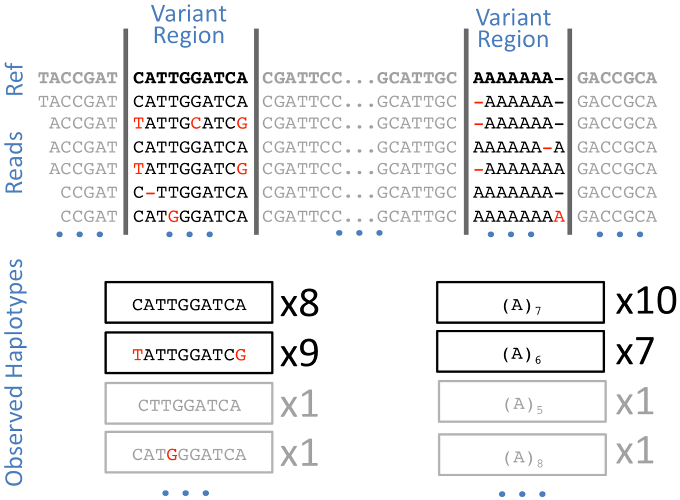
Single-cell methods



[Gawad et al., 2016]

- Often more like ‘a few’-cell sequencing
- Ploidy and viruses are still difficult
- Noisy/requiring lots of expensive samples

Reference based variant calling



Variant Region

Variant Region

Ref TACCGAT CATTGGATCA CGATTCC...GCATTGC AAAAAAA- GACCGCA

Reads

TACCGAT CATTGGATCA CGATTCC...GCATTGC -AAAAAA- GACCGCA

ACCGAT TATTGCATCG CGATTCC...GCATTGC -AAAAAA- GACCGCA

ACCGAT CATTGGATCA CGATTCC...GCATTGC AAAAAA-A GACCGCA

ACCGAT TATTGGATCG CGATTCC...GCATTGC -AAAAAAA GACCGCA

CCGAT C-TTGGATCA CGATTCC...GCATTGC AAAAAAA- GACCGCA

CCGAT CATGGGATCA CGATTCC...GCATTGC AAAAAAAA GACCGCA

Observed Haplotypes

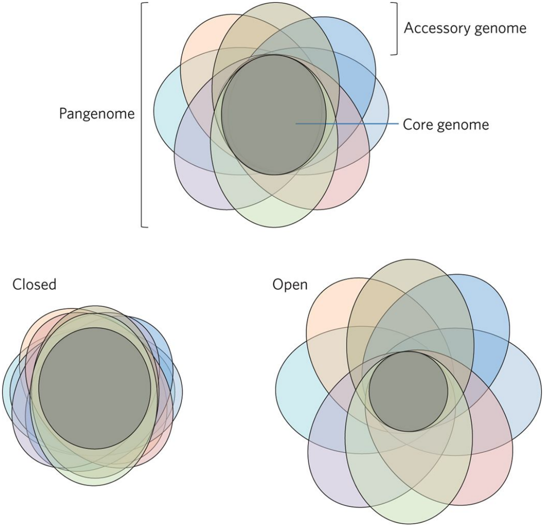
CATTGGATCA	x8	(A)$_7$	x10
TATTGGATCG	x9	(A)$_6$	x7
CTTGGATCA	x1	(A)$_5$	x1
CATGGGATCA	x1	(A)$_8$	x1

Read-mapping and variant calling *`bit.ly/2v6ZgTs`*

Choosing a reference?

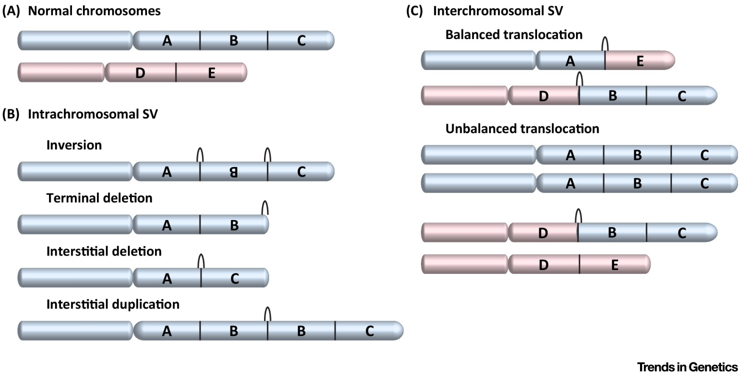
- Whatever other people used?
- Try a few and compare?
- Find closest sequence (ANI, MASH etc.)

Where this fails: outside core-genome



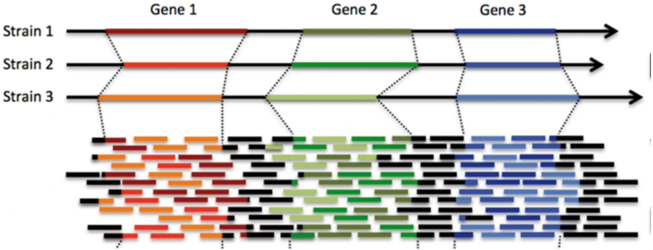
[McInerney et al., 2017]

Where this fails: structural variation



[Weckselblatt and Rudd, 2015]

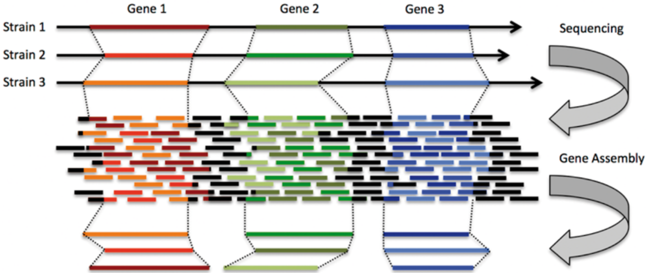
Where this fails: divergent strains



[Gregor et al., 2016]

Assembling variation

Assembly



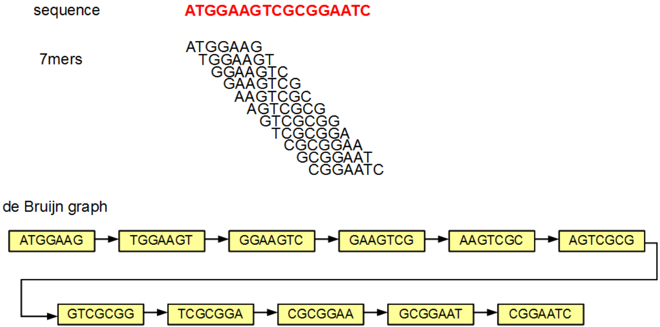
[Gregor et al., 2016]

de Bruijn graph recap

$$G = (V, E)$$

$$v \in V : v = k\text{-mer } x$$

$$\exists e(v \rightarrow v') \in E \iff x(1, k) = x'(0, k-1)$$



sequence ATGGAAGTCGCGGAATC

7mers
ATGGAAG
TGGAAGT
GGAAGTC
GAAGTCG
AAGTCGC
AGTCGCG
GTCGCGG
TCGCGGA
CGCGGAA
GCGGAAT
CGGAATC

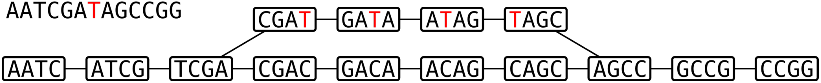
de Bruijn graph

ATGGAAG → TGGAAGT → GGAAGTC → GAAGTCG → AAGTCGC → AGTCGCG → GTCGCGG → TCGCGGA → CGCGGAA → GCGGAAT → CGGAATC

homolog.us/Tutorials/book4/p2.1.html

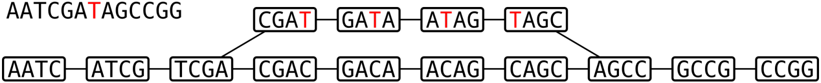
Resolving a simple SNP

AATCGACAGCCGG
AATCGATAGCCGG

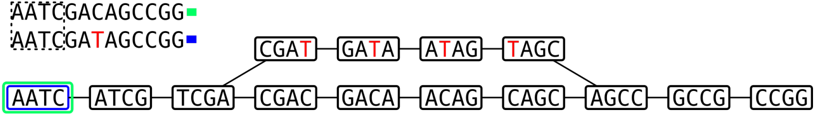


Keeping track of k-mers

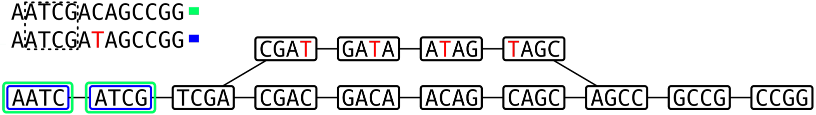
AATCGACAGCCGG
AATCGATAGCCGG



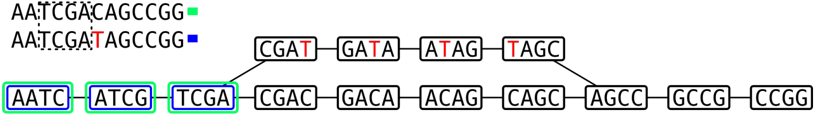
Keeping track of k-mers



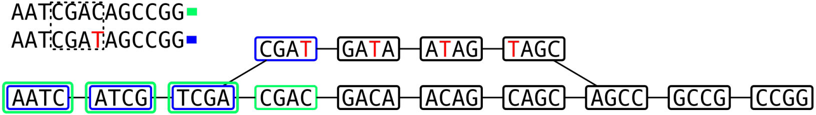
Keeping track of k-mers



Keeping track of k-mers



Keeping track of k-mers



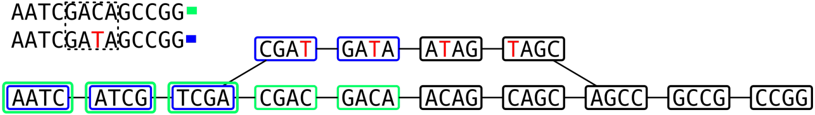
AATCGACAGCCGG 🟩

AATCGATAGCCGG 🟦

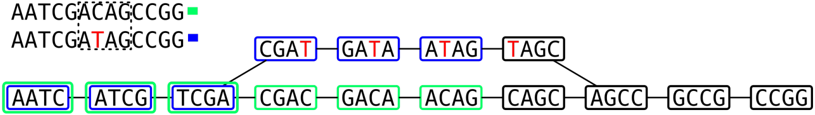
CGAT GATA ATAG TAGC

AATC ATCG TCGA CGAC GACA ACAG CAGC AGCC GCCG CCGG

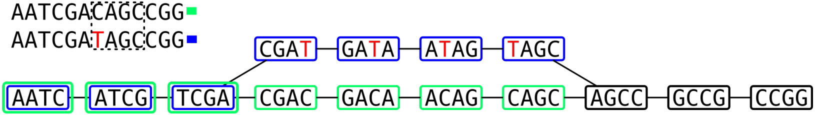
Keeping track of k-mers



Keeping track of k-mers



Keeping track of k-mers



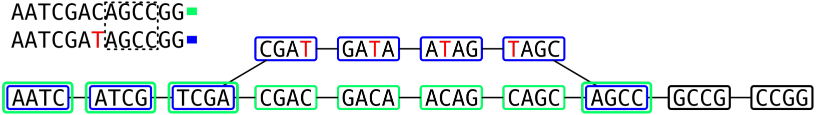
AATCGACAGCCGG 🟩

AATCGATAGCCGG 🟦

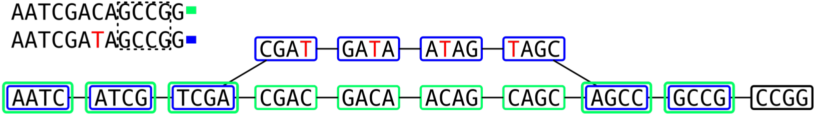
CGAT — GATA — ATAG — TAGC

AATC — ATCG — TCGA — CGAC — GACA — ACAG — CAGC — AGCC — GCCG — CCGG

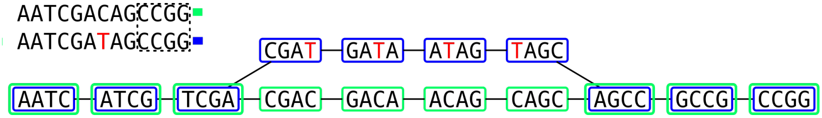
Keeping track of k-mers



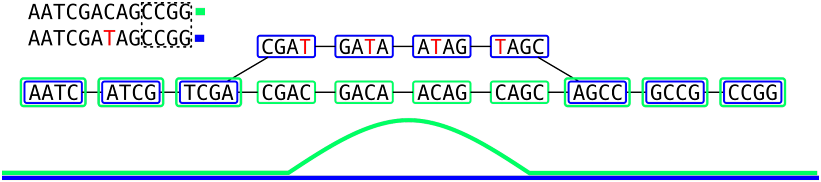
Keeping track of k-mers



Keeping track of k-mers



Keeping track of k-mers



Untangling the knot: coloured de Bruijn graphs

$$G = (V, E, C)$$

$$v \in V : v = k\text{-mer } x$$

$$\exists e(v \rightarrow v') \in E \iff x(1, k) = x'(0, k-1)$$

Given *n* samples/reads/k-mers:

$$\mathcal{C} = c_1, c_2, ...c_n$$

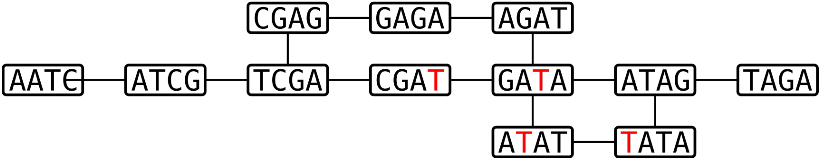
$$\forall v \in V : \exists c(v) \in \mathcal{C}$$

$$\forall e \in E : \exists c(e) \in \mathcal{C}$$

[Iqbal et al., 2012]

SNP in low-complexity region

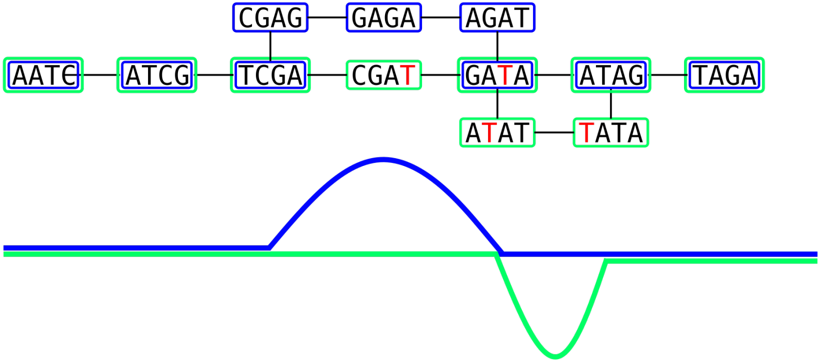
AATCGAGATAGA
AATCGATATAGA



SNP in low-complexity region

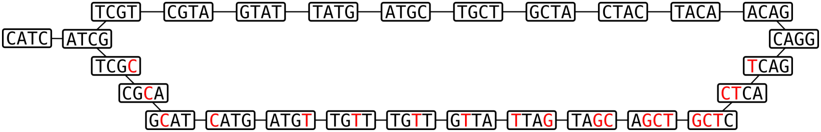
🟦 AATCGAGATAGA

🟩 AATCGATATAGA

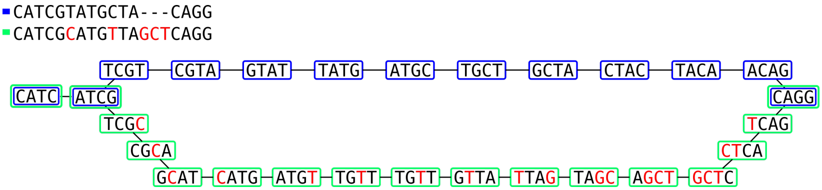


Clustered variants

```
CATCGTATGCTA---CAGG
CATCGCATGTTAGCTCAGG
```



Clustered variants



🟦 CATCGTATGCTA---CAGG

🟩 CATCGCATGTTAGCTCAGG

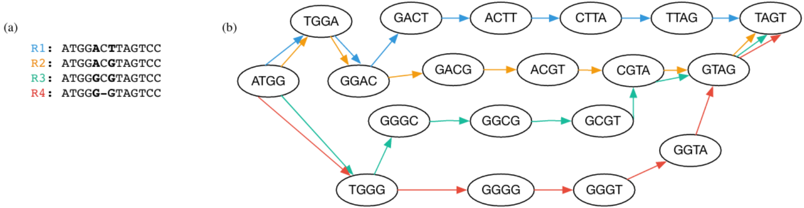
CATC — ATCG

TCGT — CGTA — GTAT — TATG — ATGC — TGCT — GCTA — CTAC — TACA — ACAG

CAGG

TCGC — CGCA — GCAT — CATG — ATGT — TGTT — TGTT — GTTA — TTAG — TAGC — AGCT — GCTC — CTCA — TCAG

Multiple reads-Multiple variants



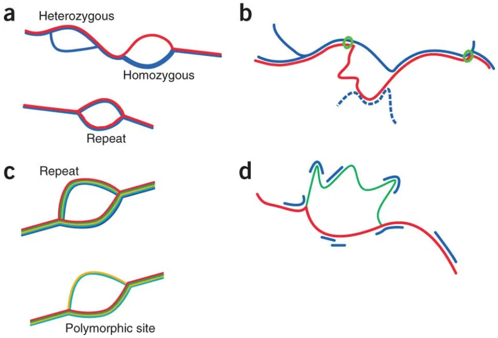
(a)

```
R1: ATGGACTTAGTCC
R2: ATGGACGTAGTCC
R3: ATGGGCGTAGTCC
R4: ATGGG-GTAGTCC
```

(b)

[Alipanahi et al., 2020]

Cortex Assembler



[Iqbal et al., 2013]

- Diploid individual (blue) with a reference sequence (red)
- Tracking longest contig
- Variant likelihood calculations based on coverage

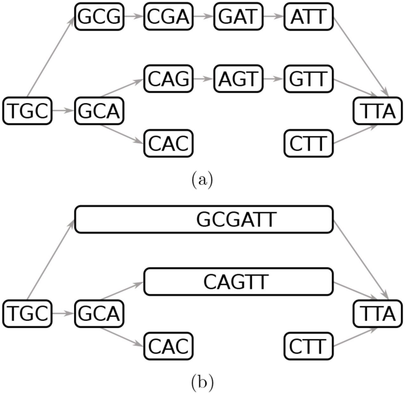
Messy details not covered

- Incorporating paired-end information
- Probabilistic colouring
- Details of using coverage and disambiguating error and variation

Downsides of coloured graphs: huge

- 88 metagenomic samples from Cattle feedlots [Noyes et al., 2016]
- 4 billion paired-end reads
- 41 billion 32-mers
- Storing k-mer:read pairing even as single bit would need 285 petabytes of space

Succinct/Compacted coloured de Bruijn graphs

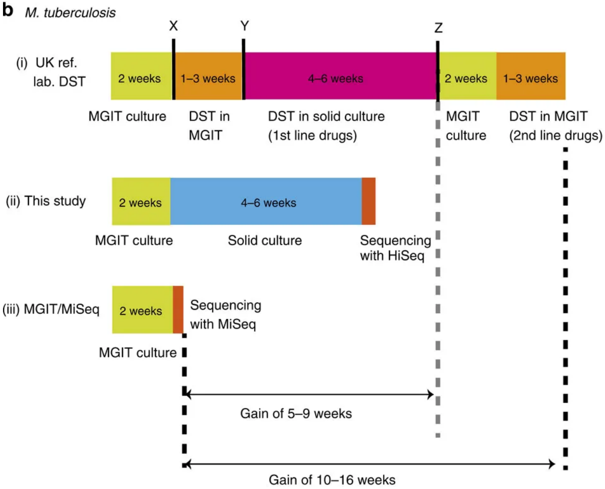


[Holley and Melsted, 2019]

- Compact maximal non-branching paths into untigs
- Use probabilistic data structures e.g. bloomfilters, minhash sketches, minimisers
- AKA make things more approximate but smaller!

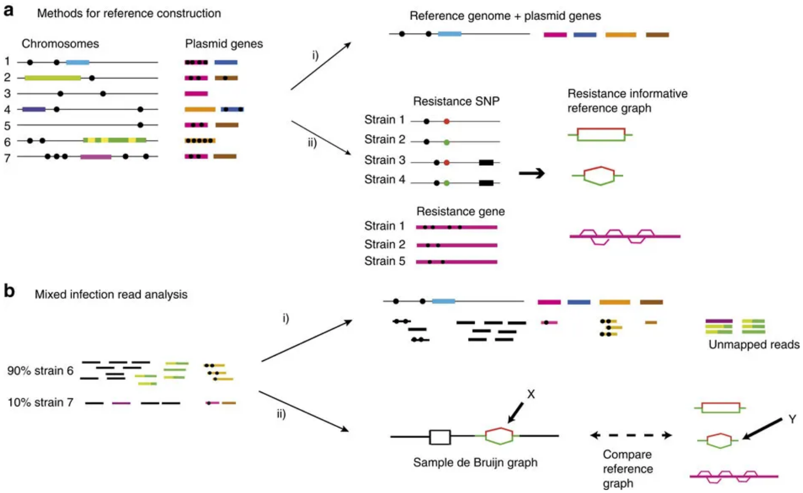
Using coloured de Bruijn graphs

Use-case: identifying AMR genes



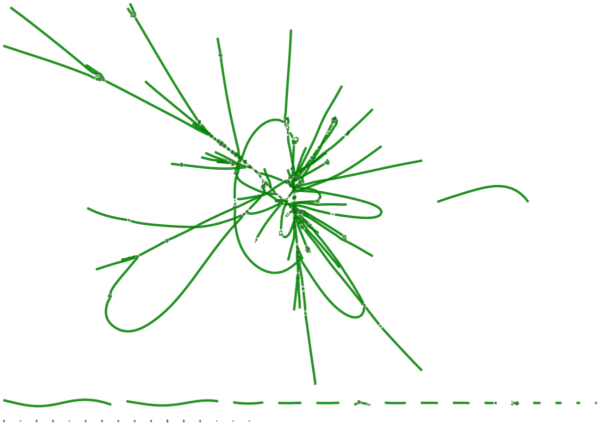
[Bradley et al., 2015]

Mykrobe



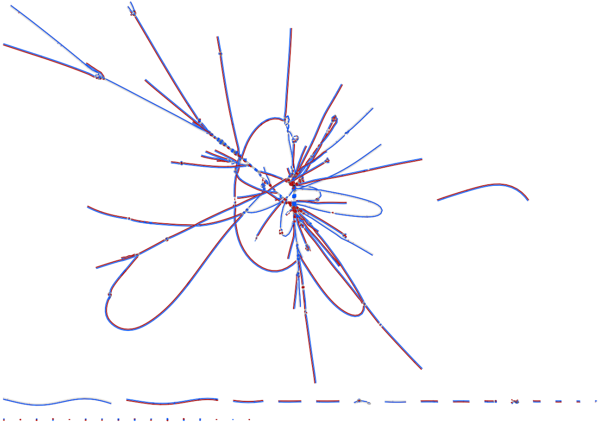
[Bradley et al., 2015]

‘Real Example’



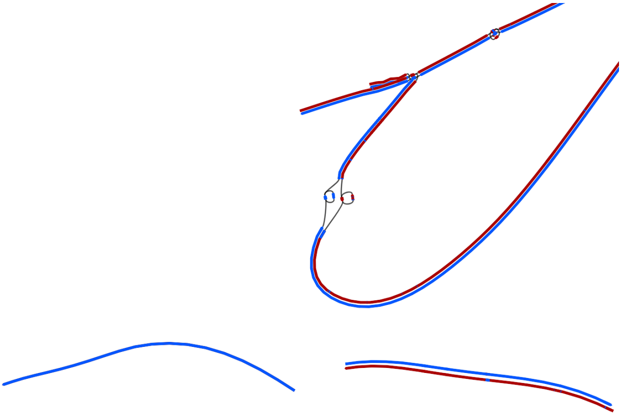
Multi-Strain *Mycobacterium tuberculosis* Infection Assembly Graph (one strain TDR the other totally susceptible)

‘Real Example’



Multi-Strain *Mycobacterium tuberculosis* Infection Assembly Graph (blue: TDR, red: totally susceptible)

‘Real Example’



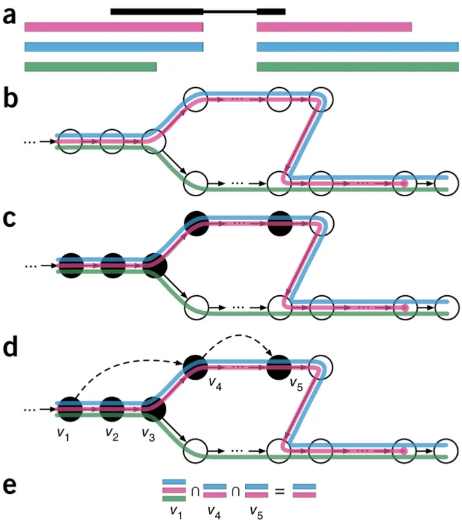
Multi-Strain *Mycobacterium tuberculosis* Infection Assembly Graph (blue: TDR, red: totally susceptible)

How well does this work in practice?

Predicting antimicrobial susceptibility in 3,206 *M. tuberculosis* samples [Hunt et al., 2019]

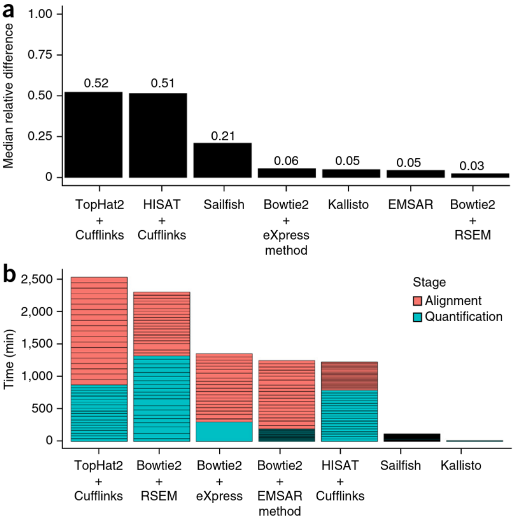
Method	Paradigm	MB	Min	Sensitivity	Specificity
Mykrobe	cdBG	1057	3.2	91.64	98.21
KvarQ	Motif	38	22.2	80.81	98.03
MTBSeq	BWT	12201	41.6	82.68	97.65
SPAdes	Assembly	18125	102.4	90.4	97.91

Transcript Quantification



[Bray et al., 2016]

Kallisto pseudoalignment



[Bray et al., 2016]

Taxonomy of graphs

Sequence graphs

- de Bruijn graphs:

Sequence graphs

- de Bruijn graphs:
    - de Bruijn graphs $G = (V, E)$

Sequence graphs

- de Bruijn graphs:
  - de Bruijn graphs $G = (V, E)$
  - compacted/succint de Bruijn graphs $G = (V, E)$ where $V =$ unitig

Sequence graphs

- de Bruijn graphs:
  - de Bruijn graphs $G = (V, E)$
  - compacted/succint de Bruijn graphs $G = (V, E)$ where $V =$ unitig
  - coloured de Bruijn graphs $G = (V, E, C)$

Sequence graphs

- de Bruijn graphs:
    - de Bruijn graphs $G = (V, E)$
    - compacted/succint de Bruijn graphs $G = (V, E)$ where $V =$ unitig
    - coloured de Bruijn graphs $G = (V, E, C)$
    - probabilistic coloured de Bruijn graphs $G = (V, E, C)$ where $C =$ p(C)

Sequence graphs

- de Bruijn graphs:
    - de Bruijn graphs $G = (V, E)$
    - compacted/succint de Bruijn graphs $G = (V, E)$ where $V = \text{unitig}$
    - coloured de Bruijn graphs $G = (V, E, C)$
    - probabilistic coloured de Bruijn graphs $G = (V, E, C)$ where $C = \text{p(C)}$
- Variation graphs:

Sequence graphs

- de Bruijn graphs:
  - de Bruijn graphs $G = (V, E)$
  - compacted/succint de Bruijn graphs $G = (V, E)$ where $V =$ unitig
  - coloured de Bruijn graphs $G = (V, E, C)$
  - probabilistic coloured de Bruijn graphs $G = (V, E, C)$ where $C =$ p(C)
- Variation graphs:
  - de Bruijn graph paths $G = (V, E, P)$ where $P =$ all the paths through G

Sequence graphs

- de Bruijn graphs:
    - de Bruijn graphs $G = (V, E)$
    - compacted/succint de Bruijn graphs $G = (V, E)$ where $V =$ unitig
    - coloured de Bruijn graphs $G = (V, E, C)$
    - probabilistic coloured de Bruijn graphs $G = (V, E, C)$ where $C =$ p(C)
- Variation graphs:
    - de Bruijn graph paths $G = (V, E, P)$ where $P =$ all the paths through G
    - compacted/coloured/probabilistic de Bruijn graphs $G = (V, E, C$ where $p_i = (v \in V : c(v) = c_i)$

Sequence graphs

- de Bruijn graphs:
    - de Bruijn graphs $G = (V, E)$
    - compacted/succint de Bruijn graphs $G = (V, E)$ where $V =$ unitig
    - coloured de Bruijn graphs $G = (V, E, C)$
    - probabilistic coloured de Bruijn graphs $G = (V, E, C)$ where $C =$ p(C)
- Variation graphs:
    - de Bruijn graph paths $G = (V, E, P)$ where $P =$ all the paths through G
    - compacted/coloured/probabilistic de Bruijn graphs $G = (V, E, C$ where $p_i = (v \in V : c(v) = c_i)$
    - looser sequence graphs $G = (V, E, P)$ where $V =$ any sequence and $E =$ adjacency in the sequence

Sequence graphs

- de Bruijn graphs:
  - de Bruijn graphs $G = (V, E)$
  - compacted/succint de Bruijn graphs $G = (V, E)$ where $V =$ unitig
  - coloured de Bruijn graphs $G = (V, E, C)$
  - probabilistic coloured de Bruijn graphs $G = (V, E, C)$ where $C =$ p(C)
- Variation graphs:
  - de Bruijn graph paths $G = (V, E, P)$ where $P =$ all the paths through G
  - compacted/coloured/probabilistic de Bruijn graphs $G = (V, E, C$ where $p_i = (v \in V : c(v) = c_i)$
  - looser sequence graphs $G = (V, E, P)$ where $V =$ any sequence and $E =$ adjacency in the sequence
- Other types of graph:

Sequence graphs

- de Bruijn graphs:
    - de Bruijn graphs $G = (V, E)$
    - compacted/succint de Bruijn graphs $G = (V, E)$ where $V =$ unitig
    - coloured de Bruijn graphs $G = (V, E, C)$
    - probabilistic coloured de Bruijn graphs $G = (V, E, C)$ where $C =$ p(C)
- Variation graphs:
    - de Bruijn graph paths $G = (V, E, P)$ where $P =$ all the paths through G
    - compacted/coloured/probabilistic de Bruijn graphs $G = (V, E, C$ where $p_i = (v \in V : c(v) = c_i)$
    - looser sequence graphs $G = (V, E, P)$ where $V =$ any sequence and $E =$ adjacency in the sequence
- Other types of graph:
    - Wheeler graphs (generalised structure)

Sequence graphs

- de Bruijn graphs:
  - de Bruijn graphs $G = (V, E)$
  - compacted/succint de Bruijn graphs $G = (V, E)$ where $V =$ unitig
  - coloured de Bruijn graphs $G = (V, E, C)$
  - probabilistic coloured de Bruijn graphs $G = (V, E, C)$ where $C =$ p(C)
- Variation graphs:
  - de Bruijn graph paths $G = (V, E, P)$ where $P =$ all the paths through G
  - compacted/coloured/probabilistic de Bruijn graphs $G = (V, E, C$ where $p_i = (v \in V : c(v) = c_i)$
  - looser sequence graphs $G = (V, E, P)$ where $V =$ any sequence and $E =$ adjacency in the sequence
- Other types of graph:
  - Wheeler graphs (generalised structure)
  - Breakpoint graphs [Lin et al., 2014] = coloured de Bruijn graphs

FASTG graph format

AATCGACAGCCGG

AATCGATAGCCGG

FASTG format (*http://fastg.sourceforge.net/*):

```
#FASTG:begin;
#FASTG:version=1.0:assembly_name="SNP example";
>chr1:chr1;
AATCGA[1:alt|C,T]CAGCCGG
```

GFA sequence graph format

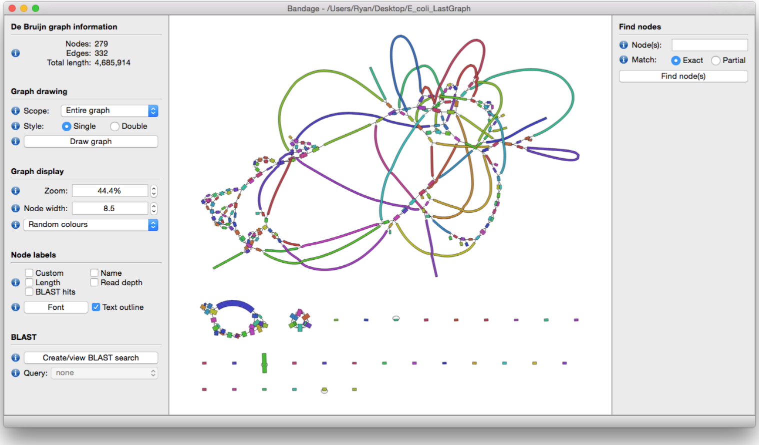
AATCGACAGCCGG

AATCGATAGCCGG

GFA format (*http://gfa-spec.github.io/GFA-spec/GFA2.html*):

```
H       VN:Z:1
S       1       AATCGA  LN:i:6
S       2       C       LN:i:1
S       3       T       LN:i:1
S       4       AGCCGG  LN:i:6
L       1       +       3       +       0M
L       1       +       2       +       0M
L       2       +       4       +       0M
L       3       +       4       +       0M
P       chr1a   1+,2+,4+        6M,1M,6M
P       chr1b   1+,3+,4+        6M,1M,6M
```

Bandage



Bandage - /Users/Ryan/Desktop/E_coli_LastGraph

De Bruijn graph information

Nodes: 279
Edges: 332
Total length: 4,685,914

Graph drawing

Scope: Entire graph

Style: ● Single ○ Double

Draw graph

Graph display

Zoom: 44.4%

Node width: 8.5

Random colours

Node labels

□ Custom □ Name
□ Length □ Read depth
□ BLAST hits

Font ☑ Text outline

BLAST

Create/view BLAST search

Query: none

Find nodes

Node(s):

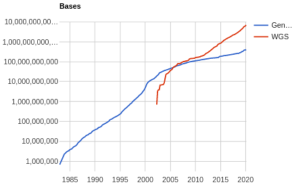
Match: ● Exact ○ Partial

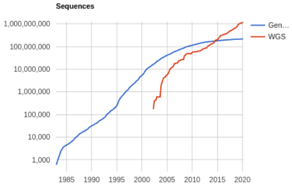
Find node(s)

[Wick et al., 2015]

Comparing sequences to databases more efficiently

Databases are growing rapidly





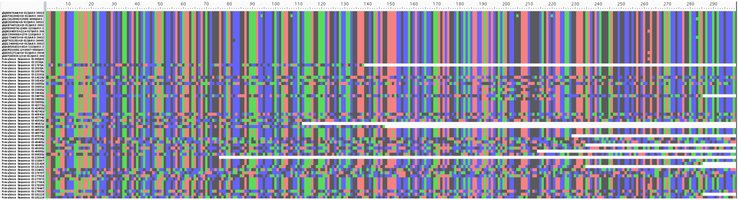
7 trillion bases in 1.2 billion sequences *https://www.ncbi.nlm.nih.gov/genbank/statistics/*

Searching databases

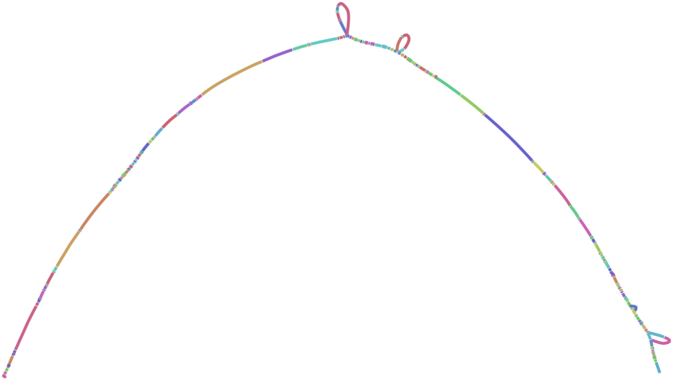
Our ability to search these databases approximately scales:

- Processing the query: (M = size of input sequence, K = word-size) $O(KM)$
- Scanning the database for partial matches (N = size of database) $O(KN)$
- Extending the match $O(MN)$

All NDM sequences



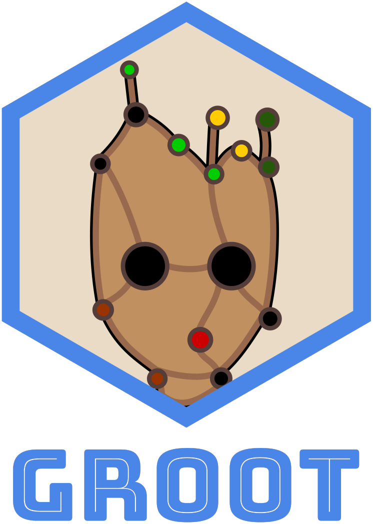
Collapsing NDM into a variation graph



How do we query these graphs?

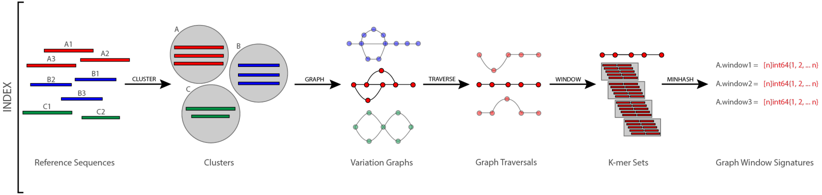
K-mer Indices

Groot



[Rowe and Winn, 2018]

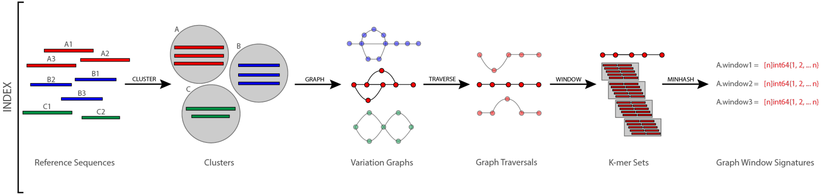
Creating and indexing variation graphs



[Rowe and Winn, 2018]

- Cluster database, align clusters, build variation graphs

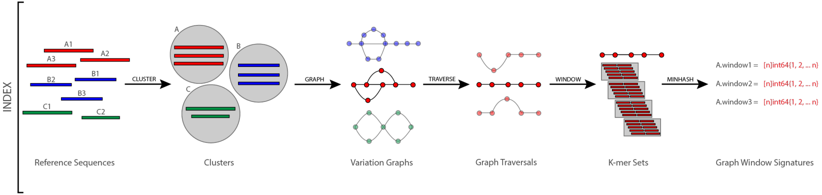
Creating and indexing variation graphs



[Rowe and Winn, 2018]

- Cluster database, align clusters, build variation graphs
- Traverse graph using sliding window and decomposed to k-mers

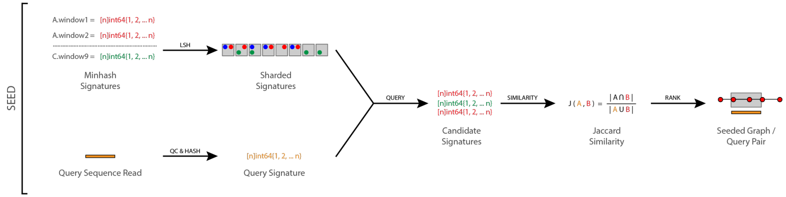
Creating and indexing variation graphs



[Rowe and Winn, 2018]

- Cluster database, align clusters, build variation graphs
- Traverse graph using sliding window and decomposed to k-mers
- Create a MinHash sketch for each window

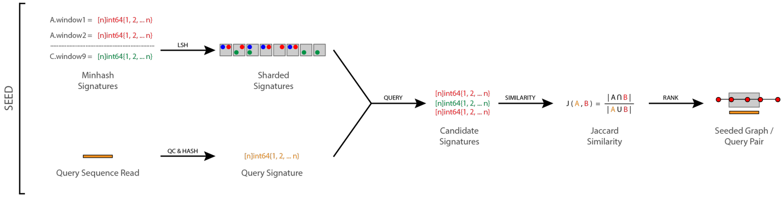
Seeding queries



[Rowe and Winn, 2018]

- Query reads are quality checked, trimmed and sketched

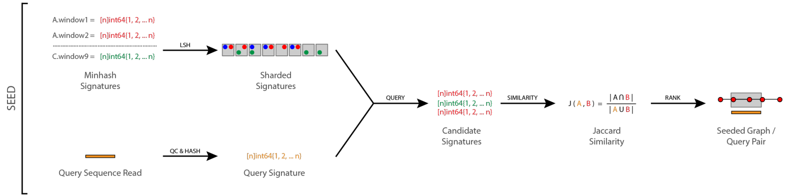
Seeding queries



[Rowe and Winn, 2018]

- Query reads are quality checked, trimmed and sketched
- Read sketch queried against the index using additional Locality Sensitive Hashing

Seeding queries



SEED

A.window1 = [n]int64{1, 2, ... n}

A.window2 = [n]int64{1, 2, ... n}

C.window9 = [n]int64{1, 2, ... n}

Minhash Signatures

LSH

Sharded Signatures

Query Sequence Read

QC & HASH

[n]int64{1, 2, ... n}

Query Signature

QUERY

[n]int64{1, 2, ... n}
[n]int64{1, 2, ... n}
[n]int64{1, 2, ... n}

Candidate Signatures

SIMILARITY

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

Jaccard Similarity

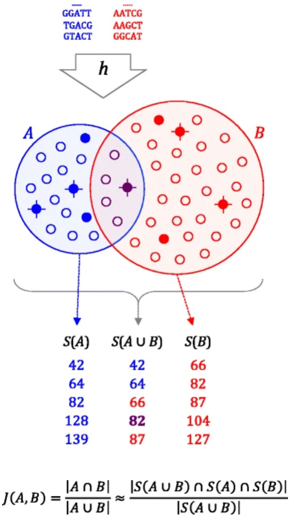
RANK

Seeded Graph / Query Pair

[Rowe and Winn, 2018]

- Query reads are quality checked, trimmed and sketched
- Read sketch queried against the index using additional Locality Sensitive Hashing
- Seeds are ranked by Jaccard Similarity estimates

MinHash Jaccard similarity

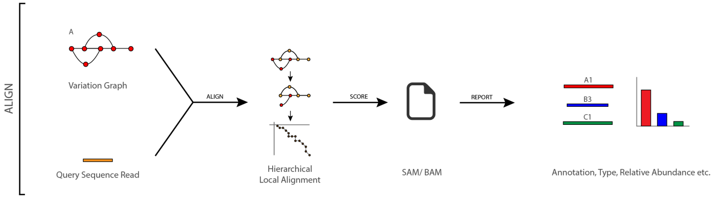


$S(A)$	$S(A\cup B)$	$S(B)$
42	42	66
64	64	82
82	66	87
128	82	104
139	87	127

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

[Ondov et al., 2016]

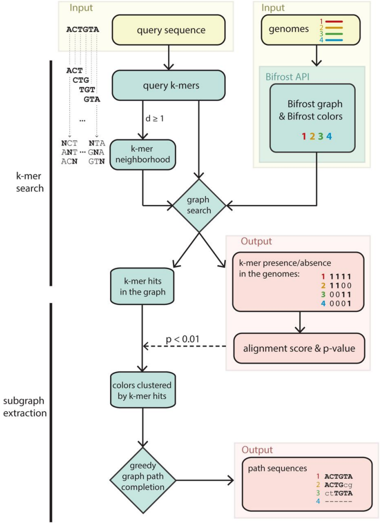
Aligning queries



[Rowe and Winn, 2018]

- Hierarchical local alignment
- Check exact matches, check partial exact, traverse graph
- Score traversal to classify an alignment (unique, perfect, partial, etc.)

BlastFrost: Similar but for bigger sequences!



[Luhmann et al., 2020]

Downsides of k-mer hashing-based methods

- Static table size means resizing is costly - bad for dynamic reference
- Search performance reduces when table capacity is reached
- Sensitive to k-mer size and sequencing error
- Aligns identical sequences multiple times
- Memory footprint can be high

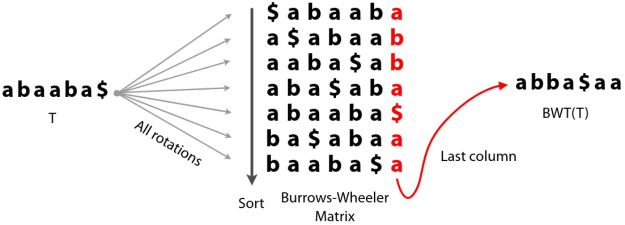
Burrows Wheeler Transform to the rescue

Disclaimer

This will skip over:

- FM-indices
- Wheeler graphs
- Fix-free parsing
- Note: BWT on graphs is still more theoretical CS than active use

Burrow-Wheeler Transform



abaaba$

T

All rotations

Sort

$abaaba
a$abaab
aaba$ab
aba$aba
abaaba$
ba$abaa
baaba$a

Burrows-Wheeler Matrix

Last column

abba$aa

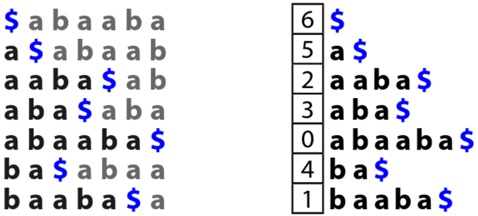
BWT(T)

Ben Langmead

Burrow-Wheeler Transform

$$BWT[i] = \begin{cases} T[SA[i]-1] & \text{if } SA[i] > 0 \\ \$ & \text{if } SA[i] = 0 \end{cases}$$

"BWT = characters just to the left of the suffixes in the suffix array"



```
$ a b a a b a
a $ a b a a b
a a b a $ a b
a b a $ a b a
a b a a b a $
b a $ a b a a
b a a b a $ a
```

BWM(T)

6	$
5	a $
2	a a b a $
3	a b a $
0	a b a a b a $
4	b a $
1	b a a b a $

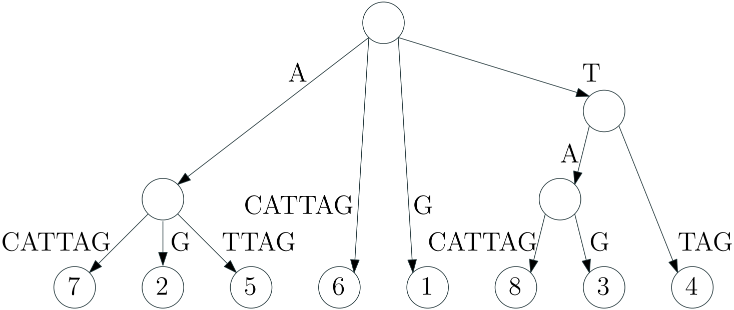
SA(T)

Ben Langmead

Searching the suffix tree for a motif

- Label leaves by string position/depth
- Search for pattern ”AT”
- Leaves in the subtree we reach are the location of that pattern

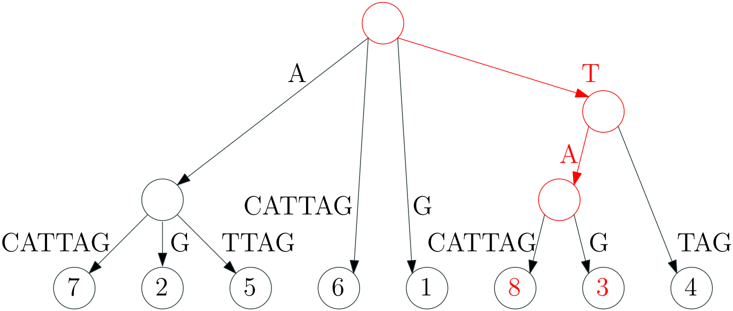
G	A	T	T	A	C	A	T
1	2	3	4	5	6	7	8



Searching the suffix tree for a motif

- Label leaves by string position/depth
- Search for pattern ”AT”
- Leaves in the subtree we reach are the location of that pattern

search for AT in G A T T A C A T
1 2 3 4 5 6 7 8

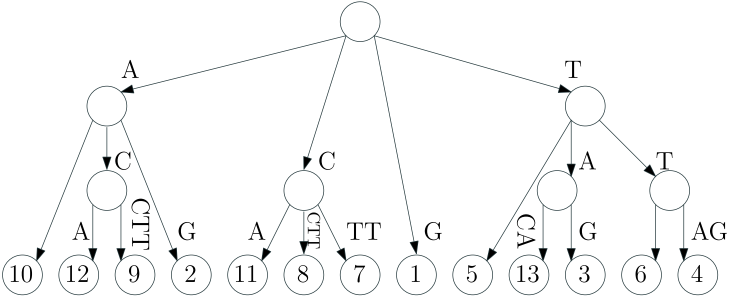


Travis Gagie

Extending this to collections of strings

- Order strings and label leaves by overall depths
- Search for pattern ”AT”
- Leaves in the subtree we reach are the location of that pattern

$$\mathcal{S} = \left\{ \begin{array}{ccccccccccccccc} \mathrm{G} & \mathrm{A} & \mathrm{T} & \mathrm{T} & , & \mathrm{T} & \mathrm{T} & \mathrm{C} & \mathrm{C} & \mathrm{A} & , & \mathrm{A} & \mathrm{C} & \mathrm{A} & \mathrm{T} \\ 1 & 2 & 3 & 4 & & 5 & 6 & 7 & 8 & 9 & & 10 & 11 & 12 & 13 \end{array} \right\}$$

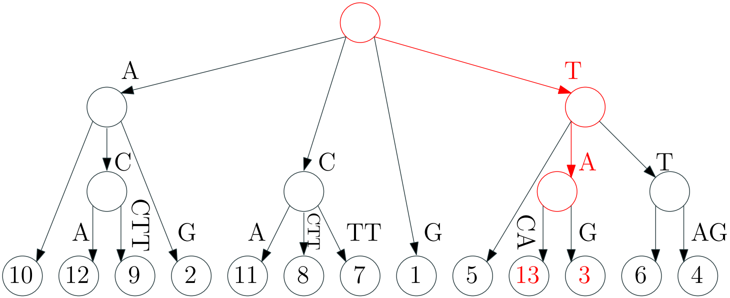


Extending this to collections of strings

- Order strings and label leaves by overall depths
- Search for pattern ”AT”
- Leaves in the subtree we reach are the location of that pattern

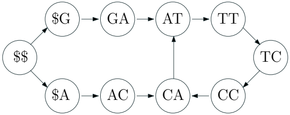
searching for AT in

$$\mathcal{S} = \left\{ \begin{array}{cccccccccccccccc} G & A & T & T & , & T & T & C & C & A & , & A & C & A & T \\ 1 & 2 & 3 & 4 & & 5 & 6 & 7 & 8 & 9 & & 10 & 11 & 12 & 13 \end{array} \right\}$$



Graphs are a collection of paths

If *G* is a de Bruijn graph (dBG) then we can sort the vertices into the co-lexicographic order of the strings labelling walks reaching them — all the strings labelling walks reaching a *k*-tuple α end with α — and so index *G*.

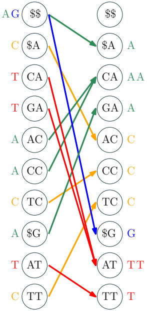


in-degrees: $0, 1, 2, 1, 1, 1, 1, 1, 2, 1$

$\to 0, 1, 2, 1^5, 2, 1$

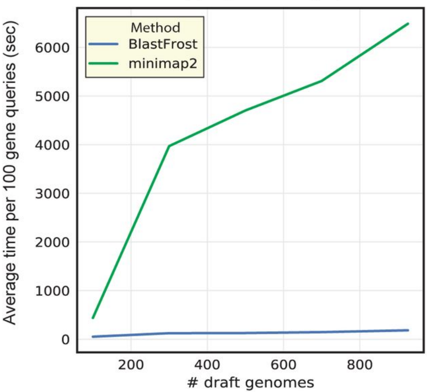
out-degrees: $2, 1, 1, 1, 1, 1, 1, 1, 1, 1 \to 2, 1^9$

BWT: AGCTTAACATC



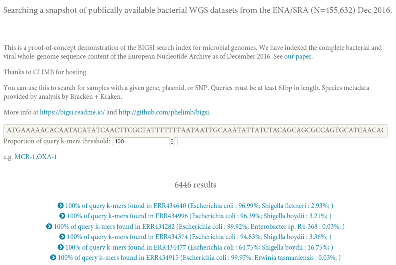
How well do reference variation graphs work?

BlastFrost scaling



[Luhmann et al., 2020]

BIGSI indexing of ENA



Searching a snapshot of publically available bacterial WGS datasets from the ENA/SRA (N=455,632) Dec 2016.

This is a proof-of-concept demonstration of the BIGSI search index for microbial genomes. We have indexed the complete bacterial and viral whole-genome sequence content of the European Nucleotide Archive as of December 2016. See our paper.

Thanks to CLIMB for hosting.

You can use this to search for samples with a given gene, plasmid, or SNP. Queries must be at least 61bp in length. Species metadata provided by analysis by Bracken + Kraken.

More info at https://bigsi.readme.io/ and http://github.com/phelimb/bigsi.

ATGAAAAACACAATACATATCAACTTCGCTATTTTTTTAATAATTGCAAATATTATCTACAGCAGCGCCAGTGCATCAACAC

Proportion of query k-mers threshold: 100

e.g. MCR-1,OXA-1

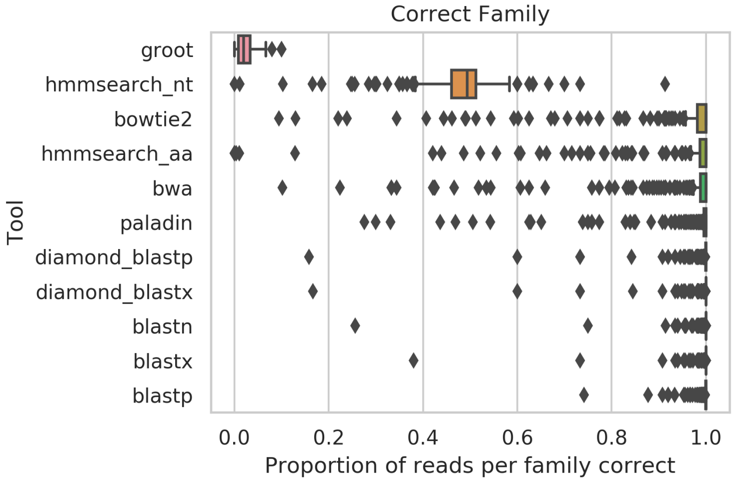
6446 results

- 100% of query k-mers found in ERR434640 (Escherichia coli : 96.99%; Shigella flexneri : 2.93%;)
- 100% of query k-mers found in ERR434996 (Escherichia coli : 96.39%; Shigella boydii : 3.21%;)
- 100% of query k-mers found in ERR434282 (Escherichia coli : 99.92%; Enterobacter sp. R4-368 : 0.03%;)
- 100% of query k-mers found in ERR434374 (Escherichia coli : 94.83%; Shigella boydii : 3.36%;)
- 100% of query k-mers found in ERR434477 (Escherichia coli : 64.75%; Shigella boydii : 16.75%;)
- 100% of query k-mers found in ERR434915 (Escherichia coli : 99.97%; Erwinia tasmaniensis : 0.03%;)

[Bradley et al., 2019]

- BIGSI: probabilistic coloured de Bruijn graph
- Indexing all bacterial, viral and parasitic reads in ENA (500,000 sets, 170TB of data)
- 1.5TB index that be queried near instantaneously

Beware: not always good



Simulated metagenome AMR family classification

Summary

Conclusions

- Coloured de Bruijn graphs represent variation within assemblies

Conclusions

- Coloured de Bruijn graphs represent variation within assemblies
- Powerful way of performing variation aware assembly (e.g. Cortex/Mykrobe)

Conclusions

- Coloured de Bruijn graphs represent variation within assemblies
- Powerful way of performing variation aware assembly (e.g. Cortex/Mykrobe)
- Succint cdBGs are a way to make them less space-intensive (e.g. Vari)

Conclusions

- Coloured de Bruijn graphs represent variation within assemblies
- Powerful way of performing variation aware assembly (e.g. Cortex/Mykrobe)
- Succint cdBGs are a way to make them less space-intensive (e.g. Vari)
- Variation graphs in general (including cdBGs like BIGSI) are an efficient way to represent large redundant reference databases

Conclusions

- Coloured de Bruijn graphs represent variation within assemblies
- Powerful way of performing variation aware assembly (e.g. Cortex/Mykrobe)
- Succint cdBGs are a way to make them less space-intensive (e.g. Vari)
- Variation graphs in general (including cdBGs like BIGSI) are an efficient way to represent large redundant reference databases
- K-mer hashing and probabilistic data-structures allow efficient querying of these references (e.g. groot, blastfrost)

Conclusions

- Coloured de Bruijn graphs represent variation within assemblies
- Powerful way of performing variation aware assembly (e.g. Cortex/Mykrobe)
- Succint cdBGs are a way to make them less space-intensive (e.g. Vari)
- Variation graphs in general (including cdBGs like BIGSI) are an efficient way to represent large redundant reference databases
- K-mer hashing and probabilistic data-structures allow efficient querying of these references (e.g. groot, blastfrost)
- K-mer methods highly parameter dependent and noise sensitive

Conclusions

- Coloured de Bruijn graphs represent variation within assemblies
- Powerful way of performing variation aware assembly (e.g. Cortex/Mykrobe)
- Succint cdBGs are a way to make them less space-intensive (e.g. Vari)
- Variation graphs in general (including cdBGs like BIGSI) are an efficient way to represent large redundant reference databases
- K-mer hashing and probabilistic data-structures allow efficient querying of these references (e.g. groot, blastfrost)
- K-mer methods highly parameter dependent and noise sensitive
- Burrows-Wheeler Transform (generalised as wheeler graphs) can also be used to index/query graphs (e.g. VG-toolkit GCSA/wheeler graphs)

Questions?

References i

(2018).
Computational pan-genomics: status, promises and challenges.
Briefings in bioinformatics, 19(1):118–135.

Alipanahi, B., Muggli, M. D., Jundi, M., Noyes, N. R., and Boucher, C. (2020).
Metagenome snp calling via read colored de bruijn graphs.
Bioinformatics.

Aury, J.-M., Jaillon, O., Duret, L., Noel, B., Jubin, C., Porcel, B. M., Ségurens, B., Daubin, V., Anthouard, V., Aiach, N., et al. (2006).
Global trends of whole-genome duplications revealed by the ciliate paramecium tetraurelia.
Nature, 444(7116):171–178.

References ii

- Bradley, P., Den Bakker, H. C., Rocha, E. P., McVean, G., and Iqbal, Z. (2019).
  Ultrafast search of all deposited bacterial and viral genomic data.
  Nature biotechnology, 37(2):152–159.

- Bradley, P., Gordon, N. C., Walker, T. M., Dunn, L., Heys, S., Huang, B., Earle, S., Pankhurst, L. J., Anson, L., De Cesare, M., et al. (2015).
  Rapid antibiotic-resistance predictions from genome sequence data for staphylococcus aureus and mycobacterium tuberculosis.
  Nature communications, 6(1):1–15.

- Bray, N. L., Pimentel, H., Melsted, P., and Pachter, L. (2016).
  Near-optimal probabilistic rna-seq quantification.
  Nature biotechnology, 34(5):525–527.

References iii

Didelot, X., Walker, A. S., Peto, T. E., Crook, D. W., and Wilson, D. J. (2016).
Within-host evolution of bacterial pathogens.
Nature Reviews Microbiology, 14(3):150.

Gawad, C., Koh, W., and Quake, S. R. (2016).
Single-cell genome sequencing: current state of the science.
Nature Reviews Genetics, 17(3):175.

Gregor, I., Schönhuth, A., and McHardy, A. C. (2016).
Snowball: strain aware gene assembly of metagenomes.
Bioinformatics, 32(17):i649–i657.

Holley, G. and Melsted, P. (2019).
Bifrost–highly parallel construction and indexing of colored and compacted de bruijn graphs.
BioRxiv, page 695338.

References iv

Hunt, M., Bradley, P., Lapierre, S. G., Heys, S., Thomsit, M., Hall, M. B., Malone, K. M., Wintringer, P., Walker, T. M., Cirillo, D. M., et al. (2019).
Antibiotic resistance prediction for mycobacterium tuberculosis from genome sequence data with mykrobe.
Wellcome Open Research, 4(191):191.

Iqbal, Z., Caccamo, M., Turner, I., Flicek, P., and McVean, G. (2012).
De novo assembly and genotyping of variants using colored de bruijn graphs.
Nature genetics, 44(2):226.

Iqbal, Z., Turner, I., and McVean, G. (2013).
High-throughput microbial population genomics using the cortex variation assembler.
Bioinformatics, 29(2):275–276.

References v

- Lin, Y., Nurk, S., and Pevzner, P. A. (2014).
  What is the difference between the breakpoint graph and the de bruijn graph?
  BMC genomics, 15(S6):S6.

- Luhmann, N., Holley, G., and Achtman, M. (2020).
  Blastfrost: Fast querying of 100,000 s of bacterial genomes in bifrost graphs.
  BioRxiv.

- McInerney, J. O., McNally, A., and O'connell, M. J. (2017).
  Why prokaryotes have pangenomes.
  Nature microbiology, 2(4):1–5.

References vi

- Noyes, N. R., Yang, X., Linke, L. M., Magnuson, R. J., Dettenwanger, A., Cook, S., Geornaras, I., Woerner, D. E., Gow, S. P., McAllister, T. A., et al. (2016).
  Resistome diversity in cattle and the environment decreases during beef production.
  Elife, 5:e13195.

- Ondov, B. D., Treangen, T. J., Melsted, P., Mallonee, A. B., Bergman, N. H., Koren, S., and Phillippy, A. M. (2016).
  Mash: fast genome and metagenome distance estimation using minhash.
  Genome biology, 17(1):132.

- Redd, A. D., Quinn, T. C., and Tobian, A. A. (2013).
  Frequency and implications of hiv superinfection.
  The Lancet infectious diseases, 13(7):622–628.

References vii

Rowe, W. P. and Winn, M. D. (2018).
Indexed variation graphs for efficient and accurate resistome profiling.
Bioinformatics, 34(21):3601–3608.

Weckselblatt, B. and Rudd, M. K. (2015).
Human structural variation: mechanisms of chromosome rearrangements.
Trends in Genetics, 31(10):587–599.

Wick, R. R., Schultz, M. B., Zobel, J., and Holt, K. E. (2015).
Bandage: interactive visualization of de novo genome assemblies.
Bioinformatics, 31(20):3350–3352.