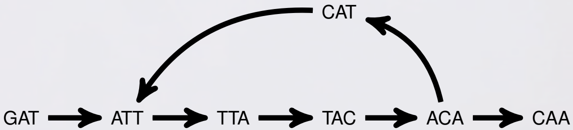
On the representation of de Bruijn Graphs

Rayan Chikhi (Penn State), Antoine Limasset (ENS),
Shaun Jackman (BCGSC), Jared Simpson (OICR),
Paul Medvedev (Penn State)

RECOMB 2014

de Bruijn Graph

```
read:   GATTACATTACAA
k-mers: GAT
(k=3)    ATT
          TTA
           ...
```

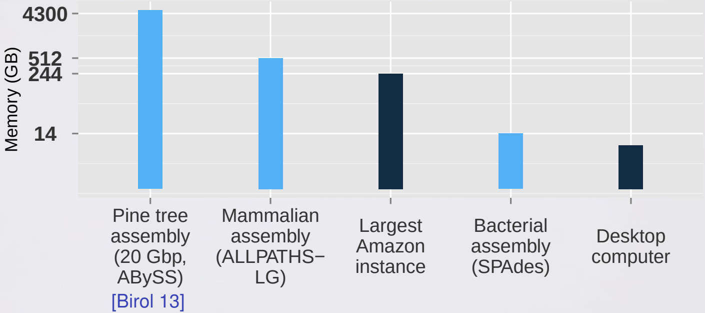


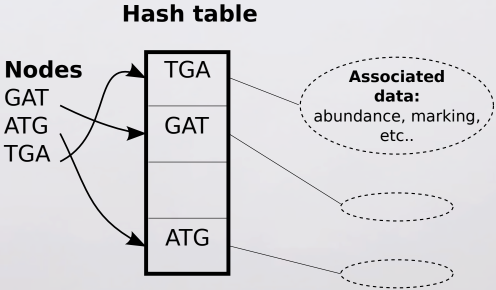
Nodes: *k*-mers

Edges: $(k-1)$-overlaps

- *de novo* assembly of sequencing data
  - DNA: Velvet, ALLPATHS-LG, SOAPdenovo2, SPAdes, . . .
  - RNA: Trinity, Oases
  - meta-DNA, meta-RNA

dBGs require a lot of memory

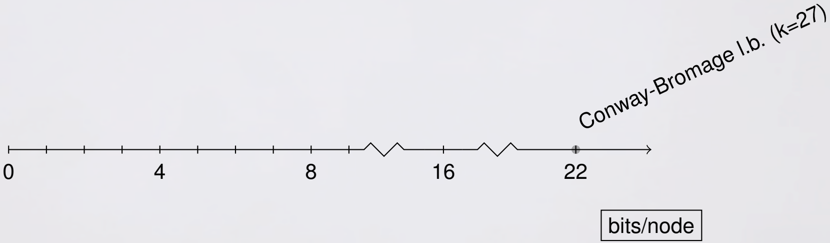




Space needed to represent the dBG

Lower bound for dBG data structures

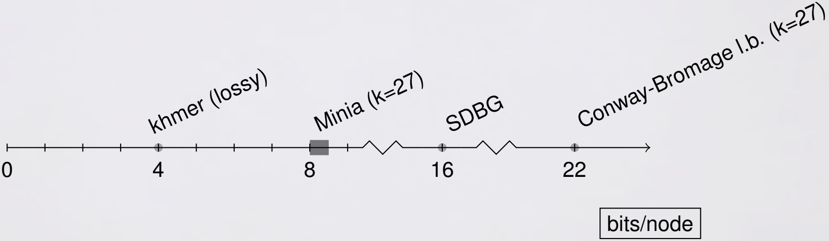
[Conway, Bromage 11]



Space needed to represent the dBG

Lower bound for dBG data structures

[Conway, Bromage 11]



Memory-efficient dBG data structures:

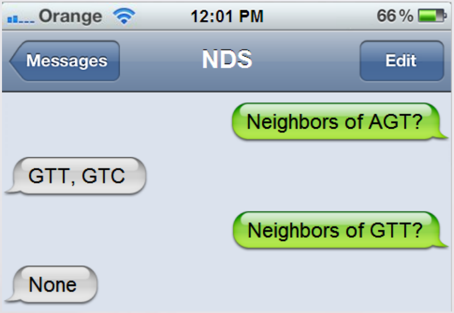
khmer	Bloom filter	[Pell et al. 11]
Minia	BF \ false positives	[Chikhi, Rizk 12], [Salikhov et al. 13]
SDBG	XBW + rank/select	[Bowe et al. 12]

Why are they doing better? → not all operations are supported

Navigational data structures

	NDS	Membership (e.g. hash table)
Traverse dBG from known nodes	✓	✓
Query **membership** of arbitrary nodes	*x*	✓
Enumerate nodes	*x*	✓

NDS has undefined behavior if query node not present.

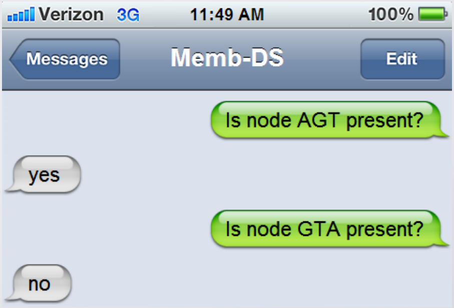


Neighbors of AGT?

GTT, GTC

Neighbors of GTT?

None



Is node AGT present?

yes

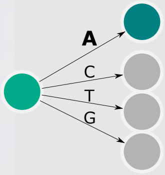
Is node GTA present?

no

Minia and SDBG are **NDS** but **not Memb-DS**

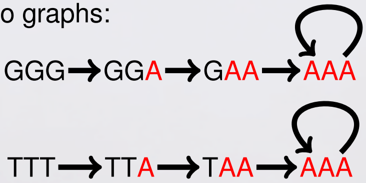
Why does a NDS beat Conway-Bromage LB?

Consider this example NDS



"The neighbor of x is $x_{2\ldots k}\mathbf{A}$"

Valid for these two graphs:



GGG → GGA → GAA → AAA

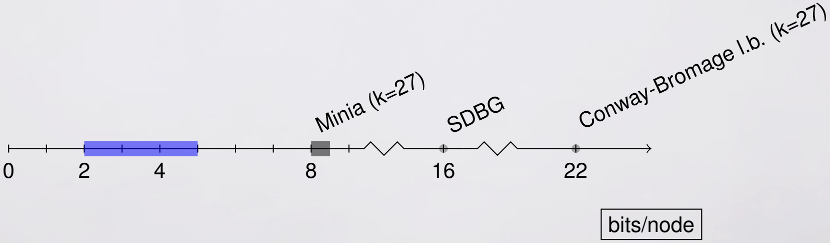
TTT → TTA → TAA → AAA

1 NDS $\longleftrightarrow$ >1 dBGs

1 Membership DS $\longleftrightarrow$ 1 dBG

This work

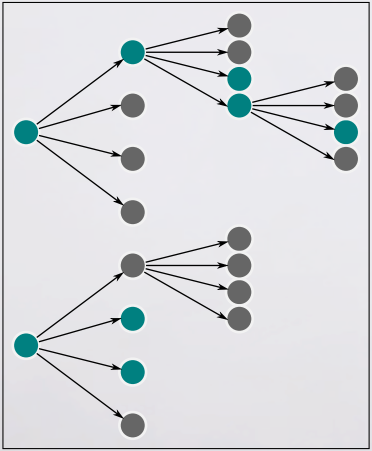
1. Lower bounds in the NDS model
2. Construction algorithm for upper bound
3. Parameterized upper bound



NDS lower bound

Theorem

A NDS needs at least 3.24 *bits/k-mer.*



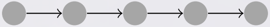
Proof sketch:

1. Let $n > 0$ k-mers
2. Construct $N = 2^{3.24n}$ graphs
3. Suppose NDS needs $< \log(N)$ bits
4. Two graphs have the same NDS (pigeonhole principle), contradiction

→ But these are not genome graphs.

NDS lower bound for linear graphs

Best case for genomes: linear graphs



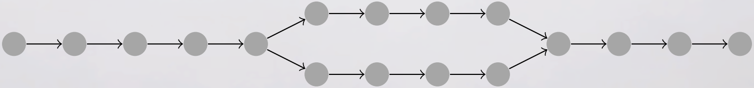
Theorem

A NDS for a linear graph needs at least 2 bits/k-mer.

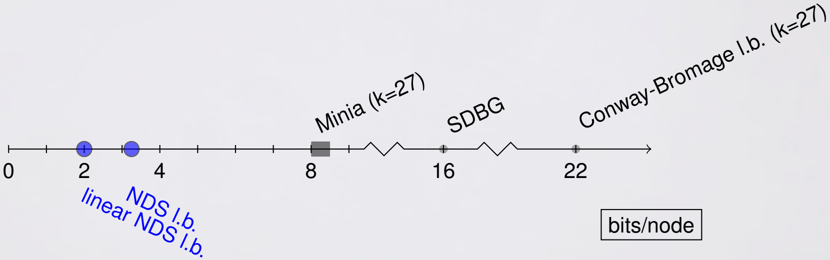
Proof sketch: (same technique, different family of graphs)

- 2^n linear dBGs have n k-mers [Gagie 12]
- Pigeonhole principle

dBGs of genomes are “linear-like”:



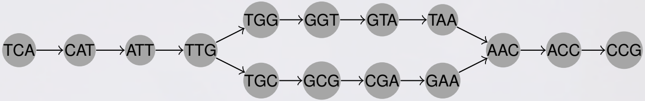
Big Picture



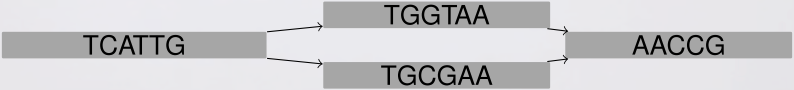
1. Lower bounds
2. **Construction of upper bound**
3. Upper bound

Constructing the compacted dBG

Input:



After **compaction**, output:



Constraint: construction memory < NDS

How? (no existing algorithms)

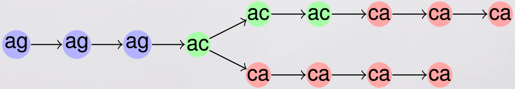
Minimizers

ℓ-**minimizer**: smallest substring of length ℓ

e.g. ($\ell = 2$, lexicographical order)

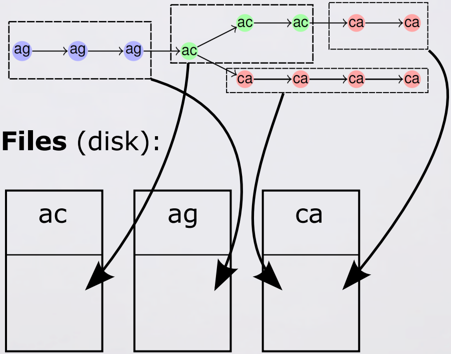
```
TGACGGG
 GACGGGT
  ACGGGTA
   CGGGTCA
    GGGTCAG
     GGTCAGA
```

dBG partitioning w.r.t minimizers:



BCALM algorithm

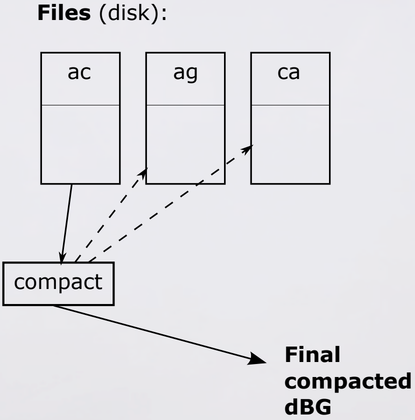
Initial step:



Nodes are partitioned to files on disk according to their minimizer

BCALM algorithm

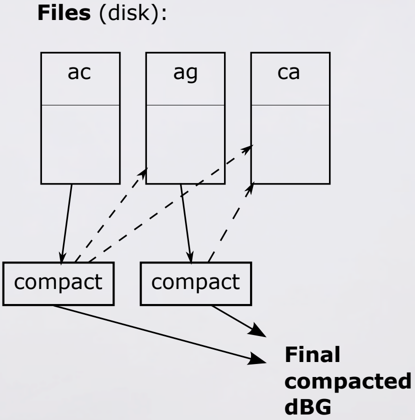
High-level overview of main loop:



Intermediate compactions are redistributed (depending on minimizers of left/right ($k-1$)-mers).

BCALM algorithm

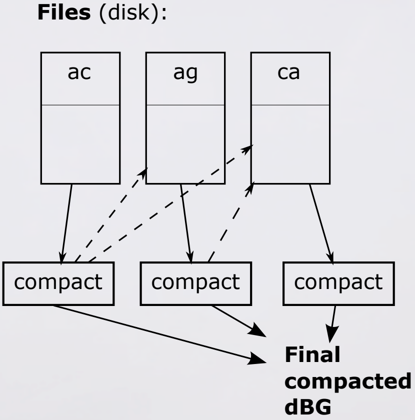
High-level overview of main loop:



Intermediate compactions are redistributed (depending on minimizers of left/right $(k-1)$-mers).

BCALM algorithm

High-level overview of main loop:

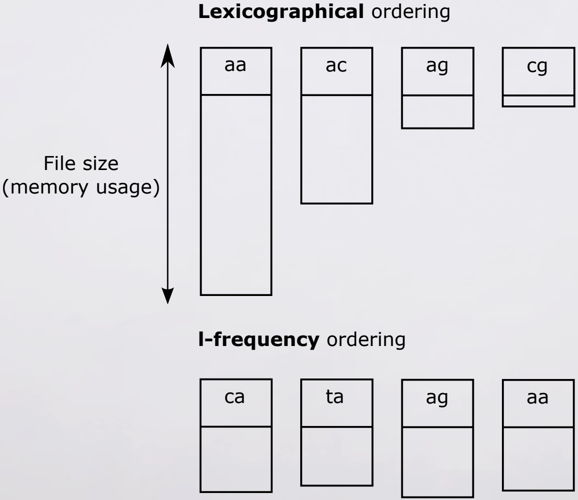


Careful redistribution ensures correctness of final graph (proof in paper)

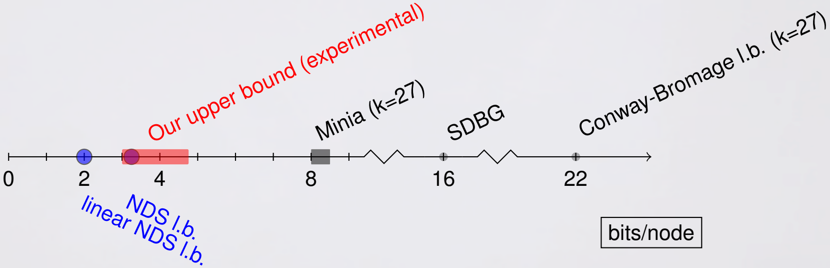
BCALM results

Whole human Illumina dataset, 46x coverage, $2.5 \cdot 10^9$ filtered 55-mers

Memory: **43 MB** Time: **12** hours

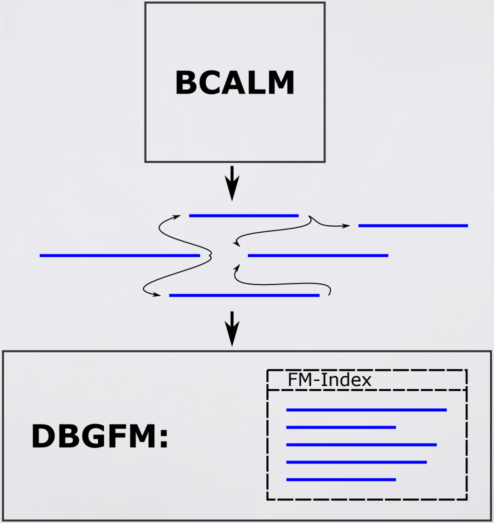


Big Picture



1. Lower bounds
2. Construction of simple paths
3. **Parameterized upper bound**

Membership DS for linear-like dBGs

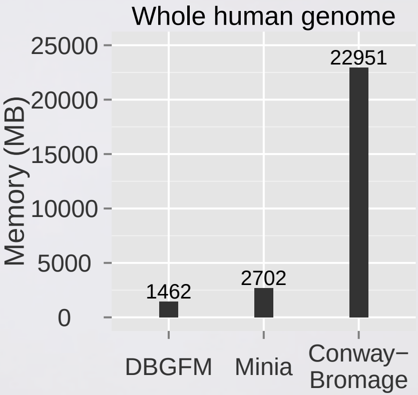


DBGFM Memory usage (bits/node) (n k-mers, c_k simple paths)

$$\approx 2n + O(kc_k)$$

DBGFM Results

Dataset: NA18507
Illumina 100bp
46x coverage
$2.5 \cdot 10^9$ filtered 55-mers
(DSK)



Integration in the **ABySS** assembler: [Simpson 09]

Chr. 14, $k = 55$	hash table	DBGFM
Memory	2.4 GB	**700** MB
Time	14 mins	21 mins

Conclusion / Perspectives

Navigational data structures:

- Model for recent dBG data struct.
- Lower bound: 3.24 bits/*k*-mer
- Gap with known upper bounds (16)

BCALM:

- dBG compaction in negligible memory
  - `http://github.com/Malfoy/bcalm`
- Reduce memory burden of other seq. analysis

DBGFM:

- dBG in $\frac{\text{genome size}}{2}$ bytes: 1.5 GB for human (experimental)
- 2× improvement from Minia
  - `http://github.com/jts/dbgfm`
  - `http://github.com/bcgsc/abyss/tree/dbgfm`