

On the representation of de Bruijn Graphs

Rayan Chikhi

joint work with P. Medvedev, A. Limasset, S. Jackman, J. Simpson

Univ. Lille

ALEA 2016

de Bruijn Graph

sequence: GATTACATTACAA

k-mers: GAT

(k=3) ATT

TTA

...

Nodes: k -mers (words of length k)

Edges: exact suffix-prefix overlaps of length $k - 1$



Usages:

- Bioinformatics
 - ▶ *de novo* assembly of sequencing data
- Distributed applications

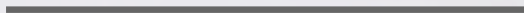
Genome sequencing



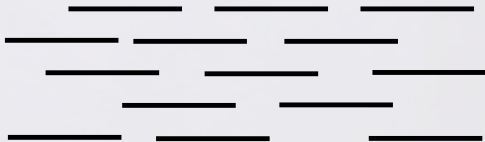
```
>1  
ACTGA..  
>2  
TGTAG..  
...
```

Genome assembly

genome
(not known)

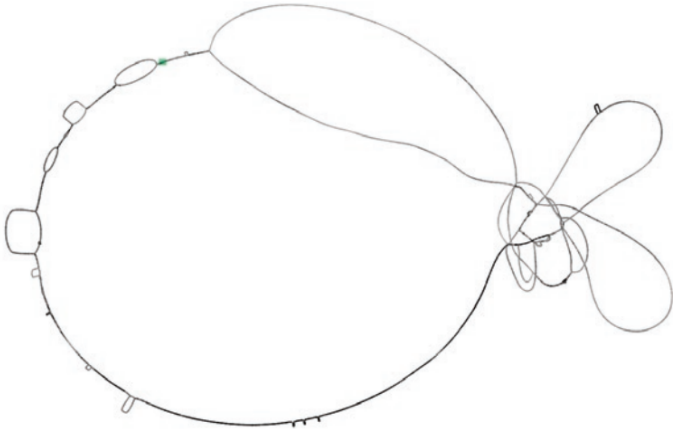


substrings from the genome, but position unknown



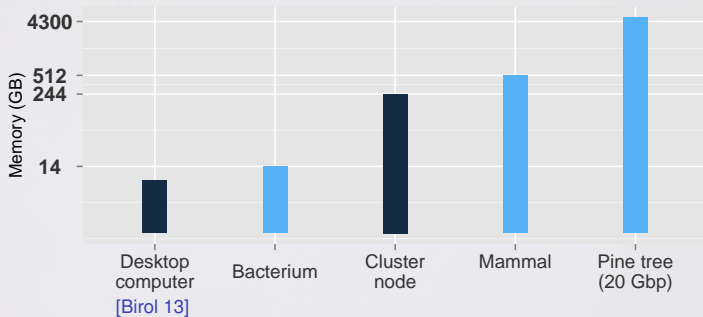
Using a de Bruijn
graph of all k-mers
present in the
substrings



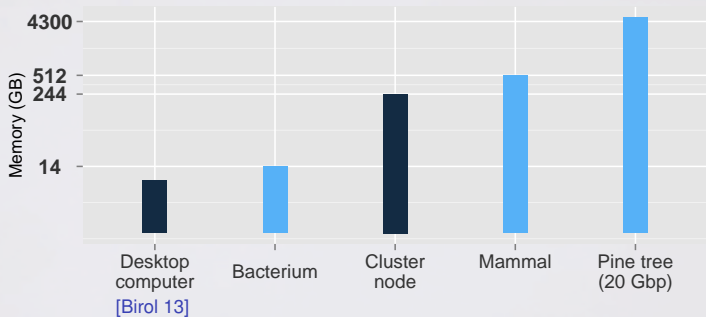


Bacterial genome assembled with a de Bruijn graph.

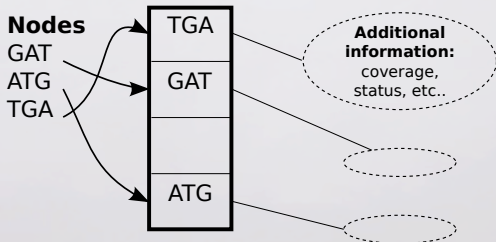
dBGs require a lot of memory



dBGs require a lot of memory



Hash table



How to encode the de Bruijn graph using as little space as possible?

nodes only: $\{GAT, ATT, \dots\}$

(human genome: $k = 75$, $n = 3 \cdot 10^9$ k -mers)

- Explicit list:

$2k \cdot n$ bits

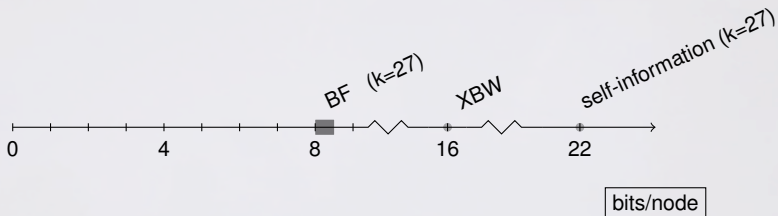
56 GB

- Self-information of n nodes:
[Conway, Bromage 11]

$$\log_2 \left(\binom{4^k}{n} \right) \text{ bits}$$

44 GB

Recent techniques



- Bloom filter of nodes (w/ tricks) [Chikhi, Rizk 12], [Salikhov et al. 13]
- XBW (Burrows-Wheeler for trees) variant [Bowe et al. 12]

Why are they doing better?

→ different types of data structures

Data structures

A **membership** data structure is a pair of algorithms
(*const*, *contains_node*), where:

$$data \leftarrow const(G)$$

contains_node(*data*, *kmer*) returns {true, false} whether $kmer \in G$

A **navigational** data structure is (*const*, *neighbors*), where:

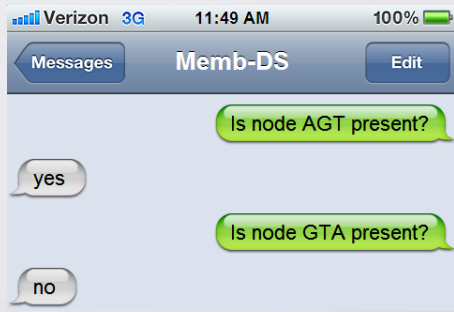
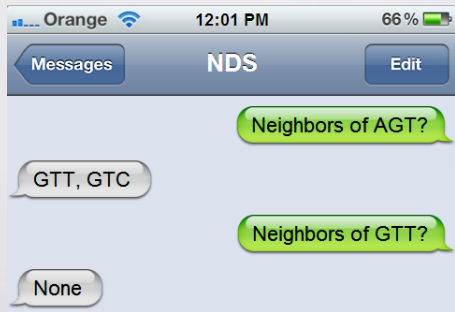
$$data \leftarrow const(G)$$

neighbors(*data*, *kmer*) returns the neighbors of *kmer* in *G*

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.



Recent techniques are **NDS** but **not Membership DS**

Why a NDS "beats" the self-information

Consider this example NDS when $k = 3$

"For each node $x = x_1x_2x_3$,

out-neighbor: $x_2x_3x_1$

in-neighbor: $x_3x_1x_2$ "

Valid for these two graphs:



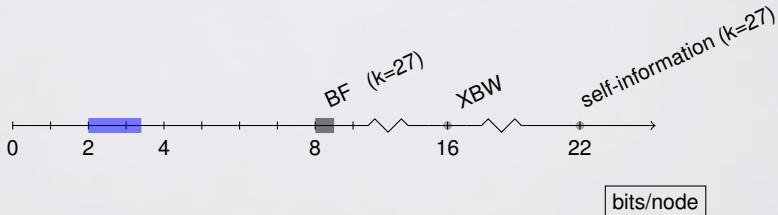
So,

1 NDS \longleftrightarrow >1 dBGs

1 Membership DS \longleftrightarrow 1 dBG

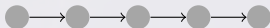
Lower bounds

We seek DBG representation lower bounds in the NDS model.



NDS lower bound for linear graphs

Linear graphs



Theorem

NDS for linear graphs need at least 2 bits/k-mer of space.

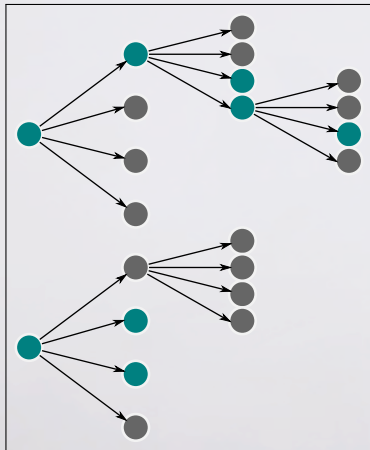
Proof sketch:

- Number of DNA strings that have n distinct k -mers and start with same k -mer: $\approx 2^{2n}$ [Gagie 12]
- Number of linear DBGs with n nodes and same left-most node: $\approx 2^{2n}$
- Suppose NDS needs $< 2n$ bits,
- Two graphs have the same NDS (pigeonhole principle)

NDS lower bound

Theorem

NDS need at least 3.24 bits/k-mer.



Proof sketch:

1. Construct a large family of N graphs, such that for any two graphs, \exists k-mer that appears in both graphs but with different neighbors.
2. Suppose NDS needs $< \log(N)$ bits
3. Two graphs have the same NDS (pigeonhole principle), contradiction

Our construction has $N = 2^{3.24n}$

Conclusion / Perspectives

Navigational data structures:

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

Open questions:

- Closing the gap above
- Entropy-compressed DBG representations

Contact/references:

- *On the Representation of de Bruijn Graphs*, 2014
- `rayan.chikhi@univ-lille1.fr`
- `http://rayan.chikhi.name`