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

Guillaume Holley* and Páll Melsted

Holley and Melsted Genome Biology 2020
https://doi.org/10.1186/s13059-020-02135-8

https://github.com/pmelsted/bifrost

Ömer Faruk Özdemir

# De Bruijn graphs

OLC framework is the defacto standard but,
De Bruijn is still used to assemble and correct long reads

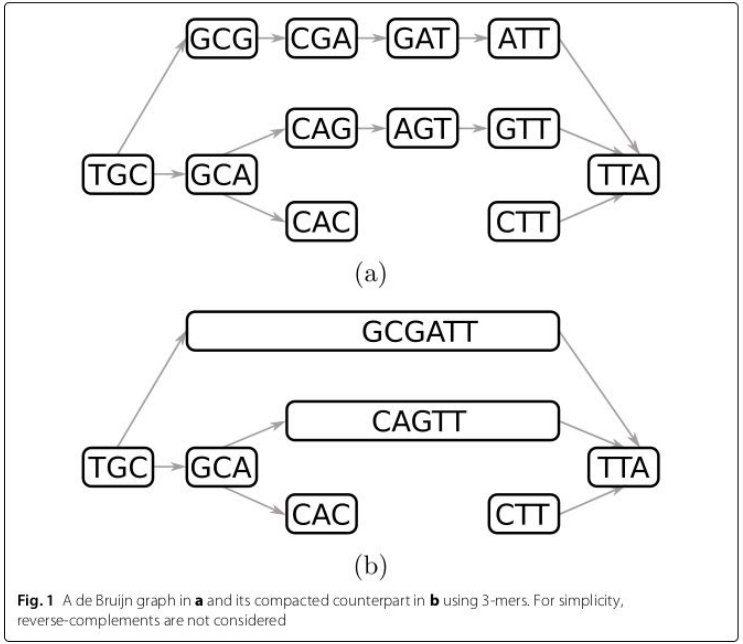


**Fig. 1** A de Bruijn graph in **a** and its compacted counterpart in **b** using 3-mers. For simplicity, reverse-complements are not considered

# De Bruijn graphs

OLC framework is the defacto standard but,
De Bruijn is still used to assemble and correct long reads

Problems such as

- de novo transcriptome assembly
- variant calling
- short read compression
- short read correction
- long read correction
- short read mapping

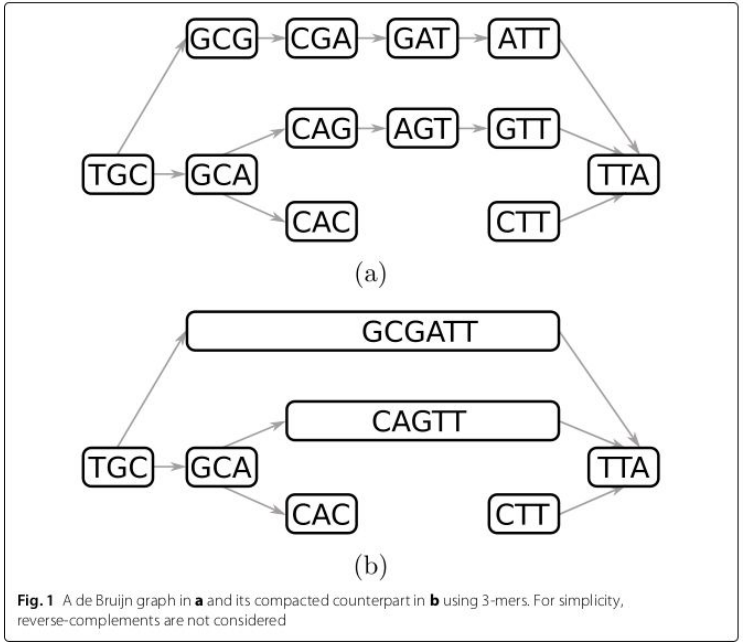


**Fig. 1** A de Bruijn graph in **a** and its compacted counterpart in **b** using 3-mers. For simplicity, reverse-complements are not considered

# De Bruijn graphs

OLC framework is the defacto standard but,
De Bruijn is still used to assemble and correct long reads

Problems such as

- de novo transcriptome assembly
- variant calling
- short read compression
- short read correction
- long read correction
- short read mapping

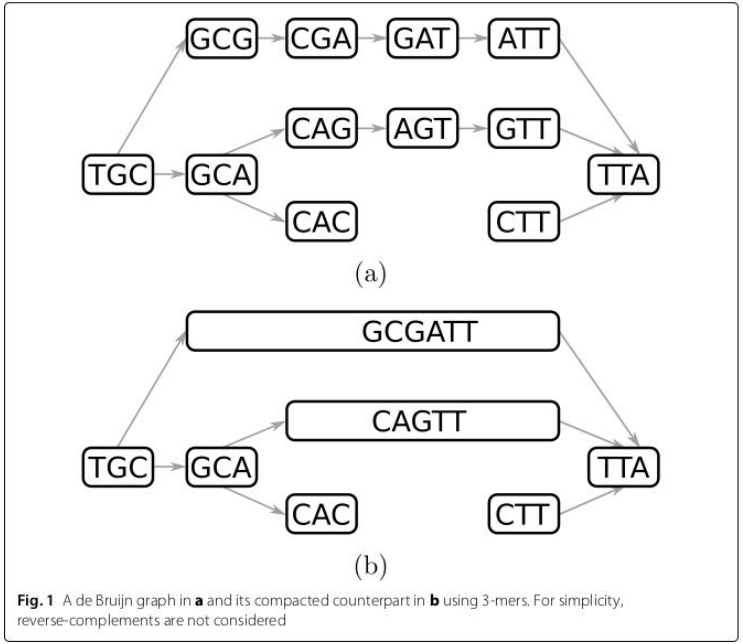


**Fig. 1** A de Bruijn graph in **a** and its compacted counterpart in **b** using 3-mers. For simplicity, reverse-complements are not considered

The colored de Bruijn graph keeps track of the source of each vertex in the graph

- assembly and genotyping
- pan-genomics
- variant calling
- transcript quantification

# Objective

- Problem: Uncompacted graph has trouble fitting in memory.

- Solution: A parallel and memory-efficient algorithm enabling construction without uncompacted graph.

# Experiments

Benchmarks:

- cdBG construction
- cdBG querying
- cdBG coloring

- Setup: 16-core Intel Xeon E5-2650, 256G of RAM

# Experiments

**Table 1** Time and memory comparison of Bifrost and BCALM2 for different *k*-mer sizes and number of threads during graph construction

| | Tool | *k*-mer size | Number of threads | | | |
|---|---|---|---|---|---|---|
| | | | 1 | 4 | 8 | 16 |
| Time (h) | Bifrost | 31 | **20.81** | **8.53** | **6.10** | **5.55** |
| | | 63 | **14.38** | **4.20** | **2.40** | **2.00** |
| | | 95 | **12.51** | **3.88** | **2.25** | **1.58** |
| | | 127 | **9.56** | **2.96** | **1.81** | **1.41** |
| | BCALM2 | 31 | 44.25 | 14.11 | 8.48 | 6.33 |
| | | 63 | N/A | 25.6 | 13.96 | 8.71 |
| | | 95 | N/A | 39.91 | 21.45 | 12.56 |
| | | 127 | N/A | N/A | 27.73 | 16.15 |
| Memory (GB) | Bifrost | 31 | 39.59 | 39.58 | 39.59 | 39.60 |
| | | 63 | 37.77 | 37.77 | 37.77 | 37.78 |
| | | 95 | 44.33 | 44.30 | 44.30 | 44.32 |
| | | 127 | 55.88 | 55.86 | 55.86 | 55.86 |
| | BCALM2 | 31 | **36.00** | **35.66** | **35.61** | **35.58** |
| | | 63 | N/A | **29.83** | **29.73** | **29.64** |
| | | 95 | N/A | **33.47** | **33.51** | **33.66** |
| | | 127 | N/A | N/A | **43.42** | **53.77** |

Best results are highlighted. N/A indicates the result is unavailable because the computation took more than 48 h

# Experiments

**Table 3** Running time and memory usage for indexing and querying a de Bruijn graph for 30 million short reads

| Tool | Process | Time (m) | Memory (GB) |
|---|---|---|---|
| Bifrost | Build | **333** | 39.6 |
| | Index | **11.1** | 26.8 |
| | Query | **4.7** | 26.8 |
| | Query-total | **16.4** | 26.8 |
| BCALM2 | Build | 380 | **35.58** |
| Blight | Index | 80 | **8.3** |
| | Query | 13.6 | **8.3** |
| | Query-total | 93.6 | **8.3** |
| Squeakr | Build | 1147 | 80 |
| Mantis | Index | 54 | 17 |
| | Query | 38.8 | 168 |
| | Query-total | 96.9 | 168 |

The total time of Bifrost and Blight is split into index and query as reported by the software, whereas query-total is the wall time measurement. For Mantis, the index is a separate process and needs only to be run once

# Experiments

**Table 5** Running time, memory usage, and external disk usage for constructing the colored de Bruijn graphs of an increasing number of *Salmonella* strains

| Number of strains | Tool | Time (h) | Memory (GB) | Disk (GB) |
|---|---|---|---|---|
| 100 | Bifrost | **0.016** | **0.16** | **0** |
| | VARI-merge + KMC2 | 0.33 | 5.1 | 17 |
| 400 | Bifrost | **0.05** | **0.29** | **0** |
| | VARI-merge + KMC2 | 1.016 | 15.4 | 51 |
| 1600 | Bifrost | **0.38** | **2.4** | **0** |
| | VARI-merge + KMC2 | 4.86 | 56.9 | 228 |
| 4000 | Bifrost | **1.66** | **3.7** | **0** |
| | VARI-merge + KMC2 | 12.35 | 138 | 449 |
| 117,913 | Bifrost | **93.35** | **102.74** | **0** |
| | VARI-merge + KMC2 | N/A | N/A | N/A |

N/A indicates the result is unavailable

# Results

- +Open source library and reusability
- +Speedup in graph indexing.
- +Significant speedup and memory decrease in colored de brujin generation.
- -Nothing I could find :)

# Algorithm

- Bloom filter (BF) is a space and time-efficient data structure that records the approximate membership of elements in a set.
- Blocked Bloom Filter (BBF), better data locality
- Minimizers, ascending minima approach to share minimizers and neighbor hashing trick to make it stronger.
- Same minimizer within a BBF block. Great data locality.

# Algorithm

- Parallelize each BBF
- Utilize AVX instructions within a block
- Later combine BBF results, chains and nodes.

# Algorithm

- Ghost k-mer, put it into a hash table and don’t block.
- Reduces fragmentation, improves memory usage and running time

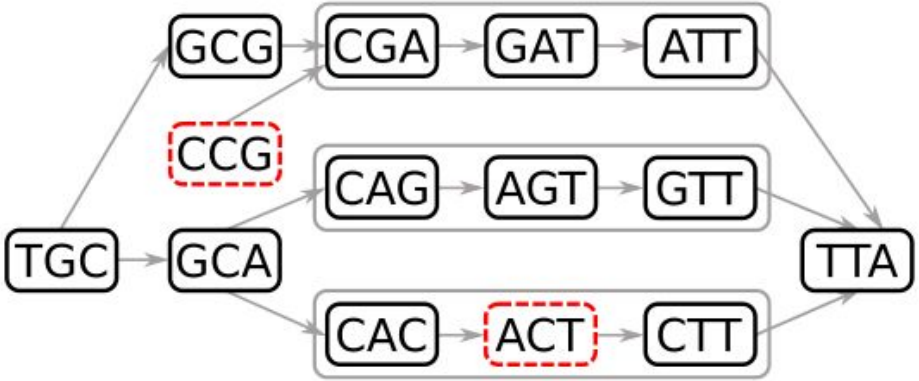


**Fig. 3** A compacted de Bruijn graph containing false positive 3-mers. Errors are represented in red dashed line vertices: *K*-mer “CCG” creates a false branching and “ACT” creates a false connection. *K*-mers that are compacted in a unitig are grouped in a gray line box

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

Guillaume Holley* and Páll Melsted

Holley and Melsted Genome Biology 2020
https://doi.org/10.186/s13059-020-02135-8

https://github.com/pmelsted/bifrost


Ömer Faruk Özdemir
linktr.ee/FarukO