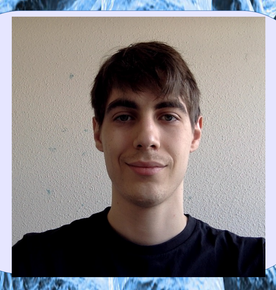
BGREAT: A De Bruijn graph read mapping tool

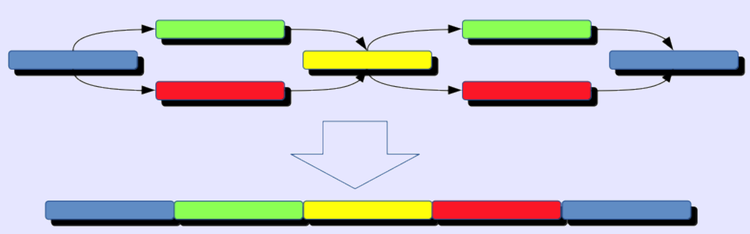
Antoine Limasset and Pierre Peterlongo
INRIA/IRISA/GenScale, Campus de Beaulieu, 35042 Rennes Cedex, France
antoine.limasset@inria.fr, pierre.peterlongo@irisa.fr



Mapping reads on references is a central task in numerous genomic studies. Since references are mainly extracted from assembly graphs, it is of high interest to map efficiently on such structures. The problem of mapping sequences on a De Bruijn graph has been shown NP-complete[1] and no scalable generic tool exists yet. We motivate here the problem of mapping reads on a de Bruijn graph and we present a practical solution and its implementation called **BGREAT**. **BGREAT** handles real world instances of billions reads with moderate resources.

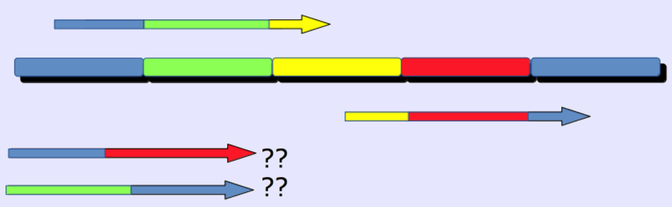
Motivation

Is a reference sequence a good representation of a genome ?



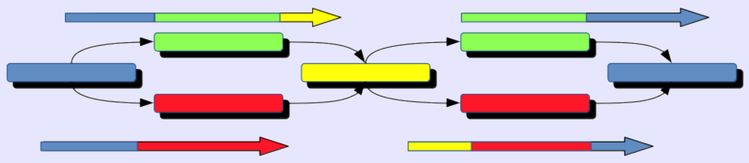
De Bruijn Graph and its assembled "reference"

Map on reference



Since some parts has been discarded, numerous reads can not be mapped

Map on graph

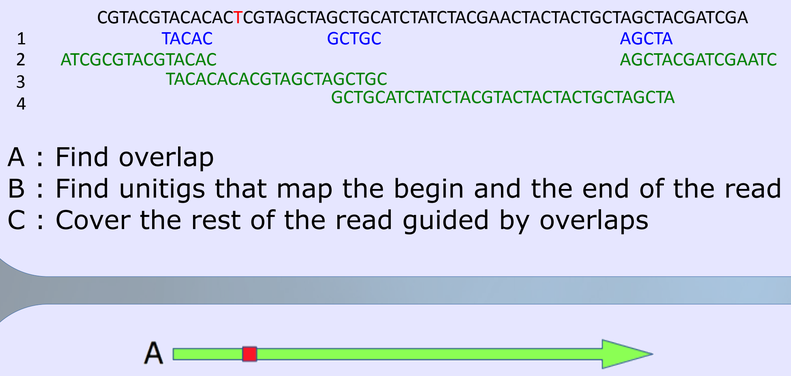


Read can be mapped even on path not chosen by the assembly. This solution enable to map on:
-The different haplotypes
-Repeated sequences
-Complex regions

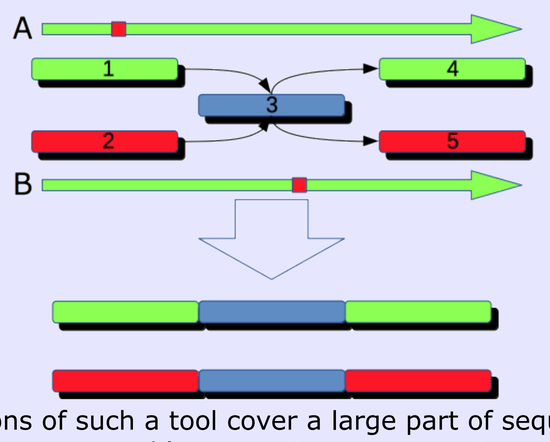
Algorithms

Our new tool, **BGREAT** maps reads on a de Bruijn graph.
Several heuristics are used to ensure an almost linear mapping process:

-Only overlaps between unitigs are indexed
-Only first overlaps seen are possible anchors
-Greedy extension between anchors



A : Find overlap
B : Find unitigs that map the begin and the end of the read
C : Cover the rest of the read guided by overlaps

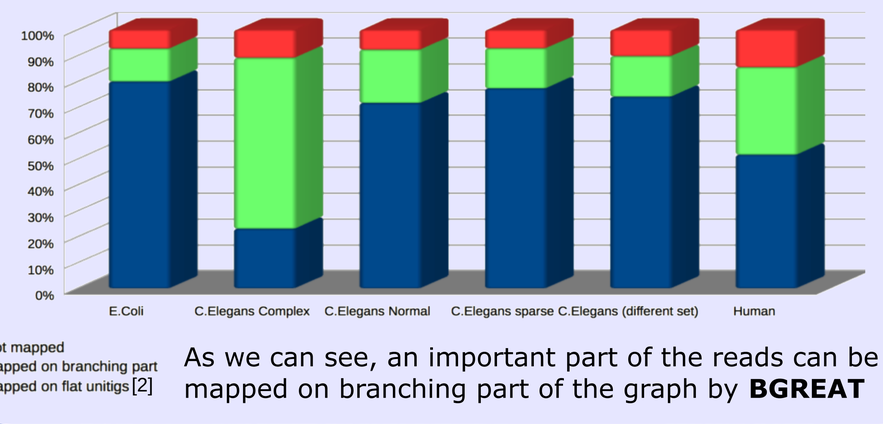


Applications of such a tool cover a large part of sequencing data treatments as assembly, correction, compression, quantification...

Results

Graph	Mapped set (nb reads)	MS	*BGREAT* results: Nb mapped on branching parts of DBG	time (RPS)	memory	Nb reads fully mapped on unitigs	Overall nb mapped reads
Coli	SRR959239 (5,372,832)	G	687,997 (12.81%)	2m2 (38,562)	15 MB	4,295,627 (79.95%)	4,983,624 (92.76%)
〃	〃	C	688,933 (12.82%)	1h24 (1,014)	〃	〃	4,984,560 (92.77%)
El_cpx	SRR065390 (67,617,092)	G	44,686,355 (66.09%)	3h08 (5,965)	1.16GB	15,592,918 (23.06%)	60,279,273 (89.15%)
El_norm	〃	〃	13,994,715 (20.70%)	1h55 (9,438)	380MB	48,442,146 (71.64%)	62,436,861 (92.34%)
El_sparse	〃	〃	10,467,181 (15.48%)	1h15 (13,093)	210MB	52,288,269 (77.33%)	62,755,450 (92.81%)
El_norm	SRR1522085 (22,509,110)	〃	3,523,416 (15.65%)	12min25s (30,213)	380MB	16,682,194 (74.11%)	20,205,610 (89.77)%
Human	SRR345593 and SRR345594 (2,967,536,821)	G	1,004,182,363 (33.84%)	11h48 (70,526)	18 GB	1,533,456,046 (51.67%)	2,537,638,409 (85.51%)

Results show that **BGREAT**:
Uses heuristics that allow a dramatic improvment in speed at at very low cost of recall
Is able to map efficiently a set of read that was not used to build the graph
Scales to large dataset



As we can see, an important part of the reads can be mapped on branching part of the graph by **BGREAT**

Mapping on de Bruijn graph enable to keep whole genomic information and get rid off possible assembly mistakes. However the problem is theoretically hard to handle on real-world dataset. Using a set of heuristics, our proposed tool is able to map million read by CPU hours even on complex human genomes.
BGREAT is available at
github.com/Malfoy/BGREAT

[1]Limasset, A., & Peterlongo, P. (2015). Read Mapping on de Bruijn graph. arXiv preprint arXiv:1505.04911.
[2]Langmead, Ben, et al. "Ultrafast and memory-efficient alignment of short DNA sequences to the human genome." Genome Biol 10.3 (2009): R25.