ABySS 2.0

Resource-efficient assembly of large genomes using a Bloom filter

Shaun D Jackman @sjackman

Benjamin P Vandervalk, Hamid Mohamadi, Justin Chu, Sarah Yeo, S Austin Hammond, Golnaz Jahesh, Hamza Khan, Lauren Coombe, Rene L Warren, Inanc Birol

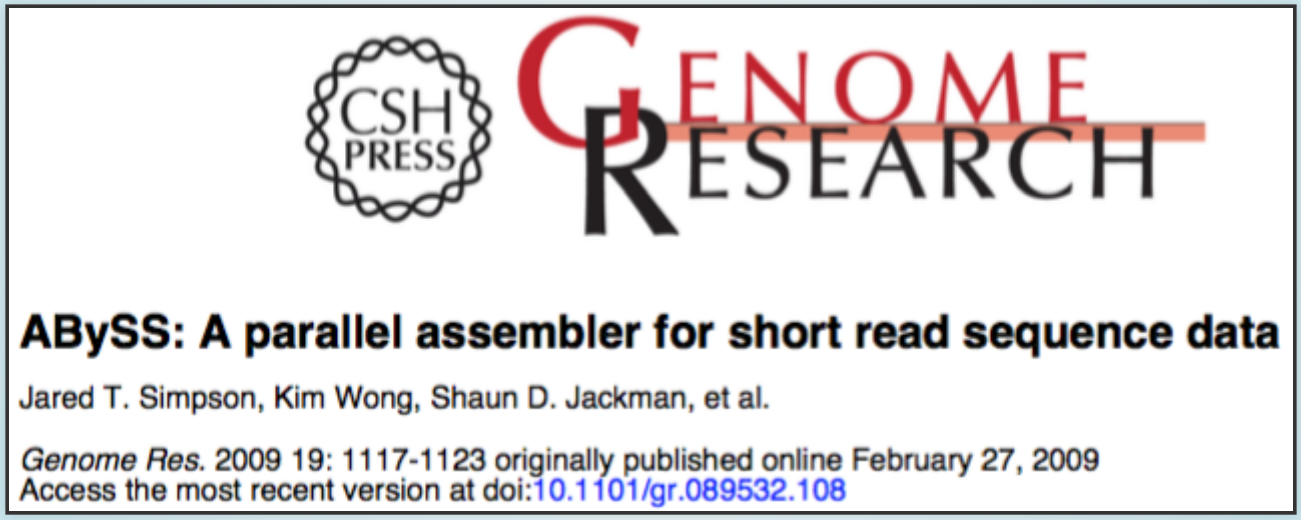
RECOMB-Seq 2018-04-19
https://sjackman.ca/abyss2-slides
Funded by Genome Canada · Genome BC · NIH · NSERC

Shaun Jackman

Birol Bioinformatics Technology Lab
BC Cancer Genome Sciences Centre · Vancouver, Canada
@sjackman · github.com/sjackman · sjackman.ca

Short Read Genome Assembly

ABySS 1.0 (2009) was the first to assemble
a human genome from short reads (42 bp!)



http://bit.ly/abyss1-paper · doi:btnjx2

ABySS 1.0

- de Bruijn graph assembler
- Stored *k*-mers in a hash table
- Distributed the hash table over many machines
- Used MPI to aggregate sufficient memory
- Assembles large genomes

ABySS 1.0

	Human	**Spruce**
Genome size	3 Gbp	20 Gbp
RAM	418 GB	4.3 TB
CPU cores	64	1,380
Wall time	14 hours	12 days
Year	2017	2013
Short DOI	doi:f9x8qp	doi:f4zzrr

Challenges

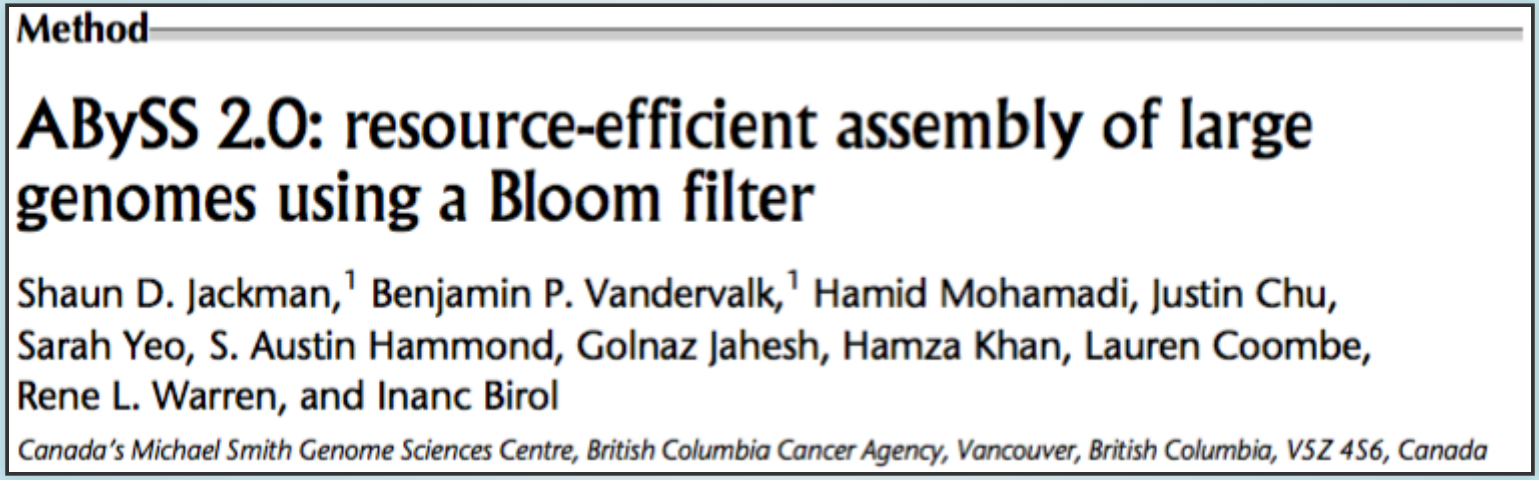
- High memory usage
- Interprocess communication is slow
- Intermachine communication is really slow

Solution

- A memory-efficient data structure reduces memory usage
- Fitting entire graph in a single machine eliminates intermachine communication
- OpenMP rather than MPI eliminates interprocess communication

ABySS 2.0

ABySS 2.0 (2017) reduces the memory usage of ABySS by ten fold.



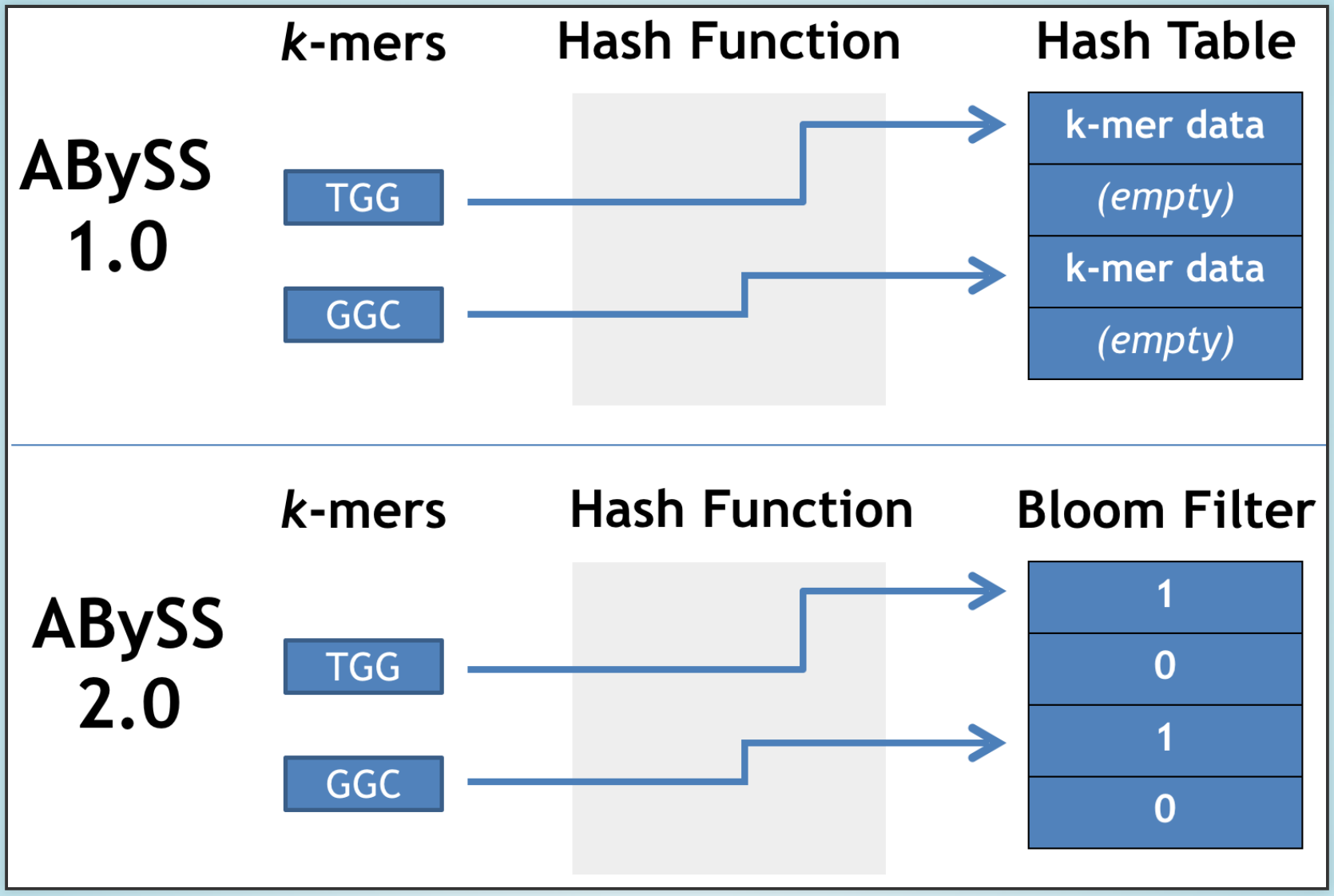
Method

ABySS 2.0: resource-efficient assembly of large genomes using a Bloom filter

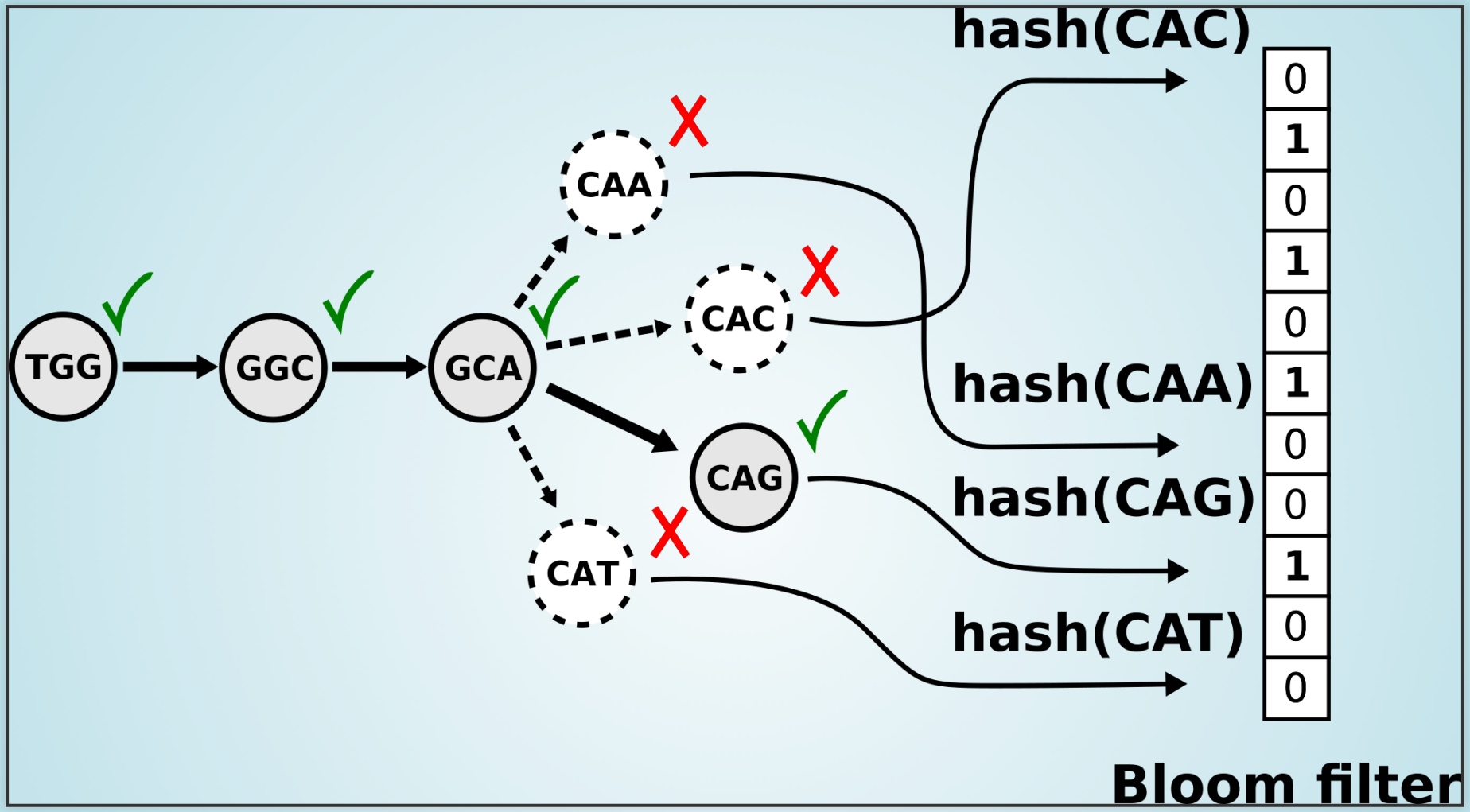
Shaun D. Jackman,[1] Benjamin P. Vandervalk,[1] Hamid Mohamadi, Justin Chu, Sarah Yeo, S. Austin Hammond, Golnaz Jahesh, Hamza Khan, Lauren Coombe, Rene L. Warren, and Inanc Birol

Canada's Michael Smith Genome Sciences Centre, British Columbia Cancer Agency, Vancouver, British Columbia, V5Z 4S6, Canada

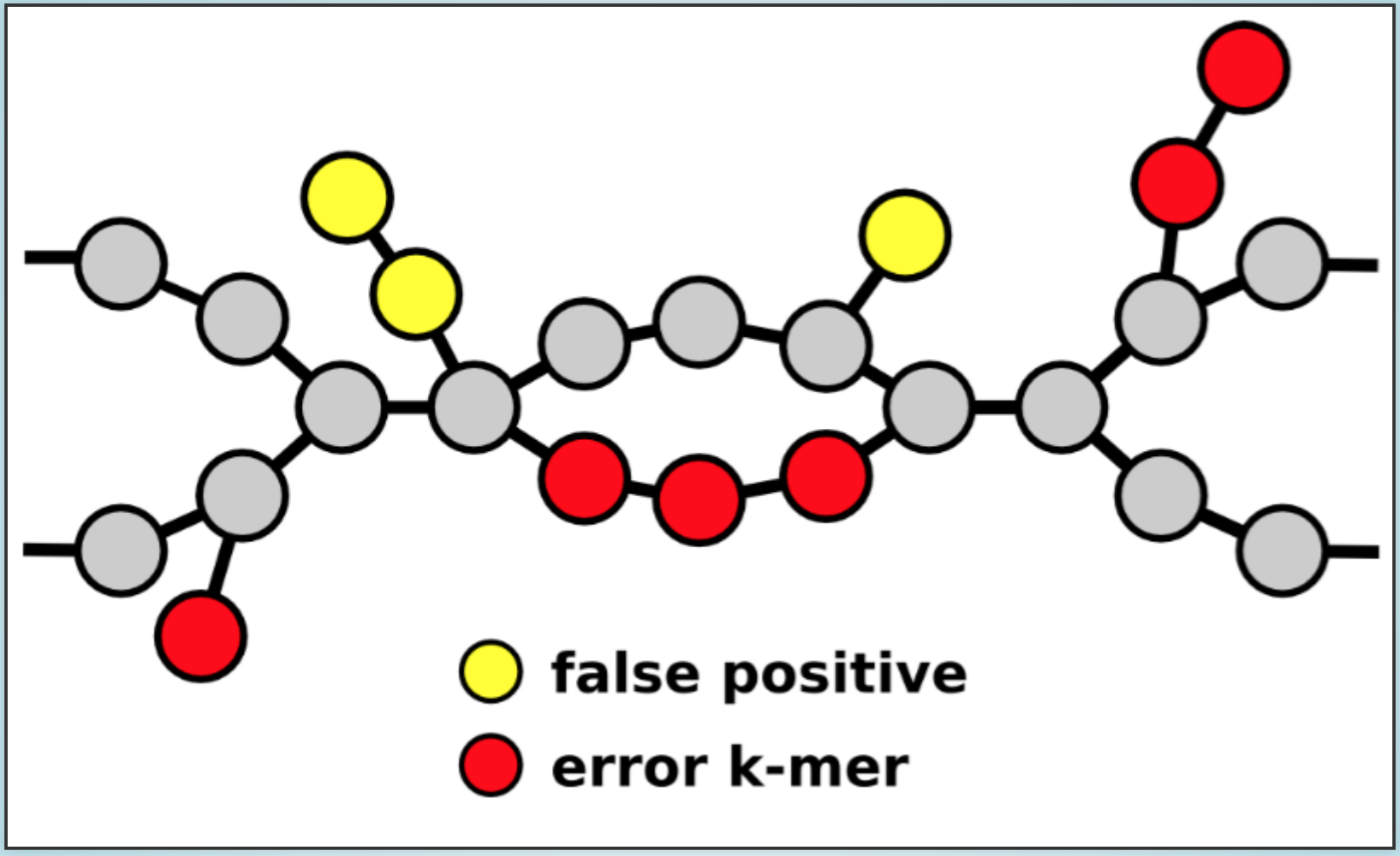
http://bit.ly/abyss2-paper · doi:f9x8qp



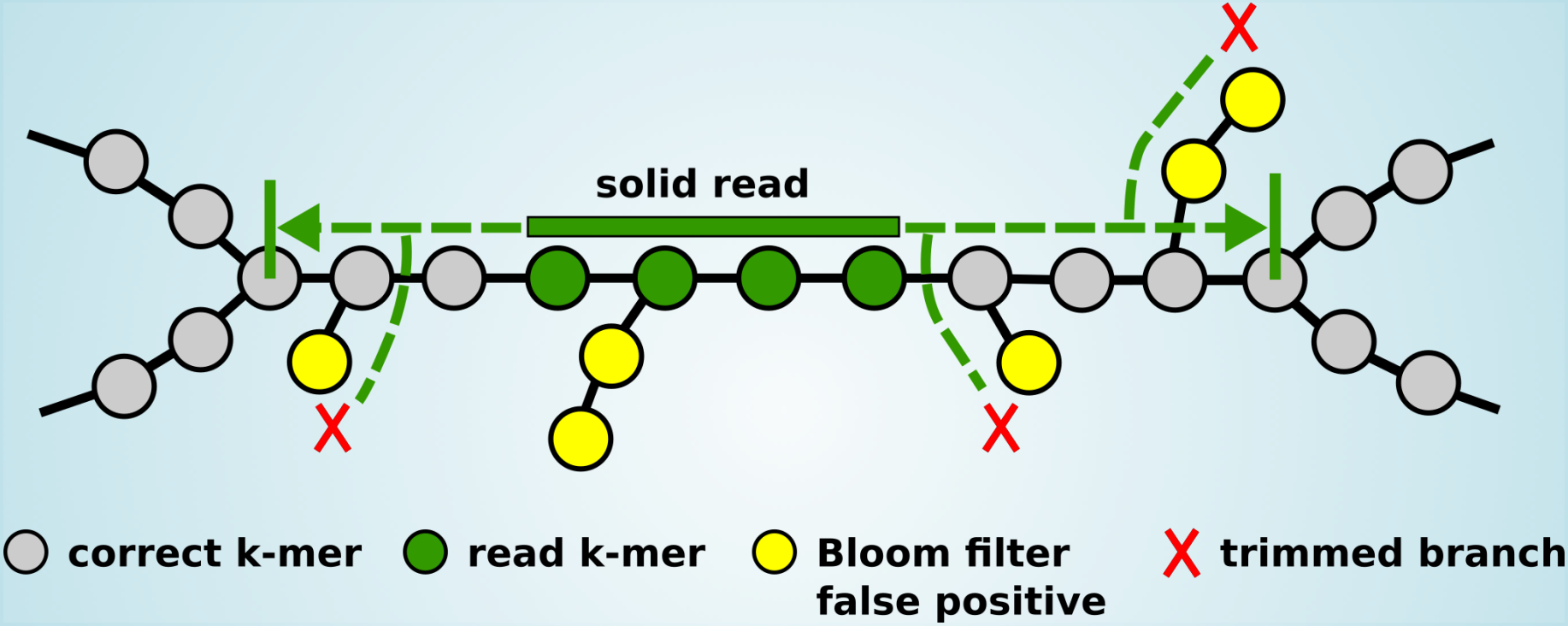
Memory efficient de Bruijn graph using a Bloom filter
Memory usage is independent of *k*



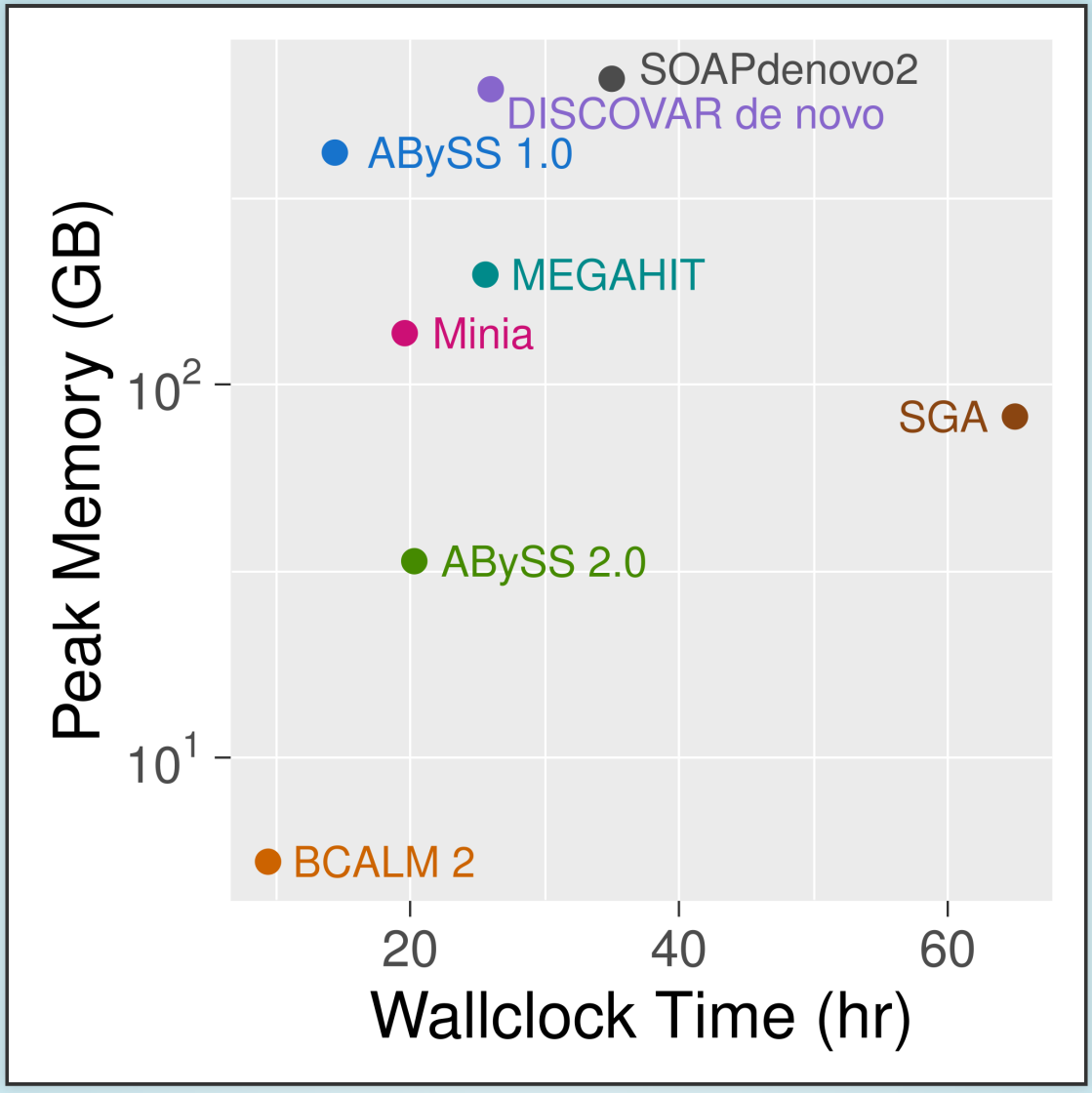
Navigating a Bloom filter de Bruijn graph
Introduced by Minia (Chikhi *et al.* 2012)



Sequencing errors and Bloom filter false positives



Solid reads are extended using the Bloom filter de Bruijn graph to assemble unitigs

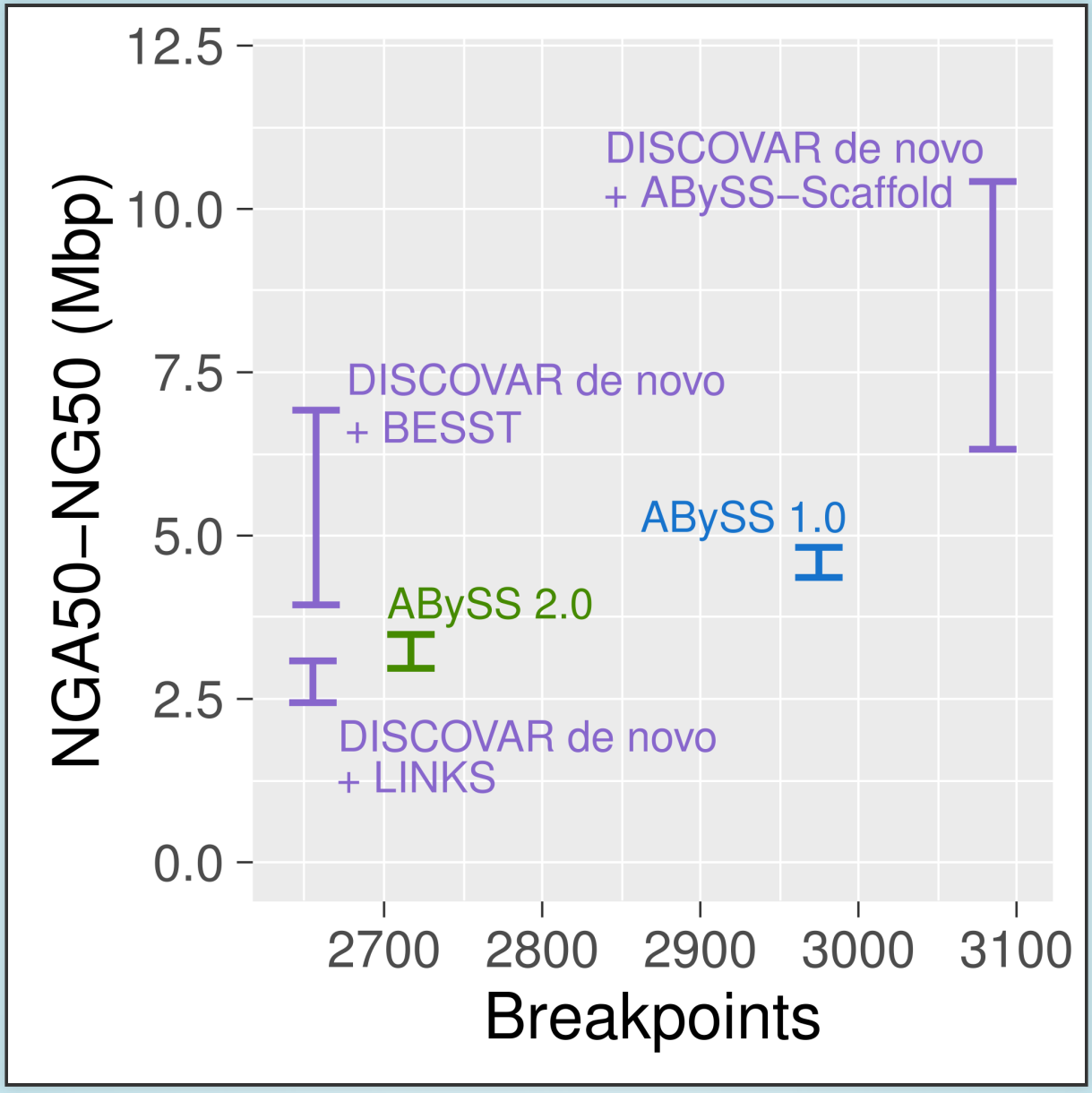


ABySS 2.0 reduces memory usage by 10 fold vs ABySS 1.0 for human genome assembly (GIAB HG004 NA24143)

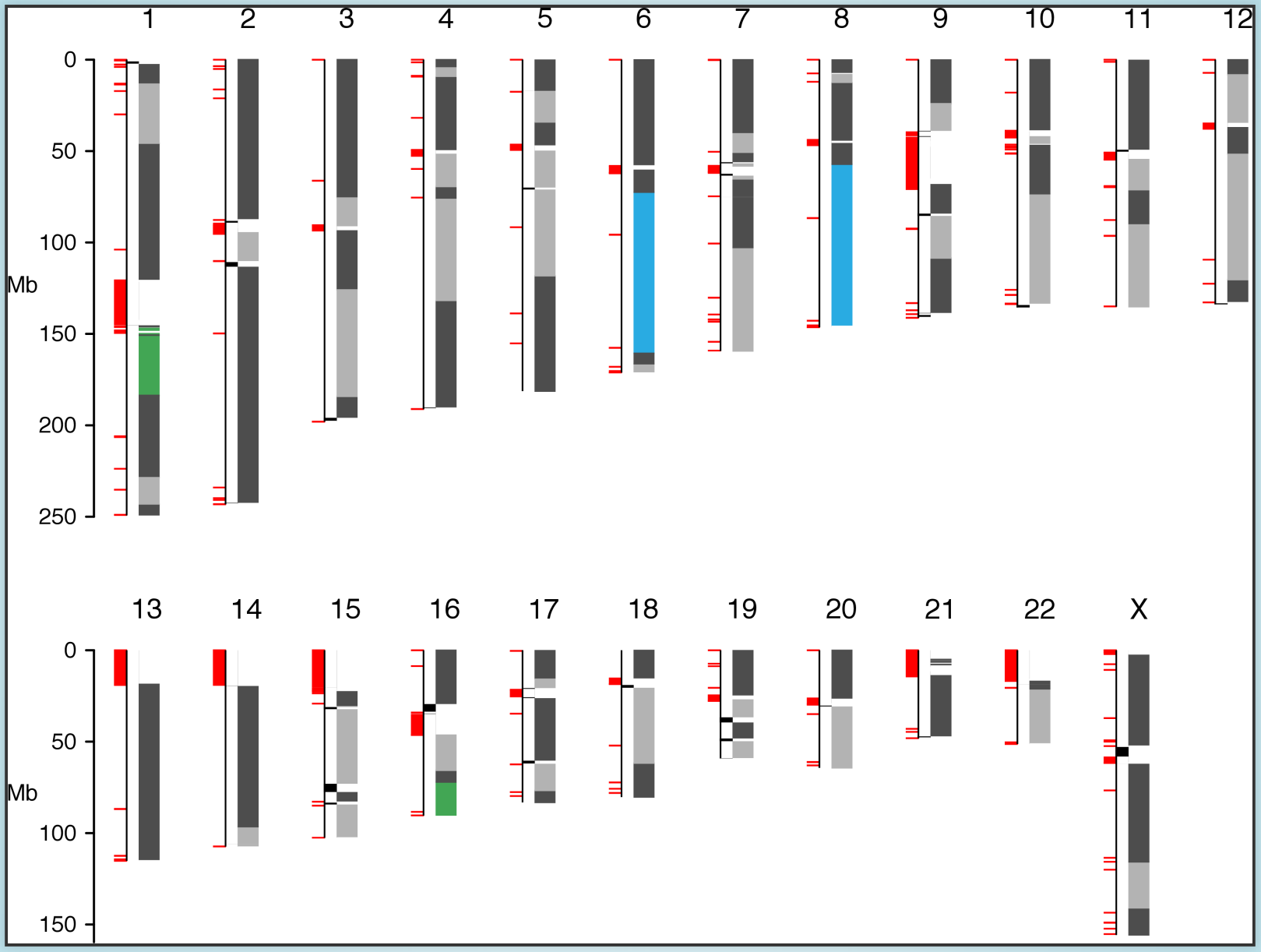
Spruce genome assemblies

ABySS	**1.3.5**	**2.0.0**
Spruce species	Interior	Sitka
Machines	115	1
RAM (GB)	4,300	500
CPU cores	1,380	64
CPU time* (years)	6.0	3.2
Wall time* (days)	1.6	18
Year	2013	2017
Short DOI	doi:f4zzrr	NA

* Time of unitig assembly without scaffolding



Contiguity and correctness are comparable



41.9 Mbp NG50 scaffolded with BioNano optical mapping

Conclusion

- ABySS 2.0 reduces memory usage by 10 fold
  from 418 GB for ABySS 1.0
  to 34 GB for ABySS 2.0
  for a human genome assembly
- High-throughput short-read sequencing
  combined with large molecule scaffolding
  such as 10X Genomics, BioNano, Hi-C
  permits cost effective assembly of large genomes

fin

Posters

- SEQ-7 **Tigmint**: Correcting Assembly Errors Using Linked Reads From Large Molecules
- SEQ-6 **ARKS**: chromosome-scale human genome scaffolding with linked read kmers
- SEQ-10 **ONTig**: Contiguating Genome Assembly using Oxford Nanopore Long Reads
- SEQ-8 **Multi-Index Bloom Filters**: A probabilistic data structure for sensitive multi-reference sequence classification with multiple spaced seeds

Shaun Jackman

@sjackman · github.com/sjackman · sjackman.ca

Benjamin P Vandervalk, Hamid Mohamadi, Justin Chu, Sarah Yeo, S Austin Hammond, Golnaz Jahesh, Hamza Khan, Lauren Coombe, Rene L Warren, Inanc Birol

ABySS 2.0

https://github.com/bcgsc/abyss

ABySS 2.0 Paper

http://bit.ly/abyss2-paper · doi:f9x8qp

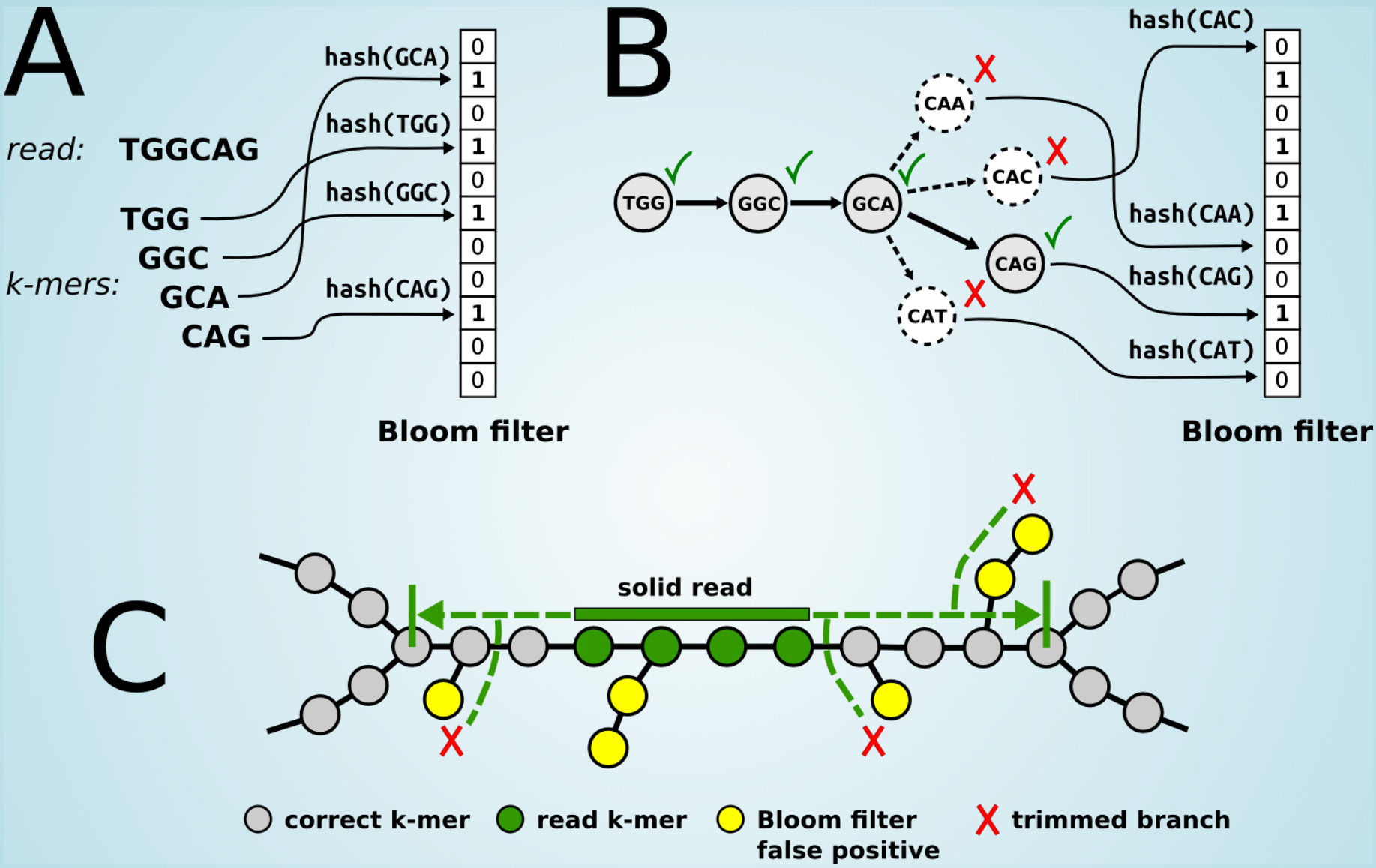
Slides

http://sjackman.ca/abyss2-slides

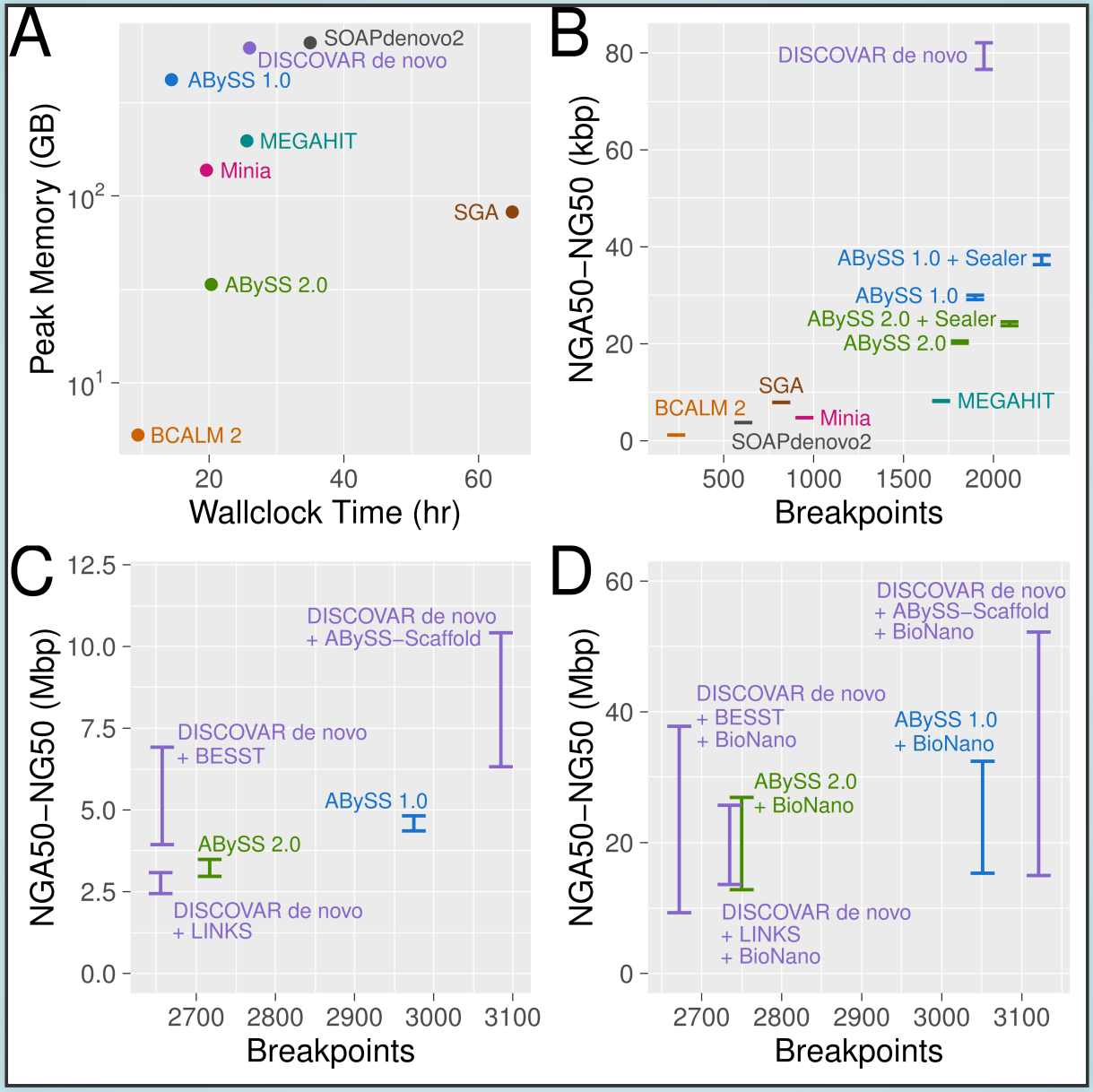
https://github.com/sjackman/abyss2-slides

Funded by Genome Canada · Genome BC · NIH · NSERC

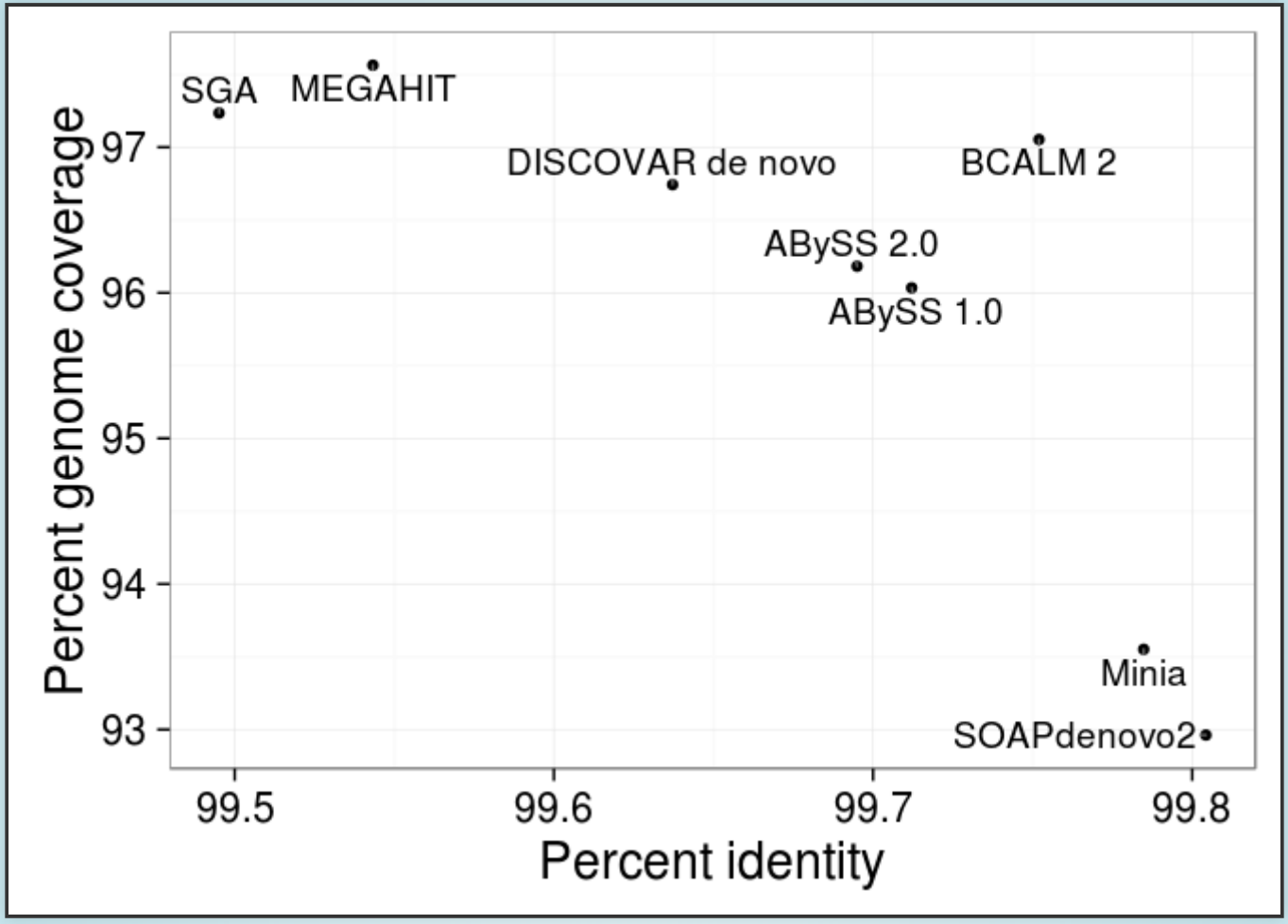
Supplemental Slides



ABySS 2.0 assembly algorithm



Assembler comparison



Genome coverage and identity

Jupiter plot of scaffolds mapped to the reference genome