Scalable Genomic Assembly through Parallel de Bruijn Graph Construction for Multiple K-mers

Kanak Mahadik, Chris Wright, Milind Kulkarni, Saurabh Bagchi, Somali Chaterji

PURDUE UNIVERSITY

Sequence assembly

- Reads : fragments of genome read by sequencing instruments
- Reconstruct the original genome from the reads
- *Approximate,* no single best assembly tool*
  - Repeating patterns in original genome
  - Uneven sampling of original genome
  - Errors in reads

* Salzberg, S. L., Phillippy, A. M., Zimin, A., Puiu, D., Magoc, T., Koren, S., & Marçais, G. (2012). GAGE: A critical evaluation of genome assemblies and assembly algorithms. *Genome research*, *22*(3), 557-567.

De bruijn Graph (DBG)

SEQUENCE:

A A T G C C

READS:

A A T

T G C

G C C

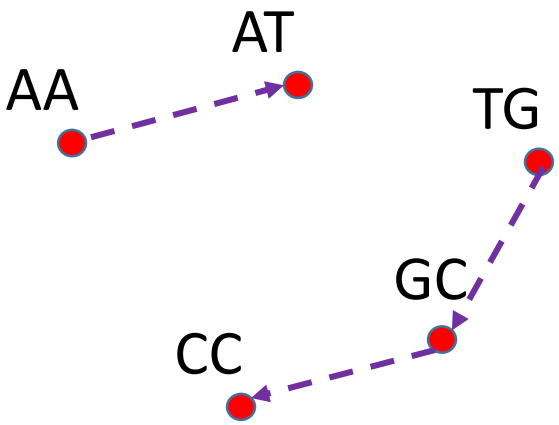
Objective: Assemble reads into longest contigs with good coverage (extent of redundancy in the final contig set)

Edge between vertices u and v in G if they are consecutive k-mers in a read & the last (k-1) nucleotides of the k-mer u are the same as the first (k-1) nucleotides of the k-mer v

Constructed by traversing the DBG to identify *maximal paths* in the graph. (All vertices have in-degree and out-degree equal to 1, except at the start and end of the path)

K-mer set for k=2

{AA, AT, TG, GC, CC}



Contig set k=2

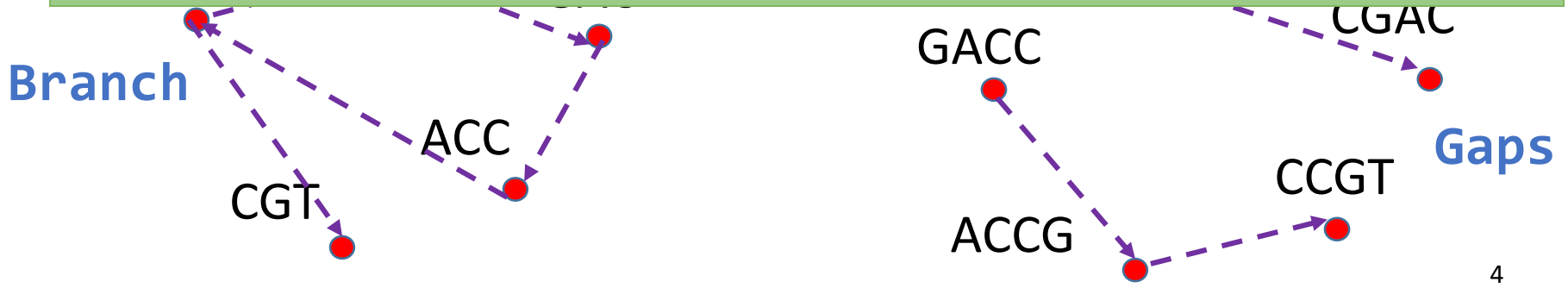
{AAT, TGCC}

Significance of K-value

SEQUENCE:

K-mer set for k=3

- Smaller k-value
  - Probability of extraction of valid k-mer from a read is higher
  - branched DBG
- Larger k-value
  - Can resolve repeats of greater length
  - fragmented DBG
- Both stop contig extension → reduced contig lengths



Iterative de bruijn graph tools

- Tools use multiple k-values sequentially : IDBA-UD, SPAdes
- Small k-value graph traversed to update larger graph
- Graph “accumulated” and “updated” at each iteration in a range
- IDBA-UD tool : medium-fast and memory efficient,

  SPAdes tool : slow and memory inefficient*

*Abbas, M. M., Malluhi, Q. M., & Balakrishnan, P. (2014). Assessment of de novo assemblers for draft genomes: a case study with fungal genomes. *BMC genomics*, *15*(9), S10.

IDBA-UD iterative process

SEQUENCE:

C C G A C C G T

READS:

C C G A C ✗

Read Set is updated to remove reads already covered in contigs

Graph $G_{k=3}$ is upgraded to $G_{k=4}$ based on information from Read $Set_{k=4}$ and Contig Set $C_{k=3}$

Contigs of $G_{k=3}$ are constructed

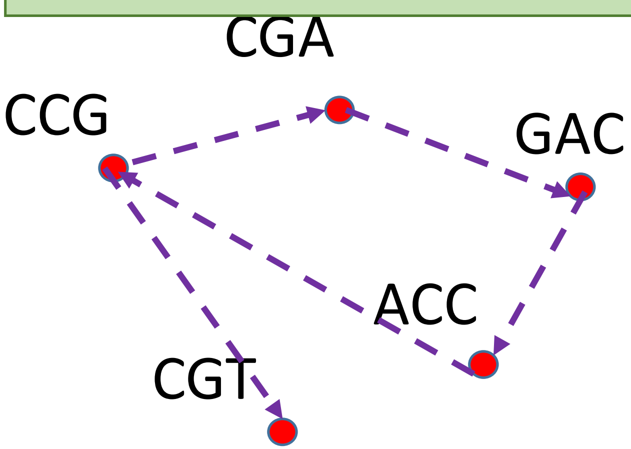
K-mer set for k=3

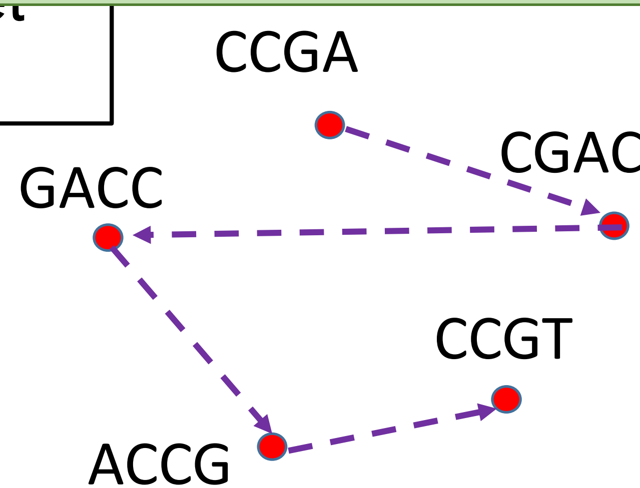
{CCG,CGA,GAC,ACC,CGT}

- Dependence of graph at iteration "i+1" on graph at iteration "i", contigs of graph at iteration "i" and read set at iteration "i"
- Forces sequential operation on a chain of k-values

New Read Set

ACCGT





Problems with IDBA-UD

- Iterative graph construction iterations dominate the overall execution time(>90%)
- Iterative de bruijn assemblers are sequential
- No parallelism for long chain of k-values
- Scalability is limited to single node
- Insufficient for large datasets

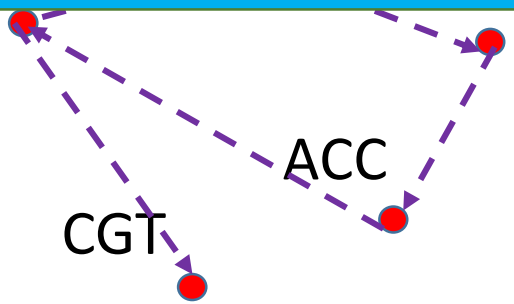
ScalaDBG Insight

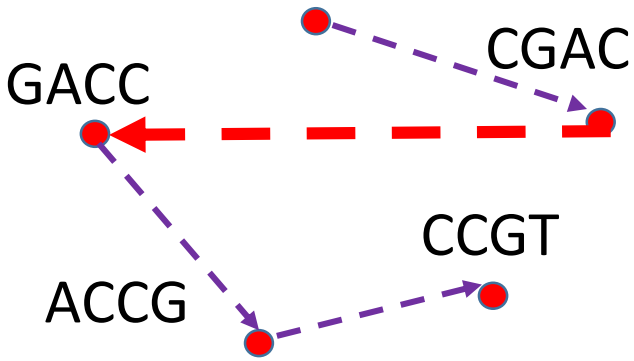
SEQUENCE:
C C G A C C G T
READS:

K-mer set for k=3
{CCG,CGA,GAC,ACC,CGT}

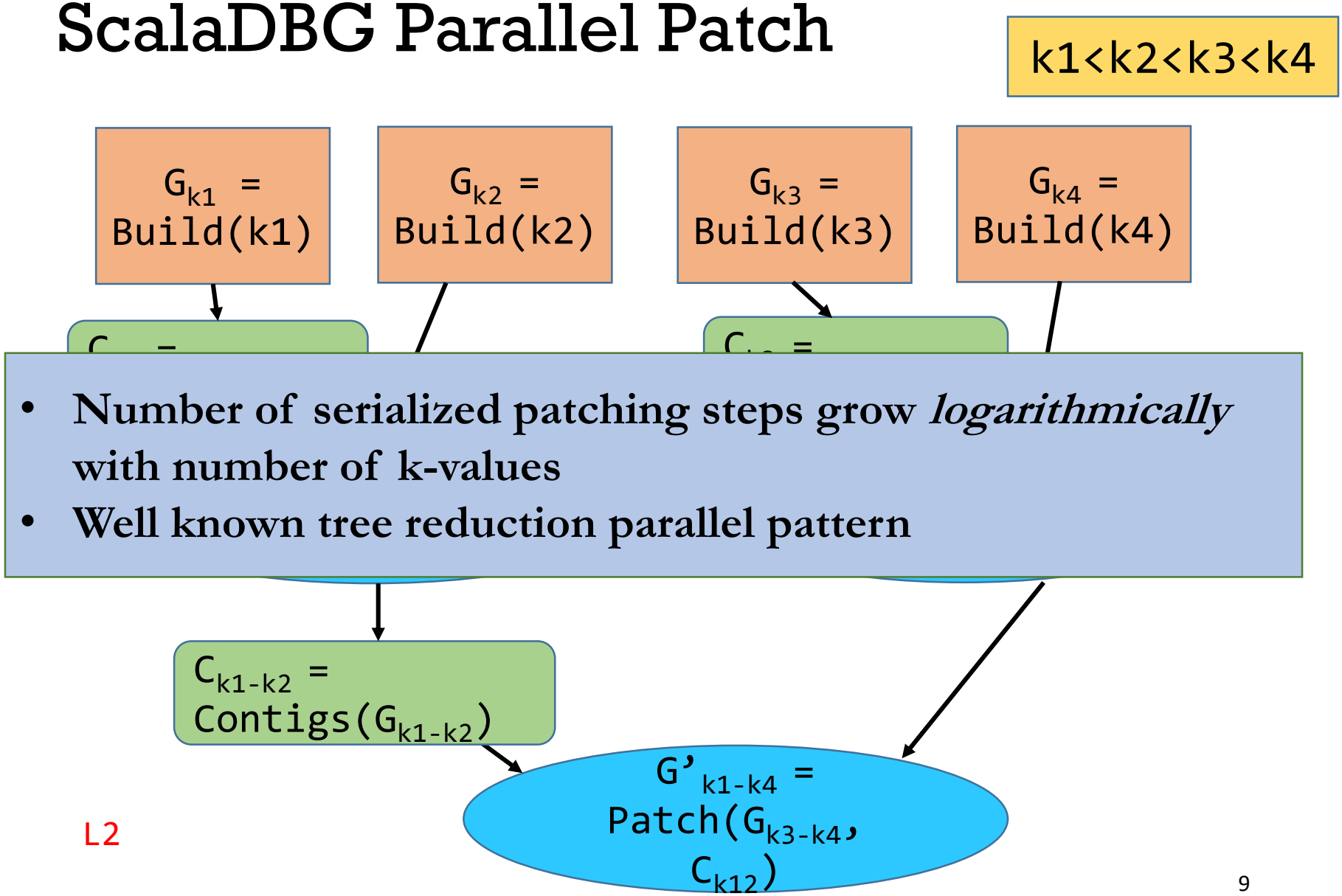
K-mer set for k=4

- Build (partial) graphs independently in parallel
- Push or "patch" information from lower k-value graph to higher k-value graph





ScalaDBG Parallel Patch

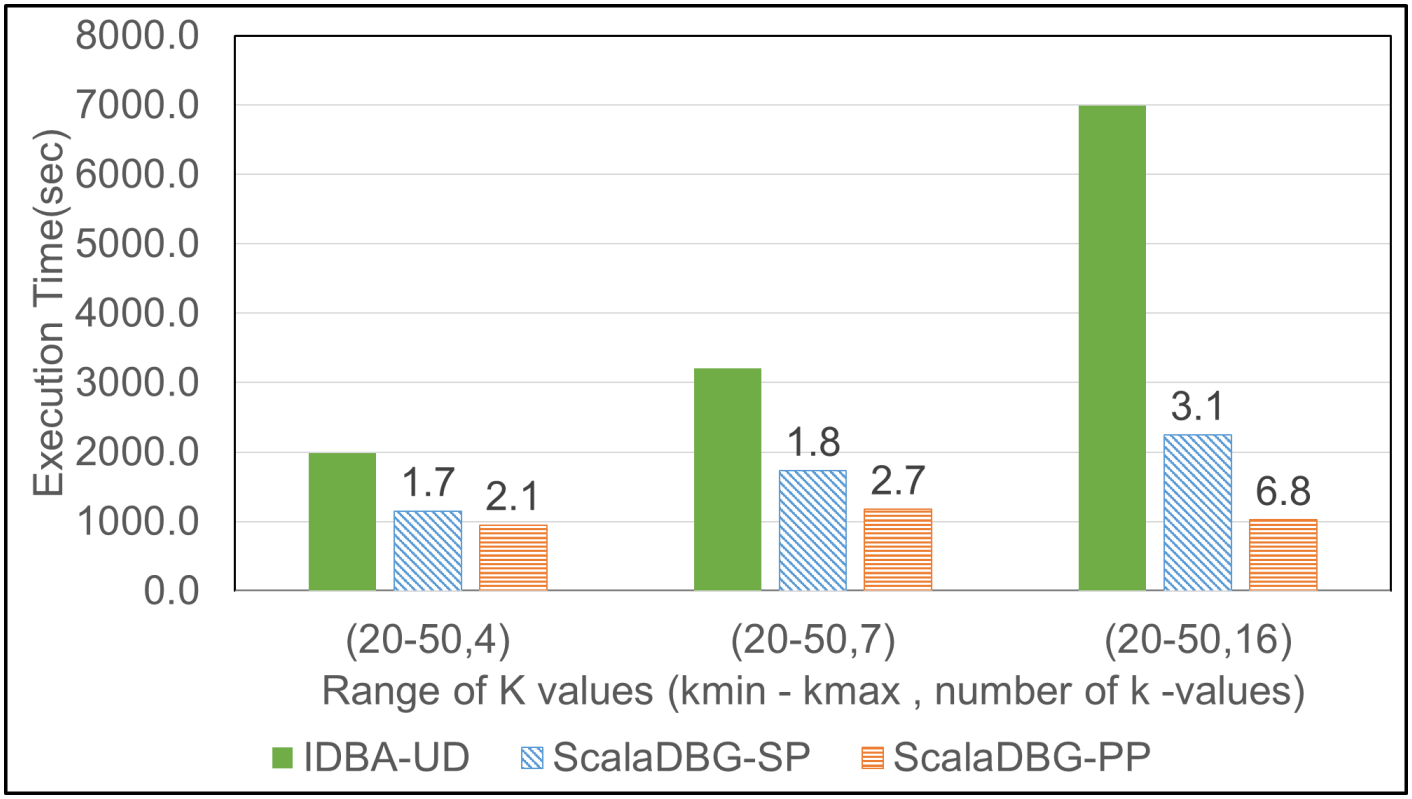


Evaluation - Setup

Name	Read Set Type	Read Length (bp)	# reads
RM1 , RM2	Real, Metagenomic	150	33,140,480
SC – E.coli	Real, Single Cell	100	23,818,596
E. coli	Simulated	75	1,882,418
SC – S.aureus	Real, Single Cell	100	66,997,488
SC – SAR324	Real, Single Cell	100	55,733,218

- Intel Xeon Infiniband cluster, Intel Xeon E5-2670, 2.6 GHz with 16 cores per node, 32 GB of memory
- IDBA-UD (1.1.1)

Performance on SC-*SAR 324* dataset



- Speedup increases with number of k-values
- 6.8X of baseline, 2.2X of serial patch for 16 k-values

Accuracy Results Overview

- Used Quast tool to analyze quality
- Difference in metric values not statistically significant
- Metrics Analyzed
  - N50 : Median value of the length of contigs
  - # Contigs : Number of assembled non-gap genomic sequences
  - Max Contig Length
  - NGA50 : Median value of the aligned contig length normalized w.r.t genome length
  - Coverage : Number of aligned bases in the contigs divided by the reference length.
  - #Missassemblies : number of positions in the contigs incorrect

Conclusion

- Technique to exploit parallelism in multi k-value DBG assembly
- Distribute on a cluster of nodes
- Provides modular stages for construction, patching, and contig generation, applicable to other assemblers

https://github.com/purdue-dcsl/Scaladbg

Thank You!