Scalable Methods for Genome Assembly

Priyanka Ghosh, Ananth Kalyanaraman*(PhD Advisor), Sriram Krishnamoorthy+ (Mentor)
*School of EECS, Washington State University, +Pacific Northwest National Laboratory

Problem Description

- Genome assembly is a fundamental problem in the field of bioinformatics
- The goal lies in reconstructing an unknown genome from short DNA fragments obtained from it.
- With the advent of high-throughput sequencing technologies, billions of reads can be generated in a few hours leading to TB/day data accumulation.
- Research focus: Implement new scalable methods for performing extreme-scale genome assembly suited for microbial genomes and other complex eukaryotic genomes.

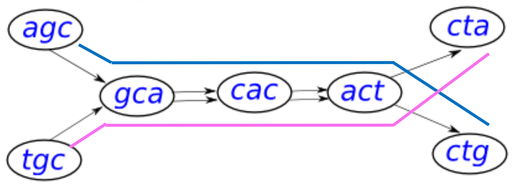
De Novo Genome Assembly

Input: Reads that may contain errors
Extract k-mers (k=3): chop reads into k-mers, process to remove errors

S_1 : AGCACTG → AGC, GCA, CAC, ACT, CTG

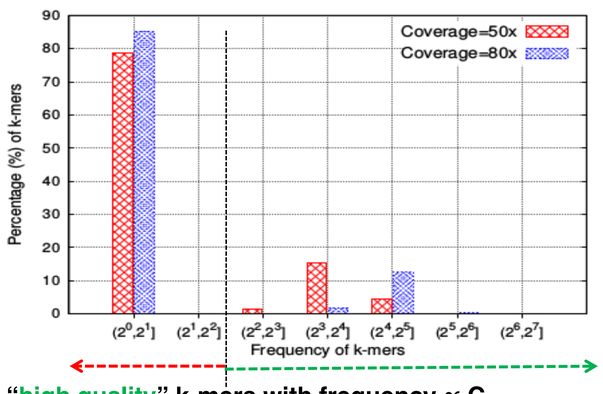
S_2 : TGCACTA → TGC, GCA, CAC, ACT, CTA

De Bruijn graph (DBG): construct graph where, nodes=k-mers, edges= (k-1)-mer overlap. **Output:** Traverse graph to enumerate long contiguous genomic regions or **contigs**



K-mer Frequency

Frequency of k-mers in read datasets depends on : a) Sequencing coverage (C) b) Error rate



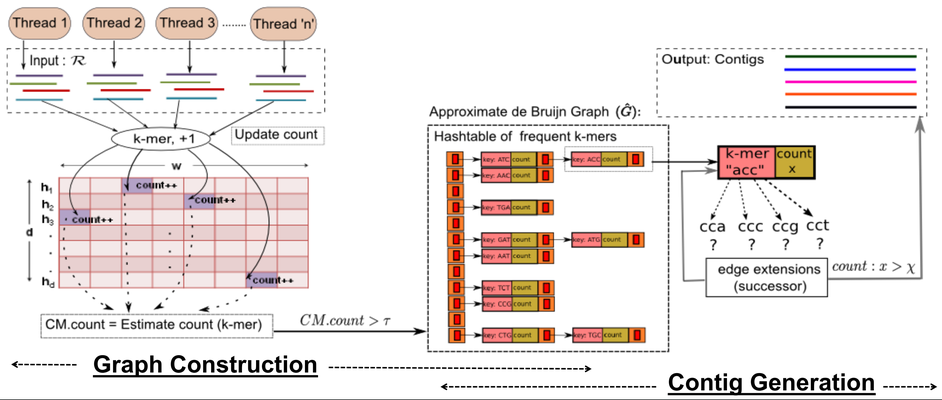
"high quality" k-mers with frequency ∝ C,
"low quality" k-mers are infrequent

Shared-memory method for assembly: *FastEtch*

Objective: To produce fast and space-efficient assemblies with shorter turnaround time.

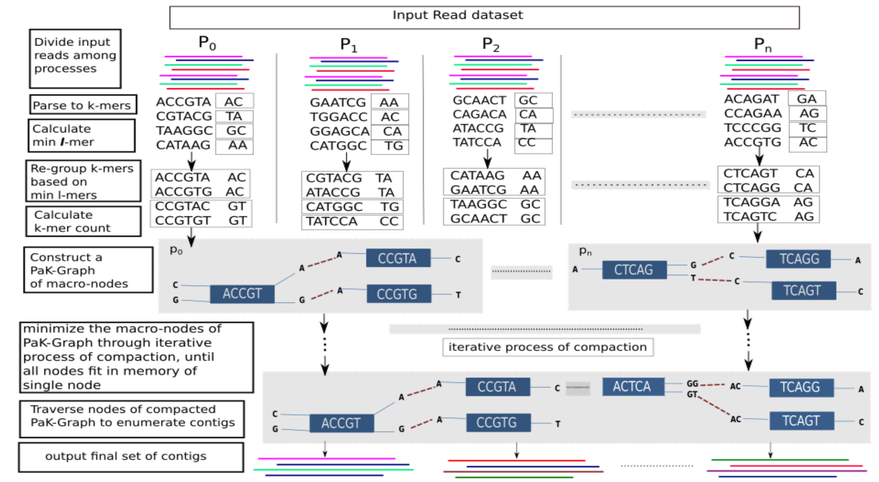
***FastEtch* Contributions:**

- A new algorithm that uses a probabilistic data structure designed for counting—viz. the Count-Min sketch [C, Muthukrishnan '04] to approximate k-mer counts
- Builds a low memory, approximate version of the de Bruijn Graph using the CM-sketch
- Stores only a subset of vertices and detects edges on-the-fly as contigs are generated
- Approximate approach aims at improving performance (both time and memory footprint)
- Two variants presented: A) FastEtch: contigs represent genomic regions from one of the DNA strands B) Bi-FastEtch: contigs switches between the two strands of the DNA



Distributed-memory method : *PaKman* (In Progress)

Objective: Implementation of a fully-distributed (MPI + OpenMP) extreme-scale genome assembler aiming to tackle the assembly of large genomes
Essential Contributions: 1) Handle I/O overheads at the time of reading and distributing read dataset. 2) Load balance the distribution of k-mers. 3) Novel data-structure to minimize inter-process communication at the time of contig generation. 4) Demonstration on shared and distributed memory systems



Future Work: 1) Incorporate support for paired-end reads (for FastEtch and PaKman) 2) Complete the design and implementation of the assembly pipeline for PaKman (include support of Scaffolding phase) 3) Extend algorithm to support other assembly frameworks (transcriptome assembly)

FastEtch: Experiments and Results

- Experiments conducted on a single 128GB DDR4 memory node of NERSC Cori

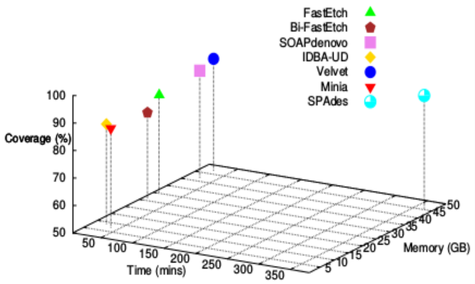
Input datasets used in our experiments

Organism	Genome Size (bp)	Coverage	# of Reads	Avg length	Data size (GB)
E.coli	4,703,541	80	3,713,280	100	0.4
Yeast	12,157,105	100	12,156,200	100	1.4
C.elegans	100,286,401	50	50,143,050	100	5.3

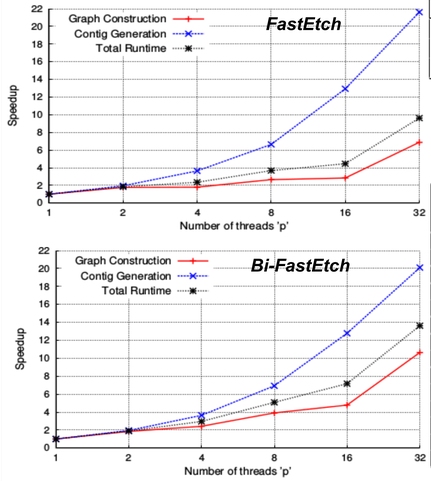
Breakdown of execution time (in secs) for different stages of the assembly conducted on 32 threads

	Input	Reading Input	Graph Construction	Contig Generation	Total Time
FastEtch	E. coli	4.38	33.48	10.71	48.57
	Yeast	12.14	113.8	46.05	171.99
	C.elegans	42.29	428.93	129.06	600.28
Bi-FastEtch	E. coli	4.74	26.92	13.91	45.57
	Yeast	18.23	87.96	49.44	155.63
	C.elegans	40.46	379.25	257.67	677.38

3-D plot depicting the performance (time, memory, quality) of all assemblers tested on the *C. elegans* (cov=50x), *k*=32

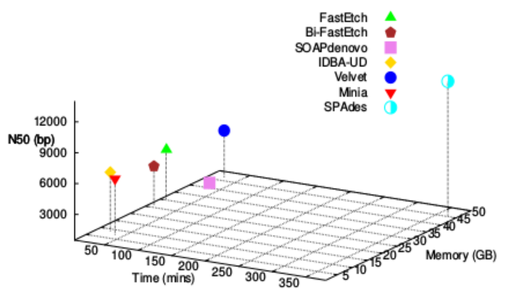


Speedup of FastEtch and Bi-FastEtch on C. elegans (50x, k=32)



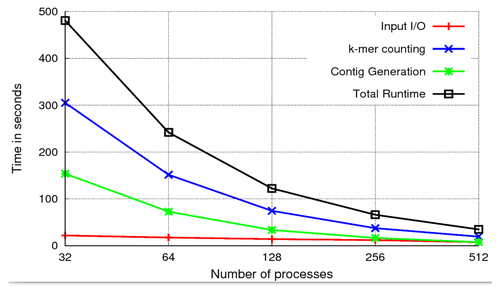
Comparing quality statistics of FastEtch vs. Bi-FastEtch

		Number of contigs	N50	Duplication factor	Number of k-mers in $\hat{G}$
E.coli	FE (τ=400)	1,473	14,122	1.94	56,029,869
	Bi-FE (τ=400)	934	12,337	1	50,208,772
Yeast	FE (τ=800)	3,565	17,805	1.92	182,127,245
	Bi-FE (τ=1600)	2,133	17,536	1	154,341,561
C.elegans	FE (τ=1k)	58,304	5,221	1.69	886,744,228
	Bi-FE (τ=6k)	38,493	4,280	1.01	712,570,949
Chr2	FE (τ=6k)	142,994	2,678	1.48	1,086,339,454
	Bi-FE (τ=24k)	103,457	2,210	1.01	810,708,459
Chr2+3	FE (τ=10k)	258,676	2,620	1.48	1,965,038,032
	Bi-FE (τ=20k)	190,154	2,146	1.02	1,527,148,071



PaKman preliminary results

Strong scaling results on NERSC Cori for dataset: C.elegans (100x) , size=11GB, with # of distinct k-mers : 1,585,416,564



References

1) Ghosh, Priyanka, and Ananth Kalyanaraman. "A Fast Sketch-based Assembler for Genomes." In Proceedings of the 7th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics, pp. 241-250. ACM, 2016.
2) Ghosh, Priyanka, and Ananth Kalyanaraman. "FastEtch: A Fast Sketch-based Assembler for Genomes." IEEE/ACM transactions on computational biology and bioinformatics (2017).

Acknowledgements

This research was supported in part by U.S. DOE grant DE-SC-0006516 and U.S. DOE under Contract No. DE-AC02-05CH11231.