KSSD-Arrayhash: Segmented Random-Array Mapping for Efficient k-mer Minimizer Construction

Xiaoxin Lu, Huiguang Yi
1. Agricultural Genomics Institute at Shenzhen, Chinese Academy of Agricultural Sciences（Shenzhen Branch, Guangdong Laboratory of Lingnan Modern Agriculture）, Shenzhen , 518000, China;
Contact information: luxiaoxin@caas.cn

Background

- Minimizers are representative k-mers selected from sliding windows.
- They reduce the number of stored seeds and accelerate candidate region detection.
- Minimizer construction requires mapping many overlapping k-mers to comparable integer values.

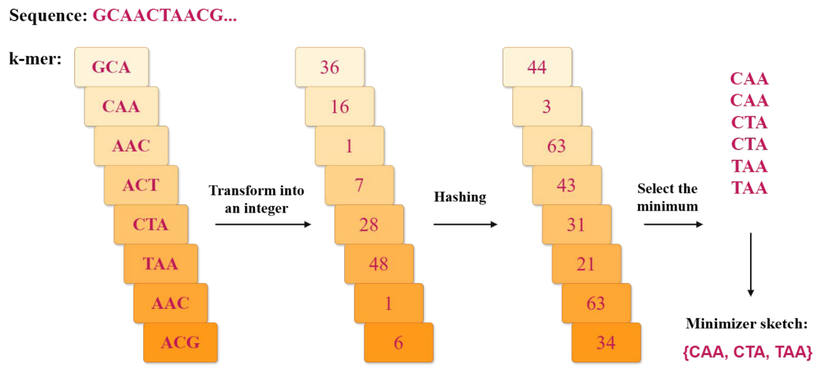

Fig.1 Minimizers selection process

Method Overview: KSSD-Arrayhash

- KSSD-Arrayhash replaces repeated hash computation with segmented random-array lookup and bitwise recombination.

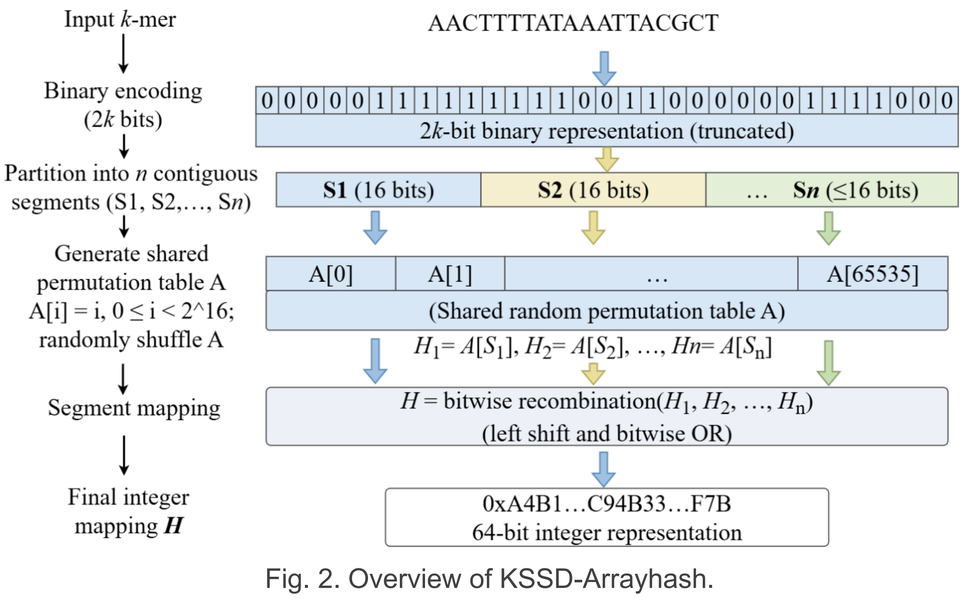

Fig. 2. Overview of KSSD-Arrayhash.

k-mer length	Number of segments
k ≤ 8	1
8 < k ≤ 16	2
16 < k ≤ 24	3
24 < k ≤ 32	4

Table 1. Adaptive segmentation rules used in KSSD-Arrayhash.

✓Each segment is limited to at most 16 bits, keeping the lookup domain within 2^{16} entries and improving cache-friendly access.

Experimental Setup

Datasets: Synthetic DNA sequence (300M) and real genomic references, including Arabidopsis, Human GRCh38, and Zea mays.

Baselines: XXH3, XXH64, MurmurHash3, Wyhash, ntHash

Environment: Single-threaded experiments

Integration: KSSD-Arrayhash was integrated into Minimap2 v2.30 by replacing only the k-mer integer mapping step. Other indexing and alignment procedures were kept unchanged

Results

◆Result 1: Faster Minimizer Construction

◆Under k = w = 8, KSSD-Arrayhash outperformed XXH3, XXH64, MurmurHash3, and Wyhash in minimizer construction throughput.

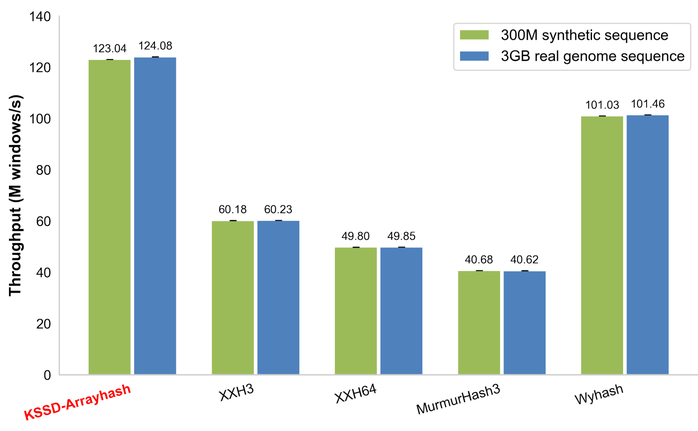

Fig. 3. Throughput comparison under k = w = 8.

Synthetic 300M
123.04 M windows/s

Real genomic sequence
124.08 M windows/s

✓Takeaway: highest throughput on both synthetic and real genomic datasets.

◆Result 2: Comparison with ntHash (k = w = 4–32)

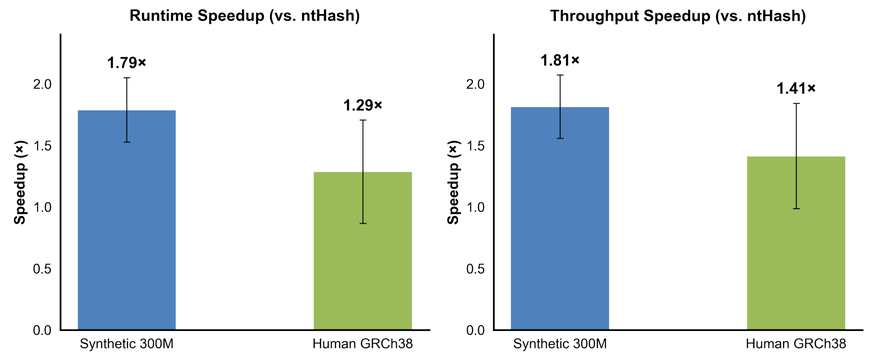

Fig. 4. Average speedup over ntHash.

✓Across k = w = 4–32, KSSD-Arrayhash was faster than ntHash on both synthetic and real genomic datasets, achieving up to 1.79× runtime speedup and 1.81× throughput speedup.

◆Result 3: Statistical Uniformity

◆Chi-square test (higher pass rate is better)

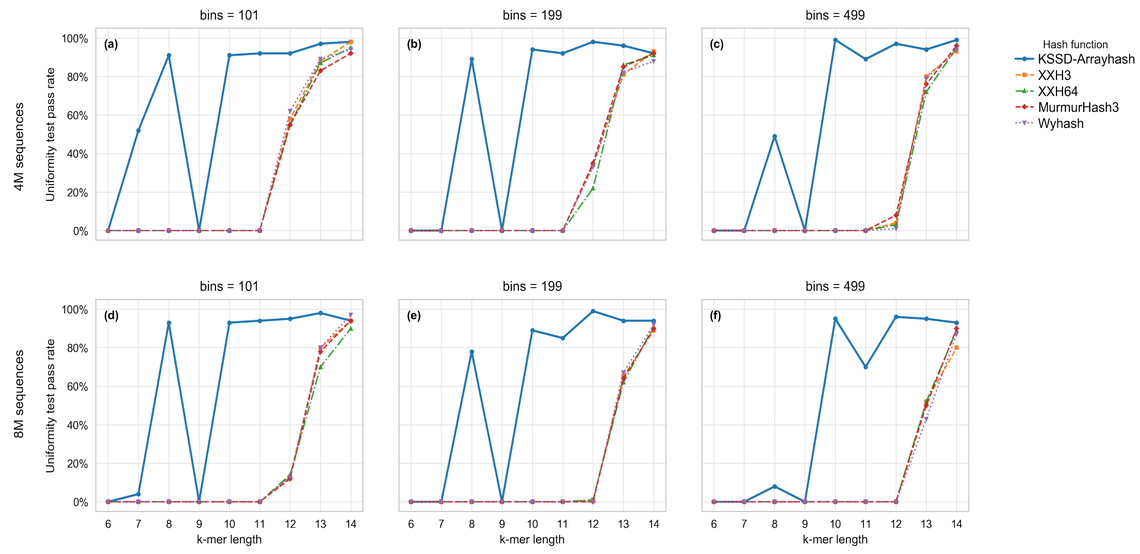

Fig. 5. Statistical uniformity evaluation.

✓KSSD-Arrayhash achieved high pass rates when k ≥ 10 under most test settings, showing comparable statistical uniformity to mainstream hash functions.

◆Result 4: Minimap2 Integration and Alignment Consistency

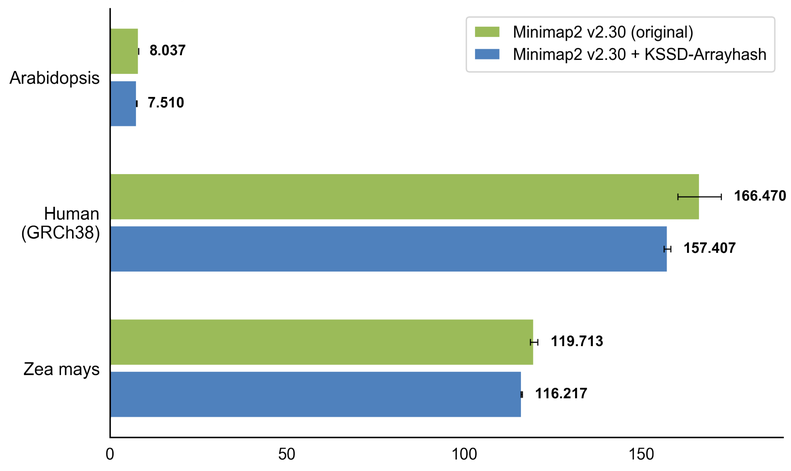

Fig. 6. Minimap2 index construction time.

✓KSSD-Arrayhash reduced index construction time across tested reference genomes.

Reference	Read length	Global acc. Δ	MAPQ=60 Δ
Human	100 bp	+0.03%	−0.037%
	150 bp	+0.02%	−0.083%
Zea mays	100 bp	−0.01%	−0.242%
	150 bp	−0.05%	−0.075%

Table 2. Alignment consistency after Minimap2 integration

✓Downstream validation showed only minor changes after KSSD-Arrayhash integration. Global accuracy differences remained within 0.05 percentage points, and MAPQ = 60 changes were below 0.25% in all tested cases.

✓Takeaway: KSSD-Arrayhash reduced Minimap2 index construction time while preserving downstream alignment consistency.

Conclusion

- KSSD-Arrayhash provides a segmented random-array mapping strategy for efficient k-mer minimizer construction.
- It improves minimizer construction throughput and remains competitive against the DNA-specific rolling hash method ntHash.
- Integration into Minimap2 reduces index construction overhead while preserving downstream alignment consistency.