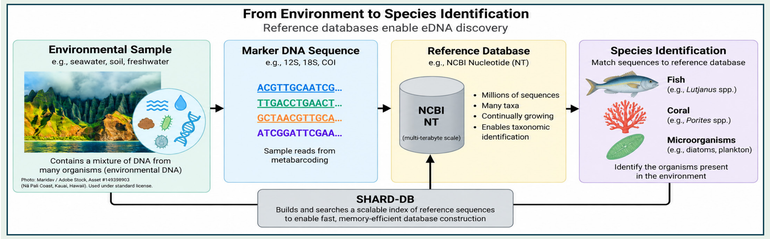
SHARD-DB: Scalable Hash-Based Retrieval for High-Throughput Reference Database Construction

Stephanie Murray, Lenore Pipes

University of Hawai'i at Mānoa

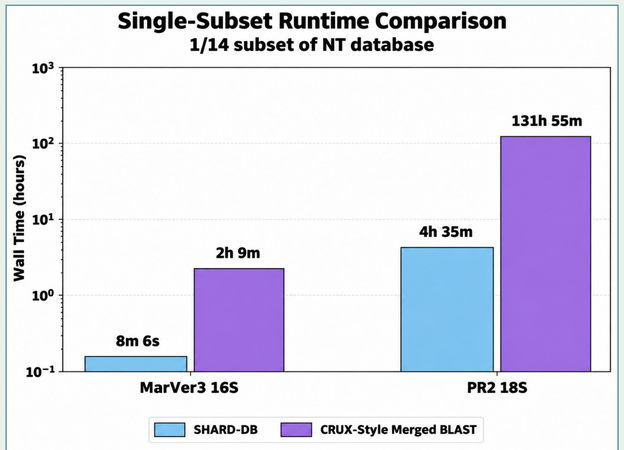
Introduction



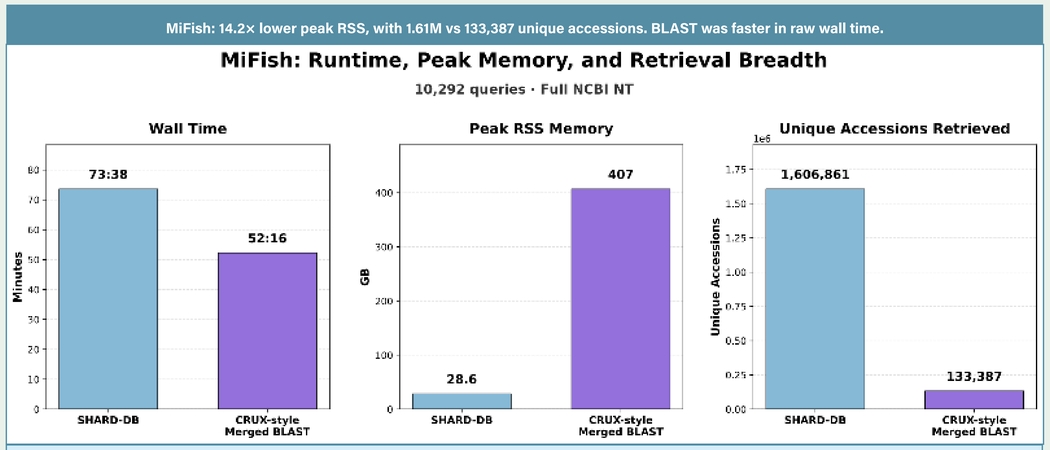
- Environmental DNA (eDNA) can reveal which organisms are present in an environment, but identification depends on the reference sequences available for the targeted genetic marker.
- Building comprehensive marker-specific reference databases requires searching thousands of marker sequences against multi-terabyte nucleotide collections.
- CRUX-style workflows commonly use repeated BLAST searches to expand the initial reference set, making large-scale construction expensive in computation, memory, and data movement.
- SHARD-DB takes a different approach: reference sequences are indexed once using compact MinHash sketches and locality-sensitive hashing (LSH), enabling approximate candidate retrieval before targeted locus localization.
- **GOAL: Make large-scale reference-database expansion practical while preserving broad reference recovery and controlling memory and I/O.**

Results

Performance is workload-dependent: SHARD-DB uses significantly less memory on full-scale MiFish and shows large runtime advantages for MarVer3 and PR2 on matched NCBI NT subsets.



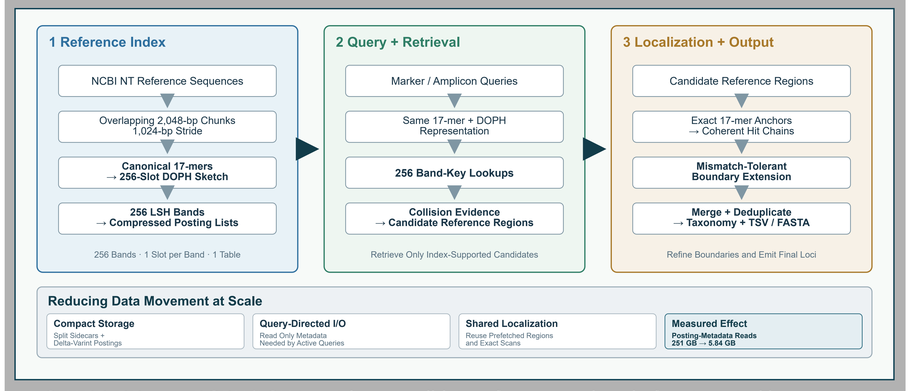
MarVer3 and PR2 show substantial runtime advantages for SHARD-DB on 1/14 NCBI NT subset.



MiFish: 14.2× lower peak RSS, with 1.61M vs 133,387 unique accessions. BLAST was faster in raw wall time.

~12× broader candidate retrieval: SHARD-DB retrieved 1.61M unique candidate accessions vs 133K for CRUX-style Merged BLAST, expanding the reference pool available for database construction.

System Architecture/ Methods



- SHARD-DB uses DOPH/LSH to retrieve candidate reference regions before targeted locus localization.
- The implementation minimizes repeated data movement using query-directed metadata access, compact postings, and shared reference-region processing.

Benchmark System: Dell PowerEdge R760 Intel Xeon Gold 6548Y+ 32 physical cores - 64 threads - 1 TB RAM - 204 TB storage - CPU-only - AVX2

Measured effect: Posting-metadata reads fell from 251 GB to 5.84 GB.

What Is New?

A different reference-database construction strategy: broad MinHash/LSH retrieval followed by targeted locus localization rather than alignment-first expansion.

Conclusion

- **Performance varies with query-set characteristics:** MiFish used 14.2× less peak memory while BLAST was faster in wall time, whereas MarVer3 and PR2 showed large SHARD-DB runtime advantages on matched NT subsets.
- **Broader retrieval enables richer reference databases:** SHARD-DB expands the candidate reference pool across marker datasets, increasing the sequence diversity available for downstream database construction.
- **Validation:** SHARD recovered 99.95% of BLAST-retrieved accessions, and additional SHARD-only candidates were validated independently with semi-global Needleman–Wunsch.

SHARD-DB shows that MinHash/LSH-based retrieval can support large-scale reference-database construction with low memory use and substantial runtime advantages for some marker workloads.

Discussion

- Marker characteristics strongly affect performance: short, conserved MiFish queries produce much broader candidate neighborhoods and more localization work.
- Broader retrieval may increase reference coverage, but also increases processing and makes validation important.
- Future work will characterize recall–precision tradeoffs and parameter choices across marker families.

References

- NCBI Nucleotide Database. National Center for Biotechnology Information. https://www.ncbi.nlm.nih.gov/nucleotide
- BLAST: Basic Local Alignment Search Tool. National Center for Biotechnology Information. https://blast.ncbi.nlm.nih.gov/Blast.cgi

Acknowledgements/Support

Stephanie Murray - PhD student, Computer Science
Lenore Pipes - Advisor / PI, Pipes Lab
UH Mānoa · Pacific Biosciences Research Center
Supported by NIH NIGMS R00GM144747.
Computational resources: UH Mānoa / PBRC

Contact: stephanie.murray@hawaii.edu