Published online 12 12 2024 *Nucleic Acids Research, 2024*

KIRAL: An Efficient Tool For Aligning Reads To 1600+ Killer Immunoglobulin-like Receptor (KIR) Allele Sequences

Erfan Golpour[1], **Amir Negahdarsaber**[1*]

[1]Department of Computer Science, University of Victoria, 3800 Finnerty Road, Victoria BC, V8P 5C2, Canada

ABSTRACT

KIRAL is a tool for aligning sequencing reads to a database of Killer Immunoglobulin-like Receptor (KIR) allele sequences. It takes advantage of high allele similarities within the same KIR gene to perform more efficient aligning compared to the naive implementation with a minor cost of losing some alignment coverage. KIR genes have been linked to controlling immune responses and variation in these genes has been found interconnected with health issues related to the immune system (1).

INTRODUCTION

The KIR gene family plays a crucial role in the immune system by regulating natural killer (NK) cell activity. These genes exhibit high variability with over 1600 known alleles, making their analysis computationally intensive. Traditional alignment approaches often require aligning sequencing reads to the entire KIR allele database, leading to significant redundancy due to the high sequence similarity among alleles of the same KIR gene. KIRAL addresses this challenge by introducing an innovative two-pass approach. In the first pass, representative alleles are selected to approximate read origins. This information guides focused alignments to alleles within the corresponding KIR genes in the second pass, reducing computational overhead. This paper outlines the design, methodology, and performance of KIRAL, highlighting the performance improvement, and alignment coverage loss cost compared to a naive implementation for aligning reads to the KIR allele sequences. KIRAL also allows for parallelization of the alignment process by taking a ’threads’ parameter that allows for more performance improvement and lower obtained run times. Our tool uses minimap2 as its alignment engine. minimap2 has its own optimized alignment algorithms which allowed us to focus on improved preprocessing of the input data to develop faster alignment methods specifically for aligning reads to KIR alleles. KIRAL takes advantage of direct access to minimap2’s C API, ensuring optimal performance which was made possible because we implemented our tool using C++.

MATERIALS AND METHODS

Technologies Used

KIRAL integrates minimap2, a state-of-the-art alignment tool, as its core alignment engine. By building around minimap2, KIRAL avoids duplicating effort in optimizing alignment algorithms. Instead, it enhances preprocessing and refines input data. Implemented in C++, KIRAL takes advantage of direct access to minimap2’s C API, ensuring seamless integration and optimal performance. Compared to Python-based approaches, C++ offers superior handling of large datasets and computationally intensive operations.

Alignment Methods

KIRAL implements three alignment strategies:

Naive Method : Aligns all reads against the entire KIR database. While comprehensive, it is computationally expensive and serves as a baseline for comparison.

- **Pros:** Guarantees maximum accuracy by considering every allele.
- **Cons:** Computationally prohibitive for large datasets, with excessive runtime and memory usage.

Categorical Method : Selects representative alleles for each KIR gene and performs an initial alignment to categorize reads. Subsequent alignments are restricted to alleles within the corresponding gene. This approach balances speed and accuracy, although some reads may be missed if not represented in the initial pass.

- **Pros:** Strikes a balance between speed and accuracy; reduces computational overhead significantly compared to the naive method.
- **Cons:** Risk of missing reads not captured by the representative allele, leading to a potential loss in accuracy.

Regional Method : Builds upon the categorical approach by grouping alignments into regions based on initial matches. Each region is trimmed from the alleles for targeted

[*]To whom correspondence should be addressed. Email: amirnegahdarsaber@uvic.ca

alignments. To address alignment shifts caused by indels, an error buffer extends the regions. While smaller buffers enhance speed, larger buffers improve accuracy at the cost of efficiency.

- **Pros:** Offers the fastest runtime for small error buffers and efficiently focuses alignments on specific regions.
- **Cons:** Highly sensitive to alignment shifts caused by gaps or indels. Accuracy depends heavily on the size of the error buffer, which can slow down performance if too large.

Representative Selection

Currently, representative alleles are chosen randomly. Future work includes identifying “ideal” representatives with minimal sequence divergence, which could significantly improve alignment precision.

RESULTS

We evaluated KIRAL based on key metrics such as runtime and alignment accuracy. The naive method served as the baseline for accuracy, while the categorical and regional methods were analyzed under various configurations to assess their performance and trade-offs.

Benchmarks

Method Performance We assessed the performance of each method in both single-threaded and multi-threaded (8 threads) settings to understand their computational efficiency under different conditions.

Single-threaded Performance The runtime of each method was measured under a single-thread configuration:

- **Naive Method:** Exhibited the longest runtime due to the exhaustive alignment of reads to the entire KIR database. This method is the most accurate but computationally expensive.
- **Categorical Method:** Achieved a significant reduction in runtime by aligning reads only to representative alleles during the first pass and narrowing the search scope in the second pass. The accuracy remained close to the naive method, with minor losses.
- **Regional Method:** Delivered the fastest runtimes, particularly with small error buffers. However, accuracy varied depending on buffer size, balancing speed and precision.

The average runtime in figures does not include the naive method’s runtime.

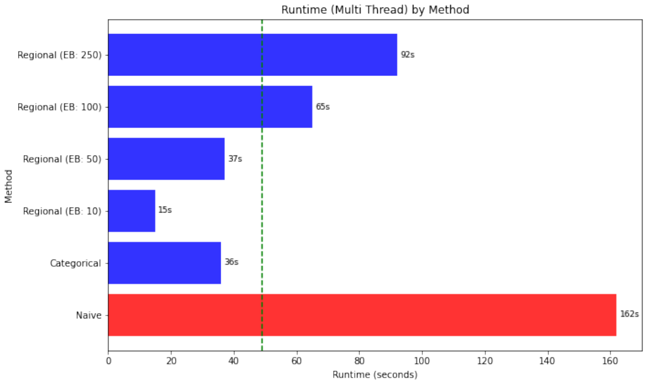

Figure 1. Runtime comparison of KIRAL methods under single-thread configuration.

Multi-threaded Performance (8 Threads) The use of multiple threads significantly improved runtime for all methods, highlighting KIRAL’s scalability:

- **Naive Method:** Still the slowest but showed substantial improvements with threading.
- **Categorical and Regional Methods:** Both demonstrated marked speed gains while maintaining similar relative differences in runtime.

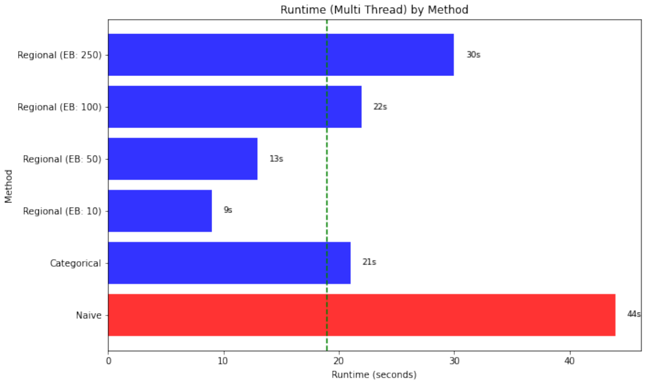

Figure 2. Runtime comparison of KIRAL methods under multi-threaded configuration (8 threads).

Method Accuracy The accuracy of each method was evaluated to determine the trade-offs made for runtime efficiency:

- **Small Buffers (Regional Method):** Provided faster results but lower accuracy due to narrower search ranges.
- **Large Buffers (Regional Method):** Increased accuracy at the cost of runtime, closely resembling the performance of the categorical method.
- **Categorical Method:** Struck a balance between speed and accuracy, making it a versatile option for most use cases.

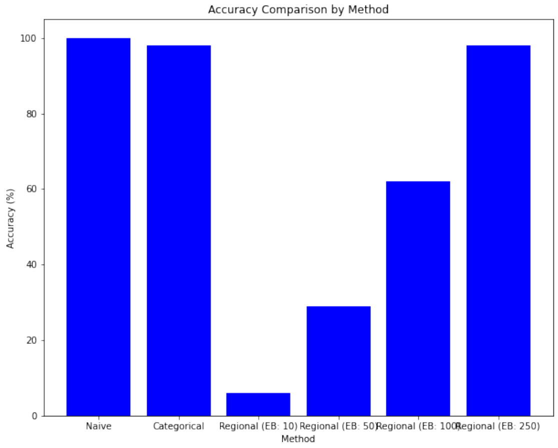

Figure 3. Alignment accuracy of KIRAL methods under various configurations.

Overall Comparison We synthesized the runtime and accuracy results into a radar chart to visually compare the trade-offs:

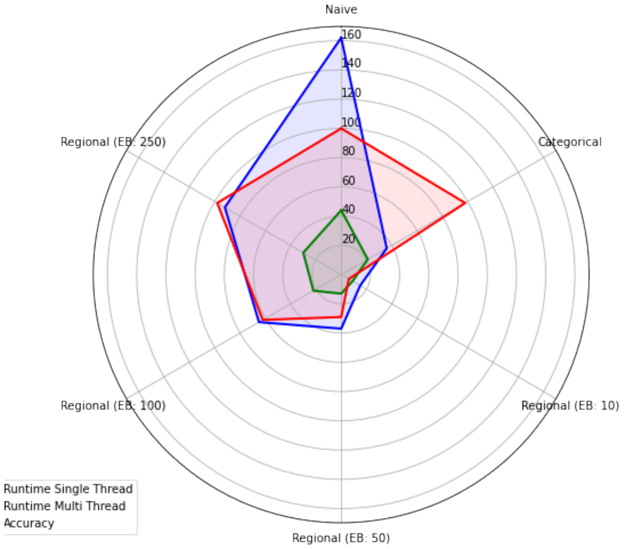

Figure 4. Radar chart comparing runtime and accuracy of KIRAL methods.

Summary KIRAL provides two innovative methods (categorical and regional) that offer clear advantages over the naive method depending on the use case:

- **Categorical Method:** Best suited for scenarios requiring a balance of accuracy and efficiency.
- **Regional Method:** Ideal for rapid alignment needs, especially when small error buffers are acceptable.
- **Naive Method:** While computationally expensive, it serves as a benchmark for maximum accuracy.

DISCUSSION AND FUTURE WORK

Improved Representative Selection

Future updates will focus on defining and identifying optimal representatives. Metrics such as sequence similarity and minimal divergence will guide selection.

Enhancing the Regional Method

Preprocessing steps to detect and handle gaps could address alignment shifts. Enhancing buffer logic and region grouping mechanisms will also improve robustness and accuracy.

Comprehensive Preprocessing with Allele Comparisons

Building a database of differences among alleles within each KIR gene could significantly enhance preprocessing. This database would simplify the approximation of read locations within representative alleles by accounting for shifts and differences in alignment regions. Although this approach requires significant upfront computation, it offers a systematic solution to the challenges posed by alignment shifts in the regional method.

Algorithm Scalability

Further optimization for large datasets will ensure KIRAL remains competitive as sequencing technologies advance.

Improved Representative Selection

Future updates will focus on defining and identifying optimal representatives. Metrics such as sequence similarity and minimal divergence will guide selection.

Enhancing the Regional Method

Preprocessing steps to detect and handle gaps could address alignment shifts. Enhancing buffer logic and region grouping mechanisms will also improve robustness and accuracy.

Algorithm Scalability

Further optimization for large datasets will ensure KIRAL remains competitive as sequencing technologies advance.

CONCLUSION

KIRAL introduces a scalable, efficient approach to aligning reads to the diverse KIR allele database. While the naive method offers maximum accuracy, the categorical and regional methods provide tunable trade-offs between speed and precision. By leveraging C++ and minimap2, KIRAL demonstrates significant improvements over traditional alignment pipelines, particularly for high-throughput genomic studies. Ongoing enhancements will focus on refining accuracy and usability, positioning KIRAL as a valuable tool in computational genomics.

AVAILABILITY AND IMPLEMENTATION

The source code for KIRAL can be found on the following Github repository: https://github.com/erfangolpour/KIRAL

ACKNOWLEDGEMENTS

Thanks to Dr. Numanagić for his continuous support throughout the project.

REFERENCES

1. Carrington M, and Norman P (2003) The KIR gene cluster. *Bethesda, MD: National Center for Biotechnology information*
2. https://lh3.github.io/minimap2/minimap2.html *Manual Page - minimap2*
3. https://github.com/lh3/minimap2 *GitHub - lh3/minimap2, A versatile pairwise aligner for genomic and spliced nucleotide sequences*
4. https://github.com/lh3/minimap2 *GitHub - lh3/minimap2, A versatile pairwise aligner for genomic and spliced nucleotide sequences*