Algorithms in Molecular Biology Course Project

Konstantinos Giatras
Alexandros Kouris

DSIT 2022-2023

Au et al. BMC Genomics (2017) 18:16
DOI 10.1186/s12864-016-3449-9

BMC Genomics

SOFTWARE Open Access

INDELseek: detection of complex insertions and deletions from next-generation sequencing data

Chun Hang Au[1], Anskar Y. H. Leung[2], Ava Kwong[3,4,5], Tsun Leung Chan[1] and Edmond S. K. Ma[1*]

Abstract

Background: Complex insertions and deletions (indels) from next-generation sequencing (NGS) data were prone to escape detection by currently available variant callers as shown by large-scale human genomics studies. Somatic and germline complex indels in key disease driver genes could be missed in NGS-based genomics studies.

Results: INDELseek is an open-source complex indel caller designed for NGS data of random fragments and PCR amplicons. The key differentiating factor of INDELseek is that each NGS read alignment was examined as a whole instead of "pileup" of each reference position across multiple alignments. In benchmarking against the reference material NA12878 genome ($n = 160$ derived from high-confidence variant calls), GATK, SAMtools and INDELseek showed complex indel detection sensitivities of 0%, 0% and 100%, respectively. INDELseek also detected all known germline (*BRCA1* and *BRCA2*) and somatic (*CALR* and *JAK2*) complex indels in human clinical samples ($n = 8$). Further experiments validated all 10 detected *KIT* complex indels in a discovery cohort of clinical samples. *In silico* semi-simulation showed sensitivities of 93.7–96.2% based on 8671 unique complex indels in >5000 genes from dbSNP and COSMIC. We also demonstrated the importance of complex indel detection in accurately annotating *BRCA1*, *BRCA2* and *TP53* mutations with gained or rescued protein-truncating effects.

Conclusions: INDELseek is an accurate and versatile tool for complex indel detection in NGS data. It complements other variant callers in NGS-based genomics studies targeting a wide spectrum of genetic variations.

Keywords: Complex indel, Variant calling, Bioinformatics, Next-generation sequencing

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5217656/

Paper Summary

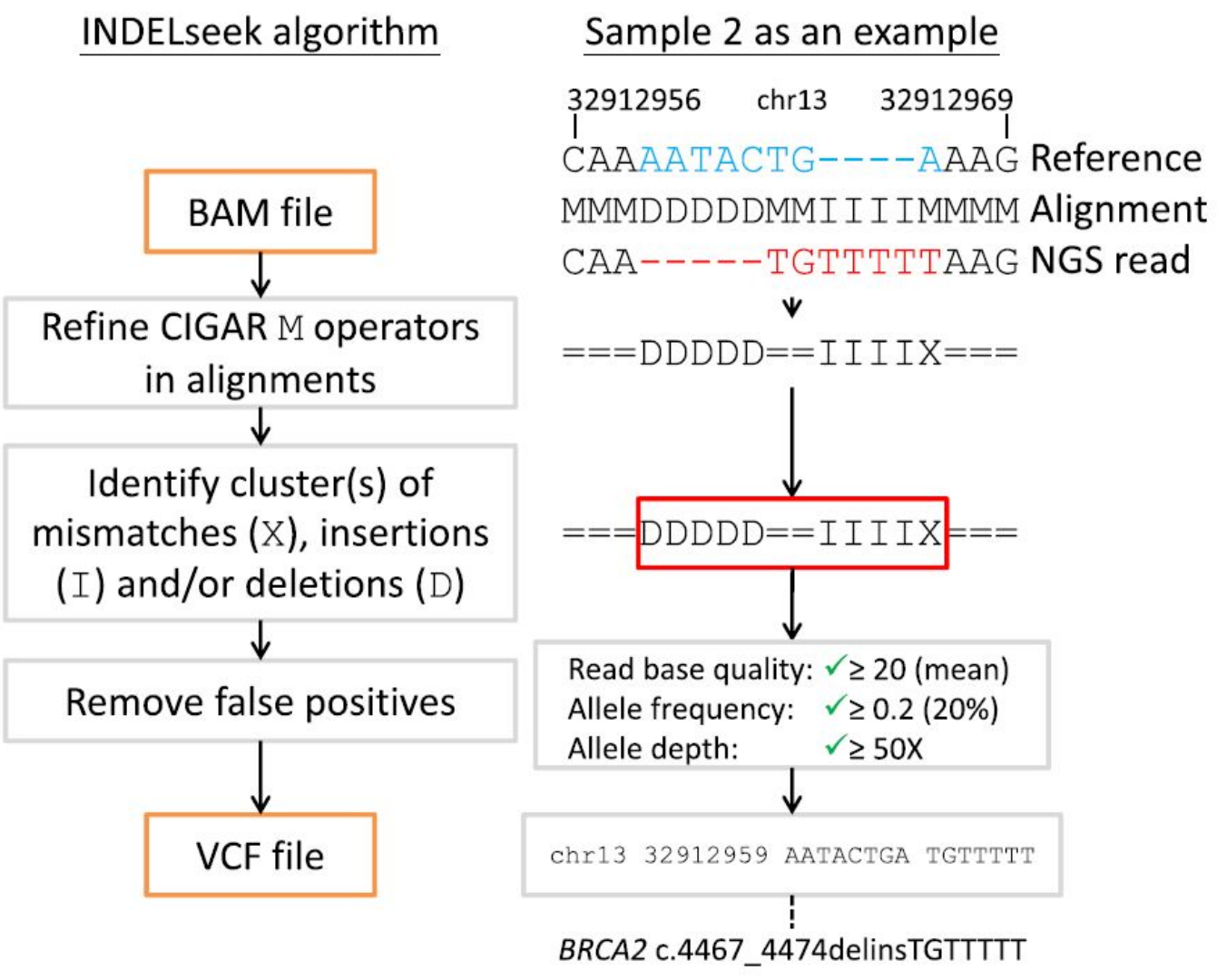
- Complex insertions and deletions (indels) are common genetic variations, formed by simultaneously deleting and inserting DNA fragments of different sizes at a common genomic location. They are primary causes for frameshift mutations and in next-generation sequencing (NGS) data often evade detection by existing variant callers.
- Significant somatic and germline complex indels in critical disease driver genes may be overlooked in NGS-based genomics studies.

Paper Summary

- INDELseek is an open-source tool designed to call complex indels in NGS data from random fragments and PCR amplicons.
- The distinguishing characteristic of INDELseek is its examination of each NGS read alignment in its entirety, as opposed to a "pileup" of each reference position across multiple alignments.
- INDELseek boasts a 100% detection sensitivity rate for complex indels, demonstrating superior performance over other variant callers in complex indel detection, validating its importance in genomics studies.

Datasets

Dataset	Sample count and description	Sensitivity	Specificity
Real NGS data			
1. Protein-coding and flanking regions from whole-genome sequencing (random fragments)	1 (NA12878) 160 putative complex indels 26 negative control loci	100%	100%
2. Hereditary breast and/or ovarian cancer panel (amplicons)	239 3 positive samples (*BRCA1* $n = 1$, *BRCA2* $n = 2$) 236 negative samples	100%	100%
3. Myeloid neoplasm panel (amplicons)	23 5 positive samples (*CALR* $n = 4$, *JAK2* $n = 1$) 18 negative samples (NA12878 and 17 healthy controls)	100%	100%
Semi-simulated data by engineering mutations to real NGS data			
1. Whole-genome sequencing (random fragments)	8671 collected from COSMIC and dbSNP	93.7%	N/A
2. Hereditary breast and/or ovarian cancer panel (amplicons)	237 collected from COSMIC and dbSNP	96.2%	N/A
3. Myeloid neoplasm panel (amplicons)	576 collected from COSMIC and dbSNP	94.6%	N/A


INDELseek algorithm
BAM file
Refine CIGAR M operators in alignments
Identify cluster(s) of mismatches (X), insertions (I) and/or deletions (D)
Remove false positives
VCF file
Sample 2 as an example
32912956
chr13
32912969
CAAAATACTG----AAAG Reference
MMMDDDDDMMIIIIMMMM Alignment
CAA-----TGTTTTTAAG NGS read
===DDDDD==IIIIX===
===DDDDD==IIIIX===
Read base quality: ✓≥ 20 (mean)
Allele frequency: ✓≥ 0.2 (20%)
Allele depth: ✓≥ 50X
chr13 32912959 AATACTGA TGTTTTT
BRCA2 c.4467_4474delinsTGTTTTT

Our Approach

- Reproduce the results of the paper to validate the authors' claims regarding the called variants (accuracy metric).
- Improve the algorithm on various aspects (original algorithm performance metrics, augment Perl script, create Python and C++ implementations to measure and compare performance).
- Compare with another variant discovery tool, namely GATK Haplotyper (accuracy metric).

Datasets We Used

- Downloaded the sample named **NA12878** (BAM file of 113GB size, 1.5bil reads) from the Illumina Basespace.
- Extracted **chr20** and **chr15** from the above dataset.
- chr20: ~33mil reads and chr15: ~42.5mil reads.
- Cancer datasets were **not** provided from the authors.
- For semi-simulated datasets, the list of complex indels were also **not** provided from the authors.
- We additionally downloaded a sample BAM file called **HG00148.chr11** from the 1000 Genomes Project for testing purposes that will be explained shortly.
- HG00148: ~4mil reads.

Hypatia VM

In order to test our implementations in a cohesive environment and not test on different machines the various versions of the algorithm, we created a VM in Hypatia, with the following specs:

- 7 CPU Cores @ 2.6GHz
- 60GB RAM
- 300GB of hard disk space

Reproducing the Results of the Authors

The only available dataset that we could acquire to replicate the results presented, was the NA12878 dataset.

- First run on the Hypatia server was done using the original implementation in Perl in order to get some basic metrics on read speed.
- After two days we stopped the run, as it seemed that the job was not gonna finish in a reasonable amount of time.
- The second step was to extract chr20 and chr15 from the dataset, and try again.
- We stumbled upon the same case of the algorithm running too slow.

Reproducing the Results of the Authors

On our 3rd attempt to get some results, we used the HG00148.chr11 bam file from 1000 Genomes.

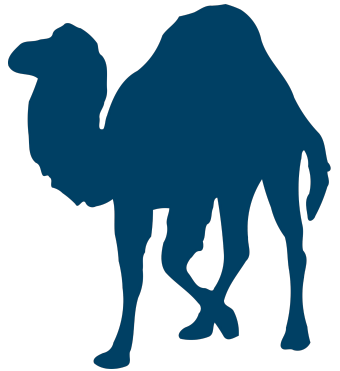
This was run on a **local machine** of ours, finishing after 5 hours and 49 minutes.

So we decided for, at least, testing purposes to use the HG00148 dataset and compare to the Perl output produced by the latest run, on the same local machine.

Augmenting the Perl script

In order to be able to measure the performance of the script we needed to add some metrics, which were also implemented in the other two implementations:

- Progress indicators
- Measure the execution time
- Periodically measure the RAM usage (every one minute) so we can create a RAM usage plot



Python Implementation

- Used dictionaries for caching purposes.
- Used lists instead of arrays in some of the cases of the original implementation.
- Resulted in ~400 lines of code (comparable to the perl script)

C++ Implementation

- All the original filtering options are available via command line arguments.
- Used hash maps for caching purposes.
- Used hash maps instead of arrays in some of the cases of the original implementation.
- Custom structs instead of strings wherever possible, to decrease memory footprint and readability of the code.
- Accompanying Makefile for easy compiling in any machine.
- Resulted in ~900 lines of code (a lot more than the perl script)

Results

After finishing our implementations in Python and C++, we started our runs using the HG00148.chr11 file on the Hypatia server.

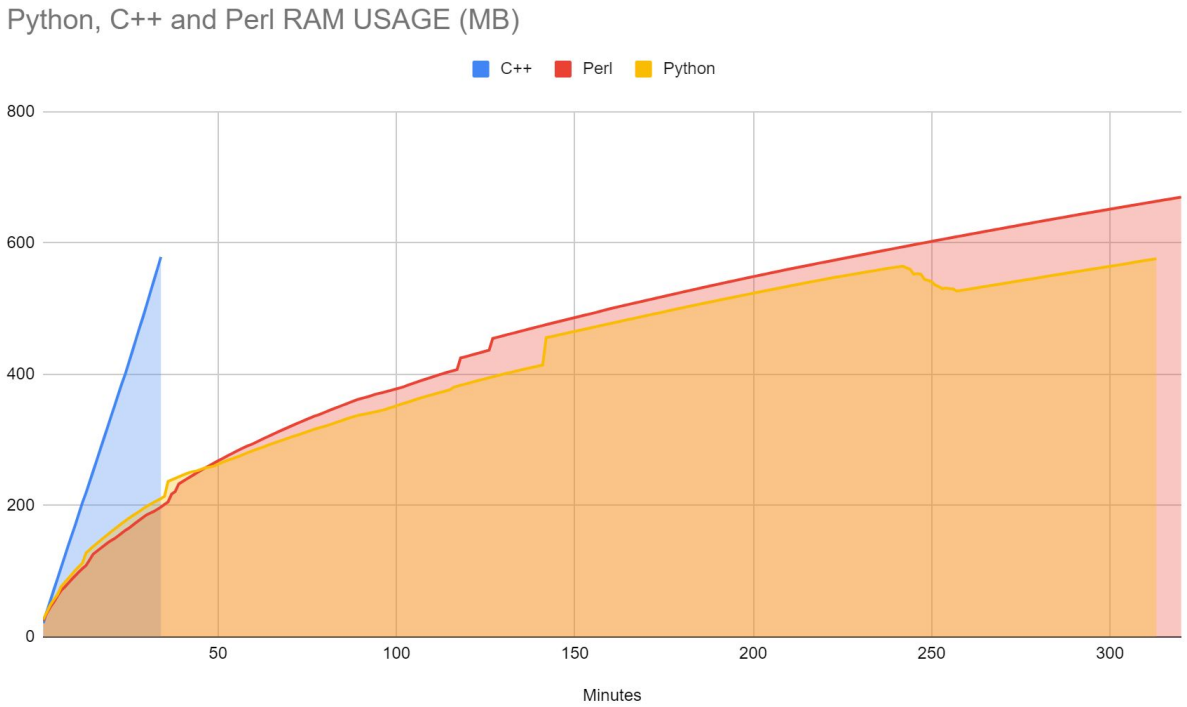
- Running speed in all 3 cases was significantly slower compared to our local runs during development. Based on our progress indicator, what the C++ could read in 2 minutes locally, would take almost 1 hour on the server.
- Our initial thought on this, is that the disk is slower on the server, resulting the program to be I/O bound. This would require further metrics to confirm if necessary.
- So finally our runs were performed locally, but on the same machine (12 cores @ 4.1GHz, 16GB RAM, 500GB SSD)

Results

Execution time:

(HG00148.chr11 of 4mil):

- Perl: 5 hours and 49 minutes
- Python: 5 hours and 42 minutes
- C++: 40 minutes



Results Discussion

- We understood why the authors in the original paper didn’t mention the execution time. It is evident that this tool has not been optimised for use in a production environment, but maybe it could find its way into research applications, where execution time is not so critical.
- Resource footprint seems to be relatively small (~600MB of RAM for 4mil reads file) (running issues on the server could be investigated further)
- Compiled languages have a clear advantage on the performance side, but the coding complexity increases.
- Due to the extremely long execution time for the original BAM file (NA12878), our research focused on performance evaluation and improvement, rather than validating the authors’ results.

Algorithm Improvement Discussion

- Algorithm is dependent on samtools for reading the reference sequences and the depth statistics of each read.
- Querying the reference genome directly or being able to read directly from the BAM file using libraries that support that, and not through samtools, should increase the processing speed.
- Mmap could be used for further increasing the reading speed of such large files. Since the reads are sequential, you don't have random queries in different parts of the file, so you could safely load/unload the pages from RAM.

Potential Next Steps

- Validation: Possible approach could be to focus on a smaller dataset of mutations from the 160 original complex indels and run our algorithms on those chromosomes/regions only.
- Comparison with GATK Haplotyper on the specific BAMs extracted from the above regions.

Thank you for your attention.

Any questions?