# gencore: An Efficient Tool to Generate Consensus Reads for Error Suppressing and Duplicate Removing of NGS data

Shifu Chen[1,2†*], Yanqing Zhou[1†], Yaru Chen[1], Tanxiao Huang[1], Wenting Liao[1], Yun Xu[1], Zhicheng Li[2], and Jia Gu[2]

[1]HaploX Biotechnology, Shenzhen, China
[2]Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China
[†]These authors have contributed equally to this work

*** Correspondence:**
Corresponding Author
Shifu Chen: chen@haplox.com

# Abstract

## Background

Removing duplicates might be considered as a well-resolved problem in next-generation sequencing (NGS) data processing domain. However, as NGS technology gains more recognition in clinical applications (i.e. cancer testing), researchers start to pay more attention to its sequencing errors, and prefer to remove these errors while performing deduplication operations. Recently, a new technology called unique molecular identifier (UMI) has been developed to better identify sequencing reads derived from different DNA fragments. Most existing duplicate removing tools cannot handle the UMI-integrated data. Some modern tools can work with UMIs, but are usually slow and use too much memory, making them not suitable for cloud-based deployment. Furthermore, existing tools rarely report rich statistical results, which are very important for quality control and downstream analysis. These unmet requirements drove us to develop an ultra-fast, simple, little-weighted but powerful tool for duplicate removing and sequence error suppressing, with features of handling UMIs and reporting informative results.

## Results

This paper presents an efficient tool *gencore* for duplicate removing and sequence error suppressing of NGS data. This tool clusters the mapped sequencing reads and merges reads in each cluster to generate one single consensus read. While the consensus read is generated, the random errors introduced by library construction and sequencing can be removed. This error-suppressing feature makes *gencore* very suitable for the application of detecting ultra-low frequency mutations from deep sequencing data. When unique molecular identifier (UMI) technology is applied, *gencore* can use them to identify the reads derived from same original DNA fragment. *gencore* reports statistical results in both HTML and JSON formats. The HTML format report contains many interactive figures plotting statistical coverage and duplication information. The JSON format report contains all the statistical results, and is interpretable for downstream programs.

**Conclusions**

Comparing to the conventional tools like Picard and SAMtools, *gencore* greatly reduces the output data's mapping mismatches, which are mostly caused by errors. Comparing to some new tools like UMI-Reducer and UMI-tools, *gencore* runs much faster, uses less memory, generates better consensus reads and provides simpler interfaces. To our best knowledge, *gencore* is the only duplicate removing tool that generates both informative HTML and JSON reports. This tool is available at: https://github.com/OpenGene/gencore

## Keywords

next-generation sequencing; unique molecular identifier; consensus reads; deduplication

55 # Introduction

56 High-depth next-generation sequencing (NGS) has been widely used for precision
57 cancer diagnosis and treatment [1]. From such deep sequencing data, somatic
58 mutations can be detected to guide personalized targeted therapy or immunotherapy.
59 Recently, circulating tumor DNA (ctDNA) sequencing has been recognized as a
60 promising biomarker for cancer treatment and monitoring. Since the tumor-derived
61 DNA is usually a small part of the total blood cell-free DNA, the mutant allele
62 frequency (MAF) of a variant detected from ctDNA sequencing data can be very low
63 (as low as 0.1%). To detect such low-frequency variants, we usually increase the
64 sequencing depth (can be higher than 10,000x). However, the processes of making
65 NGS library and sequencing are not error-free. Particularly, the library amplification
66 using PCR technology can lead to particular sequences becoming overrepresented [2],
67 and consequently cause some false positive mutations in the result of NGS data
68 analysis.

69 As a result of library amplification, NGS data can have many duplicates. Especially
70 for the high-depth data generated by sequencing low-input DNA, the duplication level
71 can be much higher. Traditionally, we just mark the duplicated reads and remove
72 them before downstream analysis. For low-depth paired-end NGS data, the read pairs
73 of same start and end mapping positions can be treated as duplicated reads derived
74 from a same original DNA fragment [3]. Then, the reads clustered together can be
75 merged to be a single read. Due to the nature that errors usually happen randomly, the
76 inconsistent mismatches in the clustered read group can be removed to generate a
77 consensus read.

78 However, for ultra-deep sequencing, it's possible that two read pairs with same
79 positions are derived from different original DNA fragments. This possibility can be
80 higher when the DNA fragments are shorter. For example, cell-free DNA usually has
81 a peak length of ~167 bp, which is much shorter than the peak length of normally
82 fragmented genomic DNA. To better identify sequencing reads derived from different
83 DNA fragments, a technology called unique molecular identifier (UMI) has been

84 developed. It has been adopted by various sequencing methods such as Duplex-Seq [4]
85 and iDES [5]. With UMI technology, each DNA fragment is ligated with unique
86 random barcodes before any DNA amplification process. The UMIs can be then used
87 for accurate clustering of sequencing reads. UMIs may be applied to almost any
88 sequencing method where confident identification of PCR duplicates by alignment
89 coordinates alone is not possible and/or an accurate quantification is required,
90 including DNA-seq karyotyping [6] and antibody repertoire sequencing [7].
91 Some tools, like SAMtools [8] and Sambamba [9], are commonly used to remove
92 duplicates, but cannot process data with UMIs. Samtools is not efficient since it has to
93 sort the data twice for marking duplicate alignments. Sambamba runs faster, but opens
94 a lot of files (much more than 1024), and may introduce problems when multiple
95 instances are run concurrently. The conventional tool Picard markDuplicates
96 (http://broadinstitute.github.io/picard) is able to handle UMIs, but cannot process bam
97 data with unmapped reads. UMI-Reducer [10] and UMI-tools [11] are two new tools
98 designed for processing UMI-integrated NGS data. However, UMI-Reducer is only
99 suitable for RNA data, and UMI-tools cannot deal with data without UMI-integrated.
100 Furthermore, these tools are usually relative slow and use too much memory, which
101 make them cost ineffective for cloud-based deployment. These unmet requirements
102 drove us to develop a new tool called *gencore*, which is fast and memory efficient,
103 with functions to eliminate errors and remove duplicates by generating consensus
104 reads for NGS data with or without UMIs. Table 1 shows a brief comparison of
105 features of different deduplication or consensus read generating tools.
106
107 **Table 1 Features comparison of different deduplication or consensus read**
108 **generating tools.**

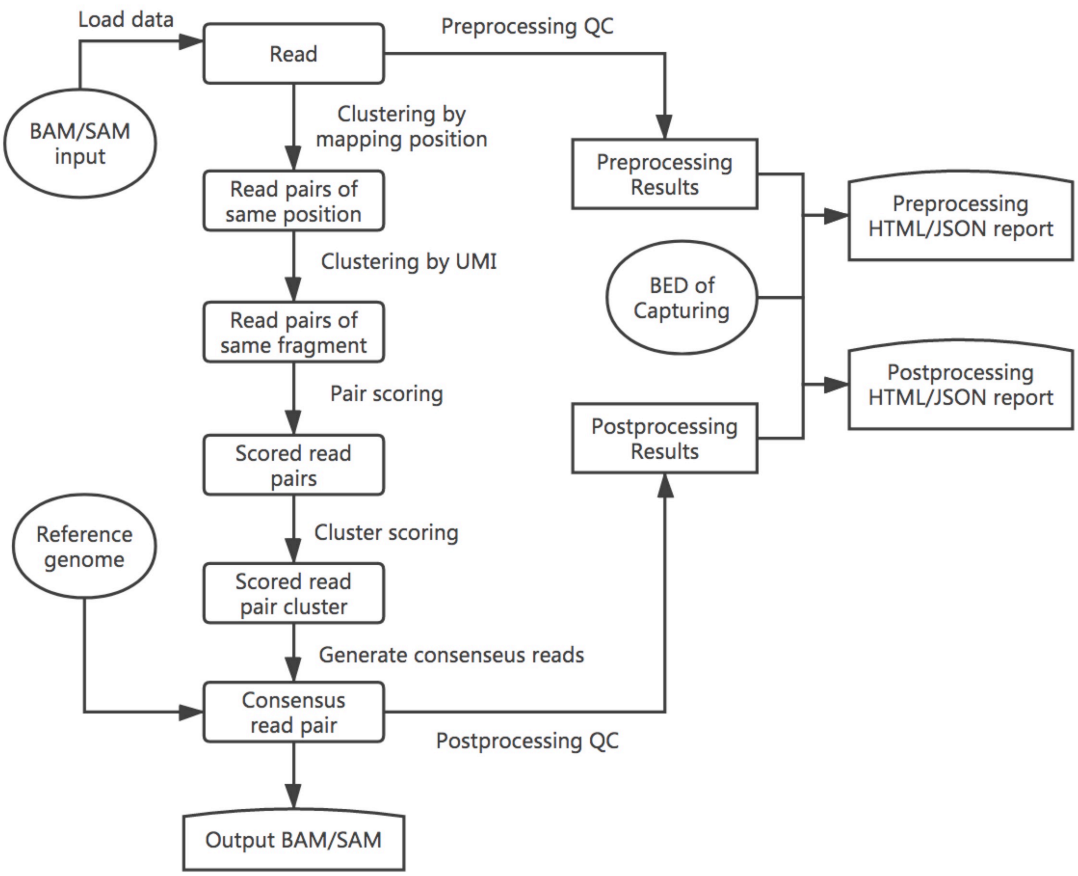
| | **SAMtools** | **Picard** | ***gencore*** | **Picard** | **UMI-tools** | ***gencore*** |
|---|---|---|---|---|---|---|
| | **Non-UMI mode** | | | **UMI mode** | | |
| No need to sort by read name | | + | + | | + | + |
| No need to sort clean up flags | | + | + | | + | + |
| No need to sort add UMI tag | + | + | + | | + | + |
| No need to sort by position again | | + | + | | + | + |

| | | | | | | |
|---|---|---|---|---|---|---|
| JSON Report | + | + | + | + | | + |
| HTML Report | | | | | | + |

In Table 1, the input for these five tools is a sorted bam. SAMtools cannot handle UMIs, whereas UMI-tools is only applicable for UMI-integrated data. Only UMI-tools and *gencore* needn't any extra BAM preporcessing before performing the deduplication. *gencore* reports metrix in JSON and HTML formats whereas UMI-tools doesn't.

# Implementation

*gencore* requires an input of position sorted BAM file and a reference genome FASTA file. If the FASTQ data has UMIs, it can be preprocessed using fastp [12] to move the UMIs from read sequences to read identifiers. The main workflow of *gencore* is described in Fig.1.



123 **Fig.1 The brief workflow of *gencore*.** Besides the input BAM/SAM file, this tool
124 accepts a reference genome input to assist consensus reads generation. If the data is
125 from targeted sequencing, a BED file can also be provided to describe the capturing
126 regions. In this case, the coverage statistics in BED regions will also be reported in the
127 HTML/JSON reports.
128
129 Simply put, *gencore* clusters read pairs by their mapping positions and UMIs (if
130 applicable), and then generates a consensus read for each cluster. The main algorithm
131 of *gencore* can be briefly introduced as following steps:
132 (1) Position clustering: all mapped read pairs are grouped by mapping position first.
133 The reads with same mapping chromosome, start position and end position will be
134 grouped together. A multi-level map [*chr*]:[*left_pos*]:[*right_pos*]:[*read_pairs*], is used
135 to store the clusters being processed, while [*left_pos*] and is [*right_pos*] the read pair's
136 leftmost and rightmost mapping position in the chromosome respectively. [*read_pairs*]
137 is a group of read pairs that share the same mapping positions. To reduce the memory
138 usage, *gencore* implements a processing-while-reading strategy, which means
139 processing one group immediately when its all possible reads are collected. For
140 example, when *gencore* finds that the mapping position of current inputting read is
141 greater than [*right_pos*] of one group, it will perform following step 2 to step 7 for
142 this group and release the group immediately.
143 (2) UMI clustering: for each group of same mapping positions, read pairs are then
144 clustered by their UMIs with one base difference tolerance. If the data has no UMIs,
145 this step is skipped. Due to the principle of Illumina paired-end sequencing, if the data
146 has dual UMIs from forward and reverse reads, the read pairs with reciprocal UMIs
147 will be clustered together. For instance, two read pairs with UMI *ATGC_GCAA* and
148 *GCAA_ATGC* will be considered as derived from different strands of same original
149 DNA fragment, and will be clustered together.
150 (3) Cluster filtering: each cluster will be filtered by comparing its supporting reads
151 number with the threshold (default = 1, which means no threshold). If it passes,
152 *gencore* will start to generate a consensus read for this cluster. For ultra-deep

153 sequencing (i.e. ctDNA sequencing with 10,000× or higher depth), it's recommended
154 to increase the threshold to 2 to discard part of reads that without any PCR duplicates.
155 (4) Pair scoring: a default score number (default = 6) will be initially assigned to
156 every base in the read. For each read pair in a cluster, the overlapped region of the
157 paired reads is computed. For each base in the overlapped region, its score is adjusted
158 according to its consistence with its paired base, with the consideration of their quality
159 scores. The default scoring schema is presented in the project repository, and can be
160 configured through options.
161 (5) Cluster scoring: in this step, the total scores are computed by summarizing the
162 scores computed in previous step. For each position in the mapping region, *gencore*
163 queries the base presented in the cluster's different reads, and summarizes them to
164 compute the score of different bases (A/T/C/G).
165 (6) Consensus read generating: for each position in a cluster, its base diversity is
166 computed according to the scores of different bases computed in last step. If *gencore*
167 finds one dominant base, this base will also be presented in the consensus read.
168 Otherwise if exists one or more reads are concordant with reference genome with high
169 quality, or all reads at this positions are with low quality, the corresponding base in
170 the reference genome will be used. The using of reference genome is one of the major
171 differences between *gencore* and other tools. Since a base is more likely an error
172 when it's not concordant with reference, *gencore* assigns lower weight to them when
173 computing the consensus reads.
174 (7) Buffered reads outputting: when one consensus read is generated, it will be
175 buffered in a position-sorted queue to be written to output. To minimize the memory
176 used by this queue, *gencore* implements a writing-while-processing strategy. With
177 this strategy, *gencore* maintains a pointer that always points to the unprocessed read
178 with least mapping position, and periodically outputs the reads in queue with mapping
179 position less than it.
180 After the processing is done, *gencore* will generate a summary of the data before and
181 after processing. Some metrics like coverage, duplication histogram, mapping rate,
182 duplication rate, passing filter rate and mismatch rate are reported in HTML/JSON

format reports. The HTML report contains no image figures but some interactive figures, which are built based on Plotly.js. Comparing to conventional HTML reports with static images, this single-page standalone JavaScript-based HTML report is much more interactive and easier to transfer. Fig. 2 shows a demonstration of the coverage statistics in both genome scale and capturing regions in the HTML reports.

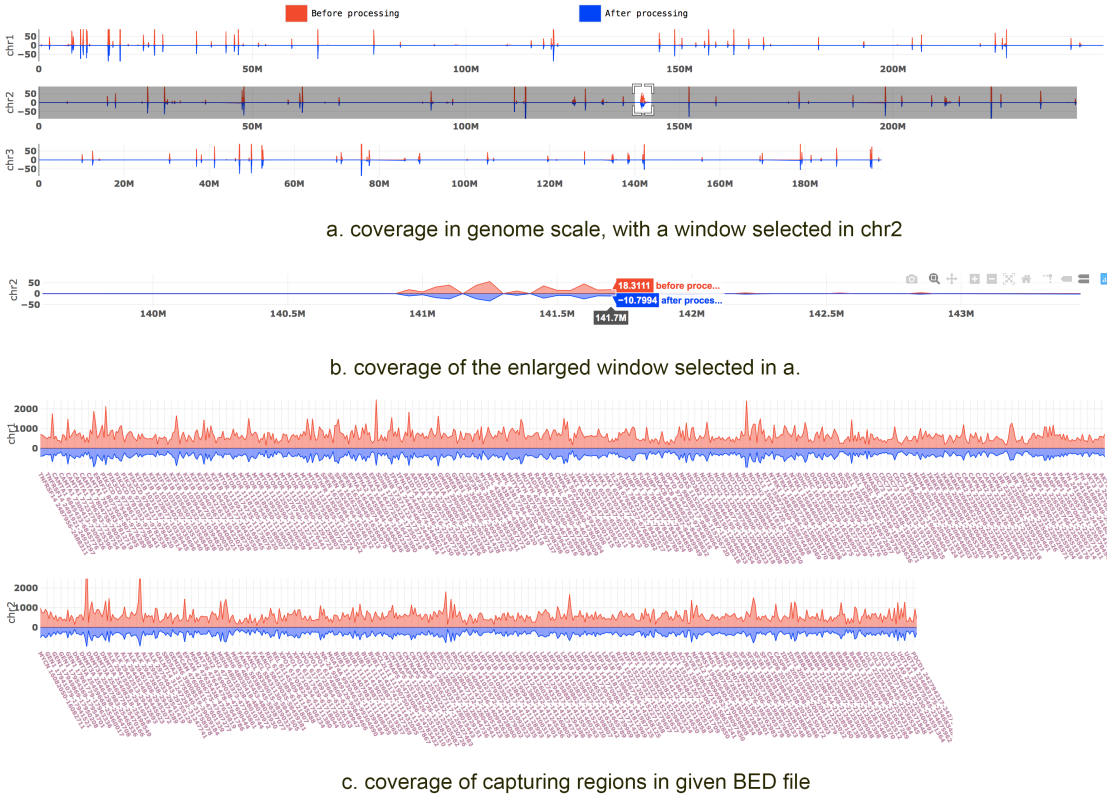



**Fig. 2 The coverage statistics figures in the HTML report.** In this interactive HTML report, a region is selected in a), and then enlarged in b). While a) and b) are coverage in genome scale, c) is the coverage only in the capturing regions. The BAM file of this report was generated by targeted sequencing using a panel with hundreds of genes. So the coverage in genome scale is very sparse, whereas the coverage in capturing regions is high and dense.

## Application

Since *gencore* can be used to reduce sequencing errors, it is very useful for the application of detecting low-frequency somatic mutations from cancer sequencing data, particularly in liquid biopsy technology [16]. When the samples are from blood,

urine or malignant effusion, the MAF of variants can be even much lower than 1%. The detection of such low-frequency variants can be seriously affected by the errors, which are usually introduced by library preparation and sequencing. *gencore* can significantly reduce the sequencing errors of deep sequencing data, and consequently reduce the false positive calling rate.

To evaluate how *gencore* can help the variant detection, we conducted two evaluation experiments using 8 DNA samples, obtained from the National Center for Clinical Laboratories (NCCL) in China. The dataset #1 (1801, 1802, 1803 and 180N) was generated by sequencing plasma cell-free DNA samples, and each data contains ~55G bases. Sample 1801, 1802 and 1803 were DNA extracted from blood of one lung adenocarcinoma patient at different time, whereas sample 180N was DNA from a healthy control. The dataset #2 (1811, 1812, 1813 and 181N) was generated by sequencing tissue DNA samples, and each data contains ~10G bases. Sample 1811 and 1812 were DNA of tumor tissues collected from two breast cancer patients, whereas sample 1813 was DNA of a biopsy tissue collected from a lung adenocarcinoma patient, and sample 181N was DNA from a healthy control. These DNA samples are publicly provided by NCCL as reference materials for conducting inter-lab quality assessments. The golden standard mutations of all samples were also provided by NCCL. Among all the mutations, the lowest frequency was about 0.15%.

In our experiment, all samples sequencing libraries were prepared using IDT xGen Dual Index Adapters, captured with a 451-gene cancer panel, and then sequenced using an Illumina NovaSeq 6000 sequencer. UMI adapters were used for 1801, 1802, 1803 and 180N samples. The detailed file sizes and commands of experiments are provided in Supplementary file 1.

The FASTQ files were preprocessed by fastp, and then mapped to reference genome hg19 using BWA [13]. After the mapped bam file was sorted using Samtools, the sorted bam files were then processed by different tools. VarScan2 [14] was used to call SNVs from the processed bam files, and ANNOVAR [15] was then used to annotate the VCF files. The missense variants detected in the coding sequences were

then filtered with conditions (dataset #1: supporting reads ⩾ 5; dataset #2: supporting reads ⩾ 8 and variant allele frequency ⩾ 2%). The variant calling results were evaluated by comparing to the golden standard results provided by NCCL, and the speed and memory performance were also compared. The comparison result is shown in Fig 3.

**Fig. 3 Comparison of speed, memory peak and processing results of different tools in both UMI and non-UMI modes.** a) memory peak and execution time of different tools. Samtools and Picard (in UMI mode) need to prepare the data before performing deduplication, whereas *gencore* and UMI_tools needn't. b) average depth of output BAM. For the cfDNA samples (1801, 1802 and 1803), the depths of UMI mode results are much higher than non-UMI mode, indicating that over-deduplication may happen when performing deduplication without UMI for ultra-deep sequencing data. c) specificity of downstream variant calling results comparing to the golden standard results provided by NCCL.

From Fig. 3, we can learn that *gencore* runs much faster than all other tools. For the comparison of memory peak, *gencore* uses much less memory than Picard and UMI_tools. Due to *gencore* consumes extra memory to load reference genome and performs more processing, *gencore* uses more memory than Samtools. But, as shown on Fig. 3a, its memory peak is still less than 8GB. This result shows that *gencore* is littleweight and fast, and is much more cost-effective to be deployed on the cloud.

The comparison of downstream variant calling results also shows that *gencore* outperforms other tools. From Fig. 3c, we can learn that *gencore* achieved higher specificity in both UMI and non-UMI modes.

For the cfDNA samples (1801, 1802 and 1803), we applied a filter with condition (supporting reads⩾5). The results showed all tools successfully detected all true positive variants for 1801 and 1802, but non-UMI mode tools missed one true positive variant for 1803 due to its variant allele frequency was too low (VAF = 0.15%). Moreover, we evaluated the detected variants by comparing their VAFs to the golden results, and considered a variant as unacceptable if its VAF exceeded 2 standard deviations. The results showed all non-UMI mode tools resulted in two variants with unacceptable VAFs. For UMI mode, UMI_tools detected one variant with unacceptable VAF, while all variants detected by Picard and *gencore* were acceptable.

For the tissue DNA samples (1811, 1812 and 1813), we applied a filter with condition (supporting reads ⩾ 8 and VAF ⩾ 2%). The results showed that all tools could detect true positive variants at 100% sensitivity. But for sample 1811, both Picard and Samtools reported one false positive variant, while *gencore* achieved 100% specificity for all three samples.

These results suggest that UMI technique is important for detecting variants with ultra-low VAFs, and *gencore* is one of the best tools to process UMI-enabled data due to its superior accuracy and performance.

# Results and discussion

By analysing the output data using downstream tools, *gencore* outperforms other tools in both non-UMI and UMI modes. By carefully exploring the data generated by these different tools, we found the major difference was that *gencore* applied reference genome based correction, whereas Picard and UMI-tools didn't. Utilization of a reference genome is important for eliminating sequencing noises. When an inconsistent position is found when making a consensus read, the reference base should be taken into account since the base different from the reference may have higher probability to be a sequencing error.

To explore how *gencore* eliminates the sequencing errors, we manually compared some the alignment files before and after *gencore* processing. In the case of sample 1802, NM_005228.3(EGFR): c.2369C>T, p.T790M variant was one true positive variant. Fig. 4 shows the alignment visualization illuminated by Integrated Genome Viewer (IGV) for the files before and after processing. In Fig. 4a, which is the original alignment file generated by mapping by BWA [13], the double-line marked mismatch T base is the true positive variant EGFR p.T790M. However, there are also some other mismatch bases, which are false positive mismatches caused by sequencing errors. In Fig. 4b, which is the alignment file after *gencore* processing, we can find these false positive mismatches are gone, while the true positive variant is kept. This result suggests that *gencore* not only removes duplicates, but also eliminates sequencing errors.

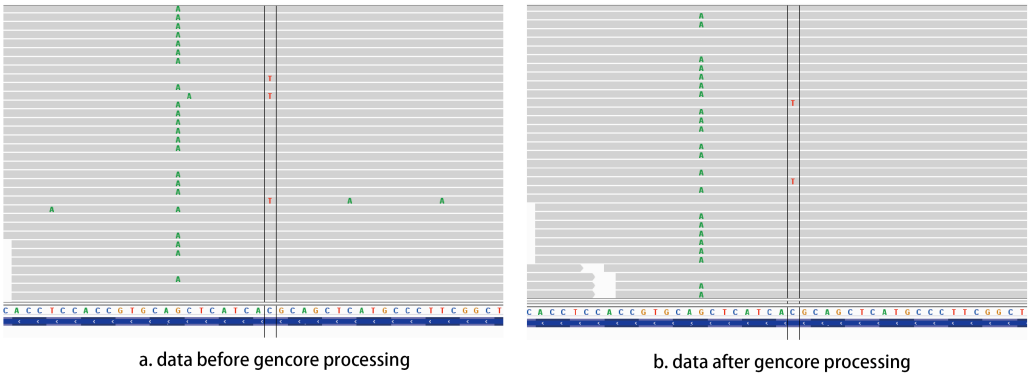



**Fig. 4 Comparison of the alignment files before and after *gencore* processing.** In this figure, the position marked by double lines is NM_005228.3(EGFR): c.2369C>T,

296 p.T790M variant. a) visualizes the mapped reads of original alignment file, b)
297 visualizes the mapped reads after *gencore* processing. We can find that the false
298 positive mismatches, which appear randomly in the original alignment file, are
299 corrected by *gencore*.
300

301 # Conclusion

302 We introduced a tool *gencore*, which is useful for performing deduplication and
303 consensus read generation for deep next-generation sequencing data. We conducted
304 several experiments to evaluate the performance of *gencore*, with comparisons to
305 Picard, Samtools and UMI-tools. The result shows that *gencore* is much faster and
306 more memory efficient, while providing similar or better results. This tool generates
307 interactive HTML reports and informative JSON reports that can help manually
308 checking and programmatically downstream analysis. According to our estimation,
309 this tool has been used to process more than 10,000 samples in the authors' institution,
310 and is now suitable to be adopted by community users.
311

312 **List of abbreviations**

313 ctDNA: cell-free tumor DNA; NGS: next generation sequencing; IGV: integrative
314 genome viewer; MAF: mutated allele frequency; INDEL: insertion and deletion; SNP:
315 single-nucleotide polymorphism; SNV: single-nucleotide variation; EGFR: epidermal
316 growth factor receptor; UMI: unique molecular identifier; HTML: Hypertext Markup
317 Language; JSON: JavaScript Object Notation; NCCL: National Center for Clinical
318 Laboratories;
319

320 # Declarations

321 ***Ethics approval and consent to participate***

Not applicable.

### *Consent to publish*

Not applicable.

**Availability and requirements**

Project name: gencore

Project home page: https://github.com/OpenGene/gencore

Operating system(s): Linux or Mac OS X

Programming language: C++

Other requirements: htslib and zlib

License: MIT License.

### *Competing interests*

The authors declare that they have no competing interests.

### *Funding*

The presented study was funded by Shenzhen Science and Technology Innovation Committee Technical Research Project (Grant No. JSGG20180703164202084) and Shenzhen Strategic Emerging Industry Development Special Fund (Grant No. 20170922151538732).

### *Authors' contributions*

SC designed the algorithm, developed the software and wrote the paper. YZ co-designed the algorithm, YC, YX, WL and TH conducted the testing work and evaluation experiments, ZL and JG contributed to data analysis and paper revision. All authors read and approved the final manuscript.

### *Acknowledgments*

We'd like to thank HaploX Biotechnology for supporting the publication cost. We'd like to express our thanks to the *gencore* community users for testing and reporting

issues. We'd like to thank the OpenGene members for their suggestions of *gencore*'s design.

351 # References


352 [1] Shendure J, Ji H. Next-generation DNA sequencing. *Nat Biotechnol* 2008, 26:
353 1135–45.

354 [2] Aird D, G Ross M, Chen W, Danielsson M, Fennell T, Russ C, et al. Analyzing
355 and minimizing PCR amplification bias in Illumina sequencing libraries. *Genome*
356 *biology* 2011, 12: p.R18.

357 [3] Sims D, Sudbery I, Ilott NE, Heger A, Ponting CP. Sequencing depth and
358 coverage: key considerations in genomic analyses. Nat Rev Genet 2014, 15: 121–32.

359 [4] Kennedy SR, Schmitt MW, Fox EJ, Kohrn BF, Salk JJ, Ahn EH, et al. Detecting
360 ultralow-frequency mutations by Duplex Sequencing. Nat Protoc 2014, 9: 2586-2606.

361 [5] Newman AM, Lovejoy AF, Klass DM, Kurtz DM, Chabon JJ, Scherer F, et al.
362 Integrated digital error suppression for improved detection of circulating tumor DNA.
363 *Nat Biotechnol* 2016, 34: 547-555.

364 [6] Karlsson K, Sahlin E, Iwarsson E, Westgren M, Nordenskjöld M, Linnarsson S.
365 Amplification-free sequencing of cell-free DNA for prenatal non-invasive diagnosis
366 of chromosomal aberrations. *Genomics* 2015, 105: 150–8.

367 [7] Vollmers C, Sit RV, Weinstein JA, Dekker CL, Quake SR. Genetic measurement
368 of memory B-cell recall using antibody repertoire sequencing. *Proc Natl Acad Sci U S*
369 *A* 2013, 110: 13463–8.

370 [8] Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, et al. The Sequence
371 Alignment/Map format and SAMtools. *Bioinformatics* 2009, 25: 2078-2079.

372 [9] Tarasov A, Vilella AJ, Cuppen E, Nijman IJ, Prins P. Sambamba: fast processing
373 of NGS alignment formats. *Bioinformatics* 2015, 31: 2032-4.

374 [10] Mangul S, Driesche SV, Martin LS, Martin KC, Eskin E. UMI-Reducer:
375 Collapsing duplicate sequencing reads via Unique Molecular Identifiers. *bioRxiv* 2017,
376 103267.

377 [11] Smith T, Heger A, Sudbery I. UMI-tools: Modelling sequencing errors in Unique
378 Molecular Identifiers to improve quantification accuracy. *Genome Res* 2017, 27:
379 491-499.

380 [12] Chen S, Zhou Y, Chen Y, Gu J. fastp: an ultra-fast all-in-one FASTQ
381 preprocessor. Bioinformatics 2018, 34: 884-890.
382 [13] Li H, Durbin R. Fast and accurate short read alignment with Burrows-Wheeler
383 transform. Bioinformatics 2009, 25: 1754-1760.
384 [14] Koboldt DC, Zhang Q, Larson DE, Shen D, McLellan MD, Lin L, et al. VarScan
385 2: somatic mutation and copy number alteration discovery in cancer by exome
386 sequencing. Genome Res 2012, 22: 568-576.
387 [15] Wang K, Li M, Hakonarson H. ANNOVAR: functional annotation of genetic
388 variants from high-throughput sequencing data. Nucleic Acids Res 2010, 38: e164.
389 [16] Esposito A, Criscitiello C, Trapani D, Curigliano G. The Emerging Role of
390 "Liquid Biopsies," Circulating Tumor Cells, and Circulating Cell-Free Tumor DNA in
391 Lung Cancer Diagnosis and Identification of Resistance Mutations. Curr Oncol Rep
392 2017, 19: 1.