Flexiplex: A versatile demultiplexer and search tool for omics data

Nadia M. Davidson[1,2]✉, Noorul Amin[1,2], Ling Min Hao[3], Changqing Wang[1,2], Oliver Cheng[1,2], Jonathan Göke[3], Matthew E. Ritchie[1,2], Shuyi Wu[1]

[1]Walter and Eliza Hall Institute of Medical Research, Parkville, Australia
[2]Department of Medical Biology, Faculty of Medicine, Dentistry and Health Sciences, The University of Melbourne, Australia
[3]Genome Institute of Singapore, Singapore

✉corresponding, lead and senior author: davidson.n@wehi.edu.au
Note: authors are listed in order of contribution

Abstract

The process of analyzing high throughput sequencing data often requires the identification and extraction of specific target sequences. This could include tasks such as identifying cellular barcodes and UMIs in single cell data, and specific genetic variants for genotyping. However, the existing tools which perform these functions are often task-specific, such as only demultiplexing barcodes for a dedicated type of experiment or are not tolerant to noise in the sequencing data. To overcome these limitations, we developed Flexiplex, a versatile and fast sequence searching and demultiplexing tool for omics data, which is based on the Levenshtein distance and thus allows imperfect matches. We demonstrate Flexiplex's application on three use cases, identifying cell line specific sequences in Illumina short read single cell data, and discovering and demultiplexing cellular barcodes from noisy long-read single cell RNA-seq data. We show that Flexiplex achieves an excellent balance of accuracy and computational efficiency compared to leading task-specific tools. Flexiplex is available at https://davidsongroup.github.io/flexiplex/.

Introduction

High throughput sequencing, from both short-read and long-read sequencing technologies, enable the genome, transcriptome and other 'omics to be profiled by reading up to billions of nucleotides. Matching and extracting sequences from 'omics data is a common analysis task. For example, searching for the presence of genetic variants or motifs, or, identifying and error correcting barcodes. Highly efficient string search command line tools such as grep and agrep [1], which find exact and approximate matches respectively, have been used for this purpose in the past [2]. However, they are designed for a small set of search patterns, and do not output the data in a format compatible with downstream analysis tools, thereby limiting their use for demultiplexing. Demultiplexing is becoming increasingly routine in 'omics data analysis. For instance, indexes, including barcodes and Unified Molecular Identifiers (UMIs), are added for a range of purposes such as pooling samples to save sequencing cost, measuring clonal expansion, labeling reads from individual cells, or individual molecules [3–6]. To address this, demultiplexing tools have been built such as ultraplex [7] for short-read data, and scTagger [8] and BLAZE [9] for noisy long-read data. Some demultiplexing tools have also been built into multi-

purpose pipelines such as FLAMES [10], sockeye [11] and SiCeLoRe [12]. However these tools are often tailored to a specific experiment type, and may not work on a wider range of customized barcodes and sequencing data, specifically, when the structure of the barcode, flanking sequences, and their locations differ from the usual specifications. For this reason, experiment agnostic tools have begun to be developed [13].

Long-read demultiplexing is a particular challenge, as the reads often contain a high number of errors, including insertions and deletions [14]. Moreover, a small percent of reads may be chimeric [15] containing multiple barcodes and require read splitting. Although there are some tools which can simultaneously split chimeric reads and demultiplex, such as poreChop [16], these may not work across a broad range of datasets or when there are thousands of barcodes. Finally, much of the available software is complex to setup and install and their computational requirements in terms of time and memory can limit their practical use when processing the volumes of data now being generated.

Here we introduce Flexiplex, which given a set of reads as either FASTQ or FASTA, will demultiplex and/or identify a sequence of interest, reporting matching reads and read-barcode assignment. Flexiplex works in two modes; a) when one or more sequences of interest are known, such as barcodes and b) discovery mode – when only the sequence which flanks the region of interest is known. Flexiplex is lightweight, multithreaded, fast and easy to install and run. The source code and pre-built binary executables for Linux and MacOS are provided at https://davidsongroup.github.io/flexiplex/.

Flexiplex Approach

Flexiplex assumes a read structure where a barcode and UMI are flanked by other known sequences (Figure 1A). To identify the barcode, Flexiplex first uses the Edlib C++ library [17] to search for the left and/or right flanking sequence. Default sequences are 22bp (base pairs) from the 10x Genomics Chemistry v3 primer and 9bp of poly T sequence. The intermediate sequence which contains the barcode and UMI, is initially left as a wildcard (default length of 28bp). For the best match to the flanking sequence within a specified edit distance (default edit distance of 8), the intermediate sequence (barcode and UMI) +/- 5 bp either side is extracted. Then, a dynamic programming algorithm implemented in Flexiplex is used to align the extracted sequence against a user-provided list of known barcodes using the Levenshtein distance [18]. The best matching barcode equal or less than a specified distance (default of two) is reported, but only if no other barcode has equal lowest distance. If a known barcode list is not provided, the sequence following the left flank is assumed as the barcode, and barcode frequencies are reported. To identify and split chimeric reads, Flexiplex will repeat the flank and barcode search with the previously found barcode and flanking region masked out. This is repeated until no new barcodes are found in the read. Subsequently, the reverse complement of the unmasked read is searched using the same algorithm.

The flank and barcode sequences provided to Flexiplex are customisable, as is the length of the barcode and UMI. Flexiplex will output identified barcodes, UMIs, and trimmed reads by default.

If the barcode, UMI and right flank are left unspecified, Flexiplex can be used as a general searching tool and will identify any read containing the left flanking sequence within a user defined distance. As it can read and write to standard input, the output of Flexiplex can be piped to another instance of Flexiplex for complex search and demultiplexing applications, such as extracting cell barcodes from all reads which contain a genetic variant of interest.

To illustrate the performance of Flexiplex, we considered several typical use cases; searching for specific sequences of interest in low error short-read data and discovering and demultiplexing single cell RNA sequencing barcodes from noisy long-read data. In all instances Flexiplex demonstrated an excellent balance between computational efficiency and accuracy. However these use cases are not exhaustive. Flexiplex is versatile, with a wide range of applications beyond the examples presented here, such as clonal tagging of single cells using barcoding approaches and sample demultiplexing in bulk data.

Use Cases

Fast and accurate sequence search

We first show Flexiplex is a reliable and general sequence searching tool using an Illumina short-read single cell RNA-Seq dataset from Chen et al. [19]. This data was generated from a mixture of 8 cell lines including MCF-7 which is known to harbor the highly expressed fusion gene BCAS4-BCAS3 [20,21]. The pool also included the HEK293T cell line which expresses E1A, an adenovirus 5 gene. Here we demonstrate how reads of interest can be extracted using a query sequence and then processed for downstream analysis. In this use case, identifying reads from BCAS4-BCAS3 and Adenovirus E1A and using them to assign cells to their cell line of origin.

Using Flexiplex, we searched the reads for 34bp from each of Adenovirus E1A and the dominant breakpoint of BCAS4-BCAS3 within an edit distance of two. As BCAS4-BCAS3 has an MCF-7 specific Single Nuclear Polymorphism (SNP) 13bp from the breakpoint, we included the SNP within the query sequence. Three alternative methods, grep which looks for an exact match, and ugrep [22] and seqkit grep [23], which both allow mismatches, were compared. Flexiplex processed approximately 200 million reads in 24 minutes using a single thread per search (Linux high-performance computer). The computational speed was slower than grep (which took 1-2 minutes), but significantly faster than the other mismatch tolerant methods, seqkit grep (~190 minutes) and ugrep (~40 minutes).

To translate identified reads into a list of cells expressing the novel sequence, we extracted cellular barcodes (first 16bp) from the pair of matched reads. The cells identified fell into clusters based on gene expression as expected (Figure 1B), allowing us to assign clusters to cell lines and assess the performance of each tool. Flexiplex identified the highest or equal highest number of cells expressing BCAS4-BCAS3 (373 of 982 cells) or Adenovirus E1A (445 of 885 cells) in the correct cluster (Figure 1C). The importance of allowing mismatches was exemplified by BCAS4-BCAS3. When using the reference allele in our query sequence rather than the MCF-7 specific

SNP, grep reported no matching reads, whereas Flexiplex identified 97% of those found using the MCF-7 variant (Figure 1C).

False positives, assessed by cells expressing the novel genes from other cell line clusters, were at a similar rate for all tools. We hypothesize that the false positives are unlikely to be a result of poor sequence matching, but rather, a result of doublets, sequencing errors in the cellular barcodes, ambient RNA, or uncertainty in gene expression clustering. Taken together, these results demonstrate that error tolerance improves sensitivity without compromising precision, even for low error Illumina data.

Demultiplexing cellular barcode from noisy long-read data

Barcode demultiplexing is an application of sequence searching, where the number of query sequences (barcodes) can be large and the best match should be reported. We validated Flexiplex's demultiplexing performance on noisy long-read Oxford Nanopore Technology (ONT) single cell RNA sequencing. We compared Flexiplex's performance with two specialized tools for long-read single cell demultiplexing, scTagger and FLAMES, on simulated data generated by Ebrahimi et al [8]. The simulation consisted of 5 million reads generated for 5 thousand cells. True barcode sequences were provided to each tool to benchmark them under a best-case scenario. Flexiplex reported the correct barcodes for more reads than any other methods across all maximum edit distances tested (0-3) and had the lowest false discovery rate (Figure 1D).

Next, we verified these results on a real dataset, scmixology 2 from Tian et al. [10]. Scmixology 2 is a pool of five cell lines which underwent FLT-Seq [24], where cells were processed with the 10x Genomics 3' v3 protocol followed by ONT long-read cDNA sequencing in addition to Illumina short-read sequencing. Approximately 25 million ONT reads were generated. 248 cells were identified in the matched short-read sequencing and their barcodes passed to the long-read demultiplexing tools.

To test the demultiplexing accuracy, we used two orthogonal methods to determine a read's cell line of origin and assessed their concordance, where one method relied on the cellular barcodes being correct. Namely, cellular barcodes had previously been assigned to cell lines using short-read gene-expression clustering [10]. Reads could therefore be assigned to cell lines by correctly demultiplexing their cellular barcodes. For a subset of 18 thousand reads, we could also determine the cell line using cell line specifics Single Nucleotide Polymorphisms (SNPs). For each of these SNP-typed reads, correct barcode assignment was inferred if the predicted cell line from each approach matched.

Using this method we estimated that Flexiplex correctly demultiplexed the highest number of reads of any tools, whilst assigning incorrect barcodes in less than 5% of reads (Figure 1E). Importantly, we found that 3% of the reads in the dataset were chimeric, and Flexiplex's ability to split these improved the demultiplexing performance further. Flexiplex took 2.68 hours to process 25 million long reads (default settings, single thread, Linux high-performance computer), which

was slower than FLAMES (1.28 hours), but faster than scTagger (3.62 hours). Both Flexiplex and FLAMES consumed less than 1 GB of RAM, compared to 41 GB for scTagger.

Discovering cellular barcode from noisy long-read data

Next we assessed Flexiplex's performance at barcode discovery, a scenario where the barcodes are unknown, and a list of valid barcodes need to be generated prior to demultiplexing with tools such as those in the previous use case. This is one of the first analysis steps required for long-read single-cell sequencing without matched short-reads. We compared Flexiplex's barcode discovery against two purpose built tools, BLAZE and Sockeye on the scmixology 2 dataset, and three technical replicates of human induced pluripotent stem cells (hiPSC) sequenced with varying depths and error rates by You et al. [9]. The associated barcodes derived from short-read data were used as truth, but not provided to the tools. Flexiplex's sensitivity and specificity were similar to the other tools (Figure 1F). Significantly, we found that Flexiplex (run with four threads, other setting default, Linux high-performance computer) completed both barcode discovery and subsequent demultiplexing in approximately 10% of the time that BLAZE took for barcode discovery alone, and 20% of the time that sockeye took for barcode region identification (call_adapter_scan) (Figure 1G). This advantage was most striking on the largest dataset (hiPSC, 62 million reads), where BLAZE took over 16 hours compared to 1.7 hours for Flexiplex, and Sockeye failed to complete due to an out-of-memory issue.

Discussion

Here we present Flexiplex, a generalized tool for sequence searching and demultiplexing. To achieve computational efficiency on large 'omics data, Flexiplex uses a combination of two Levenshtein distance alignment algorithms. Hence Flexiplex is tolerant to substitutions, insertions and deletions which can be present in the data due to sequencing errors and SNPs. Flexiplex is highly customizable, including all search sequences and lengths, and whether to split chimeric reads, making it a versatile tool for many applications.

Using barcode discovery and demultiplexing of single cell long-read RNA-Seq, and sequence searching in single cell short-read RNA-Seq as use cases, we demonstrate that Flexiplex achieves accuracy comparable or exceeding popular task-specific tools, with good run-time and memory usage. However, Flexiplex's applications are broader - beyond single cell and RNA sequencing. For example, demultiplexing pooled samples generated by bulk long read experiments, or custom clonal barcodes. Because Flexiplex can read and write to standard IO, instances of itself can be chained together in a single line of bash for more complex tasks, for example, to select reads with a mutation of interest and then demultiplex their cellular barcodes, or to search for phased variants or splicing. We designed Flexiplex to be a simple, user-friendly and fast command line utility, with no dependencies, making it straightforward to install and run. It addresses the need for a lightweight tool to rapidly search and extract subsets of reads from raw data and can be easily integrated into comprehensive downstream data analysis pipelines.

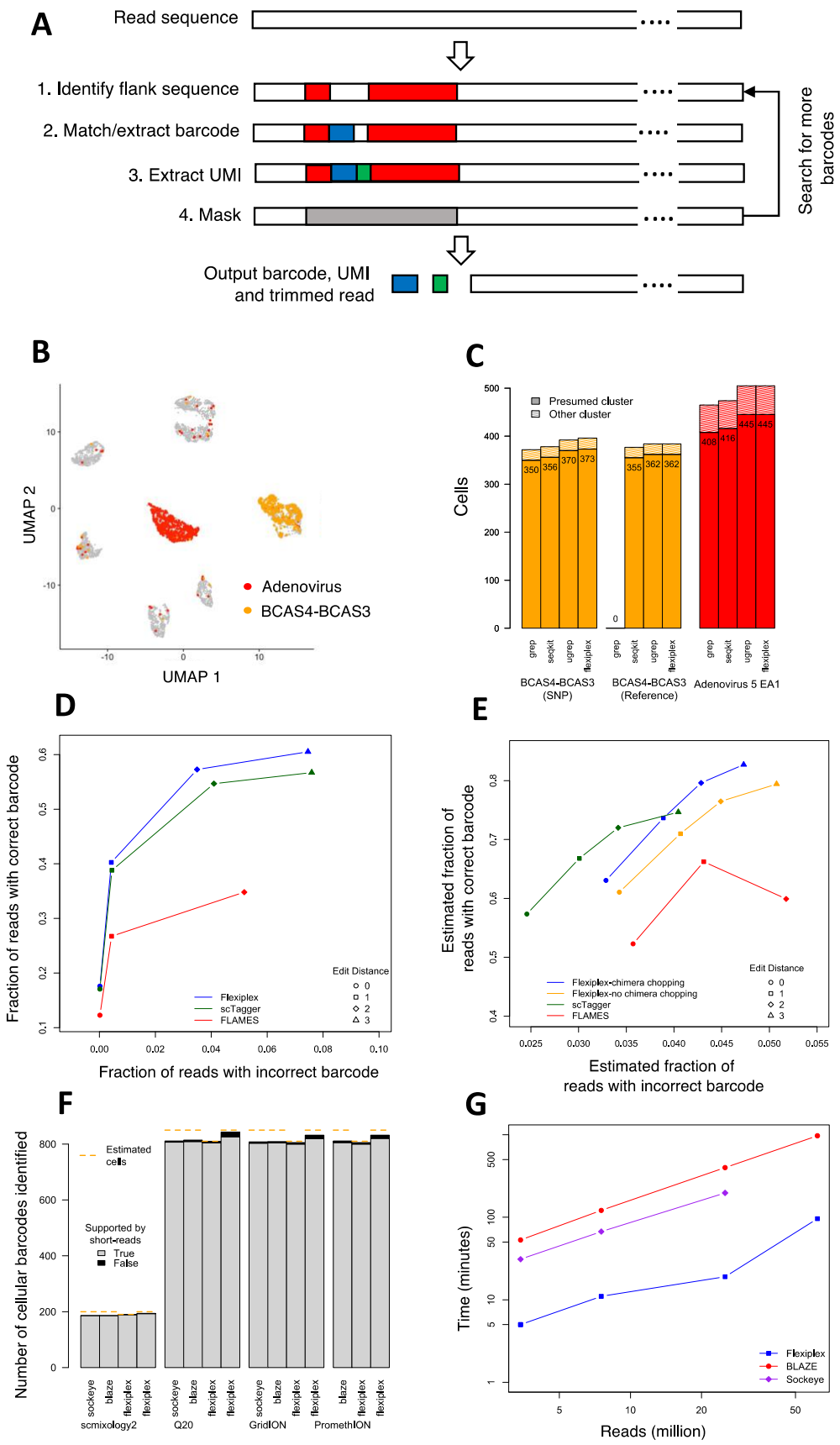
A
Read sequence
1. Identify flank sequence
2. Match/extract barcode
3. Extract UMI
4. Mask
Search for more barcodes
Output barcode, UMI and trimmed read
B
UMAP 2
UMAP 1
Adenovirus
BCAS4-BCAS3
C
Cells
Presumed cluster
Other cluster
350
356
370
373
0
355
362
362
408
416
445
445
grep
seqkit
ugrep
flexiplex
BCAS4-BCAS3 (SNP)
BCAS4-BCAS3 (Reference)
Adenovirus 5 EA1
D
Fraction of reads with correct barcode
Fraction of reads with incorrect barcode
Flexiplex
scTagger
FLAMES
Edit Distance
E
Estimated fraction of reads with correct barcode
Estimated fraction of reads with incorrect barcode
Flexiplex-chimera chopping
Flexiplex-no chimera chopping
scTagger
FLAMES
Edit Distance
F
Number of cellular barcodes identified
Estimated cells
Supported by short-reads
True
False
sockeye
blaze
flexiplex
scmixology2
Q20
GridION
PromethION
G
Time (minutes)
Reads (million)
Flexiplex
BLAZE
Sockeye

***Figure 1**: (A) The demultiplexing approach used by Flexiplex. The right and left flank are first searched for within a read. The barcode and UMI regions are then extracted from the intermediate sequence, with barcode error correction if known barcodes are provided. (B) UMAP of the short-read single cell dataset of 8 pooled cells lines. Cells positive for BCAS4-BCAS3 (orange) and Adenovirus 5 EA1 (red) are indicated. (C) The number of cells identified with grep, seqkit grep, ugrep and Flexiplex that express sequence from BCAS4-BCAS3 (SNP - using an MCF-7 specific variant or Reference - using the reference allele) and Adenovirus 5 EA1 in a short-read single cell dataset of 8 pooled cells lines. Cells which cluster away from the presumed cluster (hatched) are likely to be false positives, whereas those falling within the presumed cluster are true positives (values on bars) (D) The accuracy of barcode demultiplexing on a simulated set of 5 million single cell RNA-seq long-reads for Flexiplex, scTagger and FLAMES, varying the maximum allowed edit distance to known barcodes between 0 to 3. (E) The estimated accuracy of barcode demultiplexing on a real dataset of 248 cells sequenced with ONT for Flexiplex (with and without chimeric read splitting), scTagger and FLAMES, varying the maximum allowed edit distance to known barcodes between 0 to 3. (F) The number of barcodes recovered across four datasets when no known barcode list was provided, but the number of expected cells was given (orange dashed lines). As Flexiplex does not filter empty drops, we evaluated its performance for two estimated cell numbers: determined manually via knee plot, and using the same number of estimated cells as provided to BLAZE and sockeye. (G) The run-time (log scale, four threads) of Flexiplex, BLAZE and sockeye as a function of the number of reads processed from the four datasets used for barcode discovery evaluation.*

Acknowledgement

We would like to acknowledge Alicia Oshlack for invaluable discussions and feedback on the manuscript. We would also like to thank Givanna Putri, Belinda Phipson and Edward Yang for testing Flexiplex and providing feedback. NMD is funded by NHMRC Investigator Grant GNT2016547.

References

1. Wu S, Manber U. Agrep — a fast approximate pattern-matching tool. USENIX [Internet]. Available from: https://www.usenix.org/legacy/publications/library/proceedings/wu.pdf

2. Panagopoulos I, Gorunova L, Bjerkehagen B, Heim S. The “grep” command but not FusionMap, FusionFinder or ChimeraScan captures the CIC-DUX4 fusion gene from whole transcriptome sequencing data on a small round cell tumor with t(4;19)(q35;q13). PLoS One. 2014;9:e99439.

3. Smith AM, Heisler LE, St Onge RP, Farias-Hesson E, Wallace IM, Bodeau J, et al. Highly-multiplexed barcode sequencing: an efficient method for parallel analysis of pooled samples. Nucleic Acids Res. 2010;38:e142.

4. Bramlett C, Jiang D, Nogalska A, Eerdeng J, Contreras J, Lu R. Clonal tracking using embedded viral barcoding and high-throughput sequencing. Nat Protoc. 2020;15:1436–58.

5. Merino D, Weber TS, Serrano A, Vaillant F, Liu K, Pal B, et al. Barcoding reveals complex clonal behavior in patient-derived xenografts of metastatic triple negative breast cancer. Nat Commun. 2019;10:766.

6. Philpott M, Watson J, Thakurta A, Brown T Jr, Brown T Sr, Oppermann U, et al. Nanopore sequencing of single-cell transcriptomes with scCOLOR-seq. Nat Biotechnol. 2021;39:1517–20.

7. Wilkins OG, Capitanchik C, Luscombe NM, Ule J. Ultraplex: A rapid, flexible, all-in-one fastq demultiplexer. Wellcome Open Res. 2021;6:141.

8. Ebrahimi G, Orabi B, Robinson M, Chauve C, Flannigan R, Hach F. Fast and accurate matching of cellular barcodes across short-reads and long-reads of single-cell RNA-seq experiments. iScience [Internet]. 2022 [cited 2023 Mar 26];25. Available from: http://www.cell.com/article/S258900422200801X/abstract

9. You Y, Prawer YDJ, De Paoli-Iseppi R, Hunt CPJ, Parish CL, Shim H, et al. Identification of cell barcodes from long-read single-cell RNA-seq with BLAZE. Genome Biol. 2023;24:66.

10. Tian L, Jabbari JS, Thijssen R, Gouil Q, Amarasinghe SL, Voogd O, et al. Comprehensive characterization of single-cell full-length isoforms in human and mouse with long-read sequencing. Genome Biol. 2021;22:310.

11. sockeye: Single Cell Transcriptomics [Internet]. Github; [cited 2023 May 5]. Available from: https://github.com/nanoporetech/sockeye

12. Lebrigand K, Magnone V, Barbry P, Waldmann R. High throughput error corrected Nanopore single cell transcriptome sequencing. Nat Commun. 2020;11:4025.

13. Sullivan DK, Pachter L. Flexible parsing and preprocessing of technical sequences with splitcode [Internet]. bioRxiv. 2023 [cited 2023 Jun 8]. p. 2023.03.20.533521. Available from: https://www.biorxiv.org/content/10.1101/2023.03.20.533521v2

14. Dohm JC, Peters P, Stralis-Pavese N, Himmelbauer H. Benchmarking of long-read correction methods. NAR Genom Bioinform. 2020;2:lqaa037.

15. White R, Pellefigues C, Ronchese F, Lamiable O, Eccles D. Investigation of chimeric reads using the MinION. F1000Res. 2017;6:631.

16. Wick RR. Porechop. Github https://github. com/rrwick. Porechop; 2017.

17. Šošić M, Šikić M. Edlib: a C/C++ library for fast, exact sequence alignment using edit distance. Bioinformatics. 2017;33:1394–5.

18. Berger B, Waterman MS, Yu YW. Levenshtein Distance, Sequence Comparison and Biological Database Search. IEEE Trans Inf Theory. 2021;67:3287–94.

19. Chen F, Ding K, Priedigkeit N, Elangovan A, Levine KM, Carleton N, et al. Single-Cell Transcriptomic Heterogeneity in Invasive Ductal and Lobular Breast Cancer Cells. Cancer Res. 2021;81:268–81.

20. Edgren H, Murumagi A, Kangaspeska S, Nicorici D, Hongisto V, Kleivi K, et al. Identification of fusion genes in breast cancer by paired-end RNA-sequencing. Genome Biol. 2011;12:R6.

21. Davidson NM, Majewski IJ, Oshlack A. JAFFA: High sensitivity transcriptome-focused fusion gene detection. Genome Med. 2015;7:43.

22. ugrep. https://github.com/Genivia/ugrep.

23. Shen W, Le S, Li Y, Hu F. SeqKit: A Cross-Platform and Ultrafast Toolkit for FASTA/Q File Manipulation. PLoS One. 2016;11:e0163962.

24. Jabbari JS, Tian L. Massively parallel long-read sequencing of single cell RNA isoforms v1 (protocols.io.8d9hs96) [Internet]. Available from: https://www.protocols.io/view/massively-parallel-long-read-sequencing-of-single-8d9hs96