Copy Number Variation Detection Using Single Cell Sequencing Data

Fatima Zare
Computer Science and Engineering,
University of Connecticut
Storrs, Connecticut
fatemeh.zare@uconn.edu

Jacob Stark
Computer Science and Engineering,
University of Connecticut
Storrs, Connecticut
jacob.a.stark@uconn.edu

Sheida Nabavi*
Computer Science and Engineering,
University of Connecticut
Storrs, Connecticut
sheida.nabavi@uconn.edu

ABSTRACT

Single-cell sequencing (SCS) has emerged as a critical means of discovering important biological knowledge. Data analysis plays an essential role in extracting accurate and meaningful information from SCS data. However, compared to bulk sequencing, SCS introduces new challenges in data analysis. In this paper, we present a novel CNV detection algorithm for SCS data. The proposed method, first, finds the optimal window size for generating read count signal using the AIC approach and removes outliers from the read count signal. Then, using a novel segmentation method based on the Total Variation approach, the method identifies significant change points and detects CNV segments. Finally, it uses the hierarchical clustering of cells based on their CNV patterns and employs Z-score to improve CNV detection across the cells. We used real and simulated data to evaluate the performance of the proposed method and compared its performance with those of other commonly used CNV detection methods. We show that the proposed method outperforms the existing CNV detection methods in terms of sensitivity and false discovery rate.

KEYWORDS

Copy Number Variation, Taut String, Single Cell Sequencing

ACM Reference Format:
Fatima Zare, Jacob Stark, and Sheida Nabavi. 2021. Copy Number Variation Detection Using Single Cell Sequencing Data . In *12th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics (BCB '21), August 1–4, 2021, Gainesville, FL, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3459930.3469556

1 INTRODUCTION

Single-cell sequencing (SCS) has gained rapid popularity as an innovative technique and has been used in many biological and medical studies [8, 25, 28]. Because of the shallow depth of coverage of SCS data, copy number variation (CNV) detection methods based on split-read, paired-end, or single-nucleotide polymorphism density cannot work properly for SCS data and read count based methods are usually used. Read count signals generated from both bulk and SCS are distorted by many factors, such as GC content, mappability, and sequencing errors. The main difference between SCS and traditional bulk sequencing is that SCS requires further steps to isolate a cell and to amplify a single-cell genome [17] that introduces more biases. Due to the small amount of DNA in one cell, an amplification step is performed before sequencing in SCS. Fail to amplify entire segments results in whole-genome amplification biases and significantly distorts read counts [14]. On the other hand, the read coverage in SCS data is significantly lower than that from bulk sequencing data, resulting in much lower signal-to-noise ratios (SNR) in SCS read count signals compared to bulk sequencing read count signals. This low SNR makes CNV detection more challenging. Another challenge in SCS data analysis is inflated read counts because of poorly assembled regions of the genome and low coverage. These biases that are specific to SCS data need to be addressed [2, 6, 14] to extract meaningful and effective information and to make the most of SCS data. As a result new computational methods need to be developed for SCS data [3].

Although many computational tools are available for CNV analysis using bulk sequencing data, there are currently few completely automated and open source CNV detection tools that are designed for SCS data, including SCCNV [10], Ginkgo [14], AneuFinder [4], SCOPE [27], SCICoNE [19], CHISEL [30]. A few CNV detection methods for bulk sequencing data have also been applied/generalized to DNA SCS data [18, 26, 31].

Most of the methods and tools introduced for detecting CNVs from SCS data use circular binary segmentation (CBS) [22] or hidden Markov model (HMM) [12] for segmentation of read count signals. This is mostly because their efficient implementations are available, and they have been commonly used for bulk sequencing data. They have also shown promising results in analyzing SCS data [6, 15, 16]. These tools and methods are different in their preprocessing steps where they generate and normalize read count signals. Among these tools, SCOPE uses a cross-sample segmentation procedure to detect breakpoints that are shared across cells from the same subclones.

In this work, we developed a novel method for more precise and efficient detection of CNVs from SCS data called SCTSCNV. This method consists of three parts: preprocessing, segmentation, and cross-cell postprocessing. In the preprocessing part, first, the proposed method removes GC bias. Then, using the AIC approach, it finds the optimal window size for generating read count signals. Finally, it removes outlier from the read count signals. In the segmentation part, the proposed method detects integer copy numbers using a novel segmentation method. Considering CNVs as

*Corresponding author.

ACM ISBN 978-1-4503-8450-6/21/08…$15.00
https://doi.org/10.1145/3459930.3469556

piece-wise constant signals, we developed a segmentation method by modifying the efficient implementation of the solution to the change-point optimization problem, the Taut String (TS) algorithm [33]. Using the new segmentation algorithm, the proposed method first, detects change points, then it uses the Pettit test to assign significant values to the detected change points. Finally, the method filters out the low confidence change points to reduce false positive CNVs and assigns CNV segments for each individual cell. In the cross-cell postprocessing part, the method employs a hierarchical clustering to cluster cells base on their CNV pattern and uses Z-score method to improve the detection of breakpoints to further improve false positive rate. The codes are available at https://github.com/NabaviLab/ TSCNV.

2 METHOD

The schematic diagram of the overall proposed CNV detection method for DNA SCS data, called SCTSCNV, is shown in Figure 1. The overall workflow of SCTSCNV is divided into preprocessing, segmentation, and cross cell analysis parts. We discus these parts in the following.

2.1 Preprocessing

The preprocessing part of the proposed method include GC bias normalization, optimal window size identification, and outliner read count filtration as described below.

2.1.1 GC bias normalization. After sorting BAM files and deleting duplicated reads using SAMTools [20] and Picard [1], the proposed preprocessing method uses Lowess regression [7] to normalize read coverage values based on the GC contents of the reads. We used deepTool [23], to apply the Lowess regression method introduced in [7] to correct GC bias. The output of deepTool is a GC biased normalized BAM file.

2.1.2 Optimal window size identification. To generate read count signal, the genome needs to be divided into bins (genomic window) after GC correction. To extract maximum genomic information from very low-coverage sequencing ($< 0.1x$), finding the optimal window size for counting mapped reads is critical. If the size of the window is too small, there will be zero counts in many windows, and almost no pattern can be observed. Conversely, the patterns or genomic features would be smoothed out if the window size is too large.

In general, estimating the optimum window size is a model selection problem, where the model parameter is the size of the window, w (or, equivalently, the number of windows in the genome, m). Defining the size of the window, and hence the location of breaks, is critical during constructing a histogram so that the histogram can reflect the true underlying density of the data. We use a data-based method to estimate an optimal window size for generating read count signal proposed in [16]. Given the statistical model, the window size is considered to be optimal if Akaike's information criterion (AIC) is minimized. The AIC model does not need to approximate the underlying density of reads, which can be poorly estimated in sequencing data. AIC is minimum at the optimal window size, i.e., the expected distance of the model to the underlying probability structure that generates the data is minimal. Assume $f(n)$ is the true underlying density of reads in a genome and $n_1, n_2, ..., n_l$ are the genomic position of reads in base pair, which are randomly sampled from a density $f(n)$ and l is the length of the genome. If m denotes the number of windows in the genome, $I_i(n)$ is an i^{th} genomic window for $i = 1, 2, ..., m$, and $t_i(n)$ represents the left-hand point of $I_i(n)$, we can define $w = t_{i+1} - t_i$ to be the width of the window. The width, w, depends on the number of reads in the data. Assuming $v_i(n)$ is the number of reads that fall in the genomic window $I_i(n)$, we can define $f(n)$ as a step function $f(n) = c_i, n \in I_i(n)$. For $c_i \geq 0$ and $\sum_i c_i = 1/w$, the maximum likelihood estimation of the histogram will be the standardized read counts in the i^{th} window: $c_i = v_i(n)/lw$. We calculate AIC with different values of m. The optimal m (number of windows) is estimated when AIC as given in Equation 1 is minimized.

$$AIC = m - \sum_{i=1}^{m} v_i(n) \log\left(\frac{v_i(n)}{lw}\right). \tag{1}$$

2.1.3 Filtering out outliers. After generating read count signals using the optimal window size, we identify and remove bins that have extremely high read count values compared to those of their neighbor bins using the Hampel identifier [21]. The Hampel identifier is robust against outliers and does not require prior knowledge of the data distribution [33].

2.2 Segmentation

A read count signal can be modeled as piece-wise constant (PWC) data that are contaminated by noise as: $\mathbf{r} = \mathbf{f} + \mathbf{N}$,where $\mathbf{N} \sim N(0, \sigma^2)$ is a vector of Gaussian noise, $\mathbf{r}$ is a vector of observed noisy read counts, and $\mathbf{f}$ is a vector of the underlying read counts. The Total Variation (TV) approach is a sparse-regularized optimization that has shown outstanding performances in estimating change points of a PWC signal [32, 33]. Given observed signal $\mathbf{r}$ the TV denoising approach tries to compute the least square estimates for some regularization parameter $\varepsilon > 0$ to estimate the desired signal $\mathbf{f}$ as shown in Equation 2.

$$\min_{\mathbf{f} \in \mathbb{R}^n} \frac{1}{2}\|\mathbf{r} - \mathbf{f}\|_2^2 + \varepsilon\|\mathbf{Df}\|_1, \tag{2}$$

where $\mathbf{D}$ is the differencing matrix (all zeros except $D_{ii} = -1$ and $D_{i,i+1} = 1 (1 \leq i \leq n-1)$). By changing and defining a new set of variables [5], we can write Equation 3 instead of Equation 2:

$$\min_{\mathbf{F}} \sum_{i=1}^{n} \sqrt{1 + (F_{i-1} - F_i)^2}, s.t. \ \|\mathbf{F} - \mathbf{R}\|_\infty \leq \varepsilon, \tag{3}$$

where $R_i = \sum_{k=1}^{i} r_k$ and $F_i = \sum_{k=1}^{i} f_k$ are the cumulative sum of signal $\mathbf{r}$ and $\mathbf{f}$, respectively. Once $\mathbf{F}$ is found, it is possible to recover the original read count signal $\mathbf{f}$ by observing that:

$$F_i - F_{i-1} = R_i - u_i - (R_{i-1} - u_{i-1}) = r_i - u_i + u_{i-1} = f_i. \tag{4}$$

According to $F_i = \sum_{k=1}^{i} f_k$, f_i is the slope of $\mathbf{F}$ between points i and $i+1$. Therefore, f_i represents the derivative of $\mathbf{F}$ between points i and $i+1$. For a noisy signal with length l and standard deviation σ, ε can be chosen as $\sigma\sqrt{2\log l}$ [9, 11]. The Taut String algorithm has been proposed to solve this optimization problem efficiently with a complexity of $O(n)$. Taut String generates upper and lower

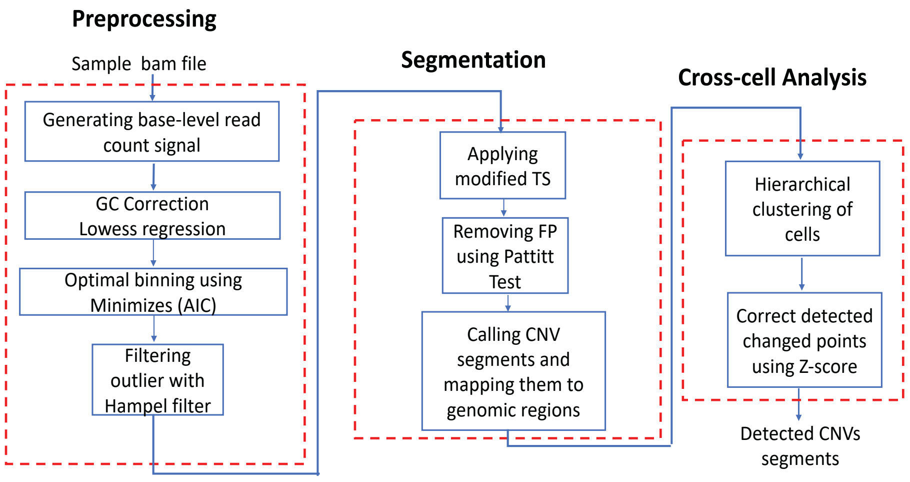
Preprocessing
Sample bam file
Generating base-level read count signal
GC Correction Lowess regression
Optimal binning using Minimizes (AIC)
Filtering outlier with Hampel filter
Segmentation
Applying modified TS
Removing FP using Pattitt Test
Calling CNV segments and mapping them to genomic regions
Cross-cell Analysis
Hierarchical clustering of cells
Correct detected changed points using Z-score
Detected CNVs segments

Figure 1: The overall workflow of SCTSCNV

bounds of the observed noisy data, $R + \varepsilon$ and $R - \varepsilon$ surrounding R as a tube. This approach starts from the fixed point of $F_0 = 0$ and gradually calculates the greatest convex minorant of the upper bound and the smallest concave majorant of the lower bound on the tube. When both curves (the greatest convex minorant and smallest concave majorant) intersect, the first segment of Taut String is fixed using the left-most point where either the majorant or the minorant touched the tube. From the end of the identified segment, the procedure is then resumed and iterated until all segments are identified.

The conventional Taut String algorithm has two main challenges: staircase effect at change points [24] and choosing an effective regularization parameter ε. It is shown that the first change point detected by Taut String does not show the staircase effect. Also, it is shown that an appropriate ε depends on the variance of the data. Build up on our previous modified Taut String algorithm for addressing these challenges [33], we developed an algorithm to also yield integer CNV values and assign significant values to the change points. As shown in Algorithm 1, we modified Taut String algorithm such that after detecting the first change point, our algorithm makes Taut String to stop, stores the first change point and resets Taut String again with the detected change point as a new start point. It means that after each pause, Taut String is applied to the rest of the signal with a new ε based on the new length l and standard deviation of the signal. In fact, after each pause, the algorithm calculates a new ε adaptively based on the rest of the signal. The algorithm repeats this procedure until the whole signal is scanned and all the potential change points are detected. However, some of the change points are not significant enough to be considered as CNV segment breakpoints. In this step, the algorithm calculates the detected change points' *p-value*s using the Pettitt test to identify non-significant change points for filtering them out. The Pettitt test does not require any assumption on the distribution of data and provides a *p-value* to test the significance of a detected change point. Finally, the algorithm calculates the means of signal $\mathbf{r}$ between each pair of detected change points, and round it to the nearest integer to assign copy number to the detected segments.

2.3 Cross-cell analysis

We map back the CNV segments to genomic positions and generate a matrix in which rows are cells, and columns are genomic positions of CNV breakpoints. To find clusters of cells that share the same CNV patterns we apply the hierarchical clustering method to the matrix. For the hierarchical clustering we use Ward's minimum variance to find clusters of cells based on their ploidy. We assume cells from the same cluster of ploidy share similar genomic locations of breakpoints. For each identified cluster, we consider its own CNV pattern matrix. The goal is to identify and adjust outlier breakpoints based on the breakpoints of other cells in the same cluster. The Z-score objective is to explain any data points by figuring out how they contribute to the standard deviation and mean of the group of data points. We calculate Z-score for a breakpoint of a cell i at location j as: $(Z_{ij} = (x_{ij} - \mu)/\sigma)$, where x_{ij} is a breakpoint of cell i in location j, and μ and σ are the mean and standard deviation of all breakpoints in the cluster containing cell i. At genomic location j across the corresponding cluster, we look for outlier breakpoints that have a Z_{ij} larger than a threshold. Then we replace the outlier breakpoints with the average of non-outlier breakpoints.

3 DATA SETS

To investigate the performance of the proposed method, we used three sets of data: 1) simulated read count signals, 2) simulated sequencing data, and 3) real sequencing data.

3.1 Simulated read count signal

Inspired by the CNV patterns of real data we generated 390 simulated read count signals (Cells), representing CNV segments with no noise, with 6 deterrent average ploidies as gold standard CNV data to evaluate the performance of our model. Using these data, we generated noisy read count signals by adding different levels

Algorithm 1 SCSegmentation

```
1:  function RSCSegmentation(r)
2:      $CNVsegments \leftarrow \emptyset$
3:      $R \leftarrow Cumsum(r)$
4:      $desiredF \leftarrow \emptyset$
5:      while $length(R) > 0$ do
6:          $l \leftarrow length(R)$, $\sigma \leftarrow std(R)$, $\epsilon \leftarrow \sigma \times \sqrt{2 \times \log(l)}$
7:          Initialization
8:          while $i < l$, $l$ is the length of noisy signal do
9:              $majorantheight \leftarrow \varepsilon$
10:             if $upperbound > \varepsilon$ then
11:                 Build valid segment up to last majorant breaking point
12:                 Start new segment after break
13:                 Break
14:             end if
15:             Update minorant height
16:             if $underbound < -\varepsilon$ then
17:                 Build valid segment up to last minorant breaking point
18:                 Start new segment after break
19:                 Break
20:             end if
21:             if $upperbound \leq -\varepsilon$ then
22:                 The majorant now touches the floor
23:             end if
24:             if $underbound \geq \varepsilon$ then
25:                 The minorant now touches the ceiling
26:             end if
27:             $i = i + 1$
28:         end while
29:         build last valid segment
30:     end while
31:     $f \leftarrow Slope(desiredF)$
32:     $ChangePoints \leftarrow NonZeroIndicator(Dx(f))$
33:     $ChangePoin.Pval \leftarrow PettitTest(ChangePoints, r)$
34:     $FinalCP \leftarrow ChangePoin.Pval < thr$
35:     while $k < length(FinalCP)$ do
36:         $CNVsegments[k] \leftarrow Round(mean(f_{k+1}, f_k)$
37:     end while
38:     return CNVsegments
39: end function
```

of Gaussian white noise to each of the simulated read count signal. We used a range of signal to noise ratios (SNRs) from 1 to 10, where SNR is defined as the ratio of the signal power (meaningful information) to the background noise power (unwanted signal).

3.2 Simulated sequencing data

We used CNSC [29], a single-cell CNV simulator to simulates sequencing data with different ploidies. Using the simulator we generated 20 synthetic data sets with different ploidies of 1.55, 2.1, 3.3.8, 5.26. We benchmarked three CNV detection methods for DNA SCS data: Our method (SCTSCNV), Ginkgo, and HMMcopy by using these data.

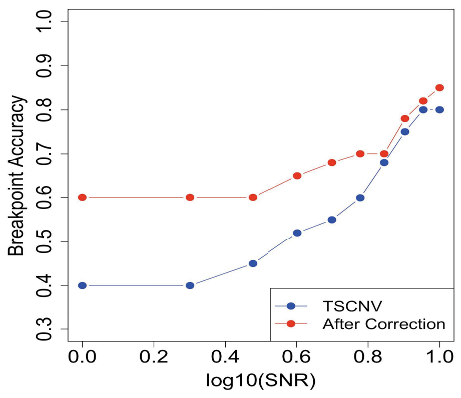

Figure 2: Breakpoint accuracy before and after applying cross-analysis for different level of noise, σ_N^2

Table 1: Overall Performance of the Proposed CNV detection Method Using the Real SCS data

Methods	Sensitivity	FDR
Our proposed method	53.14%	41.20%
HMMcopy	51.45%	53.52%
Ginkgo	55.2%	55.23%

3.3 Real data

We downloaded 67 cells generated by the study in [13]. Sequencing data are available at SRA under accession code SRP188831. The study also provides a CNV list of the data using the baseqCNV [13] tool. Due to the lack of ground truth, we evaluated the performance of our method using the provided CNV list. We also used another real biological data set that is available at SRA under accession code SRP114962. We do not have a CNV list as a benchmark for this dataset. We compared the detected amplified and deleted genes by HMMcopy, Ginkgo and our developed method.

4 RESULTS

4.1 Results on simulated read count signal

Using the simulated data, we evaluated the performance of the proposed cross-cell analysis approach. The simulated data consists of 6 clusters, and each cluster has cells with the same ploidy and breakpoint locations. We applied the postprocessing part of the proposed methods to the detected CNVs from the simulated read count signal. Figure 2 shows the performance of our method in detecting breakpoints. Breakpoints of each detected segment indicate the start and end position of the segment in the genome. In this analysis, breakpoint accuracy is defined as the percentage of the number of times the start and end points of detected CNV segments are exactly the same as those of the known CNV segments from the gold standard. As can be seen in Figure 2, using the cross-cell analysis approach increases the breakpoint accuracy. It can be seen that our proposed cross-cell model improves the breakpoint detection accuracy significantly especially when the level of noise is high.

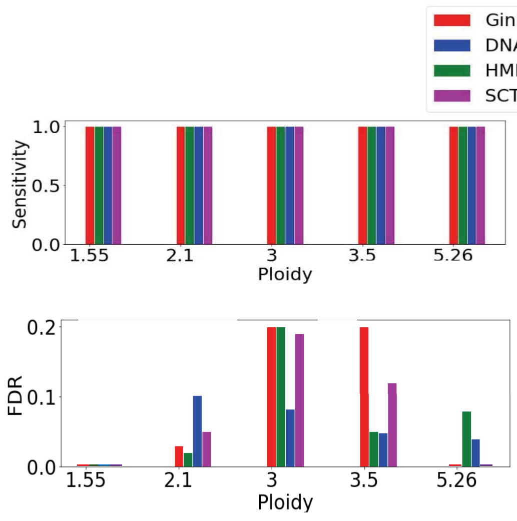

Figure 3: The performance of CNV detection tools using simulated sequencing data with different ploidies.

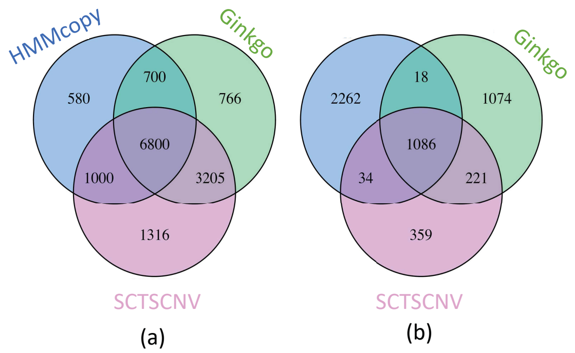

Figure 4: Venn diagrams of the average of the number of truly detected CNV genes from 3 tools: SCTSCNV, Ginko and HMMCopy, (a) amplified genes, (b) deleted genes.

4.2 Results on the simulated sequencing data

Using simulated sequencing data, we also compared the performance of our proposed method with those of three popular CNV detection tools: Ginkgo, HMMcopy and DNAcopy using simulated data with different ploidy. We annotated the detected CNV segments to obtain gene lists. We used the "cghMCR" R package from Bioconductor (http://bioconductor.org/packages/cghMCR/) to identify CNV genes using Refseq gene identifications. The average of sensitivities, and FDRs are shown in Figure 3, when comparing CNV genes. The data set contains 5 clusters with different ploidies. As can be seen in Figure 3, all the methods provide high sensitivities. The proposed model provide lower FDRs for most of the ploidies compared to the other methods.

4.3 Results on the real data

We compared the performance of our proposed method with those of the two popular SCS CNV detection tools, Ginko and HMMCopy using the real sequencing data (67 cells) explained in the Data Sets Section. The average sensitivity and FDR of the CNV detection tools on real SCS data are shown in Table 1. Our proposed method, SCTSCNV, outperforms HMMcopy and Ginkgo in detecting CNVs. Ginkgo uses a variable size approach for binning, and HMMcopy uses a fixed size to generate read count signal. These results show that performing an optimal window size approach can lead to a better performance of SCS CNV detection. Another reason for the better performance of our approach is the use of statistical tests to identify outliers and delete or modify them: the preprocessing part contains a very efficient outlier bin removal approach, the segmentation part identifies non-significant breakpoints and removes them, and the postprocessing part uses Z-score to identify outlier breakpoints and modifies them.

Also, using the other data set with 371 cells explained in the the Data Sets section, we compared amplified and deleted genes identified by SCTSCNV, Ginko and HMMCopy. Using a Venn diagram as shown in Figure 4, we observed that the proposed method has the largest shares of amplified and deleted genes compared to Ginko and HMMCopy. It means that our method can identify more true CNV genes, because compared to the other tools, more of its detected CNV genes are also detected by the other tools. Nevertheless, as also reported by studies on CNV tools for bulk sequencing data, the consistency among tools in calling CNV genes using SCS data is not high.

5 CONCLUSION

Single-cell DNA sequencing has emerged as a popular and powerful tool to assess genomic heterogeneity. This technology is widely used in stem cell, neuron, and cancer studies. One fundamental concern in analysing SCS data is the existence of high level of noise and biases in the data. Numerous technological errors can be introduced during the three major stages of cell isolation, genome amplification, and sequencing. Also, the lack of coverage and the extremely non-uniform genome-wide read counts is often observed in raw SCS coverage data. In this work, we developed a comprehensive pipeline to identify CNVs using SCS data. Our proposed method consists of preprocessing, segmentation, and postprocessing parts. For the preprocessing part, we developed a method to estimate the optimal window size for counting reads by minimizing AIC across different window sizes. For the segmentation part, we developed a method based on the Taut String algorithm to detect CNV segments and to assign p-values to change points based on the Pettitt test. For the postprocessing part of our method, we clustered cells using a cross-cell analysis based on Z-score to improve CNV detection. We showed that using advanced segmentation methods and applying pre/postprocessing analysis such as using optimal window size for generating read count signals, filtering outlier read counts, filtering low significant breakpoints, adjusting outlier breakpoints using breakpoints of similar cells, can help improving CNV detection for SCS data. We conclude that CNV detection in SCS data is still in its early stage. Single Cell DNA sequencing data are being more prevalent and interest in analysing CNV using SCS data is increasing. As a result, because SCS technology introduces unique challenges in data analysis and unlike bulk sequencing, SCS data sets contain many cells (thousands of cells), developing new approaches to address the new challenges and take advantage of information across cells to identify CNVs is necessary.

ACKNOWLEDGMENTS

This study was supported by the National Institutes of Health (NIH) grant No. R00LM011595, PI: Nabavi and the National Science Foundation (NSF) under grant No. 1942303, PI: Nabavi.

REFERENCES

[1] [n.d.]. Picard Tools - By Broad Institute. http://broadinstitute.github.io/picard/

[2] Can Alkan, Bradley P. Coe, and Evan E. Eichler. 2011. Genome structural variation discovery and genotyping. *Nature Reviews Genetics* 12, 5 (May 2011), 363–376. https://doi.org/10.1038/nrg2958

[3] Philipp Angerer, Lukas Simon, Sophie Tritschler, F Alexander Wolf, David Fischer, and Fabian J Theis. 2017. Single cells make big data: New challenges and opportunities in transcriptomics. *Current Opinion in Systems Biology* 4 (2017), 85–91.

[4] Bjorn Bakker, Aaron Taudt, Mirjam E. Belderbos, David Porubsky, Diana C. J. Spierings, Tristan V. de Jong, Nancy Halsema, Hinke G. Kazemier, Karina Hoekstra-Wakker, Allan Bradley, Eveline S. J. M. de Bont, Anke van den Berg, Victor Guryev, Peter M. Lansdorp, Maria Colomé-Tatché, and Floris Foijer. 2016. Single-cell sequencing reveals karyotype heterogeneity in murine and human malignancies. *Genome Biology* 17, 1 (Dec. 2016), 115. https://doi.org/10.1186/s13059-016-0971-7

[5] Álvaro Barbero and Suvrit Sra. 2014. Modular proximal optimization for multidimensional total-variation regularization. *arXiv:1411.0589 [math, stat]* (Nov. 2014). http://arxiv.org/abs/1411.0589 arXiv: 1411.0589.

[6] Timour Baslan, Jude Kendall, Linda Rodgers, Hilary Cox, Mike Riggs, Asya Stepansky, Jennifer Troge, Kandasamy Ravi, Diane Esposito, B Lakshmi, Michael Wigler, Nicholas Navin, and James Hicks. 2012. Genome-wide copy number analysis of single cells. *Nature Protocols* 7, 6 (June 2012), 1024–1041. https://doi.org/10.1038/nprot.2012.039

[7] Daniel J Benjamin, David Cesarini, Matthijs JHM van der Loos, Christopher T Dawes, Philipp D Koellinger, Patrik KE Magnusson, Christopher F Chabris, Dalton Conley, David Laibson, Magnus Johannesson, and others. 2012. The genetic architecture of economic and political preferences. *Proceedings of the National Academy of Sciences* 109, 21 (2012), 8026–8031.

[8] Hao Chen, John M. Bell, Nicolas A. Zavala, Hanlee P. Ji, and Nancy R. Zhang. 2015. Allele-specific copy number profiling by next-generation DNA sequencing. *Nucleic Acids Research* 43, 4 (Feb. 2015), e23–e23. https://doi.org/10.1093/nar/gku1252

[9] Haeran Cho and Piotr Fryzlewicz. 2011. Multiscale interpretation of taut string estimation and its connection to Unbalanced Haar wavelets. *Statistics and Computing* 21, 4 (Oct. 2011), 671–681. https://doi.org/10.1007/s11222-010-9200-5

[10] Xiao Dong, Lei Zhang, Xiaoxiao Hao, Tao Wang, and Jan Vijg. 2019. *SCCNV: a software tool for identifying copy number variation from single-cell whole-genome sequencing*. preprint. Bioinformatics. https://doi.org/10.1101/535807

[11] Lutz Dümbgen, Arne Kovac, et al. 2009. Extensions of smoothing via taut strings. *Electronic Journal of Statistics* 3 (2009), 41–75.

[12] Jane Fridlyand, Antoine M. Snijders, Dan Pinkel, Donna G. Albertson, and Ajay N. Jain. 2004. Hidden Markov models approach to the analysis of array CGH data. *Journal of Multivariate Analysis* 90, 1 (July 2004), 132–153. https://doi.org/10.1016/j.jmva.2004.02.008

[13] Yusi Fu, Fangli Zhang, Xiannian Zhang, Junlong Yin, Meijie Du, Mengcheng Jiang, Lu Liu, Jie Li, Yanyi Huang, and Jianbin Wang. 2019. High-throughput single-cell whole-genome amplification through centrifugal emulsification and eMDA. *Communications Biology* 2, 1 (Dec. 2019), 147. https://doi.org/10.1038/s42003-019-0401-y

[14] Tyler Garvin, Robert Aboukhalil, Jude Kendall, Timour Baslan, Gurinder S Atwal, James Hicks, Michael Wigler, and Michael C Schatz. 2015. Interactive analysis and assessment of single-cell copy-number variations. *Nature Methods* 12, 11 (Nov. 2015), 1058–1060. https://doi.org/10.1038/nmeth.3578

[15] Charles Gawad, Winston Koh, and Stephen R Quake. 2016. Single-cell genome sequencing: current state of the science. *Nature Reviews Genetics* 17, 3 (2016), 175.

[16] Arief Gusnanto, Charles C. Taylor, Ibrahim Nafisah, Henry M. Wood, Pamela Rabbitts, and Stefano Berri. 2014. Estimating optimal window size for analysis of low-coverage next-generation sequence data. *Bioinformatics* 30, 13 (July 2014), 1823–1829. https://doi.org/10.1093/bioinformatics/btu123

[17] Byungjin Hwang, Ji Hyun Lee, and Duhee Bang. 2018. Single-cell RNA sequencing technologies and bioinformatics pipelines. *Experimental & Molecular Medicine* 50, 8 (Aug. 2018), 96. https://doi.org/10.1038/s12276-018-0071-8

[18] Kristin A. Knouse, Jie Wu, and Angelika Amon. 2016. Assessment of megabase-scale somatic copy number variation using single-cell sequencing. *Genome Research* 26, 3 (March 2016), 376–384. https://doi.org/10.1101/gr.198937.115

[19] Jack Kuipers, Mustafa Anıl Tuncel, Pedro Ferreira, Katharina Jahn, and Niko Beerenwinkel. 2020. *Single-cell copy number calling and event history reconstruction*. preprint. Cancer Biology. https://doi.org/10.1101/2020.04.28.065755

[20] Heng Li, Bob Handsaker, Alec Wysoker, Tim Fennell, Jue Ruan, Nils Homer, Gabor Marth, Goncalo Abecasis, Richard Durbin, and others. 2009. The sequence alignment/map format and SAMtools. *Bioinformatics* 25, 16 (2009), 2078–2079.

[21] Hancong Liu, Sirish Shah, and Wei Jiang. 2004. On-line outlier detection and data cleaning. *Computers & Chemical Engineering* 28, 9 (Aug. 2004), 1635–1647. https://doi.org/10.1016/j.compchemeng.2004.01.009

[22] Adam B Olshen, ES Venkatraman, Robert Lucito, and Michael Wigler. 2004. Circular binary segmentation for the analysis of array-based DNA copy number data. *Biostatistics* 5, 4 (2004), 557–572.

[23] Fidel Ramírez, Devon P Ryan, Björn Grüning, Vivek Bhardwaj, Fabian Kilpert, Andreas S Richter, Steffen Heyne, Friederike Dündar, and Thomas Manke. 2016. deepTools2: a next generation web server for deep-sequencing data analysis. *Nucleic Acids Research* 44, W1 (July 2016), W160–W165. https://doi.org/10.1093/nar/gkw257

[24] Cristian R. Rojas and Bo Wahlberg. 2015. How to monitor and mitigate stair-casing in L1 trend filtering. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, South Brisbane, Queensland, Australia, 3946–3950. https://doi.org/10.1109/ICASSP.2015.7178711

[25] Jeremy J. Shen and Nancy R. Zhang. 2012. Change-point model on nonhomogeneous Poisson processes with application in copy number profiling by next-generation DNA sequencing. *The Annals of Applied Statistics* 6, 2 (June 2012), 476–496. https://doi.org/10.1214/11-AOAS517

[26] Sarah A Vitak, Kristof A Torkenczy, Jimi L Rosenkrantz, Andrew J Fields, Lena Christiansen, Melissa H Wong, Lucia Carbone, Frank J Steemers, and Andrew Adey. 2017. Sequencing thousands of single-cell genomes with combinatorial indexing. *Nature Methods* 14, 3 (March 2017), 302–308. https://doi.org/10.1038/nmeth.4154

[27] Rujin Wang, Dan-Yu Lin, and Yuchao Jiang. 2019. *SCOPE: a normalization and copy number estimation method for single-cell DNA sequencing*. preprint. Bioinformatics. https://doi.org/10.1101/594267

[28] Xuefeng Wang, Hao Chen, and Nancy R. Zhang. 2018. DNA copy number profiling using single-cell sequencing. *Briefings in Bioinformatics* 19, 5 (2018), 731–736. https://doi.org/10.1093/bib/bbx004

[29] Nickolas Navin Luay Nakhleh Xian Fan, Mohammadamin Edrisi. 2019. CNSC. https://bitbucket.org/xianfan/cnsc_simulator/src/master/

[30] Simone Zaccaria and Benjamin J. Raphael. 2020. Characterizing allele- and haplotype-specific copy numbers in single cells with CHISEL. *Nature Biotechnology* (Sept. 2020). https://doi.org/10.1038/s41587-020-0661-6

[31] Hans Zahn, Adi Steif, Emma Laks, Peter Eirew, Michael VanInsberghe, Sohrab P Shah, Samuel Aparicio, and Carl L Hansen. 2017. Scalable whole-genome single-cell library preparation without preamplification. *Nature Methods* 14, 2 (Feb. 2017), 167–173. https://doi.org/10.1038/nmeth.4140

[32] Fatima Zare, Sardar Ansari, Kayvan Najarian, and Sheida Nabavi. 2018. Copy number variation detection using partial alignment information. In *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, Madrid, Spain, 2435–2441. https://doi.org/10.1109/BIBM.2018.8621529

[33] Fatima Zare and Sheida Nabavi. 2019. Copy Number Variation Detection Using Total Variation. In *Proceedings of the 10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*. ACM, Niagara Falls NY USA, 423–428. https://doi.org/10.1145/3307339.3342181