Bioinformatics
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: Day Month Year
Manuscript Category

Genomics

User-guided local and global copy-number segmentation for tumor sequencing data

Zubair Lalani [1,†], Gillian Chu [1,†], Silas Hsu [1], Simone Zaccaria [2,3,*] and Mohammed El-Kebir [1,4,*]

[1]Department of Computer Science, University of Illinois Urbana-Champaign, Urbana, IL, USA
[2]Computational Cancer Genomics Research Group, University College London Cancer Institute, London, UK
[3]Cancer Research UK Lung Cancer Centre of Excellence, University College London Cancer Institute, London, UK
[4]Cancer Center at Illinois, University of Illinois Urbana-Champaign, Urbana, IL, USA

[†]Joint first authorship.
[*]To whom correspondence should be addressed.

Abstract

Motivation: Copy-number aberrations (CNA) are genetic alterations that amplify or delete the number of copies of large genomic segments. Although they are ubiquitous in cancer and subsequently a critical area of current cancer research, CNA identification from DNA sequencing data is challenging because it requires partitioning of the genome into complex segments that may not be contiguous. Existing segmentation algorithms address these challenges either by leveraging the local information among neighboring genomic regions, or by globally grouping genomic regions that are affected by similar CNAs across the entire genome. However, both approaches have limitations: overclustering in the case of local segmentation, or the omission of clusters corresponding to focal CNAs in the case of global segmentation. Importantly, inaccurate segmentation will lead to inaccurate identification of important CNAs.
Results: We introduce CNAViz, a web-based tool that enables the user to simultaneously perform local and global segmentation, thus overcoming the limitations of each approach. Using simulated data, we demonstrate that by several metrics, CNAViz yields more accurate segmentations relative to existing local and global segmentation methods. Moreover, we analyze six bulk DNA sequencing samples from three breast cancer patients. By validating with parallel single-cell DNA sequencing data from the same samples, we show that CNAViz's more accurate segmentation improves accuracy in downstream copy-number calling.
Availability and implementation: `https://github.com/elkebir-group/cnaviz`
Contact: s.zaccaria@ucl.ac.uk, melkebir@illinois.edu

1 Introduction

The cancer genome of most solid tumors is characterized by the accumulation of somatic genetic alterations, called *copy-number aberrations (CNAs)*, which are pervasive across different cancer types with on average 44% of the genome being affected by CNAs in solid tumors (Watkins *et al.*, 2020; Dentro *et al.*, 2021; The PCAWG Consortium *et al.*, 2020). While two distinct copies, or *alleles*, are expected to be present in the genome of normal diploid cells for every gene in autosomal chromosomes, each CNA can simultaneously alter the dosage of hundreds to thousands of genes by increasing (gain) or decreasing (loss) the number of copies of a large genomic segment, including chromosome's arms and whole chromosomes (Zack *et al.*, 2013; Beroukhim *et al.*, 2010). Therefore, the identification of CNAs has a critical impact on the understanding of cancer evolution (McGranahan and Swanton, 2015; Jamal-Hanjani *et al.*, 2017; Bielski *et al.*, 2018; Watkins *et al.*, 2020). Moreover, the identification of CNAs may inform the development of targeted therapies since CNAs can introduce novel vulnerabilities for cancer cells that can be exploited for drug design (Cohen-Sharir *et al.*, 2021; Quinton *et al.*, 2021; Memon *et al.*, 2021).

Currently, most cancer studies characterize the presence of CNAs in large cohorts of cancer patients by performing DNA sequencing of one

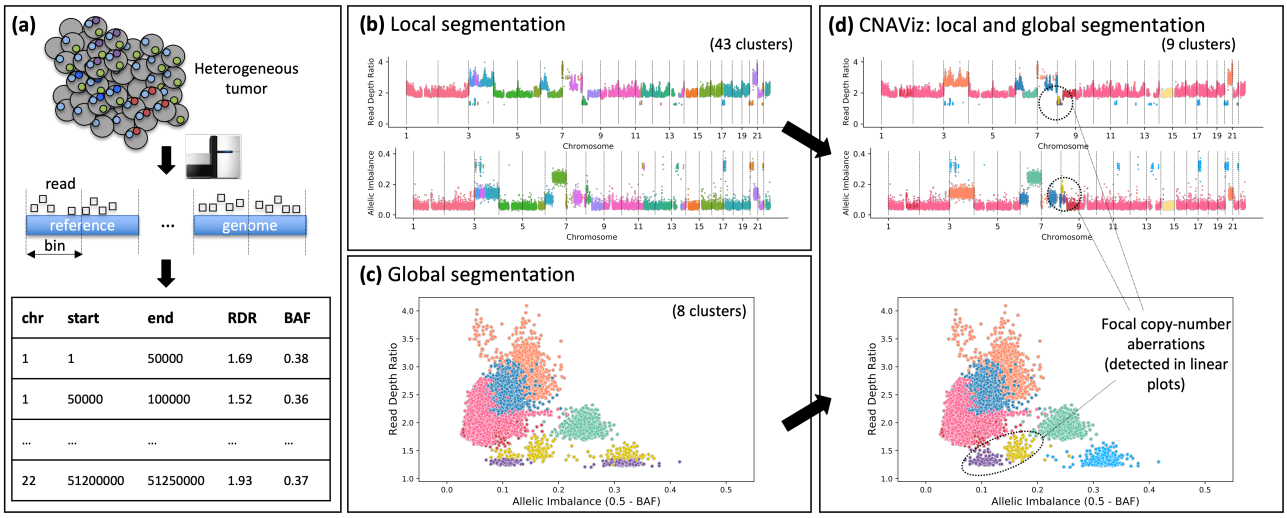


Fig. 1: **CNAViz enables user-guided segmentation for improved copy-number calling.** (a) The genome of cancer cells (gray circles) is affected by CNAs (colored dots). DNA sequencing reads obtained from these cancer cells are aligned to a human reference genome, which is partitioned into bins (defined by the start and end position of the bin in a certain chromosome). For each bin, two signals are measured from DNA sequencing reads: the RDR, which is proportional to the total number of copies of the bin in the genome, and the BAF, which measures allelic imbalance. (b) Local segmentation algorithms combine neighboring bins with identical RDR (top plot) and BAF (bottom plot, where allelic imbalance is represented instead of BAF and is measured as $0.5 - \text{BAF}$) into segments. Differences across datasets might lead to overclustering. (c) Global segmentation algorithms cluster bins with similar RDR and BAF values across the entire genome, disregarding genomic location information, which may lead to spurious clusters and omit focal CNAs. (d) CNAViz allows the user to unify local and global segmentation approaches to obtain a more accurate segmentation.

or multiple tumor samples (Jamal-Hanjani *et al.*, 2017; Watkins *et al.*, 2020; The PCAWG Consortium *et al.*, 2020). In particular, CNAs can be identified from DNA sequencing data by combining two related signals that are observed for every genomic region, or *bin* (Fig. 1a) — i.e. a sequence of neighboring genomic loci (Tarabichi *et al.*, 2021). First, the *read depth ratio (RDR)* is defined as the ratio between the observed and expected number of sequencing reads that align to a specific bin. As such, variations of RDRs indicate changes in the total number of copies: an increase/decrease in the values of RDR between different bins indicates a higher/lower number of copies. For example, while RDRs are expected to be nearly constant in normal diploid cells, higher/lower values of RDRs across the cancer genome allow the identification of related gains/losses due to CNAs. Second, the *B-allele frequency (BAF)* is defined as the proportion of sequencing reads that belong to only one of the two alleles of the bin. A value of 0.5 is expected for normal diploid bins since each allele is present in one copy and half of the sequencing reads are expected to be sequenced from each allele. As such, a significant deviation from this expected value, called *allelic imbalance*, indicate the presence of CNAs that affect the proportion of copies between the two alleles. For example, if BAF is observed to be 0.33 for a bin that is affected by a gain and has three copies (as indicated by the RDR), we can conclude that the genome contains two copies of one allele and one copy of the other; in contrast, a BAF of 0.0 would indicate that the genome contains three copies of only one allele. Thus, analyzing variations of RDR and BAF values across bins allow the identification of CNAs in cancer genomes. However, this is a challenging task for which several algorithms have been proposed.

So far, most of the proposed algorithms to identify CNAs from variations in RDRs and BAFs are based on *local segmentation* approaches. The key idea is that CNAs generally affect large genomic segments that comprise multiple bins and, therefore, neighboring bins have an increased probability to be or not be affected by the same CNA. As such, algorithms for change-point detection (e.g., Hidden Markov models) have been proposed to identify CNA-based genomic segments by grouping neighboring bins that do not have higher than expected variations in RDRs and BAFs (Fig. 1b). Examples of these algorithms for DNA sequencing data include ASCAT (Van Loo *et al.*, 2010; Ross *et al.*, 2021), BIC-seq (Xi *et al.*, 2011), Control-FREEC (Boeva *et al.*, 2012), TITAN (Ha *et al.*, 2014) for bulk tumor samples, as well as HMMcopy (Laks *et al.*, 2019) and Ginkgo (Garvin *et al.*, 2015) for single cells. However, the performance of local-segmentation algorithms can be substantially affected in different sequencing datasets by the presence of decreased or increased variance of RDR and BAF values between or within distinct genomic segments. While decreased variance is due to normal contamination, i.e. the presence of normal, non-cancerous cells in the sample (Van Loo *et al.*, 2010; Watkins *et al.*, 2020; Zaccaria and Raphael, 2020), increased variance results from differences in sequencing technologies and platforms (Zare *et al.*, 2017; Zaccaria and Raphael, 2021).

To deal with the limitations of local segmentation, *global segmentation* approaches have been proposed, which leverage the presence of distinct genomic segments affected by similar CNAs. In fact, similar CNAs are frequent across the entire genome of the same tumor, resulting in bins from across the genome with similar RDR and BAF values. Thus, global-segmentation algorithms, such as FACETS (Shen and Seshan, 2016) and CELLULOID (Notta *et al.*, 2016), leverage these shared signals from different CNAs by clustering bins that share RDR and BAF values (Fig. 1c). Moreover, the recent HATCHet (Zaccaria and Raphael, 2020) and CHISEL (Zaccaria and Raphael, 2021) algorithms have demonstrated that this global approach can be further extended to jointly leverage the signals even across multiple samples (or single cells) obtained from the same tumor, obtaining improved power to accurately identify CNAs even in contexts of low tumors purity or CNAs that are only present in distinct subpopulations of cancer cells. However, this increased power afforded by global segmentation comes at the cost of a diminished ability to identify smaller or focal CNAs, as well as CNAs that are only present in few or single tumor samples, which are frequent in cancer (Zaccaria and Raphael, 2020). Since local-segmentation algorithms generally have

improved power for these smaller and focal CNAs by leveraging the local signals of neighboring genomic regions, there is thus a trade-off between local and global segmentation approaches.

Importantly, accurate segmentation is key to identifying tumor clones with accurate copy number calls. To solve the trade-off between local and global segmentation algorithms, we introduce CNAViz, a graphical, interactive, and web-based tool to perform user-guided segmentation of tumor DNA sequencing data for the identification of CNAs (Fig. 1d). By providing an accessible and highly portable interactive platform to combine RDR and BAF values across both the entire genome and multiple samples while simultaneously revealing the presence of local genomic patterns, CNAViz represents a unifying approach that combines the advantages of local and global segmentation approaches. In particular, CNAViz is applicable to a wide range of novel and retrospective analyses, as it can be used to perform both segmentation *de novo* or to improve the segmentation performed by other existing segmentation methods. We have used simulated multi-sample tumor sequencing dataset generated by the published MASCoTE framework (Zaccaria and Raphael, 2020) to demonstrate the improved accuracy of CNAViz relative to existing local and global segmentation methods. Moreover, we have applied CNAViz to previous bulk DNA sequencing data generated from 6 tumor samples obtained from 3 breast cancer patient (Casasent *et al.*, 2018). Using these data, we have demonstrated that CNAViz is more concordant with parallel single-cell sequencing data of these samples, revealing the presence of CNAs for known breast cancer driver genes that would have been missed by current methods.

2 Requirements

We describe the input data in Section 2.1. We discuss the analysis tasks required for effective segmentation of copy-number data in Section 2.2.

2.1 Data Characteristics

Input and Output. CNAViz input data adhere to seven characteristics.

(D1) One or more samples, quantified by $m > 0$, are sequenced from a tumor. Samples may correspond to bulk DNA sequencing samples and/or single-cell DNA sequencing samples.

(D2) The genome is partitioned in n bins that may vary in size. We indicate the chromosome in which bin i occurs by $\mathrm{chr}(i)$, its start position on that chromosome by $\mathrm{start}(i)$ and end position by $\mathrm{end}(i)$.

(D3) The read depth ratio $\mathrm{RDR}(p, i)$ is provided for each bin i in each sample p.

(D4) The B-allele frequency $\mathrm{BAF}(p, i)$ is provided for each bin i in each sample p.

(D5) Optionally, each bin may be assigned to a segment/cluster $\mathrm{cluster}(i)$. These values represent the local or global segmentation performed by existing algorithms and is used for further refinement.

(D6) Optionally, a set D of driver genes may be provided with genomic coordinates $(\mathrm{chr}(d), \mathrm{start}(d), \mathrm{end}(d))$ for each driver gene d.

(D7) Export new clustering. The new clustering created is exportable for future usage. Bins that have been erased (C5) are excluded from export.

2.2 Analysis Tasks

CNAViz is characterized by five groups of analysis tasks: (i) plotting, (ii) filtering, (iii) selecting, (iv) clustering, and (v) analytics. The overarching requirement underlying these tasks is that the tool should be able to support both local and global segmentation.

Plotting. We begin by introducing the requirements regarding plotting.

(P1) Plot the RDR and BAF values of bins for each sample sorted by genomic coordinates. To facilitate local segmentation, the user can inspect RDR and BAF values ordered by genomic coordinates. This will enable the user to identify breakpoints along the genome.

(P2) Simultaneously plot RDR and BAF values of bins for each sample. To facilitate global segmentation, the user can inspect RDR and BAF values in a two-dimensional plot for each sample. This will enable the user to identify groups of bins distributed across the genome with similar RDR and BAF values across all samples.

(P3) Indicate clustering of bins with the same set of colors in all plots. To enable the user to view the current clustering, the tool should indicate cluster assignments of bins with colors. Specifically, the same set of colors is used in both the local plots (P1) as well as the global plots (P2).

(P4) Show input data and cluster assignment of an individual bin. The user can inspect the input data of an individual bin as well as its assigned cluster (if any).

Filtering. As we envision a tool that enables both local and global segmentation, the user can set filters in both a localized manner as well as a global manner.

(F1) Show only bins that occur in a localized genomic range. The user can restrict the shown bins to only those that occur within a user-specified linear genomic range via the local plots introduced in (P1).

(F2) Show only bins that occur within a range of BAF and RDR values. The user can specify a range of RDR and BAF values for each sample, restricting the tool to only show those bins whose RDR and BAF values occur within the specified ranges.

(F3) Zooming and panning can be reset to a default state where all bins are shown. This can be done on both the local and global plots.

(F4) Show only bins that occur on an individual chromosome. The user can specify an individual chromosome, restricting the tool to only show those bins that occur on the specified chromosome.

(F5) Show only bins that are assigned to a specified set of clusters. The user can specify one or more clusters, restricting the tool to only show those bins that are assigned to any of the specified clusters.

(F6) The same set of bins should be shown in both the local and global plots for each sample. To maintain visual consistency, it is important to ensure that the same set of bins is shown for each sample at all times. This is particularly important when altering filtering criteria via any of the aforementioned filtering tasks.

Selecting. We now proceed with introducing the requirements regarding selection functionality. Specifically, selection is an important step to enable the user to identify a group of bins and subsequently update their cluster assignments.

(S1) A range of localized bins can be selected and deselected. The user can add a range of bins to the current selection via the local plots introduced in (P1). Conversely, the user can remove bins from the current selection in a localized fashion.

(S2) A set of bins with similar RDR and BAF values in one sample can be selected and deselected. The user can add a set of bins with similar RDR and BAF values to the current selection via the global plots introduced in (P2). Moreover, the user can remove bins from the current selection using the same global plots.

(S3) The current selection can be cleared. The user can quickly clear the current selection (i.e. deselect all bins).

(S4) The current selection of bins must be shown in all local and global plots. The user can view the current selection in all plots across all samples.

Clustering. Next, the user can cluster the selected bins as captured by the following requirements.

(C1) The current selection of bins can be assigned to a new cluster. Specifically, the tool should identify a new cluster index that has not been previously used and assign the bins to this cluster. If a subset of the selected

bins were previously assigned to another cluster, they should be re-assigned to the new cluster.

(C2) The current selection of bins can be merged into an existing cluster. The user can select a previous cluster and assign the selected bins to this cluster. If a subset of selected bins were previously assigned to another cluster, they should be re-assigned to the cluster selected by the user.

(C3) The cluster assignments of the selected bins can be cleared. This functionality should reset the cluster assignments of the selection without assigning the selected bins to an existing cluster.

(C4) All cluster assignments can be cleared. The user can quickly clear cluster assignments of all bins.

(C5) The current selection of bins can be erased. To enable the user to remove outlier bins from consideration by downstream copy-number callers, the tool should provide functionality to erase the current selection.

(C6) Any of the aforementioned clustering tasks can be undone. To enable the user to recover from any mistakes during clustering (without starting over), the tool should maintain an undo stack.

(C7) A log of all clustering operations is maintained. To facilitate reproducibility, the tool should maintain an exportable log of all operations.

Analytics. Finally, the tool provides the user with feedback reflecting the current clustering. Specifically, we have the following analytics requirements.

(A1) Show metrics assessing homogeneity and separation for each cluster. A good clustering satisfies the following two criteria. First, bins within each cluster have similar RDR and BAF values per sample (i.e. homogeneity or cohesion). Second, distinct clusters are comprised of bins with distinct RDR and BAF values per sample (i.e.ṡeparation). Our tool should provide the user feedback of the current clustering regarding these two criteria. In particular, the tool should identify clusters that would benefit from further refinement.

(A2) Show centroids of currently selected clusters. To enable the user to visually assess the clustering, the tool can show cluster centroids in the global plots (P2). Only centroids for the currently selected clusters (F5) should be shown.

(A3) Show driver gene locations. To facilitate interpretation of CNAs, the tool should show driver gene locations in the linear plots (P1). The set of shown driver genes can be customized by the user (D6).

3 Methods

This section introduces CNAViz, a web-based tool for user-guided segmentation implemented using `D3` and `React`. CNAViz is open source and is available at: `https://github.com/elkebir-group/cnaviz`. Section 3.1 defines the input and output of the tool. Section 3.2 covers how the genomic bins are visualized and interacted with in CNAViz. The cluster analysis task are discussed in Section 3.3. Finally, Section 3.4 provides guidelines on how the tool can be used to perform *de novo* segmentation or refine an existing segmentation.

3.1 Input and Output

Input. Following (D1)-(D5), the user may upload a tab-separated values (TSV) file containing the RDR and BAF values of bins across multiple samples. The first row specifies column headers, which must contain 'CHR', 'START', 'END', 'RD', 'BAF' and, optionally, 'CLUSTER'. The order in which these columns are specified does not matter. If the 'CLUSTER' is not provided, then we consider all the genomic bins not clustered. That is, internally, we set $\mathrm{cluster}(i) = -1$ for each bin i. As these files can be large (about 10 MB for $m = 3$ whole genome samples with 50 Kb bins), we require the rows to be ordered as follows to facilitate fast processing. First, all bins part of the same chromosome must be grouped together and sorted by genomic position. Second, bins at the same genomic position, but from different samples are grouped together. Third, every genomic bin should be present in every sample. Note that the TSV input file may contain additional columns, which will not be used, but will be included in exported files as discussed below.

Per (D6), the user may also upload a list of driver genes. The input data for driver genes must have the following columns: 'symbol' and 'Genome Location'. The latter column is of the format '{CHR}:{START}-{END}'.

Output. Following (A4), the user may export the current clustering. When doing so, two files will be downloaded. One file contains a log of all clustering operations that were performed (C7). The other file adheres to the same TSV format used for input and specifies the clustering. Bins i that were erased (C5), which we internally assign cluster $\mathrm{cluster}(i) = -2$, will not be exported.

3.2 Plotting, Filtering, Selecting and Clustering

Visualization. The user interface of CNAViz is composed of a sidebar that can be hidden (Fig. 2a) as well as a main view containing the linear plots and scatter plot. We accomplish (P1) with two linear plots that both have the genomic position on the x-axis (Fig. 2c). On the y-axis, one of them contains the RDR, and the other contains the allelic imbalance. The *allelic imbalance* of a bin i in sample p is defined as $0.5 - \mathrm{BAF}(p, i)$. We accomplish (P2) using a scatter plot, where the x-axis shows the allelic imbalance and the y-axis shows the RDR of each bin (Fig. 2b). Note that rather than plotting BAFs directly, we chose to plot a transformation of the BAF so that bins that are unaffected by CNAs appear close to the origin in the scatter plot. Bins in both the linear and scatter plots are colored according to their cluster assignment (P3). Both the scatter plot and linear plots are side-by-side and display all bins from the same sample. Up to m scatter-linear plot pairs can be displayed at a time, where each pair displays all genomic bins part of one of the m samples. As a result, users can view genomic bins across multiple samples at the same time.

Each plot was created using the `D3`[1] and `D3FC`[2] libraries. In order to give the user maximum control over the clustering, all bins from the input data are plotted without any merging. We found that directly using SVG or drawing points using HTML Canvas does not scale to the number of bins that we have in our data ($\sim$ 50,000 bins). In order to efficiently plot a large number of bins, we used `D3FC` wrapper methods for WebGL. WebGL takes advantage of the rendering speed of the GPU, which allows for the efficient rendering of large amounts of data points. Each plot in CNAViz contains an SVG layer and WebGL layer to allow for both user interactivity and efficient rendering.

Interactivity. To support interactivity, our tool can be in one of three global states: (i) pan/zoom mode, (ii) add-to-selection mode and (iii) remove-from-selection mode. In all three modes, when hovering over any point in the scatter plot, a tooltip will appear with information about the genomic bin (P4). A D3 quadtree with all the points in the plot was used to achieve this effect. The genomic position of the bin being hovered over is displayed on the linear plots with a black bar.

If in pan/zoom mode (as indicated by the highlighted magnifying glass icon next to the sidebar), zooming and panning can both be done on the linear plots (F1) and scatter plot (F2). Otherwise, if in one of the two selection modes, the user can still pan/zoom on the axes, but not on the plot area. While in pan/zoom mode, if the user holds down 'shift' then bounding box zoom is enabled through clicking and dragging on the scatter plot and linear plot. The scales of the scatter plot are kept in sync with the y-axis

[1] `https://github.com/d3/d3`

[2] `https://github.com/d3fc/d3fc`

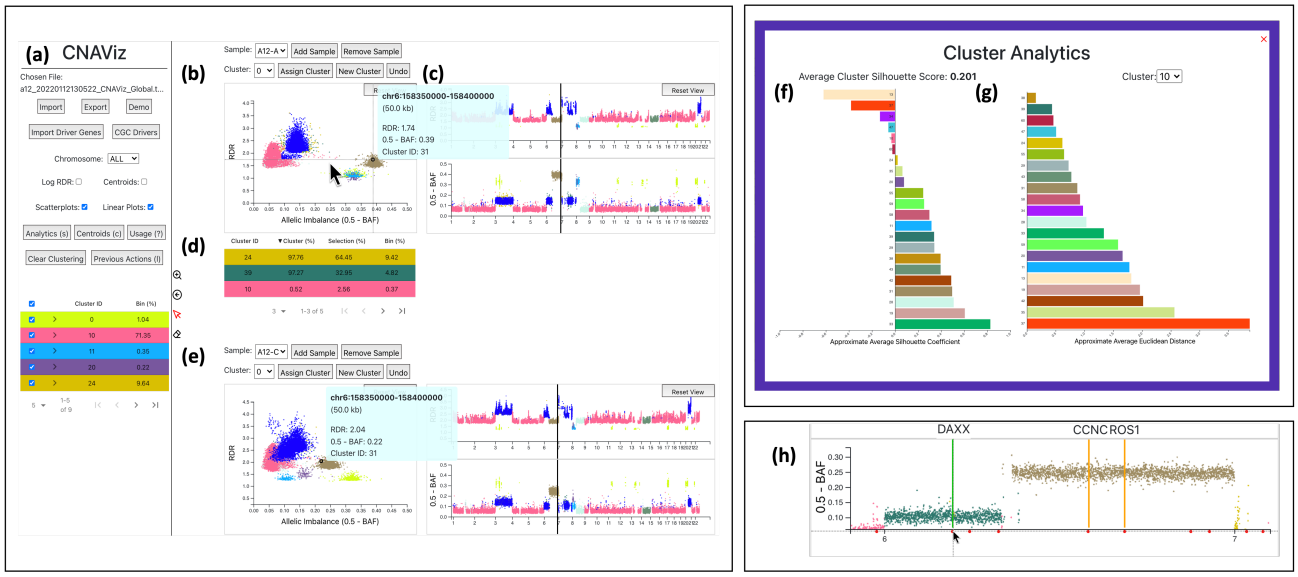


Fig. 2: **CNAViz contains a variety of options, modes, and plots to help the user create an effective segmentation.** (a) Sidebar containing import/export options, filtering by cluster or chromosome, analytics, centroids, etc. (b) Scatter plot with RDR on the y-axis and allelic imbalance on the x-axis. When hovering over a point in the scatter plot, a tooltip appears with information about the corresponding bin including the genomic position, bin size, RDR, allelic imbalance, and cluster ID. In addition, the hovered bin's position on the linear plots is indicated with a black bar. (c) RDR and allelic imbalance linear plots with genomic position on the x-axis. (d) When points are selected, the color of the bins on all plots changes to a dark blue color. The cluster composition of the selected points is displayed under the plots with a table, where the row color matches the cluster color in the plots. (e) Button to open/close sidebar (left arrow), as well as buttons to switch between three modes: pan/zoom (magnifying glass), add-to-selection (mouse pointer), and remove-from-selection (eraser). (f) Average silhouette coefficient bar plot. Above the bar plot, the average of the silhouette scores for each cluster is displayed. (g) Average Euclidean distance bar plot. Displays the average inter-cluster distance of each cluster to the cluster selected in the drop-down above the plot. (h) Driver genes are displayed as red dots along the x-axis of the linear plots. When a driver gene is clicked, it is locked in place and represented as an orange bar with the driver gene symbol above it. Hovering over one of the red dots allows the user to preview the driver gene (displayed as a green vertical bar).

scales of the linear plots, so any zooming done on the scatter plot will be reflected in the linear plots. The same is true the other way, any y-axis zoom on the linear plot will be reflected by the scatter plot. In addition, zooming along the x-axis (genomic position) of either linear plot will act as a filter for the scatter plot of the same sample. To reset any zooming or panning done on a plot, the user can click the corresponding 'Reset View' button in the top right of the plot (F3).

Beyond panning and zooming, there are two main ways of applying global filters to the genomic bins. First, the user can filter by chromosome through the use of the corresponding drop down in the sidebar (F4). Second, the user has the option to filter by cluster globally by using the cluster table in the sidebar (F5). For filtering, the `crossfilter`[3] library is used, which allows for filters along multiple dimensions to be added and removed with ease. Finally, both the scatter plots and linear plots for the same sample will always display the same bins as required by (F6).

In add-to-selection mode, points can be added to the current selection through the use of a bounding box in either the scatter or linear plots (S1)-(S2). The user can enter this mode by clicking the corresponding icon next to the sidebar (Fig. 2e). Alternatively, the user may temporarily enter this mode by holding down the control/command modifier key. To clear the selection, the user can click anywhere on the plot without dragging (S3). Points selected will be kept in sync between all plots in the visualization (between samples) as required by (S4). To erase part of a selection, the user can switch to remove-from-selection mode by clicking the corresponding icon next to the sidebar (Fig. 2e). Alternatively, the user may hold the alt/option key modifier to enter this mode. In this mode, all bins within the user-guided bounding box will be deselected (S1)-(S2). As points are selected and deselected, a variety of statistics for each cluster in the selection is displayed. For each cluster j, the user can see the percentage of bins assigned to cluster j covered by the selection, the percentage of the selection belonging to cluster j, and, finally, what percentage of bins the selection belonging to cluster j correspond to.

Once the user has selected the genomic bins desired, they can create a completely new clustering by clicking 'New Cluster', which will assign the points to the next cluster ID available (C1). Alternatively, they can choose an existing cluster from the drop down on the top of the scatter plot, and click 'Assign Cluster' (C2). This drop down also has options -1 as a temporary not clustered state (C3) and -2 which represents a deleted state (C5). The user has the ability to clear all cluster assignments by clicking on 'Clear Clustering' in the sidebar (C4). If the user wants to undo a cluster assignment, they can click 'Undo' (C6). To see information about the actions taken, the user can click the 'Previous Actions' button in the sidebar (C7).

3.3 Analytics

Visualization and Interactivity. In order to allow users to see how well they are clustering the data, we introduce a 'Cluster Analytics' tab (A1) that shows the silhouette values of the clustering (Rousseeuw, 1987) as well as pairwise cluster distances. Specifically, given m samples, we represent each bin i as a vector $\mathbf{v}_i = [\mathrm{RDR}(1,i), \ldots, \mathrm{RDR}(m,i), \mathrm{BAF}(1,i), \ldots, \mathrm{BAF}(m,i)]^\top$ in $2m$-dimensional space, combining the m RDR and the m BAF values of the

[3] `https://github.com/crossfilter/crossfilter`

bin across all m samples. This enables us to compute Euclidean distances between pairs of bins. To view analytics about the current clustering, the user can click the 'Analytics' button in the sidebar. A pop-up will appear that displays two bar plots (Fig. 2f, g).

The first bar plot shows the approximated average silhouette coefficient for each cluster j, fulfilling (A1). The silhouette value $s(i)$ of a bin i is a value between -1 and 1, where a high value indicates that the bin is well matched to other bins assigned to the same cluster (homogeneity/cohesion) and poorly matched to bins from other clusters (separation). The *silhouette coefficient* $s(j)$ of a cluster j is the mean silhouette value of all bins i assigned to cluster j. Computing the exact silhouette coefficient of each cluster is time intensive, i.e. it requires $O(n^2)$ time where the number n of bins is around 50000 for real data. Therefore, we approximate the computation silhouette coefficient via downsampling of points. The goal is to obtain a clustering with silhouette coefficients near 1.

The second bar plot represents the average Euclidean distance between the points of two clusters, which enables the user to identify pairs of clusters that can be merged. From the drop down above the plot, the user chooses a specific cluster for which to compute distances to other clusters. Clusters that have a distance near 0 to the specified cluster are good candidates for merging. The goal is to obtain clusters that show good separation, and have large pairwise Euclidean distances. Finally, we provide the user the ability to visualize cluster centroids through a checkbox in the sidebar (A2).

In order mark important driver genes on the linear plots, the user can upload a list of driver genes using the 'Import Driver Genes' button in the sidebar. The data must abide by the format described in Section 3.1. Following (A3), once uploaded, the driver genes will be represented by dots along the x-axis of the linear plots. By default, we use the driver genes published in the COSMIC Cancer gene census, and restricted ourselves to those genes for which a genomic location was provided (Tate *et al.*, 2019). Each driver gene marker acts as a toggle button, where if toggled on, the genomic region that the driver gene spans is highlighted. When hovering over one of the markers, the highlighted region can be previewed (Fig. 2h).

3.4 Usage Guidelines

In the following, we provide general guidelines on how CNAViz can be applied in each scenario. Screencasts and detailed tutorials demonstrating the application of these guidelines on real and simulated data are publicly available and can be found at `https://github.com/elkebir-group/cnaviz`.

Using CNAViz to Perform De Novo Segmentation. We begin by providing guidelines for *de novo* segmentation using CNAViz. We recommend displaying all samples in order to evaluate bins across samples concurrently. Moreover, we recommend using the scatter plot to quickly identify potential clusters that share similar RDR and BAF values across samples at a glance. However, the use of linear plots is essential to refine this clustering, especially in the presence of large number of clusters or clusters corresponding to small CNAs. Thus, both the scatter and linear plots should be used in the process of selecting relevant bins in the following three steps.

First, the user should select bins that are well separated on the scatter plot of a single sample. The user should then inspect whether these selected bins are also grouped together in other samples. In particular, selected bins that vary in one sample should be excluded from the current selection, and are good candidates for a new cluster. Second, the user should also use the linear plots to inspect whether these selected bins share RDR and BAF values across the genome. The linear plots are especially helpful to leverage the intuition that CNAs tend to occur in contiguous segments of the genome. Third, selected bins which share RDR and BAF values across samples can be made into a new cluster. This process should be repeated until each bin has been assigned to a cluster. When all bins have been clustered, the user can then proceed with the following steps to check an existing clustering.

Using CNAViz to Refine an Existing Segmentation. We now provide a few guidelines with which to evaluate and improve upon an existing clustering. The user should begin by displaying all samples. As a first step, the user should toggle the plots to show only the bins in one chromosome. This can be achieved using either the sidebar's chromosome menu, or via the zoom selection. The following steps should then be repeated for each chromosome.

First, if a pair of clusters share both RDR and BAF values across all samples, these clusters should be merged. The user may find the following subroutine for merging clusters helpful. (1) Note the cluster IDs in question. (2) Use the cluster check boxes in the left toolbar to visualize only the bins in these clusters. (3) Use the 'Reset View' button to ensure all cluster bins are visualized. (4) Select all bins and either assign them to an existing cluster or create a new cluster as appropriate. (5) Repeat this process as necessary.

Second, if a single cluster contains different RDR and BAF values, this cluster should be split into at least two clusters. We suggest the following procedure for splitting clusters. (1) Note the cluster ID in question, and the approximate corresponding range of RDR and BAF for each new cluster. (2) Use the cluster check boxes in the left toolbar to visualize only the bins in this cluster. (3) Use the 'Reset View' button to ensure all cluster bins are visualized. (4) Select the bins that should be separated, and create a new cluster. (5) Repeat this process as necessary so that each cluster has distinct RDR and BAF values.

Third, in an input clustering with several clusters which each have very few bins, it is often desirable to lessen the number of clusters by absorbing small clusters into larger ones. This is particularly relevant after inspecting and splitting each cluster, which results in the creation of several small clusters. The user should first verify that the largest clusters that incorporate the majority of bins are appropriately clustered – that is, each cluster's bins share a RDR and a BAF value that is distinct from all other bins. Next, given a small spurious cluster we suggest using the 'Analytics Tab' to identify a candidate largest cluster for merging. Finally, we recommend the user to iterate through these three steps until convergence.

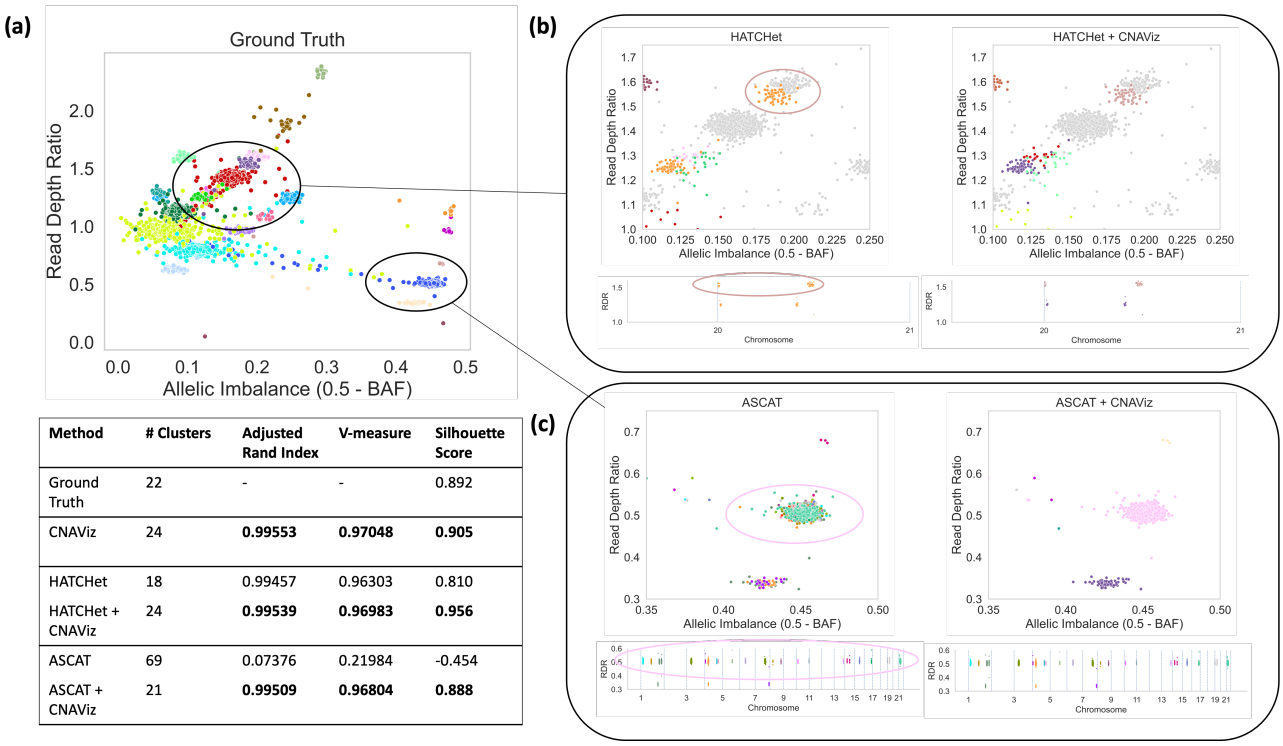
4 Results

We used published simulated datasets (Zaccaria and Raphael, 2020) generated from multi-sample DNA sequencing tumor samples to demonstrate how CNAViz improves upon existing segmentation algorithms in Section 4.1. Moreover, in Section 4.2 we demonstrate on a dataset of 6 tumor samples from 2 breast cancer patients that the novel features of CNAViz allows us to the accurately reveal CNAs affecting important cancer genes, which were previously missed by existing segmentation algorithms.

4.1 Validation of CNAViz using Simulations

Experimental Setup. To demonstrate the benefits of CNAViz, we used previously published data simulated with MASCoTE (Zaccaria and Raphael, 2020) for which ground truth is available and can be used for assessing segmentation performance. We considered the published dataset `n2_s4669/k4_01090_02008_00506035_00504055` with $m = 4$ bulk DNA sequencing samples comprising of 2 tumor clones.

To assess CNAViz's ability to perform accurate *de novo* segmentation as well as to assess improvement upon segmentations produced by existing methods, we performed three different experiments. First, we ran CNAViz in *de novo* mode by providing non-segmented data as input. Second, we provided CNAViz a segmentation solution generated by HATCHet, which performs global segmentation (Zaccaria and Raphael, 2020). Third, we



Method	# Clusters	Adjusted Rand Index	V-measure	Silhouette Score
Ground Truth	22	-	-	0.892
CNAViz	24	**0.99553**	**0.97048**	**0.905**
HATCHet	18	0.99457	0.96303	0.810
HATCHet + CNAViz	24	**0.99539**	**0.96983**	**0.956**
ASCAT	69	0.07376	0.21984	-0.454
ASCAT + CNAViz	21	**0.99509**	**0.96804**	**0.888**

Fig. 3: **CNAViz produces more accurate segmentations on simulated data in both *de novo* mode as well as when refining a given segmentation.** (a) A two-dimensional plot of RDR (y-axis) and allelic imbalance (x-axis, measured as $0.5 - \text{BAF}$) of 50 Kb genomic bins (points). Colors represent the ground-truth segments/clusters. Table shows performance metrics for each method. (b) Comparison of HATCHet's global segmentation solution before (left plots) and after refinement (HATCHet + CNAViz, right plots). (c) Comparison of ASCAT's local segmentation solution before (left plots) and after refinement (HATCHet + CNAViz, right plots). In each plot of (b) of (c) respectively, the same genomic bins are displayed, but colored according to each method's inferred segmentation.

input a segmentation solution generated by ASCAT, which performs local segmentation (Van Loo *et al.*, 2010; Ross *et al.*, 2021). We ran ASCAT in single-sample mode (aspcf) and provided it with ground-truth purity and ploidy values. We reconciled the sample-specific segmentation into a single sample-agnostic segmentation solution by retaining all breakpoints. We refer the reader to `https://github.com/elkebir-group/cnaviz` for screencasts describing the specific steps taken for this simulation instance. These follow the general guidelines described in Section 3.4.

Results. We evaluated the different clustering solutions using three performance metrics. These include the Adjusted Rand Index (ARI) (Hubert and Arabie, 1985), the V-measure (Rosenberg and Hirschberg, 2007) and the silhouette score (Rousseeuw, 1987). The ARI equals 0 when points are assigned to clusters randomly, and equals 1 when the inferred and ground-truth clustering solutions are the same. Likewise, the V-measure ranges from 0 (poor clustering) to 1 (matching ground-truth) (Rosenberg and Hirschberg, 2007). We refer to Section 3.3 for further details on interpreting the silhouette score.

We assessed the performance of five different segmentation solutions produced by (i) CNAViz, (ii) HATCHet, (iii) HATCHet + CNAViz, (iv) ASCAT, (v) ASCAT + CNAViz (Fig. 3a). Notably, the segmentation produced in CNAViz's *de novo* mode achieved the best overall clustering performance in terms of ARI and V-Measure (0.99553 and 0.97048, respectively). Given an existing solution, CNAViz also produced consistent improvements when compared to the original solution. Specifically, CNAViz produced the greatest improvement in terms of both ARI and V-measure (0.07376 to 0.99509 for ARI, and 0.21984 to 0.96804 for V-measure) when applied to the ASCAT solution. We also see modest improvements in these metrics for HATCHet.

Next, we present two specific examples of typical errors made in existing methods that CNAViz is able to fix (Fig. 3). First, CNAViz enables the user to improve the HATCHet solution by splitting a cluster. By visualizing the HATCHet solution using CNAViz's integrated scatter and linear plots, we can observe an orange cluster containing bins that separate into two distinct genomic segments along the genome (Fig. 3b). Therefore, we split the orange cluster into two separate clusters (Fig. 3b), matching ground truth (Fig. 3a). Second, CNAViz enables the user to combine distinct segments from across the genome into a single cluster. As a local segmentation method, ASCAT overclusters a single ground-truth cluster into 22 separate segments. ASCAT produces this clustering because the bins occur non-contiguously (Fig. 3c). With CNAViz's interactive scatter plot, we are able to both identify and reassign the cluster of bins (Fig. 3c), producing a cluster that matches ground truth (Fig. 3a).

4.2 Application of CNAViz to Real Data.

To investigate the impact of CNAViz's novel features, we applied CNAViz to DNA sequenced from six tumor samples across three breast cancer patients (P5, P6, P10) analyzed in the previous study of Casasent *et al.* (2018). In addition to standard bulk DNA sequencing of each tumor sample, the authors also performed matched high-resolution single-cell sequencing of every sample. As such, we can use these single-cell data to validate the CNAs inferred from the bulk sequencing data. Specifically, we plan to assess whether performing segmentation using CNAViz produces

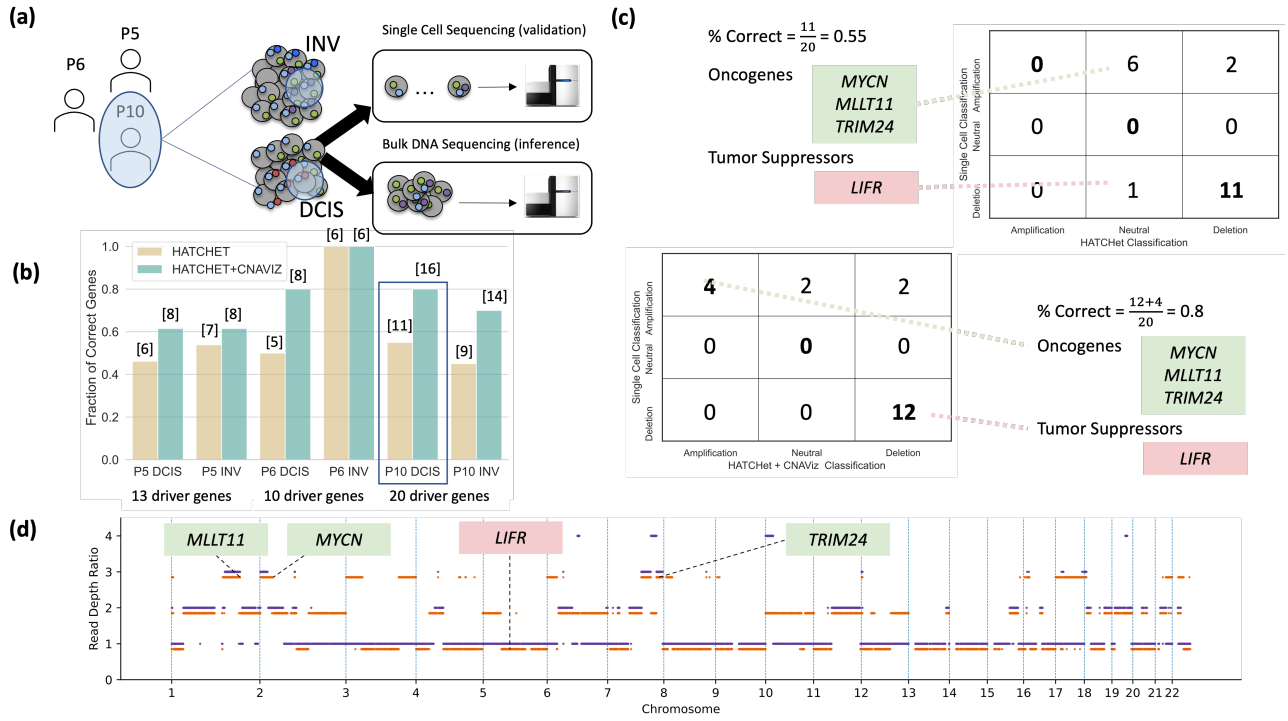


Fig. 4: **CNAViz results in more accurate identification of CNA status of breast cancer driver genes compared to an existing segmentation algorithm.** (a) CNAViz has been applied on the DNA sequencing data of two tumor samples (DCIS and INV) obtained from each of three breast cancer patients (P5, P6, and P10) analyzed by Casasent *et al.* (2018). (b) The number of correctly identified CNAs for breast cancer driver genes (y-axis) is reported across all samples of the three patients when using either the existing segmentation algorithm HATCHet (yellow) or refining its results with CNAViz (green). The number of correct driver genes is listed above each bar. (c) The number of breast-cancer driver genes with different types of CNAs inferred by either HATCHet (columns in top table) or HATCHet + CNAViz (columns in bottom table) is compared with the high-resolution CNAs measured by the matched classification in single-cell sequencing data (rows in both tables). (d) The CNAs (y-axis) inferred by HATCHet + CNAViz for two distinct sub-populations of cancer cells identified in Patient 10 are shown in orange and purple, with 0.15 separation for visual clarity.

downstream CNA calls that better match the single-cell data compared to using an existing segmentation method (Fig. 4a).

We processed the raw sequencing reads using the same pipeline reported in Casasent *et al.* (2018). After downloading the DNA sequencing data from the Sequence Read Archive (accession numbers SRP114962 and SRP116771), we aligned the reads to the human reference genome (hg19) using BWA (Li and Durbin, 2009). Then, the aligned sequencing reads were provided as input to HATCHet (Zaccaria and Raphael, 2020). Similar to other methods for copy number calling, HATCHet first performs segmentation before outputting copy number calls. Due to its modular design, it is possible to provide HATCHet with a custom segmentation. We created two sets of CNA calls for each patient. One set was obtained by running HATCHet end-to-end with its built-in global segmentation (denoted as 'HATCHet'). We extracted HATCHet's global segmentation and refined it using CNAViz (following the guidelines in Section 3.4). This enabled us to obtain a second set of CNA calls from HATCHet using the refined segmentation (denoted as 'HATCHet + CNAViz').

For each patient, Casasent *et al.* (2018) reported a small number of relevant breast cancer driver genes (ranging from 13 to 20). Using the single-cell CNA calls reported by the authors, we classified the driver genes of each patient as either unaffected, deleted, or amplified due to CNAs. We designated a driver gene as correctly classified if the CNA state inferred from bulk data matched the single-cell CNA state. We found that HATCHet + CNAViz classified a total of $44/86$ genes (51%) correctly compared to $60/86$ genes (70%) correctly classified by HATCHet (Fig. 4b). In particular, for sample P10 DCIS (ductal carcinoma *in situ*) HATCHet + CNAViz inferred 16 genes correctly compared to 15 genes correctly inferred by HATCHet. Further inspection reveals that HATCHet alone identified no amplified genes, and instead identifies 7 driver genes as neutral and 13 driver genes as deletions (Fig. 4c,d). By contrast, HATCHet + CNAViz identified 4 amplifications among driver genes, matching the ground-truth single-cell data. Among these, three are known oncogenes: *TRIM24* (Pathiraja *et al.*, 2015), *MYCN* (Schwab, 1991) and *MLLT11* (also known as AF1q) (Park *et al.*, 2015). Generally, we expect oncogenes to be amplified within tumor cells, as these mutations prove beneficial to tumor cells. Thus, the literature provides further evidence corroborating HATCHet + CNAViz's classification of these genes. Another difference between both approaches is the classification of the driver gene *LIFR*, which is a known tumor suppressor gene (Chen *et al.*, 2012). While HATCHet classified this gene as unaffected by CNAs, HATCHet + CNAViz classified the gene as affected by a deletion. This matches the expected behavior for tumor suppressor genes, which are frequently affected by deletions.

In summary, significant improvements in the accuracy of downstream copy-number analyses are possible with more accurate upstream segmentation. Here, we have illustrated improvements in the use case of driver gene classification, which were made possible by using CNAViz to refine the segmentation prior to copy number calling.

5 Discussion

Many cancer genomes are affected by copy-number aberrations (CNAs), making accurate characterization a critical step in improving our understanding of tumorigenesis as well as identifying treatment opportunities. Current CNA callers typically perform segmentation, either merging neighboring bins into segments (local segmentation) or clustering bins from across the genome (global segmentation). Importantly, both approaches suffer from limitations which result in either too many clusters in the case of local segmentation, or the omission of clusters corresponding to focal CNAs in the case of global segmentation. Here, we introduced CNAViz, a web-based tool to perform user-guided segmentation while taking both local and global perspectives into account. Thus CNAViz acquires the advantages of both approaches while overcoming their respective limitations. On simulated data, we demonstrated that CNAViz produces more accurate segmentations regardless of whether it is run in *de novo* mode or used to refine local or global segmentations. On real data, we demonstrated an example of how CNA analyses are afforded tangible downstream improvements by CNAViz.

There are several avenues for future research. First, while the 'Cluster Analytics' tab provides static feedback on the current segmentation, we envision the tool could provide real-time suggestions to further improve segmentation. Second, CNAs are often recurrent across patients with the same tumor type. Presently the tool operates on samples from one tumor at a time. In the future, we may consider generating suggestions based on segmented data from tumors in the same cohort. This will help further automate the process of generating and improving segmentation. Third, we propose an opt-in way for users to contribute segmentation solutions akin to crowd-sourcing efforts like FoldIt, enabling future developments of automated segmentation algorithms that incorporate successful strategies employed by expert users (Cooper *et al.*, 2010).

Funding

M.E-K. was supported by the National Science Foundation (grants: CCF-1850502 and CCF-2046488) as well as funding from the Cancer Center at Illinois. S.Z. was supported by the Rosetrees Trust grant reference M917.

References

Beroukhim, R. *et al.* (2010). The landscape of somatic copy-number alteration across human cancers. *Nature*, **463**(7283), 899–905.

Bielski, C. M. *et al.* (2018). Genome doubling shapes the evolution and prognosis of advanced cancers. *Nature genetics*, **50**(8), 1189–1195.

Boeva, V. *et al.* (2012). Control-freec: a tool for assessing copy number and allelic content using next-generation sequencing data. *Bioinformatics*, **28**(3), 423–425.

Casasent, A. K. *et al.* (2018). Multiclonal invasion in breast tumors identified by topographic single cell sequencing. *Cell*, **172**(1-2), 205–217.

Chen, D. *et al.* (2012). LIFR is a breast cancer metastasis suppressor upstream of the hippo-yap pathway and a prognostic marker. *Nature medicine*, **18**(10), 1511–1517.

Cohen-Sharir, Y. *et al.* (2021). Aneuploidy renders cancer cells vulnerable to mitotic checkpoint inhibition. *Nature*, **590**(7846), 486–491.

Cooper, S. *et al.* (2010). Predicting protein structures with a multiplayer online game. *Nature*, **466**(7307), 756–760.

Dentro, S. C. *et al.* (2021). Characterizing genetic intra-tumor heterogeneity across 2,658 human cancer genomes. *Cell*, **184**(8), 2239–2254.

Garvin, T. *et al.* (2015). Interactive analysis and assessment of single-cell copy-number variations. *Nature methods*, **12**(11), 1058–1060.

Ha, G. *et al.* (2014). Titan: inference of copy number architectures in clonal cell populations from tumor whole-genome sequence data. *Genome research*, **24**(11), 1881–1893.

Hubert, L. and Arabie, P. (1985). Comparing partitions. *Journal of classification*, **2**(1), 193–218.

Jamal-Hanjani, M. *et al.* (2017). Tracking the evolution of non–small-cell lung cancer. *New England Journal of Medicine*, **376**(22), 2109–2121.

Laks, E. *et al.* (2019). Clonal decomposition and dna replication states defined by scaled single-cell genome sequencing. *Cell*, **179**(5), 1207–1221.

Li, H. and Durbin, R. (2009). Fast and accurate short read alignment with burrows–wheeler transform. *bioinformatics*, **25**(14), 1754–1760.

McGranahan, N. and Swanton, C. (2015). Biological and therapeutic impact of intratumor heterogeneity in cancer evolution. *Cancer cell*, **27**(1), 15–26.

Memon, D. *et al.* (2021). Copy number aberrations drive kinase rewiring, leading to genetic vulnerabilities in cancer. *Cell reports*, **35**(7), 109155.

Notta, F. *et al.* (2016). A renewed model of pancreatic cancer evolution based on genomic rearrangement patterns. *Nature*, **538**(7625), 378–382.

Park, J. *et al.* (2015). AF1q is a novel TCF7 co-factor which activates cd44 and promotes breast cancer metastasis. *Oncotarget*, **6**(24), 20697.

Pathiraja, T. N. *et al.* (2015). TRIM24 links glucose metabolism with transformation of human mammary epithelial cells. *Oncogene*, **34**(22), 2836–2845.

Quinton, R. J. *et al.* (2021). Whole-genome doubling confers unique genetic vulnerabilities on tumour cells. *Nature*, **590**(7846), 492–497.

Rosenberg, A. and Hirschberg, J. (2007). V-measure: A conditional entropy-based external cluster evaluation measure. pages 410–420.

Ross, E. M. *et al.* (2021). Allele-specific multi-sample copy number segmentation in ASCAT. *Bioinformatics*, **37**(13), 1909–1911.

Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, **20**, 53–65.

Schwab, M. (1991). Enhanced expression of the cellular oncogene MYCN and progression of human neuroblastoma. *Advances in enzyme regulation*, **31**, 329–338.

Shen, R. and Seshan, V. E. (2016). Facets: allele-specific copy number and clonal heterogeneity analysis tool for high-throughput dna sequencing. *Nucleic acids research*, **44**(16), e131–e131.

Tarabichi, M. *et al.* (2021). A practical guide to cancer subclonal reconstruction from dna sequencing. *Nature methods*, **18**(2), 144–155.

Tate, J. G. *et al.* (2019). Cosmic: the catalogue of somatic mutations in cancer. *Nucleic acids research*, **47**(D1), D941–D947.

The PCAWG Consortium *et al.* (2020). Pan-cancer analysis of whole genomes. *Nature*, **578**(7793), 82.

Van Loo, P. *et al.* (2010). Allele-specific copy number analysis of tumors. *Proceedings of the National Academy of Sciences*, **107**(39), 16910–16915.

Watkins, T. B. *et al.* (2020). Pervasive chromosomal instability and karyotype order in tumour evolution. *Nature*, **587**(7832), 126–132.

Xi, R. *et al.* (2011). Copy number variation detection in whole-genome sequencing data using the bayesian information criterion. *Proceedings of the National Academy of Sciences*, **108**(46), E1128–E1136.

Zaccaria, S. and Raphael, B. J. (2020). Accurate quantification of copy-number aberrations and whole-genome duplications in multi-sample tumor sequencing data. *Nature Communications*, **11**(1), 4301.

Zaccaria, S. and Raphael, B. J. (2021). Characterizing allele-and haplotype-specific copy numbers in single cells with CHISEL. *Nature biotechnology*, **39**(2), 207–214.

Zack, T. I. *et al.* (2013). Pan-cancer patterns of somatic copy number alteration. *Nature genetics*, **45**(10), 1134–1140.

Zare, F. *et al.* (2017). An evaluation of copy number variation detection tools for cancer using whole exome sequencing data. *BMC bioinformatics*, **18**(1), 1–13.