XClone: detection of allele-specific subclonal copy number variations from single-cell transcriptomic data

Rongting Huang[1,*], Xianjie Huang[1,2,*], Yin Tong[3], Helen Y.N. Yan[3], Suet Yi Leung[3,4,5], Oliver Stegle[6,7], and Yuanhua Huang[1,2,8,#]

[1] School of Biomedical Sciences, The University of Hong Kong, Hong Kong SAR, China
[2] Center for Translational Stem Cell Biology, Hong Kong Science and Technology Park, Hong Kong SAR, China
[3] Department of Pathology, School of Clinical Medicine, LKS Faculty of Medicine, The University of Hong Kong, Queen Mary Hospital, Hong Kong SAR, China
[4] The Jockey Club Centre for Clinical Innovation and Discovery, LKS Faculty of Medicine, The University of Hong Kong, Pokfulam, Hong Kong SAR, China
[5] Centre for PanorOmic Sciences, LKS Faculty of Medicine, The University of Hong Kong, Pokfulam, Hong Kong SAR, China
[6] Division of Computational Genomics and Systems Genetics, German Cancer Research Center (DKFZ), Heidelberg, Germany
[7] Genome Biology Unit, European Molecular Biology Laboratory, Heidelberg, Germany
[8] Department of Statistics and Actuarial Science, The University of Hong Kong, Pokfulam, Hong Kong SAR, China
[*]: Co-first authors; [#]: Correspondence: `yuanhua@hku.hk`

Abstract. Somatic copy number variations (CNVs) are major mutations in various cancers for their development and clonal progression. A few computational methods have been proposed to detect CNVs from single-cell transcriptomic data. Still, the technical sparsity makes it challenging to identify allele-specific CNVs, especially in complex clonal structures. Here we present a statistical method, XClone, to detect haplotype-aware CNVs by integrating expression levels and allelic balance from scRNA-seq data. With well-annotated datasets on multiple cancer types, we demonstrated that XClone could accurately detect different types of allele-specific CNVs, enabling the discovery of the corresponding subclones and dissection of their phenotypic impacts.

Keywords: Copy number variation · Allelic imbalance · Somatic clone · ScRNA-seq modelling.

1 Introduction

The copy number variations (CNVs), including copy loss, gain and copy-neutral loss-of-heterozygosity (CN-LOH or interchangeably LOH hereafter), have been widely shown associations in various cancers [1,2]. CNVs have also been found predictive to clinical outcomes in patients and can reoccur in certain tumour types [3]. Accurate detection of CNVs is therefore crucial for studying and understanding the genetic factors that contribute to the development and progression of cancers. The high-throughput sequencing is one of the most widely used techniques for studying CNVs in cancers initially in a bulk manner, e.g., in the pan-cancer analysis [4]. Soon, the advent of single-cell sequencing started enlightening our understanding of CNVs and their clonal evolution. Recent advances in single-cell DNA-seq at both technology and computational levels, e.g., ACT [5] and CHISEL [6], have further enhanced the analysis of CNV heterogeneity and the potential association with clonal fitness [7].

CNV analysis has also been performed by using scRNA-seq platforms thanks to its dual readout of transcriptome and genetic makeups, yet its high technical sparsity makes it challenging to accurately call CNVs at a single-cell resolution. A few methods have recently been developed to tackle these challenges. InferCNV is one of the first methods proposed for this purpose [8] and remains widely used with achieving reasonable accuracy probably thanks to its effective smoothing by a moving-average method and multiple denoising treatments. CopyKAT is another recently proposed method that leverages a polynomial dynamic programming model to smooth the transformed data along genomic coordinates and had an emphasis on

the detection of diploid cells [9]. However, both InferCNV and CopyKAT do not consider allelic information and hence cannot detect copy-neutral LOH. On the other hand, HoneyBADGER [10] and CaSpER [11] were proposed to account for the allelic shift signal for detecting a full spectrum of CNVs. Given the coverage of a single variant is extremely low, their independent use of individual SNPs often suffers from high noise. Instead, Numbat was recently introduced to substantially resolve such challenge by aggregating multiple variants via population haplotype-based phasing [12], a strategy that has been proved to be powerful in scDNA-seq [6]. However, this method may sacrifice the resolution of clonal CNVs from a single-cell level to a pseudo-bulk level and its performance remains to be assessed in a quantitative manner, especially on samples with complex CNV clonal structures.

To overcome these challenges, we introduce *XClone*, a statistical model to detect allele-specific CNVs on individual cells by modelling the raw read/UMI counts from scRNA-seq data. Briefly, this method accounts for two modules: the B-allele frequency (BAF) of heterozygous variants and the sequencing read depth ratio (RDR) of individual genes, respectively detecting the variation states on allelic balance and absolute copy number, which are further combined to generate the final CNV states (Fig. **1**a, Supp. Fig. S1-2, Methods). Uniquely, our method introduced two major innovations to boost the signal-to-noise ratio in detecting CNV states. First, in the BAF module, we implemented a three-step of haplotyping, from individual SNPs to a gene by population-based phasing, from consecutive genes (100 by default) to a gene bin by an Expectation-Maximization algorithm, and from gene bins to a chromosome arm by a dynamic programming method. Importantly, we re-designed our original genotyping software cellsnp-lite [13] to a tailored package *xcltk* to address the issue of double counting UMIs that cover multiple nearby SNPs (see Methods). Second, we employed two orthogonal strategies to smooth the CNV-state assignments on both BAF and RDR modules: horizontally with hidden Markov models along the genome and vertically with a k-nearest-neighbours (KNN) cell-cell connectivity graph, which not only denoises the data but also preserves the single-cell resolution.

2 Results

2.1 Assessment of CNV detection with well-annotated scRNA-seq data

The assessment of CNV detection is often challenging due to the lack of reliable ground truth. To fill this gap, here we collected a fresh gastric cancer tissue and generated both single-cell RNA-seq data (5,400 cells by 5' 10x Genomics) and single-cell DNA-seq (10x CNV kit; Fig. **1**b-c), together with bulk whole exome sequencing (WES) on its frozen sample (Supp. Fig. S3). The CNV states and regions called from both bulk WES and scDNA-seq are highly concordant, including copy gain at chr8, chr20, copy loss at chr12p (chr12:7288360-16757762) and chr18, and CN-neutral LOH at chr1q, chr6p, chr10q and chr11q (Fig. **1**b; Supp. Fig.S3; Supp. Table S1-2). Additionally, the scDNA CNV clonal structure is relatively simple in this sample, consisting of two major subgroups, with/without copy number changes. The scRNA data clustering of this sample further confirmed this clonal structure by separating the cells into two significant populations: one (3227 in 5400, 59.8%) was classified as tumor microenvironment cells with immune or fibroblast markers expressed, the other one (2153 in 5400, 39.9%) was classified as gastric stem/carcinoma cells with strong expression of stem cell markers and EPCAM (Fig. **1**c, Supp. Fig. S4). These two scRNA groups are considered concordant with the two scDNA groups respectively with CN neutral and harbouring CNVs, hence it can serve as a high-quality benchmarking data set. Then, we applied XClone to this scRNA-seq data by using immune cells as diploid reference (see details in Methods) and found that its RDR module can accurately detect the regions with absolute copy number changes (chr8, chr18 and chr20; Fig. **1**e and Supp. Fig. S5) and the BAF module has the complementary capability in detecting CNVs with allele ratio deviance (chr1q, chr6p, chr8, chr10q, chr11q, chr18, chr20; Fig. **1**f). When combining the two modules, XClone precisely detects all four types of CNV states in tumour cells (copy gain, copy-neutral LOH, loss and neutral; Fig. **1**g).

As a comparison, we applied four popular methods, InferCNV, CopyKAT, CaSpER and Numbat to this dataset (Supp. Fig. S6). We also introduced a quantitative assessment metric with a receiver operating characteristic (ROC) curve at an entry level for each CNV state, namely using the assignment probability (or transformed value) of each gene in each cell against the ground truth annotation (Fig. **1**h-j; Methods).

InferCNV and CopyKAT, with only expression level information, generally work well in detecting CNVs with absolute CN changes on gain or loss (Supp. Fig. S6 visually and Fig. **1**h-i quantitatively). Interestingly, for copy gain, XClone and Numbat further improve the performance substantially (AUROC from 0.89 by InferCNV to 0.99), presumably thanks to its extra BAF information and denoise treatment. For allele-specific CNV (LOH), XClone and Numbat also achieve high AUROC (>0.98), largely outperforming the counterpart (CaSpER; AUROC=0.65). For copy loss, XClone and Numbat also achieve high AUROC (>0.98), which is comparable to InferCNV (0.9768). Overall, on this gastric cancer sample, XClone and Numbat are ranked as the top two methods for comprehensive CNV detection; they both perform accurately in copy gain, copy loss and LOH detection. As a minor note, according to the Fig. **1**e-g, XClone's property of preserving single-cell-resolution CNV states may weaken the signal to detect small CNV fragments confidently (CN loss on chr12p). By contrast, Numbat uses a preprocessing step to define the CNV segments at the bulk level, which may help concentrate on the CNV region, but this strategy may lead to the mis-recognition of the whole regions of complex clonal CNV structure (e.g., the glioma below), while XClone's high cellular resolution have better capability to avoid it. Additionally, XClone has higher computational efficiency by reducing around 60% of the running time compared to Numbat (1,025 vs 2,617 seconds; Supp. Table S5).

2.2 XClone identifies differential allele loss between subclones in glioma

To further assess its robustness on tumors with complex clonal structure, we applied XClone to a public glioma sample BCH869 with histone H3 lysine27-to-methionine mutations (H3K27M-glioma), where 489 malignant cells and 3 non-tumour cells were probed by smart-seq2 [14]. As reported in the original study, a distinct subclonal structure has been observed in this population of cells by combining scRNA-seq (by using InferCNV) and whole-genome sequencing (WGS) data, where two major clones have different allele loss on chr14 (Fig. **2**a) When using XClone's RDR module, we indeed found a copy loss on chr14 for almost all cells (Fig. **2**b and Supp. Fig. S8), which is consistent with that of InferCNV (Supp. Fig. S9b). Uniquely, when leveraging the BAF module, XClone detects the different allele loss on this chromosome 14, matching well with the reported clones (Fig. **2**c-d, Supp. Fig. S8). Interestingly, XClone's BAF module detects medium but consistent allelic shifts on chr7 (clone 1), chr17q and chr19p, endorsing the probable copy gain, instead of copy loss or LOH. Therefore, besides chr14, XClone's combined mode further detects consensus CNVs (copy gain on chr1q, chr2p and chr17q) and clone-specific CNVs including copy gain on chr7 and LOH on chr10 in clone 1 and copy gain on chr2q in both clones 1 and 3 (Fig. **2**d).

As a comparison, we also ran other tools on this dataset. Overall, InferCNV and CopyKAT well identified the absolute CN change (Supp. Fig. S9) while Numbat is the only one showing a potential to capture the allele-specific CNVs. Specifically, Numbat identifies the major copy gains and performs well on LOH, but it shows a fatal defect in copy loss detection by missing one of the two allele losses on chr14 (Fig. **2**e). By contrast, XClone identifies nearly all CNVs across clones. This visual comparison is further supported by the quantitative assessment with ROC curves as introduced above. XClone indeed achieves remarkable improvement or comparable performance on AUROC for all types of CNVs compared to the best counterpart: copy gain (0.925 vs 0.853; Fig. **2**f, with InferCNV), copy loss (0.986 vs 0.947; Fig. **2**g, with InferCNV) and LOH (0.992 vs 0.994; Fig. **2**h, with Numbat). Of note, although presented in XClone's plotting, the cell cluster information is not used in CNV estimation (a similar setting to InferCNV), hence it preserves the detection at the cell level, instead of the cluster level.

2.3 XClone detects wide existence of LOH in a TNBC sample

Next, we applied XClone to a triple-negative breast cancer (TNBC) sample that was assayed by droplet-based scRNA-seq (10x Genomics). As reported by the original study using CopyKAT, three clusters of cells were identified from the transcriptome, consisting of 300 normal cells and 797 tumour cells from two distinct CNV clones [9]. By re-analysing this dataset, our RDR module confirms both consensus CNVs (chr1q, chr6p, chr8, chr12p, chr18) and subclonal CNVs including chr2q, chr4p, chr6q, chr7q, chr15p, chr16p in Clone 1 and chrX in Clone 2 (Fig. **E1**b, Supp. Fig. S10). On the other hand, when applying XClone's BAF module, various

regions were found with copy neutral in absolute number but imbalanced allele frequency (Fig. **E1**c-d). This is further confirmed by XClone's combined mode, presenting a wide existence of copy-neutral LOH (Fig. **E1**e), which cannot be detected by methods omitting the allelic information, e.g., CopyKAT and InferCNV (Supp. Fig. S11). One example is chr1p where strong allelic bias is observed by the smoothed BAF signal (Fig. **E1**c), further detected as LOH by XClone (with a minor probability of copy loss), while it is largely missed by Numbat (Fig. **E1**f). Specifically, consensus LOH includes chr1p and chr21, potentially consensus LOH in chr11p, chr13, chr14, chr17, chr19 with strong allele bias support, while clone 2 has specific LOH on chr3p and chr22. Taking one step further, we leveraged the CNV probability detected by XClone to explore potential subclones within this sample, e.g., by K-means clustering on the flattened probability matrix here (XClone_ID in Fig. **E1**e; Methods). The clonal reconstruction by XClone is perfectly concordant with the original one (ARI metric: 0.965).

In conclusion, we introduced XClone, a statistical method that effectively models the CNV states with both RDR and BAF modules. When applying it to three types of unique cancer samples (gastric, glioma and TNBC, respectively with quality annotation, complex clonal structure, and ubiquitous CNVs), XClone showed its powerful and robust capability in detecting allele-specific CNV states and capturing a broader spectrum of genetic variations, therefore can further help reveal the phenotypic impact. Thanks to the high accuracy in detecting CNVs at a cell level, XClone may enable the discovery of fine-resolved subclonal structures, hence providing insights for therapeutic strategies. Admittedly, further validation of the potentially identified subclones will be beneficial, possibly with other techniques, e.g., mitochondrial variations [15,16].

3 Methods

3.1 Overview of XClone suite

XClone has an end-to-end pipeline for CNV analysis from scRNA-seq data. It takes bam file(s) as input and models the raw UMI or read counts; then it outputs both the smoothed BAF and RDR matrices for visualization and the detected cell-by-gene CNV states as the final results. It also requires users to specify a diploid cell group as a reference. There are four major steps, including pre-processing, pre-analysis, modelling, and post-analysis, as described in detail below and illustrated in (Fig.**1**a).

3.2 Preprocessing: extract count matrices with xcltk

In order to pre-process single-cell data for XClone, we developed a toolkit xcltk (`https://pypi.org/project/xcltk/`) to extract cell-by-gene count matrices from BAM file(s) for both RDR and BAF modules. The input can be a single BAM file containing cell barcodes (e.g., by 10x Genomics) or multiple BAM files with one file per cell (e.g., by smart-seq2). Here, we used annotated genes (32,696 for hg19; 33,472 for hg38) from GENCODE (the same as Cell Ranger) as default; users can also use any customized gene (or other feature) annotations.

For read depth count generation, xcltk calculates the UMIs (if applicable) or reads counts for each gene in each cell and returns a sparse matrix. This processing is highly similar to Cell Ranger developed by 10x Genomics, including reads filtering (by default only those reads with MAPQ$<$ 20, aligned_length$<$30nt or FLAG$>$255 are filtered).

For allele-specific count generation, xcltk requires one extra file besides the default gene and cell lists: a VCF file containing phased germline genotypes. Then, xcltk takes all heterozygous variants, counts the UMIs or reads supporting each allele, and returns two cell-by-SNPs matrices of the alternative allele (i.e., AD) or the depths (i.e., DP). By leveraging the genotype phasing information, xcltk further aggregates multiple SNPs from one gene, and compresses both AD and DP into cell-by-gene matrices. Of note, when a UMI (or read) covers two or multiple variants on one gene, it is critical to only count them once and avoid double counting; xcltk implemented a strategy for this purpose by keeping the UMI (or read) IDs for each SNP and only counting them once when aggregating all SNPs.

3.3 Pre-analysis: three-level allele phasing for haplotype-specific BAF

In XClone, in order to enhance the signal-to-noise ratio when analysing the allele information in the BAF module, we applied three-level allele phasing. First, we employed the human population haplotype reference to phase individual SNPs, i.e., **reference-based phasing**. By default, the phasing is achieved by EAGEL2 [17] with Haplotype Reference Consortium panel [18], powered by the Sanger imputation server (`https://imputation.sanger.ac.uk`). Then xcltk will count the phased alleles for SNPs across a gene (see above).

Second, to avoid the errors in the population-based phasing when it comes to a longer genome range covering many genes, we further performed a **gene-bin phasing** (local phasing) to link a group of proximate and consecutive genes (we call it a gene bin, by default 100 genes). Intuitively, in each cell, we force the allele frequency AF to be the same for all genes in a gene bin, allowing flipping the allele of each gene. Namely, it assumes the CNV states are the same within a gene bin in a cell, even though the CNV states can be different from cell to cell (e.g., some are normal cells and some are tumour cells). Similar to CHISEL [6], we implemented an EM algorithm (Supp. Algorithm 1) to achieve this by flipping the allele of each gene within a gene bin.

Third, for synchronising the BAF for visualisation, we introduced a **chromosome-range phasing** (global phasing) to further link all gene bins on each chromosome arm. Similar to the gene-bin phasing, this step can also use the genomic auto-correlation on the CNV states, which could be addressed by a dynamic programming algorithm, a Hidden Markov model or a Gaussian process model. As this step is relatively independent of CNV state identification, we adopted the dynamic programming algorithm proposed in CHISEL for simplicity and efficiency (see Supp. Algorithm 2). Note, both the gene-bin and chromosome-range phasing steps depend on the auto-correlation of BAF deviance from 0.5 (theoretical BAF value for allele balance). Otherwise, if there is no allelic imbalance, it will return arbitrary phasing, which is irrelevant to the CNV state inference. The three-level allele phasing illustration is shown in Supp. Fig. S1 and the remarkable performance of the phasing of the three datasets were visualised in Supp. Fig. S5, S8, and S10.

3.4 Pre-analysis: horizontal and vertical smoothing for visualization

Before detecting the CNV states, it is beneficial to visualise the raw signal of both RDR and BAF modules. Due to the high technical sparsity in scRNA-seq data, it is necessary to perform smoothing for the visualisation. Here, we performed both a horizontal smoothing by a weighted moving average (WMA) along with each chromosome and a vertical smoothing across cells via a k-nearest neighbour (KNN) graph.

Taking the read depth ratio (RDR) module as an example, the raw count matrix was first normalized to its total library counts, then divided by reference cell expression profile, and finally transformed to the logarithm scale as log ratio. Then this raw RDR matrix X is smoothed horizontally by a WMA connectivity matrix W and vertically by a KNN connectivity matrix M, as follows,

$$\tilde{X} = M \times X \times W, \tag{1}$$

where M is a cell-by-cell connectivity matrix generated from the KNN graph (k=10 by default) of routine scRNA-seq analysis `scanpy.pp.neighbors()`, and W is a gene-by-gene connectivity matrix linking each gene to its proximate consecutive genes on each chromosome arm (window size t=101 by default), as follows

$$W_{j,g} = \frac{W^o_{j,g}}{\sum_h W^o_{h,g}}, W^o_{j,g} = \begin{cases} t - |d(j;g)|, & 0 < |d(j;g)| < t \\ t * 2, & d(j;g) = 0 \\ 0, & |d(j;g)| \geq t, \end{cases}$$

where $d(j;g)$ refers to the order distance from any gene j to a query gene g on genome coordinate. Of note, W is column normalised and M is row normalised, namely with sum to one.

For the B allele frequency (BAF) module, we took the cell-by-gene_bin matrix as input and performed a similar smoothing both horizontally (with WMA) and vertically (with KNN). As an example, the smoothing steps for GX109-T1c scRNA were shown in Supp. Fig. S5. Once the smoothed RDR and BAF matrices are generated, they can serve both visual representation and empirical extraction of CNV parameters (see below).

3.5 Modelling: mixture model for expression data (RDR module)

The Read Depth Ratio (RDR) module aims to detect the states of absolute copy numbers, i.e., CN-neutral, loss, and gain, for each gene in each cell. To start, it requires specifying reference cells (i.e., diploid cells), which can be picked from pre-annotated cell types from transcriptome by examining markers of non-tumour cells. Given the reference cells, XClone will calculate the reference average gene expression (ref_avg) and filter the genes with low expression (default 0.5 for 10x genomics and 1.8 for smart-seq). Optionally, it will further remove top marker genes for each pre-annotation cell group (15 by default), e.g., as detected by the `scanpy.pp.rank_genes_groups()`.

Then, we used a negative binomial distribution to model the observed UMI (or read) counts for each CNV state in a generalized linear model framework, as follows

$$\begin{aligned} p(X_{c,g}|z_{c,g,k}=1) &= \texttt{NB}(X_{c,g}|\mu_{c,g,k},\phi_g) \\ \mu_{c,g,k} &= l_c \times X_{ref,g} \times C_k, \end{aligned} \tag{2}$$

where $X_{c,g}$ denotes the raw read/UMI count at gene g cell c; l_c denotes the cell-specific library size factor; $X_{ref,g}$ denotes the average count value for gene g in given reference diploid cells; C_k denotes the copy number ratio compared to reference diploid state for each CNV state. The latent variable $z_{c,g,k}$ indicates if gene g in cell c belongs to CNV state k ($z_{c,g,k}=1$) or not ($z_{c,g,k}=0$). In this module, we defined three states with index $k=0$ for copy loss, $k=1$ for copy neutral and $k=2$ for copy gain.

Considering that the reference count values $X_{ref,g}$ can be obtained directly, it leaves only three sets of parameters to be estimated: l_c, C_k and ϕ_g. Here, we estimated them separately by a maximum likelihood estimation with the `statsmodels` package [19] or using a heuristic method. To estimate the library size factor l_c for each cell c, by default, we used the ratio of its total sum count to the average of reference cells. To estimate the gene-specific dispersions ϕ_g, we fit a negative binomial GLM per gene in the form of Eq. (2) but only on the reference cells ($C_k=1$) with the fitted library size l_c from the previous step.

For the CNV ratio C_k, one can directly use a theoretical value, namely $\{0.5, 1, 1.5\}$ respectively for loss, neutral and gain. Here, we further employed a heuristic method from the empirical distribution. Based on the smoothed RDR value, we selected 3 guided genomic regions (chromosomes or arms) that are most likely to be copy loss, copy neutral and copy gain regions, as ranked by the mean. In general, CNV regions only cover a small percentage of the whole cell-by-gene matrix and the extreme ratio value at each CNV state can be used to guide CNV ratio initialization. Thus, the CNV ratio C_k will be extracted from quantiles of each of the corresponding chromosomes (or arms), specifically default quantile 0.96 in copy neutral, quantile 0.0001 in copy loss and quantile 0.99 in copy gain regions.

Once all three sets of parameters are fitted via the above procedures, we can calculate the probability of assigning each gene in each cell to a certain CNV state via Eq.(2). Namely, we can obtain a cell-gene-state probability tensor, as follows

$$p(z_{c,g,k}=1|X_{c,g},l_c,X_{ref,g},\boldsymbol{C}) = \frac{\texttt{NB}(X_{c,g}|l_c \cdot X_{ref,g} \cdot C_k,\phi_g)}{\sum_k \texttt{NB}(X_{c,g}|l_c \cdot X_{ref,g} \cdot C_k,\phi_g)}. \tag{3}$$

3.6 Modelling: mixture model for phased allele-specific data (BAF module)

For the B-Allele Frequency (BAF) module, we took both AD and DP sparse matrices from xcltk, removed marker genes provided by the RDR module, and performed the three-level phasing to generate gene_bin based matrices as inputs for XClone modelling. Here, we used a beta-binomial distribution to model the observed UMI (or read) counts for each CNV state, as follows

$$p(a_{c,g}|d_{c,g},z_{c,g,k}=1) = \texttt{BB}(a_{c,g}|d_{c,g},\rho_{g,k},\tau_{g,k}), \tag{4}$$

where $a_{c,g}$ and $d_{c,g}$ denote phased B allele count and total count at gene_bin g in cell c, respectively. The Beta-binomial is parameterised by the mean success rate ρ and concentration τ (namely $\alpha=\rho\tau$ and $\beta=(1-\rho)\tau$

in conventional parameterization). Here, we treat the concentration as a technical factor and shared by all gene_bins in droplet-based data ($\tau_{g,k} = 100$ by default) while it can vary in smart-seq data depending on the coverage of each gene_bin.

Here, $z_{c,g,k}$ is a latent variable indicating the cell c in gene_bin g is CNV state k ($z_{c,g,k} = 1$) or not ($z_{c,g,k} = 0$). The parameter $\rho_{g,k}$ denotes the expected allele frequency of gene_bin g in CNV state k. The state index $k = \{0, 1, 2, 3, 4\}$ is respectively for allele A strong bias, allele A minor bias, allele balance, allele B minor bias, and allele B strong bias, with the corresponding theoretical allele frequencies $\{0, 1/3, 0.5, 2/3, 1\}$.

The dataset-specific allele frequency parameters $\rho_{g,k}$ can also be estimated from the horizontally smoothed BAF matrix to capture more accurate CNV patterns. Specifically, we first estimated the allele frequency for copy neutral state $\rho_{g,k=2}$ from reference cells by directly using its gene_bin specific mean allele frequency. If there are too few reference cells to accurately estimate the allele frequency, it can be set to 0.5 by default. Next, to estimate the allele frequency of other BAF states $k = \{0, 1, 3, 4\}$, we applied a Gaussian mixture model to the smoothed BAF matrix to identify five components with mean values, in order, for allele A bias (strong, minor), allele balance and allele B bias (minor, strong) as an empirical estimation of the CNV ratio for the BAF model. XClone also provides another mode of 3 BAF states (allele A bias, allele balance and allele B bias) with the corresponding theoretical allele frequencies $\{0, 0.5, 1\}$ visualization for comparison.

When the parameters are optimized above, similar to the RDR module, we can also obtain a cell-by-gene_bin-by-state likelihood tensor by normalising Eq. (4), as follows,

$$p(z_{c,g,k} = 1 | a_{c,g}, d_{c,g}, \boldsymbol{\rho}_{g,\cdot}, \boldsymbol{\tau}_{g,\cdot}) = \frac{\texttt{BB}(a_{c,g} | d_{c,g}, \rho_{g,k}, \tau_{g,k})}{\sum_k \texttt{BB}(a_{c,g} | d_{c,g}, \rho_{g,k}, \tau_{g,k})}. \tag{5}$$

3.7 Post-analysis: CNV states smoothing, combination and denoise

Instead of directly using Eq. 3 and Eq. 5 for CNV states assignment at each gene (or gene_bin) in each cell respectively in RDR and BAF modules, we further introduced a smoothing strategy for the state's assignment. Specifically, for each module separately, we first performed a vertical smoothing via a KNN connectivity matrix, as used in Section 3.4, and then we treated it as the input emission probability matrix for horizontal smoothing by a hidden Markov model (HMM). For HMM, the start probability and transition probability can be customised to capture different resolutions of smoothing. Here, we set start probability $\{0.1, 0.8, 0.1\}$ for the RDR module and $\{0.2, 0.15, 0.3, 0.15, 0.2\}$ for the BAF module, and transition probability ($\{t = 1e-6,\ 1-(K-1)t\}$ respectively for cross-state transition and state keeping) as default. Then, we employed a forward-backward algorithm in HMM to achieve a maximum likelihood for CNV assignment probabilities along genomic coordinates.

To combine the CNV states of RDR and BAF modules, we first need to synchronise feature dimensions, as XClone's RDR module outputs a cell-by-gene-by-state CNV posterior probability tensor, while BAF has a shape of cell-by-gene_bin-by-state. The default way is to map BAF gene_bin scale back to the gene scale according to the RDR module. Then, we can combine the same dimensional RDR and BAF CNV posterior probability tensors following the combination strategy in Supp. Fig. S2 to get the final combined CNV posterior probability tensor.

After XClone detects the CNV states for each gene in each cell, a post-denoise step can be optionally applied to clean the regions with a low chance of carrying CNVs. Specifically, for each gene, we calculated the proportion of aneuploid cells and then fit a two-component Gaussian mixture model over the aneuploid cell proportion among all genes. Then the genes assigned to the bottom component (i.e., with the lowest proportion of aneuploid cells) will be taken as neural CNV regions, hence will be masked for CNV analysis. Alternatively, users can manually set a threshold as cell proportion to denoise the probability matrix if there is other evidence. We apply the post-denoise strategy default on the BAF module and it also can be applied on the RDR and combined module by manually setting it.

3.8 Data generation, download, preprocessing and analysis

The scRNA-seq data for a fresh gastric cancer tissue sample GX109-T1c was generated by 10x Genomics Chromium 5'-scRNA-seq platform at the Centre for PanorOmic Sciences (CPOS) at HKU. The raw FASTQ

files were aligned to the human genome (GRCh38) via Cell Ranger v2.2.0. In total, 5,400 cells were called with the same pipeline described in [20] and were analysed with Seurat by selecting the top 2,000 highly variable genes and using 50 top principal components, followed by Leiden clustering with resolution 0.1, resulting in 7 clusters (Supp. Fig. S4). We further manually annotated these clusters according to the cell type marker genes in Supplementary table S2 from [21], and aggregated them into 4 major different cell types: endothelium/fibroblast (endothelium or fibroblast cell), epithelium, immune cells, and stem/cancer cell (gastric stem or cancer cell), respectively with cell counts of 63, 20, 3164 and 2153.

The BCH869 data generated using smart-seq2 was downloaded in FASTQ files from GEO (GSE102130) by request from the authors. Within the original BCH869 dataset, there are 489 cells, with four tumor subclones composed of 227, 221, 34 and 7 cells, according to the original study[14], 3 normal cells (Oligodendrocytes and immune cells) from the same sample were combined in the dataset as reference cells in our study.

TNBC1 is a triple-negative breast cancer sample with 1097 cells (2 subclones in tumor cells)[9]. The TNBC1 data was generated using 3' 10x Genomics platform by Gao et al [9] and the aligned reads in BAM format and the called cell list were directly downloaded from GEO (GSE148673). For more details on the 3 datasets, see Supp. Table S1.

Given the aligned BAM file(s) and cell lists, we performed the genotyping by using cellsnp-lite v1.2.0 [13] (except that GX109-T1c was genotyped on scDNA-seq data) and genotype phasing by using the Haplotype Reference Consortium panel [18] with EAGLE2[17], powered by Sanger Imputation Server (`https://imputation.sanger.ac.uk`). Then, feature counting was achieved by using xcltk v0.1.15 (see above).

3.9 Benchmarking CNV detection performance

It is necessary to benchmark the CNV inference methods using gold standard annotation as ground truth. The CNV profile of the GX109-T1c sample was well characterised by scDNA-seq and WES, which is used as ground truth to identify copy number loss, copy number gain, and loss of heterozygosity (Supp. Methods and Supp. Table S2). The BCH869 dataset has a more complex clonal annotation (4 subclones with different CNV events) where the clone-specific CNV annotation was determined from the combination of scRNA-seq data (by InferCNV) and bulk WGS with haplotype analysis (Supp. Table S3).

With these two datasets, we benchmarked XClone with four other CNVs detection tools specifically designed for scRNA-seq data, InferCNV[22], CaSpER[11], CopyKAT[9] and Numbat[12] (XClone v.0.3.4, InferCNV v.1.8.0, CopyKAT v.1.0.4, CaSpER v.0.2.0 and Numbat v.1.2.1) using their default parameters (details in Supp. Methods and running efficiency in Supp. Table S5).

For CNV detection performance evaluation at the single-cell level, we built a consensus cell-by-gene signal matrix for each tool at each CNV state. For XClone and Numbat, CNV event posterior probability was used to build a cell-by-gene matrix, and for InferCNV, CaSpER and CopyKAT, the modified expression was used. For Numbat and CaSpER whose output is at genomic region resolution, we mapped them to the gene level. The ground truth matrix is a same-dimensional cell-by-gene binary matrix indicating the existence of CNV events in the consensus cells and genes obtained above. The ground truth matrices are prepared for each state (copy gain, copy loss, and LOH) independently. Ground truth matrix value 1 indicates the CNV exists and 0 means not. By flattening the matrix and varying the cutoff on signal scores, we calculated the true positive rate (TPR) and false positive rate (FPR) between the detected (cell-by-gene signal matrix) and real CNV regions (cell-by-gene binary ground truth matrix) for each tool, further generated a receiver operating characteristic (ROC) curve and calculated the area under the ROC curve (AUC) values for evaluation. It is worth mentioning that for fair benchmarking with different tools, XClone can output CNVs in two resolutions: 1) gene_scale, where the CNV probabilities are reported for each gene in each cell; 2) chromosome arm_scale, where the CNV probabilities are reported for each arm of chromosomes in each cell. We showed the gene scale benchmarking results in Fig. **1**h-j for GX109-T1c and **2**f-h for BCH869 and the arm scale benchmarking results in Supp. Fig. S7 for GX109-T1c.

For the TNBC1 sample, as a high-quality annotation is lacking and it implies complex CNV profiles, we only used it in a qualitative manner by examining the consistency between the smoothed BAF signal (without inference) and the inferred CNV states. In this sample, the original paper reports two major subclones that comprised 44% (subclone A) and 28% (subclone B) of the tumour mass within the identified tumour cells

[9]. By Combining expression with BAF analysis, we annotated the cell clusters into Clone 1, Clone 2 and Normal cells, which we used as cell annotation for assessing the clonal reconstruction performance for XClone by calculating the Adjusted Rand Index (ARI) metric (0.9653; Fig. **E1**e).

Data availability

The GX109-T1c scRNA-seq data (5' 10x Genomics) is included in the XClone package (count matrices) and the raw data will be available on GEO shortly. The BCH869 scRNA-seq data (Smart-seq2) raw FASTQ files were downloaded from GEO (GSE102130). The TNBC1 scRNA-seq data (3' 10x Genomics) BAM files were downloaded from GEO (GSE148673).

Codes availability

XClone is implemented as an open-source Python package, publicly available at `https://github.com/single-cell-genetics/XClone`. XClone documentation and tutorials are publicly available at `https://xclone-cnv.readthedocs.io/en/latest/`. The preprocessing toolkit xcltk is freely available at `https://pypi.org/project/xcltk/`. The scripts for benchmarking can be found at `https://github.com/Rongtingting/CNV_calling_Benchmark`.

Contributions

Y.H. and O.S. conceived the study. R.H. and Y.H. designed the statistical model. R.H. led the implementation of the XClone software. R.H. performed all data analysis with help from X.H. X.H. implemented the xcltk preprocessing pipeline. X.H. and R.H. performed the benchmarking part. Y.T. performed the cell annotation for GX109-T1c scRNA-seq data. H.H.Y. and S.Y.L. contributed to the GX109-T1c data and biological insights. S.Y.L. contributed to funding acquisition. R.H. and Y.H. wrote the manuscript. All authors provided feedback on and approved the paper.

Acknowledgements

We thank Lam Yin Hei for helping benchmark copy number variations detection methods and Slava Ivanov for model exploration at the early stage. We thank the Genomics Core and the Imaging and Flow cytometry Core of the Centre for PanorOmic Sciences (CPOS), LKS Faculty of Medicine, The University of Hong Kong for providing single-cell genomic analysis and flow cytometry services. This work was partially supported by a donation from the Hong Kong Jockey Club Charities Trust and a theme-based research grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. T12-710/16-R).

References

1. Zack, T. I. *et al.* Pan-cancer patterns of somatic copy number alteration. *Nature genetics* **45**, 1134–1140 (2013).
2. Steele, C. D. *et al.* Signatures of copy number alterations in human cancer. *Nature* **606**, 984–991 (2022).
3. Nowakowska, B. Clinical interpretation of copy number variants in the human genome. *Journal of applied genetics* **58**, 449–457 (2017).
4. Pan-cancer analysis of whole genomes. *Nature* **578**, 82–93 (2020).
5. Minussi, D. C. *et al.* Breast tumours maintain a reservoir of subclonal diversity during expansion. *Nature* **592**, 302–308 (2021).
6. Zaccaria, S. & Raphael, B. J. Characterizing allele-and haplotype-specific copy numbers in single cells with chisel. *Nature biotechnology* 1–8 (2020).
7. Salehi, S. *et al.* Clonal fitness inferred from time-series modelling of single-cell cancer genomes. *Nature* **595**, 585–590 (2021).

8. Patel, A. P. *et al.* Single-cell rna-seq highlights intratumoral heterogeneity in primary glioblastoma. *Science* **344**, 1396–1401 (2014).
9. Gao, R. *et al.* Delineating copy number and clonal substructure in human tumors from single-cell transcriptomes. *Nature biotechnology* **39**, 599–608 (2021).
10. Fan, J. *et al.* Linking transcriptional and genetic tumor heterogeneity through allele analysis of single-cell rna-seq data. *Genome research* **28**, 1217–1227 (2018).
11. Serin Harmanci, A., Harmanci, A. O. & Zhou, X. Casper identifies and visualizes cnv events by integrative analysis of single-cell or bulk rna-sequencing data. *Nature communications* **11**, 1–16 (2020).
12. Gao, T. *et al.* Haplotype-aware analysis of somatic copy number variations from single-cell transcriptomes. *Nature Biotechnology* 1–10 (2022).
13. Huang, X. & Huang, Y. Cellsnp-lite: an efficient tool for genotyping single cells. *Bioinformatics* **37**, 4569–4571 (2021).
14. Filbin, M. G. *et al.* Developmental and oncogenic programs in h3k27m gliomas dissected by single-cell rna-seq. *Science* **360**, 331–335 (2018).
15. Miller, T. E. *et al.* Mitochondrial variant enrichment from high-throughput single-cell RNA sequencing resolves clonal populations. *Nature Biotechnology* 1–5 (2022).
16. Kwok, A. W. C. *et al.* MQuad enables clonal substructure discovery using single cell mitochondrial variants. *Nature communications* **13**, 1–10 (2022).
17. Loh, P.-R. *et al.* Reference-based phasing using the haplotype reference consortium panel. *Nature genetics* **48**, 1443–1448 (2016).
18. McCarthy, S. *et al.* A reference panel of 64,976 haplotypes for genotype imputation. *Nature genetics* **48**, 1279–1283 (2016).
19. Seabold, S. & Perktold, J. statsmodels: Econometric and statistical modeling with python. In *9th Python in Science Conference* (2010).
20. Yan, H. H. *et al.* Organoid cultures of early-onset colorectal cancers reveal distinct and rare genetic profiles. *Gut* **69**, 2165–2179 (2020).
21. Sathe, A. *et al.* Single-cell genomic characterization reveals the cellular reprogramming of the gastric tumor microenvironment. *Clinical Cancer Research* **26**, 2640–2653 (2020).
22. Tirosh, I. *et al.* Dissecting the multicellular ecosystem of metastatic melanoma by single-cell rna-seq. *Science* **352**, 189–196 (2016).

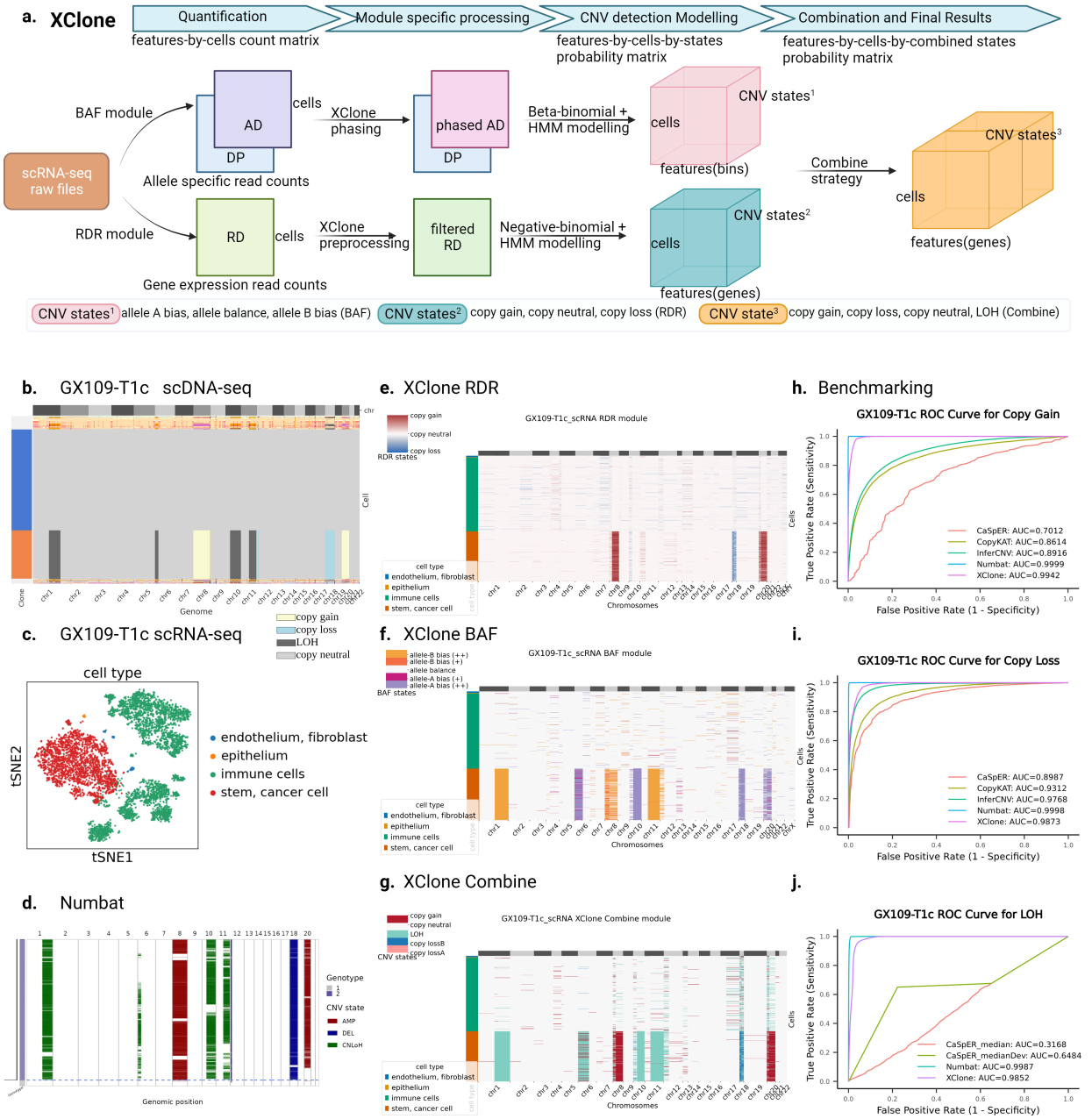


Figure 1: Overview of XClone and benchmarking of CNVs identification on GX109-T1c. (a) Overview of XClone workflow. The XClone workflow begins with scRNA-seq raw files to generate allele-specific read counts matrices and gene expression read counts matrix (features by their genomic positions and cells by assigned cell barcode order). XClone performs the BAF module and RDR module independently and combines them to call the final CNV states. (b) CHISEL analysis of cells derived from a gastric cancer tissue GX109-T1c (scDNA-seq from 10x Genomics) depicts allele-specific copy number changes of the tumour subpopulation in specific chromosomal regions. (c) TSNE of GX109-T1c scRNA-seq data delineates 4 cell types, endothelium or fibroblast, epithelium, immune cells, and stem, cancer cell. (d) Heatmap generated by Numbat using immune cells as reference for GX109-T1c. (e-g) Heatmap generated by XClone RDR module, BAF module and combination of the 2 modules, respectively using immune cells as reference for GX109-T1c. (h-j) Assessment of performance in identification of copy number gain, copy number loss and loss of heterozygosity (LOH) on GX109-T1c.

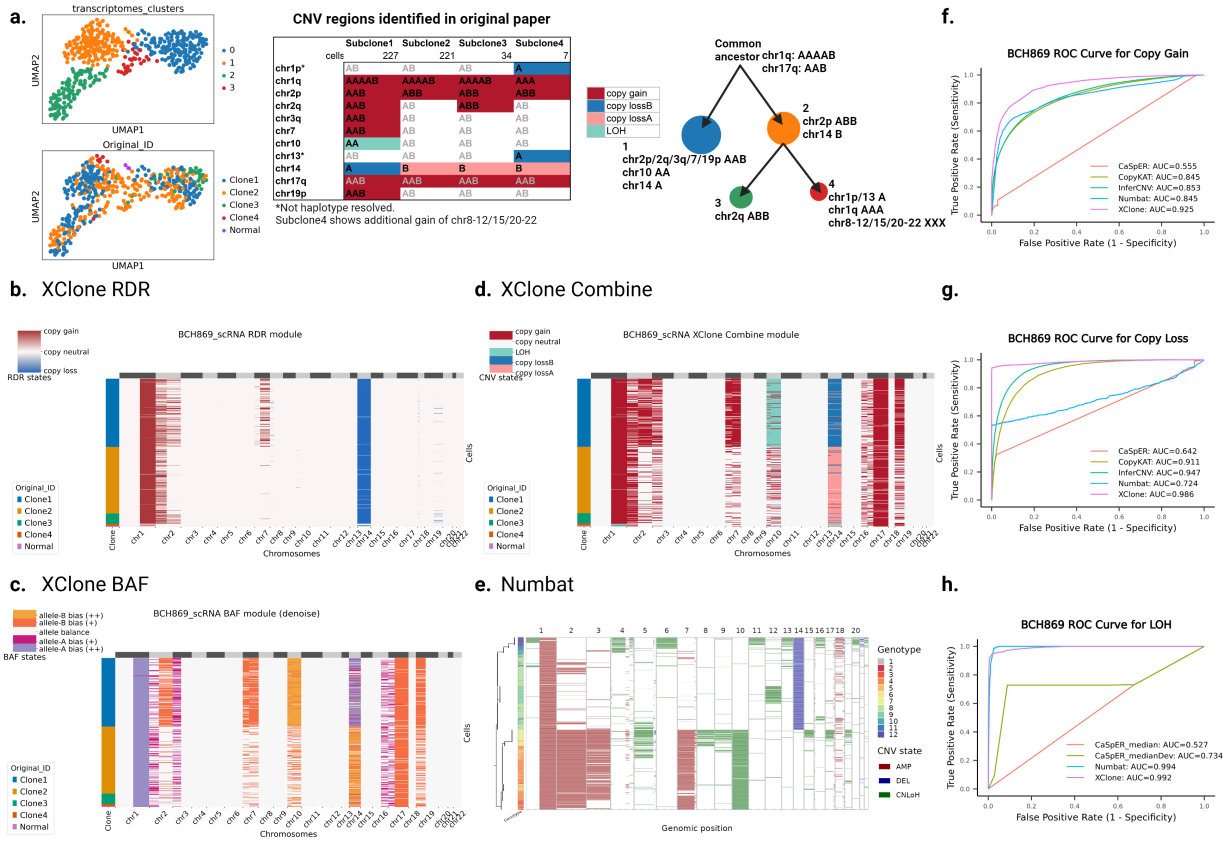
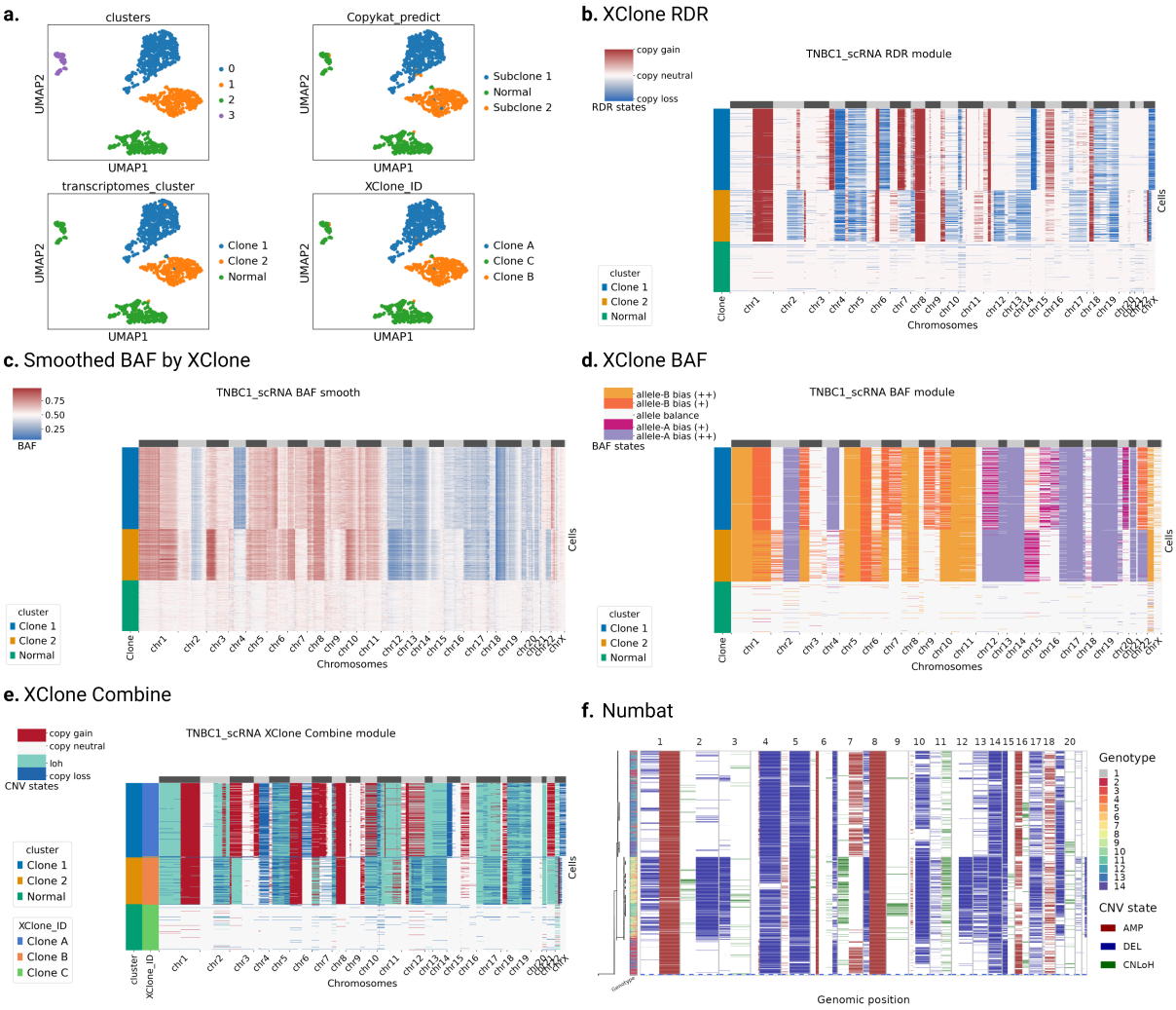


Figure 2: BCH869 CNVs identification by XClone and performance comparison between methods. (a) UMAP of BCH869 with different types of cell annotation: cell clustering by transcriptomes and CNV tumor clone ID identified by the original paper, CNV region identified for each clone in the original paper and the inference clonal tree. (b-d) Heatmap generated by XClone RDR module, BAF module and combination of the 2 modules, respectively for BCH869. (e) Heatmap generated by Numbat. (f-h) Assessment of performance in identification of copy number gain, copy number loss and loss of heterozygosity on BCH869.



Extended Data Figure E1: TNBC1 CNVs identification by XClone and subclonal detection comparison with CopyKAT. (a) UMAP of TNBC1 with different types of cell annotation. Transcriptomes cluster here is the annotation of cell clusters from expression and allele frequency combined analysis, which is used as annotation in analysis and visualization.(b-e) Heatmap generated by XClone RDR module, BAF smoothing, BAF module and combination of the 2 modules, respectively for TNBC1, here the combined result shows that XClone detects similar subclones. (f) Heatmap generated by Numbat.