Title

SCONCE2: jointly inferring single cell copy number profiles and tumor evolutionary distances

Journal

BMC Bioinformatics, 23(1)

ISSN

1471-2105

Authors

Hui, Sandra

Nielsen, Rasmus

Publication Date

2022

DOI

10.1186/s12859-022-04890-w

Peer reviewed

BMC Bioinformatics

RESEARCH ARTICLE

Open Access

SCONCE2: jointly inferring single cell copy number profiles and tumor evolutionary distances

Sandra Hui[1*] and Rasmus Nielsen[1,2,3]

*Correspondence:
sandra_hui@berkeley.edu

[1] Center for Computational Biology, University of California, Berkeley, Berkeley 94720, USA
[2] Department of Integrative Biology, University of California, Berkeley, Berkeley 94720, USA
[3] Department of Statistics, University of California, Berkeley, Berkeley 94720, USA

Abstract

Background: Single cell whole genome tumor sequencing can yield novel insights into the evolutionary history of somatic copy number alterations. Existing single cell copy number calling methods do not explicitly model the shared evolutionary process of multiple cells, and generally analyze cells independently. Additionally, existing methods for estimating tumor cell phylogenies using copy number profiles are sensitive to profile estimation errors.

Results: We present SCONCE2, a method for jointly calling copy number alterations and estimating pairwise distances for single cell sequencing data. Using simulations, we show that SCONCE2 has higher accuracy in copy number calling and phylogeny estimation than competing methods. We apply SCONCE2 to previously published single cell sequencing data to illustrate the utility of the method.

Conclusions: SCONCE2 jointly estimates copy number profiles and a distance metric for inferring tumor phylogenies in single cell whole genome tumor sequencing across multiple cells, enabling deeper understandings of tumor evolution.

Keywords: Copy number alterations, Cancer genomics, Single cell sequencing, Tumor evolution, Tumor phylogenies

Background

Cancer evolution is driven by an accumulation of somatic point mutations and large copy number alterations (CNAs) [1, 2]. For example, CNAs can affect the transcriptional landscape via dosage effects [3], and identifying intra tumor heterogeneity and cell specific changes in gene expression is clinically relevant. In particular, recent studies have shown quantifying these transcriptional changes, by measuring tumor specific total mRNA expression, is predictive of disease prognosis and progression across multiple cancer types [4]. In this manuscript, we focus on estimating the underlying copy number alterations using whole genome sequencing, in order to directly study the evolutionary process.

Single cell sequencing can offer a detailed picture of the process of CNA and mutation accumulation that is lost in bulk sequencing, in particular by estimating the

phylogenetic relationship among different cell types. A challenge in such efforts is that single cell sequencing data is typically very noisy due to variable and low sequencing depth [5], making accurate genotyping, copy number (CN) calling, and phylogeny estimation difficult. However, as we will show here, by leveraging the shared evolutionary history among cells, jointly calling CNAs across cells can lead to increased accuracy and give information about the evolutionary relationship between cells, thereby leading to improved estimates of tumor phylogenies. Different cells from the same tumor share some of their somatic evolutionary history, and information regarding CNAs, and CNA breakpoints, from one cell can, therefore, inform CNA calling in other cells.

Unfortunately, the commonly used methods for estimating single cell copy number profiles (CNPs), the collection of copy number states across the genome, do not rigorously use this shared information. Instead, most methods, including SCONCE [6] and the commonly-used AneuFinder [7, 8], independently call CNPs. Although SCONCE [6], a copy number calling method for single cell tumor data, was previously shown to outperform competing methods in absolute copy number and breakpoint detection accuracy [6], it does not utilize any information from shared evolutionary histories between cells. Other methods, such as CopyNumber [9] and SCIcONE [10], jointly call CNPs by forcing breakpoints to be shared across all cells. However, we showed in previous work that SCONCE [6] outperforms these methods as well, despite not analyzing cells jointly. In contrast, WaveDec [11], a method designed to detect shared and cell specific copy number events in copy number arrays and applied to a subset of sequencing data from [12], takes an orthogonal approach by transforming $\log_2$-ratio copy number data into the wavelet space. This transformation allows separation of common/shared and individual CNAs, as shared events are captured by the approximation coefficients and individual events are described by the detail coefficients.

Despite these limitations in copy number calling, several distance metrics for copy number profiles have been developed for estimating tumor phylogenies using algorithms such as neighbor-joining [13, 14]. Commonly used pairwise distance metrics include the Euclidean distance [12, 15], the MEDICC distance described by [16], and the cnp2cnp distance presented by [17]. Although the Euclidean distance is easy to calculate, large and/or overlapping CNAs can artificially inflate this measure, leading to overestimation of dissimilarity. The latter two methods measure distance between two CNPs by attempting to find the minimum number of deletion and amplification events needed to transform one CNP into the other, without allowing regions that are lost to be regained. The MEDICC model is limited to maximum copy number 4 and events that increase or decrease copy number by one, while the cnp2cnp metric relaxes both of these constraints. Cordonnier et al. [17] re-implemented the MEDICC algorithm to allow copy numbers greater than 4, and showed that while both the cnp2cnp and MEDICC distances outperform the Euclidean distance for the purpose of phylogeny estimation, cnp2cnp is more accurate on error free data and MEDICC is more accurate on data with errors.

However, none of these methods use explicit evolutionary models of CNAs to provide joint estimates of CNPs and evolutionary distance. Here, we present SCONCE2, an expansion on SCONCE, that further develops SCONCE's underlying tumor evolutionary model to jointly model the CNA process in two cells. SCONCE2 takes

advantage of the shared evolutionary history between cells, and produces more accurate single cell CNP estimates and pairwise estimates of the evolutionary distances between cells, by combining information across multiple cells. We show that SCONCE2 estimates more accurate CNPs and tumor phylogenies than competing methods using extensive simulations, and apply it to previously published data from [12, 18] to illustrate its utility.

Results

To infer the evolutionary history of tumor cells, SCONCE2 models the evolution of pairs of cells. We assume a pair of cells, (A, B), have a partially shared evolutionary history originating from a healthy ancestral diploid cell, D. The shared part of their evolutionary history is represented in a tree, $\mathcal{T} = [t_1, t_2, t_3]$, by a branch of length t_1, running from an non-tumor diploid cell (D) to an unobserved divergence point, Z. From Z, cells A and B evolve independently, with branch lengths t_2 and t_3, respectively (see Fig. 1). A core goal is to estimate this tree and to distinguish between shared evolutionary events and independent cell specific events.

Because the number of pairs of cells grows quadratically with the number of cells, n, full joint maximum likelihood estimation of all parameters can become computationally challenging. We, therefore, first run SCONCE on all cells independently to obtain cell specific estimates of model parameters. We then take the median of the estimates of evolutionary parameters $\{\alpha, \beta, \gamma\}$, corresponding to the rates of different types of copy number events (see One cell continuous time Markov process), to combine the disjoint SCONCE estimates into summary estimates across all cells. Then, for each pair of cells, we estimate branch lengths of tree $\mathcal{T} = [t_1, t_2, t_3]$ using maximum likelihood, and use the Viterbi algorithm to calculate paired decoded copy number profiles. Because each cell appears in $n - 1$ pairs, this produces $n - 1$ paired CNP estimates per cell. Finally, for each cell, we take the per window mean across each cell's $n - 1$ paired CNP estimates to calculate consensus CNPs. This pipeline is described in Detailed SCONCE2 pipeline and illustrated in Fig. 2.

By analyzing each cell in the context of multiple pairs, we obtain increased accuracy in copy number calls and breakpoint detection, as well as usable tree branch length estimates. We examine the properties of these estimates on both simulated and real data.

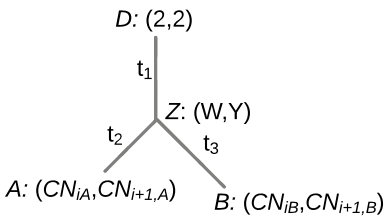

Fig. 1 Pairwise tree structure, showing the tree, $\mathcal{T} = [t_1, t_2, t_3]$, between the pair of cells A and B, where the branch with length t_1 represents their shared evolutionary history from an ancestral diploid cell, D, before diverging at the unobserved state Z. The branches with lengths t_2 and t_3 show independent evolution to cells A and B, respectively. Copy number in adjacent bins along the genome is shown in parentheses

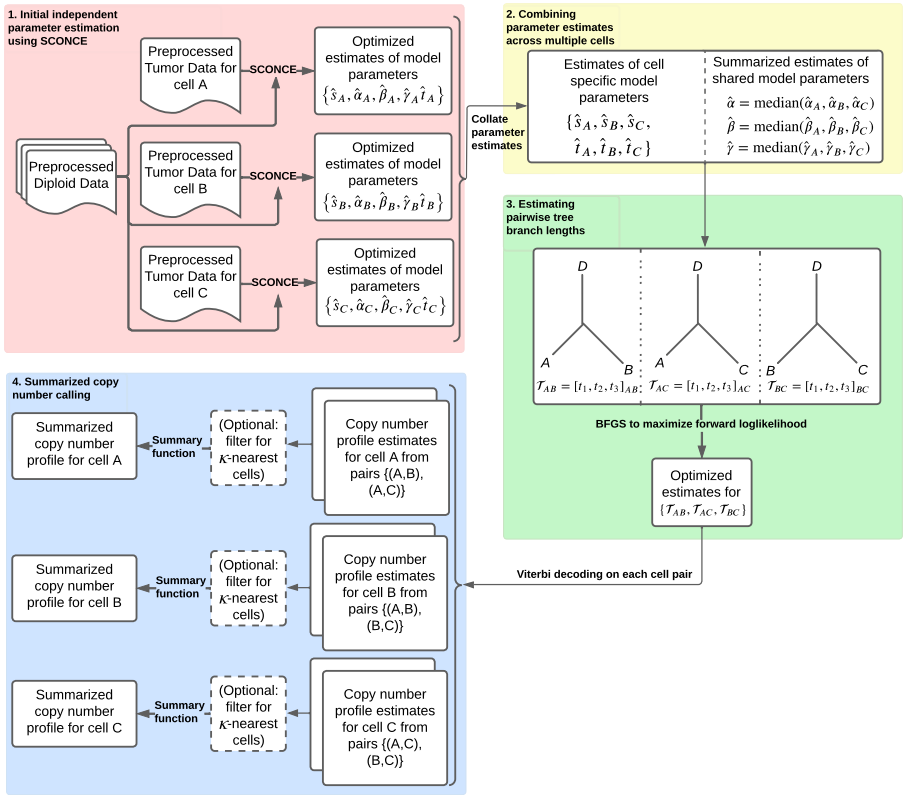

Fig. 2 Detailed flowchart of the SCONCE2 pipeline. We demonstrate the pipeline with cell triplet $\{A, B, C\}$, without loss of generality. Each tumor cell is initially independently analyzed through SCONCE, which gives parameter estimates and copy number profiles for each cell (red box). These parameter estimates are then summarized (yellow box), and branch lengths for tree $\mathcal{T} = [t_1, t_2, t_3]$ are estimated for each pair of cells (green box). Finally, for each cell, paired copy number profiles are summarized into a consensus copy number profile (blue box). Each step is fully described in Detailed SCONCE2 pipeline

Simulations

In order to rigorously test SCONCE2, we applied it to four simulated datasets and two real datasets, from [12, 18]. We simulated 128 cells on four different tree structures: tree A) is maximally imbalanced and ultrametric, tree B) is perfectly balanced and ultrametric, tree C) is maximally imbalanced and not ultrametric, where internal and terminal branches have uniform length, and tree D) is maximally imbalanced and not ultrametric, where internal branches have equal length and terminal branch lengths decay logarithmically (tree structures shown in Additional file 1: Fig. S1). Simulated cells from each tree structure were divided into five discrete test subsets of 20 cells each.

Briefly, the simulated genome is modeled as a collection of line segments, where amplifications and deletions occur according to a Markov process and have lengths sampled from a truncated exponential distribution. Copy number events occur within the tree structure, such that ancestral CNAs are propagated to descendent cells. Note, the simulation model is more biologically realistic and intentionally structured to be substantially different from the SCONCE2 inference model, in order to avoid biasing accuracy results to favor our method. We previously described this simulation model in [6], and full simulation details are given in Simulations.

Copy number and breakpoint detection accuracy

Sum of squared error on CNPs

To measure copy number accuracy, we calculated the sum of squared errors (SSE) between the inferred copy number and the true simulated copy number across genomic windows for each cell. To evaluate each step in the SCONCE2 pipeline, we calculated the SSE on copy number profiles generated from individual cell estimation (SCONCE), on profiles from each pair of cells (one pair), and on consensus profiles estimated using three different summary statistics (mean, median, mode) across multiple pairs of cells. We also compared to AneuFinder [7, 8], a commonly used method for single cell copy number calling, and the second-most accurate one, after SCONCE, among methods evaluated in previous work [6]. In all subsequent results, we report summary statistics across all subsets for each tree/simulation set. Recall full simulation descriptions are given in Simulations.

In tree A (maximally imbalanced ultrametric tree; Fig. 3A), using pairs of cells had lower SSE than individual cells (SCONCE) alone, with respective median SSE values of 26.01 and 37.31. Furthermore, using the mean had the lowest median SSE of 17.83, with median and mode at 23.68 and 24.04. These SSE values were lower than AneuFinder, which had a median SSE value of 51.78. Similar results for tree B (perfectly balanced ultrametric tree) are shown in Fig. 3B, with median SSE values of 28.37, 21.25, 14.74, 17.90, 18.09, and 41.80

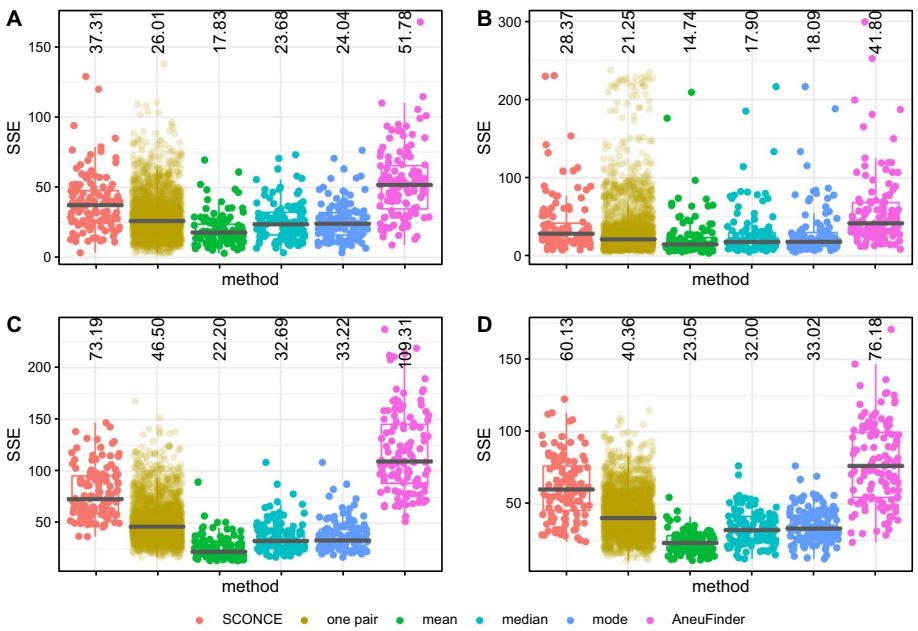

Fig. 3 Boxplots of genome wide sum of squared error (SSE) between true simulated copy number profiles and inferred copy number profiles, across methods. SSE results are shown for cell specific CNPs from SCONCE (independent cell inference); joint inference on each pair of cells (one pair); summary functions across all pairs of cells (mean, median, and mode); and AneuFinder. Different methods are shown across the x-axis, and SSE is shown on the y-axis. Median SSE for each method is printed at the top of each column. Each dot represents one cell (note, in "one pair", each cell appears multiple times), and the median SSE is printed at the top of each column. Panel letters correspond to tree labels **A** (maximally imbalanced ultrametric tree), **B** (perfectly balanced ultrametric tree), **C** (maximally imbalanced non ultrametric tree with uniform branch lengths), and **D** (maximally imbalanced non ultrametric tree with logarithmically decaying branch lengths). SSE results are consistently lower when using data from multiple cells

for individual cells, single pairs, mean, median, mode, and AneuFinder, respectively. In the same order, the median SSE values for tree C (maximally imbalanced with uniform internal branch lengths) were 73.19, 46.50, 22.20, 32.69, 33.22, and 109.31, and the median SSE values for tree D (maximally imbalanced with logarithmically decaying branch lengths) were 60.13, 40.36, 23.05, 32.00, 33.02, and 76.18 (see Fig. 3C, D). Clearly, there is a substantial improvement in accuracy by using pairs of cells instead of individual cells, and this improvement in accuracy is larger if multiple pairs are used.

We note that, as an artifact of the genome binning procedure, true fractional copy numbers may occur from small CNAs completely contained within window boundaries, or from CNAs crossing window boundaries (for example, observing windows with true copy numbers $1 \rightarrow 1.25 \rightarrow 2$). As such, the mean and median have the lowest SSE values because they allow fractional copy numbers. However, many downstream tools expect integer copy number profiles for single cells, so users may wish to round to the nearest integer or use the mode option.

Breakpoint distance and detection

In order to measure breakpoint detection accuracy, we calculated the genome wide distance between inferred and true breakpoints, penalized by the number of total inferred breakpoints. Specifically, for each simulated breakpoint, we calculated the distance to the nearest inferred breakpoint. Because erroneously inferring breakpoints at every position in the genome would artificially lower this genome wide distance, we also calculated $\omega = \frac{\text{\# inferred breakpoints}}{\text{\# true breakpoints}}$, such that lowest breakpoint distances with ω values closest to 1 indicate greatest accuracy.

In all simulation sets, using the mean consistently had ω values closest to 1, again due to fractional copy number states, as well as lower total breakpoint distance than other methods. Across trees, results from AneuFinder, followed by SCONCE, had the highest breakpoint distances and ω values further from 1. For tree A (ultrametric maximally imbalanced tree; Fig. 4A), SCONCE, single pairs, mean, median, mode, and Aneufinder had median distance values of 1167, 1006, 394, 1018.5, 1019.5, and 1172, and median ω values of 0.466, 0.490, 0.921, 0.486, 0.486, and 0.462, respectively. Similarly, for tree D (maximally imbalanced with logarithmically decaying branch lengths, Fig. 4D), median distance values were 153.5, 85, 33, 77, 77.5, and 168.5, and median ω values were 0.504, 0.535, 1.007, 0.535, 0.534, and 0.489, in the same order as above. Full median distance and ω values are given in Additional file 1: Tables S2 and S3. Similarly to the observations for the CNP estimates, breakpoint detection also improves when using pairs of cells, and improves when estimates from multiple pairs are combined, particularly if combining using the mean. As previously noted in Sum of Squared Error on CNPs, binning the genome can result in fractional copy numbers for some bins. Compared to the median and the mode, the mean is better able to capture these fractional copy number states, resulting in lower breakpoint distances and ω values closer to 1.

Optimal number of pairs to use

Because there are $\binom{n}{2}$ pairs for n cells, averaging over more pairs of cells comes at a computational cost. Furthermore, as we will show, adding too many divergent cells

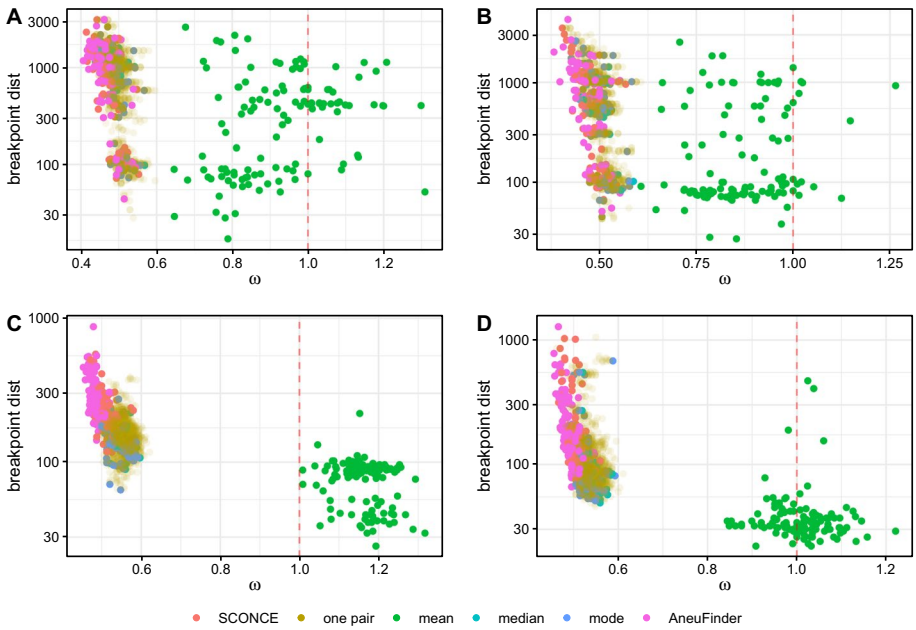

Fig. 4 Breakpoint detection accuracy results across methods. Total distance to nearest breakpoint is shown on the y-axis, and $\omega = \frac{\text{\# inferred breakpoints}}{\text{\# true breakpoints}}$ is shown along the x-axis. Each dot represents one cell, colored by method. Methods with the highest breakpoint detection accuracy cluster near $\omega = 1$ (vertical red dotted line) and have lowest total breakpoint distance. Each panel corresponds to a simulation set (A-D). In all simulation sets, using the mean, median, and mode have lower total breakpoint distance than independent cell analyses. Furthermore, using the mean results in ω values closest to 1, as it is able to infer fractional copy numbers

can reduce the accuracy, as including highly divergent cells in the average may increase the noise.

To determine the optimal number of cells to summarize across, we estimated summarized (mean) copy number profiles with increasing numbers of cells. As each cell was added, we calculated the difference in SSE relative to SCONCE (individual cells). Cells were added in three different orderings: most to least similar (i.e., nearest first, as defined by the Euclidean distance between the cells' SCONCE profiles), least to most similar (furthest first), and randomized order. In tree A, the median pairwise Euclidean distance between SCONCE profiles was 88.0625 for the nearest/most similar cells, 138.9585 for the tenth most similar cells, and 205.853 for the least similar cells. Median pairwise Euclidean distances for all datasets are given in Additional file 1: Table S4.

In Fig. 5, we summarize the change in SSE across κ cells for each tree. Across all trees, SSE improves fastest when adding nearest cells first, and slowest for adding furthest cells first, with the random ordering in between. When adding nearest cells first, the SSE initially sharply decreases, levels off and reaches the largest decrease after approximately 10 cells, and then increases. Specifically, the mean change in SSE when adding nearest cells first reached the greatest decrease in SSE from SCONCE of -20.710, -17.491, -56.185, and -38.570 when $\kappa = 12, 10, 9, 15$ cells for trees A, B, C, and D, respectively. In contrast, when $\kappa = 20$ cells, the change in SSE from SCONCE was -19.721, -16.757, -53.461, and -37.920 for trees A, B, C, and D, consistent with the results shown in Fig. 3.

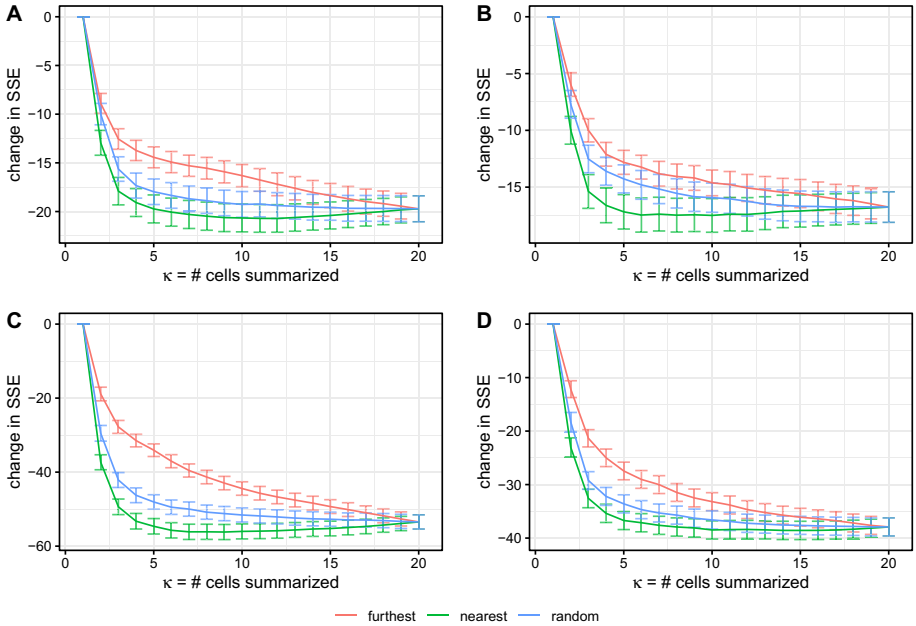

Fig. 5 Change in SSE relative to SCONCE, as more cells are added to the consensus (mean) analysis. Number of cells, κ, in the joint analysis is shown along the x-axis, and change in SSE (relative to SCONCE) is shown on the y-axis. Each line shows the mean change in SSE across cells, with error bars showing ± 1standard deviation. Colors show cell ordering, where furthest denotes adding the least similar cells first (i.e., furthest distance as defined by the Euclidean distance on SCONCE profiles) and nearest denotes adding the most similar cells first. Across all datasets and cell orderings, SSE quickly drops as more cells are added, with adding the nearest cells (green line) showing the fastest improvement. However, for this ordering, the decrease in SSE levels off after approximately 10 cells are added, and then slightly rises as more cells are added, due to rare copy number events getting averaged out

When summarizing over $\kappa < n - 1$ cells, for a given cell, some of the other cells will not be used in that cell's consensus profiles. As a time saving measure, these excluded cell combinations are not analyzed. Therefore, we recommend users summarize over $\kappa = 10$ cells, added in order of most to least similar.

For completeness, SSE and breakpoint detection across parameter sets when summarized over only the nearest 10 cells is shown in Additional files 1: Figures S2, S3, and Tables S5 and S6.

Using multiple cells results in better CNA detection

Plotting true simulated copy number profiles against inferred copy number profiles shows why performance improves when using multiple cells. For example, in Fig. 6, SCONCE erroneously combined two breakpoints for cell A, while predicting cell B's breakpoint too far to the left (column labelled SCONCE). However, when analyzed as a pair, a shared breakpoint was inferred (left arrow), and the second breakpoint (right arrow) in cell A was correctly inferred (one pair column). While the shared breakpoint was closer to the true breakpoint, it was not until CNPs are summarized across multiple cells that the breakpoint was called in the correct position. Using the mean results in slightly fuzzier boundaries due to non-integer copy number calls (middle

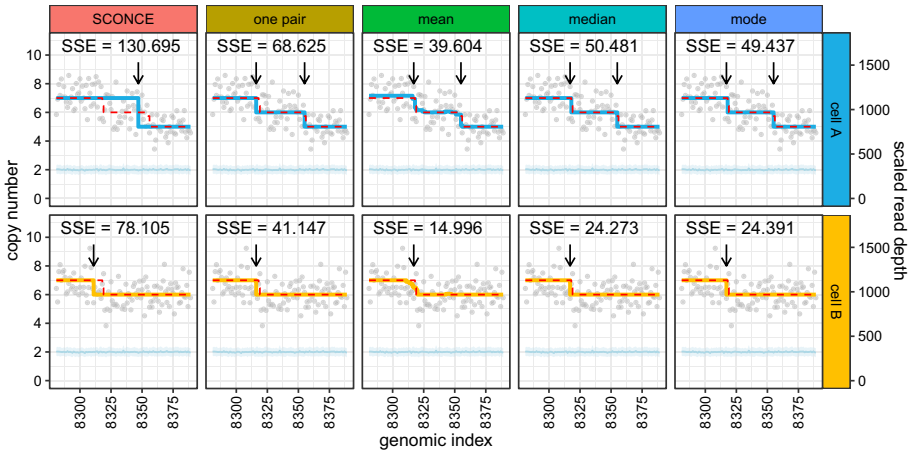

Fig. 6 Jointly calling CNAs and summarizing data across multiple cells results in more accurate boundary detection. Inferred copy number profiles and read depth data are shown for two cells (A: 111, B: 59) simulated from tree C (maximally imbalanced, uniform branch lengths). Genomic index shows 110–250 kb windows along the x-axis, while the y-axis shows copy number (left) and read depth (right). Gray dots show per window read depth, the light blue line and band show the mean and variance of the diploid read depth, the dotted red line shows the true simulated copy number, and the blue and yellow lines show the inferred copy number calls for each cell. Arrows denote inferred breakpoints, and SSE values are listed for each subpanel. Subpanels show results from different copy number calling methods: SCONCE (independent analysis); one pair (analysis of cells A and B as a pair); mean, median, and mode (consensus calls from summarizing paired CNPs for cells A and B across all relevant pairs of cells). Breakpoint detection accuracy increases as more cells are included in the joint analysis

column), which better reflected the true underlying data, while the median and mode (right two columns) result in integer jumps at bin boundaries.

Similar results are observed for real data. For example, in Additional file 1: Figure S4, SCONCE missed the left most CNA in cell B (arrow in SCONCE column). When analyzed as a pair, this CNA was detected, and was shared with cell A (left arrows). Additionally, a short deletion was called in both cells (right arrows). However, this is a rare event, as it was averaged out in the mean, median, and mode analyses (arrows in mean, median, and mode columns). Furthermore, in Additional file 1: Figure S5, SCONCE did not call a CNA in cell A, but did call a -3 deletion in cell B (arrows in SCONCE column). However, when these two cells were jointly analyzed as a pair, there was enough evidence to call a -1 deletion in both. When summarizing across multiple cells, this deletion continued to be supported (right arrow). Additionally, there was some evidence from joint analyses with other cells that an additional small deletion existed in cell B, but not in cell A (left arrow). However, this small deletion was lost when using the median and mode, although the deletion first identified in the joint analysis of cells A and B remained.

Model parameter estimates

For each pair of cells, SCONCE2 estimates the branch lengths for tree $\mathcal{T} = [t_1, t_2, t_3]$ (see Fig. 1). From the simulated trees, the corresponding tree branch lengths and node distances can be extracted for each cell pair. Recall the simulation and inference models are intentionally formulated differently to evaluate SCONCE2 in more realistic settings (see Simulations). Because the scaling of $\mathcal{T}$ is different between the simulation and inference models, we show the R^2 values between true (simulated) and inferred values of

$\mathcal{T} = [t_1, t_2, t_3]$, as well as the summed distance $t_2 + t_3$ as a distance metric between two cells.

For tree A (ultrametric, maximally imbalanced), SCONCE2 recovered $\{t_1, t_2, t_3, t_2 + t_3\}$ values with R^2 values of 0.798, 0.35, 0.286, and 0.551 (Fig. 7A), respectively. Additionally, for tree D (maximally imbalanced with uniform branch lengths), SCONCE2 had R^2 values of 0.661, 0.564, 0.59, and 0.686, for t_1, t_2, t_3, and $t_2 + t_3$. (Fig. 7D).

We note that the sum $t_2 + t3$ has higher R^2 values than those of t_2 or t_3 individually, demonstrating some uncertainty in assigning events to particular branches. Furthermore, because the simulation model generates the number of CNAs from a distribution relating to a Poisson (however, the distribution is not truly Poisson as the size of the genome changes through the simulations), the mean and variance of the number of events increases with branch length in expectation. This increased variance is reflected by the larger range of branch length estimates as branch lengths increase. Nonetheless, as we will show in the next section, SCONCE2 recovers the magnitude of cell relationships sufficiently accurately to allow improved phylogeny estimation.

Phylogeny estimation

Estimating phylogenies on copy number profiles using neighbor-joining [13, 14] requires a distance metric between cells. Existing metrics include the Euclidean distance [12], and two estimates of the minimum number of CNAs needed to transform one CNP into another: the cnp2cnp metric [17] and the MEDICC distance [16] (here, we use the implementation in the cnp2cnp program [17]). These methods require prior estimation of the CNP. See Running other methods for details on running these programs.

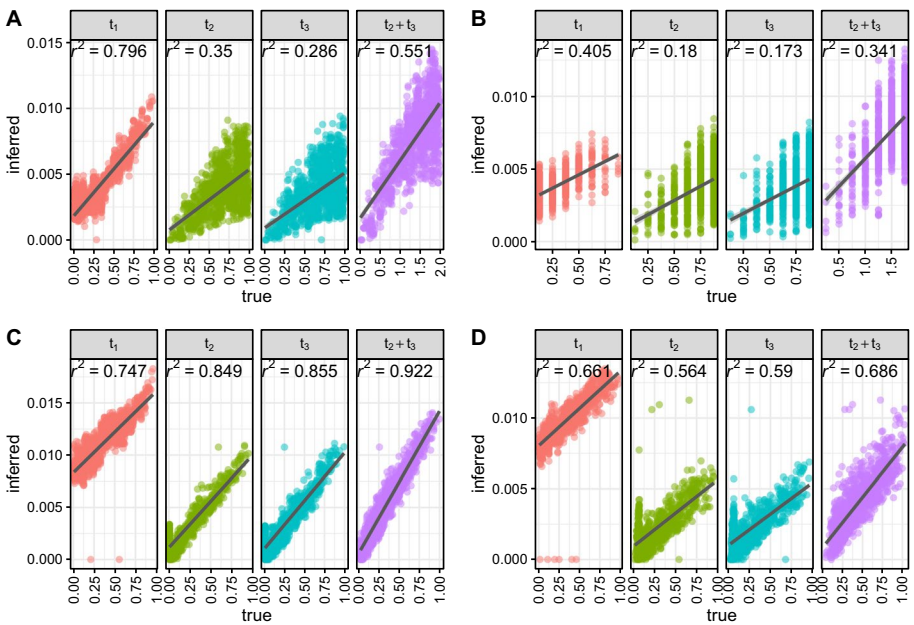

Fig. 7 Correlation between true branch lengths and estimated branch lengths across simulation sets. Each dot represents one pairwise branch length estimate, with true node distance on the x-axis and estimated branch length on the y-axis. R^2 values from a linear regression on branch length (dark gray lines) is shown for each subpanel. Across all simulation scenarios (panels A-D correspond to trees A-D), SCONCE2 consistently predicts t_1 and $t_2 + t_3$ with high R^2 values.

Under the SCONCE2 model, by construction, $t_2 + t_3$ measures the pairwise distance between two cells. To compare these different distance metrics, we first calculated distance matrices using pairwise Euclidean, cnp2cnp, and MEDICC distances on CNPs called by SCONCE (previously showed to be more accurate than other single cell copy number callers [6]), as well pairwise $t_2 + t_3$ estimates. Next, we applied neighbor-joining to estimate phylogenies and computed the Robinson–Foulds (RF) distance [19] between the true trees and the inferred trees. As shown in Fig. 8, across parameter sets, the trees inferred from estimates of $t_2 + t_3$ had lower Robinson–Foulds distances than trees inferred from other distance metrics. For example, for tree A (ultrametric, maximally imbalanced), the median RF distances were 27, 27, 29, and 20 for the Euclidean distance, cnp2cnp distance, MEDICC distance, and $t_2 + t_3$ (Fig. 8), respectively (see Additional file 1: Table S7, for all median Robinson–Foulds distances).

Furthermore, we calculated RF distances from phylogenies based on the Euclidean, cnp2cnp, and MEDICC distances on consensus CNPs and true simulated CNPs (Additional file 1: Figure S6). When summarizing over all pairs of cells, using $t_2 + t_3$ consistently had lower median RF distances than other methods on consensus CNPs. For example, in tree A (ultrametric, maximally imbalanced), the median RF distances for phylogenies estimated from mean consensus profiles were 27, 26, and 28 for the Euclidean, cnp2cnp, and MEDICC distances (Additional file 1: Figure S6A). On distances calculated from the true CNPs, $t_2 + t_3$ performed as well or better than the other metrics, with the exception of the cnp2cnp distance in tree A, where the Euclidean, cnp2cnp, MEDICC distances and $t_2 + t_3$ had respective median RF distances of 27, 19, 20, and 20

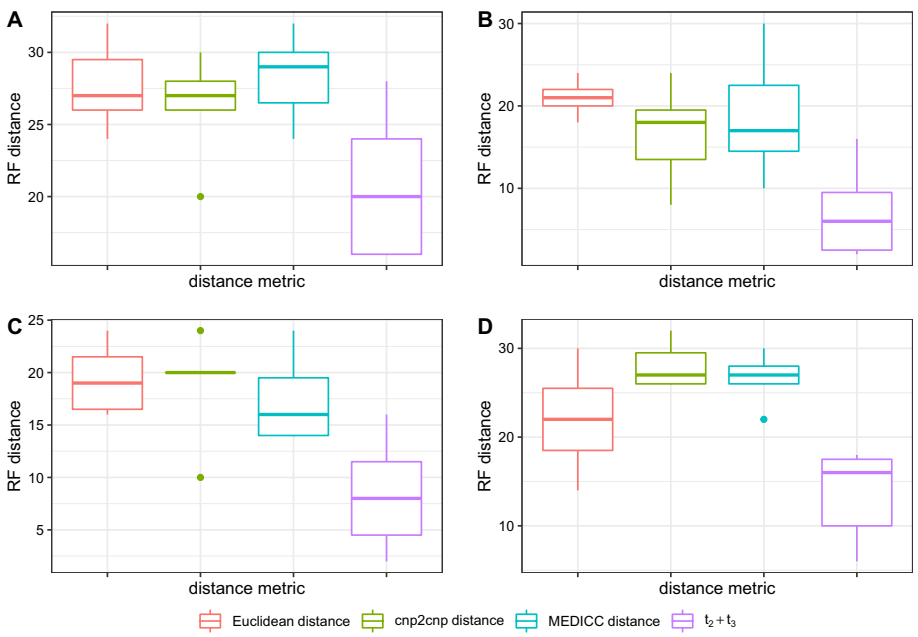

Fig. 8 Robinson–Foulds (RF) distances between true and inferred phylogenies. Phylogenies were estimated using neighbor-joining on $t_2 + t_3$ estimates, and on the Euclidean, cnp2cnp, and MEDICC distances between true CNPs and CNPs inferred from SCONCE. Methods are shown along the x-axis, while Robinson–Foulds distances are shown on the y-axis. Across multiple parameter sets (panels **A**–**D** correspond to trees **A**–**D**), using $t_2 + t_3$ estimates resulted in a lower Robinson–Foulds distance from the simulated tree, relative to all other inferred phylogenies

(Additional file 1: Figure S6A). However, under experimental conditions, the true CNP would be unknown. In all other simulation sets, the phylogenies estimated using $t_2 + t_3$ had lower median Robinson–Foulds distances than all other methods.

For completeness, we additionally calculated Robinson–Foulds distances on phylogenies estimated from consensus CNPs from summarizing over the nearest 10 cells. For tree A (ultrametric, maximally imbalanced), the median RF distances on mean consensus CNPs over the nearest 10 cells were 27, 27, and 26 for the Euclidean, cnp2cnp, and MEDICC distances, respectively (Additional file 1: Figure S7A). Note that in order to estimate phylogenies using our $t_2 + t_3$ metric, all pairs of cells must be analyzed, and cannot benefit from the time savings of analyzing a selected subset of pairs of cells (see Optimal number of pairs to use). This is a weakness of our method, as analyzing all pairs of cells comes at an increased computational cost.

Discussion

We present a novel method, SCONCE2, that combines data across single cells in a manner that is grounded in a principled model of stochastic tumor evolution. It jointly calls copy number alterations in single cell sequencing of cancer cells with higher accuracy than competing methods on both simulated and real data. Additionally, SCONCE2 calculates an informative pairwise distance metric that can be used to estimate phylogenies with less error than other methods.

Similar to SCONCE, one weakness of SCONCE2 is the requirement for matched diploid cells in order to normalize GC content and mappability biases. These diploid cells must be sequenced under the same experimental conditions for proper GC content and mappability normalization, and may not be directly of interest to investigators, thereby potentially increasing cost. However, infiltrating diploid cells are often sequenced as a byproduct of single cell sequencing, and can be identified with orthogonal methods, such as cell sorting. For example, in the two datasets analyzed here, no additional sequencing was necessary to purposefully produce matched diploid sequencing data.

Additionally, SCONCE2 does not use SNPs or genotype likelihoods, or do any allelic phasing, to inform copy number calls or $t_2 + t_3$ estimates. Although calling SNPs in low coverage and noisy single cell data is difficult, incorporating genotype likelihoods can add information and increase confidence in these procedures. For example, using the allele frequency in variable single nucleotide sites can support concordant or rule out discordant copy number states. Furthermore, estimating the counts of variable sites on specific branches in $\mathcal{T} = [t_1, t_2, t_3]$ (see Fig. 1) can increase confidence in branch length estimates. Adding genotype likelihoods of single nucleotide variants is the subject of future work.

Another weakness of SCONCE2 is that it takes longer to run, relative to other methods. However, if investigators are primarily interested in copy number calling, significant time can be saved by summarizing over a selective subset of pairs of cells (that is, noninformative pairs are not analyzed), described in Optimal number of pairs to use. But, if investigators are interested in estimating phylogenies using our $t_2 + t_3$ metric, all pairwise distances must be estimated to calculate a complete distance matrix (described in Phylogeny Estimation), thereby negating this time saving measure. Because the distance matrix dimensions and number of pairwise comparisons grow quadratically with the number of cells analyzed,

the computational complexity and run time cost grows quickly. However, if investigators are interested in both copy number calling and phylogeny estimation using our $t_2 + t_3$ metric, after all pairwise parameter estimates are calculated (the most computationally intensive step), investigators have the flexibility to quickly call consensus copy number profiles over an arbitrary number of pairs. Despite the computational complexity of this model, we propose the increased accuracy of both copy number calls and phylogeny estimation outweighs the increased computational run time cost.

Conclusions

In conclusion, we present a principled method, SCONCE2, for simultaneously and accurately calling and aggregating copy number profiles across multiple tumor cells, and estimating pairwise evolutionary distances, using single cell whole genome sequencing. This work shows jointly analyzing cells in single cell experiments to leverage their shared evolutionary history increases accuracy in copy number calling and phylogeny estimation, with implications for deepening our understanding of tumor evolution.

Methods

Evolutionary process modeling

We first review the Markov processes introduced in [6]. Briefly, we assume an evolutionary process that is continuous in time but discrete along the length of the genome. However, notice that this is just an approximation, as the true process along the length of the genome is not Markovian (see Simulations).

One cell continuous time Markov process

The one cell continuous time process from [6] models the copy numbers of two adjacent genomic bins, in positions i and $i + 1$ in the genome, on the same lineage (cell) with copy number $U, V \in \mathbb{S}_c = \{0, 1, \ldots, k\}$, respectively, where k is the maximum allowed copy number. We assume that (U, V) evolve through time with the following rate parameters:

$$\alpha = \text{rate of } \pm 1 \text{ CNA} \tag{1a}$$

$$\beta = \text{rate of any CNA} \tag{1b}$$

$$\gamma = \text{relative rate of CNAs affecting both } U \text{ and } V \tag{1c}$$

which leads to the following instantaneous rate matrix for the joint process for two bins on one lineage: $\mathbb{Q} = \{q_{(U,V),(U',V')}\}$:

$$q_{(U,V),(U',V')} = \begin{cases} \gamma(\alpha+\beta) & \text{if } (U',V') = \begin{cases} (U+n,V+n) \\ (U-n,V-n) \end{cases}, n=1 \\ \gamma\beta & \text{if } (U',V') = \begin{cases} (U+n,V+n) \\ (U-n,V-n) \end{cases}, n>1 \\ \alpha+\beta & \text{if } (U',V') = \begin{cases} (U\pm n,V) \\ (U,V\pm n) \end{cases}, n=1 \\ \beta & \text{if } (U',V') = \begin{cases} (U\pm n,V) \\ (U,V\pm n) \end{cases}, n>1 \\ r_{(U,V)} & \text{if } (U',V') = (U,V) \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

To ensure all rows sum to 0, we set the diagonal elements to the negative row sum, $r_{(U,V)} = -\sum_{(u',v')\neq(U,V)} q_{(U,V),(u',v')}$. Note that $\mathbb{Q}$ defines the instantaneous rate of events such as $(U', V') = (U+n, V-k), n>0, k>0$ (i.e., events where (U, V) are changed by different CNAs) to be equal to 0. However, (U, V) can be changed by different CNAs for any evolutionary time interval $t>0$. Additionally, note we use the sum $\alpha+\beta$ as the rate for events with ±1 CNA, to allow ±1 copy number events to have a higher rate than larger magnitude copy number events.

The corresponding probability matrix, $\mathbb{P}$, of time dependent transition probabilities of adjacent bins changes from (U, V) to (U', V') is calculated as the matrix exponential

$$P_{(U,V),(U',V')}(t) = e^{\mathbb{Q}t} \tag{3}$$

where t is evolutionary time.

Two cell evolutionary process expansion

We now extend this single lineage process to describe the joint evolutionary process in two cells. Consider a pair of cells (A, B) and their most recent common ancestor in a tree, $\mathcal{T} = [t_1, t_2, t_3]$, where t_1 denotes the branch length of their shared history and t_2 and t_3 denote the branch lengths from divergence, at unobserved state Z, to cells A and B, respectively (see Fig. 1).

Under this tree structure, adjacent bins in cells A and B have a shared evolutionary history for time t_1 from an ancestral diploid state (i.e., $D : (2, 2)$) to an intermediate unobserved state, $Z : (W, Y)$, with associated transition probability $P_{(2,2),(W,Y)}(t_1)$. After divergence, bins in cell A evolve from (W, Y) to $(CN_{iA}, CN_{i+1,A})$ in time t_2 with transition probability $P_{(W,Y),(CN_{iA},CN_{i+1,A})}(t_2)$, where $CN_{iA}, CN_{i+1,A} \in \mathbb{S}_c$ denote copy number in windows i and $i+1$ for cell A. Similarly, bins in B evolve from (W, Y) to $(CN_{iB}, CN_{i+1,B})$ in time t_3 with transition probability $P_{(W,Y),(CN_{iB},CN_{i+1,B})}(t_3)$.

Approximating discrete Markov process along the genome

Next, we convert these continuous time process transition probabilities for adjacent bins in two cells into the transition probabilities for the approximating discrete Markov process for pairs of cells along the entire length of the genome, further described in Two Cell Hidden Markov Model Description. We do this by expanding the state space to the product space of the state space for each cell, $\mathbb{S}_c$. This expansion of the state space to the joint CN state for two cells is necessary as the correlation structure along the length of

the genome prevents the use of standard tree-based dynamic programming algorithms such as Felsenstein's pruning algorithm [20].

The state space, $\mathbb{S}_d$, for this discrete process is composed of pairs (CN_{iA}, CN_{iB}), representing the copy numbers for window i in cell pair (A, B), where $CN_{iA}, CN_{iB} \in \mathbb{S}_c = \{0, 1, \ldots, k\}$ for a fixed maximal copy number k, such that

$$\mathbb{S}_d = \mathbb{S}_c \times \mathbb{S}_c = \{(0,0), (0,1), (0,2), \ldots, (k,k)\} \tag{4}$$

We define matrix $\mathbb{F}(\mathcal{T}) = \{f_{(CN_{iA},CN_{iB}),(CN_{i+1,A},CN_{i+1,B})}(\mathcal{T})\}$ as the transition probability of moving from state (CN_{iA}, CN_{iB}) in window i to $(CN_{i+1,A}, CN_{i+1,B})$ in window $i+1$, given evolutionary tree $\mathcal{T} = [t_1, t_2, t_3]$, for cell pair (A, B). Therefore, the matrix $\mathbb{F}(\mathcal{T})$ is defined as

$$\begin{aligned} f_{(CN_{iA},CN_{iB}),(CN_{i+1,A},CN_{i+1,B})}(\mathcal{T}) = \sum_{W,Y \in \mathbb{S}_c} \Big(& P_{(2,2),(W,Y)}(t_1) \times \\ & P_{(W,Y),(CN_{iA},CN_{i+1,A})}(t_2) \times \\ & P_{(W,Y),(CN_{iB},CN_{i+1,B})}(t_3) \Big) \end{aligned} \tag{5}$$

which can be used to calculate a transition matrix, $\mathbb{M}(\mathcal{T})$, along the length of the genome for pairs of cells. This is done by dividing the joint probability of the CN state in both cells (A and B) in both bins (i and $i+1$), with the marginal probability of CN state in both cells (A and B) in bin i, i.e., dividing each entry in $\mathbb{F}(\mathcal{T})$ with the corresponding row sum:

$$\mathbb{M}(\mathcal{T}) = \{m_{(CN_{iA},CN_{iB}),(CN_{i+1,A},CN_{i+1,B})}\}(\mathcal{T}) \tag{6a}$$

$$m_{(CN_{iA},CN_{iB}),(CN_{i+1,A},CN_{i+1,B})}(\mathcal{T}) = \frac{f_{(CN_{iA},CN_{iB}),(CN_{i+1,A},CN_{i+1,B})}(\mathcal{T})}{\sum_{(c,d)\in\mathbb{S}_d} f_{(CN_{iA},CN_{iB}),(c,d)}(\mathcal{T})} \tag{6b}$$

We have thereby constructed a process with state space on the copy numbers of pairs of cells, $\mathbb{S}_d$. The matrix $\mathbb{M}(\mathcal{T})$ gives the probabilities of observing transitions from (CN_{iA}, CN_{iB}) in window i to $(CN_{i+1,A}, CN_{i+1,B})$ in window $i+1$, along the genome, for cell pair (A, B), given evolutionary tree $\mathcal{T}$. We also note that the process along the length of the genome is not Markovian, as breakpoints appear in pairs, inducing an inherently non-Markovian correlation structure (see also [6]). However, to facilitate computation, we will approximate the process as a Markovian process with transition probabilities given by $\mathbb{M}(\mathcal{T})$. We note that while this model approximates the evolutionary process and paired nature of breakpoints via the genome wide transition matrix $\mathbb{M}(\mathcal{T})$, it does not explicitly model pairs of breakpoints jointly, potentially leading to unpaired breakpoints. This Markov chain will then be used for inferences in a Hidden Markov Model framework with emission probabilities similar to those described in [6].

Two cell hidden Markov model description

Expanding on the framework of [6], we define a Hidden Markov Model (HMM) [21–23] to infer copy number across the genome for pairs of tumor cells, using binned read depth data.

Recall that cells A and B are associated with the evolutionary tree, $\mathcal{T}$, shown in Fig. 1. The sample space of read data, $\mathbb{A}$, is composed of pairs of observed per window read count values, (x_{iA}, x_{iB}), $x_{iA}, x_{iB} \in \mathbb{N}_0 = \{0, 1, 2, \dots\}$:

$$\mathbb{A} = \mathbb{N}_0 \times \mathbb{N}_0 = \{(0,0), (0,1), (0,2), \dots\} \tag{7}$$

This HMM uses the state space of paired copy numbers, $\mathbb{S}_d$, defined in Eq. 4 and the transition matrix $\mathbb{M}(\mathcal{T})$, defined in Eq. 6.

Emission probabilities

Assuming conditional independence between cells, the emission probabilities of the HMM are:

$$\mathbb{P}(X_{iA} = x_{iA}, X_{iB} = x_{iB} | CN_{iA}, CN_{iB}) = \mathbb{P}(X_{iA} = x_{iA} | CN_{iA}) \mathbb{P}(X_{iB} = x_{iB} | CN_{iB}) \tag{8}$$

As these probabilities are calculated similarly for cells A and B, we only describe the derivation for cell A (note: we previously described this derivation in [6]).

We assume X_{iA} follows a negative binomial distribution, such that

$$\mathbb{E}(X_{iA}) = \lambda_{iA} = \left(CN_{iA} \times \frac{\mu_i}{2} \right) \times s_A + \varepsilon \tag{9}$$

$$X_{iA} \sim NegBinom\left(\lambda_{iA}, \sigma^2_{iA} = a\lambda^2_{iA} + b\lambda_{iA} + c \right) \tag{10}$$

where

$$CN_{iA} = \text{the copy number in window} i \text{for cell} A \tag{11a}$$

$$\mu_i = \text{the mean diploid read depth in window } i \tag{11b}$$

$$\varepsilon = \text{constant sequencing error term} \tag{11c}$$

$$s_A = \text{ library size scaling factor for cell } A \tag{11d}$$

$$\{a, b, c\} = \text{constants learned from diploid data} \tag{11e}$$

Estimation of constants $\{a, b, c\}$ is described in Initial independent parameter estimation using SCONCE. In the following, we will describe the full SCONCE2 estimation procedure in detail.

Detailed SCONCE2 pipeline

Given binned read depths for tumor and matched diploid cells, joint copy number calling in SCONCE2 takes place in four main steps: (1) independently estimating model parameters and copy number profiles for each cell using SCONCE, (2) combining independent parameter estimates across cells, (3) estimating tree branch lengths for each cell pair, and (4) creating summarized copy number profiles. This process is illustrated in Fig. 2.

Initial independent parameter estimation using SCONCE

We first estimate constants $\{a, b, c\}$, defined in Eqs. 10 and 11e, using maximum likelihood on diploid cells only, as previously described in [6]. We note that most single cell tumor sequencing projects naturally also produce data from non-tumor diploid cells as part of standard sequencing techniques, and that these cells conveniently can be used for standardization [12, 24–30].

In order to obtain initial estimates of all model parameters, we analyze all tumor cells independently through SCONCE, described in detail in [6] and briefly summarized here. This is done to avoid the computational cost of joint estimation for all model parameters across all pairs of cells. The SCONCE pipeline first estimates the transition matrix of an unconstrained CN HMM, with associated library size scaling factor s_A, for each cell using a modified Baum-Welch [31] algorithm. These estimates are then used to obtain initial starting points for each model parameter for an optimization of the likelihood function using the Broyden–Fletcher–Goldfarb-Shanno (BFGS) algorithm [32]. This results in parameter estimates $\{\hat{s}_A, \hat{\alpha}_A, \hat{\beta}_A, \hat{\gamma}_A, \hat{t}_A\}$, for cell A. Recall s_A is the library size scaling factor, defined as the coverage for the cell relative to the average diploid library size, $\{\alpha, \beta, \gamma\}$ are the instantaneous rates for copy number events, and t_A is the total branch length from the ancestral diploid cell to cell A (see the red block in Fig. 2.).

Combining parameter estimates across multiple cells

To analyze shared evolutionary history between n cells, we first combine independent estimates across all cells of the transition rate parameters, $\{\alpha, \beta, \gamma\}$, assumed to be shared among all cells, using the median to form joint estimates:

$$\hat{\alpha} = \text{median}(\hat{\alpha}_A, \hat{\alpha}_B, \ldots, \hat{\alpha}_n) \tag{12a}$$

$$\hat{\beta} = \text{median}(\hat{\beta}_A, \hat{\beta}_B, \ldots, \hat{\beta}_n) \tag{12b}$$

$$\hat{\gamma} = \text{median}(\hat{\gamma}_A, \hat{\gamma}_B, \ldots, \hat{\gamma}_n) \tag{12c}$$

We note that a full joint optimization could possibly lead to better model parameter estimates, especially for highly heterogeneous tumor populations, as the median is not strongly affected by extreme or highly variable individual estimates. However, we opted not to pursue the estimation of such estimates because of considerations of computational efficiency. This is illustrated in the yellow block in Fig. 2.

Estimating pairwise tree branch lengths

Next, we estimate parameters of the joint two-cell process for all $\binom{n}{2}$ pairs of cells. The branch lengths of tree $\mathcal{T} = [t_1, t_2, t_3]$, are specific to each pair of cells, and branch length estimates from SCONCE, $\hat{t}$, are used to inform the initial optimization starting point for $\mathcal{T}$. For example, for pair (A, B), the initial branch length estimates, denoted with *, are:

$$t_1^* = \frac{min(\hat{t}_A, \hat{t}_B)}{2} \tag{13a}$$

$$t_2^* = \hat{t}_A - t_1^* \tag{13b}$$

$$t_3^* = \hat{t}_B - t_1^* \tag{13c}$$

For each pair of cells, we use the BFGS algorithm to maximize the forward log likelihood in order to estimate $\mathcal{T}$. To calculate the forward log likelihood of an observed sequence, the HMM is reset into the initial probability vector, defined as the steady state distribution, at the beginning of each chromosome to ensure chromosomal independence.

Because each set of branch lengths, $[t_1, t_2, t_3]$, is specific to each pair of cells, this procedure is trivially parallelizable (see the green block in Fig. 2).

Summarized copy number calling

After pairwise branch lengths are estimated, we use the Viterbi algorithm [33, 34] to estimate the most likely joint copy number profile for each pair of cells. If cell A appears in $n-1$ pairs, this results in $n-1$ separate CNP estimates for cell A. In order to calculate a single consensus copy number profile, $CN_{A,consensus}$, we use either the mean (default), median, or mode of the CN in each window among the $n-1$ estimates.

While adding more information to each consensus copy number profile by summarizing across multiple cells initially increases accuracy, summarizing across too many divergent cells is not optimal because more accurate estimates about each cell are obtained using closely related cells than highly divergent cells (Fig. 5). Therefore, in order to balance combining data from multiple cells and maintaining cell specificity, the user can also choose to summarize across a subset of the κ nearest neighbors for each cell, instead of all $n-1$ pairs a particular cell appears in. The nearest neighbors for each cell is defined by the Euclidean distance between individual copy number profiles from SCONCE. Then, the consensus copy number profile is calculated only across the κ selected pairs.

Note, because of the genomic binning procedure, true copy number events may be split across bin boundaries or be completely contained within one bin, resulting in bins with non-integer average copy number. Using the mean and median summary functions can result in non-integer copy number calls, which more accurately represent the underlying biology as genomes are not truly organized in discrete bins. However, many downstream tools for single cell analyses require integer copy number profiles, so these values may need to be rounded for downstream analyses.

Simulations

In order to evaluate the accuracy of SCONCE2, we use the Line Segments model from SCONCE [6], which simulates copy number events on a fixed length reference genome as additions or deletions to a collection of line segments, and does not impose a maximum copy number limit. Note that although copy number events change the number and length of line segments, the reference genome length is constant. Additionally, copy number events create pairs of breakpoints at either end of the event, which are explicitly maintained in this simulation model, unlike the approximating discrete Markov process in the SCONCE2 inference model (see Approximating discrete Markov process along the genome), thereby making the simulation model more biologically realistic.

While we previously used neutral coalescent simulations [6]to define trees, we here instead adjust the tree structure and the length of the branch leading to the root (ie, time to first divergence event) to examine a range of highly different tree structures, each including 128 cells. We specify two ultrametric trees with uniform branch lengths of 1/128, where tree A is fully pectinate/maximally imbalanced and tree B is perfectly balanced, and two non ultrametric trees, where tree C has uniform internal and terminal branch lengths of 1/128, and tree D has uniform internal branch lengths of 1/128 and logarithmically decaying terminal branch lengths. These tree structures represent extremes in terms of how balanced the tree is and in terms of deviations from a molecular clock (ultrametric property). Following the definitions of [35], tree A models branching evolution, tree B models neutral evolution, and trees C and D model linear evolution. Under certain conditions, the structure of tree B can also be adjusted to model punctuated evolution [35] if the branch leading to the root is lengthened relative to the internal tree branches, such that more mutations fall on the shared ancestral/root branch compared to external branches. For illustration, the tree structure for 8 cells is shown for each dataset in Additional file 1: Figure S1.

For each tree, the total tree height (longest path from the root to a leaf) was scaled to 1, and the branch leading to the root was set to length 1. Simulated reference genome lengths were set to 100, with amplification and deletion rates and expected lengths shown in Additional file 1: Table S1 (note that genomic length units are arbitrary, where expected copy number event lengths are defined relative to the genome length). As previously described in [6], the locations of copy number events follow a Markov process, and the lengths of copy number events follow a truncated exponential distribution.

To simulate read depths across the genome, the human reference genome was divided into 12,397 windows (equalling the number of 250 kb non overlapping uniform windows in hg19), and the number of reads falling into each window was simulated from a negative binomial distribution with parameter $r = 50$. This results in files listing genomic window coordinates and number of reads observed in that window, for every simulated cell (similar to output from `bedtools coverage` [36] on real data). The total expected number of reads for each cell was set to 4,000,000 (322.7 expected reads per window) to approximate the observed number of reads per 250 kb window (mean 316.1, median 351.2) in diploid cells from [12]. Note the actual number of total observed reads in each cell is random.

In order to ensure tree B was a perfectly balanced binary tree and to be consistent between tree structures, read depths for 128 tumor cells and 100 diploid cells were simulated for each tree. Read depth across diploid cells was averaged per window for each tree. Tumor cells from each tree were divided into five non overlapping subsets of 20 cells to create test sets. Although healthy cells were shared for each analysis run, each test set was otherwise analyzed independently from other test sets from the same tree.

All parameter files used to generate simulations are available on GitHub, along with examples of simulated data.

Real data preprocessing

We applied SCONCE2 to two published single cell breast cancer datasets, from [12] and [18], a cancer type known for their frequent CNAs [37]. Both of these datasets were

processed as previously described [6]. Briefly, for the [12] dataset, we trimmed reads using cutadapt [38] and trimmomatic [39], removed low complexity reads with prinseq [40], aligned reads to hg19 using bowtie2 [41], removed reads with q scores less than 20 using samtools [42], and removed PCR duplicates using picard [43]. For the [18] dataset, we split downloaded preprocessed bam files into cell specific bam files using pysam [44], and removed reads with q scores less than 20 using samtools [42]. Finally, we used `bedtools coverage` to count per window read depth for each cell [36]. Cells previously and orthogonally identified as diploid cells in [12] served as the matched normal. For the [18] dataset, cells from subset A were used as the diploid samples, as previously described [29].

Running other methods

For benchmarking, we limit our comparisons to other copy number only methods (that is, no SNP or phasing information is used): SCONCE [6] and AneuFinder [7, 8] for copy number accuracy, and the cnp2cnp [17] and MEDICC [16] distances for phylogeny building.

Briefly, we ran AneuFinder with default parameters, with the exception of skipping GC and mappability corrections to avoid overcorrecting, as we did not include GC or mappability bias in our simulations. To benchmark SCONCE2's copy number calling, we first ran SCONCE [6] with default parameters (`k=10`). To run AneuFinder [7, 8], we skipped the GC and mappability corrections steps to avoid over correcting, as our simulation model does not include GC or mappability biases. We directly ran AneuFinder's `findCNVs` function (default parameters: `method="edivisive", R=10, sig.lvl=0.1`). We extracted copy number calls from the resulting the `copy.number` element, and used `bedtools intersect` [36] to split large segments into 250 kb windows.

To evaluate SCONCE2's $t_2 + t_3$ distance metric in phylogeny estimation, we compared to the cnp2cnp distance [17] and the MEDICC distance [16]. To run cnp2cnp, we first converted and rounded called CNPs into fasta files, then ran cnp2cnp in matrix mode with default parameters (`-m matrix -d any`). Because the cnp2cnp metric depends on the input sample ordering and is not symmetric, we repeated this process on the reversed sample ordering, and summed the two resulting distance matrices to make a symmetric metric. To calculate the MEDICC distance, we used the ZZS implementation of the MEDICC algorithm in the cnp2cnp program to remove the maximum copy number limit of 4 in the original MEDICC software, and ran it on the same fasta files (`-m matrix -d zzs`). Full scripts to run other methods are provided on GitHub (`runAneufinder.sh` and `runCnp2cnp.sh`).

Phylogeny estimation and Robinson–Foulds distance calculations

To estimate phylogenies, distance matrices were first read into R [45]. Next, we applied neighbor-joining [13, 14], implemented in the `ape` [46] package, to each distance matrix to estimate phylogenies.

To calculate Robinson–Foulds distances between inferred trees and true trees, we first used the `read.tree` function in the `ape` [46] package to read true trees in Newick format into R. Next, we used the `treedist` function from the `phangorn` [47, 48]

package to calculate the Robinson–Foulds distance. Full scripts to estimate phylogenies from distance matrices and calculate Robinson–Foulds distances between phylogenies are available on GitHub (`readTreeBranches.R` and `plotRFdist.R`).

Abbreviations

BFGS	Broyden–Fletcher–Goldfarb–Shanno algorithm [32]
CN	Copy number
CNA	Copy number alteration(s)
CNP	Copy number profile
HMM	Hidden Markov model [21–23]
RF	Distance: Robinson–Foulds distance [19]
SNP	Single nucleotide polymorphism
SSE	Sum of squared errors

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s12859-022-04890-w.

Additional file 1. This file contains supplementary figures showing the simulated tree structures, analyses from summarizing over the nearest 10 cells, two examples of CNA detection in real data, and Robinson-Foulds distances across more copy number profiles and from summarizing across the nearest 10 cells. Additionally, it contains tables showing simulation parameter details, full median breakpoint distances and omega values (across all cells and across the 10 nearest cells), median pairwise Euclidean distances between SCONCE profiles, and median Robinson-Foulds distances (across all cells and across the 10 nearest cells).

Acknowledgements
Not applicable

Author Contributions
SH designed and implemented the SCONCE2 pipeline, and performed all data analysis. RN conceived the study and developed the simulation program. SH and RN wrote the manuscript. All authors read and approved the final manuscript.

Funding
This work was supported by the National Institutes of Health [R01GM138634-01 to R.N.]. The funding body played no role in the design of the study and collection, analysis, and interpretation of data, or in writing the manuscript.

Availability of data and materials
SCONCE2 is implemented in C++11, requires the Boost C++ Libraries (developed on v1.65.1)and the GNU Scientific Library (developed on v2.4) [49], has been developed and tested on Ubuntu 18.04.6, and is freely available from https://github.com/NielsenBerkeleyLab/sconce2. The simulation program (written in C) and corresponding parameter files, all R scripts (developed on R v4.1.2) [45] needed to preprocess diploid data (`avgDiploid.R` and `fitMeanVarRlnshp.R`), and R plotting scripts (`readBedFilesPairs.R`, `readTreeBranches.R`, `plotBetterBoundaries.R`, `plotDiminishingReturnsNumPairs.R`, `plotIllustrativeTrees.R`, `plotRFdist.R`, `plotSSEandBreakpointPairs.R`, `plotTreeBranchCorrelation.R`) are also on GitHub. Plotting scripts require the R packages `ape` [46], `cowplot` [50], `ggplot2` [51], `ggtree` [52–54], `grid` [45], `gtools` [55], `phangorn` [47, 48], plyr [56], `reshape2` [57], `scales` [58], and `stringr` [59]. We analyzed two previously published real datasets. Data from [12] is available at the Sequence Read Archive (SRA) under accession number SRR054616. The 10x dataset from [18] is available at https://cf.10xgenomics.com/samples/cell-dna/1.1.0/breast_tissue_aggr_10k/breast_tissue_aggr_10k_web_summary.html.

Declarations

Competing interests
The authors declare that they have no competing interests.

Received: 20 May 2022 Accepted: 11 August 2022
Published online: 19 August 2022

References

1. ...Beroukhim R, Mermel CH, Porter D, Wei G, Raychaudhuri S, Donovan J, Barretina J, Boehm JS, Dobson J, Urashima M, Henry KTM, Pinchback RM, Ligon AH, Cho Y-J, Haery L, Greulich H, Reich M, Winckler W, Lawrence MS, Weir BA, Tanaka KE, Chiang DY, Bass AJ, Loo A, Hoffman C, Prensner J, Liefeld T, Gao Q, Yecies D, Signoretti S, Maher E, Kaye FJ, Sasaki H, Tepper JE, Fletcher JA, Tabernero J, Baselga J, Tsao M-S, Demichelis F, Rubin MA, Janne PA, Daly MJ, Nucera C, Levine RL, Ebert BL, Gabriel S, Rustgi AK, Antonescu CR, Ladanyi M, Letai A, Garraway LA, Loda M, Beer DG, True

LD, Okamoto A, Pomeroy SL, Singer S, Golub TR, Lander ES, Getz G, Sellers WR. Meyerson M (2010) The landscape of somatic copy-number alteration across human cancers. Nature. 2010;463(7283):899–905. https://doi.org/10.1038/nature08822.
2. ...Gerstung M, Jolly C, Leshchiner I, Dentro SC, Gonzalez S, Rosebrock D, Mitchell TJ, Rubanova Y, Anur P, Yu K, Tarabichi M, Deshwar A, Wintersinger J, Kleinheinz K, Vázquez-García I, Haase K, Jerman L, Sengupta S, Macintyre G, Malikic S, Donmez N, Livitz DG, Cmero M, Demeulemeester J, Schumacher S, Fan Y, Yao X, Lee J, Schlesner M, Boutros PC, Bowtell DD, Zhu H, Getz G, Imielinski M, Beroukhim R, Sahinalp SC, Ji Y, Peifer M, Markowetz F, Mustonen V, Yuan K, Wang W, Morris QD, Spellman PT, Wedge DC, Loo PV. The evolutionary history of 2658 cancers. Nature. 2020;578(7793):122–8. https://doi.org/10.1038/s41586-019-1907-7.
3. Upender MB, Habermann JK, McShane LM, Korn EL, Barrett JC, Difilippantonio MJ, Ried T. Chromosome transfer induced aneuploidy results in complex dysregulation of the cellular transcriptome in immortalized and cancer cells. Can Res. 2004;64(19):6941–9. https://doi.org/10.1158/0008-5472.CAN-04-0474.
4. Cao S, Wang JR, Ji S, Yang P, Dai Y, Guo S, Montierth MD, Shen JP, Zhao X, Chen J, Lee JJ, Guerrero PA, Spetsieris N, Engedal N, Taavitsainen S, Yu K, Livingstone J, Bhandari V, Hubert SM, Daw NC, Futreal PA, Efstathiou E, Lim B, Viale A, Zhang J, Nykter M, Czerniak BA, Brown PH, Swanton C, Msaouel P, Maitra A, Kopetz S, Campbell P, Speed TP, Boutros PC, Zhu H, Urbanucci A, Demeulemeester J, Van Loo P, Wang W. Estimation of tumor cell total mRNA expression in 15 cancer types predicts disease progression. Nat Biotechnol. 2022;2022:1–10. https://doi.org/10.1038/s41587-022-01342-x.
5. Kashima Y, Sakamoto Y, Kaneko K, Seki M, Suzuki Y, Suzuki A. Single-cell sequencing techniques from individual to multiomics analyses. Exp Mol Med. 2020. https://doi.org/10.1038/s12276-020-00499-2.
6. Hui S, Nielsen R. SCONCE: a method for profiling copy number alterations in cancer evolution using single-cell whole genome sequencing. Bioinformatics. 2022. https://doi.org/10.1093/bioinformatics/btac041.
7. Bakker B, Taudt A, Belderbos ME, Porubsky D, Spierings DCJJ, de Jong TV, Halsema N, Kazemier HG, Hoekstra-Wakker K, Bradley A, de Bont ESJMJM, van den Berg A, Guryev V, Lansdorp PM, Colomé-Tatché M, Foijer F. Single-cell sequencing reveals karyotype heterogeneity in murine and human malignancies. Genome Biol. 17(1), 115 (2016). https://doi.org/10.1186/s13059-016-0971-7
8. Taudt AS. Hidden Markov models for the analysis of next-generation-sequencing data. PhD thesis, University of Groningen, Groningen (2018). https://research.rug.nl/en/publications/hidden-markov-models-for-the-analysis-of-next-generation-sequenci
9. Nilsen G, Liestøl K, Loo PV, Vollan HKM, Eide MB, Rueda OM, Chin S-F, Russell R, Baumbusch LO, Caldas C, Børresen-Dale A-L, Lingjærde OC. Copynumber: efficient algorithms for single- and multi-track copy number segmentation. BMC Genomics. 2012;13(1):1–16. https://doi.org/10.1186/1471-2164-13-591.
10. Kuipers J, Tuncel MA, Ferreira P, Jahn K, Beerenwinkel N. Single-cell copy number calling and event history reconstruction. bioRxiv, 2020-0428065755 ;2020. https://doi.org/10.1101/2020.04.28.065755
11. Cai H, Chen P, Chen J, Cai J, Song Y, Han G. WaveDec: a wavelet approach to identify both shared and individual patterns of copy-number variations. IEEE Trans Biomed Eng. 2018;65(2):353–64. https://doi.org/10.1109/TBME.2017.2769677.
12. Navin N, Kendall J, Troge J, Andrews P, Rodgers L, McIndoo J, Cook K, Stepansky A, Levy D, Esposito D, Muthuswamy L, Krasnitz A, McCombie WR, Hicks J, Wigler M. Tumour evolution inferred by single-cell sequencing. Nature. 2011;472(7341):90–4. https://doi.org/10.1038/nature09807.
13. Saitou N, Nei M. The neighbor-joining method: a new method for reconstructing phylogenetic trees. Mol Biol Evol. 1987;4(4):406–25. https://doi.org/10.1093/OXFORDJOURNALS.MOLBEV.A040454.
14. Studier JA, Keppler KJ. A note on the neighbor-joining algorithm of Saitou and Nei. Mol Biol Evol. 1988;5(6):729–31. https://doi.org/10.1093/OXFORDJOURNALS.MOLBEV.A040527.
15. Schwartz R, Schäffer AA. The evolution of tumour phylogenetics: principles and practice. Nature Rev Genet. 2017;18(4):213–29. https://doi.org/10.1038/nrg.2016.170.
16. Schwarz RF, Trinh A, Sipos B, Brenton JD, Goldman N, Markowetz F. Phylogenetic quantification of intra-tumour heterogeneity. PLoS Comput Biol. 2014;10(4):1003535. https://doi.org/10.1371/JOURNAL.PCBI.1003535.
17. Cordonnier G, Lafond M. Comparing copy-number profiles under multi-copy amplifications and deletions. BMC Genom. 2020;21(2):1–12. https://doi.org/10.1186/S12864-020-6611-3/FIGURES/5.
18. 10x Genomics: Breast Tissue nuclei sections A-E (v1, 84x100) (2019). https://cf.10xgenomics.com/samples/cell-dna/1.1.0/breast_tissue_aggr_10k/breast_tissue_aggr_10k_web_summary.html
19. Robinson DF, Foulds LR. Comparison of phylogenetic trees. Math Biosci. 1981;53(1–2):131–47. https://doi.org/10.1016/0025-5564(81)90043-2.
20. Felsenstein J. Journal of molecular evolution evolutionary trees from DNA sequences: a maximum likelihood approach. J Mol Evol. 1981;17:368–76.
21. Baum LE, Petrie T. Statistical inference for probabilistic functions of finite state Markov chains. Ann Math Stat. 1966;37(6):1554–63.
22. Baum LE, Eagon JA. An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology. Bull Am Math Soc. 1967;73(3):360–3.
23. Baum LE, Petrie T, Soules G, Weiss N. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. Ann Math Stat. 1970;41(1):164–71.
24. Mallory XF, Edrisi M, Navin N, Nakhleh L. Methods for copy number aberration detection from single-cell DNA-sequencing data. Genome Biol. 2020;21(1):208. https://doi.org/10.1186/s13059-020-02119-8.
25. Wang X, Chen H, Zhang NR. DNA copy number profiling using single-cell sequencing. Brief Bioinform. 2018;19(5):731–6. https://doi.org/10.1093/BIB/BBX004.
26. Wang R, Lin DY, Jiang Y. SCOPE: a normalization and copy-number estimation method for single-cell DNA sequencing. Cell Syst. 2020;10(5):445–52. https://doi.org/10.1016/J.CELS.2020.03.005.
27. Lähnemann D, Köster J, Szczurek E, McCarthy DJ, Hicks SC, Robinson MD, Vallejos CA, Campbell KR, Beerenwinkel N, Mahfouz A, Pinello L, Skums P, Stamatakis A, Attolini CSO, Aparicio S, Baaijens J, Balvert M, Barbanson Bd, Cappuccio A, Corleone G, Dutilh BE, Florescu M, Guryev V, Holmer R, Jahn K, Lobo TJ, Keizer EM, Khatri I, Kielbasa SM, Korbel JO,

Kozlov AM, Kuo TH, Lelieveldt BPF, Mandoiu II, Marioni JC, Marschall T, Mölder F, Niknejad A, Raczkowski L, Reinders M, Ridder Jd, Saliba AE, Somarakis A, Stegle O, Theis FJ, Yang H, Zelikovsky A, McHardy AC, Raphael BJ, Shah SP, Schönhuth A. Eleven grand challenges in single-cell data science. Genome Biol. 21(1), 1–35;2020. https://doi.org/10.1186/S13059-020-1926-6
28. Casasent AK, Schalck A, Gao R, Sei E, Long A, Pangburn W, Casasent T, Meric-Bernstam F, Edgerton ME, Navin NE. Multiclonal invasion in breast tumors identified by topographic single cell sequencing. Cell. 2018;172(1–2):205–17. https://doi.org/10.1016/J.CELL.2017.12.007.
29. 10x Genomics: Application Note - Assessing Tumor Heterogeneity with Single Cell CNV (2018)
30. Zaccaria S, Raphael BJ. Characterizing allele- and haplotype-specific copy numbers in single cells with CHISEL. Nat Biotechnol. 2021;39(2):207–14. https://doi.org/10.1038/s41587-020-0661-6.
31. Rabiner LR. A tutorial on hidden Markov models and selected applications in speech recognition. Proc IEEE. 1989;77(2):257–86. https://doi.org/10.1109/5.18626.
32. Fletcher R. Newton-Like Methods. In: Practical Methods of Optimization, 2nd edn., pp. 44–79. Wiley, Chichester (2000). Chap. 3. https://doi.org/10.1002/9781118723203.ch3
33. Viterbi AJ. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. IEEE Trans Inf Theory. 1967;13(2):260–9. https://doi.org/10.1109/TIT.1967.1054010.
34. Forney GD. The viterbi algorithm. Proc IEEE. 1973;61(3):268–78. https://doi.org/10.1109/PROC.1973.9030.
35. Davis A, Gao R, Navin N. Tumor evolution: linear, branching, neutral or punctuated? Biochimica et Biophysica Acta (BBA) - Rev Cancer 2017;1867(2), 151–161. https://doi.org/10.1016/J.BBCAN.2017.01.003
36. Quinlan AR, Hall IM. BEDTools: a flexible suite of utilities for comparing genomic features. Bioinformatics (Oxford, England). 2010;26(6):841–2. https://doi.org/10.1093/bioinformatics/btq033.
37. Li Z, Zhang X, Hou C, Zhou Y, Chen J, Cai H, Ye Y, Liu J, Huang N. Comprehensive identification and characterization of somatic copy number alterations in triple-negative breast cancer. Int J Oncol. 2020;56(2):522–30. https://doi.org/10.3892/IJO.2019.4950.
38. Martin M. Cutadapt removes adapter sequences from high-throughput sequencing reads. EMBnet. Journal 2011; 17(1), 10–12 . https://doi.org/10.14806/ej.17.1.200
39. Bolger AM, Lohse M, Usadel B. Trimmomatic: a flexible trimmer for Illumina sequence data. Bioinformatics (Oxford, England). 2014;30(15):2114–20. https://doi.org/10.1093/bioinformatics/btu170.
40. Schmieder R, Edwards R. Quality control and preprocessing of metagenomic datasets. Bioinformatics. 2011;27(6):863–4. https://doi.org/10.1093/bioinformatics/btr026.
41. Langmead B, Salzberg SL. Fast gapped-read alignment with Bowtie 2. Nat Methods. 2012;9(4):357–9. https://doi.org/10.1038/nmeth.1923.
42. Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, Marth G, Abecasis G, Durbin R. The sequence alignment/map format and SAMtools. Bioinformatics (Oxford, England). 2009;25(16):2078–9. https://doi.org/10.1093/bioinformatics/btp352.
43. The Broad Institute: Picard: A set of command line tools (in Java) for manipulating high-throughput sequencing (HTS) data and formats such as SAM/BAM/CRAM and VCF. (2021). http://broadinstitute.github.io/picard/
44. Heger A, Jacobs K, et al. pysam 2021. https://github.com/pysam-developers/pysam
45. R Core Team R. A Language and Environment for Statistical Computing, Vienna, Austria (2021). https://www.R-project.org/
46. Paradis E, Schliep K. ape 5.0: an environment for modern phylogenetics and evolutionary analyses in R. Bioinformatics 2019; 35(3), 526–528 . https://doi.org/10.1093/BIOINFORMATICS/BTY633
47. Schliep KP. phangorn: phylogenetic analysis in R. Bioinformatics. 2011;27(4):592–3. https://doi.org/10.1093/BIOINFORMATICS/BTQ706.
48. Schliep K, Potts AJ, Morrison DA, Grimm GW. Intertwining phylogenetic trees and networks. Methods Ecol Evol. 2017;8(10):1212–20. https://doi.org/10.1111/2041-210X.12760.
49. Galassi M, Davies J, Theiler J, Gough B, Jungman G, Booth M, Rossi F. GNU Scientific Library Reference Manual. Network Theory Ltd. 2006
50. Wilke CO. cowplot: Streamlined Plot Theme and Plot Annotations for 'ggplot2' 2020. https://CRAN.R-project.org/package=cowplot
51. Wickham H. ggplot2: Elegant Graphics for Data Analysis. Springer 2016. https://ggplot2.tidyverse.org
52. Yu G, Smith DK, Zhu H, Guan Y, Lam TTY. ggtree: an r package for visualization and annotation of phylogenetic trees with their covariates and other associated data. Methods Ecol Evol. 2017;8(1):28–36. https://doi.org/10.1111/2041-210X.12628.
53. Yu G, Lam TTY, Zhu H, Guan Y. Two methods for mapping and visualizing associated data on phylogeny using Ggtree. Mol Biol Evol. 2018;35(12):3041–3. https://doi.org/10.1093/MOLBEV/MSY194.
54. Yu G. Using ggtree to visualize data on tree-like structures. Curr Protoc Bioinform. 2020;69(1):96. https://doi.org/10.1002/CPBI.96.
55. Warnes GR, Bolker B, Lumley T. gtools: Various R Programming Tools 2021. https://CRAN.R-project.org/package=gtools
56. Wickham H. The split-apply-combine strategy for data analysis. J Stat Software 40(1), 1–29; 2011. https://doi.org/10.18637/JSS.V040.I01
57. Wickham H. Reshaping Data with the reshape Package. Journal of Statistical Software 21(12), 1–20 ; 2007. https://doi.org/10.18637/JSS.V021.I12
58. Wickham H, Seidel D. scales: Scale Functions for Visualization 2020. https://CRAN.R-project.org/package=scales
59. Wickham H. stringr: Simple, Consistent Wrappers for Common String Operations 2019. https://CRAN.R-project.org/package=stringr

Publisher's Note