A Supplementary Material for Lei et al. (2020)

A.1 Supplementary Methods

A.1.1 Optimization model

We begin with a full description of the formal problem statement and its realization as an integer linear program. For clarity of exposition, we restate the objective function of the method:

$$\begin{aligned}\min_{\boldsymbol{C},\boldsymbol{F},\boldsymbol{S},\boldsymbol{P}} (&||\boldsymbol{B}-\boldsymbol{CPF}||_1 \\ &+\alpha_f \cdot ||\boldsymbol{F}-\boldsymbol{F'}||_1 \\ &+\alpha_p \cdot J(\boldsymbol{S},\boldsymbol{C},\boldsymbol{C'}) \\ &+\alpha_c \cdot ||\boldsymbol{X}^T\boldsymbol{CP}-\boldsymbol{H'}||_1)\end{aligned}$$

Table S1 describes the main variables used in the objective function. Additional variables and constraints are explained in the subsequent sections in defining the full program.

A.1.2 Estimating $\boldsymbol{F}$

$||\boldsymbol{B}-\boldsymbol{CPF}||_1$ *constraints:* We define the $L1$ distance of $||\boldsymbol{B}-\boldsymbol{CPF}||_1$ as:

$$||\boldsymbol{B}-\boldsymbol{CPF}||_1 = \sum_{i=1}^{m}\sum_{j=1}^{n} b_{\Delta,i,j} \tag{2}$$

with constraints:

$$b_{\Delta,i,j} \geq b_{i,j} - \sum_{r=1}^{k} c_{i,r}\cdot p_{r,r}\cdot f_{r,j}, \forall i \in \{1,...,m\}, \quad j \in \{1,...,n\} \tag{3}$$

$$b_{\Delta,i,j} \geq -b_{i,j} + \sum_{r=1}^{k} c_{i,r}\cdot p_{r,r}\cdot f_{r,j}, \forall i \in \{1,...,m\}, \quad j \in \{1,...,n\}. \tag{4}$$

where m is the number of total genomic loci, n is the number of bulk tumor samples, and k is the number of cells.

$\boldsymbol{F}$ *constraints:* Since $\boldsymbol{F}$ is a weighted matrix, each column of $\boldsymbol{F}$ should add up to 1 and all entries are non-negative.

$$0 \leq f_{r,j} \leq 1, \forall r \in \{1,...,k\}, j \in \{1,...,n\} \tag{5}$$

$$\sum_{r=1}^{n} f_{r,j} = 1, \forall j \in \{1,...,n\} \tag{6}$$

$||\boldsymbol{F}-\boldsymbol{F'}||_1$ *constraints:* We apply $L1$ distance on $||\boldsymbol{F}-\boldsymbol{F'}||_1$:

$$||\boldsymbol{F}-\boldsymbol{F'}||_1 = \sum_{r=1}^{k}\sum_{j=1}^{n} f_{\Delta,r,j} \tag{7}$$

with constraints:

$$f_{\Delta,r,j} \geq f_{r,j} - f'_{r,j}, \forall r \in \{1,...,k\}, j \in \{1,...,n\} \tag{8}$$

$$f_{\Delta,r,j} \geq -f_{r,j} + f'_{r,j}, \forall r \in \{1,...,k\}, j \in \{1,...,n\}. \tag{9}$$

In summary, when we estimate $\boldsymbol{F}$, we optimize:

$$\min_{\boldsymbol{F}} \left(||\boldsymbol{B}-\boldsymbol{CPF}||_1 + \alpha_f \cdot ||\boldsymbol{F}-\boldsymbol{F'}||_1\right)$$

with constraints (3)-(9).

A.1.3 Estimating S

$J(\boldsymbol{S},\boldsymbol{C},\boldsymbol{C'})$ *constraints:* $J(\boldsymbol{S},\boldsymbol{C},\boldsymbol{C'})$ provides an error term for the cost of an phylogenetic relationship describing shared ancestry among the inferred cell data $\boldsymbol{C}$ and the reference cell data $\boldsymbol{C'}$. In computing this term, we define a phylogenetic structure with a $K \times K$ directed adjacency matrix $\boldsymbol{S}$, where $K = 2k+1$, the first k columns indicate $\boldsymbol{C}$, the next k columns indicate $\boldsymbol{C'}$ and the last column indicates a root with normalized copy numbers all-2 (diploid). We introduce a vertex set $T = \{1,\ldots,2k+1\}$ that represents the set of all cells in $\boldsymbol{S}$. Let r be the unique, predetermined, root of T. For $t,u,v \in T$, we introduced the binary variables $g^t_{v,u}$ representing the amount of flow along edge (u,v) with destination $t \in T$. Then the full constraints are:
flow conservation on the Steiner vertices:

$$\sum_v g^t_{uv} = \sum_v g^t_{vu},\ \forall u \in T, u \neq t, u \neq r \tag{10}$$

inflow/outflow constraints on terminals in T:

$$\sum_v g^t_{uv} = \sum_v g^t_{vu},\ \forall u \in T, u \neq t, u \neq r \tag{11}$$

$$\sum_v g^t_{vt} = 1,\ \forall t \in T, t \neq r \tag{12}$$

$$g^t_{vr} = 0,\ \forall v \tag{13}$$

$$\sum_v g^t_{tv} = 0, \sum_v g^t_{rv} = 1,\ \forall t \in T \tag{14}$$

positive flow on an edge iff the edge is selected:

$$0 \leq g^t_{uv} \leq s_{uv},\ \forall t \in T \tag{15}$$

no self loops:

$$s_{uu} = 0,\ \forall u \tag{16}$$

binary variable for g^t_{uv} and $s_{u,v}$:

$$g^t_{uv}, s_{uv} \in \{0,1\} \tag{17}$$

Phylogenetic cost: We then define the measurement for evolutionary distance across each edge (u,v) in the tree as $L1$ distance of the copy number profiles of the edge endpoints(l^*_u, l^*_v) and introduce a minimum evolution model defined by $\boldsymbol{S}$ to estimate the phylogenetic cost:

$$J(\boldsymbol{S},\boldsymbol{C},\boldsymbol{C'}) = \sum_{u=1}^{K}\sum_{v=1}^{K} s_{uv}\cdot\|c^*_u - c^*_v\|_1. \tag{18}$$

We define the phylogeny objective to be derived from normalized copy numbers, effectively ignoring ploidy changes in the evolution objective and measuring distance from localized focal copy number variations only. One might plausibly improve on this model by accounting for ploidy changes separately as evolutionary events (Chowdhury *et al.*, 2014, 2015) or adopting a more nuanced general model of copy number change, such as the MEDICC model (Schwarz *et al.*, 2014). The $L1$ distance of normalized copy numbers is used as a heuristic because of the difficulty of incorporating these other model types into the ILP framework.

A.1.4 Estimating $\boldsymbol{C}$

$\boldsymbol{C}$ *constraints:* We impose some basic constraints on $\boldsymbol{C}$: (1) all copy numbers are no larger than a certain maximum number c_{max}, which is set at *10* in our tests; (2) all copy numbers must be integers.

$$c_{i,r} \leq c_{max}, \forall i \in \{1,...,m\}, r \in \{1,...,k\} \tag{19}$$

$$c_{i,r} \in \mathbb{N}_0, \forall i \in \{1,...,m\}, r \in \{1,...,k\} \tag{20}$$

$||\boldsymbol{C}-\boldsymbol{C'}||_1$ *constraints:* The $||\boldsymbol{C}-\boldsymbol{C'}||_1$ term is not explicitly expressed in the objective function. Instead, it exists in the $J(\boldsymbol{S},\boldsymbol{C},\boldsymbol{C'})$ term since we apply L_1 distance between two nodes in S as the edge weight (Eq. 18). Then we redefine:

$$w_{\Delta,u,v,i} = ||c^*_{i,u} - c^*_{i,v}|| \tag{21}$$

with constraints:

$$w_{\Delta,u,v,i} \geq c^*_{i,u} - c^*_{i,v}, \forall i \in \{1,...,2k+1\} \tag{22}$$

$$w_{\Delta,u,v,i} \geq c^*_{i,v} - c^*_{i,u}, \forall i \in \{1,...,2k+1\} \tag{23}$$

$$w_{u,v} = \sum_i w_{\Delta,u,v,i}. \tag{24}$$

$||\boldsymbol{X}^T\boldsymbol{C}\boldsymbol{P} - \boldsymbol{H'}||_1$ *constraints:* miFISH probes each cover a genomic interval spanned by SCS data, so $||\boldsymbol{X}^T\boldsymbol{C}\boldsymbol{P} - \boldsymbol{H'}||_1$ provides a way to favor consistency between miFISH and SCS data over these intervals in the optimization. We note that one might optionally weight this objective term to account for varying clonal frequencies of the miFISH cells, although we do not do so here. In this step, the known $\boldsymbol{H'}$, which represents the *unnormalized* miFISH probe counts, provides additional constraints on the copy numbers in $\boldsymbol{C}$. To get these constraints, we first define an array $Index$ mapping miFISH probe regions to SCS copy number regions, where each element indicates whether a copy number region is considered to be covered by, or strongly correlated with, a miFISH probe. We then define $\boldsymbol{X}$ as follows:

$$\boldsymbol{X}_{ij} = \begin{cases} 1, & \text{if } i \in Index \text{ and } i = Index[j] \\ 0, & \text{otherwise} \end{cases}$$

for $\forall i \in \{1,...,m\}$, and $\forall j \in \{1,...,Index.\text{length}\}$
Then we impose $L1$ distance on $||\boldsymbol{X}^T\boldsymbol{C}\boldsymbol{P} - \boldsymbol{H'}||$ and redefine it as:

$$||\boldsymbol{X}^T\boldsymbol{C}\boldsymbol{P} - \boldsymbol{H'}||_1 = \sum_{p=1}^{s}\sum_{r=1}^{k} z_{\Delta,p,r} \tag{25}$$

with constraints:

$$z_{\Delta,p,r} \geq \sum_{i=1}^{m} x_{p,i}\cdot c_{i,r}\cdot p_{r,r} - h'_{p,r}, \forall p \in \{1,...,s\}, \quad r \in \{1,...,k\} \tag{26}$$

$$z_{\Delta,p,r} \geq -\sum_{i=1}^{m} x_{p,i}\cdot c_{i,r}\cdot p_{r,r} + h'_{p,r}, \forall p \in \{1,...,s\}, \quad r \in \{1,...,k\}. \tag{27}$$

where s is the number of miFISH probes, k is the number of cells.

A.1.5 Estimating $\boldsymbol{P}$

$\boldsymbol{P}$ *constraints:* $\boldsymbol{P}$ is the diagonal matrix whose diagonal elements are the half ploidies (re-scaling factors) to transform the *normalized copy numbers to* unnormalized copy numbers. We also set lower (p_{min}) and upper (p_{max}) bounds for p_{ij}, and these are set at 0 and 8 respectively in our tests below. The complete constraints are then:

$$p_{i,j} \leq p_{max}, \forall i,j \in \{1,...,k\} \tag{28}$$

$$p_{i,j} \geq p_{min}, \forall i,j \in \{1,...,k\} \tag{29}$$

$$p_{i,j} \in \mathbb{R}^+, \forall i,j \in \{1,...,k\} \tag{30}$$

$$p_{i,j} = 0, \forall i \neq j, i,j \in \{1,...,k\} \tag{31}$$

$||\boldsymbol{X}^T\boldsymbol{C}\boldsymbol{P} - \boldsymbol{H'}||_1$ *constraints:* Unlike in A.1.4, in this step, $\boldsymbol{C}$ is known from the computation in previous step to update $\boldsymbol{C}$, and we would like to update $\boldsymbol{P}$. *Index* is defined such that $\boldsymbol{X^T}$ represents the *normalized* miFISH probes, which we can compute after updating $\boldsymbol{C}$, then $\boldsymbol{X^T C}$ can be redefined as $\boldsymbol{Y}$. Then $\boldsymbol{YP}$ is the $unnormalized$ FISH probes. We still impose L_1 distance between $\boldsymbol{YP}$ and $\boldsymbol{H'}$ and redefine it as:

$$||\boldsymbol{Y}\boldsymbol{P} - \boldsymbol{H'}||_1 = \sum_{p=1}^{s}\sum_{r=1}^{k} h_{\Delta,p,r} \tag{32}$$

with constraints:

$$h_{\Delta,p,r} \geq y_{p,r}\cdot p_{r,r} - h'_{p,r}, \forall p \in \{1,...,s\}, r \in \{1,...,k\} \tag{33}$$

$$h_{\Delta,p,r} \geq -y_{p,r}\cdot p_{r,r} + h'_{p,r}, \forall p \in \{1,...,s\}, r \in \{1,...,k\} \tag{34}$$

where s is the number of miFISH probes, k is the number of cells.

A.1.6 Coordinate descent method for deconvolution

For clarity of exposition, we restate the objective function of the method:

$$\begin{aligned}\min_{\boldsymbol{C},\boldsymbol{F},\boldsymbol{S},\boldsymbol{P}} (&||\boldsymbol{B} - \boldsymbol{C}\boldsymbol{P}\boldsymbol{F}||_1 \\ &+ \alpha_f \cdot ||\boldsymbol{F} - \boldsymbol{F'}||_1 \\ &+ \alpha_p \cdot J(\boldsymbol{S},\boldsymbol{C},\boldsymbol{C'}) \\ &+ \alpha_c \cdot ||\boldsymbol{X}^T\boldsymbol{C}\boldsymbol{P} - \boldsymbol{H'}||_1)\end{aligned}$$

Table S1 describes the main variables used in the objective function. Additional variables and constraints are explained in the main paper in Sec. 2.

The original deconvolution problem as shown above is non-convex, and it is hard to derive a closed form for the solution, so we apply a coordinate descent method to solve $\boldsymbol{F},\boldsymbol{S},\boldsymbol{C},\boldsymbol{P}$ iteratively by following the order of Sec. A.1.2 to Sec. A.1.5 with the corresponding constraints for each term (Algorithm 1).

A.1.7 Extending the reference miFISH matrix

The original $Index$ contains the indices of the 8 original FISH probes in the SCS data. However, compared to the 9934 genomic positions in the SCS data, 8 probes only contribute a very tiny portion to the copy number inference. Instead, if we find that the genomic positions around the miFISH probes are highly correlated (Fig. S1), then we extend the $Index$ by adding to it the consecutive genomic positions that are highly correlated with the miFISH probes (light blocks in the Fig. S1, threshold=0.95). We use two pointers to make sure the correlated genomic positions are consecutive to each other and to the miFISH probe (Algorithm 2), and those positions that may be also highly correlated but far away in the genomic positions or even on different chromosomes would not be considered as correlated.

A.1.8 Semi-Synthetic Data Simulation

This section describes our protocol for simulating data to test the algorithms. The guiding principle of the method is to generate a ground truth dataset in which the true clones and their mixture fractions are known and resemble the GBM data, then subsample artificial bulk, SCS, or FISH data from that single ground truth. Note that we do not know the ground truth clonal lineage tree for these semi-simulated data, since we are deriving them from actual SCS data for which the ground truth lineage is unknown. We set NUM_REGIONS=3 and NUM_PROBES=8 and MAX_COPY=10 to match the GBM data. We define this ground truth in terms of six data structures:

Algorithm 1: Modified Coordinate Descent Algorithm for Deconvolution

$i = 1$;
$\boldsymbol{H'}$ = reference copy number at FISH probes;
$\boldsymbol{F'}$ = reference mixture fractions;
$\boldsymbol{C'}$ = reference single cell;
$\boldsymbol{C}^{(i)}$ = diploid initialization;
$\boldsymbol{P}^{(i)}$ = initial ploidy;
distance = $+\infty$;
dnorm$_0$ = 0;
while distance $>$ threshold **do**
  $\boldsymbol{F}^{(i)} \leftarrow \text{argmin}_{\boldsymbol{F}}(||\boldsymbol{B} - \boldsymbol{C}^{(i)}\boldsymbol{P}^{(i)}\boldsymbol{F}|| - \alpha_f \cdot ||\boldsymbol{F} - \boldsymbol{F'}||)$ given constrains (3)-(9);
  $\boldsymbol{S}^{(i)} \leftarrow \text{argmin}_{\boldsymbol{S}}(\boldsymbol{S} \cdot ||\boldsymbol{C}^{(i)} - \boldsymbol{C'}||)$ given constrains (10)-(17);
  $\boldsymbol{C}^{(i)} \leftarrow \text{argmin}_{\boldsymbol{C}}(||\boldsymbol{B} - \boldsymbol{C}\boldsymbol{P}^{(i)}\boldsymbol{F}^{(i)}|| - \alpha_p \cdot J(\boldsymbol{S}^{(i)}, \boldsymbol{C}, \boldsymbol{C'})$ given constraints (19)-(27);
  $\boldsymbol{P}^{(i)} \leftarrow \text{argmin}_{\boldsymbol{P}}(||\boldsymbol{B} - \boldsymbol{C}^{(i)}\boldsymbol{P}\boldsymbol{F}^{(i)}|| - \alpha_c \cdot ||\boldsymbol{X}^T\boldsymbol{C}^{(i)}\boldsymbol{P} - \boldsymbol{H'}||)$ given constraints (28)-(34);
  dnorm = $||\boldsymbol{B} - \boldsymbol{C}^{(i)}\boldsymbol{F}^{(i)}||_{Fr}^2$;
  distance = ||dnorm$_0$-dnorm||;
  dnorm$_0 \leftarrow$ dnorm;
  $i \leftarrow i + 1$;
  if $i >$ Maxiter **then**
    quit the loop
  end
end

$\boldsymbol{B} \in \mathbb{R}^{+(m \times n)}$	$b_{i,j}$ is the mixed copy number of genomic location i in tumor sample j
$\boldsymbol{C} \in \mathbb{N}_0^{(m \times k)}$	$c_{i,r}$ is the integer copy number of genomic locaion i in inferred cell type r
$\boldsymbol{C'} \in \mathbb{N}_0^{(m \times k)}$	$c'_{i,r}$ is the integer copy number of genomic locaion i in reference cell type r
$\boldsymbol{F} \in \mathbb{R}^{+(k \times n)}$	$f_{r,j}$ is the mixture fraction of inferred cell type r in tumor sample j
$\boldsymbol{F'} \in \mathbb{R}^{+(k \times n)}$	$f'_{r,j}$ is the mixture fraction of reference cell type r in tumor sample j
$\boldsymbol{P} \in \mathbb{R}^{+(k \times k)}$	$p_{r,r}$ is the half ploidy of reference cell type r and $p_{i,j} = 0, \forall i \neq j$
$\boldsymbol{S} \in \{0,1\}^{(K \times K)}$	$s_{u,v} = 1$ if cell type u is ancestor of cell type v

Table S1. Variables in the objective function

Algorithm 2: Extend the $Index$ of FISH probes

corrMat $\leftarrow$ correlation matrix of genomic position in SCS;
$Extend\ Index \leftarrow$ empty list ;
for p *in* $Index$ **do**
  tempArr $\leftarrow$ empty list;
  pointer1, pointer2 = p, p;
  while TRUE do
    if pointer1 $>= 0$ *and* corrMat[pointer1, p] $>=$ threshold **then**
      add pointer1 to tempArr;
      pointer1 = pointer1 + 1;
    else if pointer2 $>= 0$ *and* corrMat[pointer2, p] $>=$ threshold **then**
      add pointer2 to tempArr;
      pointer2 = pointer2 - 1;
    else
      quit the loop;
    end
  end
  add every element in tempArr to $Extend\ Index$;
end
$Index \leftarrow Extend\ Index$

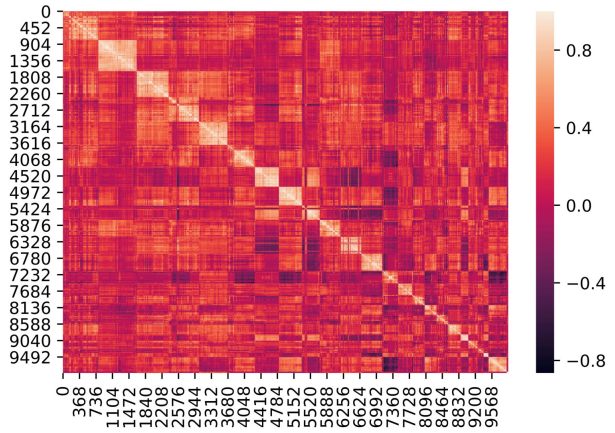

Fig. S1. Correlation matrix for genomic positions. The light blocks indicates the neighbouring genomic position are highly correlated in positive direction. For each one of 8 original FISH probe indexes, We search the consecutive genomic positions that are highly correlated with it and add it to $Index$, so that we extend the original $Index$ from length of 8 to the length around 100 (please also refer to Fig. S2, step (9)).

1. $\tilde{\boldsymbol{C}}$: a matrix of normalized copy number profiles of all selected cells, including major, minor and tiny clones. Each column of $\tilde{\boldsymbol{C}}$ corresponds to a ground truth single cell and each row to the mean copy number at a single genomic locus, where it is assumed the rows collectively span the full genome. We assume each cell (column) is normalized to mean diploid count.
2. $\hat{\boldsymbol{C}}$: a matrix of normalized copy number profiles of major clones in each tumor region. According to previous description, $\hat{\boldsymbol{C}}$ was generated by picking the first two components in $\tilde{\boldsymbol{C}}$ and used to calculate copy number accuracy and RMSD for performance estimation.
3. $\tilde{\boldsymbol{P}}$: a diagonal matrix of half ploidies, where each non-zero element $\tilde{p}_{ii}$ provides a scaling factor to convert the diploid row $\tilde{c}_i$ to absolute (unnormalized) copy numbers.
4. $\hat{\boldsymbol{P}}$: a diagonal matrix of half ploidies, where each non-zero element $\hat{p}_{ii}$ provides a scaling factor to convert the diploid row $\hat{c}_i$ to absolute (unnormalized) copy numbers.
5. $\tilde{\boldsymbol{F}}$: a matrix of mixture fractions, where each row corresponds to a selected cell and column defines a probability density describing frequency of occurrence of each cell type in the bulk samples.
6. $\hat{\boldsymbol{F}}$: a matrix of mixture fractions, where each row corresponds to a major clone and column defines a probability density describing approximate frequency of occurrence of each major clone in the bulk samples. $\hat{\boldsymbol{F}}$ is derived from $\tilde{\boldsymbol{F}}$, but column is also normalized to 1.

We first define this ground truth model, then generate simulated data of each needed type by sampling from the model. These processes are described step-by-step below.

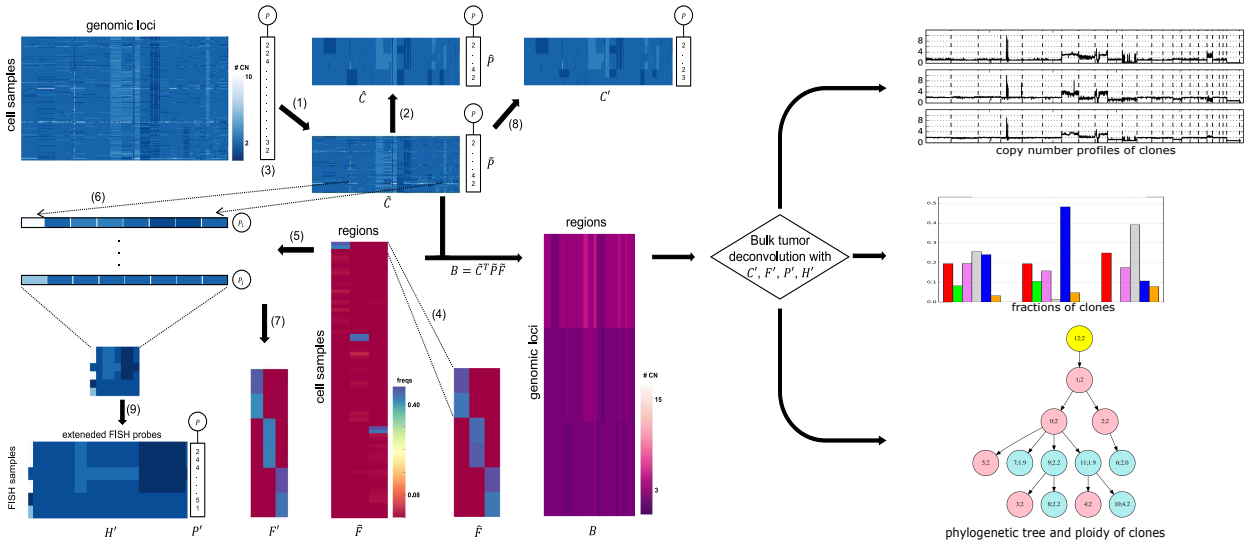

Fig. S2. Workflow of the simulation and deconvolution. The figure shows the process from real SCS data to select SCS clones, sample ploidies, simulate mixture fractions, simulate FISH and simulate bulk genomic data (step(1)-(9)). We then deconvolve the bulk data into copy number profiles of a set of inferred clones each with a defined ploidy and set of mixture fractions across tumor regions, as well as a phylogenetic tree relating these clones. We then compare these outputs with the ground truth data to evaluate our model. Further methodological details are provided in the text. Note that the images in this figure are purely illustrative and do not show true data from any particular analysis.

Selecting clones from SCS data: We first select copy number vectors to instantiate the normalized copy numbers in $\tilde{\boldsymbol{C}}$ and identify these as clones of the model. We use the true SCS data for this purpose. We uniformly at random select 25 single cells from each of the NUM_REGIONS regions to have 75 cells in total, of which the copy number and ploidy make nonzero contribution in the simulated bulk tumor sample later. The true copy number data of the selected cells define the columns of $\tilde{\boldsymbol{C}}$. Of the 25 single cells from each region, we denote the first 2 as *major* clones or high-frequency clones, and the remaining 23 cells as *minor*, or low-frequency, clones for that region. For each region, we model the assumption that, within the tumor, cells from the other occur but with very small frequency. Thus for each region, we designate the 50 cells from the other two regions as *tiny* clones, which will let these cells effectively serve as noise in the analysis (Fig. S2, step (1)). The two major clones from each regions to compose $\hat{\boldsymbol{C}}$, which has 6 clones in total (Fig. S2, step (2)).

Sampling ploidies: Since the real single-cell sequencing data have been normalized, the ploidy profiles for all samples have been set to 2 (diploidy) by default, and we call them $normalized\ cells$. The $normalized\ cells$ are a standard target to study tumor evolution, however, the ploidy information is also important during tumor evolution (Dewhurst *et al.*, 2014; Bielski *et al.*, 2018). Since we do not know the correspondence between ploidies and WGS copy number vectors in the ground-truth data, we sample a ploidy independently for each ground truth cell. We note that this practice may result in ploidy combinations that are biologically implausible given the phylogeny, as we only know the phylogeny for the fully-synthetic data simulation. We give each ground truth cell i a probability β_1 of being diploid, corresponding to $\hat{p}_{ii} = 1$. We give it a probability β_2 of tetraploidy, corresponding to $\hat{p}_{ii} = 1$. We then allow a probability $\beta_3 (= 1 - \beta_1 - \beta_2)$ of some other ploidy, selected uniformly from $[1, 3, 5, 6, 7, 8]$. Currently, $\beta_1 = 60\%$, $\beta_2 = 30\%$, $\beta_3 = 10\%$. Thus, at present:

$$\mathrm{P}(L = i) = \begin{cases} 0.3, & i = 4 \\ 0.6, & i = 2 \\ 0.1/6, & i \in \{1, 3, 5, 6, 7, 8\} \end{cases}$$

where L represents the ploidy number for and $P(L)$ is the probability of each ploidy number, then we have an additional tag of ploidy number for each SCS sample (Fig. S2, step (3)).

Simulating mixture fractions: We next assign mixture fractions $\tilde{\boldsymbol{F}}$ to the clones. We follow our previous work (Lei *et al.*, 2019) to use a Dirichlet distribution $Dir(\boldsymbol{\gamma})$, to assign multinomial frequencies to clones selected as in A.1.8. $\boldsymbol{\gamma}$ is a vector of concentration parameters that allows different cell components to have different contributions in the bulk tumor. The vector $\boldsymbol{\gamma}$ is generated to model that in the Dirichlet distribution, all regions have a equal prior probability of contributing to the bulk tumor. Following our previous work (Lei *et al.*, 2019), for each region, we set γ to be 100 for these major clones, 1 for the these minor clones and 0.01 for these tiny clones. Because there are three regions, we take the sum of the three vectors γ, one for each region, and use the sum as the parameters to the Dirichlet distribution. Then we retrieve the simulated mixture fractions of major clones to compose $\hat{\boldsymbol{F}}$, and normalized each column to 1 (Fig. S2, step (4)), This is used as the mixture fractions for RMSD calculation later.

Simulating bulk genomic data: Once we have defined a ground truth dataset, we simulate each source of input data for a given problem instance from this common ground truth. We first simulate bulk data from the reference model by assuming that each regions samples all clones from their ground truth proportions and with the ground truth copy number vectors and mixture fractions. That is, we simulate the input bulk matrix $\boldsymbol{B}$ as $\tilde{\boldsymbol{C}}\tilde{\boldsymbol{P}}\tilde{\boldsymbol{F}}$.

Simulating miFISH copy number profiles: We next simulate miFISH data using the genomic positions of the same NUM_PROBES loci as in the

real data. Because we require the ground truth mapping of simulated miFISH to whole-genome copy number vectors, we do not use true FISH probe counts or assigned ploidies for this simulation. We assume known absolute genomic positions of S_{begin} and S_{end} of each genomic interval in $\tilde{\boldsymbol{C}}$ and absolute genomic loci H_{begin} and H_{end} of all FISH probes according to the reference genome hg19. This provides us a way to retrieve corresponding copy number as the copy number of probes in FISH. We also save an array $Index$ mapping overlaps of SCS intervals and FISH probes for later use.

To simulate a FISH cell in a region, we use the two major clones for the region and restrict their copy numbers to the intervals overlapping the FISH probe. If the interval for a given FISH probe is included in a given SCS interval, we assign the FISH probe count to be the copy number of the corresponding SCS interval. If the FISH probe crosses two SCS intervals, we assign the FISH probe count to be a weighted average of the copy numbers of the two SCS intervals, weighted by the length of the FISH probe in each SCS interval (Fig. S2, step (6)). No FISH probe covers more than two SCS intervals in the real data, so we do not consider any other cases.

We also optionally randomly perturb copy numbers to simulate errors in FISH probe counts before transferring them to be unnormalized. This can be represented in terms of noise parameter q_f, where with probability q_f a probe count will be increased by 1, with probability q_f it will be decreased by 1 unless already zero, and with probability $1 - 2q_f$ it will be unaltered. Both before and after adding noise, the miFISH copy numbers are capped at MAX_COPY.

We repeat this process for 1000 FISH cells in each of NUM_REGIONS tumor regions to generate a simulated miFISH data set (Fig. S2, step (5)).

Simulating miFISH frequencies: We assume that the miFISH data provide an approximate measure of the distribution of mixture fractions. From the 1000 miFISH cells simulated in A.1.8, we calculate the fraction of each miFISH copy number combination for each region by calculating the proportion of each combination out of the total number of miFISH cells (1000), and then extract the mixture fractions of the first two largest clones from each region. We combine these fractions and allow the sum of each column to be less than 1, since in real data, it is possible that there would be a small proportion of cells that are not represented by the major clones. Then the resulting mixture fraction matrix $\boldsymbol{F'}$ represents the fraction of each major clone across the miFISH cells for each region, which can be used as reference for the mixture fractions of the major clones in SCS data (Fig. S2, step (7)).

Simulating SCS data: To simulate a set of SCS data, we select cells independently at random from $\tilde{\boldsymbol{C}}$ with probabilities for each cell in each region as defined in $\tilde{\boldsymbol{F}}$. The resulting SCS matrix $\boldsymbol{C'}$ would then consist of normalized single cells, where each column of $\boldsymbol{C'}$ initially corresponds to some column of $\hat{\boldsymbol{C}}$, allowing for repetition (Fig. S2, step (8)).

We further allow the data to be perturbed by a noise model with parameter q_s, where with probability q_s each copy number will be increased by 1, with probability q_s it will be decreased by 1 unless already zero, and with probability $1 - 2q_s$ it will be unaltered. Also, we would not allow for copy number to exceed MAX_COPY after perturbing the noise.

A.1.9 Fully-Synthetic Data Simulation

Since it is impossible to establish the ground truth phylogeny with certainty for the real or semi-simulated data on which we focus in this manuscript, we also created a fully simulated SCS dataset for which we would have known ground truth trees in order to better assess effectiveness of the methods and tree inference specifically. The simulation approach is based on similar SCS simulations used for the same purpose in (Lei *et al.*, 2019) but extended to include FISH data.

Simulating binary tree based on real data for each region: As described in (Lei *et al.*, 2019), we modeled the fully simulated data to approximate the true GBM data. We therefore began with the true number of tumor regions R ($R = 3$ in our case). For each region, we start from a root and create a complete binary tree by following level-order-traversal (LOT) such that the depth D of the tree (we define the depth of root is 0) is sufficient that the total number of nodes in the tree exceeds the number of cells sampled in the real SCS data in that region ($D = 6$ in our case). We modeled the estimated rate r_{ai} of copy number variation a per region ($a \in \{0, 1, ..., 10\}$), and probability p_{mi} that each genomic position has a non-diploid copy number ($m \in \{1, 2, ..., 9934\}, i \in \{1, 2, 3\}$) empirically from the real data. We started from the root, creating a copy number vectors for each node by extending the copy number profile of its parent node according to a Poisson distributed mutation model with the empirical rates r_{ai} and p_{mi} to mutate the copy number in different genomic positions so that the overall copy number distribution will be similar to that of the real SCS samples (Lei *et al.*, 2019). This yields a complete binary tree with CNA for each region as shown in Fig. S3 (a). We then sampled a subset of nodes by a walk from root to the leaves level-by-level (define the level of the root to be 0, which is the same as depth), for each node at current level, we picked its left, right or both children at the next deeper level with probability of 0.2, 0.2 and 0.6, respectively (red circles in (a), Fig. S3; if a node was not selected, all the nodes derived from such node would not be selected either). We repeated this process until we had as many selected nodes (cell samples) per region as we had in the real data, stopping and resetting to a tree with only the root node if we did not generate sufficient nodes before we exceed a level of 6. This process followed the parent-child pair convention to successively deeper nodes in the tree, establishing an adjacency matrix $\boldsymbol{T}^{(i)} (i = 1, 2, 3)$, where $\boldsymbol{T}^{(i)}_{u,v} = 1$ means node u is ancestor of node v, describing the topology of each tree (Table S1).

Simulating ploidy in the tree: The previous step yields an independent sub-tree for each region (Fig. S3 (b)) meant to mimic the characteristics of the real data. We then modeled the ploidy of the nodes in the tree as an independent process from focal CNAs. While most of this simulation is similar to that described in the *Sampling ploidies* step in Sec. A.1.8 for semi-simulated data, we modified the protocol to accommodate the fact that each child node here inherits the ploidy from the parent node: i) we made the ploidy of the child node to be equal to the ploidy of its parent if the child ploidy is less than the parent ploidy; ii) we only allowed the ploidy to be 2 or 4 with probability of 0.9 and 0.1, respectively, which yields a more biologically realistic ploidy distribution as findings in (Boisselier *et al.*, 2018).

Constructing the ground truth phylogenetic tree: We select six true and six inferred cells (nodes) from the data (two true and two inferred cells from each region) as proved effective with the real data. Unlike with the uniform selection for each of the sample in the semi-simulated data where we do not know the phylogenetic relations (Sec. A.1.8) here, we followed a set of constraints implied by parent-child pairs to select nodes that would allow us to test tree inference accuracy: i) among the selected nodes, we require that there is only one root and rest of the nodes descended from such a root, directly or indirectly, will then yield five types of quartet if we select four nodes in each region (Fig. S3 (c)); ii) we allow for the fact that some nodes of the true tree are not observed. We accomplish these goals by selecting a root node for each region and performing a random walk, assigning parent-child relations by collapsing the tree around the unobserved nodes and finding the most recent parent that was selected (lowest red circle in blue region in (b) of Fig. S3). We then manually add a diploid node as the common root of the three regions to build a ground truth phylogenetic tree encompassing the whole tumor (Fig. S3 (e)). For the four selected nodes in each region, we chose with equal probability

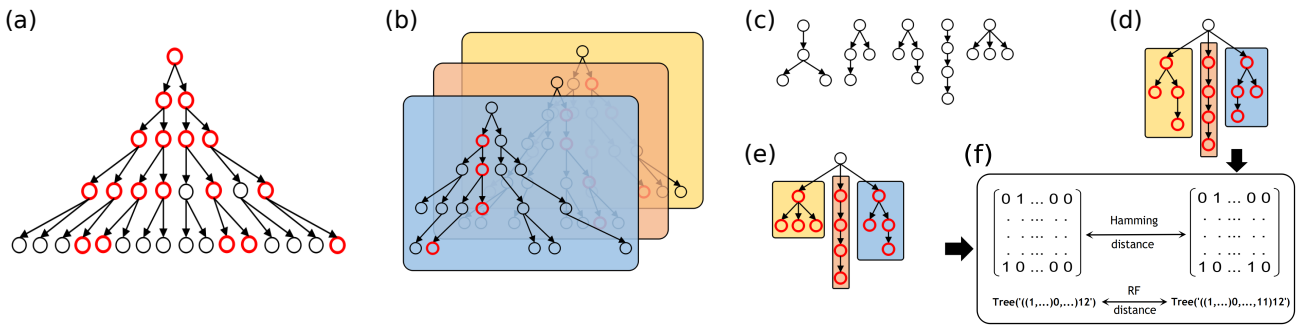

Fig. S3. Workflow of the fully-simulated data analysis. The figure shows some important steps to fully simulate SCS and FISH data and evaluate tree inferences from these data. (a) Complete binary tree for each tumor region. (b) Random walk in (a) to construct the sub-tree for each region. (c) Five types of quartet. (d) A tree example from the inference result. (e) A tree example derived from the ground truth tree by randomly picking parent-child pairs in (b). (f) Comparison between trees from (e) and (d) of matrix representation using Hamming distance (upper) and the Newick tree format representation using Robinson–Foulds distance (bottom). Note that (1) the images in this figure are purely illustrative and do not show true data from any particular analysis, (2) red circles indicates the nodes were selected for next step, and (3) different colors of the boxes in (b)(c)(d) indicate different tumor regions.

two of them as true cells and the other two as inferred cells and constructed the matrix representation $\hat{\boldsymbol{S}}$ based on previous selection and the adjacency matrix $\boldsymbol{T}^{(i)}(i = 1, 2, 3)$ from each region.

Robinson–Foulds distance for tree comparison: The Robinson–Foulds (RF) metric ((Robinson and Foulds, 1981)) is widely used to measure distances between phylogenetic trees by calculating the number of partitions in one tree that are not found in the other. We used the Python ETE3 package ((Huerta-Cepas *et al.*, 2016)) to calculate the RF distance between the true and inferred trees (Fig. S3 (f) bottom) based on common leaf nodes between the trees. Smaller RF distance means higher similarity between two trees.

Hamming distance for tree comparison: While RF distance provides a good standard metric to compare trees, it is not an ideal measure for tumor phylogeny trees in which we have labeled internal nodes that may differ between trees. There are now specialized methods for handling some of the particular challenges of comparing tumor phylogeny inference methods (DiNardo *et al.*, 2020), but none to our knowledge well suited to whole-genome copy number data like ours that cannot be easily partitioned into a discrete set of mutations. We use a comparison of the full adjacency matrices to provide a more discriminatory measurement of tree distance, specifically using the Hamming distance between the two ordered adjacency matrices. The Hamming distance between two 1-D vectors u, v is defined as:

$$d(u, v) = \frac{n_{ij}}{N} \tag{35}$$

where n_{ij} is the number of occurrences of $u[k] = i, v[k] = j, i \neq j, i, j \in \{0, 1\}, k \in \{1, ..., N\}$. Based on our definition of the adjacent matrix that only the row elements can be the parents of the column elements (Table S1), we define the Hamming distance between two matrices $\hat{\boldsymbol{S}}, \boldsymbol{S}$ as:

$$Dist(\hat{\boldsymbol{S}}, \boldsymbol{S}) = \sum_{i=1}^{K} \Big(d(\hat{\boldsymbol{S}}[i, :], \boldsymbol{S}[i, :]) \Big) \tag{36}$$

where K is the number of total nodes in one tree, and $\hat{\boldsymbol{S}}$ is the true adjacent matrix in fully-simulated data while $\boldsymbol{S}$ is the inferred adjacent matrix from the method (Fig. S3 (d), (e) and (f) top). Then smaller $Dist(\hat{\boldsymbol{S}}, \boldsymbol{S})$ means higher similarity between $\hat{\boldsymbol{S}}$ and $\boldsymbol{S}$.

We used the same process to construct bulk tumors from fully-simulated data and to infer trees from the generated data as we did for semi-simulated data in Sec.A.1.1 - A.1.8.

A.2 Supplementary Results

A.2.1 Phylogenetic tree comparison

With the real single-cell sequencing data, the true phylogenetic relationship between cells is usually unknown, limiting our ability to compare the phylogenetic outputs of our method to any certain ground truth for the semi-simulated data. For the complex combinations of copy number events at different scales that we seek to understand, there is sufficient uncertainty about the biology that even establishing realistic fully simulated data is challenging. We therefore seek to validate the effectiveness of our methods at phylogeny building more indirectly, based on the plausibility of the trees it constructs. For this purpose, we assume that the principle of minimum evolution (or maximum parsimony) should approximately hold and that the true phylogenetic tree for a given data set is likely to be one that comes close to minimizing the evolutionary changes among the taxa. Here, we define the evolutionary changes across the edges of any pair of nodes as the $L1$ distance of the unnormalized copy number profiles of the edge endpoints as we described in Sec. A.1.3, providing an indirect but informative criterion would be comparing the total distance along all the edges in different trees.

Fig. S4 shows results of a series of experiments to test the effectiveness of our model at finding trees with low evolutionary cost. For a given tree, we sum up all the edge distances in the tree tree and normalized it by dividing the total number of genomic position (9934), averaged for 10 instances, and compared these costs in different variants of our model. Since only with the phylogenetic weight turned on ($\alpha_p = 0.2$) could we get the phylogenetic output, we only compared the results from four models (as the legend shows in the Fig. S4). We also calculated the results for all the cases as we showed in Sec. 3.1.1 and Sec. 3.1.2. Comparing different subplots from (a) to (d) in Fig. S4, we can find that when the ploidy is variable and/or the noise was introduced, the total distance of the trees is somewhat increased , which is consistent with our findings on accuracy of the copy number, frequency and ploidy inferences in Sec. 3.1.1 and Sec. 3.1.2. Comparing different bars in each subplot, we find that when we add information from miFISH data, the total distance decreases significantly compared to the model that only utilized single-cell sequence data (blue bars, $\alpha_p = 0.0, \alpha_c = 0.2, \alpha_3 = 0.0$, Fig. S4), and the complete model has the minimum average distance among all the model for all the cases (coral bars, $\alpha_p = 0.2, \alpha_c = 0.2, \alpha_3 = 0.2$, Fig. S4). This again confirms, though indirectly, that miFISH information helps in phylogenetic inference and improves performance relative to a model omitting miFISH data, and further that the complete model performed the best in inferring parsimonious trees.

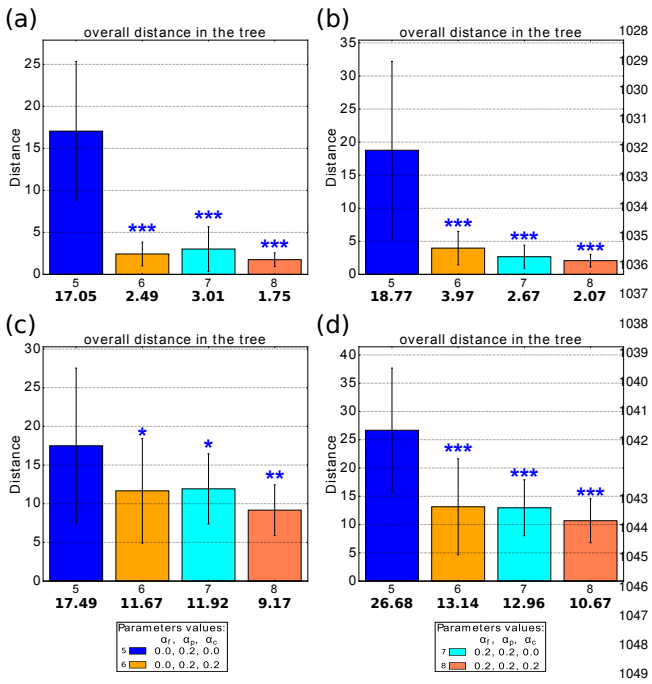

Fig. S4. Average total distance in the phylogenetic trees (n=10). Each bar with different color represents a deconvolution model distinguished by the data of which it took advantage. In the legend at the bottom, the numbers represent the values of the mixture fraction weight, phylogenetic weight, and copy number weight, which are coefficients for $||\boldsymbol{F}-\boldsymbol{F'}||$, $J(\boldsymbol{S},\boldsymbol{C},\boldsymbol{C'})$ and $||\boldsymbol{X^T}\boldsymbol{C}\boldsymbol{P}-\boldsymbol{H'}||$, respectively. 0.0 means the corresponding term is not included in the model, and we only show the results that have the phylogenetic output ($\alpha_p = 0.2$). The number in the first row under the bar indicates different model, the number in the second row under the bar indicates the mean of each model and the whiskers show the standard deviation. Statistically significant improvements from incorporating FISH data were assessed by paired sample *t*-test, comparing orange(6), cyan(7), coral(8) bars to the single-cell only model (blue bar(5)) (*: $0.05 <$ *p*-value≤ 0.1, **: $0.01 <$ *p*-value≤ 0.05, ***: *p*-value≤ 0.01). (a) Without ploidy change and without noise. (b) Without ploidy change and with 10% noise. (c) With ploidy change and without noise. (d) With ploidy change and with 10% noise.

A.2.2 Deconvolution without SCS Data

We initially tested our model in the scenario where we ido not have real SCS data but only have miFISH available. To incorporate the tree part of the objective function, we made an artificial reference cell matrix with all diploid copy number for every entry. We did the same process as described in Sec. 3. Fig.S5 shows the average result. From the top to bottom are the results without noise and without ploidy change, with 10% noise but without ploidy change, without noise but with ploidy change, with 10% noise and with ploidy change, respectively. We found that, compared to the results in Fig. 1 (a) and Fig. 2 (a), the performance was worse for most of the cases. This observation suggests that the real SCS data plays an important role in the reference, which is consistent with the conclusion of our previous work (Lei *et al.*, 2019). However, this loss is not obvious if we do not perturb the ploidy, as assuming diploid reference cells effectively provides an informative prior probability for the inference. When we implemented the change of ploidy, the difference of performance with and without real SCS became evident. Nonetheless, the addition of miFISH data substantially improves accuracy relative to inference from bulk sequence data alone.

A.2.3 Deconvolution with different number of iterations

As mentioned in Sec. 3.1.1, our current model reduced the maximum number of iterations for the Gurobi solver from 100 to 10 relative to our earlier work, as we found that increasing the number of iterations could greatly increase run time while generally not substantially improving our quantitative measures of performance. Here, we evaluated the effects of this change by showing performances with two different maximum numbers of iterations (Fig. S6) in the case of 10% noise and with variable ploidy 3.1.2. In all cases, optimization may terminate before the maximum number of iterations based on the convergence test of Algorithm 1. The cyan box shows the results of maximum iteration $= 10$ and violet box shows the results of maximum iteration $= 100$. We can see that though there is some variation between each pair of results, the average values showed no consistent pattern of improvement with increasing numbers of iterations and no significant difference between the two. While additional rounds of optimization did sometimes lead to better solutions, the results suggest that improvement was generally small and that further refinement of the objective function does not reliably translate to better solutions as assessed by our performance measures.

A.2.4 Effects of different initialization schemes

We tested the effect of different initialization schemes on the effectiveness of the present method as well as in comparison to our previous work ((Lei *et al.*, 2019)). We created 10 instances of semi-simulated data and set up 5 different experiments as shown in Table S2. In Case 1 and Case 3, we initialized $\boldsymbol{C}$ with 0 and random real single-cell data, respectively and applied the phylogeny-based method as described in (Lei *et al.*, 2019). In Case 2 and 4, we we initialized $\boldsymbol{C}$ with 0 and random real single-cell data, respectively and applied the method in this paper. Since the method in (Lei *et al.*, 2019) did not infer the ploidy, we did another experiment (Case 5) in which we modified the code run the ploidy inference, but then ignore the result and reset the ploidy to diploid for every iteration, (as mentioned in Sec. A.1.5) thereby eliminating the effect of ploidy inference while verifying that the ploidy inference had no unexpected side-effects. Also, for the same reason, when we simulated the data, we assumed the ploidy for each cell clone to be 2 rather to be random and we did not introduce noise into the simulated data. In other words, the process here is the same as we describe in Sec. 3.1.1 but we only compared the performance on the copy number inference and frequency inference.

Case 1	phylogeny-based method in (Lei *et al.*, 2019), using **0** to initialize $\boldsymbol{C}$
Case 2	method in this paper, using **0** to initialize $\boldsymbol{C}$
Case 3	phylogeny-based method in (Lei *et al.*, 2019), using **real single-cell** data to initialize $\boldsymbol{C}$
Case 4	method in this paper, **real single-cell** data to initialize $\boldsymbol{C}$
Case 5	method in this paper, **real single-cell** data as to initialize $\boldsymbol{C}$, forcing ploidy to be 2

Table S2. Different experimental cases for initialization comparison

For all of the cases, we did not introduce any penalty (dubbed the NULL model: $\alpha_f = 0.0, \alpha_p = 0.0, \alpha_c = 0.0$) in order to just test the effect of initialization. As shown in Fig. S7, we found that using real single-cell data as initialization, the overall performance is better (comparing cyan bars with blue bars in Fig. S7), which is consistent with our findings in (Lei *et al.*, 2019), while the results from the method in (Lei *et al.*, 2019) and this paper did not show significant difference (comparing Case 2 with Case 1, comparing Case 4, 5 with Case 3, respectively in Fig. S7 by paired sample *t*-test).

Although using real single-cell data would yield better performance in the NULL model, we found that such initialization is vulnerable to noise (results not shown). One possible reason would be that the miFISH information is still much less informative than SCS information even after extending it to correlated adjacent regions (Sec. A.1.7). Therefore, when

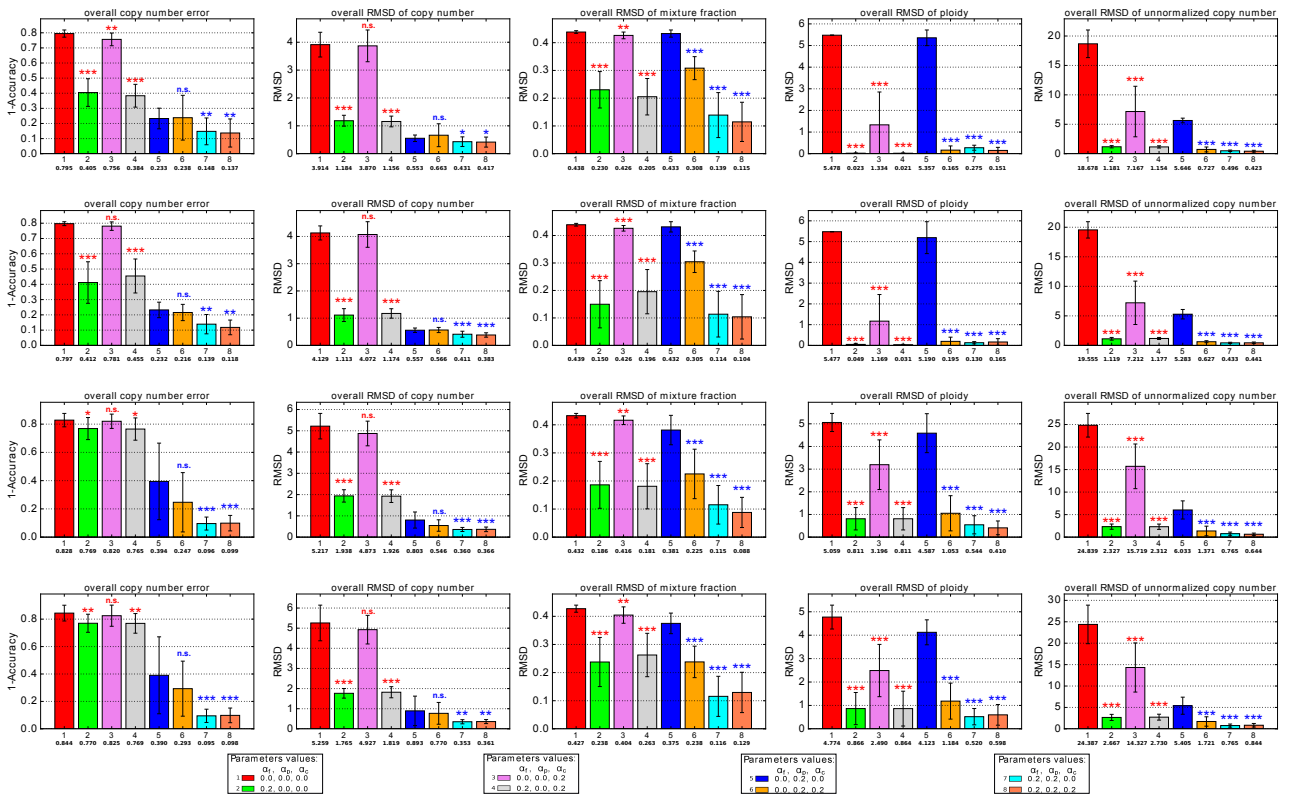

Fig. S5. Average accuracy and RMSD of the deconvolution without real SCS data. From top to bottom are the results without noise and without ploidy change, with 10% noise but without ploidy change, without noise but with ploidy change, and with 10% and with ploidy change, respectively. In each subplot, the barplot from left to right shows the average error (1-accuracy) of copy number, average RMSD of copy number, average RMSD of mixture fraction, average RMSD of ploidy and average RMSD of unnormalized copy number. All the labels are the same and the numbers represent the values of the mixture fraction weight, phylogenetic weight, and copy number weight, which are coefficients for $||\boldsymbol{F} - \boldsymbol{F'}||$, $J(\boldsymbol{S}, \boldsymbol{C}, \boldsymbol{C'})$ and $||\boldsymbol{X^T C P} - \boldsymbol{H'}||$, respectively. 0.0 means the corresponding term is not included in the model. The number in the first row under the bar indicates different model, the number in the second row under the bar indicates the mean of each model and the whiskers show the standard deviation. Statistically significant improvements from incorporating FISH data were assessed by paired sample *t*-test, comparing green(2), pink(3) and gray(4) bars to the NULL model (red bar(1)) and orange(6), cyan(7), coral(8) bars to the single-cell only model (blue bar(5)) (n.s.: not significant, *: $0.05 <$ *p*-value≤ 0.1, **: $0.01 <$ *p*-value≤ 0.05, ***: *p*-value≤ 0.01)

we introduce small amounts of noise to the real single-cell data before using it for initialization, we actually perturb the inference substantially. We choose not to focus on this single-cell initialization approach as our default in this work because the effect confounds the effect of miFISH information, which is the major focus of the present work. In addition, since single-cell data are usually noisy and sometimes limited in quantity or unavailable altogether, we prefer not to make additional assumptions on the initialization. We therefore in the present work focus primarily on results from initializing with zeroes in the main paper or with all-diploid initial guesses, as mentioned in Sec. A.2.2.

A.2.5 Sensitivity to parameter changes

In the previous sections, we turned on or off the three weights $(\alpha_f, \alpha_p, \alpha_c)$ by setting them either to 0.2 or 0.0. We chose 0.2 heuristically as a good default value for similar regularizations in our previous work (Lei *et al.*, 2019). In this section, we explored the question of sensitivity of the parameters to determine whether the results would be highly dependent on parameter choices. To evaluate this, we performed a parameter scan around the value of 0.2 to test different combinations of the three parameters, focusing specifically on the case of 10% noise and with variable ploidy. In the set of parameter combinations, we found that the model was minimally sensitive to changes of parameters in the measurement of normalized copy number but somewhat more sensitive to the change of parameters in the measurement of frequency, ploidy and unnormalized copy number (Fig. S8). For example, when we fixed the copy number weight, α_c, to be 0.1, the average performances in each heatmap did not change much in copy number inference (1^{st} and 2^{nd} rows in Fig. S8) but showed more oscillation in the rest of measurements when we increased the mixture fraction weight, α_f, and/or the phylogenetic weight, α_p, ($3^{rd}, 4^{th}$ and 5^{th} rows in Fig. S8). When we fixed α_f and α_p, we observed that ploidy inference did not reveal a simple pattern of better or worse average performance across different combinations of parameters, which indicates that the parameters may influence performance in a more complicated way. Further, there was no single ideal parameter set for all measures, but rather improvement by different measures with different parameter variations. Nevertheless, the default setting of parameters $(\alpha_f = 0.2, \alpha_p = 0.2, \alpha_c = 0.2)$ seems to yield a good consensus that provides a reasonable set of trade-offs in the performance across all the measurements (3^{rd} column in the middle of Fig. S8).

A.2.6 Robustness to real data

To test the robustness of our method to random variation in data, we conducted an analysis of sub-samples of the real GBM data. In each experiment, we sample 80% of the real SCS and FISH samples without replacement, then perform k-median clustering as described in Sec. 3.2. We still utilized the predefineded parameters $(\alpha_f = 0.2, \alpha_p = 0.2, \alpha_c =$

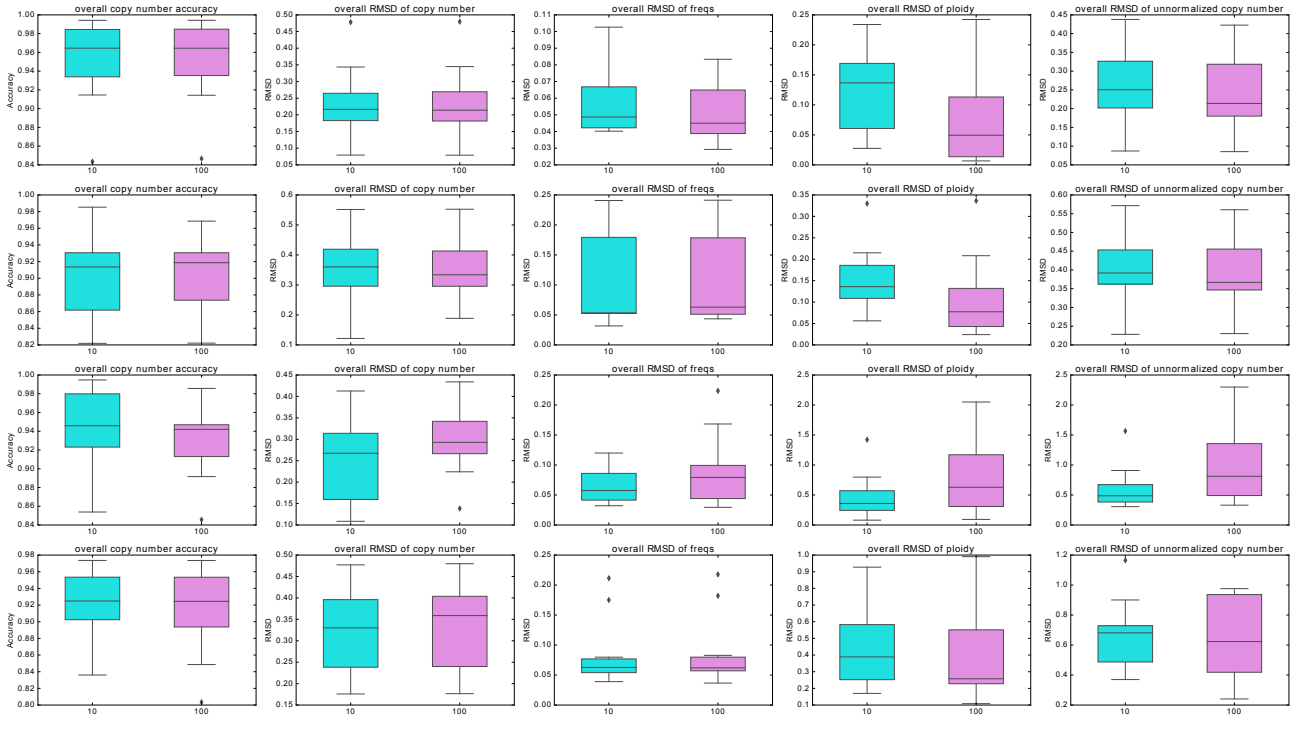

Fig. S6. Performance comparison for varying numbers of maximum iterations of optimization. This figure shows differences in performance for 10 versus 100 maximum iterations of optimization in the case of 10% noise and variable ploidy. Cyan boxes represent results of maximum iterations $= 10$ and violet boxes represent results of maximum iterations $= 100$. From the left to the right, we present the performance comparison in overall copy number accuracy, overall RMSD of copy number, overall RMSD of frequency, overall RMSD of ploidy and overall RMSD of unnormalized copy number, respectively.

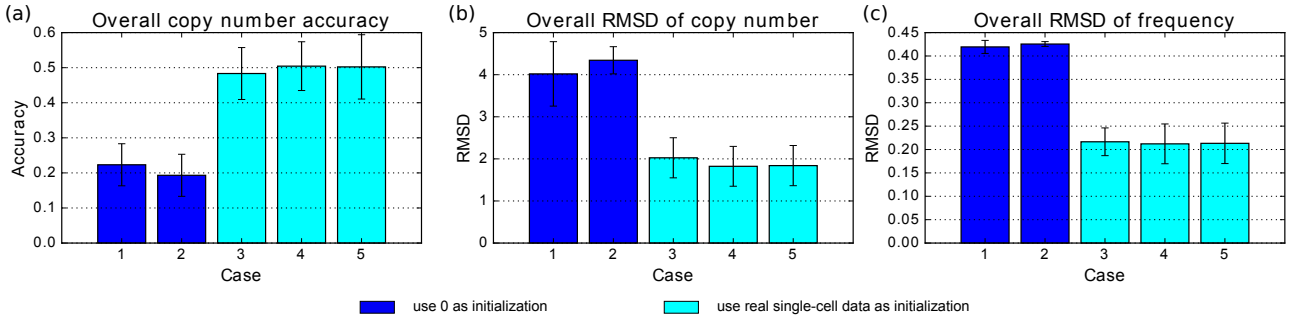

Fig. S7. Comparison between the method in (Lei *et al.*, 2019) and this paper. (a) Overall accuracy in copy number. (b) Overall RMSD in copy number. (c) Overall RMSD in frequency. Blue bars indicate the results using 0 as initialization. Cyan bars indicate the results using random real single-cell data as initialization. Bars of Case 1 and 3 stand for the results from phylogeny-based method in (Lei *et al.*, 2019) while bars of Case 2, 4, and 5 stand for the results from the method in this paper. All the tests were run on the same data.

0.2) and run 40 replicates on GBM07 data set. When we calculated the mean and standard deviation for all the replicates, we chose the inferred copy number matrix X_1 from the first replicate as a standard and reorder the column (the order of the cell components) of other replicates to get the ordered index $\mathcal{O}_i$ such that L_1 distance between X_1 and $X_i, (i \in \{2, ..., 40\})$ is minimal. We then use $\mathcal{O}_i$ to reorder the rows of the mixture fraction and ploidy matrices (the order of the cell components) and get the mean and standard deviation for each cell component across all experiments.

We provide here an expanded version of Fig. 4, Fig. S9. The copy number results on subsampled data shows that for the majority genomic loci, the standard deviation is less than 0.5, which indicates the inference consistency of our method (Fig. S10 (a)). We also found that at some specific genomic loci, the variation is much larger. These loci are usually located on chromosomes 7, 9, 10 (e.g. Fig. S10 (b)). We believe this is due to high levels of variation in these chromosomes leading to heterogeneity within defined cell clones, which is consistent with our previous analysis in (Lei *et al.*, 2019) and Sec. 3.2. We note that the pattern of mixture fractions of cell components appears different from the representative in Fig. S9 (e) in part because the order cell components has been changed. We also note the variance can be relatively large, however, this is not beyond our expectation since sub-sampling the real data produces clusters with somewhat different inferred mixture fractions. This also reinforces the importance of accurate prior mixture fraction information from miFISH data, as we mentioned in Sec. A.1.2 and Sec. 3. Nevertheless, we can still clearly see that different cell components take on distinct proportions in the tumor, which also explains the intra-tumor heterogeneity (Fig. S10 (c)). The ploidy inferences are more stable, as we can see there are

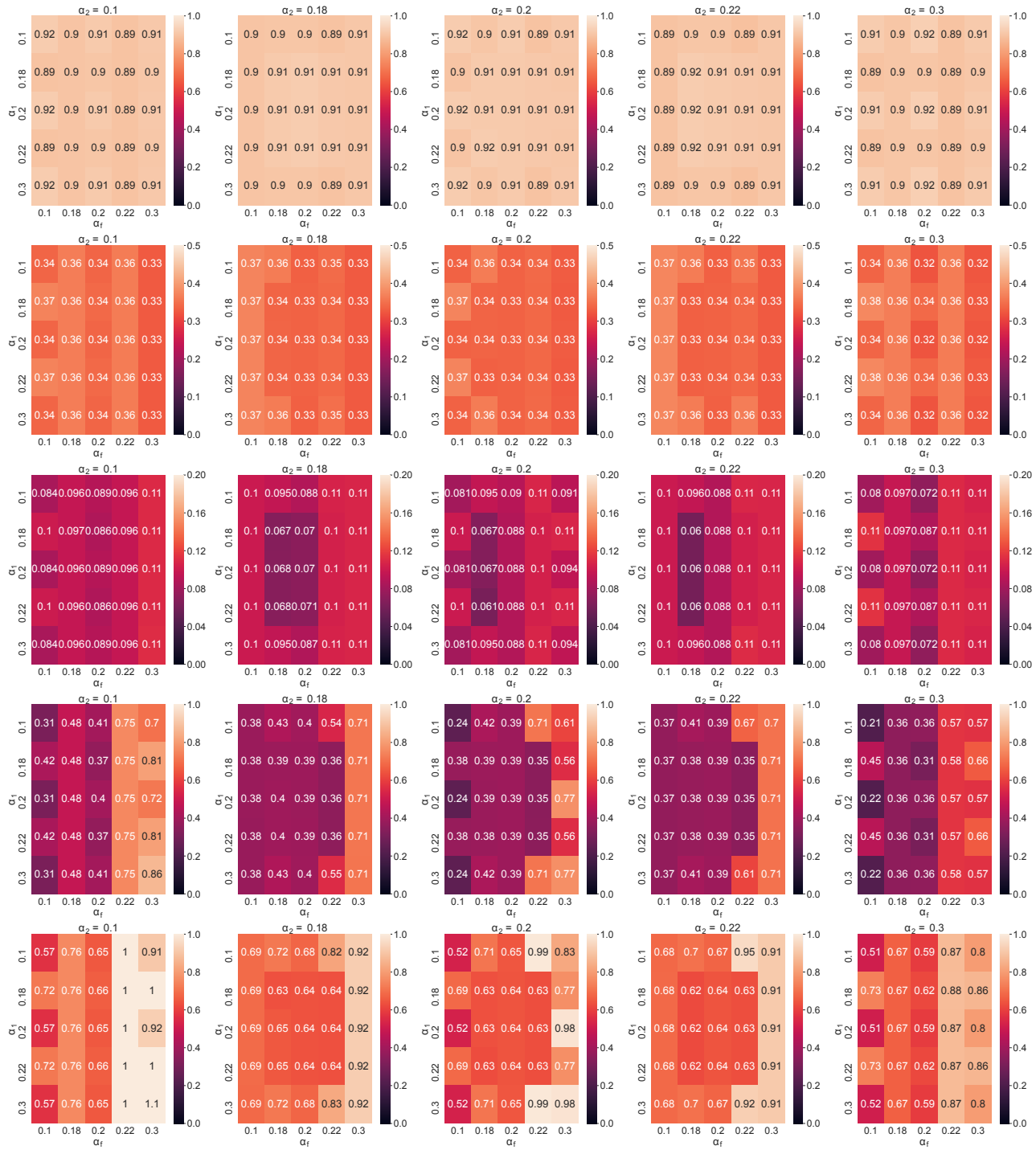

Fig. S8. Model performance for different combinations of parameters. This figure shows the results of sensitivity tests on semi-simulated data, where we vary the value of α_f, α_p and α_c to be 0.1, 0.18, 0.2, 0.22, 0.3, respectively, to form $5 \times 5 \times 5 = 125$ combinations of all the there parameters. From the top to the bottom, we present the performance in overall copy number accuracy, overall RMSD of copy number, overall RMSD of frequency, overall RMSD of ploidy and overall RMSD of unnormalized copy number, respectively. In each case, we modeled 10% noise and variable ploidy. The value in each block represents the average performance of $n = 10$ experiments.

diploid, (pseudo) triploid and tetraploid cell components in the inferences (Fig. S10 (d)). These results demonstrate the robustness of our method on real data and further reinforce the importance of integrating information from different data types.

A.2.7 Deconvolution using fully-simulated data

In this section, we analyzed the results from application of our method to fully-simulated data, as described in Sec. A.1.9. We first evaluated the average error (1-accuracy) of copy number, average RMSD of normalized copy number, average RMSD of mixture fraction, average RMSD of ploidy, and average RMSD of unnormalized copy number. These tests were

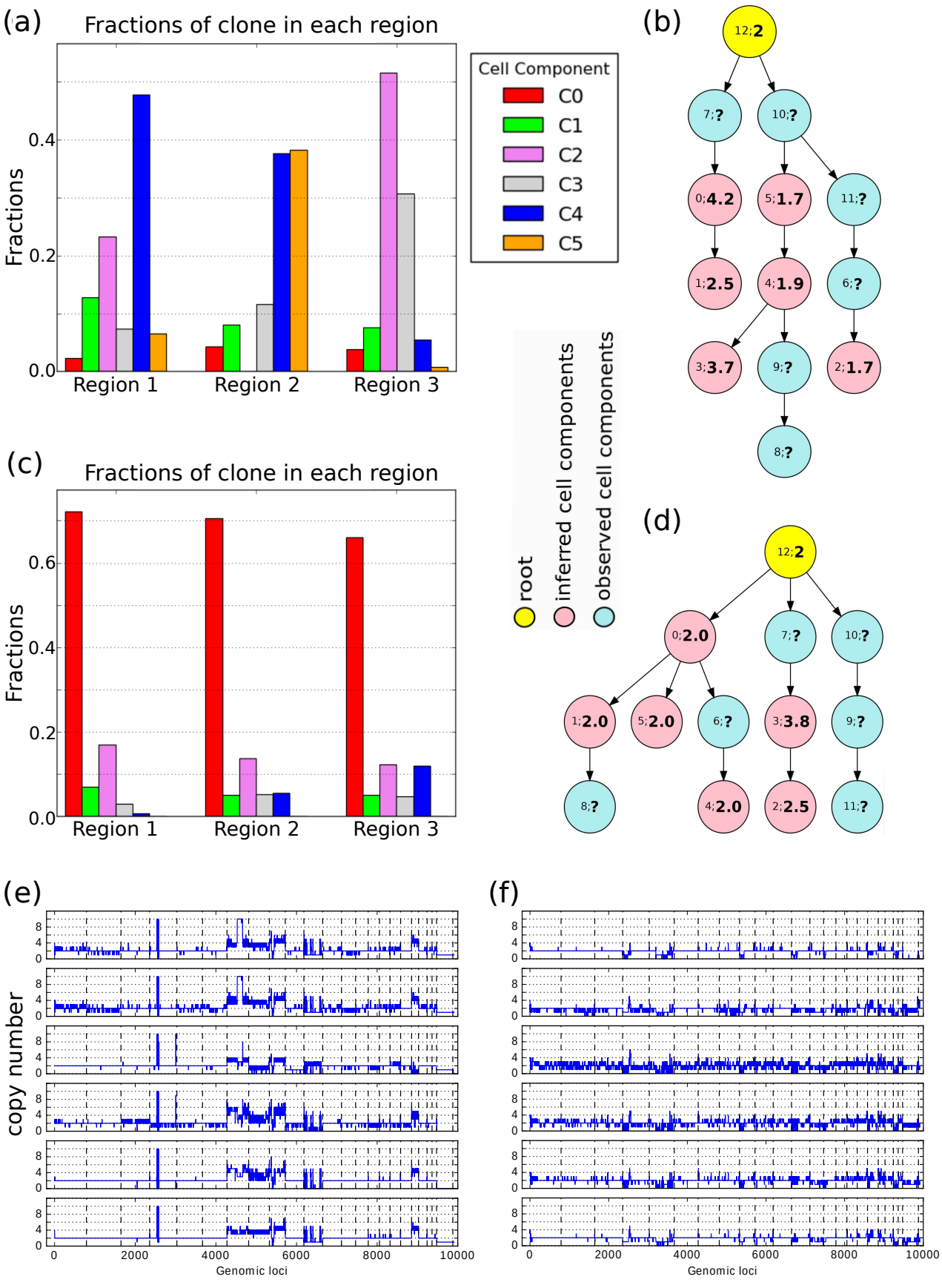

Fig. S9. Expanded version of Fig. 4. (a), (c): The corresponding mixture fraction of each inferred cell component. (b), (d): The phylogenetic relationship among the inferred cell components (pink) and observed cell components (light blue). (e), (f): The copy number of each chromosome in inferred cell component C0 (top) to C5 (bottom). The X-axis corresponds to the genomic loci and the intervals between vertical dashed lines indicate chromosomes 1 to 22, X and Y.

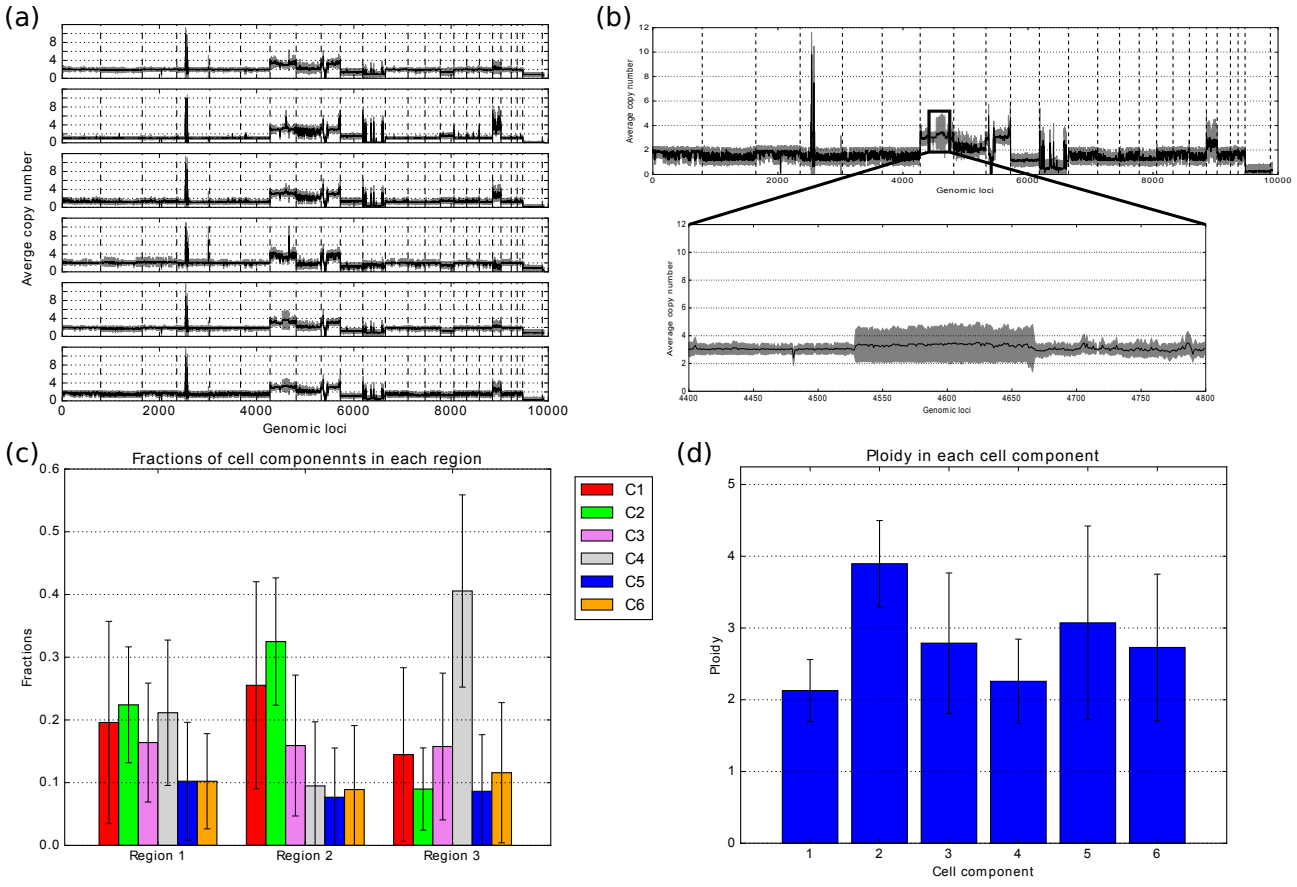

Fig. S10. Robustness to sub-sampling GBM data. (a) Average copy numbers with standard deviation as shade. From top to bottom are cell components 1 to 6. The X-axis corresponds to the genomic loci and the intervals between vertical dashed lines indicate chromosomes 1 to 22, X and Y. (b) Copy numbers of cell component 6 with a zoom-in window showing the high variations on chromosome 7. (c) Average clonal mixture fraction results for each cell component in different regions in GBM, different bars indicate different cell components (d) Ploidy results for each cell component.

conducted for four data models (with and without noise, with and without ploidy changes) as in Sec. 3.1.1 and 3.1.2. The results are qualitatively consistent with those of we found with semi-simulated data: miFISH data reliably improves the deconvolution accuracy, and the complete model $(\alpha_f = 0.2, \alpha_p = 0.2, \alpha_c = 0.2)$ performed the best in all cases (Fig. S11 (a)). One notable observation is that the model only using single-cell data $(\alpha_f = 0.0, \alpha_p = 0.2, \alpha_c = 0.0)$ achieved higher average performance for fully simulated than for semi-simulated data (Fig. S11 blue bars). One possible reason is that the fully simulated data imposes some constraints on clonal phylogenies to maintain parent-child pair relations when we match true and observed cells, as described in Sec. A.1.9, while we are necessarily unsure about such relationships in semi-simulated data. When the selected samples have simple phylogenetic relationships, our phylogenetic penalty $J(\boldsymbol{S}, \boldsymbol{C}, \boldsymbol{C'})$ would be expected to reconstruct them more easily than if the true trees were more complicated than our simulated model assumes, for example if a substantially larger number of unobserved ancestral clones were needed to explain the relationships between the observed cells. This again confirms that the application of phylogenetic penalty $J(\boldsymbol{S}, \boldsymbol{C}, \boldsymbol{C'})$ is reasonable in inferring tumor progress since tumor progress is generally a clonal evolutionary model that has parent-child pair relations ((Nowell, 1976)), while suggesting that more sophisticated phylogenetic models might lead to better performance with more complicated phylogenies that might be found in real data.

We also conducted an the indirect assessment of the phylogenetic output by calculating the total evolutionary distance ($L1$ distance) along all the edges of the tree. We again found that the results are qualitatively comparable to those for semi-simulated data (Fig. S4) but that the average performance is also quantitatively somewhat better for full simulated than semi-simulated data in many cases (Fig. S11 (b)). In the direct assessment of the phylogenetic output by calculating the Hamming distance between matrix representations of true and inferred trees (Fig.S11 (c)) and the Robinson–Foulds (RF) distance between true and inferred trees (Fig.S11 (d)), we see that integrating more information from miFISH in addition to single-cell data leads to inference of more similar trees (smaller distance) relative to the known ground truth by both measures. Further, this improved inference is also robust to ploidy changes and noise perturbation.

We also plot one example of a simulated tree and its inference by variants of our method, shown in Fig. S12. We chose an example with ploidy change and without noise to provide a clearest illustration. Compared with the ground truth (Fig. S12 (a)), we find that the tree from the model only with single-cell sequence data $(\alpha_f = 0.0, \alpha_p = 0.2, \alpha_c = 0.0)$ tends to partition inferred and observed cell components separately from one another and to infer some large ploidies not found in the true tree, similar to the results from semi-simulated data. When we integrate different components of the information from miFISH data, we find the branches of the tree and the ploidy of the nodes become more accurate, which is also consistent with the results from semi-simulated data

(Fig. 3). With the complete model ($\alpha_f = 0.2, \alpha_p = 0.2, \alpha_c = 0.2$), we find more parent-child pair relations were restored even though the reconstruction is still not perfect (Fig. S12 (e)). Our deconvolution results above suggests the fully-simulated data is reconstructed somewhat more accurately than semi-simulated data and thus may not truly capture the complexity of the real data. Nonetheless, these tests together with those on semi-simulated and real data, provide evidence that bringing miFISH data into the analysis can improve inference of phylogenetic relationships over single cell sequence data alone.

A.2.8 Comparison to MEDALT

We compared the phylogenetic inference part of our method to MEDALT (Wang *et al.*, 2021), which is a new method to infer phylogenetic trees from single-cell copy number data. MEDALT does not use bulk data or fluorescence in situ hybridization data, but provides a basis for comparison that can be run on a subset of the data used by our method. Therefore, we gave as input only the single-cell copy number part of our simulated data. For all MEDALT parameters that have default values, we used the default values. MEDALT does not have a default for the reference genome, so we selected hg19, which matches our data. We analyzed 10 replicates for each of two of our simulation model: i) no noise and no ploidy change ii) added noise but no ploidy change.

In Fig. S13, we show by example that we could get MEDALT to run and produce output (panel a) and we could convert the MEDALT output format to our alternative tree representation, showing node numbers and ploidies in each node (panel b). For purposes of comparing the example output, we show the true tree underlying the simulation of this replicate (panel c) and the tree that our method infers (panel d). In this example, the tree inferred by MEDALT (panel b) is much shallower and broader than the true tree (panel c), whereas our tree (panel d) is one level deeper than the true tree but closer to it in structure (panels c,d). We found that MEDALT tends to produce shallow trees on our input data. We quantified the comparative accuracy of the two methods in three ways: overall distance between the inferred trees and the true trees (panel e left), Hamming distance between the true trees and the MEDALT trees (panel e middle), and Robinson-Foulds distance (panel e right). Our method shows substantially better results than MEDALT by all three measures. We caution that this result should not be interpreted as a criticism of MEDALT, as our method is intended to use additional data unavailable to MEDALT. Rather, it shows the value of these additional data sources to accurate phylogenetic inference in situations where single-cell data is limited.

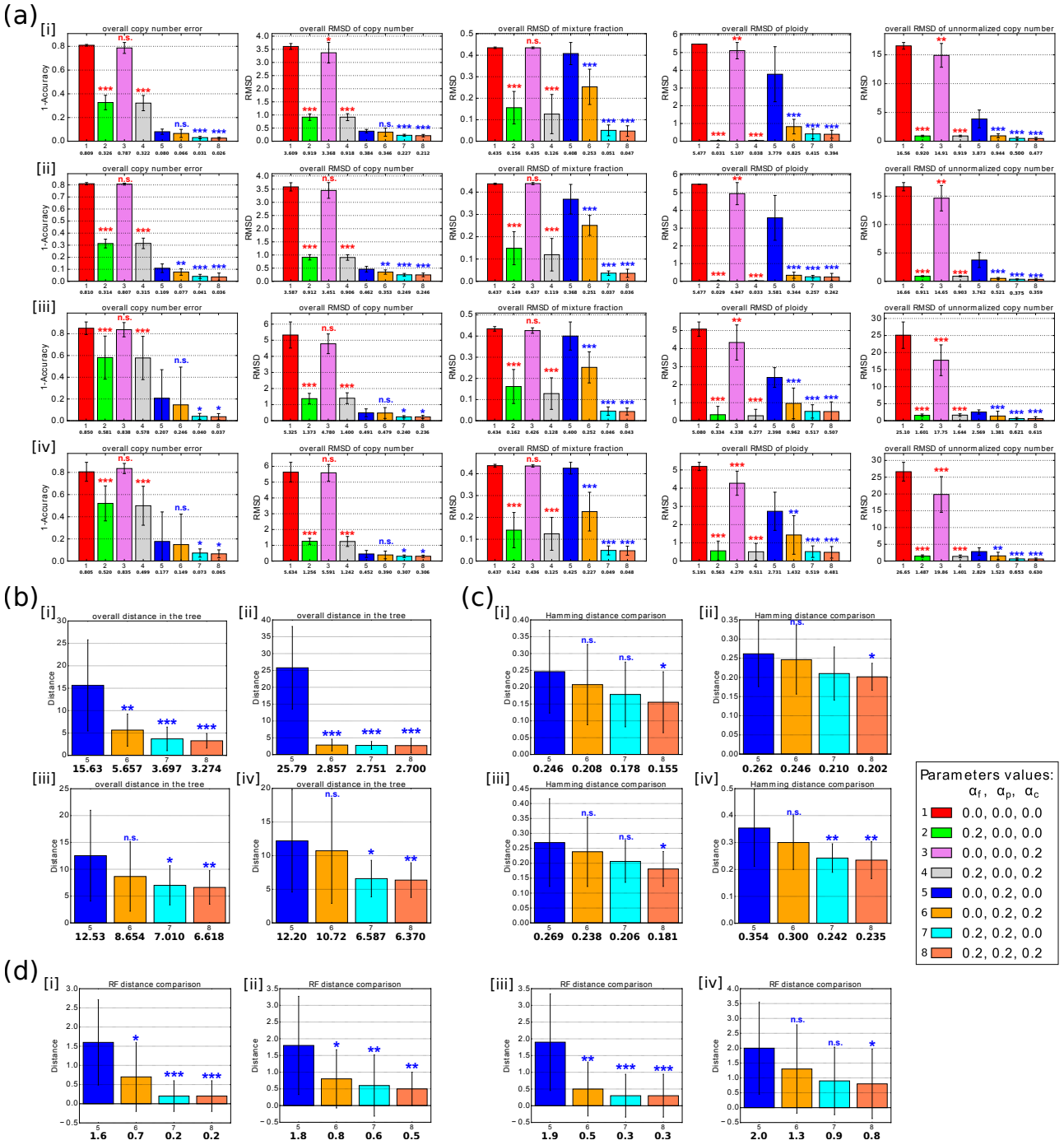

Fig. S11. Performance of deconvolution with fully-simulated data. (a) Overall performance across $n = 10$ instances by using the metrics in Sec. 3.1.1 and 3.1.2. (b) Overall distance along all edges in the tree by using the metric in Sec. A.2.1. (c) Hamming distance between the true phylogenetic tree and the inferred phylogenetic trees. (d) Robinson–Foulds (RF) distance between the true phylogenetic tree and the inferred phylogenetic trees. In each subplot of (a),(b),(c) and (d), [i]-[iv] indicate the results without noise and without ploidy change, with 10% noise but without ploidy change, without noise but with ploidy change, and with 10% and with ploidy change, respectively. All the labels are the same and the numbers represent the values of the mixture fraction weight, phylogenetic weight, and copy number weight, which are coefficients for $||\boldsymbol{F} - \boldsymbol{F'}||$, $J(\boldsymbol{S}, \boldsymbol{C}, \boldsymbol{C'})$ and $||\boldsymbol{X}^T\boldsymbol{C}\boldsymbol{P} - \boldsymbol{H'}||$, respectively. 0.0 means the corresponding term is not included in the model. The number in the first row under the bar indicates different model, the number in the second row under the bar indicates the mean of each model and the whiskers show the standard deviation. Statistically significant improvements from incorporating FISH data were assessed by paired sample *t*-test, comparing green(2), pink(3) and gray(4) bars to the NULL model (red bar(1)) and orange(6), cyan(7), coral(8) bars to the single-cell only model (blue bar(5)) (n.s.: not significant, *: $0.05 <$*p*-value≤ 0.1, **: $0.01 <$*p*-value≤ 0.05, ***: *p*-value≤ 0.01). Please note that in the performance involved with tree ((b),(c) and (d)), we could only get phylogenetic output when the term of $J(\boldsymbol{S}, \boldsymbol{C}, \boldsymbol{C'})$ was enabled, so we only have four models (5, 6, 7, 8) for the comparison in each case.

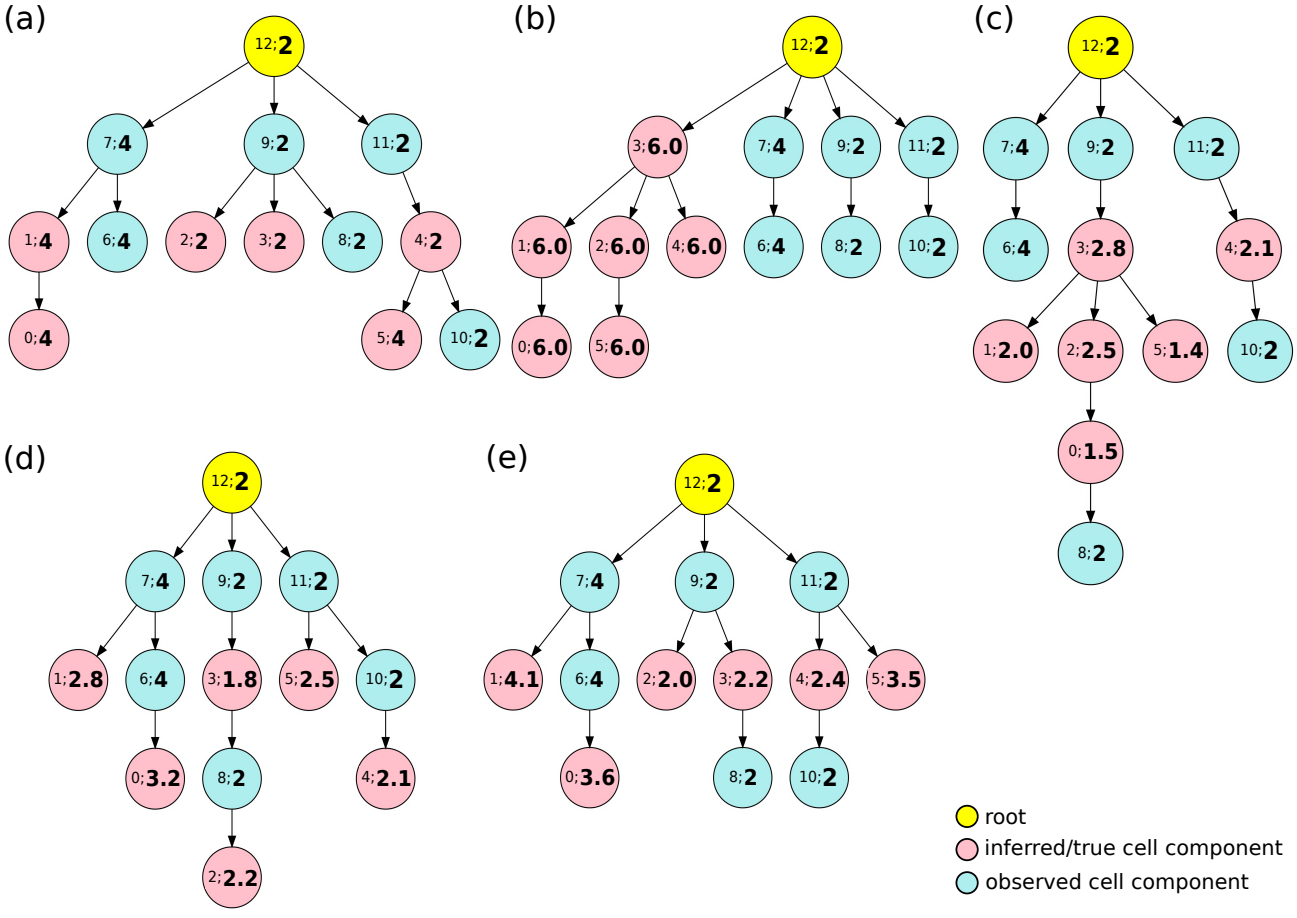

Fig. S12. Phylogenetic output example from fully-simulated data with ploidy change and without noise. (a) Ground truth example in the fully-simulated data. (b) Tree example from single-cell sequence-only model $(\alpha_f = 0.0, \alpha_p = 0.2, \alpha_c = 0.0)$. (c) Tree example from model with coefficients of $J(\boldsymbol{S}, \boldsymbol{C}, \boldsymbol{C'})$ and $||\boldsymbol{X^T CP} - \boldsymbol{H'}||$ $(\alpha_f = 0.0, \alpha_p = 0.2, \alpha_c = 0.2)$. (d) Tree example from model with coefficients of $||\boldsymbol{F} - \boldsymbol{F'}||$ and $J(\boldsymbol{S}, \boldsymbol{C}, \boldsymbol{C'})$ $(\alpha_f = 0.2, \alpha_p = 0.2, \alpha_c = 0.0)$. (e) Tree example from the complete model $(\alpha_f = 0.2, \alpha_p = 0.2, \alpha_c = 0.2)$. The yellow node represents a diploid root cell, the pink nodes represent inferred cell components in from the method or true cell components of the ground truth, and the light blue nodes are observed cell components. The number pair inside each node provides $NodeIndex$; ***Ploidy***. Note that all the nodes in the ground truth tree (a) have integer ploidy since our simulations only used integer ploidy values (Sec. A.1.9), even though the inference method allows for fractional ploidy (Sec. A.1.5).

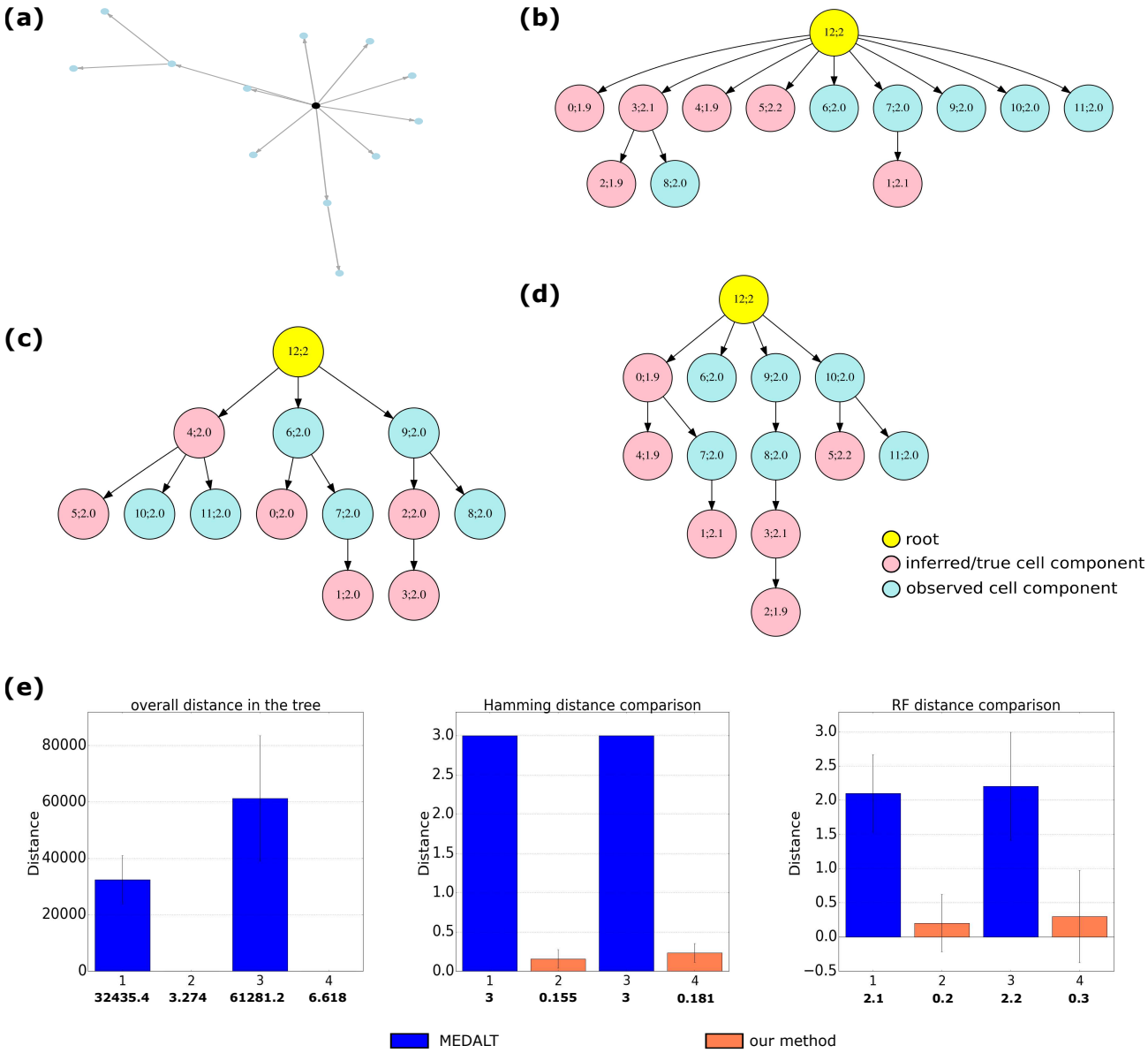

Fig. S13. Performance comparison with MEDALT. (a)-(d) Visualization of a tree example. (a) original output from MEDALT. (b) transform MEDALT tree output to the tree representation in this paper. (c) corresponding true tree. (d) corresponding tree output from inference in this paper. The number pair inside each node represents NodeIndex; Ploidy. (e) presents the tree distance comparisons between MEDALT and our method. From left to right, the measures are Overall distance along all edges in the tree by calculating L1 distance between node pairs, Hamming distance between the true phylogenetic tree and the inferred phylogenetic trees and Robinson–Foulds (RF) distance between the true phylogenetic tree and the inferred phylogenetic trees. In each subplot, bars 1 and 2 present results without noise and without ploidy change while bars 3 and 4 present results without noise but with ploidy change. The bold floating point number under each bar is represent the average among 10 replicates; the whiskers in each bar represent the standard deviation.