SUPPLEMENTARY MATERIAL: PROBABILISTIC LEARNING OF TREATMENT TREES IN CANCER

CONTENTS

S1. Proof of Proposition 1 We provide a proof for a tree with four leaves (see Figure S1) and extension to trees with a larger number of leaves follows by induction. The main idea is to merge subtrees backward and integrate out responses of internal nodes when merging subtrees.

PROOF. Consider a subtree $\mathcal{T}'$ rooted at $(t_1, \boldsymbol{X}_1')$ with two leaves $(1, \boldsymbol{X}_1)$ and $(1, \boldsymbol{X}_2)$, and one internal node $(t_2, \boldsymbol{X}_2')$ (see Panel (A) of Figure S1). Assume that the root $(t_1, \boldsymbol{X}_1')$ of the subtree is fixed, and responses $\boldsymbol{X}_i, \boldsymbol{X}_i' \in \mathbb{R}^J, J \geq 1, i = 1, 2$. With $\boldsymbol{t} = (t_1, t_2, t_3)^\mathsf{T}$, the conditional distribution for leaf responses would be $\boldsymbol{X}_i | \boldsymbol{X}_2', \mathcal{T}, \boldsymbol{t} \sim N_J(\boldsymbol{X}_2', (1-t_2)\sigma^2 \boldsymbol{I}), i = 1, 2$. Since $\boldsymbol{X}_2' | \boldsymbol{X}_1', \mathcal{T}, \boldsymbol{t} \sim N_J(\boldsymbol{X}_1', (t_2 - t_1)\sigma^2 \boldsymbol{I})$, based on the conjugacy of the normal distribution, the marginal distribution is also normal. Conditional on $\boldsymbol{t}$ and $\mathcal{T}$, mean and covariance of $\boldsymbol{X}_i, i = 1, 2$ can be derived by the law of iterated expectations and results in the distribution of the subtree $\mathcal{T}'$ with two leaves:

$$E[\boldsymbol{X}_i] = E[E[\boldsymbol{X}_i|\boldsymbol{X}_2']] = E[\boldsymbol{X}_2'] = \boldsymbol{X}_1', \quad i = 1, 2;$$

$$Var[\boldsymbol{X}_i] = Var[E[\boldsymbol{X}_i|\boldsymbol{X}_2']] + E[Var[\boldsymbol{X}_i|\boldsymbol{X}_2']] = Var[\boldsymbol{X}_2'] + E[(1-t_2)\sigma^2\boldsymbol{I}] = (1-t_1)\sigma^2\boldsymbol{I}_J;$$

$$Cov[\boldsymbol{X}_1, \boldsymbol{X}_2] = Cov[E[\boldsymbol{X}_1|\boldsymbol{X}_2'], E[\boldsymbol{X}_2|\boldsymbol{X}_2']] + E[Cov[\boldsymbol{X}_1, \boldsymbol{X}_2|\boldsymbol{X}_2']] = Var[\boldsymbol{X}_2'] + E[0] = (t_2 - t_1)\sigma^2\boldsymbol{I}_J;$$

The marginal distribution for the subtree $\mathcal{T}'$ with two leaves is

$$\left[\boldsymbol{X}_1\ \boldsymbol{X}_2\right] \sim \mathsf{MN}_{J\times 2}\left(\left[\boldsymbol{X}_1'\ \boldsymbol{X}_1'\right], \boldsymbol{I}_J, \sigma^2\boldsymbol{\Sigma}^{\mathcal{T}'}\right), \quad \boldsymbol{\Sigma}^{\mathcal{T}'} = \begin{bmatrix} 1-t_1 & t_2-t_1 \\ t_2-t_1 & 1-t_1 \end{bmatrix}.$$

Therefore, we can merge two leaves responses $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$. Similarly, we can also merge the other subtree $\mathcal{T}''$ to obtain.

$$\left[\boldsymbol{X}_3\ \boldsymbol{X}_4\right] \sim \mathsf{MN}_{J\times 2}\left(\left[\boldsymbol{X}_1'\ \boldsymbol{X}_1'\right], \boldsymbol{I}_J, \sigma^2\boldsymbol{\Sigma}^{\mathcal{T}''}\right), \quad \boldsymbol{\Sigma}^{\mathcal{T}''} = \begin{bmatrix} 1-t_1 & t_3-t_1 \\ t_3-t_1 & 1-t_1 \end{bmatrix}.$$

Eventually, we can merge two subtrees (see Panel (B) of Figure S1), $\mathcal{T}'$ and $\mathcal{T}''$. From conjugacy of the normal distribution, the resulting joint marginal distribution of $\boldsymbol{X}_i, i = 1, 2, 3, 4$ is normal. The mean and the variance can be derived along identical lines as above. The only term left is the covariance, and we need to (re-)compute them for locations within and between the combined subtrees. Explicitly,

$$Cov[\boldsymbol{X}_1, \boldsymbol{X}_2] = Cov[E[\boldsymbol{X}_1|\boldsymbol{X}_1'], E[\boldsymbol{X}_2|\boldsymbol{X}_1']] + E[Cov[\boldsymbol{X}_1, \boldsymbol{X}_2|\boldsymbol{X}_1']] = Var[\boldsymbol{X}_1'] + E[(t_2 - t_1)\sigma^2\boldsymbol{I}_J] = t_2\sigma^2\boldsymbol{I}_J$$

$$Cov[\boldsymbol{X}_1, \boldsymbol{X}_3] = Cov[E[\boldsymbol{X}_1|\boldsymbol{X}_1'], E[\boldsymbol{X}_3|\boldsymbol{X}_1']] + E[Cov[\boldsymbol{X}_1, \boldsymbol{X}_3|\boldsymbol{X}_1']] = Var[\boldsymbol{X}_1'] + E[0] = t_1\sigma^2\boldsymbol{I}_J\,.$$

This ensures that

$$\mathbf{X}^{\mathsf{T}} = \left[\boldsymbol{X}_1\ \boldsymbol{X}_2\ \boldsymbol{X}_3\ \boldsymbol{X}_4\right] \sim \mathsf{MN}_{J\times 4}\left(\left[0\,0\,0\,0\right], \boldsymbol{I}_J, \sigma^2\boldsymbol{\Sigma}^{\mathcal{T}}\right), \boldsymbol{\Sigma}^{\mathcal{T}} = \begin{bmatrix} 1 & t_2 & t_1 & t_1 \\ t_2 & 1 & t_1 & t_1 \\ t_1 & t_1 & 1 & t_3 \\ t_1 & t_1 & t_3 & 1 \end{bmatrix},$$

as required. Moreover, denote $t_{i,i'}$ as the most recent divergence time of leaves i and i'. We observe that $t_1 = t_{1,3} = t_{1,4} = t_{2,3} = t_{2,4}, t_2 = t_{1,2}$, and $t_3 = t_{3,4}$ and complete the Proposition 1. □

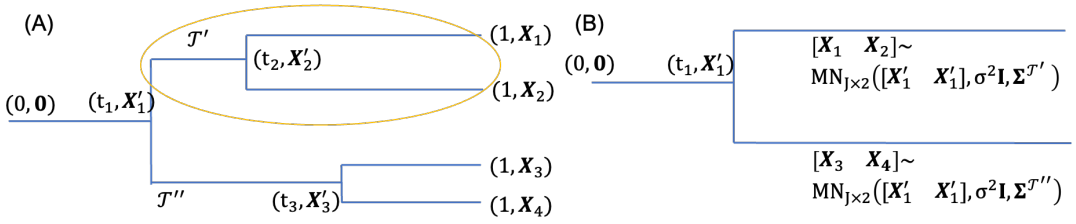


Figure S1: Merging subtrees for the integration process. (A) First step of merging upper subtree, and (B) Final step of merging all subtrees.

S2. Efficient Two-Stage Hybrid ABC-MH Algorithm Here we offer details of two-stage algorithm with pseudo code. In the Section S2.1, we describe the full algorithm of the ABC with the following posterior summary of Euclidean parameters (c, σ^2). The Section S2.2 includes the implementation of the proposal function and the acceptance probability of MH stage. Pseudo code for the full two-stage algorithm is presented below in Algorithm S1

S2.1. *ABC Stage and the Posterior Summary of c and σ^2* The Section 3 of the Main Paper states the main idea of ABC and we offer the full algorithm of ABC including (i) the synthetic data generation process, (ii) the regression adjustment (Blum, 2010) of ABC, and (iii) posterior summary of the Euclidean parameters.

Data generation in ABC. Following Section 2 in the Main Paper, a synthetic data is generated from DDT as follows: (i) given $c_l \sim \text{Gamma}(a_c, b_c)$, generate a tree $\mathcal{T}_l$ through the divergence function $a(t) = c_l(1-t)^{-1}$, and (ii) given $\mathcal{T}_l$ and $1/\sigma_l^2 \sim \text{Gamma}(a_{\sigma^2}, b_{\sigma^2})$, generate triples $(t_j, \boldsymbol{X}_i', \boldsymbol{X}_i), i' = 1 \ldots I-1, i = 1 \ldots I$ by a scaled Brownian motion upon $\mathcal{T}_l$. After discarding $(\mathcal{T}_l, t_i, \boldsymbol{X}_i')$, the leaf locations $\boldsymbol{X}_i$ form an I by J observed data matrix $\mathbf{X}_l$. In Algorithm S1, ABC repeats the procedure above to generate N^{syn} synthetic data (see Figure S2).

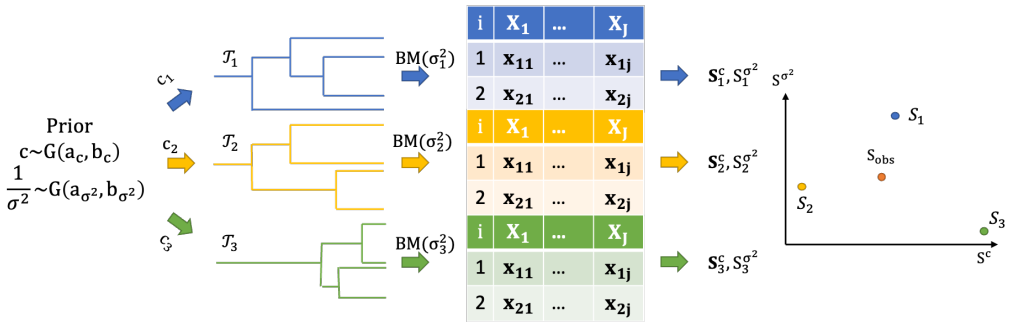


Figure S2: Schematic diagram of synthetic data generation and the calculation of summary statistics (first stage of Algorithm S1). S_{obs} is calculated based on the actual observed data.

Regression adjustment in ABC. Originally proposed in Beaumont, Zhang and Balding (2002) and later generalized by Blum (2010), regression adjustment for ABC is performed in Step 8 of Algorithm S1. The motivation is to use smoothing technique to weaken the effect of the discrepancy between the summary statistic calculated from synthetic data and that from the observed data. We briefly describe the the procedure of c. Additional details can be found in Beaumont, Zhang and Balding (2002) and Blum (2010). Suppose we are given the observed summary statistics $\boldsymbol{S}_{\text{obs}}^{(c)}$ and unadjusted samples $(c_l^{\text{unadj}}, \boldsymbol{S}_l^{(c)}), l = 1, \ldots, k$, we can calculate the weight for each sample by

$$w_l^{(c)} = K_h(\|\boldsymbol{S}_l^{(c)} - \boldsymbol{S}_{\text{obs}}^{(c)}\|) \tag{S1}$$

, where the bandwidth h is set at the largest value, such that $K_h(\max_{l=1\ldots k} \|\boldsymbol{S}_l^{(c)} - \boldsymbol{S}_{\text{obs}}^{(c)}\|) = 0$ to ensure non-zero importance weight for k samples (Sisson, Fan and Beaumont, 2019) and mean integrated square error consistency (Biau, Cérou and Guyader, 2015). Regression adjustment seeks to produce adjusted samples c_l but maintain the sample weights and thus assumes the following model for the unadjusted samples c^{unadj} with mean-zero i.i.d errors ϵ_l where $E(\epsilon_l^2) < \infty$ for $l = 1 \ldots, k$:

$$c_l^{\text{unadj}} = m(\boldsymbol{S}_l^{(c)}) + \epsilon_l\,. \tag{S2}$$

The estimated regression function $\widehat{m}$ is then a kernel-based local-linear polynomial obtained as a solution of $\text{argmin}_{\alpha,\beta} \sum_{l=1}^{k} [c_l^{\text{unadj}} - (\alpha + \beta(\boldsymbol{S}_l^{(c)} - \boldsymbol{S}_{\text{obs}}^{(c)}))]^2 w_l^{(c)}$. Using the empirical residuals $\widehat{\epsilon}_l = c_l^{\text{unadj}} - \widehat{m}(\boldsymbol{S}_l^{(c)})$, we then construct the adjusted values $c_l = \widehat{m}(\boldsymbol{S}_{\text{obs}}^{(c)}) + \widehat{\epsilon}_l$.

Posterior summary of Euclidean parameters (c, σ^2)**.** The first stage of our ABC-MH algorithm produces weighted samples $\{c_\ell, w_l^{(c)}\}$, $\{\sigma_\ell^2, w_l^{(\sigma^2)}\}$, $l = 1, \ldots, k$, and we summarize the weighted samples as follows. We illustrate the calculations with c, and the calculations for σ^2 follow similarly. We calculate the posterior median and 95% credible interval by finding the 50, 2.5 and 97.5% quantiles, and use the posterior median for the second stage of the proposed ABC-MH algorithm when sampling the tree. In general, for calculating the $q \times 100\%$ quantile, we fit an intercept-only quantile regression of c_ℓ with weights $w_l^{(c)}$; this is implemented by `rq` wrapped in the summary function `summary.abc` in the R package `abc`.

S2.2. *MH Algorithm for Updating the Tree in the DDT Model.* In the second stage of Algorithm S1, we have used existing MH tree updates (Knowles and Ghahramani, 2015). We briefly describe the proposal for generating a candidate tree $\mathcal{T}'$ from the current tree $\mathcal{T}$ and the acceptance probability. Given the current tree, a candidate tree is proposed in two steps: (i) detaching a subtree from the original tree, and (ii) reattaching the subtree back to the remaining tree (see Figure S3). In Step i, let $(\mathcal{S}, \mathcal{R})$ be the output of the random detach function that divides the original tree $\mathcal{T}$ into two parts at the detaching point u, where $\mathcal{S}$ is the detached subtree and $\mathcal{R}$ is the remaining tree. In this paper, we generate the detaching point u by uniformly selecting a node and taking the parent of the node as the detaching point. In Step ii, for the re-attaching point v, we follow the divergence and branching behaviors of the generative DDT model by treating subtree $\mathcal{S}$ as a single datum and adding a new datum $\mathcal{S}$ to $\mathcal{R}$. Given the point v, a candidate tree $\mathcal{T}'$ results by re-attaching $\mathcal{S}$ back to $\mathcal{R}$ at point v. The time of re-attaching point t_v is then earlier than the time of the root of $\mathcal{S}$ to avoid distortion of $\mathcal{S}$: $t_v < t(\text{root}(\mathcal{S}))$. By choosing u and v as above, we have described the proposal distribution from $\mathcal{T}$ to $\mathcal{T}'$, $q(v, \mathcal{R})$, which is essentially the probability of diverging at v on the subtree $\mathcal{R}$. The acceptance probability is then

$$\min\left\{1, \frac{f(\mathcal{T}', \mathbf{X}) q(u, \mathcal{R})}{f(\mathcal{T}, \mathbf{X}) q(v, \mathcal{R})}\right\} \tag{S3}$$

, where $f(\mathcal{T}, \mathbf{X}) = f(\mathcal{T}, \mathbf{X} | c_0, \sigma_0^2) = P(\mathbf{X} | \mathcal{T}, \sigma_0^2) P(\mathcal{T} | c_0)$, $P(\mathbf{X} | \mathcal{T}, \sigma_0^2)$ is the likelihood of the tree structure (Proposition 1), $P(\mathcal{T} | c_0)$ is the prior for the tree (the first two terms in Equation (4)), and c_0 and σ_0^2 are representative value chosen from the posterior sample of c and σ^2, respectively.

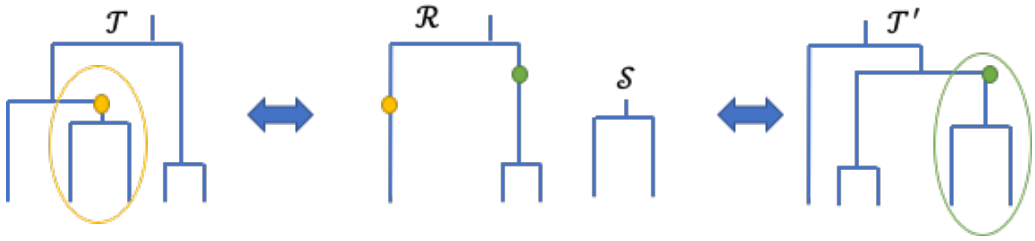


Figure S3: Schematic diagram of proposing a candidate tree in MH. (Left) Current tree $\mathcal{T}$ with detach point u (yellow); (Middle) Intermediate subtrees with remaining tree $\mathcal{R}$ and the detached subtree $\mathcal{S}$; (Right) The proposed tree $\mathcal{T}'$ with reattached point v (green).

Algorithm S1 Two-stage hybrid ABC-MH algorithm

Input:

(a) Observed data: $\mathbf{X} = [\boldsymbol{X}_1, \ldots, \boldsymbol{X}_I]^{\mathsf{T}}$ consisting of I points in $\mathbb{R}^J$;
(b) Summary statistics $\boldsymbol{S}^{(c)}, S^{(\sigma^2)}$ defined in the Main Paper Section 3.1.1;
(c) Synthetic data of size N^{syn} and threshold $d \in (0,1)$ with $k = \lceil N^{\text{syn}} d \rceil$, the number of nearest synthetic data sets to retain;
(d) Prior for model parameters: $c \sim \mathsf{Gamma}(a_c, b_c)$, $\frac{1}{\sigma^2} \sim \mathsf{Gamma}(a_{\sigma^2}, b_{\sigma^2})$;
(e) Univariate Kernel $K_h(\cdot)$ with bandwidth $h > 0$ and compact support.

Output:

(a) Posterior samples of c and σ^2 of size $k = N^{\text{syn}} d$;
(b) posterior samples of $(\mathcal{T}, \boldsymbol{t})$.

procedure EUCLIDEAN PARAMETERS(c, σ^2)
  for $l = 1 \ldots N^{\text{syn}}$ **do**
    Sample Euclidean parameters from prior $c_l \sim \mathsf{Gamma}(a_c, b_c), \sigma_l^2 \sim \mathsf{Gamma}(a_{\sigma^2}, b_{\sigma^2})$;
    Simulate data $\mathbf{X}_l$ from DDT using (c_l, σ_l^2);
    Compute: $\boldsymbol{S}_l^{(c)}$ and $S_l^{(\sigma^2)}$ along with $\|\boldsymbol{S}_l^{(c)} - \boldsymbol{S}_{\text{obs}}^{(c)}\|$ and $\|S_l^{(\sigma^2)} - S_{\text{obs}}^{(\sigma^2)}\|$.
  Choose $\{(c_{l_s}, \sigma_{l_s}^2), s = 1, \ldots, k\}$ corresponding to k smallest $\|\boldsymbol{S}_l^{(c)} - \boldsymbol{S}_{\text{obs}}^{(c)}\|$ and $\|S_l^{(\sigma^2)} - S_{\text{obs}}^{(\sigma^2)}\|$
  Calculate the sample weights $w_{l_s}^{(c)} = K_h(\|\boldsymbol{S}_{l_s}^{(c)} - \boldsymbol{S}_{\text{obs}}^{(c)}\|)$ and $w_{l_s}^{(\sigma^2)} = K_{h'}(\|S_{l_s}^{(\sigma^2)} - S_{\text{obs}}^{(\sigma^2)}\|)$ based on Equation (S1);
  Compute regression adjusted samples c_{l_s} and $\sigma_{l_s}^2$ with weights $w_{l_s}^{(c)}$ and $w_{l_s}^{(\sigma^2)}$ with the model (S2) and calculate posterior summary c_0 and σ_0^2 plugging the adjusted c_{l_s} and $\sigma_{l_s}^2$.
procedure TREE PARAMETERS($(\mathcal{T}, \boldsymbol{t})$)
  Follow the MH algorithm in Section S2.2 with fixed c_0 and σ_0^2 at the posterior median values and compute acceptance probabilities with Equation S3.

S3. Tree Projection of Pairwise iPCP Matrix In the Main Paper Section 3.2, we mentioned that a pairwise iPCP matrix $\boldsymbol{\Sigma}$ with entries $\text{iPCP}_{i,i'}, i, i' = 1, \ldots, I$ need not to be a tree-structured matrix and we address the projection of Σ on to the space of tree-structured matrices here. Given $L > 1$ posterior trees with I leaves and the corresponding pairwise iPCP matrix $\boldsymbol{\Sigma} = \left(\text{iPCP}_{i,i'}\right)$, each entry of iPCP matrix can be express as $\text{iPCP}_{i,i'} = \frac{\sum_{l=1}^{L} t_{i,i'}^{(l)}}{L}$, where $t_{i,i'}^{(l)}$ is the divergence time of leaves i and i' in the l-th posterior tree. Obviously, every entry of the iPCP matrix takes the element-wise Monte Carlo average over L tree-structured matrix and breaks the inequalities (2) and (3) in the Main Paper. Following the work of Bravo et al. (2009), by representing a tree as a tree-structured matrix, we can project Σ on to the closest tree-structured matrix in terms of Frobenius norm. The projection can be formulated as a constrained mixed-integer programming (MIP) problem:

$$\underset{\boldsymbol{\Sigma}^{\mathcal{T}}}{\text{argmin}} \quad ||\boldsymbol{\Sigma} - \boldsymbol{\Sigma}^{\mathcal{T}}||_F$$

$$\text{s.t.} \quad \Sigma_{i,i'}^{\mathcal{T}} \geq 0;\ \Sigma_{i,i}^{\mathcal{T}} \geq \Sigma_{i,i'}^{\mathcal{T}};\ \Sigma_{i,i'}^{\mathcal{T}} \geq \min(\Sigma_{i,i''}^{\mathcal{T}}, \Sigma_{i',i''}^{\mathcal{T}}),\ \text{for all } i \neq i' \neq i''.$$

We applied the projection on the pairwise iPCP matrix from the breast cancer (panel (A)), colorectal cancer (panel (B)) and melanoma (panel (C)) data of NIBR-PDXE and show the result in the Figure S4. In Figure S4, the MAP tree, the tree representation of projected iPCP matrix (MIP tree), the original iPCP matrix and the projected iPCP matrix are shown in from the left to the right columns, respectively. From the left two columns of the tree structures, we found that trees from the MAP and MIP show similar pattern and the MIP tree allows a non-binary tree structure. For example, three combination therapies and two PI3K inhibitors (CLR457 and BKM120) framed by a box form a tight subtree in both MAP

and MIP tree, but the subtree in the MIP is non-binary. For the iPCP matrix, high element-wise correlation $\text{Cor}(\Sigma^{\mathcal{T}}_{i,i'}, \Sigma_{i,i'})$ between the original iPCP Σ and the projected iPCP $\Sigma^{\mathcal{T}}$ are presented (BRCA: 0.9987; CRC: 0.9962; CM: 0.9918).

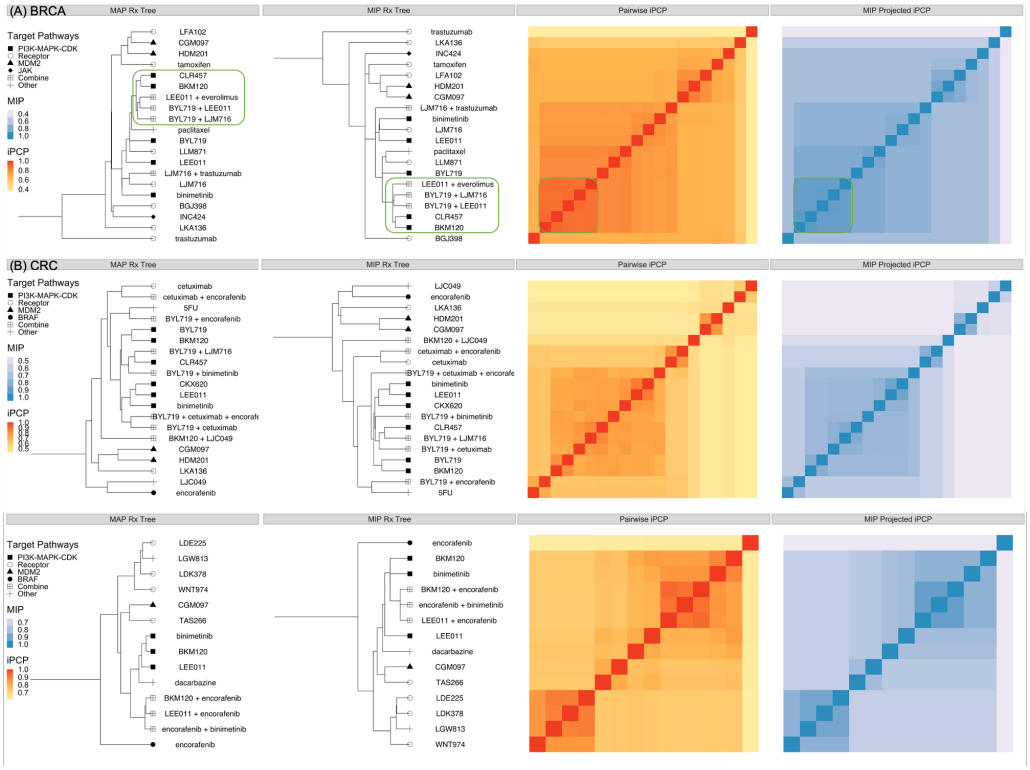


Figure S4: Comparison between (Left two columns) the tree structure from the MAP and the projected iPCP matrix (MIP tree) and (Right two columns) the matrix from the original iPCP matrix and the projected iPCP matrix for (A) breast cancer, (B) colorectal cancer and (C) melanoma. The matrix from the original iPCP and the MIP projected iPCP matrix are aligned by the MIP tree.

S4. Simulation Studies of Euclidean Parameters In this section, we empirically compare the Euclidean parameters of c and σ^2 from ABC of the proposed two-stage algorithm and single-stage MCMC. We organize this section as follows. We first compare other candidate summary statistics of c and σ^2 for ABC in Section S4.1. In Section S4.2, we illustrate the superior inference performance of Euclidean parameters from ABC than single-stage MCMC through a series of simulations. Section S4.3 offers the diagnostic statistics and the sensitivity analysis for ABC stage of the proposed two-stage algorithm and checks the convergence of c and σ^2 for the single-stage MCMC.

Simulation setup. For illustrative purposes, we fixed the observed PDX data matrix with 50 treatments ($I = 50$) and 10 PDX mice ($J = 10$) in all simulation scenarios. In addition, we let c and σ^2 take values from $\{0.3, 0.5, 0.7, 1\}$ and $\{0.5, 1\}$ respectively to mimic the PDX data with tight and well-separated clusters. For each pair of (c, σ^2), 200 replicated experiments with different tree and observed PDX data matrices were independently drawn according to the DDT generating model. We specify a prior distribution for $c \sim \mathsf{Gamma}(2, 2)$ with shape and rate parameterization. For diffusion variance σ^2, let $1/\sigma^2 \sim \mathsf{Gamma}(1, 1)$. We compare ABC-MH of the proposed two-stage algorithm against two alternatives based on single-stage

MH algorithms (Neal, 2003) (see details in Section S2.2). The first one initializes at the true parameter values and the true tree, referred to as $\text{MH}_{\mathsf{true}}$. The idealistic initialization at the truth is a best case scenario in applying existing MH algorithm to inferring DDT models. The second alternative, referred to as $\text{MH}_{\mathsf{default}}$, initializes (c, σ^2) by a random draw from the prior; the unknown tree is initialized by agglomerative hierarchical clustering with Euclidean distance and squared Ward's linkage (Murtagh and Legendre, 2014) – thus providing a fair apples-to-apples comparison. For the ABC, we generated N^{syn} synthetic data of c and σ^2 and kept $k = \lceil N^{\text{syn}} d \rceil$ nearest samples in terms of the $\|\boldsymbol{S}_l^{(c)} - \boldsymbol{S}_{\text{obs}}^{(c)}\|$ and $\|S_l^{(\sigma^2)} - S_{\text{obs}}^{(\sigma^2)}\|$. We varied the number of synthetic data N^{syn} and the threshold parameter $d \in (0, 1)$ under different settings and we specified N^{syn} and d in each of the following sections. We ran two MH algorithms with 10,000 iterations and discarded the first 7,000 iterations.

Performance metrics for Euclidean parameters. We used two algorithm performance metrics to compare our algorithm to the classical single-stage MCMC algorithms. First we computed the effective sample sizes for each Euclidean parameter c and σ^2 (ESS_c and ESS_{σ^2}) given a nominal sample size (NSS) kept for posterior inference. ESS for each parameter represents the number of independent draws equivalent to NSS posterior draws of correlated ($\text{MH}_{\mathsf{true}}$ and $\text{MH}_{\mathsf{default}}$) or independent and unequally weighted samples (ABC stage of the proposed algorithm). We let NSS for MH algorithms be the number of consecutive posterior samples in a single chain after a burn-in period; let NSS for ABC be k as in Step 6, Algorithm S1. For c and σ^2, the ESS of MH (Gelman et al., 2013) is estimated by $\text{NSS}/(1 + \sum_{t=1}^{\infty} \widehat{\rho}_t)$ where $\widehat{\rho}_t$ is the estimated autocorrelation function with lag t (Geyer, 2011). The ESS for ABC (Sisson, Fan and Beaumont, 2019) is the reciprocal of the sum of squared normalized weights, $1/\sum_{l=1}^{k} \widetilde{W}_l^2$, where $\widetilde{W}_l = w_l / \sum_{l'=1}^{k} w_{l'}$ (see weights, w_l, in Equation (S1)). Second, we evaluated how well did the posterior distributions recover the true (c, σ^2). We computed the mean absolute percent bias for c and σ^2: $|\mathbb{E}\{c \mid \mathbf{X}\} - c|/c$ and $|\mathbb{E}\{\sigma^2 \mid \mathbf{X}\} - \sigma^2|/\sigma^2$, respectively. We also computed the empirical coverage rates of the nominal 95% credible intervals (CrI) for c and σ^2.

S4.1. *Other Choices of Summary Statistics* Proposition 1 points towards other potential summary statistics for the first stage of Algorithm S1 that uses ABC to produce weighted samples to approximate the posterior distributions for c and σ^2. Here we consider a few such alternatives with $N^{\text{syn}} = 600,000$ and $d = 0.5\%$ and empirically compare their performances to the summary statistics used in the Main Paper ($\boldsymbol{S}^{(c)}$ and $S^{(\sigma^2)}$) in terms of the mean absolute percent bias in recovering the true parameter values of c and σ^2.

Summary statistic for c. Unlike building $\boldsymbol{S}^{(c)}$ based on the inter-point distance, the off-diagonal terms of $\boldsymbol{T} = \sum_j \boldsymbol{X}_{\cdot,j} \boldsymbol{X}_{\cdot,j}^{\mathsf{T}}$ (see the definition of $\boldsymbol{T}$ in Lemma 1 in Main Paper) is another potential summary statistic for c. Since the divergence parameter c affects the marginal likelihood implicitly through the divergence time $\boldsymbol{t}$, the summary statistics for $\boldsymbol{t}$ is informative for c. From Proposition 1, $\boldsymbol{T}$ is sufficient for $\sigma^2 \Sigma_{\mathcal{T}}$, where the off-diagonal terms of $\sigma^2 \Sigma_{\mathcal{T}}$ taking the form $\sigma^2 t_d, d = 1 \ldots n-1$ and containing unrelated information from σ^2. Let $\boldsymbol{Q_T}$ be a vector of the 10th, 25th, 50th, 75th and 90th percentiles of the off-diagonal terms of $\boldsymbol{T}$. Because $\boldsymbol{T}$ is sufficient for $\sigma^2 \Sigma_{\mathcal{T}}$ and involves extra Gaussian diffusion variance parameter, we can design alternative summary statistics based on $\boldsymbol{Q_T}$ through (i) augmentation, $(\boldsymbol{Q_T}, S^{(\sigma^2)})$ or (ii) scaling, $\boldsymbol{Q_T}/S^{(\sigma^2)}$. From Figure S5, $\boldsymbol{S}^{(c)}$ proposed in the Main Paper outperformed the summary statistics from $\boldsymbol{Q_T}$ by producing less biased posterior mean estimates.

Summary statistic for σ^2. Following Proposition 1, several matrix functionals on the data $\mathbf{X}$ or statistics $\boldsymbol{T}$ can be considered as alternatives to $S^{(\sigma^2)}$. We compare performance of three candidates: (i) average L_1 norm (AvgL1) of columns: $\frac{1}{J} \sum_{j=1}^{J} |\boldsymbol{X}_{\cdot,j}|_1$; (ii) Frobenius norm

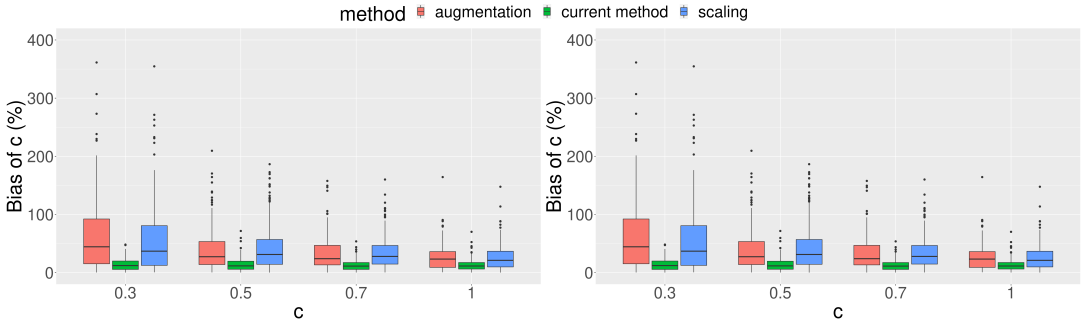


Figure S5: Comparison among different summary statistics for c (red: $(\boldsymbol{Q_T}, S^{(\sigma^2)})$; green: $\boldsymbol{S}^{(c)}$; blue: $\boldsymbol{Q_T}/S^{(\sigma^2)}$) under different values of σ^2 in terms of the mean absolute percent bias. (Left) $\sigma^2 = 0.5$; (Right) $\sigma^2 = 1$.

of $\mathbf{X}$; and, (iii) vector containing 10th, 25th, 50th, 75th and 90th percentiles of first principal component (PC1) of $\mathbf{X}$. From Figure S6, the first three methods are comparable while ABC based on principal components shows larger bias due to the information loss.

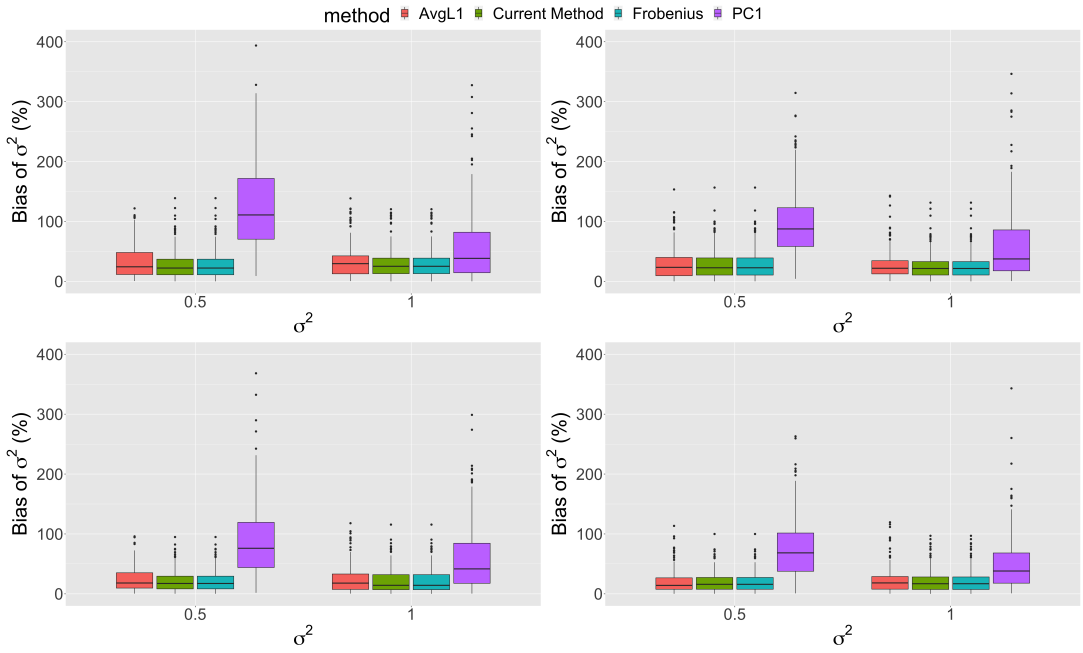


Figure S6: Comparison among different summary statistics for σ^2 under different values of c in terms of the mean absolute percent bias. (Upper Left) $c = 0.3$; (Upper Right) $c = 0.5$; (Lower Left) $c = 0.7$; (Upper Right) $c = 1.0$.

S4.2. *Posterior Inference of Euclidean Parameters* In this section, we show that two-stage algorithm (ABC-MH) outperforms the single-stage MCMC (MH) for real parameters in terms of (i) stable effective sample size (ESS) for (c, σ^2); (ii) similar or better inference on (c, σ^2), as ascertained using mean absolute percent bias and nominal 95% credible intervals.

S4.2.1. *Stable Effective Sample Sizes of ABC-MH* We calculated ESS-to-NSS ratios at varying truths of c and σ^2. To illustrate, we matched the NSS budget of ABC with that of MH (NSS = $3,000$) by keeping $d = 0.5\%$ of $N^{\text{syn}} = 600,000$ synthetic data sets that are closest to the observed data in terms of the summary statistic for each parameter (Step 6 of Algorithm S1). Table S1 shows that the ESS_c/NSS and $\text{ESS}_{\sigma^2}/\text{NSS}$ ratio from ABC is stable between 0.64 to 0.68 and around 0.83 across different c and σ^2 values, respectively. In contrast, the ESS_c/NSS ratio for MH quickly deteriorates ($\text{MH}_{\mathsf{true}}$: 0.97 to 0.41; $\text{MH}_{\mathsf{default}}$: 0.73 to 0.35) as c increases from 0.3 to 1 and $\text{ESS}_{\sigma^2}/\text{NSS}$ for MH are extremely poor (< 0.06) across different values of c and σ^2. MH produced very good ESS_c under small value $c = 0.3$ but poor ESS_c under $c = 1$. As a result, under larger values of c, MH algorithms must run longer to reach a target ESS_c. Although ESS_c for ABC is not as high as $\text{MH}_{\mathsf{true}}$ or $\text{MH}_{\mathsf{default}}$ at $c = 0.3$, the stability of ESS_c of ABC means that a predictably constant NSS is needed for conducting posterior inference across different values of c. Finally, the ESS_{σ^2} for the diffusion variance parameter from MH algorithms are strikingly smaller than ABC, indicating ABC should be preferred.

TABLE S1

ESS-to-NSS ratios between ABC-MH ($d = 0.5\%$), $\text{MH}_{\mathsf{true}}$, and $\text{MH}_{\mathsf{default}}$. All values here are obtained from 200 independent replications. For each random replication at (c, σ^2). All methods were controlled to produce identical NSS with size $3,000$.

		ESS/NSS(sd) for c		ESS/NSS(sd) for σ^2	
c	method	$\sigma^2 = 0.5$	$\sigma^2 = 1$	$\sigma^2 = 0.5$	$\sigma^2 = 1$
0.3	ABC-MH	0.68(0.032)	0.67(0.027)	0.83(0.0048)	0.83(0.0042)
	$\text{MH}_{\mathsf{true}}$	0.97(0.11)	0.96(0.13)	0.051(0.061)	0.056(0.072)
	$\text{MH}_{\mathsf{default}}$	0.73(0.33)	0.67(0.34)	0.028(0.043)	0.038(0.08)
0.5	ABC-MH	0.66(0.02)	0.65(0.018)	0.83(0.0047)	0.83(0.0044)
	$\text{MH}_{\mathsf{true}}$	0.85(0.23)	0.83(0.24)	0.034(0.042)	0.045(0.067)
	$\text{MH}_{\mathsf{default}}$	0.66(0.35)	0.62(0.34)	0.033(0.051)	0.041(0.067)
0.7	ABC-MH	0.65(0.017)	0.64(0.017)	0.83(0.0047)	0.83(0.004)
	$\text{MH}_{\mathsf{true}}$	0.63(0.31)	0.67(0.32)	0.024(0.027)	0.029(0.038)
	$\text{MH}_{\mathsf{default}}$	0.53(0.33)	0.51(0.35)	0.028(0.039)	0.038(0.072)
1.0	ABC-MH	0.65(0.017)	0.64(0.017)	0.83(0.0044)	0.83(0.0041)
	$\text{MH}_{\mathsf{true}}$	0.41(0.3)	0.44(0.32)	0.019(0.026)	0.019(0.023)
	$\text{MH}_{\mathsf{default}}$	0.35(0.29)	0.35(0.29)	0.022(0.026)	0.022(0.027)

S4.2.2. *Superior Quality Posterior Inference of ABC-MH* Does ABC give better posterior inference with a fixed computational budget? To make fair comparisons, we fixed a total CPU time and used the same computing processor to run the ABC (1st stage of Algorithm 1) and MH algorithms. Let t_{MH} and t_{ABC} be the estimated CPU time for generating one iteration in MH and one synthetic data in ABC on the same processor. Note, t_{MH} includes the additional time for proposing a valid tree. By varying the number of synthetic samples, we can match the total CPU time used by ABC with that of MH algorithms which were run for $10,000$ iterations. We generated $10,000 t_{\mathsf{MH}}/t_{\mathsf{ABC}} = 17,345$ synthetic data sets and took $d = 5\%$ with summary statistics $\boldsymbol{S}^{(c)}$ and $S^{(\sigma^2)}$ (see different values of d in Section S4.3.3) for ABC. Table S2 shows that ABC produced posterior samples that confer comparable inferences about c in terms of the bias and coverage of nominal 95% CrIs. The posterior mean of c from ABC is comparable to that from $\text{MH}_{\mathsf{true}}$ and less biased than $\text{MH}_{\mathsf{default}}$ for all settings. The coverage rates of the nominal 95% CrIs from ABC are comparable to $\text{MH}_{\mathsf{true}}$ but higher

than $\text{MH}_{\text{default}}$. MH_{true}, however, is initialized at true values and is unrealistic in practice. We observed MH_{true} sometimes failed to converge (Table S3), stuck around the initial true values and resulted in deceptively low biases and good coverage rates. Turning to the inference of σ^2, ABC offers a much better alternative to MH algorithms in terms of smaller bias in the posterior mean and better coverage of the 95% credible intervals (Table S2). This is primarily caused by the difficulty of MH in exploring the posterior distribution of σ^2 resulting in chains with high auto-correlations. The squeezed boxplots in Figure S7 indicate that the chains for σ^2 in MH_{true} and $\text{MH}_{\text{default}}$ were almost always slowly mixing and stuck around the initial values. In addition, unlike the serial nature of MH, ABC can be further parallelized to reduce the wall clock time to a fraction of what is required by MH using multicore processors. Although parallelizing MH with techniques such as consensus MCMC (e.g., Scott et al., 2016) is possible, the parallelized ABC does not require data splitting and will not trade the quality of posterior inference for computational speed.

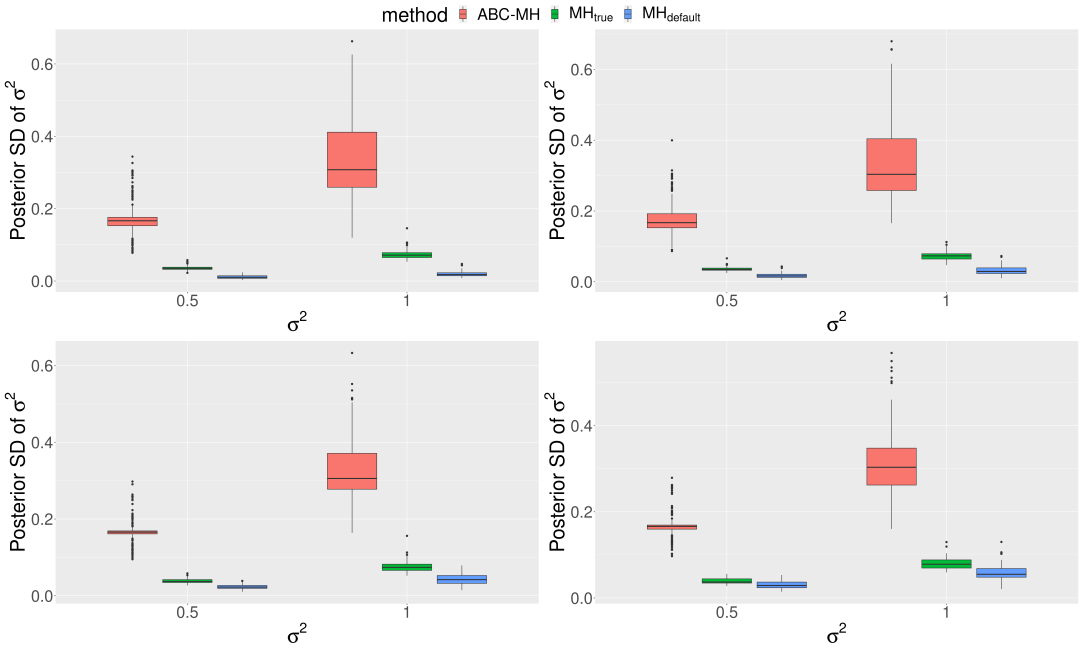


Figure S7: (Upper left) $c = 0.3$; (Upper right) $c = 0.5$; (Lower left) $c = 0.7$; (Lower right) $c = 1.0$. The posterior standard deviation of σ^2 from MH (green and blue) are close to zero across different true c showing MH is stuck. Results are based on 200 replications.

S4.3. *Algorithm Diagnostics* Here we examine the convergence of MH through the Geweke statistics (Geweke, 1992) and the goodness of fit for ABC. Specifically, two important hyper-parameters are involved in ABC: (i) the kernel bandwidth h for samples weights in Equation S1 and (ii) the threshold d for $k = \lceil N^{\text{syn}} d \rceil$ nearest samples in the Step 6 of Algorithm S1. We follow the test from Prangle et al. (2014) to justify the kernel bandwidth h and conduct the sensitivity analysis for threshold d to understand how threshold d affects the result in terms of the inferential performance.

S4.3.1. *Convergence of MH Chains in Simulations* In all of our simulations, we ran MH for $10,000$ iterations. Table S3 shows that the percentages of the converged MH chains for 200 replications are between 12.5 and 68.5% within a total $10,000$ iterations (based on

Table S2: Comparison of inferential performance for c and σ^2 between ABC-MH ($d = 5\%$), $\text{MH}_{\mathsf{true}}$, and $\text{MH}_{\mathsf{default}}$. All values here are obtained from 200 independent replications. For each random replication at (c, σ^2), all methods were run for identical total CPU time and only converged chains from MH algorithms were included.

		Percent Bias(sd) for c		Coverage(sd) for c		Percent Bias(sd) for σ^2		Coverage(sd) for σ^2	
c	method	$\sigma^2 = 0.5$	$\sigma^2 = 1$	$\sigma^2 = 0.5$	$\sigma^2 = 1$	$\sigma^2 = 0.5$	$\sigma^2 = 1$	$\sigma^2 = 0.5$	$\sigma^2 = 1$
0.3	ABC-MH	12(9.4)	13(9.9)	98(0.99)	99(0.71)	28(25)	31(25)	90(2.1)	88(2.3)
	$\text{MH}_{\mathsf{true}}$	13(9.8)	12(9.5)	94(2)	95(1.9)	9.4(7.1)	9(6.6)	80(3.4)	82(3.3)
	$\text{MH}_{\mathsf{default}}$	45(20)	46(20)	33(5.5)	30(6.1)	71(12)	72(11)	0(0)	0(0)
0.5	ABC-MH	15(11)	15(11)	92(1.9)	93(1.8)	28(26)	27(22)	90(2.2)	94(1.7)
	$\text{MH}_{\mathsf{true}}$	11(9)	11(8.6)	97(1.7)	97(1.6)	8.6(6.7)	9.9(7)	80(4)	78(4.1)
	$\text{MH}_{\mathsf{default}}$	33(18)	31(19)	60(5.5)	57(5.7)	54(17)	57(16)	1.2(1.2)	1.3(1.3)
0.7	ABC-MH	13(10)	14(11)	96(1.5)	93(1.8)	21(18)	22(20)	97(1.2)	94(1.6)
	$\text{MH}_{\mathsf{true}}$	12(9.1)	12(9.1)	95(2.6)	96(2.1)	11(8.3)	11(8.3)	70(5.3)	68(4.9)
	$\text{MH}_{\mathsf{default}}$	25(15)	27(16)	73(5.5)	69(5.9)	38(17)	41(19)	12(4)	8.1(3.5)
1.0	ABC-MH	14(11)	14(13)	95(1.5)	94(1.6)	19(18)	21(18)	98(1.1)	96(1.5)
	$\text{MH}_{\mathsf{true}}$	11(7.6)	13(11)	97(2)	92(3.5)	13(9.1)	10(7.7)	64(5.8)	85(4.6)
	$\text{MH}_{\mathsf{default}}$	14(11)	16(14)	93(3.5)	89(3.8)	24(15)	27(16)	35(6.5)	21(5.1)

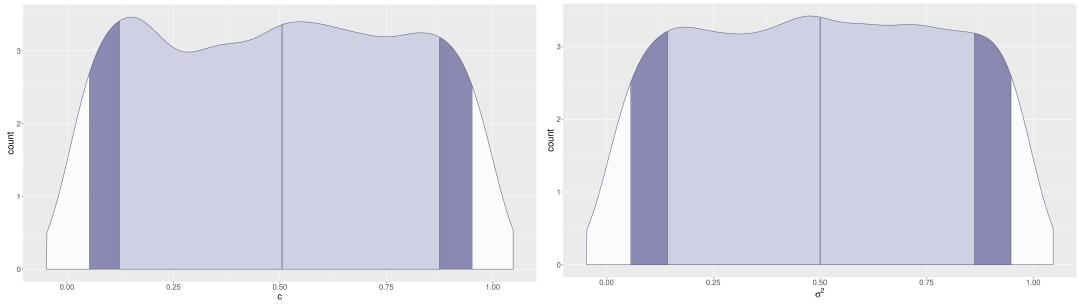
Geweke statistic). Running the chains longer will increase these percentages. In contrast, with appropriate choice of bandwidth and the fraction of synthetic samples to keep, ABC does not involve convergence issues and according to Section S4.2 achieves better ESS for a fixed NSS and similar or better quality posterior inference for fixed CPU time.

TABLE S3
Percentage of converged chains for (i) MH initialized at true (c, σ^2) (MH_{true}), and (ii) MH initialized randomly from prior (MH_{default}). All values here are obtained from 200 independent replications.

		Convergence % for c		Convergence % for σ^2	
c	method	$\sigma^2 = 0.5$	$\sigma^2 = 1$	$\sigma^2 = 0.5$	$\sigma^2 = 1$
0.3	$\text{MH}_{\mathsf{true}}$	68.0	68.5	16.5	22.5
	$\text{MH}_{\mathsf{default}}$	36.5	28.5	12.5	16.0
0.5	$\text{MH}_{\mathsf{true}}$	50.0	52.0	23.0	29.5
	$\text{MH}_{\mathsf{default}}$	40.5	37.5	18.5	23.5
0.7	$\text{MH}_{\mathsf{true}}$	38.0	46.0	26.5	27.0
	$\text{MH}_{\mathsf{default}}$	33.5	31.0	14.5	25.0
1.0	$\text{MH}_{\mathsf{true}}$	35.0	30.5	20.5	30.5
	$\text{MH}_{\mathsf{default}}$	27.5	33.0	14.0	25.0

S4.3.2. *Diagnostics for ABC* We empirically justify the choice of the kernel bandwidth h and the goodness of approximation in ABC algorithm by the calibration method from Prangle et al. (2014) based on the coverage property of the credible interval. Suppose we generated pseudo-observed data $\mathbf{X}_e$ in the eth replication from the DDT model with parameter (c_e, σ_e^2), where c_e and σ_e^2 are random draws from the prior ($c_e \sim \mathsf{Gamma}(a_c, b_c), 1/\sigma_e^2 \sim \mathsf{Gamma}(a_{\sigma^2}, b_{\sigma^2})$) and $e = 1 \dots E$. Once the tuning parameters (N^{syn}, d, h) are decided, Algorithm S1 will output regression adjusted sample (c_ℓ, σ_ℓ^2) with size $\ell = 1, \dots, k; k = \lceil N^{\text{syn}} d \rceil$ based on the input data D. We describe diagnostics for c, and note that an identical description applies to σ^2 as well. According to Cook, Gelman and Rubin (2006), the ABC procedure produces reliable approximations of the posterior if the random variables $q_e^{(c)} := \frac{1}{k} \sum_{l=1}^{k} \mathbb{I}_{\{c_\ell > c_e\}}$ follow a uniform distribution over the interval $(0, 1)$. Accordingly, Prangle et al. (2014) suggest a goodness-of-fit test $H_0 : q_e^{(c)} \sim \mathsf{Unif}(0, 1)$ as a diagnostic in order to calibrate ABC. If the test fails to reject the null hypothesis, the empirical quantiles can be viewed as being indistinguishable from the uniform distribution, and the credible interval from the posterior samples would show the asserted coverage. We use the Kolmogorov–Smirnov statistic to carry out the test, follow the simulation setting with $I = 50$ and $J = 10$, and reuse $600,000$ synthetic data sets. The synthetic data is randomly split into two non-overlapping subsets: training data with size $597,000$ and pseudo-observed data with size $E = 3,000$. Again, we run the ABC part of Algorithm S1 by treating each of the pseudo-observed data sets as the actually observed data with $N^{\text{syn}} = 597,000$ and $d = 0.5\%$. We obtained statistically non-significant KS statistics for c and σ^2 (p-values: 0.61 for c, 0.71 for σ^2). The 95% credible intervals from ABC showed 94.9% and 95.93% empirical coverage rates which are close to the nominal level.

S4.3.3. *Sensitivity Analysis of k Nearest Samples* In the previous section, we have used a simple diagnostic procedure to show the choice of bandwidth parameter h is reasonable. Here we focus on conducting additional simulations to investigate how does varying values of d in the Step 6 of Algorithm S1 impact the inferential performance of ABC. We focus on



(a) Empirical quantiles at true c **(b)** Empirical quantiles at true σ^2

Figure S8: The empirical quantiles at the true value follow the standard uniform distribution indicating calibrated ABC. Results are based on $3,000$ independent draws from the prior.

c to illustrate the main points. Similar to Table S2 in the Section S4.2 where $d = 5\%$, in the following we show the results for $d = 0.5\%$ and $d = 1\%$ in Table S4. First, for ABC itself, the bias in the posterior mean is similar, e.g. the mean bias is 14% for all three different d when $c = 1.0$ and $\sigma^2 = 0.5$. For each pair of (c, σ^2), the empirical coverage rate of the 95% credible interval decreases when d increases from 0.5% to 5%. Specifically, the empirical coverage range from 92% to 99% for $d = 5\%$, 88% to 97% for $d = 1\%$ and 84% to 94% for $d = 0.5\%$. This is likely caused by a smaller sample size $k = \lceil N^{\text{syn}} d \rceil$ and a higher posterior variance under a similar level of bias.

TABLE S4

Sensitivity analysis of d for ABC-MH. We compare the inferential performance for c among ABC-MH with $d = 5\%$, ABC-MH with $d = 1\%$, ABC-MH with $d = 0.5\%$, MH_{true}, and MH_{default}. All values here are obtained from 200 independent replications. For each random replication at (c, σ^2), all methods were run for identical total CPU time and only converged chains from MH algorithms were included.

		Percent Bias(sd)		Coverage(sd)	
c	method	$\sigma^2 = 0.5$	$\sigma^2 = 1$	$\sigma^2 = 0.5$	$\sigma^2 = 1$
0.3	ABC-MH with $d = 5\%$	12(9.4)	13(9.9)	98(0.99)	99(0.71)
	ABC-MH with $d = 1\%$	13(9.8)	14(10)	97(1.2)	96(1.5)
	ABC-MH with $d = 0.5\%$	14(10)	15(11)	94(1.6)	92(1.9)
	$\text{MH}_{\mathsf{true}}$	13(9.8)	12(9.5)	94(2)	95(1.9)
	$\text{MH}_{\mathsf{default}}$	45(20)	46(20)	33(5.5)	30(6.1)
0.5	ABC-MH with $d = 5\%$	15(11)	15(11)	92(1.9)	93(1.8)
	ABC-MH with $d = 1\%$	15(12)	16(12)	88(2.3)	90(2.1)
	ABC-MH with $d = 0.5\%$	16(12)	16(12)	84(2.6)	86(2.4)
	$\text{MH}_{\mathsf{true}}$	11(9)	11(8.6)	97(1.7)	97(1.6)
	$\text{MH}_{\mathsf{default}}$	33(18)	31(19)	60(5.5)	57(5.7)
0.7	ABC-MH with $d = 5\%$	13(10)	14(11)	96(1.5)	93(1.8)
	ABC-MH with $d = 1\%$	13(10)	13(11)	94(1.7)	90(2.1)
	ABC-MH with $d = 0.5\%$	13(11)	14(11)	90(2.1)	89(2.2)
	$\text{MH}_{\mathsf{true}}$	12(9.1)	12(9.1)	95(2.6)	96(2.1)
	$\text{MH}_{\mathsf{default}}$	25(15)	27(16)	73(5.5)	69(5.9)
1.0	ABC-MH with $d = 5\%$	14(11)	14(13)	95(1.5)	94(1.6)
	ABC-MH with $d = 1\%$	14(10)	14(13)	88(2.3)	92(1.9)
	ABC-MH with $d = 0.5\%$	14(11)	15(13)	86(2.4)	86(2.4)
	$\text{MH}_{\mathsf{true}}$	11(7.6)	13(11)	97(2)	92(3.5)
	$\text{MH}_{\mathsf{default}}$	14(11)	16(14)	93(3.5)	89(3.8)

S4.4. *Sensitivity Analysis of the Number of Synthetic Data in ABC* To our knowledge, only two packages available from: (i) Neal (2003) on the website `https://www.cs.toronto.edu/~radford/dft.software.html` and (ii) Knowles and Ghahramani (2015) on the Github `https://github.com/davidaknowles/pydt`. Neal's code is implemented on `R`, and does not implement the inference algorithm, while Knowles programmed the C++ code from scratch including the library for the tree structure. However, the C++ libraries from Knowles and Ghahramani (2015) are deprecated and require additional updates for the version updates of the C++ compiler. Without additional documentation, the C++ code is hard to adapt in our context. Thus, we implemented our algorithm in `R` based on the existing libraries for tree structure (e.g. `ape` and `phylobase`) and the ABC algorithm (e.g. `ABC`).

The main computation bottleneck for our algorithm on `R` is the ABC stage (141 hours for 600,000 synthetic data), which is much slower than the MH stage (1.7 hours for 10,000 iterations) and the single stage MCMC (2.5 hours for 10,000 iterations). However, the ABC can be easily parallelized to reduce the wall-clock time given a sufficient number of CPU cores. In addition, we may reduce the number of synthetic data (N^{Syn}) in ABC to further improve speed. We have now conducted a simulation study to empirically demonstrate the acceleration of the ABC through the reduction of N^{Syn}. Specifically, we ran the ABC and measured the posterior median under a lower N^{Syn}.

We show the simulation results in Table S5. From Table S5, the $\widehat{\sigma}^2$ are relatively stable in terms of the mean and standard deviation under a lower N^{Syn}. On the other hand, the standard deviation of $\widehat{c}$ grows rapidly ($sd: 0.0280$ for $N^{\text{Syn}} = 600,000$ and $sd: 0.260$ for $N^{\text{Syn}} = 5,000$) when the N^{Syn} decreases. For our main analyses, we went with the conservative choice of $N^{\text{Syn}} = 600,000$ for the confirmatory results.

N^{Syn}	Total CPU Hour	$\widehat{c}$ (sd)	$\widehat{\sigma}^2$ (sd)
600,000	141	1.18 (0.0280)	1.87 (0.245)
300,000	70.5	1.18 (0.0278)	1.87 (0.246)
100,000	23.5	1.16 (0.0429)	1.87 (0.246)
50,000	11.8	1.18 (0.0707)	1.86 (0.235)
10,000	2.35	1.17 (0.159)	1.88 (0.240)
5,000	1.18	1.25 (0.260)	1.84 (0.249)

TABLE S5

The total CPU time and the median of the real parameters (mean and the standard deviation in the bracket) under different numbers of synthetic data (N^{Syn}) for the ABC stage. All values are obtained from 30 *independent replicates from the correct specified data generating mechanism. The underlying true* $c = 1.220$ *and* $\sigma^2 = 1.755$.

S5. Additional Simulation Results of $\mathbf{R_x}$-Trees In this Section, we provide more simulation results for the Section 4.2 in the Main Paper. We empirically compare the the proposed two-stage ABC-MH with the single-stage MCMC in terms of the MAP tree estimation (Section S5.1) and recovery of pairwise treatment similarities (Section S5.2).

Simulation setup. For the following simulations, we followed the same setup as in Section S4 with $I = 50$ and $J = 10$, and let c and σ^2 take values from $\{0.3, 0.5, 0.7, 1.0\}$ and $\{0.5, 1.0\}$, respectively. For each pair of (c, σ^2), 50 pairs of tree and data on the leaves were independently drawn based on the DDT model. For ABC, we generated $N^{\text{syn}} = 600,000$ synthetic data sets from the DDT model with threshold parameter $d = 0.5\%$. We assigned priors on $c \sim \mathsf{Gamma}(2, 2)$ and $1/\sigma^2 \sim \mathsf{Gamma}(1, 1)$ with shape and rate parameterization. We

compare the proposed algorithm against two alternatives based on MH algorithms (MH_{true} and $\text{MH}_{\text{default}}$. We ran MH algorithms (the 2nd stage of the proposed algorithm, MH_{true} and $\text{MH}_{\text{default}}$) with 10,000 iterations and discarded the first 7,000 iterations.

Performance metrics. We assess the accuracy of tree estimation using Billera–Holmes–Vogtmann (BHV) distance (Billera, Holmes and Vogtmann, 2001) between the true tree and the *maximum a posteriori* (MAP) tree obtained from ABC-MH, MH_{true} and $\text{MH}_{\text{default}}$, or between the true tree and the dendrogram obtained from hierarchical clustering, respectively. For the pairwise similarities, we follow the Section 4.1 and calculate iPCPs for all pairs of treatments and evaluate the iPCPs by correlation of correlation for estimated similarities and true branching time and the Frobenius norm for the overall matrix.

S5.1. *Recovery of the True Tree* The proposed two-stage algorithm decoupled the real and tree parameters, produced better inference for Euclidean parameters (See Section S4.2), resulting in better inference for the unknown treatment tree. In particular, Figure S9 shows that, in terms of the BHV distance, the MAP tree estimates from ABC-MH better recovers the trees than $\text{MH}_{\text{default}}$ and hierarchical clustering with Euclidean distance and squared Ward linkage (Hclust). On average, MAP from MH_{true} is the closest to the true underlying tree. However, MH_{true} requires knowledge about the truth and is unrealistic in practice. In addition, we observed that the chains from MH_{true} in fact did not mix well and were stuck at the initial values hence falsely appearing accurate. The second stage MH for sampling the tree built on the high-quality posterior samples of c and σ^2 obtained from the 1st stage ABC and produced better MAP tree estimates that are on average closer to the simulation truths than $\text{MH}_{\text{default}}$ and Hclust.

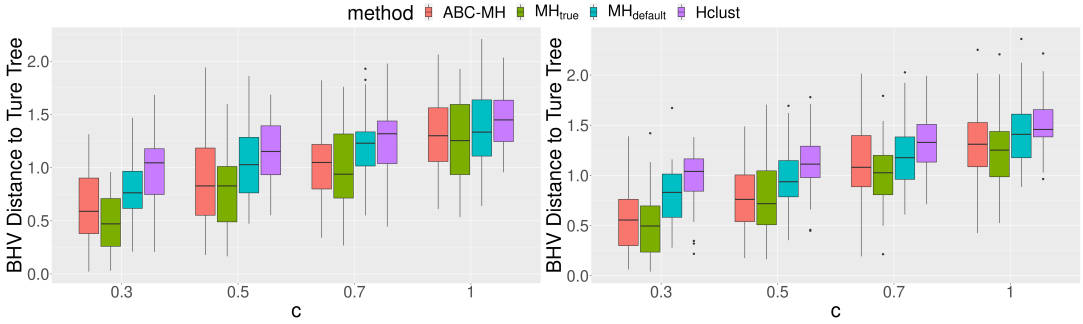


Figure S9: (Left) $\sigma^2 = 0.5$; (Right) $\sigma^2 = 1$. The BHV distance between the MAP estimate and the underlying tree for each algorithm. Results are based on 50 replications.

S5.2. *Estimation of Treatment Similarities* The two-stage algorithm also produces better iPCPs due to decoupling strategy and superior inference for Euclidean parameters in the first stage. Similar to the results for MAP, pairwise iPCPs from ABC-MH better recover the true branching time than $\text{MH}_{\text{default}}$, Hclust and Pearson correlation and reach similar quality to the iPCPs from MH_{true} (See Figure S10). Since MH_{true} requires unrealistic true parameters, MH_{true} is not attainable. From the simulations above, MAP and iPCPs from ABC-MH outperform $\text{MH}_{\text{default}}$ and take care of overall and local tree details, respectively. We apply the ABC-MH to obtain posterior DDT samples for the real data analysis section.

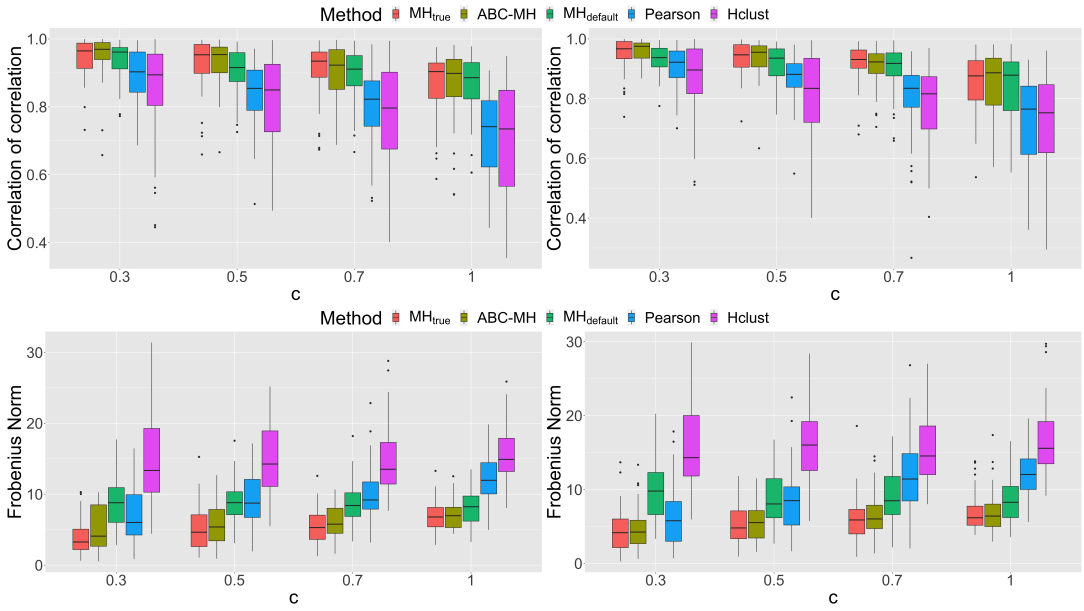


Figure S10: Under different c and σ^2, two-stage algorithm better estimates the pairwise similarities than classical single-stage MCMC in terms of correlation of correlation (upper panels) and Frobenius norm (lower panels). (Left) $\sigma^2 = 0.5$; (Right) $\sigma^2 = 1$. Results are based on 50 replications.

S5.3. *Computation Time of the Gaussian Likelihood Evaluation* Computationally, the complexity for the belief propagation is faster in theory, but the computation time also replies on the implementation. We empirically compare the running time of the evaluation of Gaussian likelihood on `R` for (i) the naive method of the Cholesky decomposition and (ii) the belief propagation algorithm. Specifically, we ran the `dmvnorm` function for naive method from the package `mvtnotm` and the `Marginals` function for belief propagation from the package `BayesNetBP` (Yu, Moharil and Blair, 2020). To our knowledge, the package `BayesNetBP` is the only `R` package implements exact belief propagation for the Gaussian data without commercial dependencies (Yu, Moharil and Blair, 2020). We ran each function 500 times on the Breast cancer data with the dimension of 20×38 given the same tree structure. All computation are executed on the same local computer of the Mac mini with M1 CPU and 8Gb memory. On `R`, the belief propagation (0.0566 second) is slower than the naive likelihood calculation (0.000148 second). The hindered belief propagation might be the result of the for-loop, which is slow in `R` (Burns, 2011).

S5.4. *Inference using the Whole Posterior Samples of c and σ^2* Our algorithm runs the approximate Bayesian computation (ABC) rejection algorithm (Sisson, Fan and Beaumont, 2019) to obtain the posterior samples of c and σ^2 and uses the posterior median of c and σ^2 as the common and fixed input for different chains of the MH algorithm. The ABC merges all synthetic data into a larger dataset and re-use the same synthetic data for different chains of the MH, which is advocated by Bertorelle, Benazzo and Mona (2010) and Blum et al. (2013). Under the ABC framework, the same synthetic data results in the identical posterior samples of c and σ^2 as the common input for different chains of the MH.

Once MH algorithm receives the posterior samples, another viable option is to use the whole posterior sample instead of using the fixed representative statistics only. We provide a set of simulations to empirically compare two algorithms using: (i) fixed posterior median only and (ii) the whole posterior samples. The algorithm (i) plugins the fixed posterior medians of c and σ^2, while the algorithm (ii) randomly picks one posterior sample at each iteration

in MH. Specifically, given L weighted posterior samples of c_l and σ_l^2, with the weights w_l^c and $w_l^\sigma, l = 1 \ldots, L$, algorithm (ii) draws a posterior sample of c_l and σ_l^2 with corresponding weights at each iteration. Eventually, we measure the results through the pairwise similarity with the correlation of correlation and the Frobenius norm.

We show our simulation results in Figure S11 using pairwise similarity. In Figure S11, the algorithm (i) (DDT) and (ii) (DDT.all) perform similarly in terms of the correlation of correlation (mean for DDT: $(0.944, 0.971, 0.981, 0.882)$ and DDT.all: $(0.945, 0.966, 0.979, 0.877)$) and the matrix norm (mean for DDT: $(1.154, 1.415, 1.558, 1.811)$ and DDT.all: $(1.156, 1.503, 1.516, 1.814)$) under four different data generating scenarios.

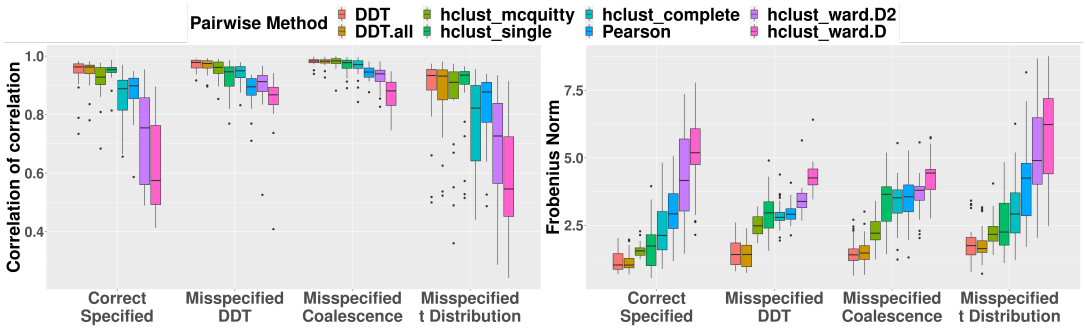


Figure S11: Simulation studies for comparing the quality of estimated treatment similarities based on DDT (DDT: median of (c, σ^2) and DDT.all: re-sample from the whole posterior samples of (c, σ^2)), hierarchical clustering, and empirical Pearson correlation. Two performance metrics are used: (Left) Correlation of correlation (higher values are better); (Right) Matrix distances with Frobenius norm for pairwise similarity and max norm for three-way similarity (lower values are better). DDT captures true similarity best under four levels of misspecification scenarios.

S5.5. *PDX Experiment with a Smaller Dimension* We investigated the performance of our method on smaller scale simulated datasets. Specifically, we applied our algorithm to two datasets with smaller dimensions (treatments, patients): 5×5 and 10×15. We show the simulation results in Figure S12 through the pairwise similarity (the correlation of correlation and the Frobenius norm). Overall, our algorithm outperforms the distance based hierarchical clustering (hclust) and the pairwise Pearson correlation in terms of the pairwise similarity except for two cases. Specifically, our algorithm is the best or the second best except for two cases: (i) the correlation of correlation under the scenario of the misspecified t-distribution with the dimension of 5×5 and (ii) the Frobenius norm under the scenario of the misspecified DDT with the dimension of 10×15. However, even under these two cases, our algorithm still have a highest lower bound in case (i) and a lowest upper bound of the Frobenius norm in case (ii), which indicates the advantage of avoiding the worst case for our algorithm. In summary, under the $1 \times 1 \times 1$ experimental design, we recommend our algorithm even under an extremely small dataset such as the dimension of 5 by 5, given enough computation resources.

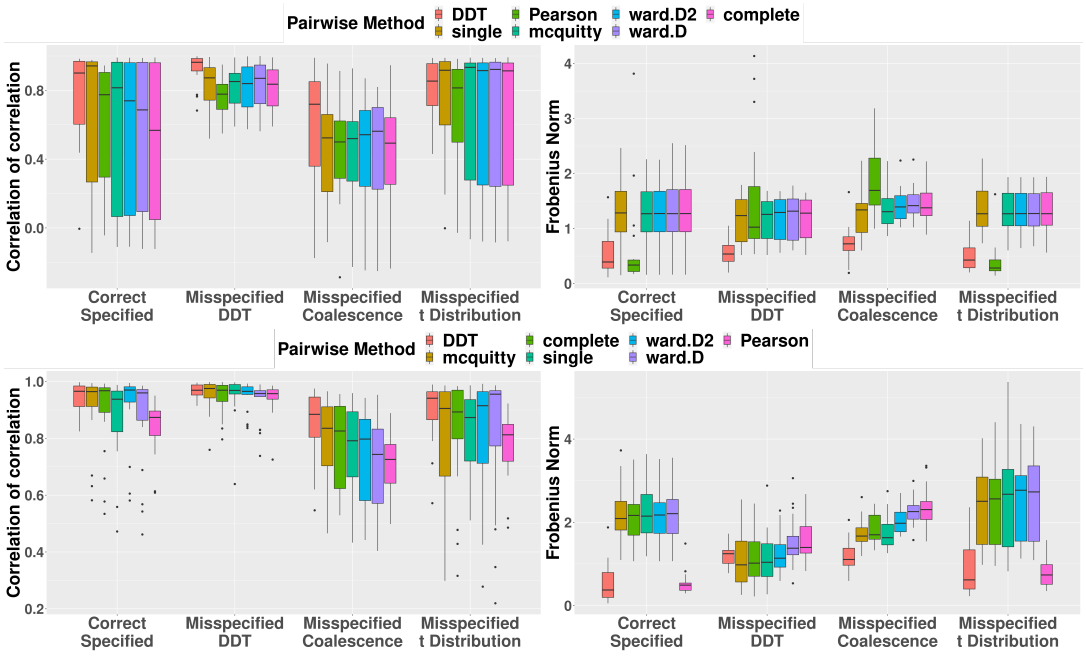


Figure S12: The pairwise similarity for the PDX experiment with a small number of dimensions. (top): 5 treatments and 5 patients; (bottom): 10 treatments and 15 patients. The results are obtained through 30 replicates.

S6. Additional Results for PDX Analysis In this section, we provide the pre-processing procedures of NIBR-PDXE and present the results for non-small lung cancer (NSCLC) and pancreatic ductal adenocarcinoma (PDAC) with tables including treatment and pathway information.

S6.1. *PDX Data Pre-Processing* We followed pre-processing procedure in Rashid et al. (2020) and imputed the missing data by k-nearest neighbor method. We take the best average response (BAR) as the response and scale the BAR by the standard deviation over all patients, treatments and across five cancers. Since the scaled BAR contains missing values, we impute the missing data by the k-nearest neighbor with $k = 10$ and compare all treatments to the untreated group. Specifically, we take $x_{ij} = \mathrm{BAR}_{ij} - \mathrm{BAR}_{0j}, i = 1 \dots I, j = 1 \dots J$ as the observed data, where $\mathrm{BAR}_{0,j}$ is the untreated BAR for patient j.

S6.2. *Test for Distributional Assumption* Our main interest of the paper is the tree-structured covariance that models the treatment similarity. The relevant class of distributions for modeling thus consists of those whose properties are fully described through a tree-structured covariance matrix (with mean equal to zero). A natural candidate is the parameterized family of mean-zero symmetric elliptical distributions indexed by tree-structured covariance matrices, which includes the Gaussian as a special case.

From a methodological perspective, restriction in the paper to the Gaussian setting is to be viewed as a first step towards modelling using the more general elliptical family, mainly driven by computational considerations and interpretability within the context of the scientific application. Notwithstanding this, the Gaussian setup, which facilitates scalable and explicit computations, does not appear unreasonable: multivariate normality tests with the multivariate qq-plot (Figure S13) demonstrate that BRCA (panel (A)) and CM (panel (B))

roughly fall on the 45-degree line, but CRC (panel (C)) slightly deviates from the the 45-degree lines indicating some departure from normality; this is further corroborated with the Doornik-Hansen (Doornik and Hansen, 2008) multivariate normal test, which resulted in p-values 0.0969 (BRCA), 0.0833 (CM) and <0.001 (CRC) for testing the null hypothesis that the responses were Gaussian.

With an eye towards future extensions to the elliptical family, we carried out hypothesis tests to assess the multivariate elliptical symmetry assumption; using the test proposed by Babic et al. (2021) available in the `R` package `ellipticalsymmetry` (Babic, Ley and Palangetic, 2021), we fail to reject the null hypothesis of elliptical symmetry with the p-values of 0.6805 for BRCA, 0.8679 for CRC, and 0.4385 for CM.

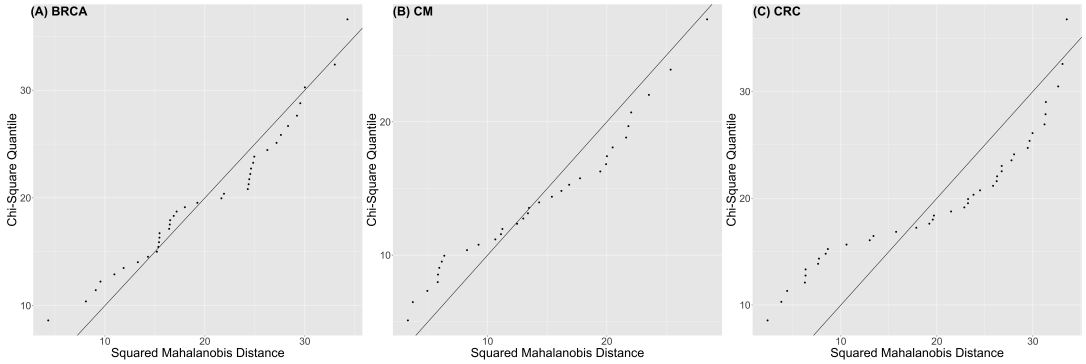


Figure S13: The multivariate normality QQ-plot for (A) breast cancer, (B) melanoma, and (C) colorectal cancer

S6.3. *Threshold of the Co-Clustering* Generally, it is hard to recommend a universal threshold for co-clustering without considering unique patterns in each dataset. For example, different cancers may respond differently to treatments, resulting in varying degrees of tumor size shrinkage. This is reflected by the varying distributions for all the pairwise iPCPs obtained from datasets for three cancers (BRCA, CRC and CM); See the three sets of different empirical quantiles in Table S6. Recognizing the practical utility of iPCP cutoffs, in the following, we use pairwise iPCPs to illustrate a practical strategy for determining such cut-offs; similarly for multi-way iPCPs.

First, for a “fully-exploratory" analysis, where one does not assume any prior knowledge about multiple monotherapies that share the same mechanism, we recommend ranking all the pairwise iPCPs as in Table S6 and setting the cut-off at the 75-th percentile.

Second, for a “partially-exploratory" analysis, where one incorporates prior knowledge by assuming the PDX dataset contains two or more specific monotherapies with known and the same mechanism, we recommend using a cut-off determined by their corresponding iPCP. For example, two treatments (BKM120 and BYL719) are both PI3K inhibitors and were tested in the BRCA data with a pairwise iPCP of 0.8002, which we recommend as a practical cut-off. If multiple such iPCPs are available for other pairs of treatments with a common mechanism, we recommend the lowest iPCP as the cut-off. In this scenario, a question may be raised regarding whether the biologically-motivated cut-off is similar to the cut-off determined by the empirical 75 percentile and which one to use. In fact, we observed that two cut-offs were practically similar. For example, the 75-th percentile of all pairwise iPCPs for BRCA is 0.753 and two treatments (binimetinib and BKM120) targeting the same pathway PI3K-MAPK-CDK have a pairwise iPCP of 0.7427. As another example, in the CM data set,

the 75-th percentile of pairwise iPCPs is 0.801; the two treatments (LEE011, binimetinib) targeting the same pathway PI3K-MAPK-CDK have a iPCP of 0.8210. In practice, when both are available, we recommend using the 75 percentile cut-off for fully-exploratory analyses and using the biologically-motivated cut-off for partially-exploratory analyses.

Cancer	Min	25-th	Median	75-th	Max
BRCA	0.357	0.664	0.680	0.753	0.899
CRC	0.420	0.441	0.515	0.687	0.862
CM	0.610	0.723	0.742	0.801	0.939

TABLE S6

The descriptive statistics for all possible pairs of pairwise iPCP for the breast cancer (top), colorectal cancer (middle) and the melanoma (bottom).

S6.4. *Additional Results for Monotherapy* In Main Paper, we listed the results for monotherapies targeting the cell regulated pathways. We offer more monotherapies targeting the rest two categories of the pathways.

ERBB3 and tubulin inhibitors. Our model also found high iPCP values among ERBB3, tubulin and PI3K-MAPK-CDK inhibitors in BRCA. ERRB3 inhibitor, LJM716, exhibits high pairwise iPCP values with PI3K (BKM120: 0.7501, BYL719: 0.7513, CLR457: 0.7500), MAPK (binimetinib: 0.7811), CDK (LEE011: 0.7847) and tubulin (paclitaxel: 0.7505) inhibitors. Since PI3K and MAPK are downstream pathways of ERBB3 (Balko et al., 2012) and CDK works closely with PI3K and MAPK (Kurtzeborn, Kwon and Kuure, 2019; Repetto et al., 2018), high iPCPs between ERBB3 inhibitor and PI3K-MAPK-CDK inhibitors are not surprising. For ERBB3 and tubulin, ERBB3 is a critical regulator of microtubule assembly (Wu et al., 2021) and tubulin plays an important role in building microtubules. Since microtubules form the skeletons of cells and are essential for cell division (Gunning et al., 2015; Haider et al., 2019), tubulin inhibitor, paclitaxel, kills cancer cell by interfering cell division and is an FDA-approved treatment. In congruence with the above results, tubulin inhibitor paclitaxel also shares high iPCPs with PI3K (BKM120: 0.8076, BYL719: 0.8063, CLR457: 0.8076), MAPK (binimetinib: 0.7433), CDK (LEE011: 0.7587) and ERBB3 (LJM716: 0.7505). In addition, another CDK4 inhibitor BPT also inhibits tubulin (Mahale et al., 2015) and PI3K inhibitor BKM120 inhibits the formation of microtubule (Bohnacker et al., 2017). Both offer additional reasons for high iPCP between tubulin and PI3K-MAPK-CDK inhibitors.

MDM2 inhibitors. We found two drugs: CGM097 and HDM201 share high iPCP values in BRCA (0.8365) and CRC (0.7860). Since CGM097 and HDM201 target the same pathway, MDM2, high iPCPs suggest a high similarity between CGM097 and HDM201 and show consistent results between our model and underlying biological mechanism. MDM2 negatively regulates the tumor suppressor, p53 (Zhao, Yu and Hu, 2014) and if MDM2 is suppressed by inhibitors, p53 is able to prevent tumor formation. Both CGM097 and HDM201 entered phase I clinical trial (Konopleva et al., 2020) for wild-type p53 solid tumors and leukemia, respectively.

S6.5. *R_x-Tree for Non-Small Lung Cancer (NSCLC) and Pancreatic Ductal Adenocarcinoma (PDAC)* We applied the R_x-tree on the rest two cancers in the data: non-small lung cancer (NSCLC) and pancreatic ductal adenocarcinoma (PDAC). Similar to the Figure 5 in the Main Paper, R_x-tree, pairwise iPCP and (scaled) Pearson correlation are shown in the

left, middle and right panels in Figure S14, respectively. Again, we observe that the R_x-tree and the pairwise iPCP matrix show the similar clustering patterns. For example, three PI3K inhibitors (BKM120, BYL719 and CLR457) and a combination therapy (BKM120 + binimetibin) in NSCLC form a tight subtree and are labeled by a box in the R_x-tree of Figure S14 and a block with higher values of iPCP among therapies above also shows up in the corresponding iPCP matrix. The R_x-tree roughly clusters monotherapies targeting oncogenic process (PI3K-MAPK-CDK, MDM2 and JAK) and agrees with the biology mechanism. For example, three PI3K inhibitors (BKM120, BYL719 and CLR457) belong to a tighter subtree in both cancers. Following the same idea as the Main Paper, we further quantify the treatment similarity through iPCP. However, compared to three cancers (BRCA, CRC and CM) in the Main Paper, different problems of model fitting or interpretation lie in NSCLC and PDAC: NSCLC deviates from the normal assumption of Equation (4) (Figure S13) and PDAC shows lower iPCP (average iPCP of PDAC: 0.4119 $<$ BRCA: 0.6734, CRC: 0.5653, CM: 0.7535, NSCLC: 0.5817). For concerns raised above, we only verify the model through the monotherapies with known biology for each cancer.

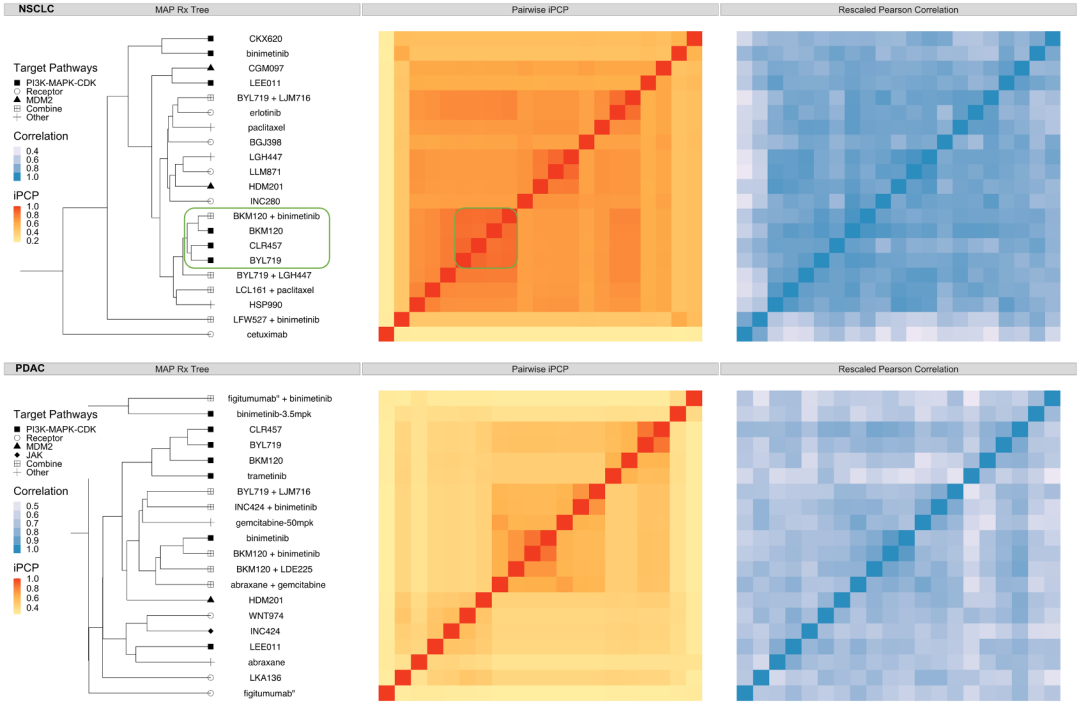


Figure S14: The R_x-tree and iPCP for non-small cell lung cancer (NSCLC, top row) and pancreatic ductal adenocarcinoma (PDAC, lower row). Three panels in each row represent: (left) estimated R_x-tree (MAP); distinct external target pathway information is shown in distinct shapes for groups of treatments on the leaves; (middle) Estimated pairwise iPCP, i.e., the posterior mean divergence time for pairs of entities on the leaves (see the result paragraph for definition for any subset of entities); (right) Scaled Pearson correlation for each pair of treatments. The Pearson correlation $\rho \in [-1, 1]$ was scaled by $\frac{\rho+1}{2}$ to fall into $[0, 1]$. Note that the MAP visualizes the hierarchy amongst treatments; the iPCP is not calculated based on the MAP, but based on posterior tree samples (see definition in Main Paper Section 3.2)

Non-small cell lung cancer. Our model suggests high iPCP values for treatments share the same target. For example, our model shows a high iPCP among three PI3K (BKM120,

BYL719 and CLR457) inhibitors: (BKM120, BYL719): 0.8402, (BKM120, CLR457): 0.8321, (BYL719, CLR457): 0.8710. For treatments with different targets, our model also exhibits a high iPCP values. For example, the monotherapy HSP990 that inhibits the heat shock protein 90 (HSP90) shows a high iPCP with PI3K inhibitors ((BKM120, HSP990): 0.7108, (BYL719, HSP990): 0.7114, (CLR457,HSP990): 0.7109). Since the inhibiting of HSP90 also suppresses PI3K (Giulino-Roth et al., 2017), it is not surprising to ses a high iPCP between PI3K and HSP90 inhibitors.

Pancreatic ductal adenocarcinoma. For PDAC, our model overall suggests a lower iPCP (average iPCP of PDAC: 0.4119). Out of 91 pairs of monotherapies, only BYL719 and CLR457 share a higher iPCP (0.8415). The higher iPCP can be explained by the common target PI3K of BYL719 and CLR457.

S6.6. *R Shiny Application* We illustrate the input and outputs of the proposed method via a R Shiny application hosted on the web (Figure S15). The visualizations are based on completed posterior computations for illustrative purposes. A user needs to specify the following inputs:

(A) Cancer type to choose the subset of data for analysis
(B) Number of treatments of interest in the subset $\mathcal{A}$ to evaluate synergy via iPCP
(C) Names of the treatments in the subset $\mathcal{A}$

Given the inputs above, the Shiny app visualizes the outputs:

(D) *maximum a posteriori* treatment tree for all the available treatments
(E) $\text{PCP}_{\mathcal{A}}(t)$ curve for the subset of treatments, $\mathcal{A}$
(F) $\text{iPCP}_{\mathcal{A}}$ value calculated from the corresponding $\text{PCP}_{\mathcal{A}}(t)$

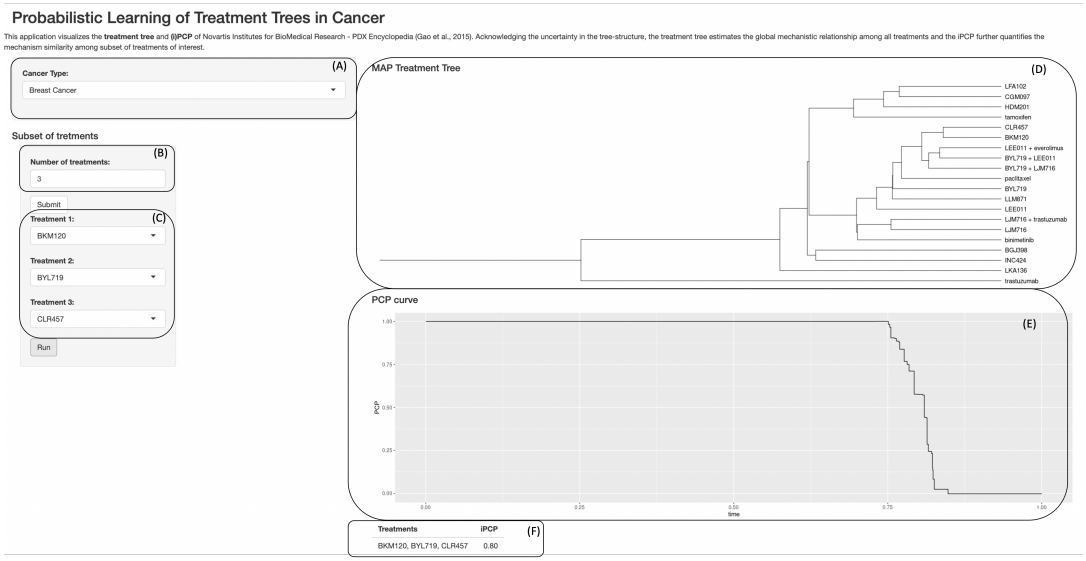


Figure S15: R Shiny app screenshot for illustrating model inputs and outputs for analyzing PDX data (20 treatments for breast cancer); the PCP curve and iPCP value are computed for a subset of three selected treatments.

S7. Random Effects Model for Multiple Animals Design The current work is built upon the $1 \times 1 \times 1$ design, but multi-replicate experiment set-up is extremely relevant in practice, and is an interesting direction for future work. Several possible modeling options can extend our work to adapt to the multi-replicate experimental design. Following the comment, we consider two different scenarios for the response: (i) homogeneous and (ii) heterogeneous responses. First, recent literature (Evrard et al., 2020) suggests robustness for PDX studies (including BAR and other tumor volume measurements) under different protocol and mice replicates and implies the homogeneous responses. When the responses are homogeneous, we can simply average the outcomes over the replicates, which makes out method directly applicable. Alternatively, when the responses are heterogeneous, we can use random effects for multiple replicates nested within each patient. We can incorporate the random effects either in the mean structure or in the variance structure. Specifically, given a PDX experiment with I treatments and J patients, for each treatment, we consider K_j independent mice replicates for the j-th patient, $j = 1, \ldots, J$. Let $\boldsymbol{X}_{\cdot jk} = [X_{1jk}, \ldots, X_{Ijk}] \in \mathbb{R}^I$ be a vector of BAR response across I treatments from the k-th replicate of patient j. Following Proposition 1, we may consider adding random effects in the mean structure:

$$\boldsymbol{X}_{\cdot jk} \overset{iid}{\sim} \boldsymbol{N}_I(\boldsymbol{\mu}_{jk}, \sigma^2 \boldsymbol{\Sigma}^{\mathcal{T}});\ \boldsymbol{\mu}_{jk} \sim \boldsymbol{N}_I(\mathbf{0}, \boldsymbol{\Omega}),\ j = 1, \ldots, J; k = 1, \ldots, K_j,$$

where the $\boldsymbol{\mu}_{jk} = [\mu_{1jk}, \ldots, \mu_{Ijk}]$ is the normal random effect with mean zero and a variance $\boldsymbol{\Omega}$. We assume $\boldsymbol{\Omega}$ to be diagonal to maintain the ultrametric property for the marginal variance of $\text{Var}(\boldsymbol{X}_{\cdot jk}) = \sigma^2 \boldsymbol{\Sigma}^{\mathcal{T}} + \boldsymbol{\Omega}$.

One may instead include random effects in the variance and the corresponding tree-structured matrix. Following the same notation, we can formulate the distribution as

$$\boldsymbol{X}_{\cdot jk} \overset{iid}{\sim} \boldsymbol{N}_I(\mathbf{0}, \sigma^2 \boldsymbol{\Sigma}^{\mathcal{T}}_k),\ k = 1, \ldots, K_j,$$

where $\boldsymbol{\Sigma}^{\mathcal{T}}_k$ is the tree-structured matrix for each replicate. We can further consider two cases with (i) pooling all tree-structured matrix of $\boldsymbol{\Sigma}^{\mathcal{T}}_k = \boldsymbol{\Sigma}^{\mathcal{T}}$ for all $k = 1, \ldots, K_j$ and (ii) assigning different $\boldsymbol{\Sigma}^{\mathcal{T}}_k$ for each k. The case (i) of pooling all tree-structured matrix is the same as the original Proposition 1 and ignores the heterogeneity of the responses. For the case (ii), we can further assign a prior distribution on each tree-structured matrix and include the external covariate information (e.g. heterogeneity of the response) in the prior distribution.

TABLE S7
Full CPUs series used for computations.

Intel Xeon X series	X5660@2.80GHz X5680@3.33GHz
Intel Xeon E series	E5-24400@2.40GHz E5-24700@2.30GHz E5-24500@2.10GHz E5-2650v3@2.30GHz E5-2650v4@2.20GHz E5-2690v4@2.60GHz E5-2690v4@2.60GHz

TABLE S8
Pathways full names and the corresponding abbreviations.

Abbreviation	Target Name
PI3K	Phosphoinositide 3-kinases
CDK	Cyclin-dependent kinases
MAPK	Mitogen-activated protein kinases
JAK	Janus kinase
MDM2	Murine double minute 2
BRAF	Serine/threonine-protein kinase B-Raf
MTOR	Mechanistic target of rapamycin
EGFR/ERBB	Epidermal growth factor receptor
SMO	Smoothened
TNKS	Tankyrase
PIM	Proto-oncogene serine/threonine-protein kinase Pim-1
BIRC2	Baculoviral IAP repeat-containing protein 2
IGF1R	Insulin-like growth factor 1 receptor

TABLE S9

Monotherapy names with targets. Different target groups are labeled differently in the Figure 5 and Figure S14.

Treatment name	Other names	Trade name	Target	Target Group
5FU	Fluorouracil	Adrucil	chemotherapy	Other
abraxane	nab-paclitaxel	abraxane	Tubulin	Other
BGJ398	Infigratinib		FGFR	Receptor
binimetinib	MEK162	Mektovi	MAPK	PI3K-MAPK-CDK
BKM120	Buparlisib		PI3K	PI3K-MAPK-CDK
BYL719	Alpelisib	Piqray	PI3K	PI3K-MAPK-CDK
cetuximab		Erbitux	EGFR	Receptor
CGM097			MDM2	MDM2
CKX620			MAPK	PI3K-MAPK-CDK
CLR457			PI3K	PI3K-MAPK-CDK
dacarbazine		DTIC-Dome	chemotherapy	Other
encorafenib	LGX818	Braftovi	BRAF	BRAF
erlotinib	Erlotinib hydrochloride	Tarceva	EGFR	Receptor
figitumumab	CP-751871		IGF1R	Receptor
gemcitabine		Gemzar	chemotherapy	Other
HDM201	Siremadlin		MDM2	MDM2
HSP990			HSP90	Other
INC280	Capmatinib	Tabrecta	MET	Receptor
INC424	Ruxolitinib	Jakafi and Jakavi	JAK	JAK
LDE225	Sonidegib	Odomzo	SMO	Receptor
LDK378	Ceritinib	Zykadia	ALK	Receptor
LEE011	Ribociclib	Kisqal	CDK	PI3K-MAPK-CDK
LFA102			PRLR	Receptor
LGH447			PIM	Other
LGW813			IAP	Other
LJC049			TNKS	Other
LJM716	Elgemtumab		ERBB3	Receptor
LKA136			NTRK	Receptor
LLM871			FGFR2/4	Receptor
paclitaxel		Taxol	Tubulin	Other
tamoxifen		Nolvadex	ESR1	Receptor
TAS266			DR5	Receptor
trametinib	GSK1120212	Mekinist	MAPK	PI3K-MAPK-CDK
trastuzumab		Herceptin	ERBB2	Receptor
WNT974			PORCN	Receptor

TABLE S10
Combination therapy full names with known targets.

Combination Therapies	Known Target Pathways	Cancer
abraxane+gemcitabine	Tubulin+chemotherapy	PDAC
BKM120+binimetinib	PI3K+MAPK	NSCLC,PDAC
BKM120+encorafenib	PI3K+BRAF	CM
BKM120+LDE225	PI3K+SMO	PDAC
BKM120+LJC049	PI3K+TNKS	CRC
BYL719+binimetinib	PI3K+MAPK	CRC
BYL719+cetuximab	PI3K+EGFR	CRC
BYL719+cetuximab+encorafenib	PI3K+EGFR+BRAF	CRC
BYL719+encorafenib	PI3K+BRAF	CRC
BYL719+LEE011	PI3K+CDK	BRCA
BYL719+LGH447	PI3K+PIM	NSCLC
BYL719+LJM716	PI3K+ERBB3	BRCA,CRC,NSCLC,PDAC
cetuximab+encorafenib	EGFR+BRAF	CRC
encorafenib+binimetinib	BRAF+MAPK	CM
figitumumab+binimetinib	IGF1R+MAPK	PDAC
INC424+binimetinib	JAK+MAPK	PDAC
LCL161+paclitaxel	BIRC2+Tubulin	NSCLC
LEE011+encorafenib	CDK+BRAF	CM
LEE011+everolimus	CDK+MTOR	BRCA
LFW527+binimetinib	IGF1R+MAPK	NSCLC
LJM716+trastuzumab	ERBB3+ERBB2	BRCA

REFERENCES

BABIC, S., LEY, C. and PALANGETIC, M. (2021). The R Journal: Elliptical Symmetry Tests in R. *The R Journal* **13** 661-672. https://doi.org/10.32614/RJ-2021-078.

BABIC, S., GELBGRAS, L., HALLIN, M. and LEY, C. (2021). Optimal tests for elliptical symmetry: Specified and unspecified location. *Bernoulli* **27** 2189 – 2216.

BALKO, J. M., MILLER, T. W., MORRISON, M. M., HUTCHINSON, K., YOUNG, C., RINEHART, C., SÁNCHEZ, V., JEE, D., POLYAK, K., PRAT, A., PEROU, C. M., ARTEAGA, C. L. and COOK, R. S. (2012). The receptor tyrosine kinase ErbB3 maintains the balance between luminal and basal breast epithelium. *Proc Natl Acad Sci U S A* **109** 221–226.

BEAUMONT, M. A., ZHANG, W. and BALDING, D. J. (2002). Approximate Bayesian Computation in Population Genetics. *Genetics* **162** 2025–2035.

BERTORELLE, G., BENAZZO, A. and MONA, S. (2010). ABC as a flexible framework to estimate demography over space and time: some cons, many pros. *Mol Ecol* **19** 2609–2625.

BIAU, G., CÉROU, F. and GUYADER, A. (2015). New insights into approximate Bayesian computation. *Ann. Inst. H. Poincaré Probab. Statist.* **51** 376–403.

BILLERA, L. J., HOLMES, S. P. and VOGTMANN, K. (2001). Geometry of the Space of Phylogenetic Trees. *Advances in Applied Mathematics* **27** 733 - 767.

BLUM, M. G. B. (2010). Approximate Bayesian computation: A nonparametric perspective. *Journal of the American Statistical Association* **105** 1178–1187.

BLUM, M. G. B., NUNES, M. A., PRANGLE, D. and SISSON, S. A. (2013). A Comparative Review of Dimension Reduction Methods in Approximate Bayesian Computation. *Statistical Science* **28** 189 – 208.

BOHNACKER, T., PROTA, A. E., BEAUFILS, F., BURKE, J. E., MELONE, A., INGLIS, A. J., RAGEOT, D., SELE, A. M., CMILJANOVIC, V., CMILJANOVIC, N., BARGSTEN, K., AHER, A., AKHMANOVA, A., DÍAZ, J. F., FABBRO, D., ZVELEBIL, M., WILLIAMS, R. L., STEINMETZ, M. O. and WYMANN, M. P. (2017). Deconvolution of Buparlisib's mechanism of action defines specific PI3K and tubulin inhibitors for therapeutic intervention. *Nat Commun* **8** 14683.

BRAVO, H. C., WRIGHT, S., ENG, K. H., KELES, S. and WAHBA, G. (2009). Estimating tree-structured covariance matrices via mixed-integer programming. *J Mach Learn Res* **5** 41–48.

BURNS, P. (2011). *The R Inferno*. Lulu.com.

COOK, S. R., GELMAN, A. and RUBIN, D. B. (2006). Validation of software for Bayesian models using posterior quantiles. *Journal of Computational and Graphical Statistics* **15** 675–692.

DOORNIK, J. A. and HANSEN, H. (2008). An Omnibus Test for Univariate and Multivariate Normality*. *Oxford Bulletin of Economics and Statistics* **70** 927-939.

EVRARD, Y. A., SRIVASTAVA, A., RANDJELOVIC, J., DOROSHOW, J. H., DEAN, D. A., MORRIS, J. S. and CHUANG, J. H. (2020). Systematic Establishment of Robustness and Standards in Patient-Derived Xenograft Experiments and Analysis. *Cancer Res* **80** 2286–2297.

GELMAN, A., CARLIN, J. B., STERN, H. S. and RUBIN, D. B. (2013). *Bayesian Data Analysis*, 3rd ed. ed. Chapman and Hall/CRC.

GEWEKE, J. (1992). Evaluating the Accuracy of Sampling-Based Approaches to the Calculation of Posterior Moments. In *In Bayesian Statistics* 169–193. University Press.

GEYER, C. J. (2011). Introduction to Markov chain Monte Carlo. In *Handbook of Markov chain Monte Carlo* (S. Brooks, A. Gelman, G. L. Jones and X.-L. Meng, eds.) 1, 3-48. Chapman and Hall/CRC.

GIULINO-ROTH, L., VAN BESIEN, H. J., DALTON, T., TOTONCHY, J. E., RODINA, A., TALDONE, T., BOLAENDER, A., ERDJUMENT-BROMAGE, H., SADEK, J., CHADBURN, A., BARTH, M. J., DELA CRUZ, F. S., RAINEY, A., KUNG, A. L., CHIOSIS, G. and CESARMAN, E. (2017). Inhibition of Hsp90 Suppresses PI3K/AKT/mTOR Signaling and Has Antitumor Activity in Burkitt Lymphoma. *Mol Cancer Ther* **16** 1779–1790.

GUNNING, P. W., GHOSHDASTIDER, U., WHITAKER, S., POPP, D. and ROBINSON, R. C. (2015). The evolution of compositionally and functionally distinct actin filaments. *J Cell Sci* **128** 2009–2019.

HAIDER, K., RAHAMAN, S., YAR, M. S. and KAMAL, A. (2019). Tubulin inhibitors as novel anticancer agents: an overview on patents (2013-2018). *Expert Opin Ther Pat* **29** 623–641.

KNOWLES, D. A. and GHAHRAMANI, Z. (2015). Pitman-Yor diffusion trees for Bayesian hierarchical clustering. *IEEE Trans Pattern Anal Mach Intell* **37** 271–289.

KONOPLEVA, M., MARTINELLI, G., DAVER, N., PAPAYANNIDIS, C., WEI, A., HIGGINS, B., OTT, M., MASCARENHAS, J. and ANDREEFF, M. (2020). MDM2 inhibition: an important step forward in cancer therapy. *Leukemia* **34** 2858–2874.

KURTZEBORN, K., KWON, H. N. and KUURE, S. (2019). MAPK/ERK Signaling in regulation of renal differentiation. *Int J Mol Sci* **20**.

MAHALE, S., BHARATE, S. B., MANDA, S., JOSHI, P., JENKINS, P. R., VISHWAKARMA, R. A. and CHAUDHURI, B. (2015). Antitumour potential of BPT: a dual inhibitor of CDK4 and tubulin polymerization. *Cell Death Dis* **6** e1743.

MURTAGH, F. and LEGENDRE, P. (2014). Ward's Hierarchical Agglomerative Clustering Method: Which Algorithms Implement Ward's Criterion? *Journal of Classification* **31** 274-295.

NEAL, R. (2003). Density modeling and clustering using Dirichlet diffusion trees. *Bayesian Statistics* **7** 619-629.

PRANGLE, D., BLUM, M. G. B., POPOVIC, G. and SISSON, S. A. (2014). Diagnostic tools for approximate Bayesian computation using the coverage property. *Australian & New Zealand Journal of Statistics* **56** 309-329.

RASHID, N. U., LUCKETT, D. J., CHEN, J., LAWSON, M. T., WANG, L., ZHANG, Y., LABER, E. B., LIU, Y., YEH, J. J., ZENG, D. and KOSOROK, M. R. (2020). High-Dimensional Precision Medicine From Patient-Derived Xenografts. *Journal of the American Statistical Association* **0** 1-15.

REPETTO, M. V., WINTERS, M. J., BUSH, A., REITER, W., HOLLENSTEIN, D. M., AMMERER, G., PRYCIAK, P. M. and COLMAN-LERNER, A. (2018). CDK and MAPK synergistically regulate signaling dynamics via a shared multi-site phosphorylation region on the scaffold protein Ste5. *Mol Cell* **69** 938–952.

SCOTT, S. L., BLOCKER, A. W., BONASSI, F. V., CHIPMAN, H. A., GEORGE, E. I. and MCCULLOCH, R. E. (2016). Bayes and big data: The consensus Monte Carlo algorithm. *International Journal of Management Science and Engineering Management* **11** 78–88.

SISSON, S. A., FAN, Y. and BEAUMONT, M. (2019). *Handbook of Approximate Bayesian Computation*, 1st ed. ed. Chapman and Hall/CRC.

WU, L., ISLAM, M. R., LEE, J., TAKASE, H., GUO, S., ANDREWS, A. M., BUZHDYGAN, T. P., MATHEW, J., LI, W., ARAI, K., LO, E. H., RAMIREZ, S. H. and LOK, J. (2021). ErbB3 is a critical regulator of cytoskeletal dynamics in brain microvascular endothelial cells: implications for vascular remodeling and blood-brain-barrier modulation. *J Cereb Blood Flow Metab* 271678X20984976.

YU, H., MOHARIL, J. and BLAIR, R. H. (2020). BayesNetBP: An R Package for Probabilistic Reasoning in Bayesian Networks. *Journal of Statistical Software* **94** 1–31.

ZHAO, Y., YU, H. and HU, W. (2014). The regulation of MDM2 oncogene and its impact on human cancers. *Acta Biochim Biophys Sin (Shanghai)* **46** 180–189.