

Joint inference of exclusivity patterns and recurrent trajectories from tumor mutation trees: Supplementary material

Xiang Ge Luo^{1,2}, Jack Kuipers^{1,2}, Niko Beerenwinkel^{1,2*}

¹Department of Biosystems Science and Engineering, ETH Zurich, Mattenstrasse 26, 4058 Basel, Switzerland

²SIB Swiss Institute of Bioinformatics, Mattenstrasse 26, 4058 Basel

Contents

A	Tree generating process and parameter estimation methods for TreeMHN	2
A.1	Tree generating process pseudocode	2
A.2	Tree generating process as a continuous-time Markov chain	3
A.3	Maximum likelihood estimation of an MHN given a set of trees	8
A.4	Hybrid EM and Monte Carlo EM algorithm for the estimation of an MHN given a set of trees	10
A.5	Parameter estimation based on non-empty trees	12
B	TreeMHN with stability selection	13
C	Computing probabilities of trajectories and mutational events with TreeMHN	14
C.1	Most probable evolutionary trajectories	14
C.2	Alternative methods to compute the probabilities of mutational events	14
D	Additional simulation details	16
D.1	Precision and recall of identifying the true MHN	16
D.2	Computing trajectory probabilities with REVOLVER and HINTRA	16
D.3	Introducing noise into simulated trees	17
D.4	Additional figures from simulations	18
E	Application to acute myeloid leukemia data	23
E.1	Preprocessing of the data	23
E.2	Additional figures	23
F	Application to non-small-cell lung cancer data	30
F.1	Preprocessing of the data	30
F.2	Additional figures	30
G	Application to breast cancer data	33
G.1	Preprocessing of the data	33
G.2	Additional figures	34
	References	39

*Correspondence: niko.beerenwinkel@bsse.ethz.ch

A Tree generating process and parameter estimation methods for TreeMHN

A.1 Tree generating process pseudocode

In Algorithm 1 we provide the pseudocode for the tree-generating process described in the Methods section. A tree \mathcal{T} consists of a vertex set V and an edge set E . Each node in the vertex set is of the form (i, π_i) , where i is the unique index of the node in \mathcal{T} , and π_i is a mutational trajectory that runs from the root to the node i . An illustrative example is given in Supplementary Figure S1(a) and (b).

```

Input:
   $n$ : Number of mutations
   $\Theta$ : Mutual Hazard Network represented by an  $n$ -by- $n$  matrix
   $\lambda_s$ : Sampling rate
Output:
   $\mathcal{T}$ : A tree structure
1  $t_s \sim \text{Exp}(\lambda_s)$  // Draw a sampling time
2  $V \leftarrow \{(1, (0))\}$  // The root node has index 1
3  $E \leftarrow \emptyset$  // The edge set is empty
4  $t_{(0)} = 0$  // The root node is at time 0
5  $\mathcal{U}_{\text{current}} \leftarrow \{1\}$  // Index set of the nodes to be visited
6 while  $\mathcal{U}_{\text{current}}$  is non-empty do
7    $\mathcal{U}_{\text{next}} \leftarrow \{\}$  // Initialize the set of nodes to visit next
8   for  $i$  in  $\mathcal{U}_{\text{current}}$  do
9     for  $j$  in  $[n] \setminus \pi_i$  do
10       $\pi \leftarrow (\pi_i, j)$  // Extend the trajectory  $\pi_i$  by one mutation in  $[n] \setminus \pi_i$ 
11       $t_\pi \sim t_{\pi_i} + \text{Exp}(\lambda_{(\pi_i, j)})$ ,  $\lambda_{(\pi_i, j)} = \Theta_{jj} \prod_{l \in \pi_i} \Theta_{jl}$  // Draw a waiting time for  $\pi$ 
12      if  $t_\pi < t_s$  then
13         $k \leftarrow |V| + 1$ 
14         $V \leftarrow V \cup \{(k, \pi)\}$  // Expand the tree by one node
15         $E \leftarrow E \cup \{i \rightarrow k\}$ 
16         $\mathcal{U}_{\text{next}} \leftarrow \mathcal{U}_{\text{next}} \cup \{k\}$ 
17      end
18    end
19  end
20   $\mathcal{U}_{\text{current}} \leftarrow \mathcal{U}_{\text{next}}$ 
21 end
22  $\mathcal{T} \leftarrow (V, E)$ 

```

Algorithm 1: Tree generating process

All subclones except the root in \mathcal{T} satisfy the following partial order relation: if π is a subclone of \mathcal{T} , then its parent $\text{pa}(\pi)$ must also be a subclone of \mathcal{T} with an edge $\text{pa}(\pi) \rightarrow \pi$, since by definition π can only occur after $\text{pa}(\pi)$. We can view a tree \mathcal{T} as a set of subclones compatible with the partial order relation and omit the implicitly implied edge set. For example, the tree displayed in Supplementary Figure S1(b) consists of four subclones $\{(0), (0, 1), (0, 3), (0, 1, 3)\}$, and one can read off the edges by enumerating all parent-child pairs, *i.e.* $(0) \rightarrow (0, 1)$, $(0) \rightarrow (0, 3)$, and $(0, 1) \rightarrow (0, 1, 3)$. Removing a leaf like $(0, 1, 3)$ from the set results in another smaller tree with three vertices.

Since tumor mutation trees are effectively sets of subclones, the notations and binary operations on sets can be applied directly:

- $\pi \in \mathcal{T}$ means that π is a subclone of the tree \mathcal{T} .
- We use $|\mathcal{T}|$ to denote the number of subclones in tree \mathcal{T} .
- We say that \mathcal{T}_1 is a subtree of \mathcal{T}_2 if all subclones in \mathcal{T}_1 can be found in \mathcal{T}_2 and write $\mathcal{T}_1 \subseteq \mathcal{T}_2$. If \mathcal{T}_1 is a subtree of \mathcal{T}_2 and $|\mathcal{T}_1| < |\mathcal{T}_2|$, then we write $\mathcal{T}_1 \subset \mathcal{T}_2$.
- We use $\mathcal{T}_2 \setminus \mathcal{T}_1$ to denote the set of subclones in \mathcal{T}_2 that are not subclones of \mathcal{T}_1 . The number of such subclones are denoted by $|\mathcal{T}_2 \setminus \mathcal{T}_1|$.
- We use $\mathcal{T}_1 \cup \mathcal{T}_2$ to denote the union of subclones in \mathcal{T}_1 and \mathcal{T}_2 , which constitutes a valid tree since the partial order relation still holds. We may also use $\mathcal{T} \cup S$ to denote the union of subclones in a tree \mathcal{T} and a set of subclones S , which may not satisfy the partial order relation.
- Given a tree \mathcal{T} , we define the exit set of \mathcal{T} as

$$\text{Exit}(\mathcal{T}) = \{\pi \mid \pi \notin \mathcal{T}, \text{pa}(\pi) \in \mathcal{T}\}, \quad (1)$$

which contains the children of the existing subclones that are not yet in \mathcal{T} , *i.e.* all subclones that could appear next. Then, we define the augmented tree as

$$A(\mathcal{T}) := \mathcal{T} \cup \text{Exit}(\mathcal{T}), \quad \text{i.e., } \text{Exit}(\mathcal{T}) = A(\mathcal{T}) \setminus \mathcal{T}. \quad (2)$$

We model tumor progression as a continuous-time Markov chain $(X_t)_{t \geq 0}$ [1] on the state space of all tumor mutation trees, denoted by $S_{\mathcal{T}}^n$. The size of the state space grows super-exponentially in the number of mutations n since

$$|S_{\mathcal{T}}^1| = 2, \quad |S_{\mathcal{T}}^n| = (|S_{\mathcal{T}}^{n-1}| + 1)^n. \quad (3)$$

The initial state of the chain is the tree with only the root, $X_0 = \mathcal{T}_0$, meaning that no mutations of interest are present at the start of the evolution. It is equivalent to say that the initial distribution of the chain has a point mass of 1 on \mathcal{T}_0 and 0 on all other trees at time $t = 0$, denoted by the Kronecker delta $\delta_{\mathcal{T}_0}$. The generator Q -matrix $Q = (q_{\mathcal{T}_1, \mathcal{T}_2} : \mathcal{T}_1, \mathcal{T}_2 \in S_{\mathcal{T}}^n)$ is indexed by trees and defined as

$$q_{\mathcal{T}_1, \mathcal{T}_2} := \begin{cases} \lambda_{\mathcal{T}_2 \setminus \mathcal{T}_1} & \mathcal{T}_1 \subset \mathcal{T}_2, |\mathcal{T}_2 \setminus \mathcal{T}_1| = 1 \\ -\sum_{\mathcal{T} \neq \mathcal{T}_1} q_{\mathcal{T}_1, \mathcal{T}} & \mathcal{T}_1 = \mathcal{T}_2 \\ 0 & \text{otherwise.} \end{cases} \quad (4)$$

The dimension of Q is $|S_{\mathcal{T}}^n| \times |S_{\mathcal{T}}^n|$. We say $(X_t)_{t \geq 0}$ is continuous-time *Markov*($\delta_{\mathcal{T}_0}, Q$). The interpretation of the off-diagonal entries is that transitioning from \mathcal{T}_1 to \mathcal{T}_2 is only possible if \mathcal{T}_2 has exactly one extra subclone than \mathcal{T}_1 , and this new subclone $\pi = \mathcal{T}_2 \setminus \mathcal{T}_1$ must be a child of one of the existing subclones in \mathcal{T}_1 , *i.e.* $\pi \in \text{Exit}(\mathcal{T}_1)$. The transition is achieved by accumulating one subclone at a time, and this process is irreversible. The transition rate is associated with all mutations in $\pi = (\text{pa}(\pi), i)$ for some $i \in [n] \setminus \text{pa}(\pi)$ and parameterized by a Mutual Hazard Network $\Theta \in \mathbb{R}^{n \times n}$,

$$\lambda_{\mathcal{T}_2 \setminus \mathcal{T}_1} = \lambda_{\pi} := \exp\left(\theta_{ii} + \sum_{j \in \text{pa}(\pi)} \theta_{ij}\right) = \Theta_{ii} \prod_{j \in \text{pa}(\pi)} \Theta_{ij}, \quad (5)$$

which is equivalent to the definition in Equation (1) of the main text (notation-wise shifted by one mutation). The diagonal entries are chosen such that the row sums of Q are zero. We interpret the off-diagonal row sum

$$q_{\mathcal{T}_1} := \sum_{\mathcal{T} \neq \mathcal{T}_1} q_{\mathcal{T}_1, \mathcal{T}} = \sum_{\pi \in \text{Exit}(\mathcal{T}_1)} \lambda_\pi, \quad (6)$$

as the rate of leaving the state \mathcal{T}_1 , which is the sum of the rates of all events that could happen next.

Following Chapter 2 of [1], we define the jump times J_0, J_1, \dots and the holding times S_1, S_2, \dots of $(X_t)_{t \geq 0}$ to be

$$J_0 = 0, \quad J_{k+1} = \inf\{t \geq J_k : X_t \neq X_{J_k}\} \quad (7)$$

for $k = 0, 1, \dots$, with $\inf \emptyset = \infty$, and for $k = 1, 2, \dots$,

$$S_k = \begin{cases} J_k - J_{k-1} & J_{k-1} < \infty \\ \infty & \text{otherwise.} \end{cases} \quad (8)$$

Furthermore, we define the jump matrix $\Xi = (\xi_{\mathcal{T}_1, \mathcal{T}_2} : \mathcal{T}_1, \mathcal{T}_2 \in S_{\mathcal{T}}^n)$ of Q as

$$\begin{aligned} \xi_{\mathcal{T}_1, \mathcal{T}_2} &:= \begin{cases} q_{\mathcal{T}_1, \mathcal{T}_2} / q_{\mathcal{T}_1} & \mathcal{T}_1 \neq \mathcal{T}_2 \text{ and } q_{\mathcal{T}_1} \neq 0 \\ 0 & \mathcal{T}_1 \neq \mathcal{T}_2 \text{ and } q_{\mathcal{T}_1} = 0, \end{cases} \\ \xi_{\mathcal{T}_1, \mathcal{T}_1} &:= \begin{cases} 0 & q_{\mathcal{T}_1} \neq 0 \\ 1 & q_{\mathcal{T}_1} = 0. \end{cases} \end{aligned} \quad (9)$$

By Theorem 2.8.2 of [1], the jump chain $(Y_k)_{k \geq 0}$ associated with $(X_t)_{t \geq 0}$ is a discrete-time Markov chain with initial distribution $\delta_{\mathcal{T}_0}$ and jump matrix Ξ , where the entry $\xi_{\mathcal{T}_1, \mathcal{T}_2}$ is the probability of jumping from \mathcal{T}_1 to \mathcal{T}_2 . We say $(Y_k)_{k \geq 0}$ is discrete-time *Markov*($\delta_{\mathcal{T}_0}, \Xi$). In particular, conditioned on the chain of states Y_0, Y_1, \dots, Y_{k-1} for each $k \geq 1$, the holding times S_1, S_2, \dots, S_k are independent exponential random variables with parameters $q_{Y_0}, q_{Y_1}, \dots, q_{Y_{k-1}}$.

Next we show the equivalence between the formulation in terms of Markov chains above and the tree-generating process defined in the main text.

Theorem 1. *A right-continuous process $(X_t)_{t \geq 0}$ with values in the space of tumor mutation trees $S_{\mathcal{T}}^n$ is continuous-time Markov($\delta_{\mathcal{T}_0}, Q$) if and only if the waiting times of subclones $\pi \in \mathcal{T}$ for any $\mathcal{T} \in S_{\mathcal{T}}^n$ are exponential random variables such that*

$$T_{(0)} = 0, \quad T_\pi \sim T_{\text{pa}(\pi)} + \text{Exp}(\lambda_\pi). \quad (10)$$

Proof. (\implies) Suppose $(X_t)_{t \geq 0}$ is continuous-time *Markov*($\delta_{\mathcal{T}_0}, Q$). The initial state $X_0 = \mathcal{T}_0$ corresponds to the initial wild-type subclone (0) at time 0, *i.e.* $T_{(0)} = 0$. By the infinitesimal definition in Theorem 2.8.2 of [1], for all $t, h \geq 0$, conditioned on $X_t = \mathcal{T}_1$ for some $\mathcal{T}_1 \in S_{\mathcal{T}}^n$, X_{t+h} is independent of $(X_s : s \leq t)$. As $h \downarrow 0$, uniformly in t , for all $\mathcal{T}_2 \in S_{\mathcal{T}}^n$,

$$P(X_{t+h} = \mathcal{T}_2 \mid X_t = \mathcal{T}_1) = \delta_{\mathcal{T}_1, \mathcal{T}_2} + q_{\mathcal{T}_1, \mathcal{T}_2} h + o(h), \quad (11)$$

where the Kronecker delta $\delta_{\mathcal{T}_1, \mathcal{T}_2} = 1$ if $\mathcal{T}_1 = \mathcal{T}_2$ and 0 otherwise. For any subclone $\pi \neq (0)$, we can construct a smaller chain $(Z_t^\pi)_{t \geq 0}$ on two states $\{\text{pa}(\pi), \pi\}$ with initial state $Z_0^\pi = \text{pa}(\pi)$ from $(X_t)_{t \geq 0}$, in order to obtain the marginal probability distribution of π . For all $t \geq 0$, as $h \rightarrow 0$,

$$\begin{aligned} P(Z_{t+h}^\pi = \pi, Z_t^\pi = \text{pa}(\pi)) &= \sum_{\mathcal{T} : \text{pa}(\pi) \in \mathcal{T}, \pi \notin \mathcal{T}} P(X_{t+h} = \mathcal{T} \cup \{\pi\}, X_t = \mathcal{T}) \\ &= \sum_{\mathcal{T} : \text{pa}(\pi) \in \mathcal{T}, \pi \notin \mathcal{T}} P(X_{t+h} = \mathcal{T} \cup \{\pi\} \mid X_t = \mathcal{T}) P(X_t = \mathcal{T}) \\ &= \sum_{\mathcal{T} : \text{pa}(\pi) \in \mathcal{T}, \pi \notin \mathcal{T}} (\lambda_\pi + o(h)) P(X_t = \mathcal{T}) \\ &= (\lambda_\pi + o(h)) \sum_{\mathcal{T} : \text{pa}(\pi) \in \mathcal{T}, \pi \notin \mathcal{T}} P(X_t = \mathcal{T}) \\ &= (\lambda_\pi + o(h)) P(Z_t^\pi = \text{pa}(\pi)). \end{aligned} \quad (12)$$

Hence,

$$\begin{aligned} P(Z_{t+h}^\pi = \pi \mid Z_t^\pi = \text{pa}(\pi)) &= \frac{P(Z_{t+h}^\pi = \pi, Z_t^\pi = \text{pa}(\pi))}{P(Z_t^\pi = \text{pa}(\pi))} \\ &= \frac{(\lambda_\pi + o(h))P(Z_t^\pi = \text{pa}(\pi))}{P(Z_t^\pi = \text{pa}(\pi))} \\ &= \lambda_\pi + o(h), \end{aligned} \quad (13)$$

$$P(Z_{t+h}^\pi = \text{pa}(\pi) \mid Z_t^\pi = \text{pa}(\pi)) = 1 - \lambda_\pi + o(h). \quad (14)$$

Similarly,

$$P(Z_{t+h}^\pi = \text{pa}(\pi) \mid Z_t^\pi = \pi) = o(h), \quad \text{and} \quad P(Z_{t+h}^\pi = \pi \mid Z_t^\pi = \pi) = 1 + o(h). \quad (15)$$

By Theorem 2.8.2 of [1], $(Z_t^\pi)_{t \geq 0}$ is also a continuous-time Markov chain with the generator matrix

$$\begin{array}{c} \text{pa}(\pi) \\ \pi \end{array} \begin{bmatrix} \text{pa}(\pi) & \pi \\ -\lambda_\pi & \lambda_\pi \\ 0 & 0 \end{bmatrix}. \quad (16)$$

It follows that the holding time from $\text{pa}(\pi)$ to π is an exponential random variable with rate λ_π , as required.

(\Leftarrow) Let $(X_t)_{t \geq 0}$ be a right-continuous process with values in $S_{\mathcal{T}}^n$, which tracks the change of state resulting from the occurrence of new subclones with exponentially-distributed waiting times as defined in Supplementary Eq. (10). Let $(Y_k)_{k \geq 0}$ be the associated jump chain. We will use the jump chain/holding time definition in Theorem 2.8.2 of [1] to prove that $(X_t)_{t \geq 0}$ is continuous-time *Markov*($\delta_{\mathcal{T}_0}, Q$).

At time $t = 0$, there is only the initial wild-type subclone $\pi = (0)$, which corresponds to the initial state $X_0 = Y_0 = \mathcal{T}_0$. Let $W(k)$ be the statement that for each $k \geq 1$, conditioned on the chain of states Y_0, Y_1, \dots, Y_{k-1} , the holding times S_1, S_2, \dots, S_k are independent exponential random variables with parameters $q_{Y_0}, q_{Y_1}, \dots, q_{Y_{k-1}}$. We will prove $W(k)$ by induction on $k \geq 1$.

- Base case: Conditioned on $Y_0 = \mathcal{T}_0$, we have

$$S_1 = J_1 = \inf\{t \geq 0 : X_t \neq \mathcal{T}_0\} = \inf\{T_{(0,i)} \geq 0 : i \in [n]\}, \quad (17)$$

because the set of mutations that could happen next is $[n]$, and whichever mutation appears first will change the state of the chain. Each waiting time $T_{(0,i)}$ is an independent exponential random variable with rate $\lambda_{(0,i)}$. By Theorem 2.3.3 of [1] (competing exponentials), the first holding time S_1 is an exponential random variable with rate

$$\sum_{i \in [n]} \lambda_{(0,i)} = \sum_{\pi \in \text{Exit}(\mathcal{T}_0)} \lambda_\pi = q_{\mathcal{T}_0} = q_{Y_0}, \quad (18)$$

independent of the next state Y_1 , which has distribution $(\xi_{\mathcal{T}_0, \mathcal{T}_1} : \mathcal{T}_1 \in S_{\mathcal{T}}^n)$ because

$$P(Y_1 = \mathcal{T}_1 \mid Y_0 = \mathcal{T}_0) = \begin{cases} \frac{\lambda_{(0,i)}}{\sum_{i \in [n]} \lambda_{(0,i)}} = \frac{q_{\mathcal{T}_0, \mathcal{T}_1}}{q_{\mathcal{T}_0}} & \mathcal{T}_1 = \mathcal{T}_0 \cup \{(0, i)\} \text{ for } i \in [n] \\ 0 & \text{otherwise.} \end{cases} \quad (19)$$

Hence, the statement $W(1)$ is true.

- Induction step: Assume that for a given value $k = l \geq 1$, $W(k)$ holds. Next, conditioned on $Y_l = \mathcal{T}_l \in S_{\mathcal{T}}^n$, the next holding time depends on the waiting times of the subclones that can appear next,

$$S_{l+1} = J_{l+1} - J_l = \inf\{t \geq J_l : X_t \neq X_{J_l}\} - J_l = \inf\{T_\pi - J_l \geq 0 : \pi \in \text{Exit}(\mathcal{T}_l)\}. \quad (20)$$

We first assume that $\text{Exit}(\mathcal{T}_l) \neq \emptyset$. For each $\pi \in \text{Exit}(\mathcal{T}_l)$, by definition, we have $\text{pa}(\pi) \in \mathcal{T}_l$ and $T_{\text{pa}(\pi)} \leq J_l \Leftrightarrow J_l - T_{\text{pa}(\pi)} \geq 0$. By Supplementary Eq. (10), the waiting time from $\text{pa}(\pi)$ to π is an exponentially-distributed random variable,

$$T_\pi - T_{\text{pa}(\pi)} \sim \text{Exp}(\lambda_\pi), \quad (21)$$

then by Theorem 2.3.1 of [1] (memoryless property), the waiting time from the most recent jump to the occurrence of π remains exponentially distributed with the same rate,

$$T_\pi - J_l = (T_\pi - T_{\text{pa}(\pi)}) - (J_l - T_{\text{pa}(\pi)}) \sim \text{Exp}(\lambda_\pi), \quad (22)$$

and independent of $T_{\pi'} - J_l$ for $\pi' \neq \pi \in \mathcal{T}_l$. By Theorem 2.3.3 of [1], S_{l+1} is an exponential random variable with rate

$$\sum_{\pi \in \text{Exit}(Y_l)} \lambda_\pi = q_{\mathcal{T}_l} = q_{Y_l}, \quad (23)$$

independent of S_1, S_2, \dots, S_l as required. S_{l+1} is also independent of the next state Y_{l+1} , which has distribution $(\xi_{\mathcal{T}_l, \mathcal{T}_{l+1}} : \mathcal{T}_{l+1} \in S_{\mathcal{T}}^n)$ because

$$P(Y_{l+1} = \mathcal{T}_{l+1} \mid Y_l = \mathcal{T}_l) = \begin{cases} \frac{\lambda_\pi}{\sum_{\pi' \in \text{Exit}(Y_l)} \lambda_{\pi'}} = \frac{q_{\mathcal{T}_l, \mathcal{T}_{l+1}}}{q_{\mathcal{T}_l}} & \mathcal{T}_{l+1} = \mathcal{T}_l \cup \{\pi\} \text{ for } \pi \in \text{Exit}(Y_l) \\ 0 & \text{otherwise.} \end{cases} \quad (24)$$

If $\text{Exit}(\mathcal{T}_l) = \emptyset$, *i.e.* no subclones can appear next, then by the definition in [1], $S_{l+1} = \inf \emptyset = \infty$ is exponentially distributed with rate $q_{\mathcal{T}_l} = 0$, and $Y_{l+1} = Y_l = \mathcal{T}_l$ is an absorbing state with $P(Y_{l+1} = \mathcal{T}_l \mid Y_l = \mathcal{T}_l) = 1$.

Therefore, the statement $W(l+1)$ is also true. By induction, $W(k)$ holds for all $k \geq 1$. Since $Y_0 = \mathcal{T}_0$ and the transition probabilities $P(Y_k = \mathcal{T}_k \mid Y_{k-1} = \mathcal{T}_{k-1})$ for $k \geq 1$ match the definition of Ξ in Supplementary Eq. (9), the jump chain $(Y_k)_{k \geq 0}$ is discrete-time *Markov* $(\delta_{\mathcal{T}_0}, \Xi)$. By the jump chain/holding time definition in Theorem 2.8.2 of [1], $(X_t)_{t \geq 0}$ is continuous-time *Markov* $(\delta_{\mathcal{T}_0}, Q)$.

We have proved that $(X_t)_{t \geq 0}$ is continuous-time *Markov* $(\delta_{\mathcal{T}_0}, Q)$ if and only if the subclones $\pi \in \mathcal{T}$ for any $\mathcal{T} \in S_{\mathcal{T}}^n$ have exponentially-distributed waiting times as defined in Supplementary Eq. (10). \square

Theorem 2. *Let $\Theta \in \mathbb{R}^{n \times n}$ be a Mutual Hazard Network, where $n \in \mathbb{N}^+$ is the number of mutations. Let $(X_t)_{t \geq 0}$ be continuous-time *Markov* $(\delta_{\mathcal{T}_0}, Q)$. Suppose $\mathcal{T} \in S_{\mathcal{T}}^n$ is an output of the tree generating process. Then the marginal probability of observing \mathcal{T} is given by*

$$p(\mathcal{T} \mid \Theta) := P \left(\max_{\pi \in \mathcal{T}} T_\pi < T_s < \min_{\pi' \in \text{Exit}(\mathcal{T})} T_{\pi'} \mid \Theta \right) = (\lambda_s (\lambda_s I - Q)^{-1})_{\mathcal{T}_0, \mathcal{T}}. \quad (25)$$

Proof. In the tree generating process, the waiting times of the subclones are exponential random variables as defined in Supplementary Eq. (10). By Theorem 1, the right-continuous process $(X_t)_{t \geq 0}$ that tracks the change of state resulting from the occurrence of new subclones is continuous-time *Markov* $(\delta_{\mathcal{T}_0}, Q)$. That is, for all $t > 0$, and $\mathcal{T} \in S_{\mathcal{T}}^n$,

$$X_t = \mathcal{T} \Leftrightarrow \max_{\pi \in \mathcal{T}} T_\pi < t < \min_{\pi' \in \text{Exit}(\mathcal{T})} T_{\pi'}, \quad (26)$$

because if there exists $\pi \in \mathcal{T}$ such that $T_\pi > t$, or $\pi' \in \text{Exit}(\mathcal{T})$ such that $T_{\pi'} < t$, then $X_t \neq \mathcal{T}$, which is a contradiction. Next, the matrix $P(t) = e^{tQ}$ is the transition probability matrix of $(X_t)_{t \geq 0}$, which by Theorem 2.1.1 of [1] solves the backward equation,

$$\frac{d}{dt} P(t) = QP(t), \quad P(0) = I. \quad (27)$$

The transition probability from \mathcal{T}_1 to \mathcal{T}_2 in time t is given by

$$P(X_t = \mathcal{T}_2 \mid X_0 = \mathcal{T}_1) = p_{\mathcal{T}_1, \mathcal{T}_2}(t), \quad (28)$$

where $p_{\mathcal{T}_1, \mathcal{T}_2}(t)$ is the $(\mathcal{T}_1, \mathcal{T}_2)$ entry in e^{tQ} , the row is indexed by \mathcal{T}_1 , and the column is indexed by \mathcal{T}_2 . In particular, starting with $X_0 = \mathcal{T}_0$, the probability of observing \mathcal{T} at sampling time t_s is

$$P(X_{t_s} = \mathcal{T} \mid X_0 = \mathcal{T}_0) = p_{\mathcal{T}_0, \mathcal{T}}(t_s) = (e^{t_s Q})_{\mathcal{T}_0, \mathcal{T}}, \quad (29)$$

which is the $(\mathcal{T}_0, \mathcal{T})$ entry in $e^{t_s Q}$. It follows that

$$P\left(\max_{\pi \in \mathcal{T}} T_\pi < t_s < \min_{\pi' \in \text{Exit}(\mathcal{T})} T_{\pi'} \mid \Theta\right) = P(X_{t_s} = \mathcal{T} \mid X_0 = \mathcal{T}_0) = (e^{t_s Q})_{\mathcal{T}_0, \mathcal{T}}. \quad (30)$$

In the tree generating process, we assume that the unknown sampling time follows an independent exponential distribution $T_s \sim \text{Exp}(\lambda_s)$ with density function $f_{\lambda_s}(t_s) = \lambda_s \exp(-\lambda_s t_s)$. Integrating it out gives us

$$\begin{aligned} P\left(\max_{\pi \in \mathcal{T}} T_\pi < T_s < \min_{\pi' \in \text{Exit}(\mathcal{T})} T_{\pi'} \mid \Theta\right) &= \int_0^\infty f_{\lambda_s}(t_s) P\left(\max_{\pi \in \mathcal{T}} T_\pi < t_s < \min_{\pi' \in \text{Exit}(\mathcal{T})} T_{\pi'} \mid \Theta\right) dt_s \\ &= \left(\int_0^\infty f_{\lambda_s}(t_s) e^{t_s Q} dt_s\right)_{\mathcal{T}_0, \mathcal{T}} \\ &= \left(\int_0^\infty \lambda_s e^{-\lambda_s t_s} e^{t_s Q} dt_s\right)_{\mathcal{T}_0, \mathcal{T}} \\ &= \left(\lambda_s \int_0^\infty e^{-\lambda_s t_s} e^{t_s Q} dt_s\right)_{\mathcal{T}_0, \mathcal{T}}. \end{aligned} \quad (31)$$

Here, $\mathcal{L}(P(t)) = \int_0^\infty e^{-\lambda_s t_s} P(t_s) dt_s = \int_0^\infty e^{-\lambda_s t_s} e^{t_s Q} dt_s = A(\lambda_s)$ is the Laplace transform of $P(t)$ [2]. By the Laplace transform of derivatives [2], we have

$$\mathcal{L}\left(\frac{d}{dt} P(t)\right) = \lambda_s A(\lambda_s) - P(0) = \lambda_s A(\lambda_s) - I. \quad (32)$$

Applying Laplace transforms to both sides of Supplementary Eq. (27) gives

$$\lambda_s A(\lambda_s) - I = Q A(\lambda_s) \Leftrightarrow (\lambda_s I - Q) A(\lambda_s) = I. \quad (33)$$

By the definition in Supplementary Eq. (4), we can permute the rows and columns of the generator matrix Q such that Q is upper-triangular, *e.g.* by sorting the trees by increasing tree size, such that the matrix $V := \lambda_s I - Q$ is also upper-triangular. Since all transition rates are positive by Supplementary Eq. (5), the diagonal entries of Q are non-positive. Hence, the diagonal entries of V are positive. It follows that the determinant of V , as the product of the diagonal entries, is non-zero, *i.e.* $\det(V) \neq 0$. Hence, V is non-singular, V^{-1} exists, and

$$A(\lambda_s) = (\lambda_s I - Q)^{-1}. \quad (34)$$

Therefore, the marginal probability of observing \mathcal{T} is given by

$$p(\mathcal{T} \mid \Theta) := P\left(\max_{\pi \in \mathcal{T}} T_\pi < T_s < \min_{\pi' \in \text{Exit}(\mathcal{T})} T_{\pi'} \mid \Theta\right) = (\lambda_s A(\lambda_s))_{\mathcal{T}_0, \mathcal{T}} = (\lambda_s (\lambda_s I - Q)^{-1})_{\mathcal{T}_0, \mathcal{T}}. \quad (35)$$

□

A.3 Maximum likelihood estimation of an MHN given a set of trees

In the previous section, we have shown in Theorem 2 that computing $p(\mathcal{T} \mid \Theta)$ involves inverting the matrix $V = \lambda_s I - Q$ of dimension $|S_{\mathcal{T}}^n| \times |S_{\mathcal{T}}^n|$, where $|S_{\mathcal{T}}^n|$ grows super-exponentially in n (Supplementary Eq. (3)). Hence, the matrix inversion is intractable for even a moderate size n . In this section, we seek to simplify the calculation of Supplementary Eq. (25).

We sort the rows and columns of Q by increasing tree size $(\mathcal{T}_0, \mathcal{T}_1, \dots, \mathcal{T}_L)$ such that $|\mathcal{T}_0| < |\mathcal{T}_1| < \dots < |\mathcal{T}_L|$, where $L := |S_{\mathcal{T}}^n|$ is the total number of trees in the state space. In other words, Q is upper-triangular, and so is V . Let $\mathbf{e}_{\mathcal{T}}$ be a unit vector on $S_{\mathcal{T}}^n$ with 1 at the position of $\mathcal{T} \in S_{\mathcal{T}}^n$ and 0 at other positions. The dimension of $\mathbf{e}_{\mathcal{T}}$ is L . For example, $\mathbf{e}_{\mathcal{T}_0} = (1, 0, \dots, 0)^\top$. Then, we can rewrite Supplementary Eq. (25) as

$$p(\mathcal{T} \mid \Theta) = (\lambda_s V^{-1})_{\mathcal{T}_0, \mathcal{T}} = \lambda_s \mathbf{e}_{\mathcal{T}_0}^\top V^{-1} \mathbf{e}_{\mathcal{T}}. \quad (36)$$

numbers of the equations in this paper. The term $\partial p(\mathcal{T} | \Theta) / \partial \lambda_{(\pi,i)}$ can be obtained from the first row and the last column of the matrix

$$\lambda_s(\lambda_s I - R_{\mathcal{T}}^{(\pi,i)})^{-1}, \quad R_{\mathcal{T}}^{(\pi,i)} := \begin{bmatrix} Q_{\mathcal{T}} & \frac{\partial Q_{\mathcal{T}}}{\partial \lambda_{(\pi,i)}} \\ 0 & Q_{\mathcal{T}} \end{bmatrix}, \quad (41)$$

with forward substitution, because

$$(\lambda_s I - R_{\mathcal{T}}^{(\pi,i)})^{-1} = \begin{bmatrix} \lambda_s I - Q_{\mathcal{T}} & -\frac{\partial Q_{\mathcal{T}}}{\partial \lambda_{(\pi,i)}} \\ 0 & \lambda_s I - Q_{\mathcal{T}} \end{bmatrix}^{-1} = \begin{bmatrix} V_{\mathcal{T}} & \frac{\partial V_{\mathcal{T}}}{\partial \lambda_{(\pi,i)}} \\ 0 & V_{\mathcal{T}} \end{bmatrix}^{-1} = \begin{bmatrix} V_{\mathcal{T}}^{-1} & -V_{\mathcal{T}}^{-1} \frac{\partial V_{\mathcal{T}}}{\partial \lambda_{(\pi,i)}} V_{\mathcal{T}}^{-1} \\ 0 & V_{\mathcal{T}}^{-1} \end{bmatrix} \quad (42)$$

by (PP399) and (PP400) of the Matrix Cookbook. Also, the gradient of the penalty term $\gamma \sum_{i \neq j} |\log \Theta_{ij}|$ with respect to each Θ_{ij} is

$$\frac{\gamma \text{sign}(\theta_{ij})}{\Theta_{ij}}, \quad \text{sign}(\theta_{ij}) = \begin{cases} 1 & \theta_{ij} > 0 \\ 0 & \theta_{ij} = 0 \\ -1 & \theta_{ij} < 0 \end{cases}. \quad (43)$$

Therefore, given N mutation trees $\mathcal{T} = \{\mathcal{T}^1, \dots, \mathcal{T}^N\}$, we can find the maximum likelihood estimation of Θ by gradient ascent optimization.

A.4 Hybrid EM and Monte Carlo EM algorithm for the estimation of an MHN given a set of trees

The joint density of a collection of waiting times \mathbf{t} , sampling time t_s , and the corresponding tree structure \mathcal{T} is

$$\begin{aligned} p(\mathcal{T}, \mathbf{t}, t_s | \Theta) &= f_{\lambda_s}(t_s) \prod_{i=1}^n \prod_{\pi \in \mathcal{T}: (\pi,i) \in A(\mathcal{T})} f_{\lambda_{(\pi,i)}}(t_{(\pi,i)} - t_{\pi}) \times \prod_{\pi \in \mathcal{T}} \mathbb{1}(t_{\pi} < t_s) \times \prod_{\pi \in A(\mathcal{T}): \pi \notin \mathcal{T}} \mathbb{1}(t_{\pi} > t_s) \\ &= f_{\lambda_s}(t_s) \prod_{i=1}^n \prod_{\pi \in \mathcal{T}: (\pi,i) \in A(\mathcal{T})} \left[\lambda_{(\pi,i)} \exp(-\lambda_{(\pi,i)}(t_{(\pi,i)} - t_{\pi})) \right] \\ &\quad \times \prod_{\pi \in \mathcal{T}} \mathbb{1}(t_{\pi} < t_s) \times \prod_{\pi \in A(\mathcal{T}): \pi \notin \mathcal{T}} \mathbb{1}(t_{\pi} > t_s) \\ &= f_{\lambda_s}(t_s) \prod_{i=1}^n \prod_{\pi \in \mathcal{T}: (\pi,i) \in A(\mathcal{T})} \left[\Theta_{ii} \prod_{j \in \pi} \Theta_{ij} \exp(-\Theta_{ii} \prod_{j \in \pi} \Theta_{ij} (t_{(\pi,i)} - t_{\pi})) \right] \\ &\quad \times \prod_{\pi \in \mathcal{T}} \mathbb{1}(t_{\pi} < t_s) \times \prod_{\pi \in A(\mathcal{T}): \pi \notin \mathcal{T}} \mathbb{1}(t_{\pi} > t_s) \end{aligned} \quad (44)$$

where f_{λ} is the exponential density function with parameter λ and $\lambda_s = 1$. The complete-data log-likelihood is then

$$\ell^{\text{full}}(\Theta) = \sum_{l=1}^N \sum_{i=1}^n \sum_{\pi \in \mathcal{T}^l: (\pi,i) \in A(\mathcal{T}^l)} \left[\log \Theta_{ii} + \sum_{j \in \pi} \log \Theta_{ij} - \Theta_{ii} \prod_{j \in \pi} \Theta_{ij} (t_{(\pi,i)}^l - t_{\pi}^l) \right] + \log f_{\lambda_s}(t_s^l) \quad (45)$$

and $\log f_{\lambda_s}(t_s^l)$ is a constant with respect to Θ . The expected complete-data log-likelihood in the E step of the EM algorithm follows by replacing each $(t_{(\pi,i)} - t_{\pi})$ with $\mathbb{E}_{\mathbf{T}, T_s | \mathcal{T}, \Theta} [T_{(\pi,i)} - T_{\pi}]$.

To compute the expected time differences, we first use the definition of the expected value,

$$\begin{aligned} \mathbb{E}_{\mathbf{T}, T_s | \mathcal{T}, \Theta} [T_{(\pi,i)} - T_{\pi}] &= \int_{\mathbb{R}_{\geq 0}^{|A(\mathcal{T})|+1}} (t_{(\pi,i)} - t_{\pi}) p(\mathbf{t}, t_s | \mathcal{T}, \Theta) dt dt_s \\ &= \frac{1}{P(\mathcal{T} | \Theta)} \int_{\mathbb{R}_{\geq 0}^{|A(\mathcal{T})|+1}} (t_{(\pi,i)} - t_{\pi}) p(\mathcal{T}, \mathbf{t}, t_s | \Theta) dt dt_s. \end{aligned} \quad (46)$$

Notice that

$$\begin{aligned}
\frac{\partial p(\mathcal{T} | \Theta)}{\partial \lambda_{(\pi,i)}} &= \frac{\partial}{\partial \lambda_{(\pi,i)}} \int_{\mathbb{R}_{\geq 0}^{|A(\mathcal{T})|+1}} p(\mathcal{T}, \mathbf{t}, t_s | \Theta) dt dt_s \\
&= \int_{\mathbb{R}_{\geq 0}^{|A(\mathcal{T})|+1}} \frac{\partial}{\partial \lambda_{(\pi,i)}} p(\mathcal{T}, \mathbf{t}, t_s | \Theta) dt dt_s \\
&= \int_{\mathbb{R}_{\geq 0}^{|A(\mathcal{T})|+1}} \left(\frac{1}{\lambda_{(\pi,i)}} - (t_{(\pi,i)} - t_\pi) \right) p(\mathcal{T}, \mathbf{t}, t_s | \Theta) dt dt_s \\
&= p(\mathcal{T} | \Theta) \left(\frac{1}{\lambda_{(\pi,i)}} - \mathbb{E}_{\mathbf{T}, T_s | \mathcal{T}, \Theta} [T_{(\pi,i)} - T_\pi] \right).
\end{aligned} \tag{47}$$

Therefore,

$$\mathbb{E}_{\mathbf{T}, T_s | \mathcal{T}, \Theta} [T_{(\pi,i)} - T_\pi] = \frac{1}{\lambda_{(\pi,i)}} - \frac{1}{P(\mathcal{T} | \Theta)} \frac{\partial P(\mathcal{T} | \Theta)}{\partial \lambda_{(\pi,i)}}, \tag{48}$$

where both $P(\mathcal{T} | \Theta)$ and $\frac{\partial P(\mathcal{T} | \Theta)}{\partial \lambda_{(\pi,i)}}$ can be computed as in Section A.3. For large trees (*e.g.* with more than 500 subtrees), however, computing this exact form can be very slow. In this case, we can approximate it by drawing M samples from the proposal distribution with density g as defined in Methods. To recapitulate, we first draw the sampling time T_s from $\text{Exp}(\lambda_s)$ with $\lambda_s = 1$, followed by drawing the difference in waiting times between subclones π and (π, i) recursively as

$$Z_{(\pi,i)} \sim \begin{cases} \text{TExp}(\lambda_{(\pi,i)}, 0, t_s - t_\pi) & \text{if } (\pi, i) \in \mathcal{T} \\ \text{TExp}(\lambda_{(\pi,i)}, t_s - t_\pi, \infty) & \text{if } (\pi, i) \in A(\mathcal{T}) \setminus \mathcal{T} \end{cases} \tag{49}$$

where $\text{TExp}(\lambda, a, b)$ is the truncated exponential distribution with parameter λ and bounds $0 \leq a < b < \infty$. The importance sampling weights are defined as

$$w^{(m)} := w(\mathbf{t}^{(m)} | \mathcal{T}, \Theta) = \frac{p(\mathcal{T}, \mathbf{t}^{(m)}, t_s^{(m)} | \Theta)}{g(\mathbf{t}^{(m)}, t_s^{(m)} | \mathcal{T}, \Theta)}, \quad m = 1, \dots, M. \tag{50}$$

The densities of the truncated exponential distributions $\text{TExp}(\lambda, 0, a)$ and $\text{TExp}(\lambda, b, \infty)$ are

$$f_{\lambda, 0, a}^{\text{TExp}}(t) = \frac{\lambda e^{-\lambda t} \mathbb{1}_{[0, a]}(t)}{1 - e^{-\lambda a}} \quad \text{and} \quad f_{\lambda, b, \infty}^{\text{TExp}}(t) = \frac{\lambda e^{-\lambda t} \mathbb{1}_{[b, \infty)}(t)}{e^{-\lambda b}} \quad \text{respectively.} \tag{51}$$

Since all terms in the numerator of $w^{(m)}$ cancel out, it follows that

$$w^{(m)} = \prod_{i=1}^n \prod_{(\pi,i) \in \mathcal{T}} (1 - e^{-\lambda_{(\pi,i)}(t_s^{(m)} - t_\pi^{(m)})}) \prod_{(\pi,i) \in A(\mathcal{T}), (\pi,i) \notin \mathcal{T}} e^{-\lambda_{(\pi,i)}(t_s^{(m)} - t_\pi^{(m)})}. \tag{52}$$

The approximation becomes

$$\mathbb{E}_{\mathbf{T}, T_s | \mathcal{T}, \Theta} [T_{(\pi,i)} - T_\pi] \approx \frac{\frac{1}{M} \sum_{m=1}^M w^{(m)} (t_{(\pi,i)}^{(m)} - t_\pi^{(m)})}{\frac{1}{M} \sum_{m=1}^M w^{(m)}}. \tag{53}$$

In the M step, we update Θ by maximizing the penalized expected complete-data log-likelihood, where the gradients with respect to each Θ_{ij} are computed as follows:

$$\begin{cases} \frac{\partial Q(\Theta, \Theta^{(k)})}{\partial \Theta_{ii}} = \sum_{l=1}^N \sum_{\pi \in \mathcal{T}^l: (\pi,i) \in A(\mathcal{T}^l)} \left[\frac{1}{\Theta_{ii}} - \prod_{j \in \pi} \Theta_{ij} \mathbb{E}_{\mathbf{T}^l, T_s^l | \mathcal{T}^l, \Theta^{(k)}} [T_{(\pi,i)}^l - T_\pi^l] \right] & i \in [n] \\ \frac{\partial Q(\Theta, \Theta^{(k)})}{\partial \Theta_{ij}} = \sum_{l=1}^N \sum_{\pi \in \mathcal{T}^l: (\pi,i) \in A(\mathcal{T}^l), j \in \pi} \left[\frac{1}{\Theta_{ij}} - \Theta_{ii} \prod_{k \in \pi: k \neq j} \Theta_{ik} \mathbb{E}_{\mathbf{T}^l, T_s^l | \mathcal{T}^l, \Theta^{(k)}} [T_{(\pi,i)}^l - T_\pi^l] \right] & i, j \in [n], i \neq j \end{cases} \tag{54}$$

A.5 Parameter estimation based on non-empty trees

In practice, we typically cannot observe an empty mutation tree containing only the root node for a cancer patient and all observed trees in \mathcal{T} have at least one mutation. In this case, we need to condition the likelihood on the trees being non-empty, which amounts to adding the following term to the objective functions in both the MLE and the EM parameter estimation procedure:

$$-N \log(1 - p(\mathcal{T}_0 | \Theta)) = -N \left(1 - \frac{\lambda_s}{\lambda_s + \sum_{i=1}^n \Theta_{ii}} \right). \quad (55)$$

To see this, consider the marginal probability of a tree \mathcal{T} given Θ . It can be decomposed into two parts,

$$\begin{aligned} p(\mathcal{T} | \Theta) &= p(\mathcal{T}, \mathcal{T} \text{ is empty} | \Theta) + p(\mathcal{T}, \mathcal{T} \text{ is non-empty} | \Theta) \\ &= p(\mathcal{T} | \mathcal{T} \text{ is empty}, \Theta) p(\mathcal{T} \text{ is empty} | \Theta) \mathbb{1}(\mathcal{T} \text{ is empty}) \\ &\quad + p(\mathcal{T} | \mathcal{T} \text{ is non-empty}, \Theta) p(\mathcal{T} \text{ is non-empty} | \Theta) \mathbb{1}(\mathcal{T} \text{ is non-empty}). \end{aligned} \quad (56)$$

Since \mathcal{T} can be either empty or non-empty, only one term in the summation is left. It follows that

$$p(\mathcal{T} | \mathcal{T} \text{ is non-empty}, \Theta) = \frac{p(\mathcal{T} | \Theta)}{p(\mathcal{T} \text{ is non-empty} | \Theta)} = \frac{p(\mathcal{T} | \Theta)}{1 - p(\mathcal{T} \text{ is empty} | \Theta)}. \quad (57)$$

Now, \mathcal{T} is empty if and only if $T_s = \min\{T_s, T_{(0,1)}, \dots, T_{(0,n)}\}$, which is the minimum of a set of independent exponential random variables. Therefore,

$$p(\mathcal{T} \text{ is empty} | \Theta) = \frac{\lambda_s}{\lambda_s + \lambda_{(0,1)} + \dots + \lambda_{(0,n)}} = \frac{\lambda_s}{\lambda_s + \sum_{i=1}^n \Theta_{ii}}, \quad (58)$$

and

$$\frac{\partial}{\partial \Theta_{ii}} \log(1 - p(\mathcal{T} \text{ is empty} | \Theta)) = \frac{p(\mathcal{T} \text{ is empty} | \Theta)^2}{\lambda_s (1 - p(\mathcal{T} \text{ is empty} | \Theta))}. \quad (59)$$

Finally, the MLE objective function becomes

$$\begin{aligned} \hat{\Theta} &= \arg \max_{\Theta} \sum_{l=1}^N \log p(\mathcal{T}^l | \mathcal{T} \text{ is non-empty}, \Theta) - \gamma \sum_{i \neq j} |\log \Theta_{ij}| \\ &= \arg \max_{\Theta} \sum_{l=1}^N \log p(\mathcal{T}^l | \Theta) - \gamma \sum_{i \neq j} |\log \Theta_{ij}| - N \left(1 - \frac{\lambda_s}{\lambda_s + \sum_{i=1}^n \Theta_{ii}} \right). \end{aligned} \quad (60)$$

Note that the same term can also be added to the objective function of the genotype MHN method if none of the input genotypes correspond to the wild type (*i.e.* a vector of zeros).

B TreeMHN with stability selection

TreeMHN can be viewed as a probabilistic graphical model with n^2 parameters, where the $(n^2 - n)$ off-diagonal entries represent the edges (\rightarrow or \neg) among the mutations (Figure S1 (a)). When the sample size N is small, it is often difficult to identify the true edges. In this case, we use a procedure called stability selection, which is particularly suitable for controlling the false discovery rate in high-dimensional graphical modelling problems [3].

More specifically, we take a random subsample I of size $\lfloor \frac{N}{2} \rfloor$ without replacement from \mathcal{J} and train TreeMHN for a given regularization parameter $\gamma > 0$. An off-diagonal entry θ_{ij} is considered as selected if it is non-zero. The set of selected edges is

$$\hat{S}_\gamma(I) = \{(i, j) \mid |\theta_{ij}| > 0, i \neq j\}. \quad (61)$$

Repeating this process for B times (B large), the probability that an entry at position (i, j) for $i \neq j$ is selected can be approximated as

$$\hat{q}_{(i,j)}(\gamma) = \frac{1}{B} \sum_{b=1}^B \mathbb{1}\{(i, j) \in \hat{S}_\gamma(I^b)\}. \quad (62)$$

For a pre-specified threshold δ and a regularization parameter γ , the set of stable edges is defined as [3]

$$\hat{S}_{\text{stable}}(\gamma) = \{(i, j) \mid \hat{q}_{(i,j)}(\gamma) \geq \delta\}. \quad (63)$$

Let S_0 be the set of true edges, S_0^c be the complement of S_0 , and $V = |S_0^c \cap \hat{S}_{\text{stable}}|$ be the falsely selected edges. To control for the expected number of false positives (*e.g.* $\mathbb{E}[V] \leq \nu = 5$), we choose to fix the threshold δ (*e.g.* 0.95) and vary γ such that the number of stable edges is bounded (Meinshausen and Bühlmann, 2010),

$$|\hat{S}_{\text{stable}}(\gamma)| \leq \lfloor \sqrt{\nu(n^2 - n)(2\delta - 1)} \rfloor. \quad (64)$$

After estimating $\hat{S}_{\text{stable}}(\gamma)$, we can then refit TreeMHN on the full dataset \mathcal{J} by masking the entries in Θ that are not selected.

C Computing probabilities of trajectories and mutational events with TreeMHN

C.1 Most probable evolutionary trajectories

In Methods, we have introduced how to compute the trajectory probability distribution over Π_S , the set of evolutionary trajectories that end with the sampling event, and Π_d , the set of evolutionary trajectories of a fixed length d . There are advantages and limitations associated with both formulations. Using trajectory probabilities over Π_d , we can compare the inferred trajectory distribution from TreeMHN with the ones from methods that do not have the notion of a sampling event. It is also easy to compute for small d . However, we cannot compare trajectories of different lengths or with the frequencies of the trajectories observed in the tumor trees based on this formulation. The trajectory probability distribution over Π_S , on the other hand, does not have these problems, but it can become infeasible to compute for even a moderate number of mutations n , because the space grows super-exponentially with n . Instead, we can enumerate the most probable trajectories that have at least one mutation dynamically with the pseudocode in Algorithm 2.

<p>Input: Θ: Mutual Hazard Network τ: Number of most probable trajectories to output λ_s: Sampling rate</p> <p>Output: $\Pi_\tau := \{\pi_{(1)}, \dots, \pi_{(\tau)}\}$: The top τ most probable trajectories $\mathbf{p}_\tau := \{p_{(1)}, \dots, p_{(\tau)}\}$: Probabilities of the τ most probable trajectories</p> <pre> 1 $\Pi_\tau \leftarrow \{(0), \dots, (0)\}$ // Initialize to trajectories with no mutations 2 $\mathbf{p}_\tau \leftarrow (0, \dots, 0)$ // Initialize to zero for all trajectory probabilities 3 $\Pi_{\text{current}} \leftarrow \{(0, 1), \dots, (0, n)\}$ // Trajectories to visit 4 while Π_{current} is non-empty do 5 $\Pi_{\text{next}} \leftarrow \{\}$ // Initialize the set of trajectories to visit next 6 for π in Π_{current} do 7 $p_\pi \leftarrow$ probability of π computed using Eq. (10) in the main text 8 if $p_\pi > \min(\mathbf{p}_\tau)$ then 9 $r \leftarrow$ index of the first minimum in \mathbf{p}_τ 10 $\pi_{(r)} \leftarrow \pi$ // Replace the r^{th} most probable trajectory by π 11 $p_{(r)} \leftarrow p_\pi$ 12 $\Pi_{\text{ch}(\pi)} \leftarrow \{(\pi, j) \mid j \in [n] \setminus \pi\}$ // Children trajectories of π by extending π by one mutation 13 $\Pi_{\text{next}} \leftarrow \{\Pi_{\text{next}}, \Pi_{\text{ch}(\pi)}\}$ // Append the children trajectories to the trajectories to visit next 14 end 15 end 16 $\Pi_{\text{current}} \leftarrow \Pi_{\text{next}}$ 17 end 18 $\mathbf{p}_\tau \leftarrow \mathbf{p}_\tau / (1 - \frac{\lambda_s}{\lambda_s + \sum_{j \in [n]} \lambda_{(0,j)}})$ // Conditioned on trajectories with at least one mutation </pre>

Algorithm 2: Most probable evolutionary trajectories

C.2 Alternative methods to compute the probabilities of mutational events

Although TreeMHN is unique in its ability to compute the probabilities of the next mutational events given a tree (Methods), we adapt five alternative methods for comparison:

1. **A frequency-based model:** A frequency-based model (N_0) predicts the next event using the relative frequencies of the mutations in the cohort, (f_1, \dots, f_n) with $f_i > 0$ and $\sum_{i=1}^n f_i = 1$. The new event can be placed randomly after any node (including the root) in \mathcal{T} with probability $1/|\mathcal{T}|$. Then, the

conditional probability of an event $\pi = (0, \sigma_1, \dots, \sigma_d) \in A(\mathcal{T}) \setminus \mathcal{T}$ happening before all the other events depends only on the relative frequency of σ_d and is given by

$$p(\pi \mid \mathcal{T}, N_0) = \frac{1}{|\mathcal{T}|} \times f_{\sigma_d} \times \mathbb{1}\{\pi \in A(\mathcal{T}) \setminus \mathcal{T}\} \times \frac{1}{Z(\mathcal{T})}, \quad \sigma_d \in [n], \quad 1 \leq d \leq n, \quad (65)$$

where $Z(\mathcal{T})$ is a normalizing constant depending on \mathcal{T} such that $\sum_{\pi \in A(\mathcal{T}) \setminus \mathcal{T}} p(\pi \mid \mathcal{T}, N_0) = 1$. We can estimate the relative frequencies $(\hat{f}_1, \dots, \hat{f}_n)$ using the subclonal genotypes weighted by their relative sizes (Supplementary Figure S2).

2. **TreeMHN with only the baseline rates:** We can run TreeMHN on a set of mutation trees with the restriction that all off-diagonal elements in the estimated network are zero. In other words, this approach assumes that all mutations are independent of each other while respecting the tree structures. The ways to compute the trajectory probabilities and the probabilities of the next mutational events given a tree stay the same.
3. **Genotype MHN using the consensus genotypes:** We can run the genotype MHN method of [4] on the consensus genotypes of patient samples (Supplementary Figure S2). Then, we use the estimated network to compute the trajectory probabilities and the probabilities of the next mutational events under the TreeMHN framework.
4. **Genotype MHN using the weighted subclonal genotypes:** Same as 2 except that we use the subclonal genotypes weighted by the size of the subclones (Supplementary Figure S2).
5. **REVOLVER:** The row-normalized matrix \mathbf{w} in REVOLVER summarizes the relative frequency of mutation j being the descendant of i in its entry w_{ij} . Given \mathbf{w} , the probability of a new event with mutation j given a tree \mathcal{T} can be computed as

$$\frac{1}{|\mathcal{T}|} \times w_{ij} \times \mathbb{1}\{j \text{ is placed after a node } i\} \times \frac{1}{Z(\mathcal{T})}, \quad (66)$$

where $Z(\mathcal{T})$ is a normalizing constant depending on \mathcal{T} such that the probabilities sum up to 1 over all possible events.

Sample	Subclone size	Mutation 1	Mutation 2	Mutation 3
A	0.55	1	0	0
	0.3	0	1	0
	0.15	0	1	1
B	0.38	1	0	1
	0.62	1	1	0
C	0.2	1	0	0
	0.35	0	1	1
	0.18	1	1	0
	0.27	1	0	0

(a)

Sample	Consensus (> 50%)		
A	1	0	0
B	1	1	0
C	1	1	0

(b)

Mutation 1	Mutation 2	Mutation 3
0.47	0.34	0.19

(c)

Figure S2: An example of converting subclonal genotypes (a) to consensus genotypes (b) using a threshold of 50%. The relative frequencies of the three mutations are shown in (c).

D Additional simulation details

D.1 Precision and recall of identifying the true MHN

In the objective functions of both TreeMHN and the genotype MHN method, there is a penalization parameter γ that controls the sparsity of the estimated network $\hat{\Theta}$. The larger γ is, the more zero entries are in $\hat{\Theta}$. We evaluate the structural differences between $\hat{\Theta}$ and a ground truth network Θ by defining the true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN) as indicated in Figure S3. In particular, an edge in $\hat{\Theta}$ is considered as a TP if and only if it has the correct direction ($\text{sign}(\hat{\theta}_{ij}) = \text{sign}(\theta_{ij})$). Next, we define the precision and recall as

$$\text{Precision} = \frac{\text{TP}}{\hat{P}} \quad \text{and} \quad \text{Recall} = \frac{\text{TP}}{P}, \quad (67)$$

where P and \hat{P} are the number of edges in the true network Θ and the estimated network $\hat{\Theta}$ respectively. The F_1 score is the harmonic mean between precision and recall, which can be computed as

$$F_1 \text{ score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \quad (68)$$

		$\hat{\theta}_{ij} = \log \hat{\Theta}_{ij}$		
		$j \ i$	$j \rightarrow i$	$j \dashv i$
$\theta_{ij} = \log \Theta_{ij}$	$j \ i$	TN	FP	FP
	$j \rightarrow i$	FN	TP	FP
	$j \dashv i$	FN	FP	TP
		<div style="display: flex; justify-content: center; align-items: center;"> } P </div>		
		<div style="display: flex; justify-content: center; align-items: center;"> } \hat{P} </div>		

Figure S3: Definition of true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN) for the structural differences between an estimated network $\hat{\Theta}$ and a true network Θ .

On average, a network estimated by randomly guessing the direction of the edges can only achieve a maximum recall of 50%. The precision of random guess is equal to $0.5(1 - s)$, where s is the proportion of zero off-diagonal entries in Θ , also called network sparsity. If 50% of the entries in Θ are zero, then the precision is only 25%. The corresponding F_1 score is $\frac{1}{3}$.

D.2 Computing trajectory probabilities with REVOLVER and HINTRA

REVOLVER [5] and HINTRA [6] are two probabilistic methods for inferring repeated trajectories from cross-sectional multi-region sequencing data. Both methods learn one tree for each patient by transferring the information across tumors, which is summarized in matrix \mathbf{w} and β , respectively. For REVOLVER, \mathbf{w} is an n -by- n matrix, where an entry w_{ij} is the number of times mutation i occurs before mutation j in the trees. By row-normalizing \mathbf{w} , one can obtain the empirical probability of mutation j being the descendant of mutation i . For HINTRA, the columns of β also correspond to the n mutations, but the rows are all possible ancestry sets, which increases super-exponentially in n . To obtain the matrices \mathbf{w} and β , we can count the frequency of every possible ancestor-descendant pair that appears in the simulated trees. Since we only consider evolutionary trajectories of length 4, it is still feasible to compute the HINTRA matrix β . Adding pseudocounts and row-normalizing the two matrices gives us $\tilde{\mathbf{w}}$ and $\tilde{\beta}$, which can be used to estimate trajectory probabilities. Given $\tilde{\mathbf{w}}$ or $\tilde{\beta}$, one simply multiplies the entries in the matrices that correspond

to the edges in the trajectories. For example, if an evolutionary trajectory is $A \rightarrow B \rightarrow C$, then we have $\tilde{w}_{\emptyset,A} \times \tilde{w}_{A,B} \times \tilde{w}_{B,C}$ for REVOLVER and $\tilde{\beta}_{\emptyset,A} \times \tilde{\beta}_{\{A\},B} \times \tilde{\beta}_{\{A,B\},C}$ for HINTRA. Moreover, REVOLVER requires an additional step of normalizing over all trajectories in Π_d to ensure that the probabilities sum up to 1.

D.3 Introducing noise into simulated trees

To evaluate the robustness of TreeMHN against tree uncertainty, we define a noise level $\epsilon \in (0, 1)$ and use it to perturb the tree structures. For each node except the root in a tree, we generate a random number $r \sim \text{Uniform}(0, 1)$ and perturb the node if $r < \epsilon$. Depending on which of the following categories the node falls into, we randomly choose one way of perturbation:

- Internal node with multiple children:
 - randomly insert a parent node (the original parent is shifted up);
 - randomly add a child node.
- Internal node with exactly one child:
 - randomly insert a parent node;
 - randomly add a child node;
 - switch order with the child;
 - be removed.
- Leaf node:
 - randomly insert a parent node;
 - randomly add a child node;
 - prune the node and reattach it to one of its siblings or the parent of its parent;
 - be removed.

To remove a node, we also need to ensure that it is not the only node in the tree.

D.4 Additional figures from simulations

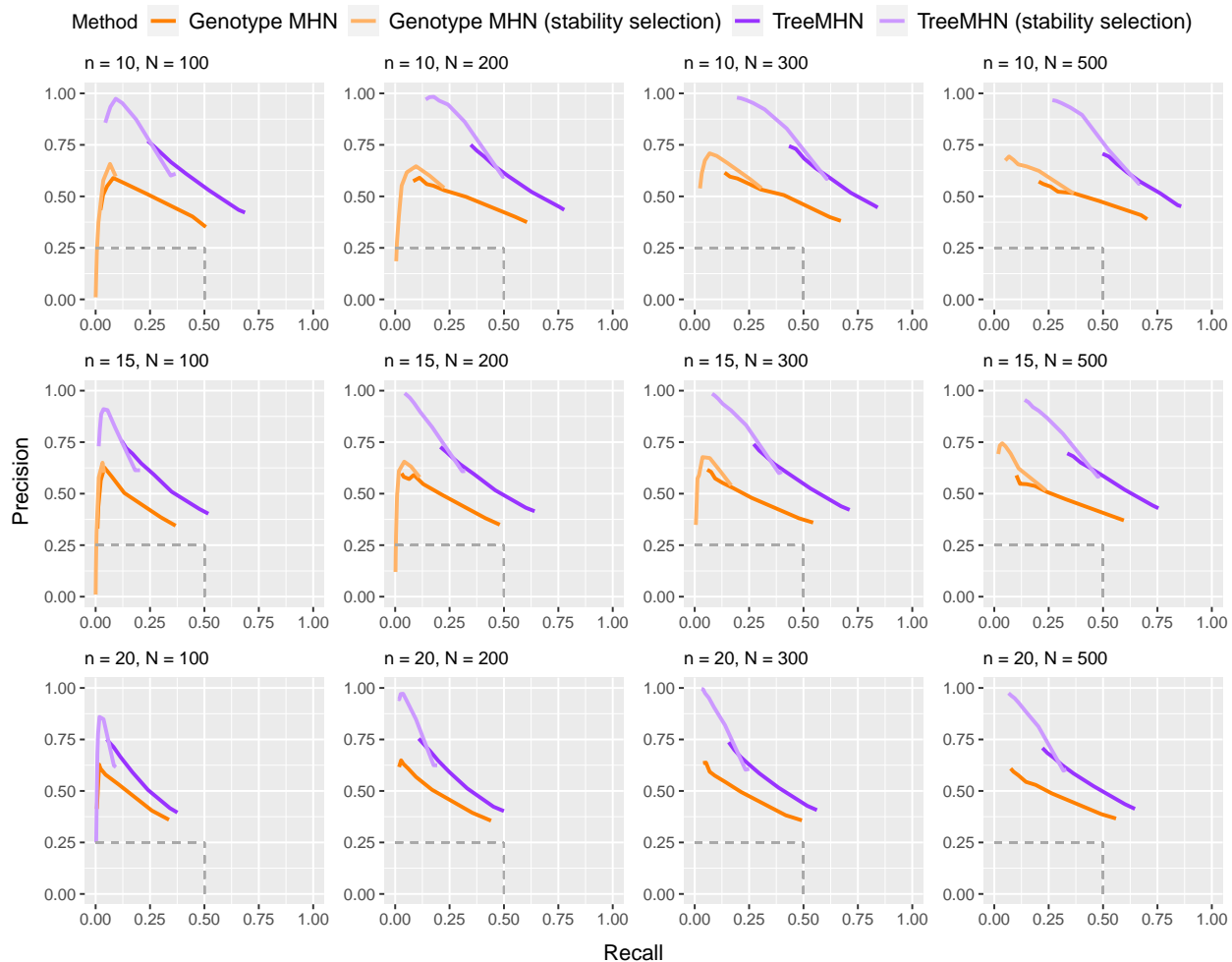


Figure S4: Performance of TreeMHN and the genotype MHN methods (with and without stability selection) in estimating the true network Θ on simulated data for $n \in \{10, 15, 20\}$ and $N \in \{200, 300, 500\}$. The precision and recall curves averaged over 100 simulation runs are plotted by comparing the full matrices Θ and $\hat{\Theta}$ (see Figure 3 for comparing sub-matrices restricted to top 50% of the mutations ranked by baseline rates). Each point on the curves corresponds to a penalization level $\gamma \in \{0.05, 0.1, 0.5, 1, 1.5, 2, 2.5, 3\}$. The dash lines indicate the performance of randomly guessing the edge directions in Θ .

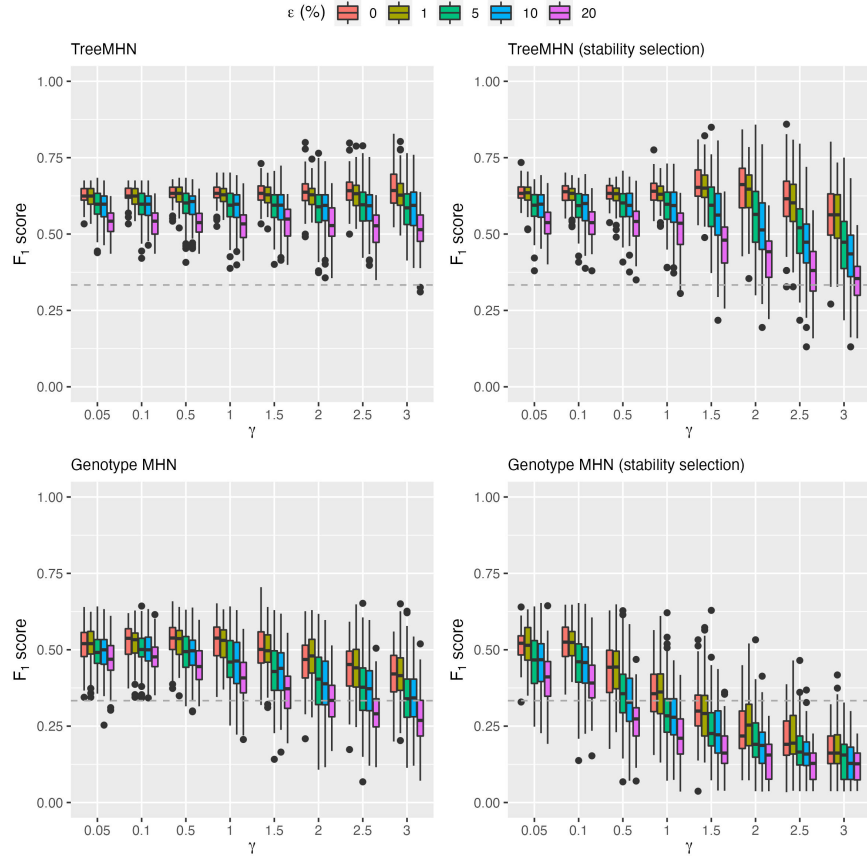


Figure S5: Performance of TreeMHN and the genotype MHN method (with and without stability selection) on perturbed simulated trees at noise level $\epsilon \in \{1\%, 5\%, 10\%, 20\%\}$ for $n = 10$ and $N = 500$. For each method, we plot the F_1 score over 100 simulation runs at $\gamma \in \{0.05, 0.1, 0.5, 1, 1.5, 2, 2.5, 3\}$. F_1 score measures the harmonic mean between precision and recall. In general, F_1 score decreases as noise level increases. In the box plots, the box represents the interquartile range (IQR) with the median inside, while the whiskers extend to the minimum and maximum values within 1.5 times the IQR, and any data outside the whiskers are shown as individual points.

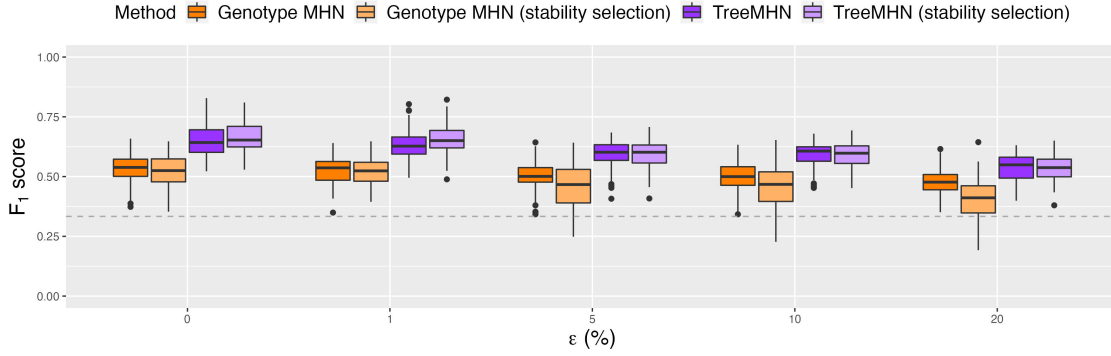


Figure S6: Performance of TreeMHN and the genotype MHN method (with and without stability selection) in estimating the true network Θ on perturbed simulated trees at noise level $\epsilon \in \{0\%, 1\%, 5\%, 10\%, 20\%\}$ for $n = 10$ and $N = 500$. F_1 score measures the harmonic mean between precision and recall. For each method and each ϵ , the box contains 100 simulation runs and corresponds to the optimal regularization parameter with respect to the average F_1 score (Figure S5). The dash line indicates the performance of random guess. In the box plots, the box represents the interquartile range (IQR) with the median inside, while the whiskers extend to the minimum and maximum values within 1.5 times the IQR, and any data outside the whiskers are shown as individual points.

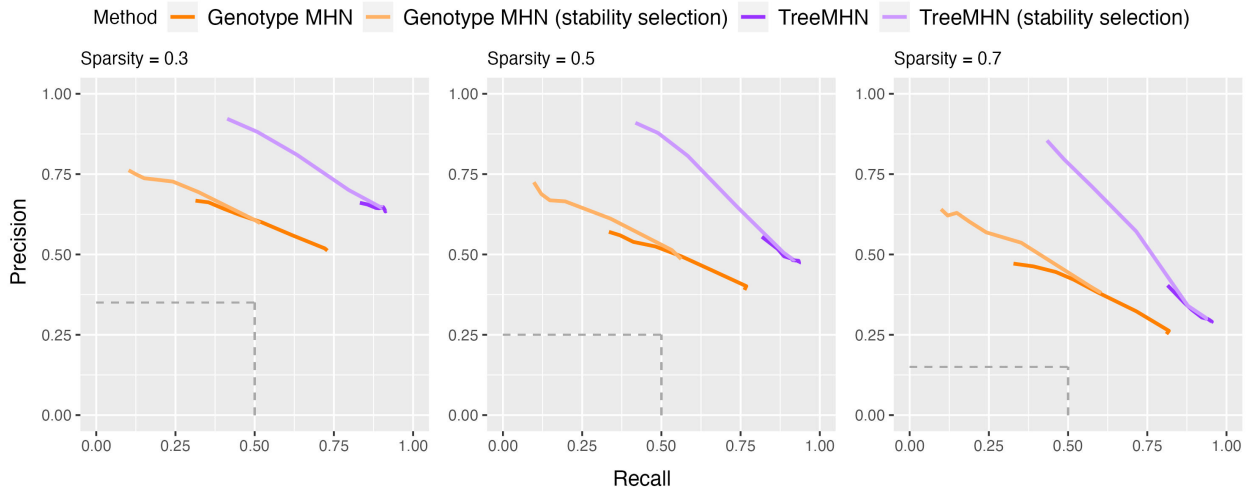


Figure S7: Performance of TreeMHN and the genotype MHN method (with and without stability selection) on simulated trees evaluated at different sparsity levels of the networks for $n = 10$ and $N = 500$. A sparsity level of 0.5 means that 50% of the off-diagonal entries in Θ are zero. The precision of random guess decreases as network sparsity increases.

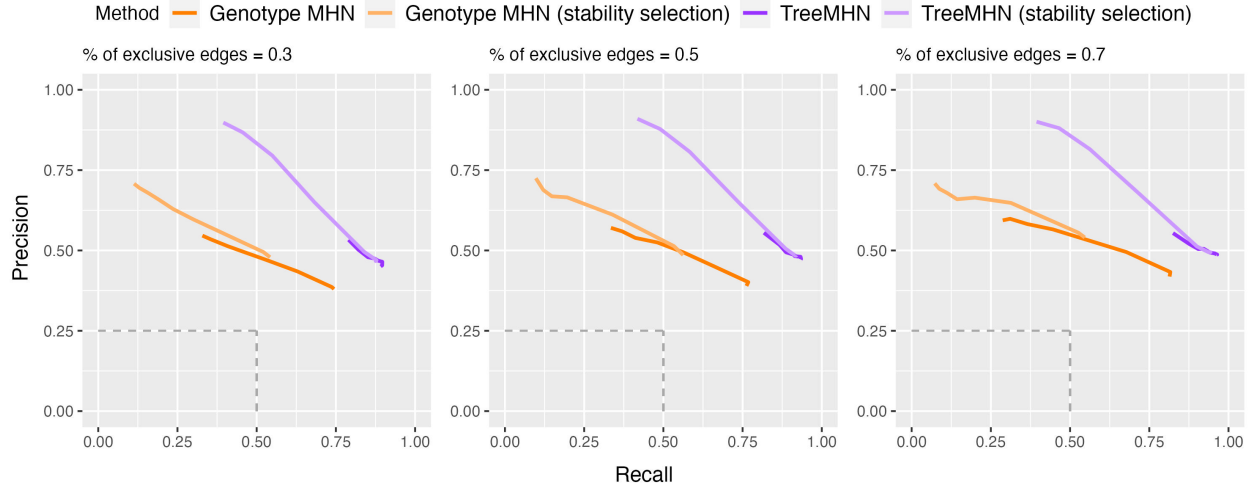


Figure S8: Performance of TreeMHN and the genotype MHN method (with and without stability selection) on simulated trees evaluated at different percentages of exclusive edges in Θ for $n = 10$ and $N = 500$. A percentage of exclusive edges equal to 0.5 means that 50% of the non-zero off-diagonal entries in Θ are negative.

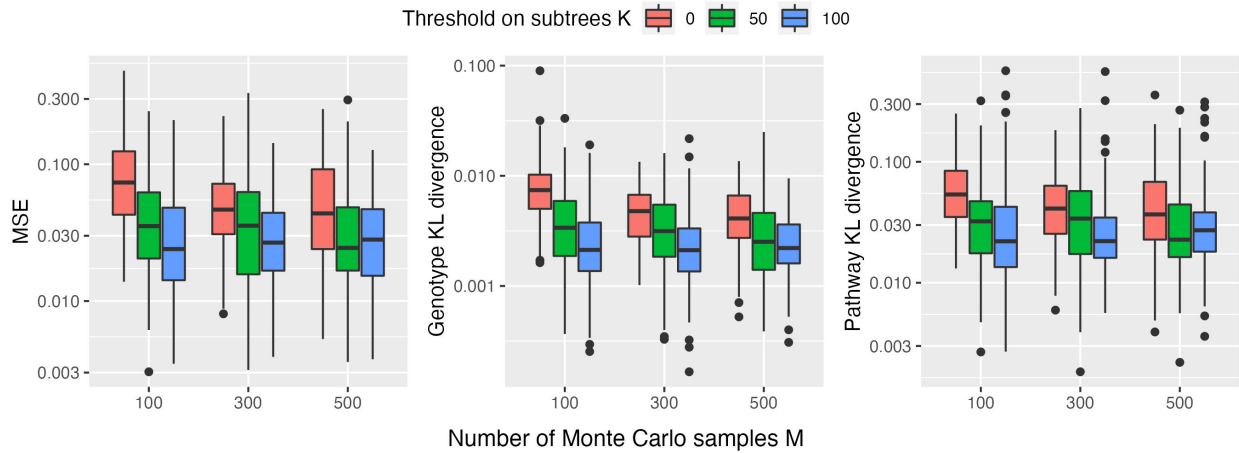


Figure S9: Comparison between the TreeMHN networks estimated using the MLE and the hybrid MC-EM methods for $n = 10$ and $N = 200$. For the hybrid MC-EM algorithm, we consider the thresholds on the subtrees $K \in \{0, 50, 100\}$ and the number of Monte Carlo samples $M \in \{100, 300, 500\}$. For each tree in the data set, we compute the expected time differences in exact form if its number of subtrees is below K . Otherwise, we use the approximation with importance sampling (Supplementary Section A.4). For each combination of K and M , we run 100 simulations and evaluate the mean squared error (MSE), the genotype KL divergence [4], and the trajectory KL divergence (Methods) between the two output matrices. Overall, the higher the threshold K and the more samples M , the smaller the differences among the networks estimated using the two inference methods. In the box plots, the box represents the interquartile range (IQR) with the median inside, while the whiskers extend to the minimum and maximum values within 1.5 times the IQR, and any data outside the whiskers are shown as individual points.

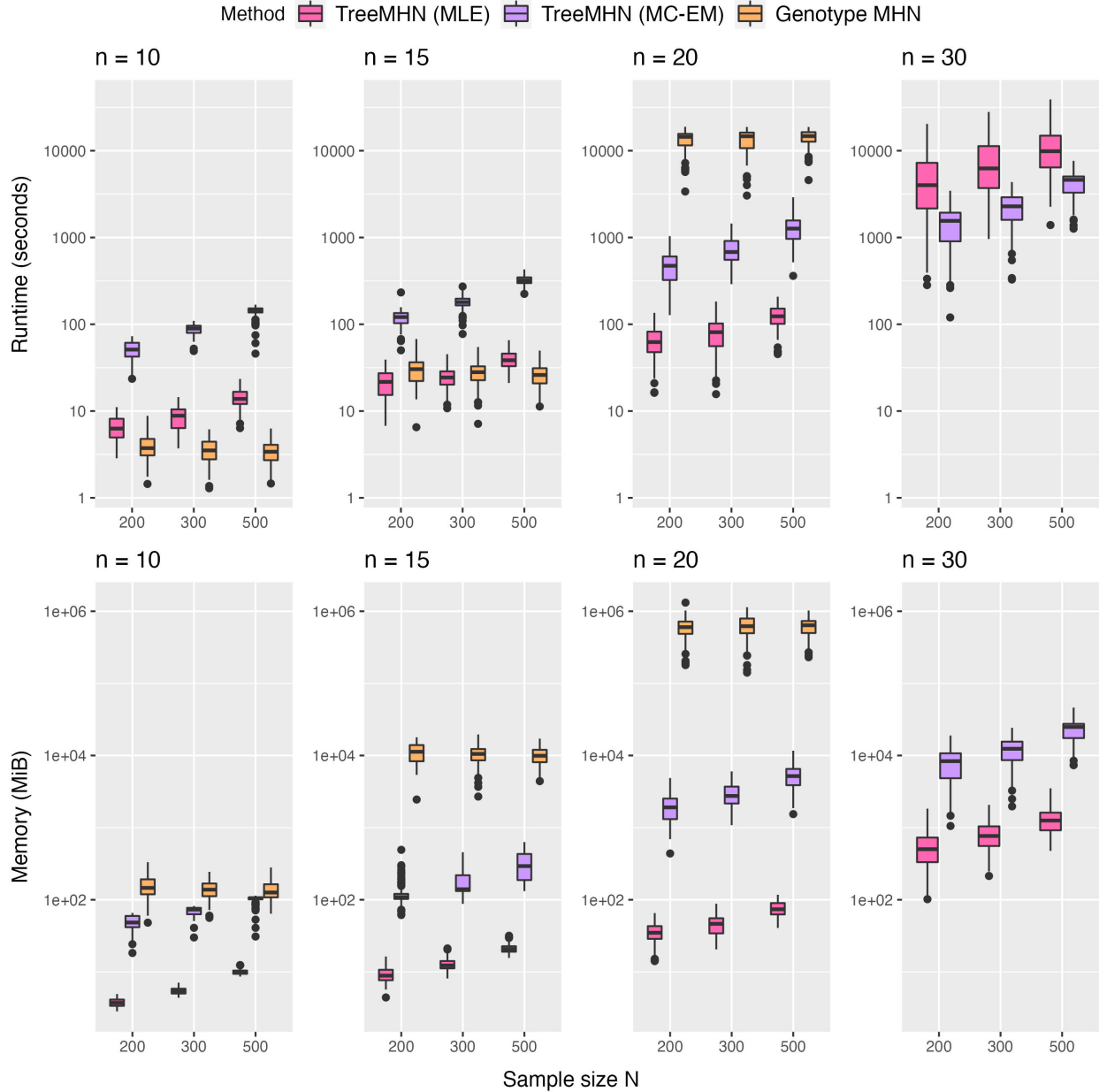


Figure S10: (a) Runtimes (wall-clock time in seconds) and (b) memory usage (in MiB) of TreeMHN (MLE and MC-EM) and the genotype MHN over 100 simulation runs for $n \in \{10, 15, 20, 30\}$, $N \in \{200, 300, 500\}$, and $\gamma = 1$. For the MC-EM version of TreeMHN, the threshold on the number of subtrees is $K = 50$, and the number of Monte Carlo samples is $M = 300$ for all configurations and simulation runs. For each method, each simulation run was performed on one CPU core of the AMD EPYC 7H12 processor on the ETH Euler cluster. The space and time complexity of the genotype MHN method increase exponentially in the number of mutations, so it is excluded in the $n = 30$ case. The runtime and memory usage of TreeMHN increase with the sample size, because the algorithm needs to go through every edge in the trees (Supplementary Section A.4). On the contrary, the genotype MHN method always summarize the genotypes into a vector of length 2^n , so larger sample size may even improve the convergence of the optimization problem. Since we did not run REVOLVER and HINTRAs using the original code but rather computed their key matrices (Methods), the runtime and memory usage of these two methods are not directly comparable. We use the R package `profmem` for memory profiling [7]. In the box plots, the box represents the interquartile range (IQR) with the median inside, while the whiskers extend to the minimum and maximum values within 1.5 times the IQR, and any data outside the whiskers are shown as individual points.

E Application to acute myeloid leukemia data

E.1 Preprocessing of the data

The dataset we analyze is a cohort of $N = 123$ AML patient samples, which contains 543 somatic mutations in $n = 31$ cancer-associated genes [8]. Each mutation tree, reconstructed by SCITE [9], corresponds to the complete evolutionary history of a tumor. SCITE assumes the infinite sites assumption for individual genomic bases. We summarize the point mutations at the gene level such that the same gene may appear in parallel lineages of a tree. Also, there may be repeated mutations along the same lineage ($A \rightarrow B \rightarrow \dots \rightarrow B$) due to gene-level summary, so we merge all such mutations to the position of their first occurrence. This rule also applies to the same mutations originating from a common direct ancestor ($B \leftarrow A \rightarrow B$).

E.2 Additional figures

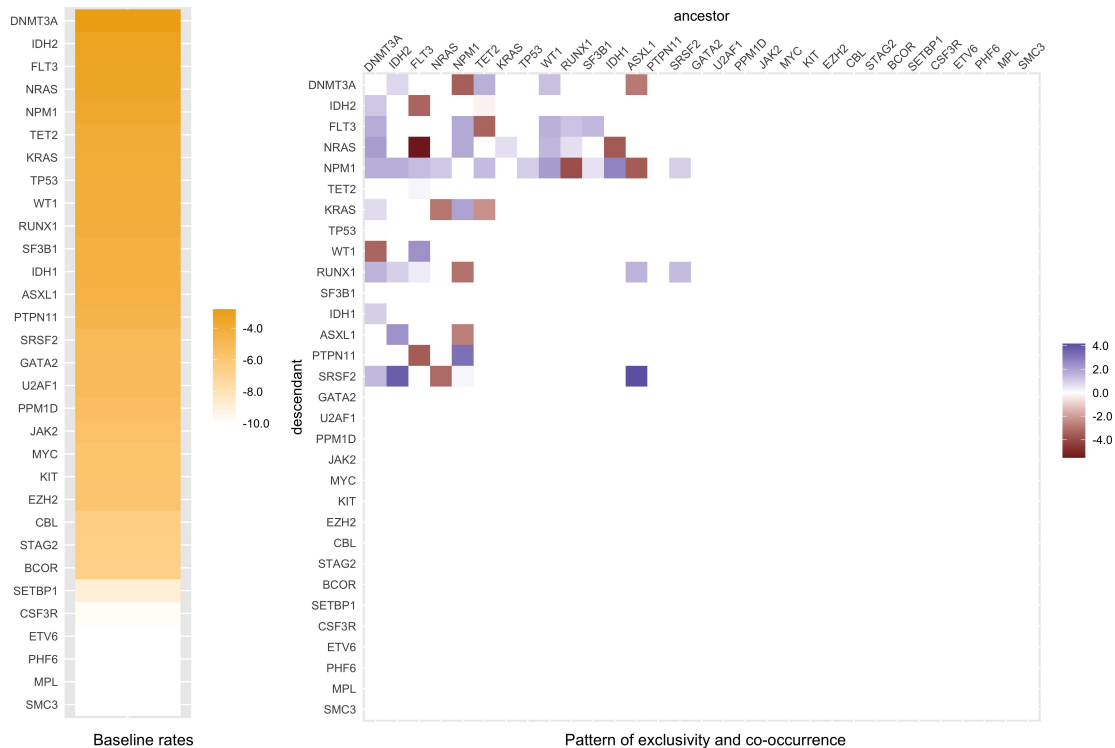
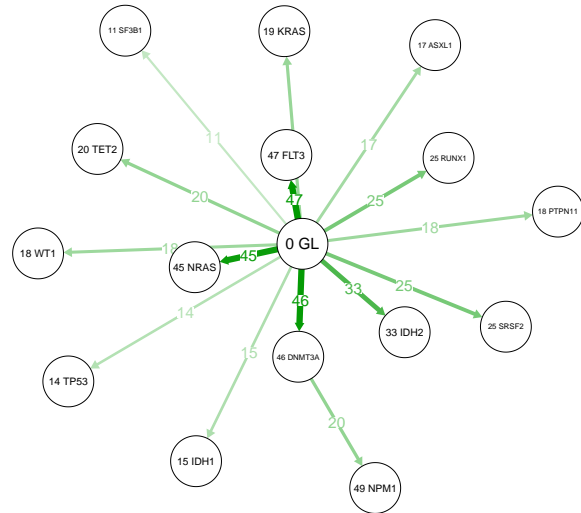


Figure S11: Full Mutual Hazard Network learned from a cohort of 123 AML patient samples [8]. This network is learned by following the stability selection procedure (Supplementary Section B) with threshold $\delta = 95\%$ and penalization parameter $\gamma = 0.1$. On the left it shows the baseline rates of the 31 mutational events, whereas the pairwise exclusive and co-occurring effects are illustrated on the right. Note that the values are displayed in logarithmic scale.

	Trajectory	Probability	Count
1	Root->DNMT3A	0.086	46
2	Root->NRAS	0.0755	45
3	Root->FLT3	0.0721	47
4	Root->IDH2	0.0601	33
5	Root->NPM1	0.0402	49
6	Root->KRAS	0.0398	19
7	Root->TET2	0.0371	20
8	Root->TP53	0.0364	14
9	Root->RUNX1	0.0325	25
10	Root->SF3B1	0.0272	11
11	Root->IDH1	0.0238	15
12	Root->WT1	0.0238	18
13	Root->PTPN11	0.0226	18
14	Root->ASXL1	0.0209	17
15	Root->SRSF2	0.0129	25
16	Root->DNMT3A->NRAS	0.0121	18
17	Root->DNMT3A->FLT3	0.0096	14
18	Root->IDH2->SRSF2	0.0081	12
19	Root->NPM1->FLT3	0.0066	14
20	Root->NPM1->PTPN11	0.0061	7
21	Root->FLT3->WT1	0.0057	6
22	Root->FLT3->NPM1	0.0056	9
23	Root->ASXL1->SRSF2	0.0056	7
24	Root->IDH1->NPM1	0.0055	5
25	Root->NPM1->NRAS	0.0054	10
26	Root->TET2->DNMT3A	0.0054	7
27	Root->FLT3->WT1->NPM1	0.0043	2
28	Root->NRAS->NPM1	0.004	6
29	Root->IDH2->NPM1	0.004	8
30	Root->DNMT3A->FLT3->NPM1	0.0039	2
31	Root->RUNX1->FLT3	0.0031	4
32	Root->SF3B1->FLT3	0.0028	5
33	Root->NPM1->KRAS	0.0027	6
34	Root->DNMT3A->IDH2	0.0027	8
35	Root->TET2->NPM1	0.0026	8
36	Root->DNMT3A->RUNX1	0.0026	6
37	Root->DNMT3A->NPM1	0.0026	20
38	Root->DNMT3A->NPM1->FLT3	0.0024	6
39	Root->NRAS->DNMT3A	0.0023	1
40	Root->FLT3->DNMT3A	0.0023	2
	Median		10.5

(a)



(b)

Figure S12: Comparison between TreeMHN and CONETT [10] on the inferred evolutionary trajectories from the AML dataset. (a) Table of top 40 most probable evolutionary trajectories inferred by TreeMHN. Each row corresponds to a trajectory along with its inferred probability and the number of times it appears in the transitive closures of the mutation trees (*i.e.* if there are directed edges from A to B and from B to C, then we add an edge from A to C). (b) Germline-rooted evolutionary trajectory tree computed by CONETT, which is conserved in at least 10 patients at 10% significance level. The number on an edge indicates the number of patients having the path from the GL node to the node which the edge is pointing to.

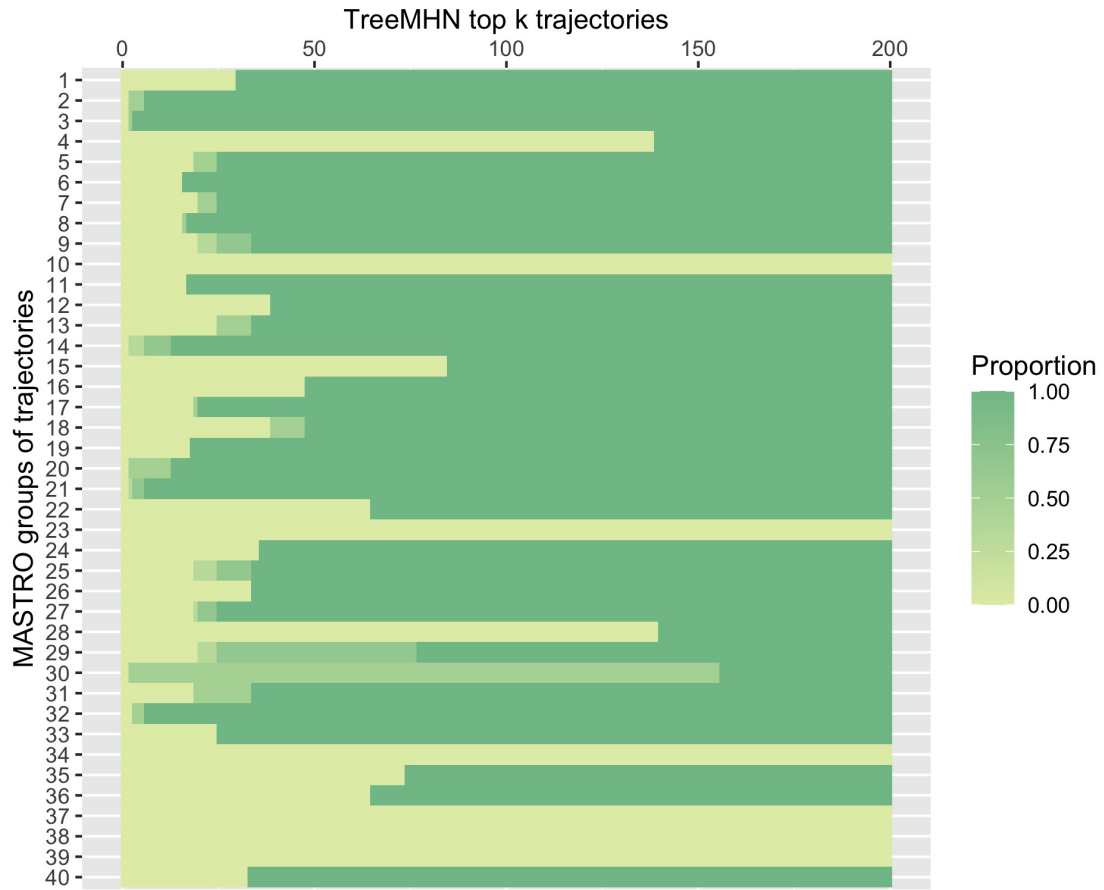


Figure S13: Proportion of significantly conserved trajectories reported by MASTRO [11] found in the top k most probable trajectories predicted by TreeMHN for the AML dataset. Here we consider $k \leq 200$. Each row corresponds to one group in Supplementary Figure 5 of [11].

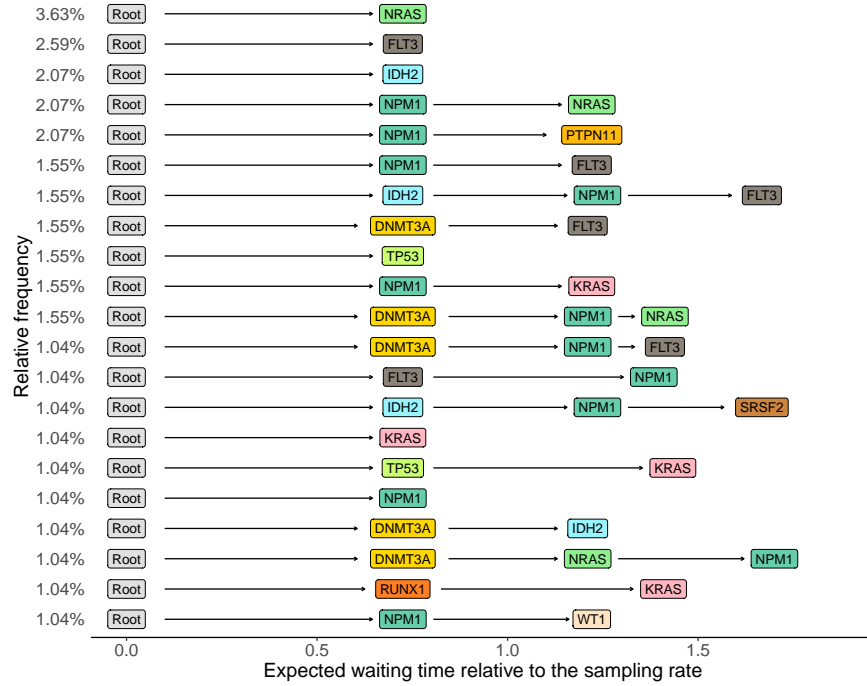


Figure S14: Observed evolutionary trajectories that appear at least twice in the AML tumor trees [8]. The rows represent the evolutionary trajectories, labeled and ordered by their relative frequencies. The horizontal positions of the mutations correspond to their expected waiting times relative to the sampling rate of $\lambda_s = 1$, which are computed based on the estimated AML network. Despite the small sample size, the relative frequencies of the trajectories still match closely with the estimated trajectory probabilities.

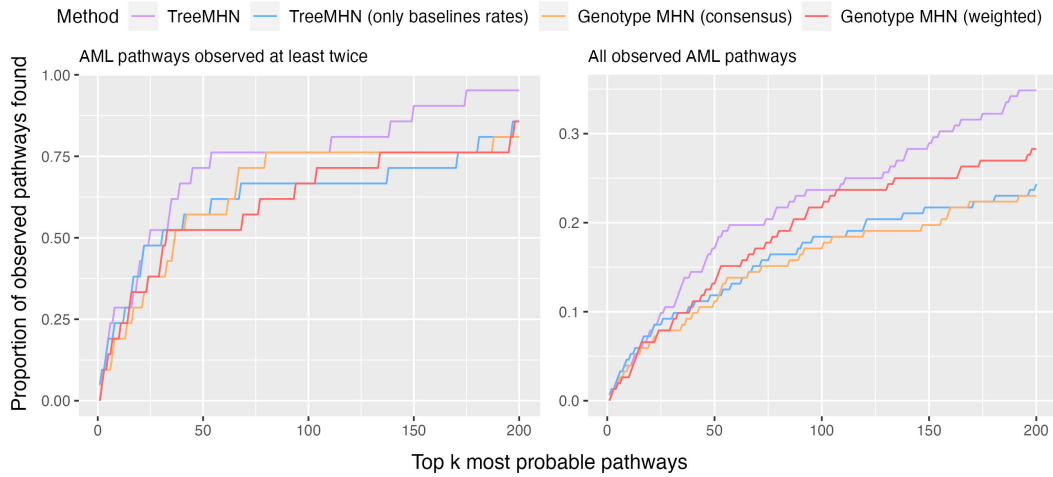


Figure S15: Proportion of observed AML trajectories found in the top k most probable trajectories predicted by four methods: TreeMHN, TreeMHN with only baseline rates, genotype MHN using the consensus subclonal genotypes, genotype MHN using the weighted subclonal genotypes (Supplementary Section C.2). Here we consider $k \leq 200$. On the left we show the comparison on the AML trajectories observed at least twice in the cohort, whereas the one on the right includes all observed trajectories. For scalability, we run both variants of genotype MHNs by focusing on the top 15 mutations that have non-zero off-diagonal entries in the estimated network (Supplementary Figure S11).

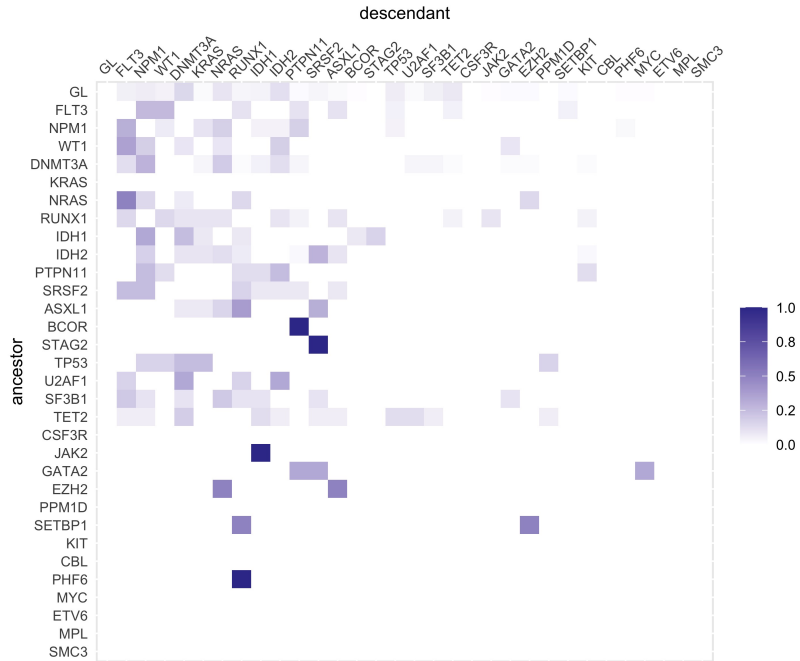


Figure S16: REVOLVER’s information transfer matrix \mathbf{w} estimated on the AML dataset after normalization. Each entry w_{ij} represents the empirical probability of mutation j being the descendant of mutation i .

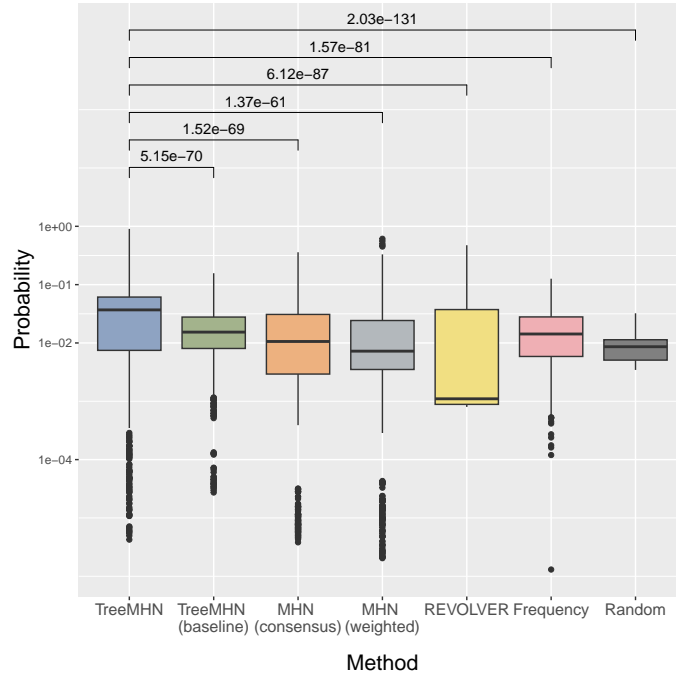


Figure S17: Performance assessment on retrospective predictions on the AML dataset. Each boxplot shows the probabilities computed on the 1370 pairs of rooted subtrees and their corresponding downstream events, enumerated from the 123 primary tumor trees. Each point in the plot corresponds to the probability of a downstream event given a rooted subtree of a primary tumor tree. The last column shows the probabilities of random guess, where each possible event gets equal weight. Since for each rooted subtree, the number of possible events is different depending on the size of the tree, the last boxplot is not a line. The p-values with Bonferroni correction for the two-sided Wilcoxon signed-rank tests between TreeMHN and alternative methods are also displayed. In the box plots, the box represents the interquartile range (IQR) with the median inside, while the whiskers extend to the minimum and maximum values within 1.5 times the IQR, and any data outside the whiskers are shown as individual points.

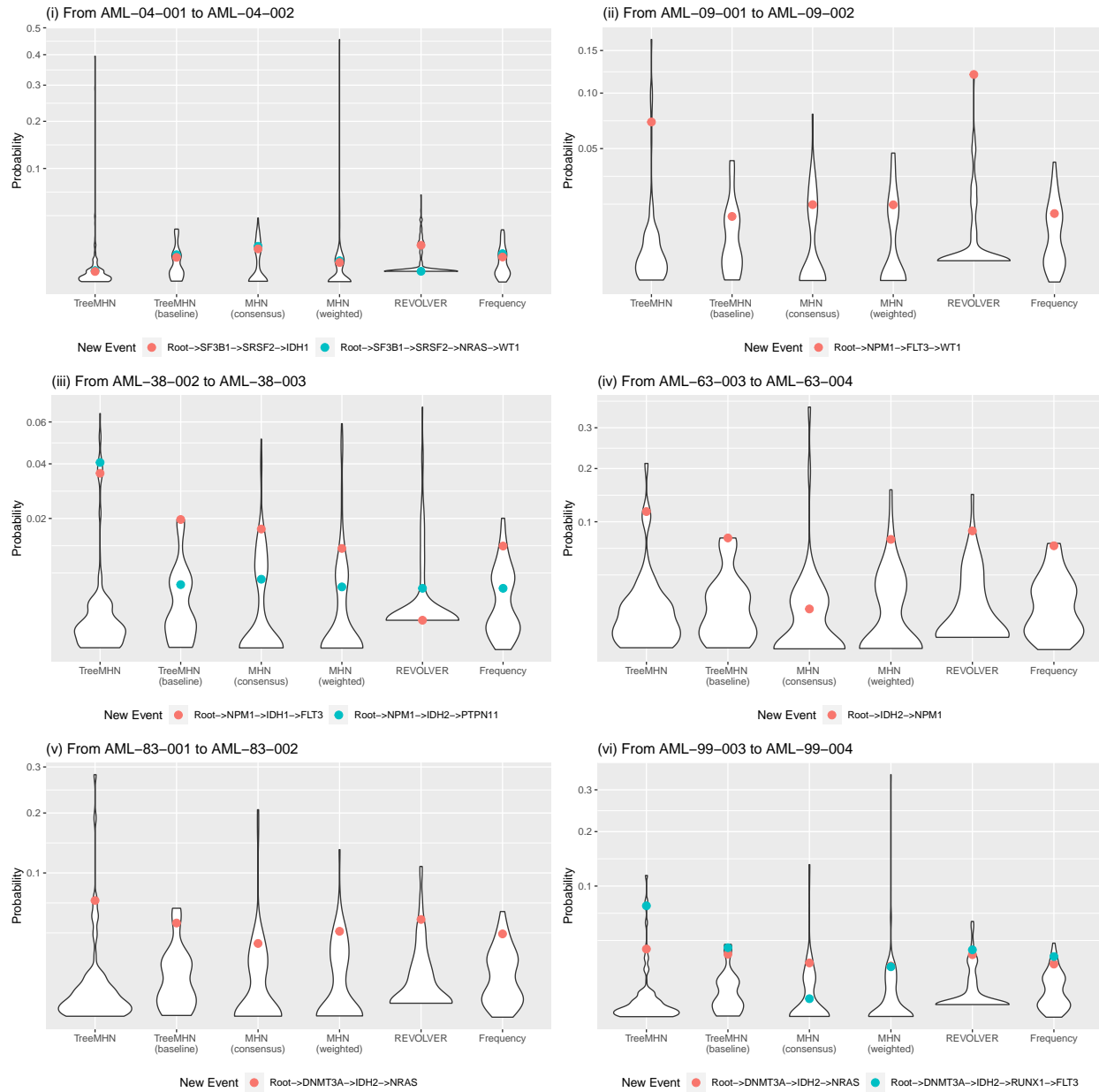


Figure S18: Performance assessment on forward predictions on the AML longitudinal samples. Each panel corresponds to a pair of consecutive trees where the second tree has at least one new event. For each method, the violin plot shows the estimated probability distribution of all possible events, and the new events are colored dots. These events correspond to the events shown in Figure 6c of the main text.

F Application to non-small-cell lung cancer data

F.1 Preprocessing of the data

To apply TreeMHN to the TRACERx NSCLC data set [12], we retrieve the uncorrelated phylogenetic trees for $N = 99$ patients and $n = 79$ driver mutations from the R package `evoverse.datasets` [13]. Each patient sample has multiple clone trees with different scores, and each node in the clone tree can have multiple mutations, whose ordering cannot be determined. Hence, for each clone tree, we enumerate or randomly sample the permutations of the mutations in the clones (depending on the clone size), and assign equal weights to the resulting fully resolved trees. Then, we multiply the weights by the score of the original clone tree. Finally, we use a weighted version of TreeMHN, where the total weights of the trees for each patient sum up to 1.

F.2 Additional figures

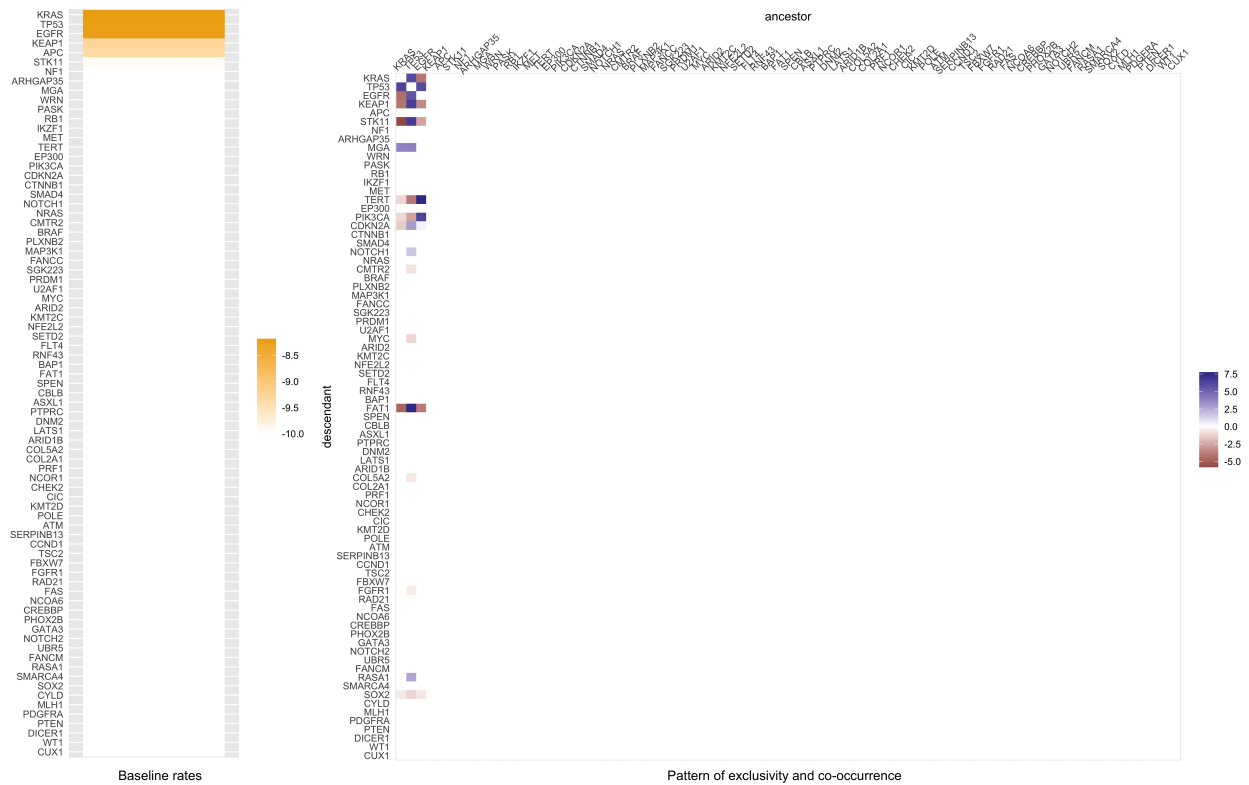


Figure S19: Full Mutual Hazard Network learned from a cohort of 99 NSCLC patient samples [5, 12]. This network is learned by following the stability selection procedure (Supplementary Section B) with threshold $\delta = 99\%$ and penalization parameter $\gamma = 0.001$. On the left it shows the baseline rates of the 79 mutational events, whereas the pairwise exclusive and co-occurring effects are illustrated on the right. Note that the values are displayed in logarithmic scale.

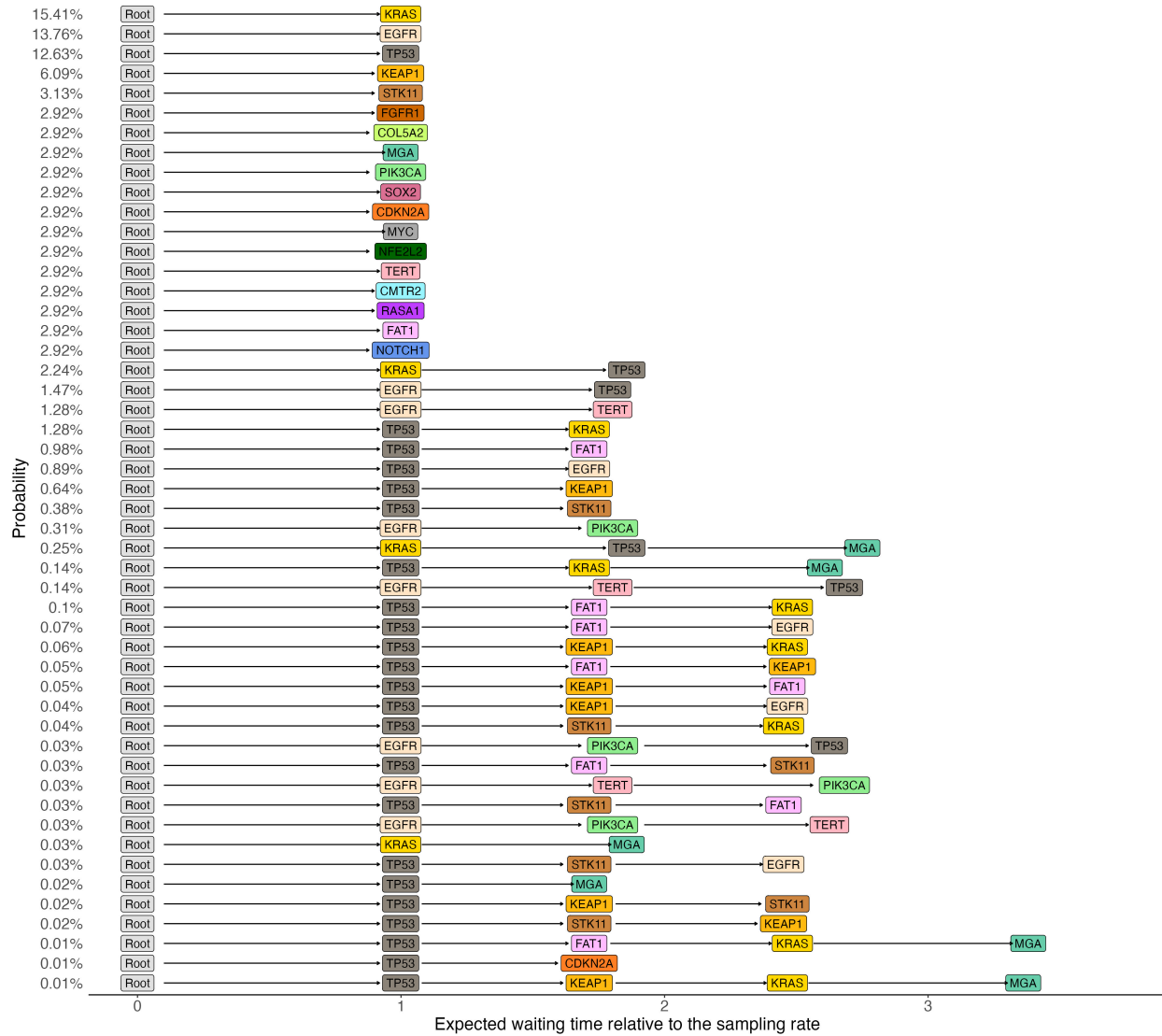


Figure S20: Top 50 most probable evolutionary trajectories inferred from the partial Mutual Hazard Network of the TRACERx NSCLC dataset (Figure 8). Note that the total number of evolutionary trajectories is in the order of $18!$.

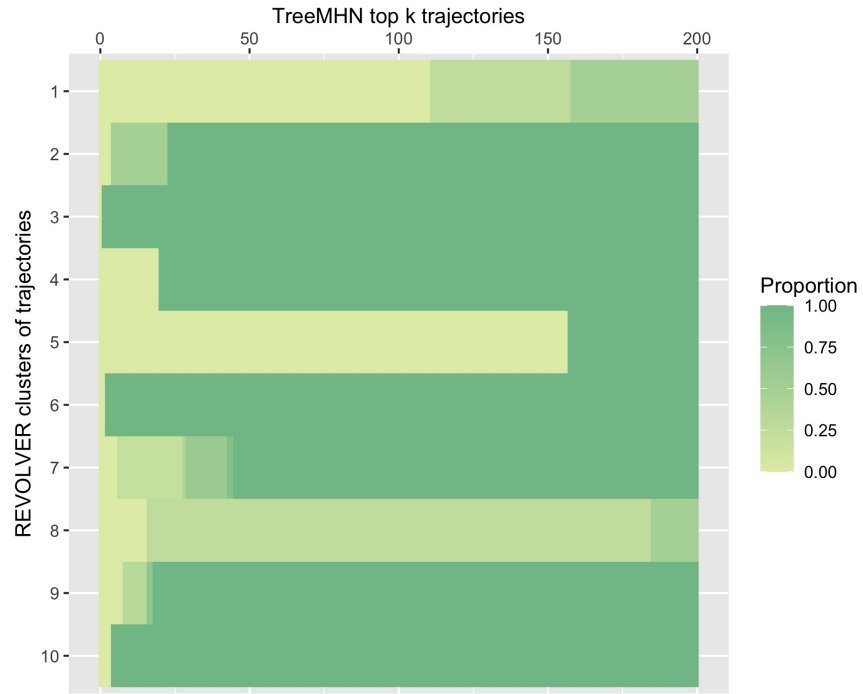


Figure S21: Proportion of repeated trajectories reported by REVOLVER found in the top k most probable trajectories predicted by TreeMHN for the NSCLC dataset. Here we consider $k \leq 200$. Each line corresponds to one cluster in Supplementary Figure 7 of [5].

G Application to breast cancer data

G.1 Preprocessing of the data

Following [14], we use the phylogenetic trees inferred by SPRUCE [15] for the breast cancer data in [16]. The number of patients in the source data is $N = 1756$. Christensen et al. [14] reduce this number to 1315 by focusing on SNVs in copy-neutral autosomal regions. We partition the patients into 8 subgroups according to their hormone receptor status (HR+/HER2+, HR+/HER2-, HR-/HER2+, Triple Negative) and sample type (primary vs. metastasis). For primary tumors, we consider only the treatment-free samples to avoid confounding signals. Moreover, we restrict the analysis to the union of mutations that appear in at least 10% of the patients in each subgroup. These preprocessing steps result in $n = 19$ mutations and $N = 1152$ patients with 1232 phylogenetic trees. Trees for the same patient sample get equal weights. The distributions of patients and mutations across subgroups are provided in Supplementary Table S1 and Supplementary Table S2 respectively.

	HR+/HER2+	HR+/HER2-	HR-/HER2+	Triple Negative
Primary	31	442	14	57
Metastasis	73	444	26	65

Table S1: Number of advanced breast cancer patients in each of the eight subgroup after the preprocessing steps.

	HR+/HER2+	HR+/HER2-	HR-/HER2+	Triple Negative
Primary	10	17	8	14
Metastasis	15	19	12	12

Table S2: Number of mutations in each of the eight subgroup after the preprocessing steps.

G.2 Additional figures

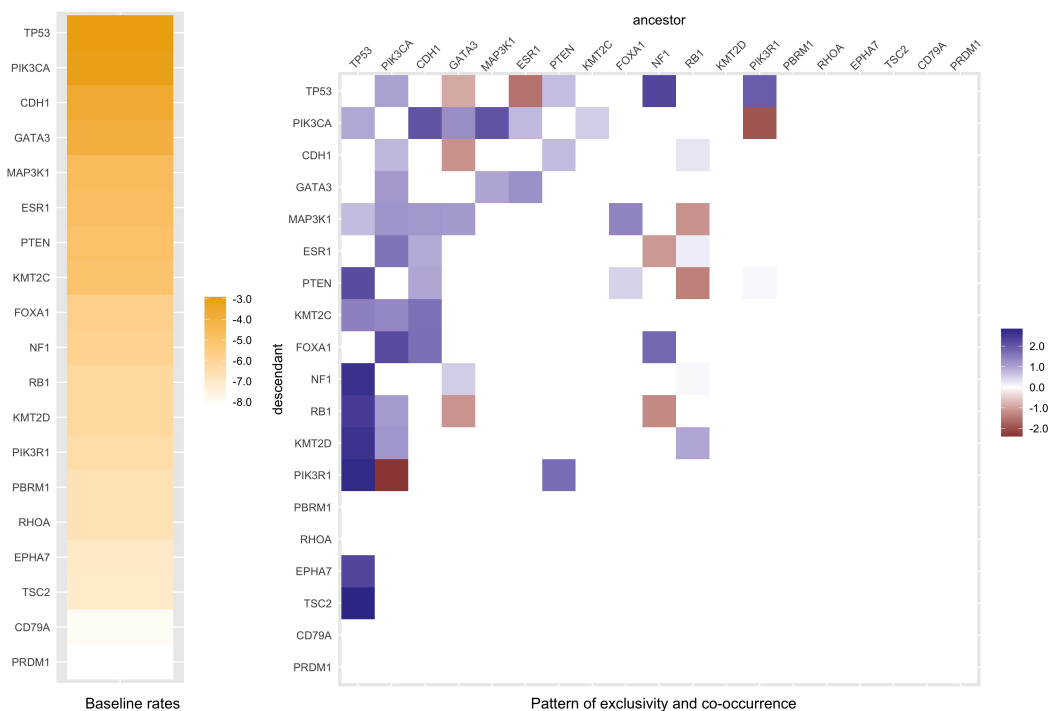


Figure S22: Full Mutual Hazard Network learned from a cohort of breast cancer patient samples [14, 16]. This network is learned by following the stability selection procedure (Supplementary Section B) with threshold $\delta = 99\%$ and penalization parameter $\gamma = 0.01$. On the left it shows the baseline rates of the 19 mutational events, whereas the pairwise exclusive and co-occurring effects are illustrated on the right. Note that the values are displayed in logarithmic scale.

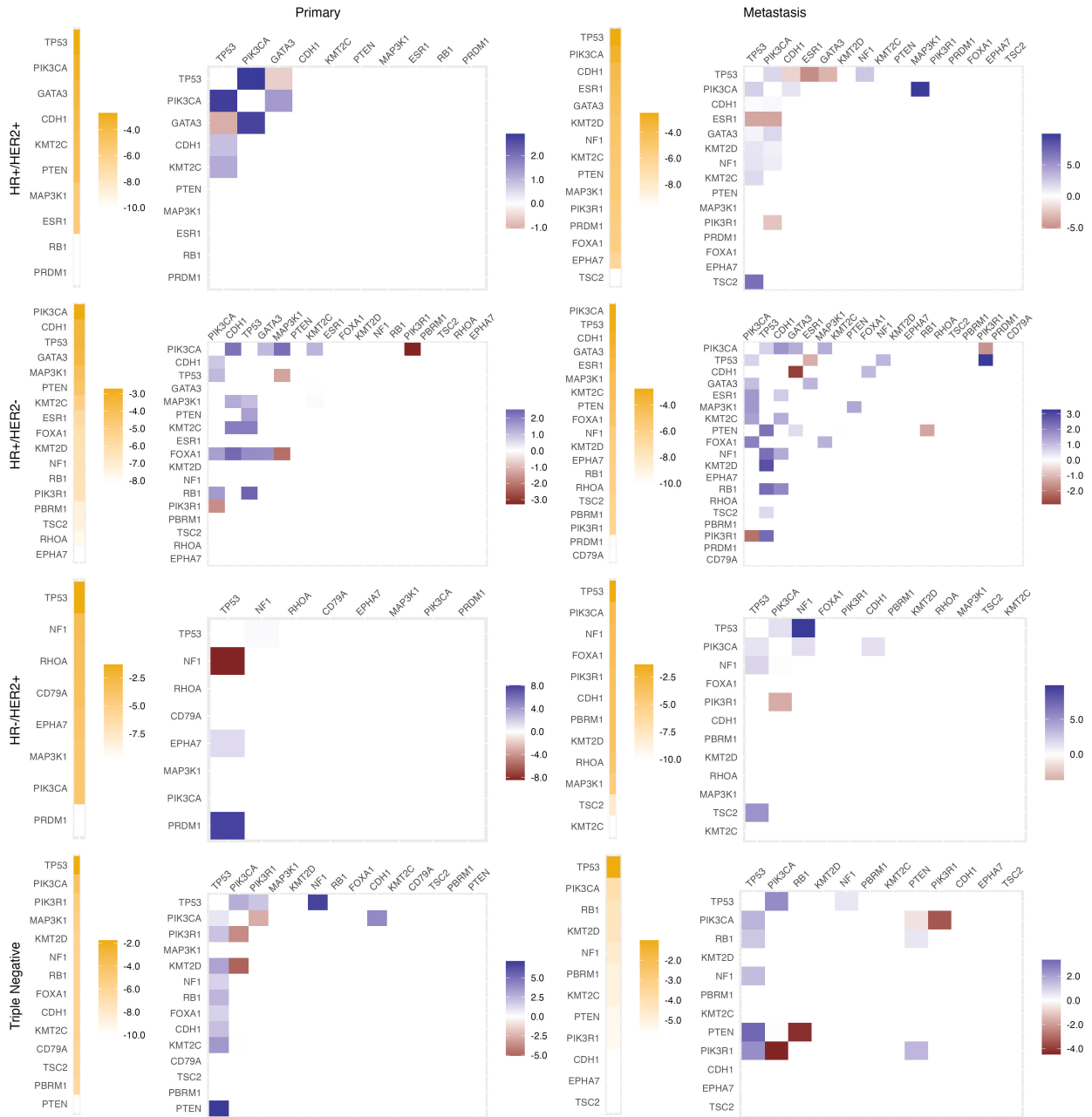


Figure S23: Full Mutual Hazard Networks for the eight subgroups defined in Supplementary Section G.1. Using the stability selection procedure with threshold $\delta = 99\%$ (Supplementary Section B), we control the expected false positive rate to be at most 10% by varying the penalty level γ for each subgroup based on the number of mutations (Supplementary Table S2).

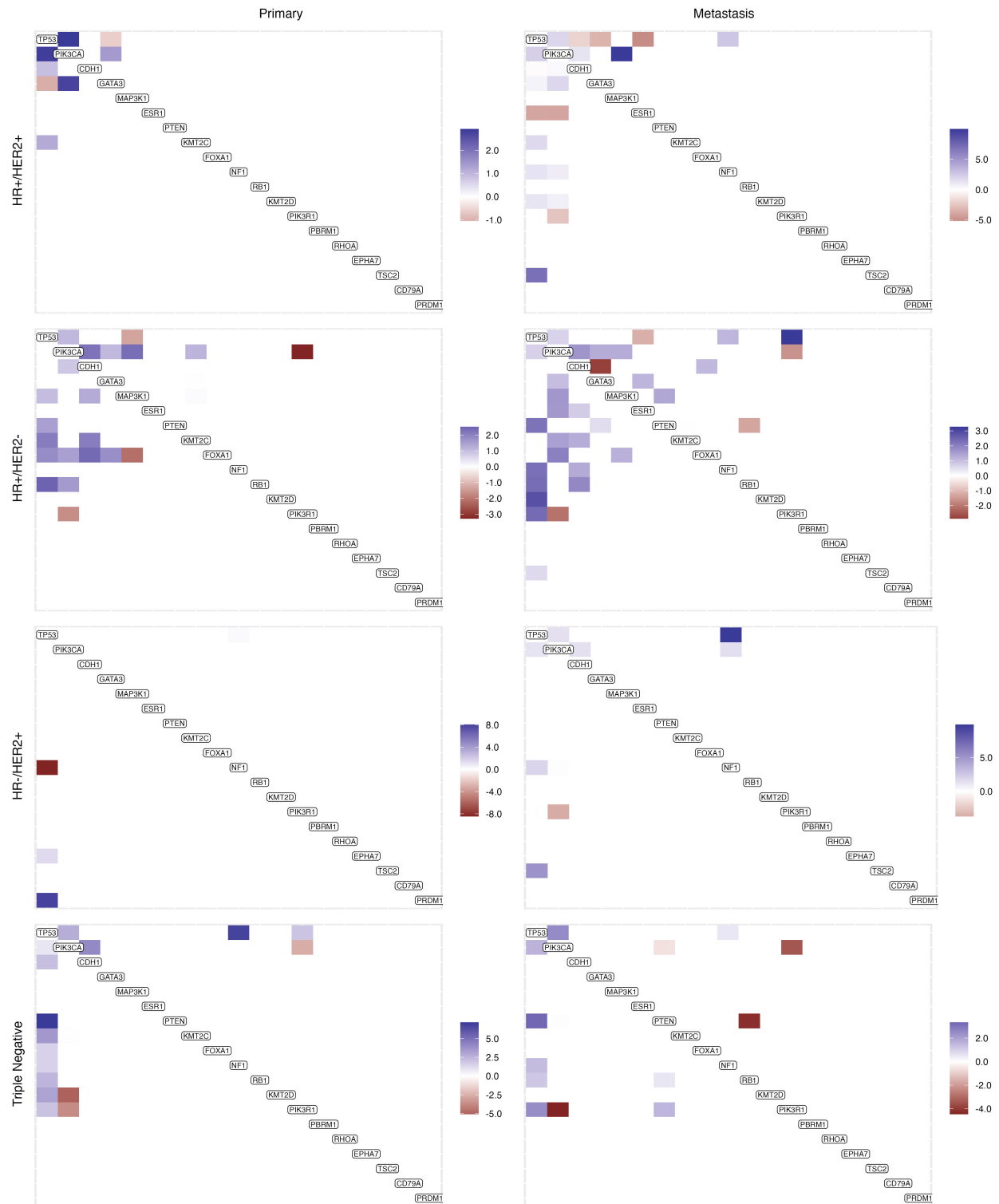


Figure S24: Partial Mutual Hazard Networks for the eight subgroups defined in Supplementary Section G.1. The columns and rows of these matrices are ordered by decreasing baseline rates in the main matrix (Supplementary Figure S22).

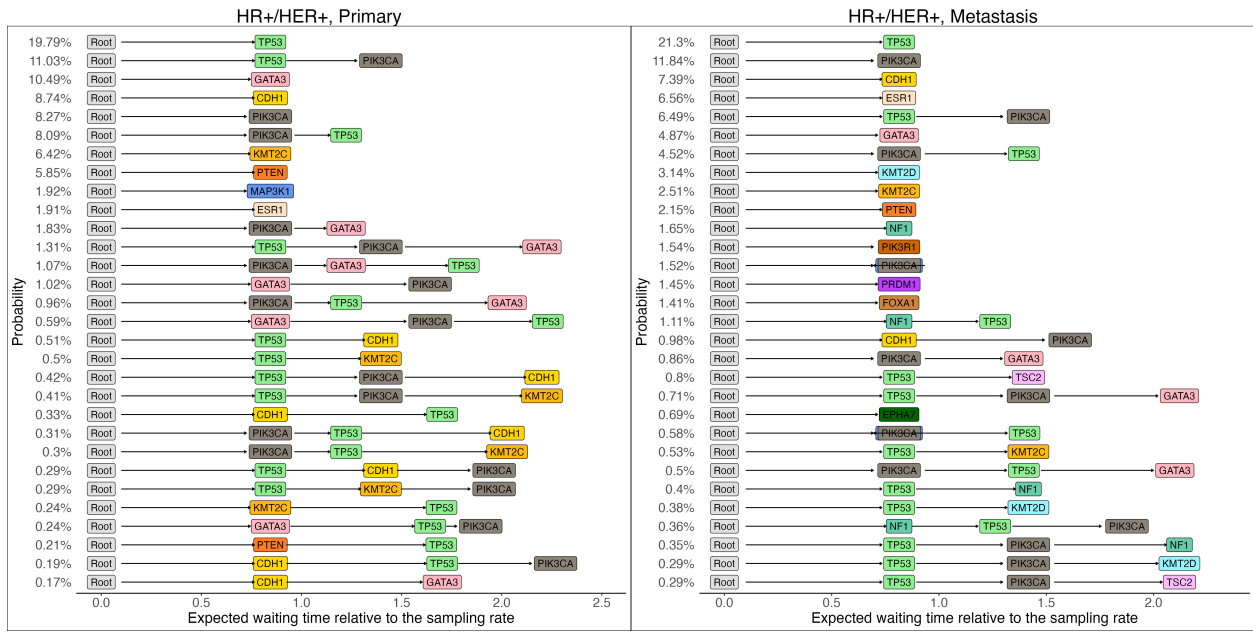


Figure S25: Top 30 most probable evolutionary trajectories of the HR+/HER+ primary (left panel) and metastasis (right panel) subgroups for the breast cancer data [16].

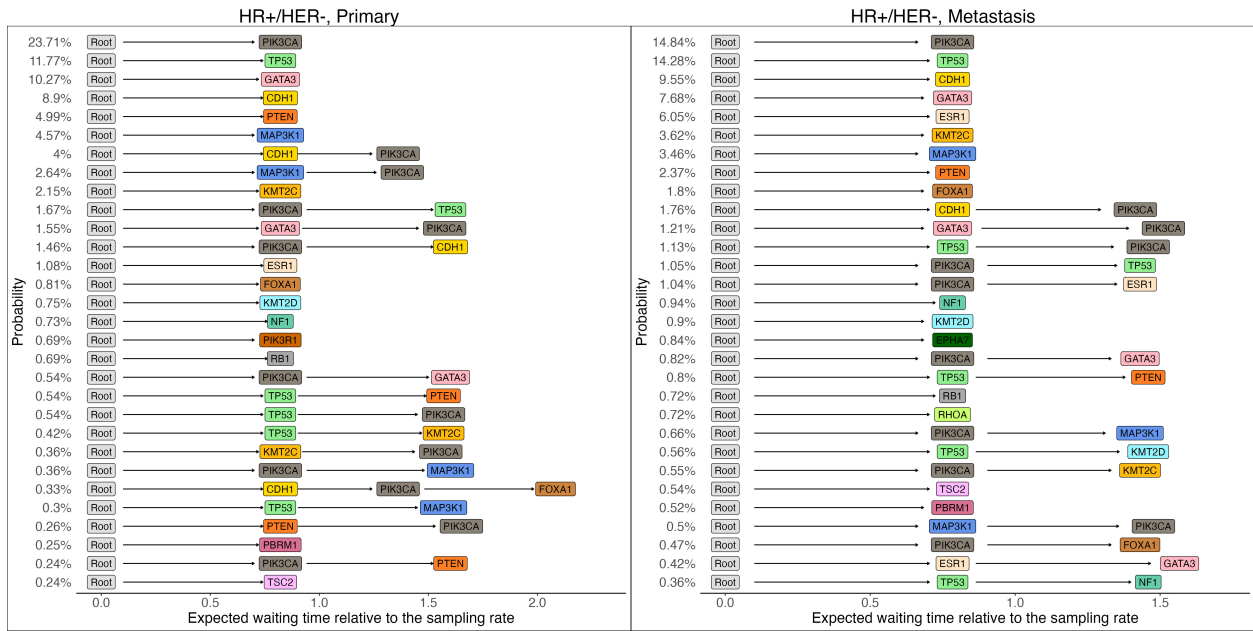


Figure S26: Top 30 most probable evolutionary trajectories of the HR+/HER- primary (left panel) and metastasis (right panel) subgroups for the breast cancer data [16].

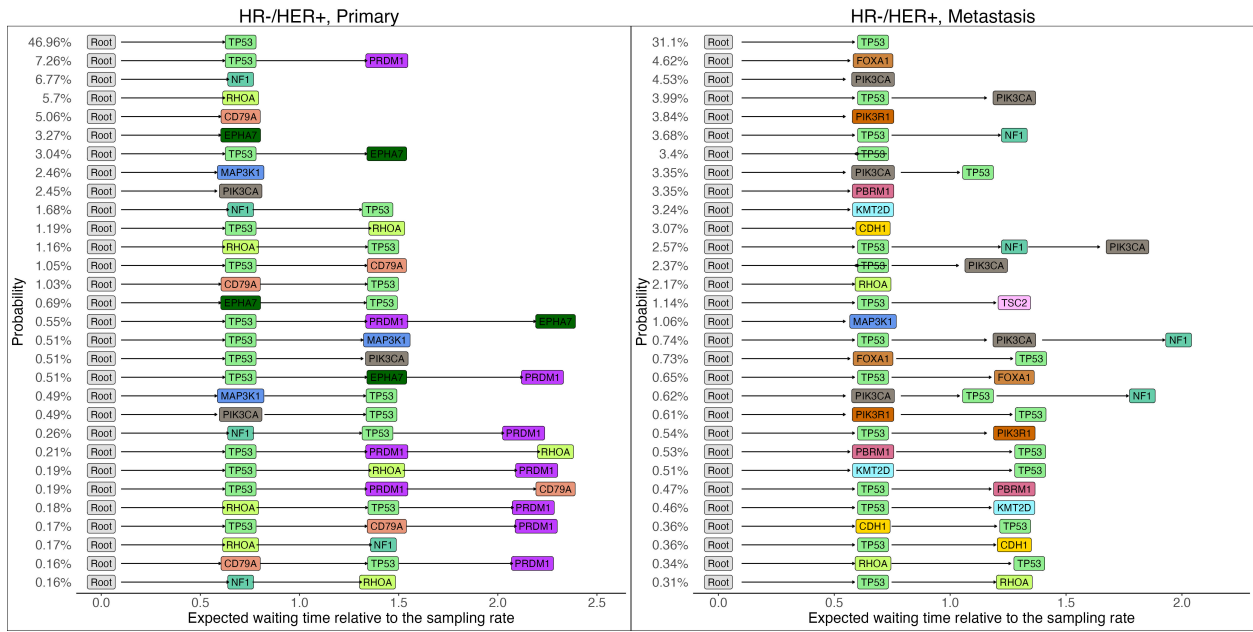


Figure S27: Top 30 most probable evolutionary trajectories of the HR-/HER+ primary (left panel) and metastasis (right panel) subgroups for the breast cancer data [16].

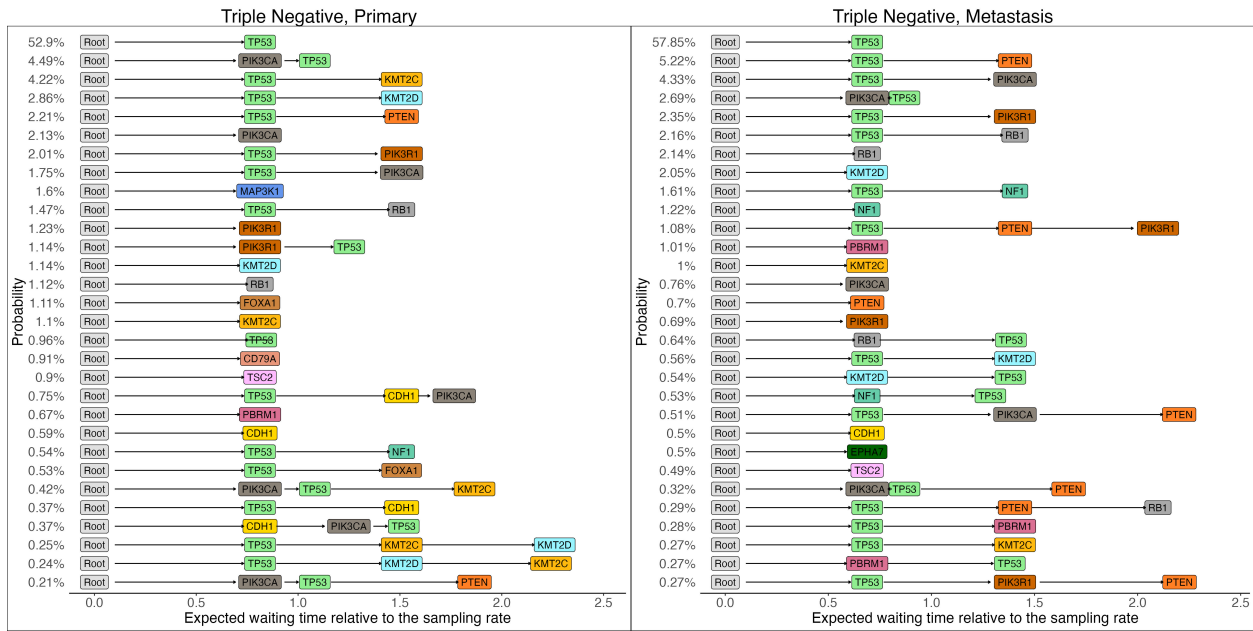


Figure S28: Top 30 most probable evolutionary trajectories of the triple-negative primary (left panel) and metastasis (right panel) subgroups for the breast cancer data [16].

References

1. Norris, J. R. *Markov chains* (1998).
2. Spiegel, M. R. *Laplace transforms* (1965).
3. Meinshausen, N. & Bühlmann, P. Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **72**, 417–473 (2010).
4. Schill, R., Solbrig, S., Wettig, T. & Spang, R. Modelling cancer progression using Mutual Hazard Networks. *Bioinformatics* **36**, 241–249 (2020).
5. Caravagna, G. *et al.* Detecting repeated cancer evolution from multi-region tumor sequencing data. *Nature Methods* **15**, 707–714 (2018).
6. Khakabimamaghani, S. *et al.* Collaborative intra-tumor heterogeneity detection. *Bioinformatics* **35**, i379–i388 (2019).
7. Bengtsson, H. *profmem: Simple Memory Profiling for R* R package version 0.6.0 (2020). <https://CRAN.R-project.org/package=profmem>.
8. Morita, K. *et al.* Clonal evolution of acute myeloid leukemia revealed by high-throughput single-cell genomics. *Nature Communications* **11**, 5327 (2020).
9. Jahn, K., Kuipers, J. & Beerenwinkel, N. Tree inference for single-cell data. *Genome Biology* **17**, 86 (2016).
10. Hodzic, E. *et al.* Identification of conserved evolutionary trajectories in tumors. *Bioinformatics* **36**, i427–i435 (2020).
11. Pellegrina, L. & Vandin, F. Discovering significant evolutionary trajectories in cancer phylogenies. *Bioinformatics* **38**, ii49–ii55 (2022).
12. Jamal-Hanjani, M. *et al.* Tracking the evolution of non–small-cell lung cancer. *New England Journal of Medicine* **376**, 2109–2121 (2017).
13. Caravagna, G. *evoverse.datasets: Data released in the evoverse*. R package version 0.1.0 (2021).
14. Christensen, S., Kim, J., Chia, N., Koyejo, O. & El-Kebir, M. Detecting evolutionary patterns of cancers using consensus trees. *Bioinformatics* **36**, i684–i691 (2020).
15. El-Kebir, M., Satas, G., Oesper, L. & Raphael, B. J. Inferring the mutational history of a tumor using multi-state perfect phylogeny mixtures. *Cell Systems* **3**, 43–53 (2016).
16. Razavi, P. *et al.* The genomic landscape of endocrine-resistant advanced breast cancers. *Cancer Cell* **34**, 427–438 (2018).