Supplementary Material — SPhyR: Tumor Phylogeny Estimation from Single-Cell Sequencing Data

Mohammed El-Kebir[1]*

[1]Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL 61801

Contents

List of Figures

List of Tables

*To whom correspondence should be addressed.

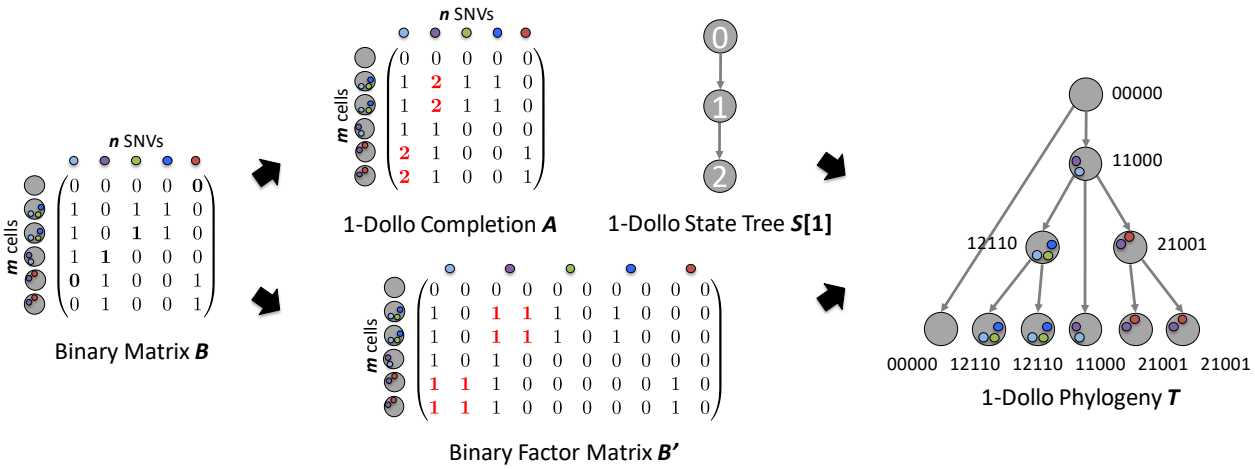

Figure A1: **Main concepts.** Binary matrix B is a 1-Dollo phylogeny matrix because there exists a 1-Dollo completion A without any forbidden submatrices (Table A1). From A and the 1-Dollo state tree $S[1]$, we can obtain the 1-Dollo phylogeny T by constructing the binary factor matrix B'.

A Supplementary Material

A.1 Results from the Main Text

We start by recapitulating the definitions and theorems from the main text. Fig. A1 illustrates the main concepts.

Definition 1. A *k-Dollo phylogeny* T is a rooted, node-labeled tree subject to the following conditions.

1. Each node v of T is labeled by a vector $\mathbf{b}_v \in \{0,1\}^n$.
2. The root r of T is labeled by vector $\mathbf{b}_r = [0,\ldots,0]^T$.
3. For each character $c \in [n]$, there is exactly one *gain edge* (v,w) in T such that $b_{v,c} = 0$ and $b_{w,c} = 1$.
4. For each character $c \in [n]$, there are at most k *loss edges* (v,w) in T such that $b_{v,c} = 1$ and $b_{w,c} = 0$.

k-Dollo Phylogeny problem (k-DP)**.** Given a binary matrix $B \in \{0,1\}^{m\times n}$ and parameter $k \in \mathbb{N}$, determine whether there exists a k-Dollo phylogeny for B, and if so construct one.

k-Dollo Phylogeny Flip and Cluster problem (k-DPFC)**.** Given matrix $D \in \{0,1,?\}^{m\times n}$, error rates $\alpha,\beta \in [0,1]$, integers $k,s,t \in \mathbb{N}$, find matrix $B \in \{0,1\}^{m\times n}$ and tree T such that: (1) B has at most s unique rows and at most t unique columns; (2) $\Pr(D \mid B,\alpha,\beta)$ is maximum; and (3) T is a k-Dollo phylogeny for B.

Definition 2 (Estabrook *et al.* (1975); Gusfield (1991))**.** A rooted, node-labeled tree T is a *perfect phylogeny* provided the following conditions hold.

1. Each node v of T is labeled by a vector $\mathbf{a}_v \in \{0,\ldots,k+1\}^n$.
2. The root r of T is labeled by vector $\mathbf{a}_r = [0,\ldots,0]^T$.
3. Nodes labeled with state i for character c form a connected subtree $T_{(c,i)}$ of T.

Theorem 1 (Perfect Phylogeny Theorem (Gusfield, 1991))**.** A binary matrix $A \in \{0,1\}^{m\times n}$ is a perfect phylogeny matrix if and only if no two columns of A contain the three pairs $(1,0)$; $(0,1)$ and $(1,1)$.

Definition 3 (Fernández-Baca (2000))**.** A *state tree* S is a rooted, node-labeled tree, whose root node is labeled by state 0, and whose other nodes are uniquely labeled by states $\{1,\ldots,k+1\}$.

Table A1: **There are 25 forbidden submatrices for $k = 1$.** Let $I^{(i)} = \{i, \ldots, k+1\}$. Here, $i_1, i'_1, j_1, j'_1 \in I^{(1)}$, $i_2, j_2 \in I^{(2)}$, $i''_1 \in I^{(1)} \setminus \{i_2\}$ and $j''_1 \in I^{(1)} \setminus \{j_2\}$.

case	forbidden matrices
$\begin{pmatrix} i_1 & 0 \\ 0 & j_1 \\ i'_1 & j'_1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{pmatrix}$ $\begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 2 \end{pmatrix}$ $\begin{pmatrix} 1 & 0 \\ 0 & 2 \\ 1 & 1 \end{pmatrix}$ $\begin{pmatrix} 1 & 0 \\ 0 & 2 \\ 1 & 2 \end{pmatrix}$ $\begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 2 & 1 \end{pmatrix}$ $\begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 2 & 2 \end{pmatrix}$ $\begin{pmatrix} 1 & 0 \\ 0 & 2 \\ 2 & 1 \end{pmatrix}$ $\begin{pmatrix} 1 & 0 \\ 0 & 2 \\ 2 & 2 \end{pmatrix}$ $\begin{pmatrix} 2 & 0 \\ 0 & 1 \\ 1 & 1 \end{pmatrix}$ $\begin{pmatrix} 2 & 0 \\ 0 & 1 \\ 1 & 2 \end{pmatrix}$ $\begin{pmatrix} 2 & 0 \\ 0 & 2 \\ 1 & 1 \end{pmatrix}$ $\begin{pmatrix} 2 & 0 \\ 0 & 2 \\ 1 & 2 \end{pmatrix}$ $\begin{pmatrix} 2 & 0 \\ 0 & 1 \\ 2 & 1 \end{pmatrix}$ $\begin{pmatrix} 2 & 0 \\ 0 & 1 \\ 2 & 2 \end{pmatrix}$ $\begin{pmatrix} 2 & 0 \\ 0 & 2 \\ 2 & 1 \end{pmatrix}$ $\begin{pmatrix} 2 & 0 \\ 0 & 2 \\ 2 & 2 \end{pmatrix}$
$\begin{pmatrix} i_1 & j''_1 \\ 0 & j_2 \\ i'_1 & j_2 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 \\ 0 & 2 \\ 1 & 2 \end{pmatrix}$ $\begin{pmatrix} 1 & 1 \\ 0 & 2 \\ 2 & 2 \end{pmatrix}$ $\begin{pmatrix} 2 & 1 \\ 0 & 2 \\ 1 & 2 \end{pmatrix}$ $\begin{pmatrix} 2 & 1 \\ 0 & 2 \\ 2 & 2 \end{pmatrix}$
$\begin{pmatrix} i_2 & 0 \\ i''_1 & j_1 \\ i_2 & j'_1 \end{pmatrix}$	$\begin{pmatrix} 2 & 0 \\ 1 & 1 \\ 2 & 1 \end{pmatrix}$ $\begin{pmatrix} 2 & 0 \\ 1 & 1 \\ 2 & 2 \end{pmatrix}$ $\begin{pmatrix} 2 & 0 \\ 1 & 2 \\ 2 & 1 \end{pmatrix}$ $\begin{pmatrix} 2 & 0 \\ 1 & 2 \\ 2 & 2 \end{pmatrix}$
$\begin{pmatrix} i_2 & j''_1 \\ i''_1 & j_2 \\ i_2 & j_2 \end{pmatrix}$	$\begin{pmatrix} 2 & 1 \\ 1 & 2 \\ 2 & 2 \end{pmatrix}$

Definition 4 (Fernández-Baca (2000))**.** Let $A \in \{0, \ldots, k+1\}^{m \times n}$ and let $\mathcal{S} = \{S_1, \ldots, S_n\}$ be a set of state trees for each character. The *binary factor matrix* $B' = [b'_{p,e}]$ of $(A, \mathcal{S})$ has dimensions $m \times n(k+1)$, and entries

$$b'_{p,e} = \begin{cases} 0, & \text{if } i \not\preceq_{S_c} a_{p,c}, \\ 1, & \text{if } i \preceq_{S_c} a_{p,c}. \end{cases} \tag{1}$$

where $c = \lfloor e/(k+1) \rfloor + 1$, $i = (e \bmod (k+1)) + 1$ and S_c is the state tree of character c.

Theorem 2 (Fernández-Baca (2000))**.** Matrix A has a perfect phylogeny consistent with states trees $\mathcal{S} = \{S_1, \ldots, S_n\}$ if and only if the binary factor matrix B' of $(A, \mathcal{S})$ is a perfect phylogeny matrix.

Definition 5. The *k-Dollo state tree* $S[k]$ is a state tree with nodes $\{0, \ldots, k+1\}$ and edges $\{(0,1)\} \cup \{(1,i) \mid i \in \{2, \ldots, k+1\}\}$.

Definition 6. Let $B \in \{0,1\}^{m \times n}$. Matrix $A \in \{0, \ldots, k+1\}^{m \times n}$ is a *k-completion of B* provided (1) $a_{p,c} \in \{0, \ldots, k+1\} \setminus \{1\}$ if and only if $b_{p,c} = 0$; and (2) $a_{p,c} = 1$ if and only if $b_{p,c} = 1$.

Definition 7. Let $I^{(i)} = \{i, \ldots, k+1\}$. Matrix $A \in \{0, \ldots, k+1\}^{m \times n}$ is a *k-Dollo completion* provided there exist no two columns and three rows in A of the following form:

$$\begin{pmatrix} i_1 & 0 \\ 0 & j_1 \\ i'_1 & j'_1 \end{pmatrix} \text{ or } \begin{pmatrix} i_1 & j''_1 \\ 0 & j_2 \\ i'_1 & j_2 \end{pmatrix} \text{ or } \begin{pmatrix} i_2 & 0 \\ i''_1 & j_1 \\ i_2 & j'_1 \end{pmatrix} \text{ or } \begin{pmatrix} i_2 & j''_1 \\ i''_1 & j_2 \\ i_2 & j_2 \end{pmatrix}$$

where $i_1, i'_1, j_1, j'_1 \in I^{(1)}$, $i_2, j_2 \in I^{(2)}$, $i''_1 \in I^{(1)} \setminus \{i_2\}$ and $j''_1 \in I^{(1)} \setminus \{j_2\}$.

Table A1 lists the forbidden submatrices for $k = 1$.

A.2 Combinatorial Characterization

Let $A \in \{0, \ldots, k+1\}^{m \times n}$ and let $\mathcal{S}[k]$ be a set comprised of n k-Dollo state trees. By Definition 5, we have that the binary factor matrix B' of $(A, \mathcal{S}[k])$ is defined as follows.

Definition 8. The *binary factor matrix* $B' = [b'_{p,e}]$ of $(A, \mathcal{S}[k])$ has dimensions $m \times n(k+1)$, and entries

$$b'_{p,e} = \begin{cases} 0, & \text{if } a_{p,c} = 0 \text{ and } i = 1 \\ 1, & \text{if } a_{p,c} \in \{1, \ldots, k+1\} \text{ and } i = 1, \\ 0, & \text{if } a_{p,c} \neq i \text{ and } i > 1, \\ 1, & \text{if } a_{p,c} = i \text{ and } i > 1, \end{cases} \tag{2}$$

where $c = \lfloor e/(k+1) \rfloor + 1$, $i = (e \bmod (k+1)) + 1$.

We now prove the key theorem that underlies SPHYR.

Theorem 3. Let $B \in \{0,1\}^{m \times n}$. The following statements are equivalent.

1. There exists a k-Dollo phylogeny T for B.
2. There exists a k-Dollo completion A of B.
3. There exists a k-completion A of B such that the binary factor matrix B' of $(A, \mathcal{S}[k])$ is a perfect phylogeny matrix.
4. There exists a k-completion A of B, and perfect phylogeny T for A whose characters are consistent with $S[k]$.

Proof. We proof the theorem by first proving that statement 1 implies statement 2. Then, we show that statement 2 implies statement 3. By Lemma 4.3 in (Fernández-Baca, 2000), we have that statement 3 implies statement 4. Finally, we prove that statement 4 implies statement 1.

$(1 \Rightarrow 2)$ Let T be a k-Dollo phylogeny for B. Recall that there exists a bijection between the leaves of T and the rows of B, and that each node v of T is labeled by binary vector $\mathbf{b}_v \in \{0,1\}^n$. We describe how to construct a matrix $A \in \{0, \ldots, k+1\}^{m \times n}$ from T. First, for each character $c \in [n]$, we identify all edges (v, w) of T where c is lost—i.e. (v, w) is a *loss edge* for c if $b_{v,c} = 1$ and $b_{w,c} = 0$. We number each loss edge (v, w) for c by $\sigma(v, w, c)$ starting from 1.

Consider row vector $\mathbf{b}_p$ of B and its corresponding leaf v_p of T. We define the corresponding row vector $\mathbf{a}_p = [a_{p,c}]$ of A by considering each character $c \in [n]$. If $b_{p,c} = 1$, we set $a_{p,c} = 1$. If $b_{p,c} = 0$ and there exists a loss edge (v, w) for c on the unique path from v_p to the root r of T, we set $a_{p,c} = \sigma(v, w, c) + 1$. Otherwise, we set $a_{p,c} = 0$. By definition of T, there are at most k loss edges for each character c. Each entry $a_{p,c}$ thus has a integer value in $\{0, \ldots, k+1\}$. Hence, matrix A is a k-completion.

Now, assume for a contradiction that A is not a valid k-completion of B. Thus, there exist two characters (columns) c, d and three leaves (rows) u, v, w containing three forbidden pairs (Definition 7). We distinguish four cases.

1. There exist $i_1, i'_1, j_1, j'_1 \in I^{(1)}$ such that

$$\begin{pmatrix} a_{u,c} & a_{u,d} \\ a_{v,c} & a_{v,d} \\ a_{w,c} & a_{w,d} \end{pmatrix} = \begin{pmatrix} i_1 & 0 \\ 0 & j_1 \\ i'_1 & j'_1 \end{pmatrix}. \tag{3}$$

We focus on edges $e_{(c,1)}$ and $e_{(d,1)}$. There are three subcases.

(a) Edge $e_{(c,1)}$ precedes edge $e_{(d,1)}$, i.e. $(c,1) \preceq_T (d,1)$:
Leaf v has character states $(c,0)$ and (d,j_1). As T is a k-Dollo phylogeny, we have that $(d,1) \preceq_T (d,j_1) \preceq_T v$. This means that $(c,1) \preceq_T (c,0)$, which yields a contradiction.

(b) Edge $e_{(d,1)}$ precedes edge $e_{(c,1)}$, i.e. $(d,1) \preceq_T (c,1)$:
Leaf u has character states (c,i_1) and $(d,0)$. As T is a k-Dollo phylogeny, we have that $(c,1) \preceq_T (c,i_1) \preceq_T u$. This means that $(d,1) \preceq_T (d,0)$, which yields a contradiction.

(c) Edges $e_{(c,1)}$ and $e_{(d,1)}$ occur on distinct branches:
As $i'_1, j'_1 \geq 1$, we have that $(c,1) \preceq_T (c,i'_1)$ and $(d,1) \preceq_T (d,j'_1)$. Moreover, leaf w has character states (c,i'_1) and (d,j'_1), and thus it holds that $(c,i'_1) \preceq_T w$ and $(d,j'_1) \preceq_T w$. This, however contradicts the premise that edges $e_{(c,1)}$ and $e_{(d,1)}$ occur on distinct branches.

2. There exist $i_1, i'_1 \in I^{(1)}, j_2 \in I^{(2)}, j''_1 \in I^{(1)} \setminus \{j_2\}$ such that

$$\begin{pmatrix} a_{u,c} & a_{u,d} \\ a_{v,c} & a_{v,d} \\ a_{w,c} & a_{w,d} \end{pmatrix} = \begin{pmatrix} i_1 & j''_1 \\ 0 & j_2 \\ i'_1 & j_2 \end{pmatrix}. \tag{4}$$

We focus on edges $e_{(c,1)}$ and $e_{(d,j_2)}$. There are three subcases.

(a) Edge $e_{(c,1)}$ precedes edge $e_{(d,j_2)}$, i.e. $(c,1) \preceq_T (d,j_2)$:
Leaf v has character states $(c,0)$ and (d,j_2). This means that $(c,1) \preceq_T (c,0)$, which yields a contradiction.

(b) Edge $e_{(d,j_2)}$ precedes edge $e_{(c,1)}$, i.e. $(d,j_2) \preceq_T (c,1)$:
By definition we have that $(c,1) \preceq_T (c,i_1)$ and $(c,1) \preceq_T (c,i'_1)$. Thus the two paths from the root to leaves u and w share the edges $e_{(d,j_2)}$ and $e_{(c,1)}$. Now, leaf u has character state (d,j''_1) and thus we have that $(d,j_2) \preceq_T (d,j''_1)$. This means that path from the root to leaf u contains either two distinct loss edges (if $j''_1 \geq 2$) or a gain after a loss (if $j''_1 = 1$) for character d. Both cases yield a contradiction.

(c) Edges $e_{(c,1)}$ and $e_{(d,j_2)}$ occur on distinct branches:
As $i''_1 \in I^{(1)} \setminus \{i_2\}$, we have $(c,1) \preceq_T (c,i''_1)$. Leaf w has character states (c,i''_1) and (d,j_2). Thus, it holds that $(c,1) \preceq_T (c,i''_1) \preceq_T w$ and $(d,j_2) \preceq_T w$. This, however contradicts the premise that edges $e_{(c,1)}$ and $e_{(d,j_2)}$ occur on distinct branches.

3. There exist $i_2 \in I^{(2)}, i''_1 \in I^{(1)} \setminus \{i_2\}, j_1, j'_1 \in I^{(1)}$ such that

$$\begin{pmatrix} a_{u,c} & a_{u,d} \\ a_{v,c} & a_{v,d} \\ a_{w,c} & a_{w,d} \end{pmatrix} = \begin{pmatrix} i_2 & 0 \\ i''_1 & j_1 \\ i_2 & j'_1 \end{pmatrix}. \tag{5}$$

We focus on edges $e_{(c,i_2)}$ and $e_{(d,1)}$. There are four subcases.

(a) Edge $e_{(c,i_2)}$ precedes edge $e_{(d,1)}$, i.e. $(c,i_2) \preceq_T (d,1)$:
As $i_2 \in I^{(2)}$, we have that $(c,1) \preceq_T (c,i_2)$. Leaf v has character states (c,i''_1) and (d,j_1). This means that $(c,i''_1) \preceq_T w$ and $(d,j_1) \preceq_T w$. As $(c,i_2) \preceq_T (d,1) \preceq_T (d,j_1)$ and $i_2 \neq i''_1$, we have that the path from the root to w contains either two distinct loss edges (if $i''_1 \geq 2$) or a gain after a loss (if $i''_1 = 1$) for character c. Both cases yield a contradiction.

(b) Edge $e_{(d,1)}$ precedes edge $e_{(c,i_2)}$, i.e. $(d,1) \preceq_T (c,i_2)$:
Leaf u has character states (c,i_2) and $(d,0)$. This means that $(d,1) \preceq_T (d,0)$, which yields a contradiction.

(c) Edges $e_{(c,i_2)}$ and $e_{(d,1)}$ occur on distinct branches:
As $j_1'' \in I^{(1)} \setminus \{j_2\}$, we have that $(d,1) \preceq_T (d,j_1'')$. Leaf w has character states (c,i_2) and (d,j_1''). Thus, it holds that $(c,i_2) \preceq_T w$ and $(d,1) \preceq_T (d,j_1'') \preceq_T w$. This contradicts the premise that edges $e_{(c,i_2)}$ and $e_{(d,1)}$ occur on distinct branches.

4. There exist $i_2 \in I^{(2)}, i_1'' \in I^{(1)} \setminus \{i_2\}, j_2 \in I^{(2)}, j_1'' \in I^{(1)} \setminus \{j_2\}$ such that

$$\begin{pmatrix} a_{u,c} & a_{u,d} \\ a_{v,c} & a_{v,d} \\ a_{w,c} & a_{w,d} \end{pmatrix} = \begin{pmatrix} i_2 & j_1'' \\ i_1'' & j_2 \\ i_2 & j_2 \end{pmatrix}. \tag{6}$$

We focus on edges $e_{(c,i_2)}$ and $e_{(d,j_2)}$. There are three subcases.

(a) Edge $e_{(c,i_2)}$ precedes edge $e_{(d,j_2)}$, i.e. $(c,i_2) \preceq_T (d,j_2)$:
Leaf v has character states (c,i_1'') and (d,j_2). This means that $(c,i_1'') \preceq_T v$ and $(c,i_2) \preceq_T (d,j_2) \preceq_T v$. As $i_2 \in I^{(2)}, i_1'' \in I^{(1)} \setminus \{i_2\}$, we have that the path from the root to v contains either two distinct loss edges (if $j_1'' \geq 2$) or a gain after a loss (if $j_1'' = 1$) for character d. Both cases yield a contradiction.

(b) Edge $e_{(d,j_2)}$ precedes edge $e_{(c,i_2)}$, i.e. $(d,j_2) \preceq_T (c,i_2)$:
Leaf u has character states (c,i_2) and (d,j_1''). This means that $(d,j_2) \preceq_T (c,i_2) \preceq_T v$ and $(d,j_1'') \preceq_T u$. As $j \in I^{(2)}, j_1'' \in I^{(1)} \setminus \{j_2\}$, we have that the path from the root to u contains either two distinct loss edges (if $j_1'' \geq 2$) or a gain after a loss (if $j_1'' = 1$) for character d. Both cases yield a contradiction.

(c) Edges $e_{(c,i_2)}$ and $e_{(d,j_2)}$ occur on distinct branches:
Leaf w has character states (c,i_2) and (d,j_2). Thus, $(c,i_2) \preceq_T w$ and $(d,j_2) \preceq_T w$, which contradicts the premise.

Each case results in a contradiction, thus A must be a k-Dollo completion of B.

$(2 \Rightarrow 3)$ Let A be a k-Dollo completion of B, and let B' be the binary factor matrix of A. Assume for a contradiction that B' is not a perfect phylogeny matrix. Thus, by Theorem 1, there exist three taxa $u, v, w \in [m]$ and two characters $e, f \in [n(k+1)]$ such that

$$\begin{pmatrix} b'_{u,e} & b'_{u,f} \\ b'_{v,e} & b'_{v,f} \\ b'_{w,e} & b'_{w,f} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{pmatrix}. \tag{7}$$

Let $c = \lfloor e/(k+1) \rfloor + 1$, $\phi = (e \bmod (k+1)) + 1$, $d = \lfloor f/(k+1) \rfloor + 1$ and $\psi = (f \bmod (k+1)) + 1$. We distinguish four cases.

1. $\phi = 1$ and $\psi = 1$: This means that matrix A contains the following submatrix.

$$\begin{pmatrix} a_{u,c} & a_{u,d} \\ a_{v,c} & a_{v,d} \\ a_{w,c} & a_{w,d} \end{pmatrix} = \begin{pmatrix} i_1 & 0 \\ 0 & j_1 \\ i_1' & j_1' \end{pmatrix}, \tag{8}$$

where $i_1, i_1', j_1, j_1' \in I^{(1)}$. Thus, matrix A violates the first condition of Definition 7. Hence, matrix A is not a k-Dollo completion, which contradicts the premise.

2. $\phi = 1$ and $\psi > 1$: This means that matrix A contains the following submatrix.

$$\begin{pmatrix} a_{u,c} & a_{u,d} \\ a_{v,c} & a_{v,d} \\ a_{w,c} & a_{w,d} \end{pmatrix} = \begin{pmatrix} i_1 & j' \\ 0 & j_2 \\ i'_1 & j_2 \end{pmatrix}, \tag{9}$$

where $i_1, i'_1 \in I^{(1)}, j_2 \in I^{(2)}, j' \in I \setminus \{j_2\}$. If $j' = 0$ then matrix A violates the first condition of Definition 7. If $j' \neq 0$ then the second condition of Definition 7 is violated. Hence, matrix A is not a k-Dollo completion, which contradicts the premise.

3. $\phi > 1$ and $\psi = 1$: This means that matrix A contains the following submatrix.

$$\begin{pmatrix} a_{u,c} & a_{u,d} \\ a_{v,c} & a_{v,d} \\ a_{w,c} & a_{w,d} \end{pmatrix} = \begin{pmatrix} i_2 & 0 \\ i' & j_1 \\ i_2 & j'_1 \end{pmatrix}, \tag{10}$$

where $i_2 \in I^{(2)}, i' \in I \setminus \{i_2\}, j_1, j'_1 \in I^{(1)}$. If $i' = 0$ then condition 1 of Definition 7 is violated. On the other hand, of $i' \neq 0$ then matrix A violates the third condition of Definition 7. Hence, matrix A is not a k-Dollo completion, which contradicts the premise.

4. $\phi > 1$ and $\psi > 1$: This means that matrix A contains the following submatrix.

$$\begin{pmatrix} a_{u,c} & a_{u,d} \\ a_{v,c} & a_{v,d} \\ a_{w,c} & a_{w,d} \end{pmatrix} = \begin{pmatrix} i_2 & j' \\ i' & j_2 \\ i_2 & j_2 \end{pmatrix}, \tag{11}$$

where $i, j \in I^{(2)}, i' \in I \setminus \{i_2\}, j' \in I \setminus \{j_2\}$. If $i' = 0$ and $j' = 0$ then condition 1 of Definition 7 is violated. If $i' = 0$ and $j' \neq 0$ then condition 2 of Definition 7 is violated. If $i' \neq 0$ and $j' = 0$ then condition 3 of Definition 7 is violated. If $i' \neq 0$ and $j' \neq 0$ then condition 4 of Definition 7 is violated. Hence, matrix A is not a k-Dollo completion, which contradicts the premise.

Each case results in a contradiction, thus the binary factor matrix B' is a perfect phylogeny matrix. This means that the binary factor matrix of a k-Dollo completion is a perfect phylogeny matrix.

$(3 \Rightarrow 4)$ This direction follows from Theorem 2 by Fernández-Baca (2000), which states that a matrix $A \in \{0, \ldots, k+1\}^{m \times n}$ has a perfect phylogeny consistent with states trees $\mathcal{S} = \{S_1, \ldots, S_n\}$ if and only if the binary factor matrix B of $(A, \mathcal{S})$ is a perfect phylogeny matrix.

$(4 \Rightarrow 1)$ Let A be a k-completion of B. Let T be a perfect phylogeny for A whose characters are consistent with $S[k]$. Let $\mathbf{a}_v = [a_{v,c}]$ be the vector associated with each node v of T. For each node v of T, we define the vector $\mathbf{b}_v = [b_{v,c}]$ as follows.

$$b_{v,c} = \begin{cases} 0, & \text{if } a_{v,c} = 0, \\ 1, & \text{if } a_{v,c} = 1, \\ 0, & \text{if } a_{v,c} > 1. \end{cases} \tag{12}$$

We claim that the tree T, where each node v is labeled by $\mathbf{b}_v$ is a k-Dollo phylogeny. For the root node r of T, we have that $a_{r,c} = 0$ for each character $c \in [n]$. Thus, by definition $b_{r,c} = 0$ for each character c. Consider a character $c \in [n]$. Since each node v with $a_{v,c} = b_{v,c} = 1$ forms a connected subtree of T and c is consistent with $S[k]$, we have that there is exactly one edge (v, w) in T such that $a_{v,c} = b_{v,c} = 0$ and $a_{v,c} = b_{w,c} = 1$. By construction c has at most k loss states, numbered $I^{(2)} = \{2, \ldots, k+1\}$. Again, by consistency of T with $S[k]$ and the fact that for all states $i \in I^{(2)}$ each node v with $a_{v,c} = i$ forms a connected subtree of T, there exist at most k edges (v, w) in T such that $a_{v,c} = 1$ and $a_{v,c} = i$ for some $i \in I^{(2)}$. Hence, there are at most k edges (v, w) in T such that $b_{v,c} = 1$ and $b_{v,c} = 0$. This proves that the tree T whose nodes v are labeled by $\mathbf{b}_v$ is a k-Dollo phylogeny. □

A.3 Column Generation for k-DP and k-DPFC

In this section, we provide additional implementation details. For both the k-DP and the k-DPFC problem, we preprocess the input matrix $D \in \{0, 1, ?\}^{m \times n}$ and remove characters (columns) and taxa (rows) of the following form.

1. Characters c such that $d_{p,c} = 0$ or $d_{p,c} =?$ for all taxa $p \in [m]$.
2. Characters c such that there exists exactly one taxon $p \in [m]$ where $d_{p,c} = 1$ and $d_{q,c} \in \{0, ?\}$ for all taxa $q \in [m] \setminus \{p\}$.
3. Characters c such that $d_{p,c} = 1$ or $d_{p,c} =?$ for all taxa $p \in [m]$.
4. Taxa p such that $d_{p,c} = 0$ or $d_{p,c} =?$ for all characters $c \in [n]$.

These taxa and characters can be safely removed due to the fact that $\alpha < 0.5$ and $\beta < 0.5$, and that the corresponding columns and rows do not contribute to conflicts. It is not hard to show that there exist optimal solutions B^* to the k-DPFC problem where identical columns and rows of input matrix D are identical in B^* as well. Thus, we remove repeated rows and columns from $D = [d_{p,c}]$ yielding $D' = [d'_{p,c}]$ and include a multiplicative factor in the objective function that accounts for the number of entries in D that corespond to each entry $d'_{p,c}$.

We now provide additional details for the column generation procedure used for solving the k-DP problem. As described in the main text, the ILP is as follows:

$$\min \sum_{p=1}^{m} \sum_{c=1}^{n} \sum_{i=2}^{k+1} a_{p,c,i} \left(\frac{1}{mn}\right)^{k+1-i} \tag{13}$$

$$\text{s.t. } a_{p,c,i} \in \{0,1\} \qquad \forall p \in [m], c \in [n], i \in \{0, \ldots, k+1\} \tag{14}$$

$$\sum_{i=0}^{k+1} a_{p,c,i} = 1 \qquad \forall p \in [m], c \in [n] \tag{15}$$

$$a_{p,c,1} = 0 \qquad \forall p \in [m], c \in [n] \text{ s.t. } b_{p,c} = 0 \tag{16}$$

$$a_{p,c,1} = 1 \qquad \forall p \in [m], c \in [n] \text{ s.t. } b_{p,c} = 1 \tag{17}$$

$$\sum_{p=1}^{m} \sum_{c=1}^{n} a_{p,c,i} \geq \sum_{p=1}^{m} \sum_{c=1}^{n} a_{p,c,i-1} \qquad \forall i \in \{3, \ldots, k+1\} \tag{18}$$

$$a_{p,d,0} + a_{q,c,0} + a_{q,d,j_1} + a_{r,c,i'_1} + a_{r,d,j'_1} \leq 5 \tag{19}$$

$$a_{p,c,i_1} + a_{p,d,j''_1} + a_{q,c,0} + a_{q,d,j_2} + a_{r,c,i'_1} + a_{r,d,j_2} \leq 5 \tag{20}$$

$$a_{p,c,i_2} + a_{p,d,0} + a_{q,c,i''_1} + a_{q,d,j_1} + a_{r,c,i_2} + a_{r,d,j'_1} \leq 5 \tag{21}$$

$$a_{p,c,i_2} + a_{p,d,j''_1} + a_{q,c,i''_1} + a_{q,d,j_2} + a_{r,c,i_2} + a_{r,d,j_2} \leq 5 \tag{22}$$

Algorithm 1 provides pseudocode for the overall procedure, and invokes functions SEPARATE1 (Algorithm 2), SEPARATE2 (Algorithm 3), SEPARATE3 (Algorithm 4) and SEPARATE4 (Algorithm 5). In the main text, we describe that separation proceeds in $O(mk^3)$ time for each pair c, d of distinct characters. This time bound can be achieved by considering only a single element of each set P, Q and R. In practice, however, considering all elements of these sets considerably strengthens the formulation and leads to better performance. Doing so leads to output-sensitive asymptotic run times of $O(mn^2 + |\mathcal{C}'|)$ for SEPARATE1, and $O(mn^2k^3 + |\mathcal{C}'|)$ for SEPARATE2, SEPARATE3 and SEPARATE4.

The ILP for the k-DPFC problem is as follows.

$$\min \sum_{\substack{(p,c)\in X:\\ d_{p,c}=0}} \left[a_{\pi(p),\psi(c),1}\log\beta + (1-a_{\pi(p),\psi(c),1})\log(1-\beta)\right] + \sum_{\substack{(p,c)\in X:\\ d_{p,c}=1}} \left[a_{\pi(p),\psi(c),1}\log(1-\alpha) + (1-a_{\pi(p),\psi(c),1})\log(\alpha)\right] \tag{23}$$

$$\text{s.t. } a_{h,f,i} \in \{0,1\} \qquad \forall h \in [s], f \in [t], i \in \{0,\ldots,k+1\} \tag{24}$$

$$\sum_{i=0}^{k+1} a_{h,f,i} = 1 \qquad \forall h \in [s], f \in [t] \tag{25}$$

$(18) - (22)$

Algorithm 6 provides pseudocode for the column generation procedure used for solving a variant of the k-DPFC problem, where one is given a row clustering and column clustering. This procedure uses the same separation functions as in Algorithm 1. In contrast to the previous algorithm, a feasible solution always exists for the k-DPFC problem, as 1-entries of the input matrix might be edited. As such, the ILP solver will never determine that the model is infeasible.

A.4 Simulation Results for k-DP

Table A2 shows additional statistics of SPHYR's performance on the simulated k-DP instances.

A.5 Simulation Results for k-DPFC

We use default arguments for SiFit with 100 restarts and 10,000 MCMC iterations for each restart. That is,

```
java -jar SiFit.jar -r 100 -m $m$ -n $n$ -fp $\alpha$ -fn $\beta$ \
  -iter 10000 -df 0 -ipMat <INPUT>
```

The above command produces an output tree in NEWICK format called `<INPUT>_mlTree.newick`, and also infers a false negative rate FN, loss-of-heterozygosity rate LOH and deletion rate del. To infer the vertex labeling, we use

```
java -cp SiFit.jar SiFit.algorithm.InferAncestralStates \
  -fp $\alpha$ -fn FN -w LOH -d del -df 0 -ipMat <INPUT> \
  -tree <INPUT>_mlTree.newick -geneNames <GENE_LABELS> \
  -cellNames <CELL_LABELS> -expectedMatrix <INPUT>.leaves
```

We use default arguments for SCITE with 100 restarts and 1000000 MCMC iterations for each restart. That is,

```
scite -i <INPUT> -r 100 -l 1000000 -a -m $m$ -n $n$ \
  -fd $\alpha$ -ad $\beta$ -o <OUTPUT> -e 0.1
```

We have the following figures, where we consider the effect of varying $\alpha \in \{0.0001, 0.001, 0.01\}$ and $\beta \in \{0.1, 0.2, 0.3\}$.

Algorithm 1: k-DP(B, k)

Input: Input matrix $B \in \{0,1\}^{m \times n}$ and natural number k
Output: k-Dollo completion $A \in \{0, \ldots, k+1\}^{m \times n}$ of B, if one exists

Let $\mathcal{C}$ be comprised of (15) − (18)
Set objective function to (13)
for $p \leftarrow 1$ **to** m **do**
    for $c \leftarrow 1$ **to** n **do**
        for $i \leftarrow 0$ **to** $k+1$ **do**
            if $b_{p,c} = i$ **then**
                $\mathcal{C} \leftarrow \mathcal{C} \cup \{a_{p,c,i} \in \{0,1\}\}$
            else
                $\mathcal{C} \leftarrow \mathcal{C} \cup \{a_{p,c,i} \in \{0\}\}$
$\mathcal{C}' \leftarrow \emptyset$
repeat
    Solve ILP
    if *ILP is infeasible* **then return** INFEASIBLE
    Let A be the ILP solution
    $\mathcal{C}' \leftarrow$ SEPARATE1$(A, k) \cup$ SEPARATE2$(A, k) \cup$ SEPARATE3$(A, k) \cup$ SEPARATE4(A, k)
    foreach *variable* $a_{p,c,i} \in \mathcal{C}'$ **do**
        if $i = 0$ **then**
            Extend domain of $a_{p,c,2}$ in $\mathcal{C}$ to $\{0,1\}$
        if $2 \leq i < k+1$ **then**
            Extend domain of $a_{p,c,i+1}$ in $\mathcal{C}$ to $\{0,1\}$
    $\mathcal{C} \leftarrow \mathcal{C} \cup \mathcal{C}'$
until $\mathcal{C}' = \emptyset$
return A

Algorithm 2: SEPARATE1(A, k)

Input: Matrix $A \in \{0, \ldots, k+1\}^{m \times n}$ and natural number k
Output: Set $\mathcal{C}'$ of violated constraints of the form (19)

$\mathcal{C}' \leftarrow \emptyset$
for $c \leftarrow 1$ **to** n **do**
  for $d \leftarrow c+1$ **to** n **do**
    // Condition 1
    $P \leftarrow \emptyset$
    for $p \leftarrow 1$ **to** m **do**
      foreach $i_1 \in I^{(1)}$ **do**
        if $a_{p,c,i_1} = 1$ *and* $a_{p,d,0} = 1$ **then** $P \leftarrow P \cup \{(a_{p,c,i_1}, a_{p,d,0})\}$
    $Q \leftarrow \emptyset$
    for $q \leftarrow 1$ **to** m **do**
      foreach $j_1 \in I^{(1)}$ **do**
        if $a_{q,c,0} = 1$ *and* $a_{q,d,j_1} = 1$ **then** $Q \leftarrow Q \cup \{(a_{q,c,0}, a_{q,d,j_1})\}$
    $R \leftarrow \emptyset$
    for $r \leftarrow 1$ **to** m **do**
      foreach $(i'_1, j'_1) \in I^{(1)} \times I^{(1)}$ **do**
        if $a_{r,c,i'_1} = 1$ *and* $a_{r,d,j'_1} = 1$ **then** $R \leftarrow R \cup \{(a_{r,c,i'_1}, a_{r,d,j'_1})\}$
    foreach $(a_{p,c,i_1}, a_{p,d,0}) \in P$ **do**
      foreach $(a_{q,c,0}, a_{q,d,j_1}) \in Q$ **do**
        foreach $(a_{r,c,i'_1}, a_{r,d,j'_1}) \in R$ **do**
          $\mathcal{C}' \leftarrow \mathcal{C}' \cup \{a_{p,c,i_1} + a_{p,d,0} + a_{q,c,0} + a_{q,d,j_1} + a_{r,c,i'_1} + a_{r,d,j'_1} \leq 5\}$
return $\mathcal{C}'$

Algorithm 3: SEPARATE2(A, k)

Input: Matrix $A \in \{0, \ldots, k+1\}^{m \times n}$ and natural number k
Output: Set $\mathcal{C}'$ of violated constraints of the form (20)

$\mathcal{C}' \leftarrow \emptyset$
for $c \leftarrow 1$ **to** n **do**
  for $d \leftarrow c+1$ **to** n **do**
    for $j_2 \in I^{(2)}$ **do**
      `// Condition 2`
      $P \leftarrow \emptyset$
      for $p \leftarrow 1$ **to** m **do**
        foreach $(i_1, j_1'') \in I^{(1)} \times (I^{(1)} \setminus \{j_2\})$ **do**
          if $a_{p,c,i_1} = 1$ *and* $a_{p,d,j_1''} = 1$ **then** $P \leftarrow P \cup \{(a_{p,c,i_1}, a_{p,d,j_1''})\}$
      $Q \leftarrow \emptyset$
      for $q \leftarrow 1$ **to** m **do**
        if $a_{q,c,0} = 1$ *and* $a_{q,d,j_2} = 1$ **then** $Q \leftarrow Q \cup \{(a_{q,c,0}, a_{q,d,j_2})\}$
      $R \leftarrow \emptyset$
      for $r \leftarrow 1$ **to** m **do**
        foreach $i_1' \in I^{(1)}$ **do**
          if $a_{r,c,i_1'} = 1$ *and* $a_{r,d,j_2} = 1$ **then** $R \leftarrow R \cup \{(a_{r,c,i_1'}, a_{r,d,j_2})\}$
      foreach $(a_{p,c,i_1}, a_{p,d,j_1''}) \in P$ **do**
        foreach $(a_{q,c,0}, a_{q,d,j_2}) \in Q$ **do**
          foreach $(a_{r,c,i_1'}, a_{r,d,j_2}) \in R$ **do**
            $\mathcal{C}' \leftarrow \mathcal{C}' \cup \{a_{p,c,i_1} + a_{p,d,j_1''} + a_{q,c,0} + a_{q,d,j_2} + a_{r,c,i_1'} + a_{r,d,j_2} \leq 5\}$
return $\mathcal{C}'$

Algorithm 4: SEPARATE3(A, k)

Input: Matrix $A \in \{0, \ldots, k+1\}^{m \times n}$ and natural number k
Output: Set $\mathcal{C}'$ of violated constraints of the form (21)

$\mathcal{C}' \leftarrow \emptyset$
for $c \leftarrow 1$ **to** n **do**
  for $d \leftarrow c+1$ **to** n **do**
    for $i_2 \in I^{(2)}$ **do**
      `// Condition 3`
      $P \leftarrow \emptyset$
      for $p \leftarrow 1$ **to** m **do**
        if $a_{p,c,i_2} = 1$ *and* $a_{p,d,0} = 1$ **then** $P \leftarrow P \cup \{(a_{p,c,i_2}, a_{p,d,0})\}$
      $Q \leftarrow \emptyset$
      for $q \leftarrow 1$ **to** m **do**
        foreach $(i_1'', j_1) \in (I^{(1)} \setminus \{i_2\}) \times I^{(1)}$ **do**
          if $a_{q,c,i_1''} = 1$ *and* $a_{q,d,j_1} = 1$ **then** $Q \leftarrow Q \cup \{(a_{q,c,i_1''}, a_{q,d,j_1})\}$
      $R \leftarrow \emptyset$
      for $r \leftarrow 1$ **to** m **do**
        foreach $j_1' \in I^{(1)}$ **do**
          if $a_{r,c,i_2} = 1$ *and* $a_{r,d,j_1'} = 1$ **then** $R \leftarrow R \cup \{(a_{r,c,i_2}, a_{r,d,j_1'})\}$
      foreach $(a_{p,c,i_2}, a_{p,d,0}) \in P$ **do**
        foreach $(a_{q,c,i_1''}, a_{q,d,j_1}) \in Q$ **do**
          foreach $(a_{r,c,i_2}, a_{r,d,j_1'}) \in R$ **do**
            $\mathcal{C}' \leftarrow \mathcal{C}' \cup \{a_{p,c,i_2} + a_{p,d,0}) + a_{q,c,i_1''} + a_{q,d,j_1}) + a_{r,c,i_2} + a_{r,d,j_1'} \leq 5\}$
return $\mathcal{C}'$

Algorithm 5: SEPARATE4(A, k)

Input: Matrix $A \in \{0, \dots, k+1\}^{m \times n}$ and natural number k
Output: Set $\mathcal{C}'$ of violated constraints of the form (22)

$\mathcal{C}' \leftarrow \emptyset$
for $c \leftarrow 1$ **to** n **do**
  for $d \leftarrow c+1$ **to** n **do**
    for $i_2 \in I^{(2)}$ **do**
      for $j_2 \in I^{(2)}$ **do**
        // Condition 4
        $P \leftarrow \emptyset$
        for $p \leftarrow 1$ **to** m **do**
          foreach $j_1'' \in I^{(1)} \setminus \{j_2\}$ **do**
            if $a_{p,c,i_2} = 1$ *and* $a_{p,d,j_1''} = 1$ **then** $P \leftarrow P \cup \{(a_{p,c,i_2}, a_{p,d,j_1''})\}$
        $Q \leftarrow \emptyset$
        for $q \leftarrow 1$ **to** m **do**
          foreach $i_1'' \in (I^{(1)} \setminus \{i_2\})$ **do**
            if $a_{q,c,i_1''} = 1$ *and* $a_{q,d,j_2} = 1$ **then** $Q \leftarrow Q \cup \{(a_{q,c,i_1''}, a_{q,d,j_2})\}$
        $R \leftarrow \emptyset$
        for $r \leftarrow 1$ **to** m **do**
          if $a_{r,c,i_2} = 1$ *and* $a_{r,d,j_2} = 1$ **then** $R \leftarrow R \cup \{(a_{r,c,i_2}, a_{r,d,j_2})\}$
        foreach $(a_{p,c,i_2}, a_{p,d,j_1''}) \in P$ **do**
          foreach $(a_{q,c,i_1''}, a_{q,d,j_2}) \in Q$ **do**
            foreach $(a_{r,c,i_2}, a_{r,d,j_2}) \in R$ **do**
              $\mathcal{C}' \leftarrow \mathcal{C}' \cup \{a_{p,c,i_2} + a_{p,d,j_1''} + a_{q,c,i_1''} + a_{q,d,j_w}) + a_{r,c,i_2} + a_{r,d,j_2} \leq 5\}$
return $\mathcal{C}'$

Algorithm 6: SOLVEAB$(D, \alpha, \beta, s, t, k, \pi, \psi)$

Input: Input matrix $D \in \{0, 1, ?\}^{m \times n}$, a false positive rate $\alpha \in [0, 1]$, a false negative rate $\beta \in [0, 1]$, natural numbers k, s, t, row clustering $\pi : [m] \to [s]$ and column clustering $\psi : [n] \to [t]$

Output: k-Dollo completion $A \in \{0, \ldots, k+1\}^{s \times t}$ with maximum likelihood $\sum_{p=1}^{m} \sum_{c=1}^{n} \log \Pr(d_{p,c} \mid b_{\pi(p),\psi(c)}, \alpha, \beta)$

Let $\mathcal{C}$ be comprised of (18), (24) and (25)
Set objective function to (23)
Let $\pi^{-1}(h) = \{p \in [m] \mid \pi(m) = h\}$ for each $h \in [s]$
Let $\psi^{-1}(f) = \{c \in [n] \mid \psi(c) = f\}$ for each $f \in [t]$
for $h \leftarrow 1$ **to** s **do**
    for $f \leftarrow 1$ **to** t **do**
        $L_0 \leftarrow 0$
        $L_1 \leftarrow 0$
        foreach $p \in \pi^{-1}(h)$ **do**
            foreach $c \in \psi^{-1}(f)$ **do**
                if $d_{p,c} = 0$ **then**
                    $L_0 \leftarrow L_0 + \log(1 - \beta)$
                    $L_1 \leftarrow L_1 + \log \beta$
                else if $d_{p,c} = 1$ **then**
                    $L_0 \leftarrow L_0 + \log \alpha$
                    $L_1 \leftarrow L_1 + \log(1 - \alpha)$
        if $L_0 > L_1$ **then**
            $\mathcal{C} \leftarrow \mathcal{C} \cup \{a_{h,f,1} \in \{0\}\}$
            foreach $j \in \{0, 2, \ldots, k+1\}$ **do**
                $\mathcal{C} \leftarrow \mathcal{C} \cup \{a_{h,f,j} \in \{0, 1\}\}$
        else
            $\mathcal{C} \leftarrow \mathcal{C} \cup \{a_{h,f,1} \in \{0, 1\}\}$
            foreach $j \in \{0, 2, \ldots, k+1\}$ **do**
                $\mathcal{C} \leftarrow \mathcal{C} \cup \{a_{h,f,j} \in \{0\}\}$
$\mathcal{C}' \leftarrow \emptyset$
repeat
    Solve ILP
    Let A be the ILP solution
    $\mathcal{C}' \leftarrow$ SEPARATE1$(A, k) \cup$ SEPARATE2$(A, k) \cup$ SEPARATE3$(A, k) \cup$ SEPARATE4(A, k)
    foreach *variable* $a_{p,c,i} \in \mathcal{C}'$ **do**
        if $i = 1$ **then**
            Extend domain of $a_{p,c,j}$ in $\mathcal{C}$ to $\{0, 1\}$ for each $j \in \{0, 2, \ldots, k+1\}$
        else
            Extend domain of $a_{p,c,1}$ in $\mathcal{C}$ to $\{0, 1\}$
    $\mathcal{C} \leftarrow \mathcal{C} \cup \mathcal{C}'$
until $\mathcal{C}' = \emptyset$
return A

Table A2: **Simulation results for k-DP instances with varying number m of taxa, number n of characters and maximum number k of losses.** For each combination of m, n and k, we simulated 60 instances with varying loss rates $\{0.1, 0.2, 0.4\}$. See the main text for simulation details. For each instance, we show the percentage of instances solved to optimality (within 5 hours), the median run time in seconds, the median number of iterations of the column generation procedure, the median percentage of variables that were included in the model and the median percentage of constraints that were included in the model.

m	k	$n = 25$					$n = 50$					$n = 100$				
		solved	sec	#it.	vars.	cons.	solved	sec	#it.	vars.	cons.	solved	sec	#it.	vars.	cons.
25	1	100%	1.9	2	3.0%	1e-04%	100%	2.7	2	4.2%	9e-05%	100%	2.8	2	3.5%	4e-05%
	2	100%	1.3	1	2.5%	2e-05%	100%	2.8	2	3.5%	2e-05%	100%	3.2	6	2.9%	8e-06%
	3	100%	1.6	1	2.0%	7e-06%	100%	3.0	2	2.8%	4e-06%	100%	3.3	6	2.4%	2e-06%
50	1	100%	2.6	2	2.9%	3e-05%	100%	2.6	2	3.3%	2e-05%	100%	2.8	2	3.7%	1e-05%
	2	100%	2.5	2	2.3%	5e-06%	100%	2.8	4	2.6%	3e-06%	100%	4.2	6	3.6%	3e-06%
	3	100%	2.7	2	1.9%	2e-06%	100%	3.1	4	2.1%	8e-07%	95%	9.2	7	3.0%	1e-06%
100	1	100%	3.1	2	2.1%	6e-06%	100%	3.1	2	2.9%	4e-06%	100%	3.2	2	3.4%	3e-06%
	2	100%	2.8	2	1.6%	9e-07%	100%	3.1	4	2.3%	6e-07%	100%	7.1	7	3.2%	8e-07%
	3	100%	2.6	2	1.3%	2e-07%	100%	3.3	5	1.9%	2e-07%	75%	15.6	9	2.6%	3e-07%

- Fig. A2 shows the tradeoff between the false positive rate and false negative rate for varying α, β and k.
- Fig. A3 shows the effect of α, β and k (for SPHYR) on the false positive rate (FPR).
- Fig. A4 shows the effect of α, β and k (for SPHYR) on the false negative rate.
- Fig. A5 shows the effect of α, β and k (for SPHYR) on the ancestral pair recall.
- Fig. A6 shows the effect of α, β and k (for SPHYR) on the incomparable pair recall.
- Fig. A7 shows the effect of α, β and k (for SPHYR) on the clustered pair recall.
- Fig. A8 shows the effect of α, β and k (for SPHYR) on the run time.

A.6 Metastatic Colorectal Cancer (CRC) Patient 1

We have the following figures.

- Fig. A9 shows the input and output matrices of colorectal patient CRC1 from (Leung *et al.*, 2017).
- Fig. A10 shows the SCITE output tree reported by Leung *et al.* (2017).
- Fig. A11 shows the SiFit output tree.
- Fig. A12 shows the SPhyR output tree.

References

Estabrook, G. F. *et al.* (1975). An idealized concept of the true cladistic character. *Mathematical Biosciences*, **23**(3-4), 263–272.

Fernández-Baca, D. (2000). The perfect phylogeny problem. In D. Z. Zu and X. Cheng, editors, *Steiner Trees in Industries*. Kluwer Acedemic Publishers.

Gusfield, D. (1991). Efficient algorithms for inferring evolutionary trees. *Networks*, **21**(1), 19–28.

Leung, M. L. *et al.* (2017). Single cell DNA sequencing reveals a late-dissemination model in metastatic colorectal cancer. *Genome Research*, page gr.209973.116.

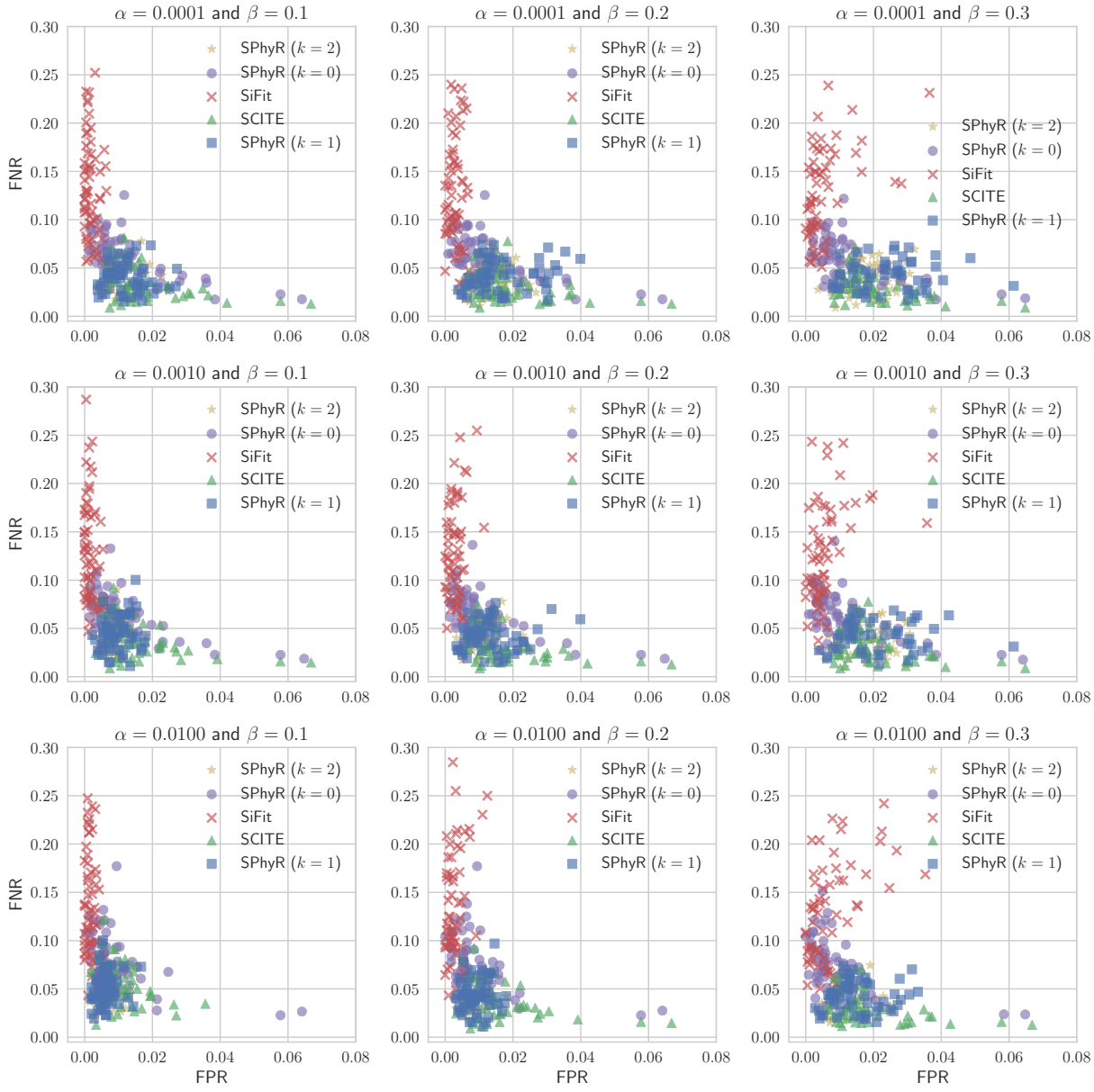

Figure A2: **Tradeoff between false positive rate (FPR) and false negative rate (FNR) for varying α, β and k (for SPhyR).** A *false positive* (FP) is a 1-entry in the output matrix B that is a 0-entry in the simulated matrix B^*. The *false positive rate* (FPR) is the fraction of false positives among the 1-entries of B. A *false negative* (FN) is a 0-entry in the output matrix B that is a 1-entry in the simulated matrix B^*. The *false negative rate* (FNR) is the fraction of false negatives among the 0-entries of B.

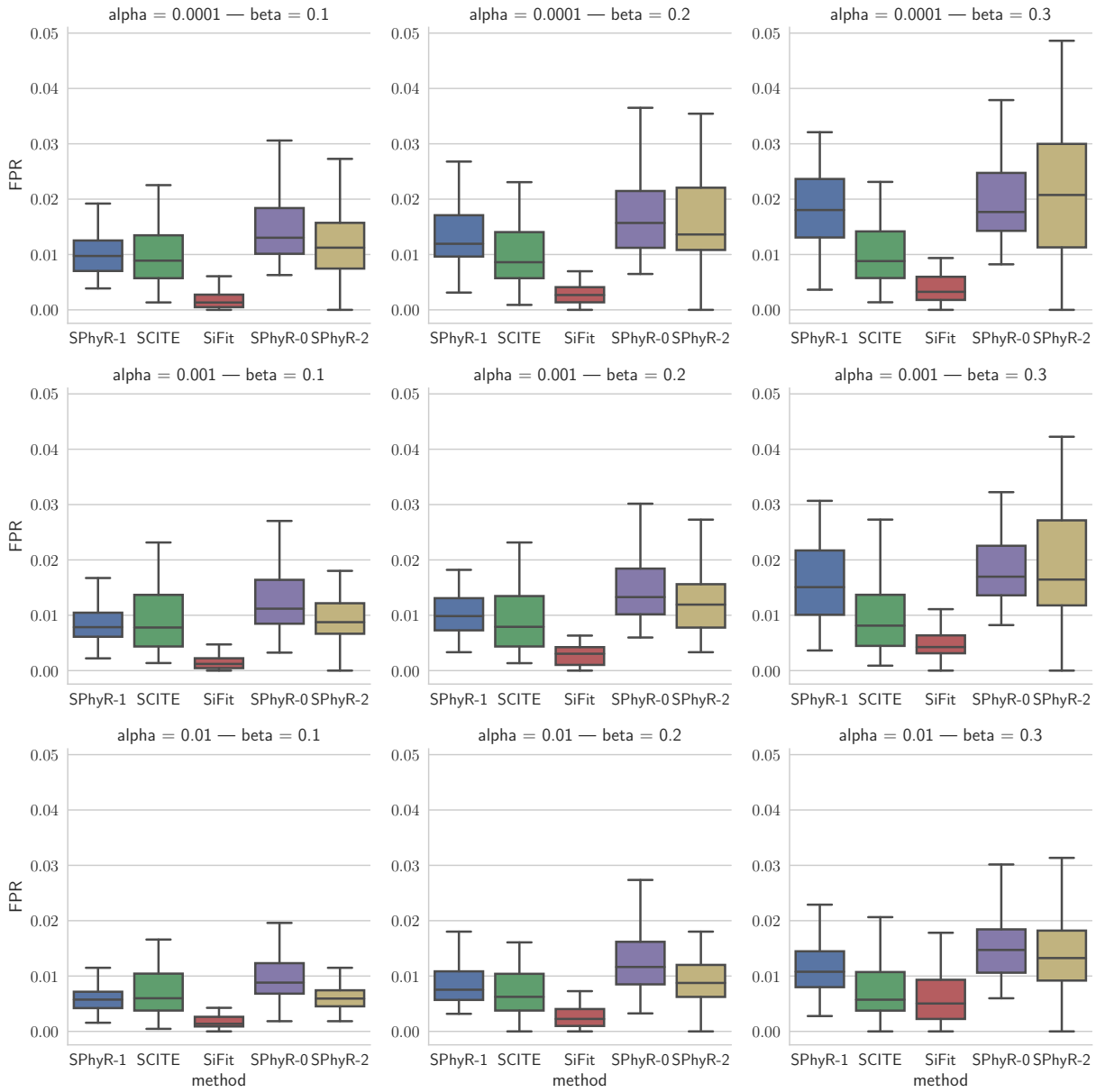

Figure A3: **The effect of α, β and k (for SPHYR) on the false positive rate (FPR).** For each method, the FPR decreases with increasing α. For SPHYR (all k) and SiFit, the FPR increases with increasing β. SCITE is fairly robust to changes in β with respect to the FPR.

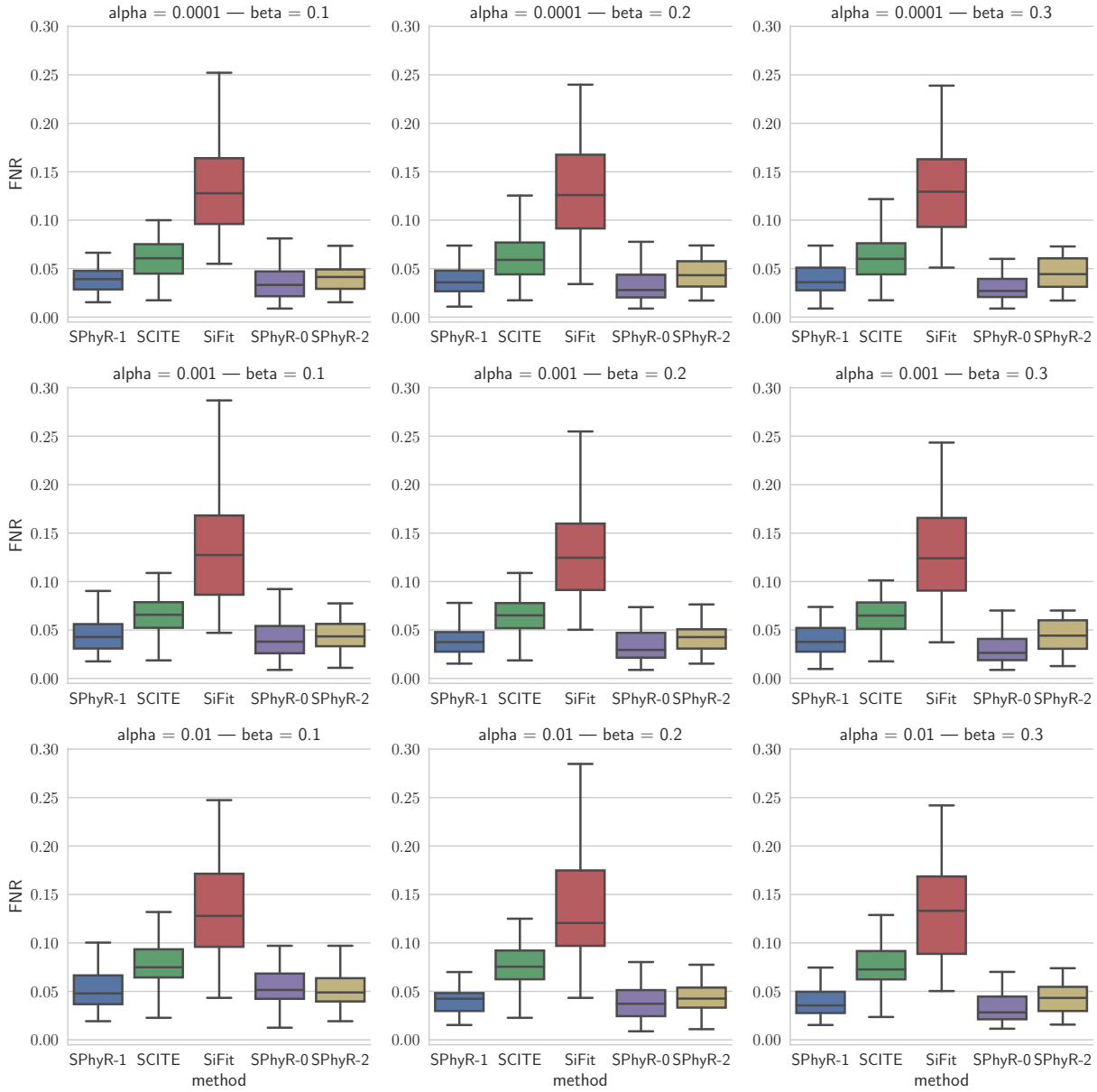

Figure A4: **The effect of α, β and k (for SPhyR) on the false negative rate (FNR).** For each method, the FNR increases with increasing α. On the other hand, the methods are fairly robust to changes in β with respect to the FNR.

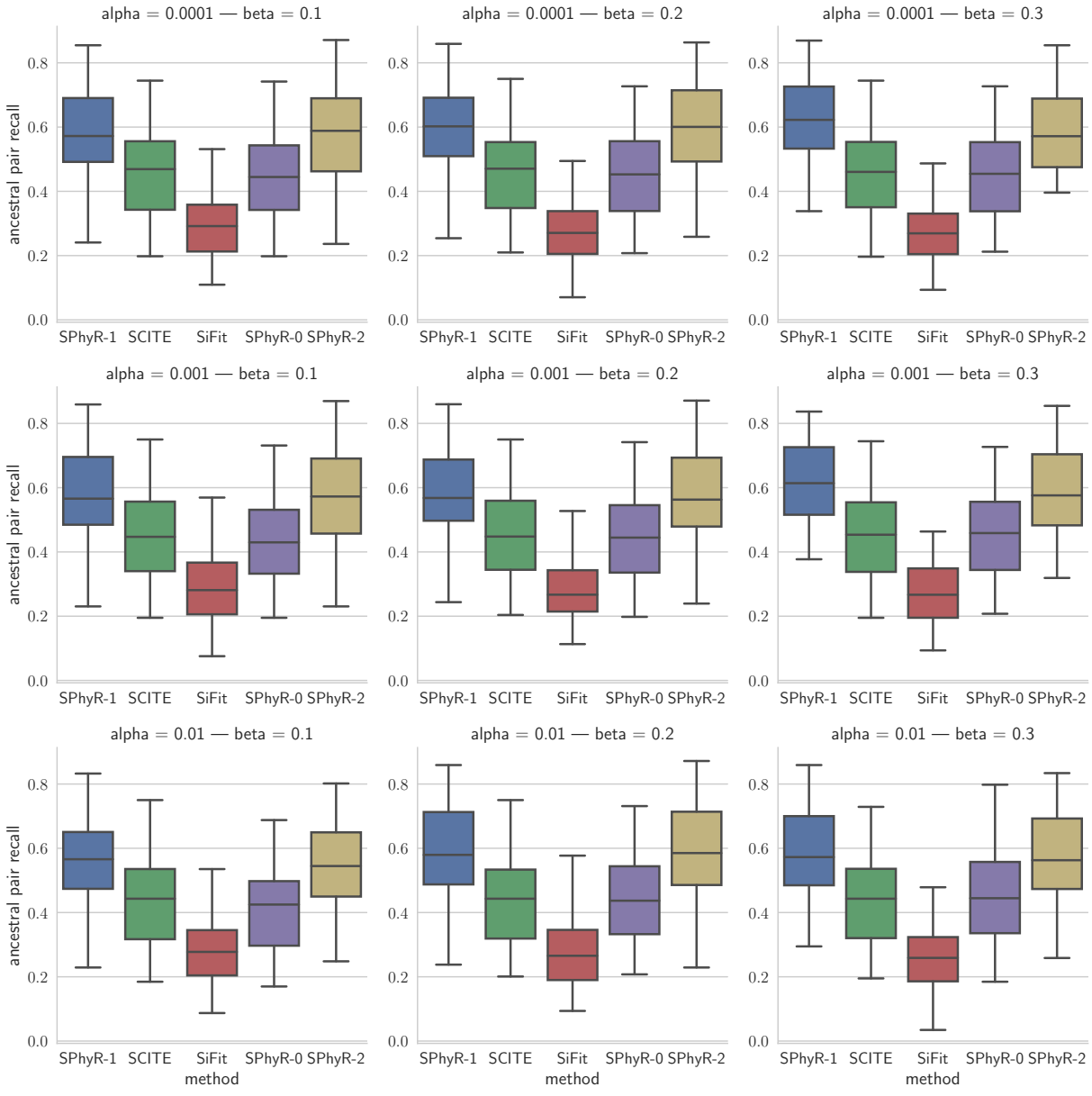

Figure A5: **The effect of α, β and k (for SPHYR) on the ancestral pair recall.** The *ancestral pair recall* is the fraction of pairs of ancestral character states of the simulated tree T^* that are retained as such in the output tree T. For this measure, the methods are fairly robust to changes in α and β. Observe that $k = 0$ performs worse than $k = 1$ and $k = 2$ for SPHYR, illustrating the necessity of allowing character loss.

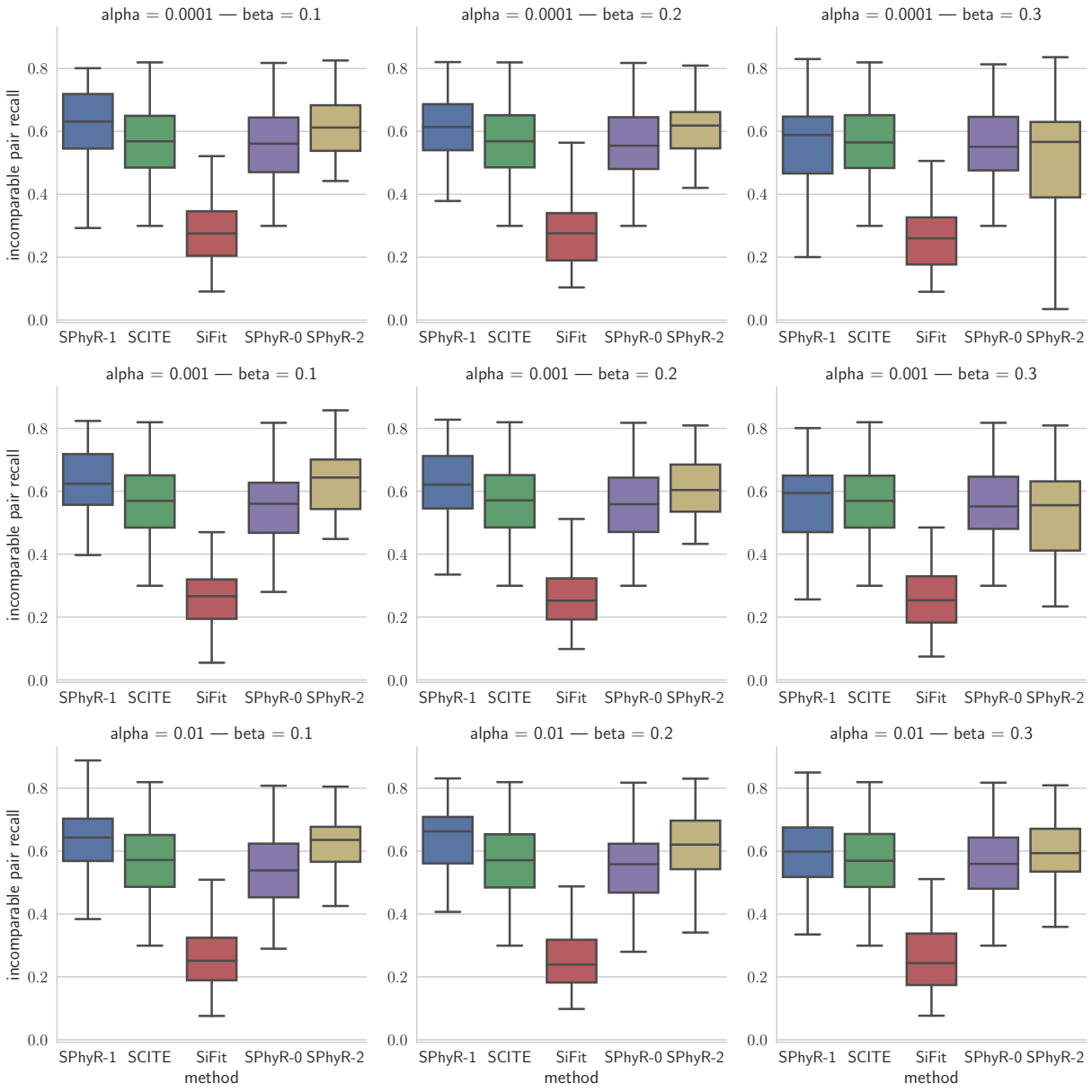

Figure A6: **The effect of α, β and k (for SPHYR) on the incomparable pair recall.** The *incomparable pair recall* is the fraction of pairs of incomparable character states of the simulated tree T^* that are retained as such in the output tree T. For this measure, the methods are fairly robust to changes in α and β.

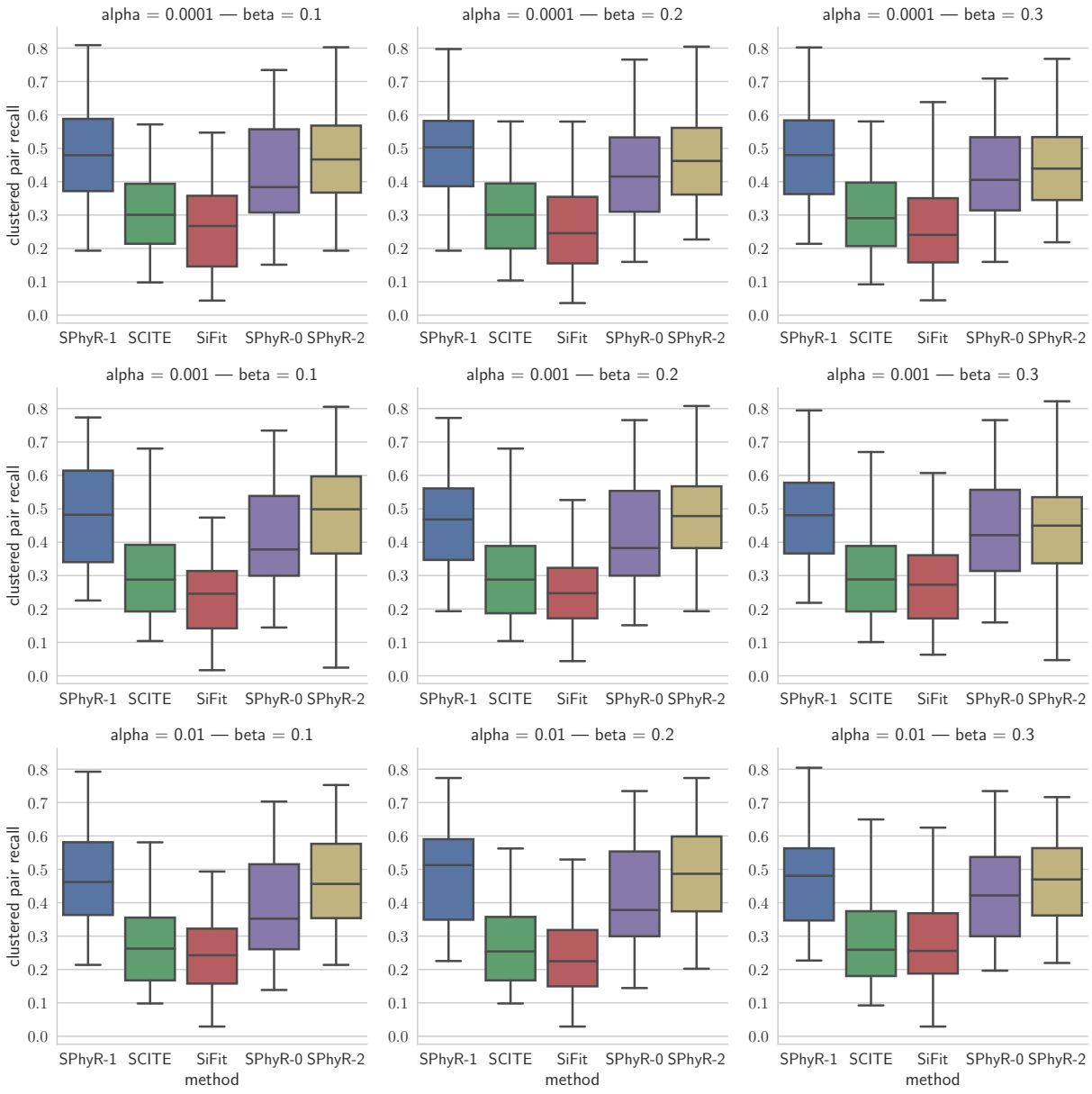

Figure A7: **The effect of α, β and k (for SPHYR) on the clustered pair recall.** The *clustered pair recall* is the fraction of pairs of clustered character states of the simulated tree T^* that are retained as such in the output tree T. For this measure, the methods are fairly robust to changes in α and β.

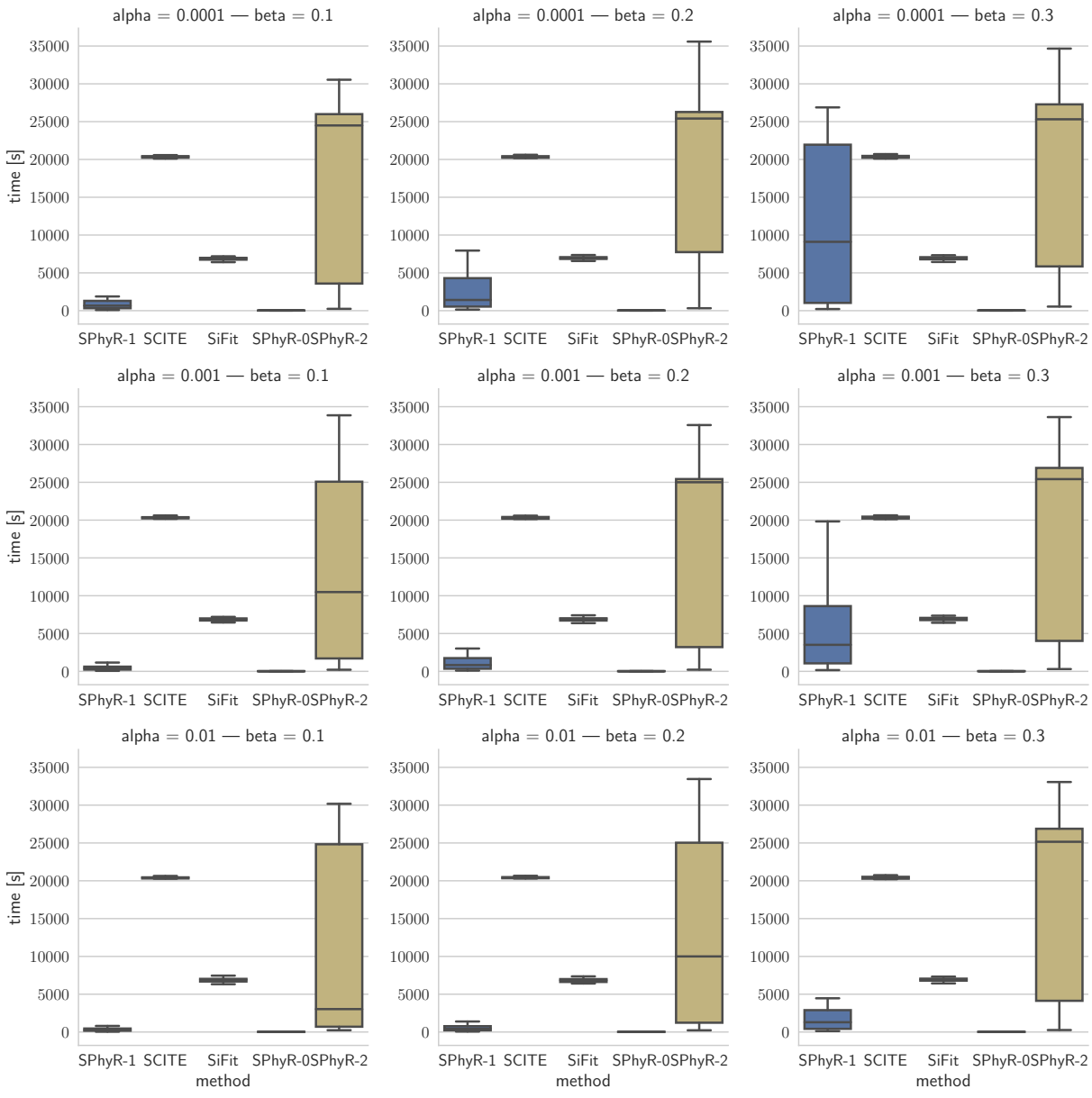

Figure A8: **The effect of α, β and k (for SPHYR) on the run time.** SiFit and SCITE are Markov-chain Monte Carlo methods, and the run time of these methods is not affected by α and β. In contrast, the run time of SPHYR increases with increasing β and k. On the other hand, the run time for SPHYR increases with decreasing α.

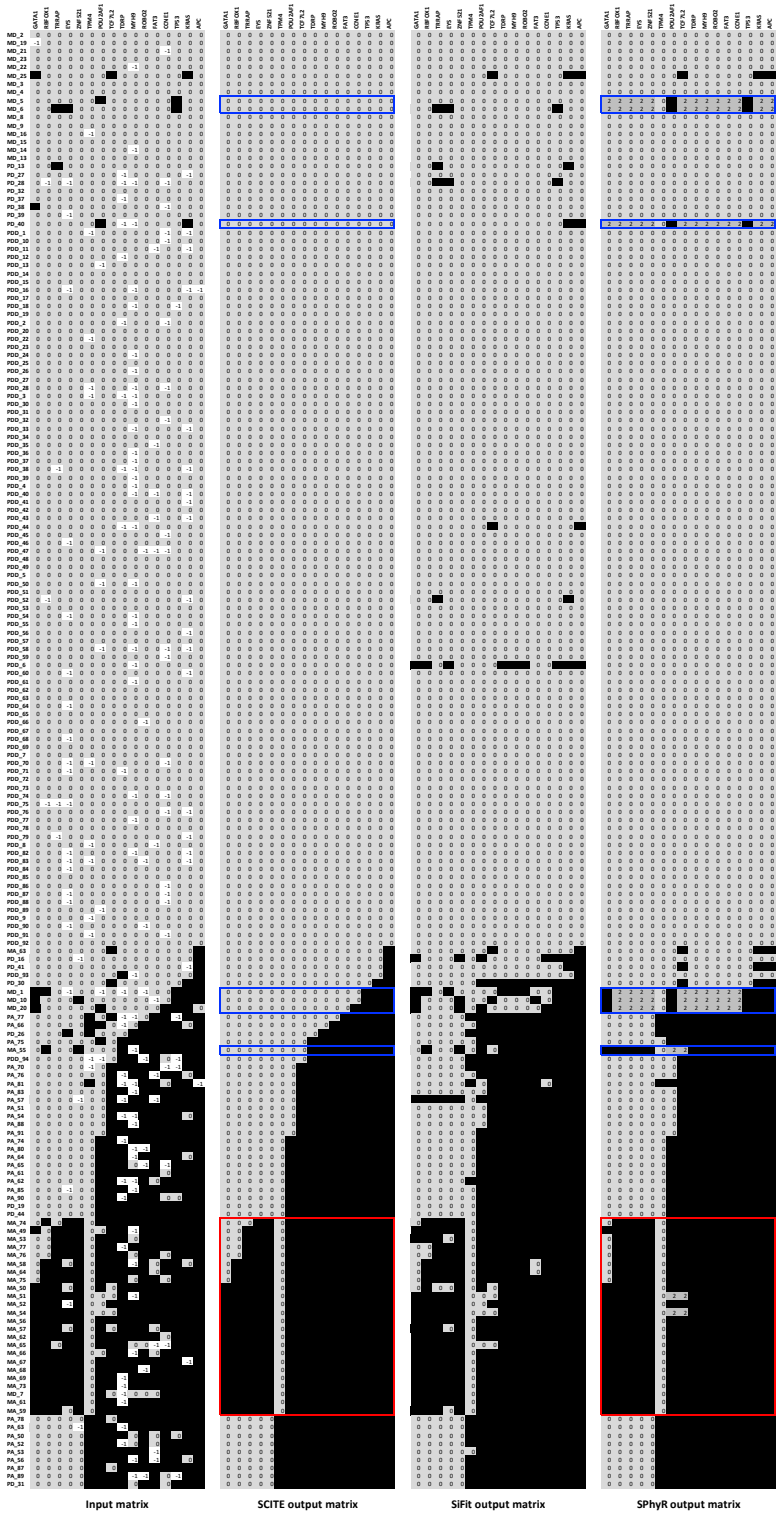

Figure A9: **Input and output matrices for metastatic colorectal cancer patient 1.** Light gray entries are 0; black entries are 1; white entries are missing (input matrix only); and dark gray entries are losses (SPHYR output matrix only). From left to right: Input matrix D, SCITE output matrix B_{SCITE}; SiFit output matrix B_{SiFit}; and SPHYR output matrix B_{SPhyR} (with $s = 12$, $t = 16$ and $k = 1$). B_{SCITE} has 278 edits from D and data log-likelihood -447.66. B_{SiFit} has 301 edits from D and data log-likelihood -471.62. B_{SPhyR} has 278 edits from D and data log-likelihood -413.38. Red boxes correspond to cells that form the metastatic clade in T_{SCITE}. Blue boxes are additional cells that SPHYR infers to be part of the metastatic clade.

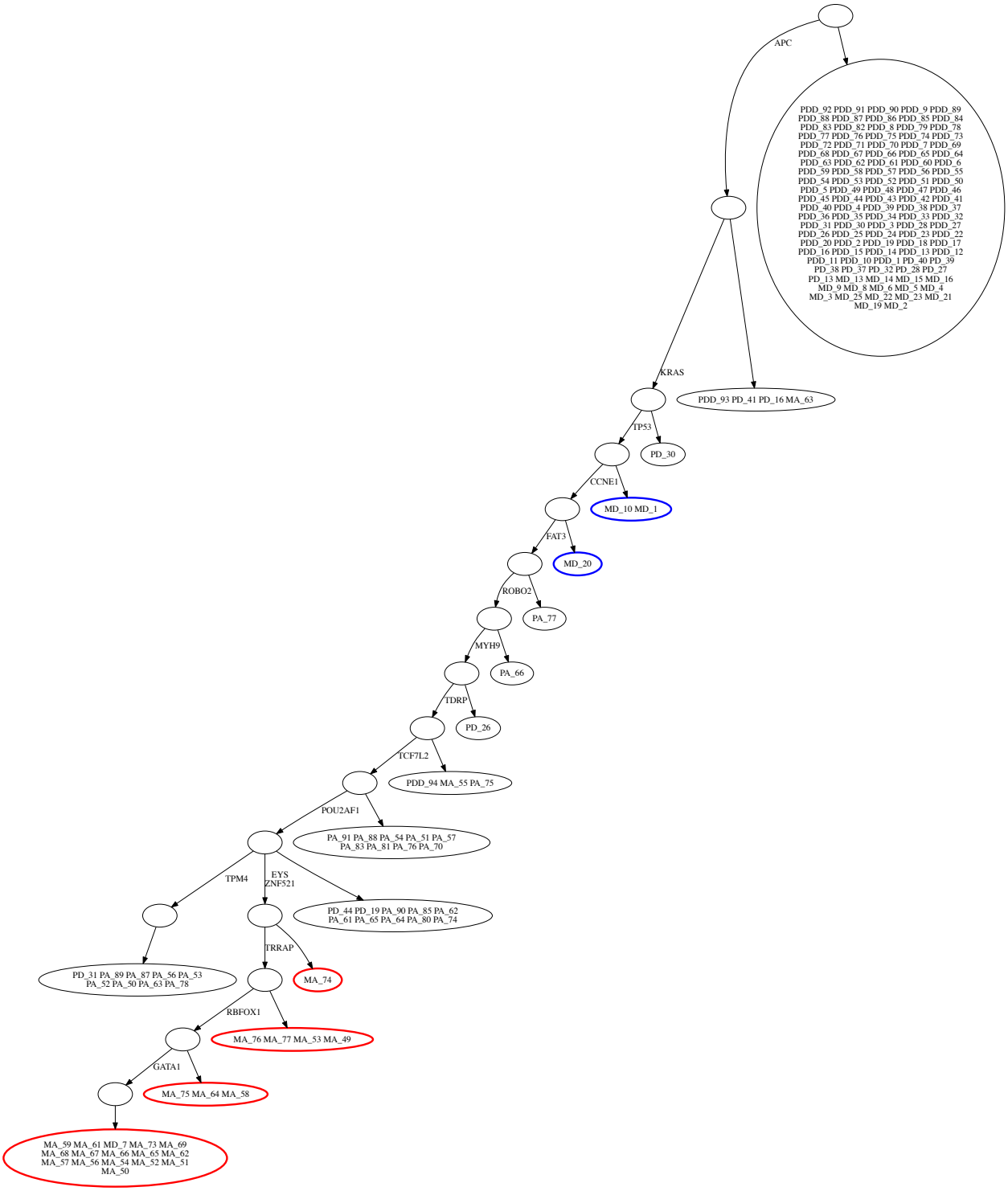

Figure A10: **SCITE output tree T_{SCITE} reported by Leung *et al.* (2017).** Edge labels are placed to the right of each edge. Red leaves form the metastatic clade. Blue leaves are additional cells that SPHYR infers to be part of this clade.

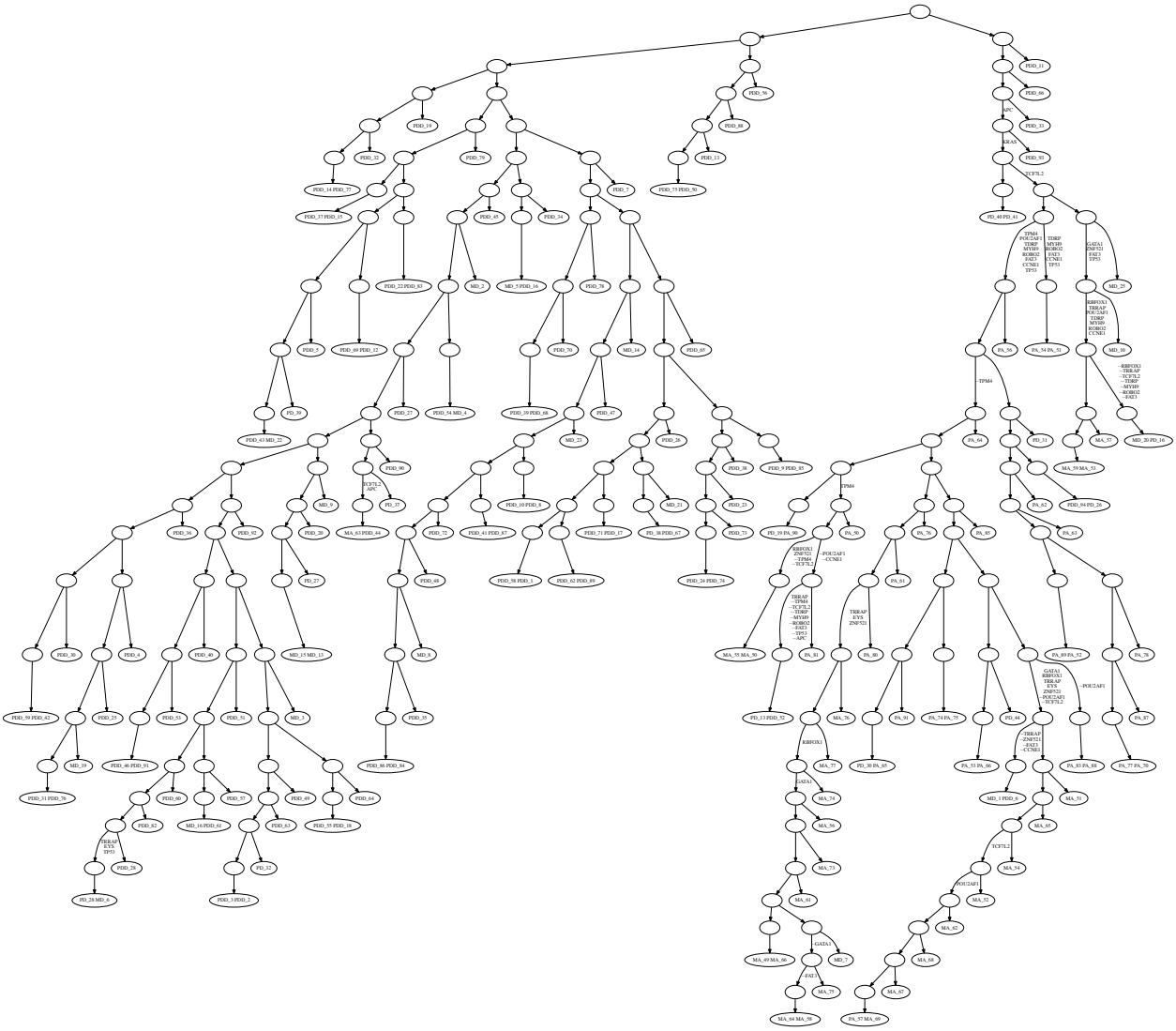

Figure A11: **SiFit output tree** T_{SiFit} **with** $\alpha = 0.0152$ **and** $\beta = 0.0789$**.** Edge labels are placed to the right of each edge. 15 SNVs on this tree have undergone parallel evolution, and 14 have been lost (prefixed by '–').

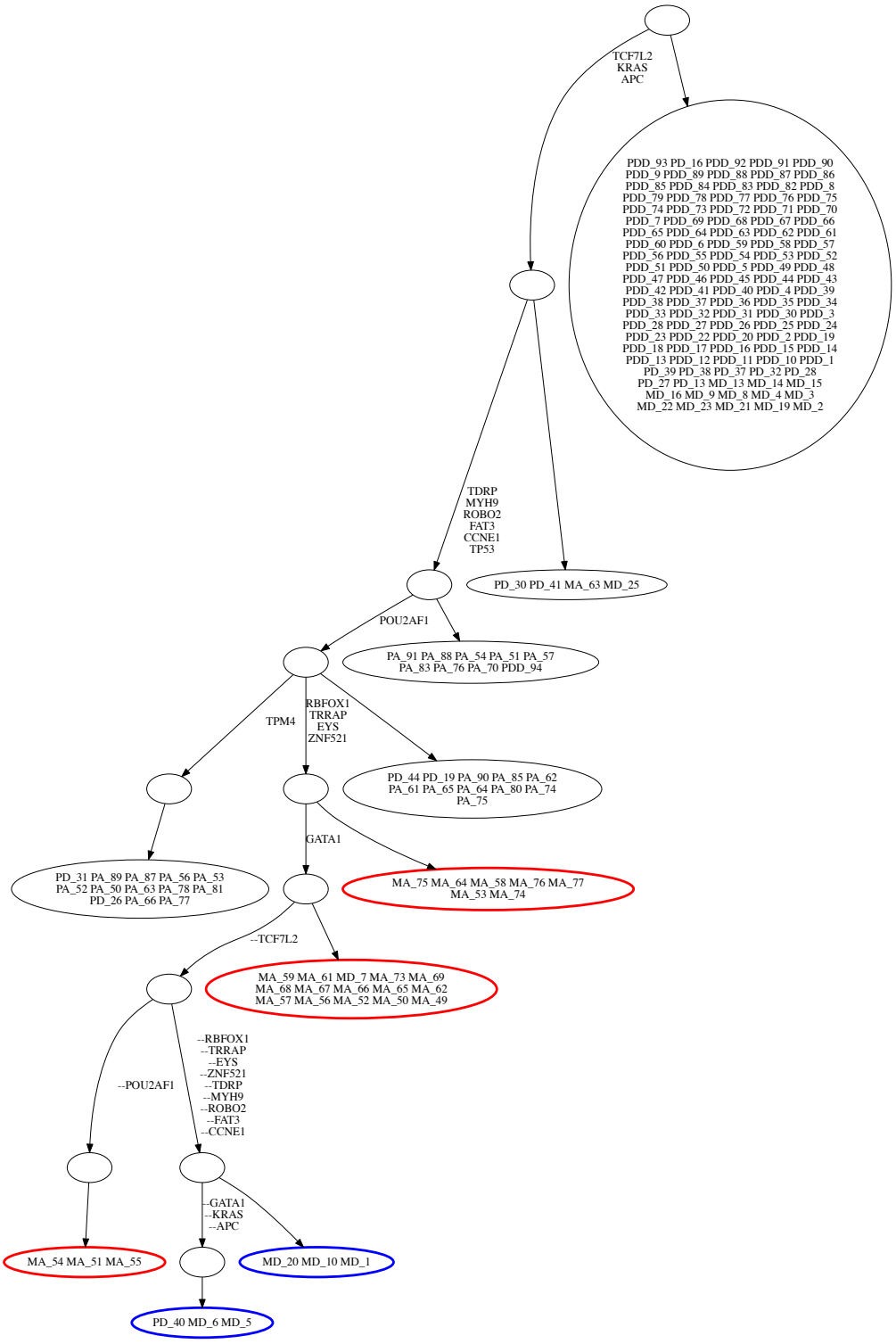

Figure A12: **SPhyR output tree T_{SPhyR} with $s = 10$, $t = 15$, $\alpha = 0.0152$ and $\beta = 0.0789$.** Edge labels are placed to the right of each edge. No SNVs on this tree have undergone parallel evolution, and 14 have been lost (prefixed by '–'). Red leaves correspond to cells that form the metastatic clade in T_{SPhyR}. Blue leaves correspond to cells that SPHYR infers to also be part of the metastatic clade. With the exception of PD_40, these cells originate from the metastatic anatomical site and were designated by Leung *et al.* (2017) as metastatic diploid (MD).