Cancer Evolution: Problems, Complexity, and Algorithms

Xinhao Liu **Yiming Zuo**

Madelyne Xiao

1 Introduction

Studying the evolutionary history of tumors can yield valuable insight into tumor development. Understanding how tumors evolve starting from a non-cancer normal cell can assist in both diagnosis and prognosis. Tumor data, however, is often messy to work with – there might be multiple lineages of tumor cells within a single sample of tissue, each consisting of distinct mutation patterns. Until recently, sequencing technologies were unable to identify single-cell samples at a sufficiently high resolution to reflect these nuances. Luckily, with the advent of single-cell sequencing, it is now possible to sequence and identify the lineages of single cells within a sample. This survey covers both early works that works with bulk sequencing data and the papers that harness single-cell sequencing technology to build more detailed and accurate phylogenies of cancer cells. These papers also seek to distinguish the causes of observed mutations, or loss of mutation, which might be due to 1) true loss of mutation, or 2) a false positive, e.g., the result of an entire segment of genomic data having been lost (a copy-number aberration/alteration). A few papers also seek to find the consensus among the CNAs and SNVs, two common types of mutations in cancer, to improve the performance. In general, these papers employ LP programming to find efficient solutions to hard problems in phylogeny reconstruction (with one exception: the last paper uses a heuristic to find a consensus tree among cancer patients).

2 Background

In this section we introduce some terms, including both mathematical definitions and their biological implications, that are used throughout the survey.

Phylogeny As per the theory of evolution, all organisms in the world share a common ancestor; thus, their evolutionary relationships can be summarized by a tree-like structure which we call a *phylogeny*. A binary phylogenetic tree on a set $\mathcal{X}$ of taxa (e.g., $\mathcal{X} = \{human, chimp, gorilla\}$) is a binary tree whose leaves are bijectively labeled by the elements in $\mathcal{X}$. That is, if we have n taxa in our dataset, then the phylogenetic tree has n leaves. The meaning of the taxa depends on the applications. For example, in *species phylogeny*, all the taxa in $\mathcal{X}$ are extant species (e.g., human, chimp, gorilla), and each internal node of the phylogeny typically represents an ancestor species to the extant species. Meanwhile, in a *cancer phylogeny*, all the taxa, as well as internal nodes, are cancer *clones*, which we will define shortly. Usually, the phylogenetic tree is rooted – we designate an internal (non-leaf) node as the root, which represents the common ancestor of all nodes in the tree. In the case of cancer phylogeny, the root is usually either the clone of normal (non-cancer) cells, or a *founding clone*, which is a set of initial cancer cells that caused the onset of the cancer. It is worth noting that, while phylogenetic trees are usually assumed to be binary trees, that does not need to be the case.

Given a set of taxa $\mathcal{X}$, the *phylogeny inference problem* tries to infer the phylogenetic tree on $\mathcal{X}$, based on the data provided by $\mathcal{X}$ (e.g. genome sequences of each species), according to some criteria (e.g. lowest number of mutations, highest likelihood according to some evolutionary model, etc.)

Cancer clones Cancer is a disease that results from somatic mutations that arise and accumulate during an individual's lifetime that eventually lead to uncontrolled growth of a collection of cells which form a tumor. The clonal theory of cancer posits that all cells in the tumor have descended from a single founding cell, which is the original cancer cell. Subsequently, additional advantageous mutations lead to *clonal expansions* of the founding clone, which gives rise to multiple cancer clones, each with heterogeneous mutational profiles, that form a tumor. As a result, a tumor consists of *clones*, which are subpopulation of cells sharing a unique combination of somatic mutations. Cancer phylogenetic inference aims at building a phylogenetic tree on the clones within a tumor.

Single-nucleotide variants (SNV) Single-nucleotide variant is one common type of somatic mutation in cancer. All cells are supposed to have the same copy of the genome. When a cell goes through a SNV, one mutation happens at one specific nucleotide of the genome of the cell (e.g., turns a T into a A), which makes this cell's genome differ from other cell's genome by one nucleotide.

Copy-number alterations (CNA) Other than only mutating one single nucleotide, somatic mutations in cancer could also amplify or delete large genomic regions. In the extreme case, the whole genome could be duplicated or deleted. To model CNAs, we represent a genome as a sequence of intervals which we call positions, labeled 1 to n from left to right. The *copy-number profile* of a clone specifies the number of copies of each of the n positions. Formally, a profile $\mathbf{y} = [y_s]$ is a vector of length n. An entry $y_s \in \mathbb{N}$ indicates the number of copies of position s in the genome of the clone.

3 The Variant Allele Frequency Factorization (VAFFP) Problem

During the earlier days of cancer phylogeny reconstruction method developments, single-cell sequencing of tumor cells was uncommon. Most of the data sets at the time measure somatic mutations in multiple spatially distinct samples from the same tumor, where each sample is a mixture of cancerous and normal cells. Sequencing one sample gives the variant allele frequency (VAF) of the sample at each of n genomic positions. Intuitively, the VAF at a genome position is proportional to the fraction of cells in the sample that contains mutation at that position. Suppose we sequence m samples from a tumor with n mutation sites. The data are described by an $m \times n$ *frequency matrix* F, where F_{pi} indicates the observed VAF in sample p for mutation i. Given F, we want to build the clonal history of tumor evolution. This section is based on *(El-Kebir et. al., 2015)* [5].

3.1 Problem statement

We encode each clone by a binary vector in $\{0,1\}^n$, where n is the number of possible mutation sites. 1 represents a somatic mutation at that site, and 0 represents no mutation. We follow the *infinite sites assumption* of mutations, which states that a mutation occurs at a single site at most once during the clonal evolution of the tumor. The evolution of the tumor clones is described by a phylogenetic tree where each vertex represents a clone that has existed during the tumor's evolution, and each edge (i,j) represents the mutation(s) that turns clone i into clone j, In practice, we group individual mutations into sets and regard them as single mutations. This leads us to the definition of an *n-clonal tree*.

Definition 1 *A rooted tree T of n vertices is an n-clonal tree for a mutation set* $[n] = \{1, ..., n\}$ *provided each edge is labeled with exactly one mutation from* $[n]$ *and no mutation appears more than once in T. Let* $\mathcal{T}_n$ *be the set of all n-clonal trees.*

In addition, we assume the root v_r of an n-clonal tree only contains one mutation $r \in [n]$. v_r is the *founding clone* of the tumor. We denote the remaining vertices by v_j where $j \neq r$ is the mutation on the last edge in the path from v_r to v_j. We may encode an n-clonal tree T by an $n \times n$ binary matrix B. We label each vertex (clone) v_j of T by a binary row vector $b_j \in \mathbb{R}^n$ with 1's at the r-th position and positions indicated by the edge labels of the unique path from v_r to v_j and 0's at the remaining positions. That is, b_j represents the set of mutations present in v_j. Let B be the $n \times n$ binary matrix whose j-th row is b_j. We call B an *n-clonal matrix*, and let $\mathcal{B}_n$ be the set of all n-clonal matrices. It turns out that $\mathcal{B}_n$ is a subset of *perfect phylogeny matrices*[10] that satisfies a set of conditions.

The observed frequency matrix F of the frequency of each mutation in each sample is related to the clonal tree T by the proportion of cells from each clone and normal cells in each sample. Therefore,

we define a $m \times n$ usage matrix U, such that U_{pi} indicates the fraction of cells in sample p that come from clone v_i. Notice that $U_{pi} \geq 0$ and $\sum_{j=1}^{n} U_{pj} \leq 1$. Let the set of all $m \times n$ usage matrices be $\mathcal{U}_{mn}$. Since each sample is a mixture of clones from T with the mixture proportions defined by U, we have

$$F = \frac{1}{2}UB \tag{1}$$

The coefficient $\frac{1}{2}$ comes from the biological fact that mutations are heterozygous, and thus each $F_{pj} \in [0, 0.5]$. Now we can define the Variant Allele Frequency Factorization (VAFFP) problem.

Definition 2 *(VAFFP) Given an $m \times n$ frequency matrix F, find a usage matrix $U \in \mathcal{U}_{mn}$ and a clonal matrix $B \in \mathcal{B}_n$ such that $F = \frac{1}{2}UB$.*

3.2 Solving the VAFFP

In this section, we characterize the solution to the VAFFP as constrained spanning arborescences of a directed acyclic graph (DAG), and give an ILP algorithm for solving the problem. We first prove two conditions for the existence of solutions.

3.2.1 A necessary condition and the ancestry graph

We say that a B (or T, they are equivalent) *generates* a frequency matrix F if and only if there exists a matrix $U \in \mathcal{U}_{mn}$ such that $F = \frac{1}{2}UB$. We start by observing that any T induces a partial ordering on vertices (clones), such that for $j, k \in [n]$, $j \prec_T k$ if and only if vertex v_j is an ancestor of vertex v_k. Otherwise j and k are incomparable. We now prove the following necessary condition for a clonal tree T to generate a frequency matrix F.

Lemma 3 *(Ancestry Condition) If T generates F and $j \prec_T k$ then $f_{pj} \geq f_{pk}$ for all samples $p \in [m]$.*

Proof. There is a corresponding n-clonal matrix B of T. Because B is a perfect phylogeny matrix, there is a partial order on the columns of B [11], such that for $j, k \in [n]$, we have $j \prec_B k$ if and only if $I(j) \supseteq I(k)$, where $I(j)$ is the positions of the 1 entries of column j. Since B and T are equivalent, and $j \prec_T k$, we have $I(j) \supseteq I(k)$. Because each entry in U is non-negative, we have the following for all samples $p \in [m]$:

$$\begin{aligned} f_{pj} &= \frac{1}{2}\sum_{i=1}^{n} u_{pi} b_{ij} \\ &= \frac{1}{2}\sum_{i \in I(j)} u_{pi} \\ &\geq \frac{1}{2}\sum_{i \in I(k)} u_{pi} \\ &= \frac{1}{2}\sum_{i=1}^{n} u_{pi} b_{ik} \\ &= f_{pk} \end{aligned}$$

■

We summarize all possible ancestral relationships between mutations in F in a graph.

Definition 4 *Given an $m \times n$ frequency matrix F, the ancestry graph $G = (V, E)$ is the directed graph with vertices $V = \{v_1, ..., v_n\}$ and edges $E = \{(v_j, v_k) | f_{pj} \geq f_{pk}$ for all $p \in [m]\}$.*

Notice that if all columns of F are distinct, which we can safely assume, then the ancestry graph G is a DAG. A *spanning arborescence* of G is a subgraph $G' = (V, E')$ with $E' \subseteq E$ such that there exists a unique path from the root vertex v_r to every other vertex $v \in V$.

Theorem 5 *If T generates F then T is a spanning arborescence of G.*

Therefore, we have proved a necessary condition for an n-clonal tree T to generate a frequency matrix F. Namely, T must be a spanning arborescence of the ancestry graph G associated with F.

3.2.2 A second condition for sufficiency

We have shown that the *ancestry condition* (or equivalently, being a spanning aborescence of G) is a necessary condition for T to generate F. However, not all ancestry graphs have a spanning arborescence, and not all spanning arborescences generate F. In this section, we prove a second condition for T to achieve sufficiency.

Lemma 6 *(Sum Condition) Give a clonal tree T, let $\delta(v_j)$ denote the children of a vertex v_j in T. If T generates F then for all samples $p \in [m]$ and mutations $j \in [n]$,*

$$f_{pj} \geq \sum_{v_k \in \delta(v_j)} f_{pk} \tag{2}$$

Proof. All $v_k \in \delta(v_j)$ are pairwise incomparable. Therefore all $I(k)$ with $v_k \in \delta(v_j)$ are pairwise disjoint [11]. Moreover, $I(j) = \bigcup_{k \in \delta(v_j)} I(k) \cup \{j\}$. Because each entry in U is non-negative, we have the following for all samples $p \in [m]$:

$$\begin{aligned}
f_{pj} &= \frac{1}{2} \sum_{k=1}^{n} u_{pk} b_{kj} = \frac{1}{2} \sum_{k \in I(j)} u_{pk} \\
&= \frac{1}{2} u_{pj} + \frac{1}{2} \sum_{v_k \in \delta(v_j)} \sum_{l \in I(k)} u_{pl} \\
&= \frac{1}{2} u_{pj} + \sum_{v_k \in \delta(v_j)} \frac{1}{2} \sum_{i=1}^{n} u_{pl} b_{lk} \\
&= \frac{1}{2} u_{pj} + \sum_{v_k \in \delta(v_j)} f_{pk} \\
&\geq \sum_{v_k \in \delta(v_j)} f_{pk}
\end{aligned}$$

■

We now proceed to prove that the *sum condition*, together with the *ancestry condition*, are necessary and sufficient. We first state the following lemma without proof in the interest of space.

Lemma 7 *If an $m \times n$ frequency matrix F satisfies Equation 2 for the tree T corresponding to $B \in \mathcal{B}_n$, then B generates F.*

Now we are ready to prove the following theorem, which establishes that the sum condition and the ancestry condition are necessary and sufficient.

Theorem 8 *T generates $F = [f_{pj}]$ if and only if T is a spanning arborescence of G such that Equation 2 holds for all f_{pj}.*

Proof. The only if direction follows from Theorem 5 and Lemma 6.

For the if direction, we know that T spans all the vertices of G, and thus is a valid clonal tree of F, with a corresponding clonal matrix B. By Lemma 7, B generates F. ■

Theorem 8 provides a characterization of the solutions to the VAFFP that allows us to only focus on finding B (equivalently, T) without considering U in solving the problem. This leads us to an integer linear programming algorithm.

3.3 An integer linear programming solution

In this section, we provide an ILP formulation to find the largest arborescence in an ancestry graph $G = (V, E)$ that satisfies the sum condition. If this is a spanning arborescence then we have found a solution to the VAFFP.

We introduce an artificial root v_r that has an outgoing edge to all the vertex in V. Let $E' = E \cup \{(v_r, v) | v \in V\}$ be the extended edge set. For $v \in V \cup \{v_r\}$, define $\delta^+(v) = \{w \in V | (v, w) \in E'\}$ be the set of all vertices connected to v by an outgoing edge from v, and similarly define $\delta^-(v) = \{w \in V | (w, v) \in E'\}$ be the set of all vertices connected to v by an incoming edge to v. Let decision variables $\mathbf{x} \in \{0, 1\}^{|E'|}$ be binary variables indicating the presence of each edge in the arborescence.

$$
\begin{aligned}
\max \quad & \sum_{(v_j, v_k) \in E'} x_{jk} \\
s.t. \quad & \sum_{v_j \in \delta^+(v_r)} x_{rj} = 1 \\
& x_{kl} \leq \sum_{v_j \in \delta^-(v_k)} x_{jk} && \forall (v_k, v_l) \in E \\
& \sum_{v_j \in \delta^-(v_k)} x_{jk} \leq 1 && \forall v_k \in V \\
& \sum_{v_j \in \delta^-(v_k)} f_{pk} x_{jk} \geq \sum_{v_l \in \delta^+(v_k)} f_{pl} x_{kl}. && \forall p \in [m], v_k \in V \\
& x_{jk} \in \{0, 1\} && \forall (v_j, v_k) \in E'
\end{aligned}
$$

The first constraint ensures that there is only one root vertex in the arborescence. The second constraint ensures that each vertex is connected in T. The third constraint ensures each clone in T has only one ancestor. The fourth constraint is the sum condition. The objective maximizes the number of edges in the arborescence.

4 The Copy-number Tree (CNT) Problem

The VAFFP problem assumes single nucleotide variations as input data, hence ignoring the effect of copy number aberrations on VAFs. With the increased prevalence of single-cell sequencing in recent years, we are able to sequence the genome of individual cancer cells and, hence, the genome of each cancer clone. As a result, cancer phylogenetic methods that take in single-cell data need to be developed. In this section, we study the problem of constructing a phylogenetic tree on clones from a tumor in the case where input data are copy number aberrations of each clone. We formally define the copy-number tree (CNT) problem, which asks to construct a phylogenetic tree whose k leaves are the k given copy-number profiles, and internal vertices are profiles such that the sum of distances over all edges is minimum. We prove that the CNT problem is NP-hard and present an ILP formulation. This section is based on *(El-Kebir et. al., 2017)* [6].

4.1 Problem statement

To recap, a *copy-number profile* of a clone specifies the number of copies of each interval (or equivalently, position) along the genome. Normal cells have two copies of the genome at each interval, and somatic mutations make this number deviate from two. An *event* acting on profile $\mathbf{y}_i$ increases or decreases copy-numbers of a contiguous segments of $\mathbf{y}_i$. Formally, an event is a triple (s, t, b) where $s \leq t$ and $b \in \mathbb{Z}$. If b is positive then the copy number at positions $s, ..., t$ are incremented by b, while if b is negative then the copy number at position $s, ..., t$ are decremented by at most $|b|$. The copy number at a position cannot go below 0 because we cannot have a negative amount of genetic material at a position. Also, once the copy number at a position goes to 0, it can never go above 0 again because somatic mutations cannot recreate a gene. Formally, applying event (s, t, b) to $\mathbf{y}_i$ results in a new profile $\mathbf{y}'_i$ such that

$$
y'_{i,l} = \begin{cases} max\{y_{i,l} + b, 0\}, & \text{if } s \leq l \leq t \text{ and } y_{i,l} \neq 0. \\ y_{i,l}, & \text{otherwise.} \end{cases}
$$

We describe the evolutionary process that gives rise to the tumor clones by a phylogenetic tree, which we call the *copy-number tree*. A copy-number tree T is a rooted binary tree. Each vertex v_i of T corresponds to a clone, thus is labeled by a copy-number profile $\mathbf{y}_i$. The root of T corresponds to

normal cells, which have 2 copies of genome at each position. Thus, $y_{r,s} = 2$ for all positions s of root r. Each edge $(v_i, v_j) \in E(T)$ represents the sequence of events that turn clone i into clone j, and is labeled by a sequence of events $\sigma(i,j) = (s_1, t_1, b_1), ..., (s_q, t_q, b_q)$. Notice that the order of the events matters. The *cost* of an event (s, t, b) is the number of changes, which is equal to $|b|$, and the cost of an edge (v_i, v_j) is the total cost of the events in $\sigma(i,j)$. The cost $\Delta(T)$ of a copy-number tree T is the sum of the costs of all edges in T.

Our data, or observations, are k profiles $\mathbf{c}_1, ..., \mathbf{c}_k$ of the k extant clones. We obtain this observation via single-cell sequencing of one individual cell from each clone. Our goal is to find a copy-number tree T^* such that there are k leaves of T^*, labeled by $\mathbf{c}_1, ..., \mathbf{c}_k$, and the cost $\Delta(T^*)$ is minimum among all such tree T. Furthermore, we assume the maximum copy-number of each profile in the phylogeny is bounded by a constant $e \in \mathbb{N}$. We now define the CNT problem.

Definition 9 *(Copy-number tree (CNT) problem) Give profiles $\mathbf{c}_1, ..., \mathbf{c}_k$ on n positions and an integer $e \in \mathbb{N}$, find a copy-number tree T^*, vertex labeling $\mathbf{y}_i$ and edge labeling $\sigma(i,j)$ such that*

- *T^* has k leaves labeled $1, ..., k$ and $\mathbf{y}_i = \mathbf{c}_i$ for all $i \in \{1, ..., k\}$.*
- *$y_{i,s} \leq e$ for all $v_i \in V(T^*)$ and $s \in \{1, ..., n\}$.*
- *$y_{r,s} = 2$ for the root r and $s \in \{1, ..., n\}$.*
- *$\Delta(T^*)$ is minimum.*

4.2 Complexity

In this section, we show that the CNT problem is NP-hard by reduction from the maximum parsimony phylogeny (MPP) problem [7]. In MPP, we seek to find a binary phylogeny T, which is a full binary tree whose vertices are labeled by binary vectors of size n. The cost of a binary phylogeny T is the sum of the Hamming distance between the vectors at the two ends of an edge for all edges in T. Given k binary vectors $\mathbf{b}_1, ..., \mathbf{b}_k$ of size n, we seek a full binary tree T such that T has k leaves that are labeled by the k given binary vectors, the root of T is labeled by a vector of all 0's, and T is of minimum cost among all possible phylogenies. We consider the decision version where we are asked whethter there exists a binary phylogeny T with cost at most h. This problem is known to be NP-complete.

Theorem 10 *The CNT problem is NP-hard.*

Proof. We define the reduction from the MPP problem. Let $\mathbf{b}_1, ..., \mathbf{b}_k$ be an instance of MPP, where each $\mathbf{b}_i$ is of length n. The corresponding CNT instance has parameters $e = 2$, and profiles $\mathbf{c}_1, ..., \mathbf{c}_{k+1}$ of length $n + (n-1)nk$. Each input profile $\mathbf{c}_i$, where $i \in \{1, ..., k\}$, is defined as

$$\mathbf{c}_i = \phi(\mathbf{b}_i) = (\phi(b_{i,1})\Omega\phi(b_{i,2})\Omega...\Omega\phi(b_{i,n})) \tag{3}$$

where *true positions*

$$\phi(b_{i,s}) = \begin{cases} 1, & \text{if } b_{i,s} = 1. \\ 2, & \text{otherwise.} \end{cases} \tag{4}$$

and Ω is called a *wall*, which is a vector of size nk such that for each $j \in \{1, ..., nk\}$

$$\Omega_j = \begin{cases} 2, & \text{if } j \text{ is odd.} \\ 1, & \text{otherwise.} \end{cases} \tag{5}$$

Finally, $\mathbf{c}_{k+1} = (2, 2, ..., 2)$. This reduction can be computed in polynomial time.

We claim that an MPP instance, with input $\mathbf{b}_1, ..., \mathbf{b}_k$ and each $|\mathbf{b}_i| = n$, admits a binary phylogeny T with cost at most h if and only if the corresponding CNT instance, with input $\mathbf{c}_1, ..., \mathbf{c}_{k+1}$ and $e = 2$, admits a copy-number tree T' with cost at most $h + W$, where $W = \frac{(n-1)nk}{2}$. We defer the readers to the original paper [6] for proof. ■

4.3 An integer linear programming solution

In this section, we present an ILP formulation of the solution to the CNT problem $\mathbf{c}_1, ..., \mathbf{c}_k$ and e. Recall that we want to build a full binary tree with k leaves corresponding to the k input copy

number profiles. Therefore, we define a directed graph G that contains any full binary tree with k leaves as a spanning tree. We define an order $v_1, ..., v_k, ..., v_{2k-1}$ on the nodes of G such that v_1 is the root and $v_k, ..., v_{2k-1}$ are the leaves. The orders on the internal nodes are defined by the edge set $\{(v_i, v_j) | 1 \leq i < k, 1 \leq i < j \leq 2k-1\}$.

We want the ILP to select a spanning tree T of G that is a full binary tree. To this end, we introduce a binary variable $x_{i,j} \in \{0, 1\}$ for each edge $(v_i, v_j) \in E(G)$, indicating whether the edge is in T, and introduce constraints of the form

$$\sum_{i \in N^-(j)} x_{i,j} = 1$$

$\forall 1 < j \leq 2k-1$, where $N^-(j)$ is the set of all nodes which have an incoming edge to j. This constraint ensures that each node in T only has one parent. A similar constraint ensures each internal node has exactly two children.

We introduce variables $y_{i,s} \in \{0, ..., e\}$ to encode the copy number of position s in node i. To model the cost of any edge (v_i, v_j), we first model the cost of amplifications and the cost of deletions covering any position s with two separate variables $a_{i,j,s} \in \{0, ..., e\}$ and $b_{i,j,s} \in \{0, ..., e\}$, which encodes the cost of the amplifications/deletions covering position s in the sequence of events $\sigma(v_i, v_j)$ labeling edge (v_i, v_j). We have constraints of the form

$$y_{j,s} + d_{i,j,s} = y_{i,s} + a_{i,j,s}$$

From $a_{i,j,s}$ and $b_{i,j,s}$, we can deduce the number of amplifications and deletions *starting* at each position s. Variables $\overline{a}_{i,j,s}$ and $\overline{b}_{i,j,s}$ represent the number of new amplification and new deletions that start at position s in edge (v_i, v_j). Thus, the cost of an edge (v_i, v_j) is the sum of the number of amplifications and deletions that start at each position s. The objective of the ILP is then the sum of the costs of all the edges in T

$$\min \sum_{(v_i, v_j) \in E(G)} \sum_{1 \leq s \leq n} x_{i,j} \cdot (\overline{a}_{i,j,s} + \overline{d}_{i,j,s})$$

where the product is modeled using the following constraint

$$w_{i,j,s} \geq \overline{a}_{i,j,s} + \overline{d}_{i,j,s} - (1 - x_{i,j}) \cdot 2e$$

for each position s, each edge $(v_i, v_j) \in E(G)$, and $w_{i,j,s} \geq 0$.

5 Phylogeny Estimation under Loss and Error

5.1 The k-Dollo parsimony model

There are multiple well-known models for molecular evolution. According to the infinite sites model, a mutation might be gained once (0 -> 1) and never lost (1 -> 0); according to the finite sites model, a mutation might be gained *and* lost multiple times. The Dollo model imposes a few restrictions on the finite sites model: a mutation might be gained once, but can be lost multiple times per character. The *k*-Dollo parsimony model specifies that such a mutation might be lost at most k times. This paper (and the next one) concern the k-Dollo model. In particular, (*El-Kebir et al. 2020*)[4] describes an algorithm, SPhyR, that accepts as input a binary matrix of character states and then proceeds to reconstruct a corresponding k-Dollo phylogeny (if, in fact, such a tree exists for that input).

Definition 11 *A **k-Dollo phylogeny** is a rooted, node-labeled tree subject to the following constraints:*

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

Let $B \in \{0,1\}^{m \times n}$. Then, a tree T is a k-Dollo phylogeny for B iff T is a k-Dollo phylogeny with m leaves such that each row of B labels exactly one leaf of T. B is a k-Dollo phylogeny matrix if there exists a k-Dollo phylogeny T for B.

The problem of constructing this tree, the k-DP problem, assumes perfect data. In reality, we must account for false positives and negatives and missing entries. We'll elide some of the details of the correction process for these matrices for the sake of this survey; importantly, the resulting *perfect phylogeny matrix* A allows us to construct a *perfect phylogeny*, for which each row of A labels exactly one leaf of T. The criterion for such a matrix A is below:

Theorem 12 ***Perfect Phylogeny Theorem*** *A binary matrix 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). We also refer to this as the* ***three gametes condition.***

Definition 13 *Additionally, given an input matrix $B \in \{0,1\}^{m \times n}$, matrix A is a k-completion of B provided 1) $a_{p,c} \in \{0, ..., k+1\} \setminus \{1\}$ iff $b_{p,c} = 0$, and 2) $a_{p,c} = 1$ iff $b_{p,c} = 1$.*

The existence of such a completion matrix A is an equivalent condition for the existence of a k-Dollo phylogeny T for B.

5.2 An integer linear program for k-DP

As before, let B be an $m \times n$ binary input matrix, and let $k \in \mathbb{N}$ be the maximum number of losses per character.

We introduce constraints to ensure that each entry of the output matrix A is 1 iff a mutation gain was recorded in the original input matrix entry:

$$a_{p,c,i} \in \{0,1\}, \forall p \in [m], c \in [n], i \in \{0, ..., k+1\} \tag{6}$$

$$\sum_{i=0}^{k+1} a_{p,c,i} = 1, \forall p \in [m], c \in [n] \tag{7}$$

$$\tag{8}$$

...and additional constraints to ensure that A is, in fact, a k-completion of B, as described in **Definition 13**:

$$a_{p,c,1} = 0, \forall p \in [m], c \in [n] \text{ s.t. } b_{p,c} = 0 \tag{9}$$

$$a_{p,c,1} = 1, \forall p \in [m], c \in [n] \text{ s.t. } b_{p,c} = 1 \tag{10}$$

$$\tag{11}$$

We also include a symmetry-breaking constraint (but will omit it here, but the sake of concision). Further, we recall from our previous definition of the **Perfect Phylogeny Theorem** that there are certain pairs of values which constitute forbidden submatrices. For the sake of concision, we won't list them exhaustively here; only one of these constraints is written below, for indication:

$$a_{p,c,i_1} + a_{p,d,0} + a_{q,c,0} + a_{q,d,j_1} + a_{r,c,i'_1} + a_{r,d,j'_1} \leq 5 \tag{12}$$

Finally, here is our objective function:

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

...in brief, we seek to minimize the maximum number of losses across all characters, with the knowledge that at most k losses are permitted per character. The authors note that a naive implementation

of this ILP doesn't scale well to large datasets (the number of variables here is $O(mnk)$ and the number of constraints is $O(m^3n^2k^4)$). One of SPhyR's major contributions is that it is able to solve the k-DP problem in a matter of seconds.

6 Phylogeny Inference with CNA and SNV Data

We mentioned, in our introduction, that SNVs and CNAs are both sources of mutation loss in sequencing data. Algorithms developed in this paper (SCARLET) and the proceeding publication (section 7) attempt to account for both sources of mutation loss. In particular, SCARLET uses CNA data to support or refute the loss of a mutation at a locus. This constraint is vitally important to the finite-sites model approach to phylogeny reconstruction, which usually permits multiple possible phylogenetic trees corresponding to an input matrix.

Copy number profiles indicate areas affected by loss or amplification (i.e., CNAs) along the genome, and are used to generate two inputs for SCARLET: 1) loss sets indicating mutations affected by deletions, and 2) a copy-number tree that determines relationships between cells as described by the copy number profile.

The SCARLET algorithm has three primary features, which we'll discuss in greater depth. (The following is quoted, in part, from (*Satas et al*., 2020))[14]:

- a loss-supported phylogeny model, which constrains mutation loss to loci where there has also been a corresponding decrease in copy number;
- an algorithm that computes a loss-supported phylogeny by refining a phylogenetic tree derived from copy-number data only;
- and maximum-likelihood inference of SNVs using a probabilistic model of observed read counts in single-cell data.

Whereas the k-Dollo model described in the preceding section limited the number of mutation losses per character to k, the loss-supported model, in this case, constrains mutations losses with sets of supported losses as dictated by CNAs at those loci. This brings us to the following problem statement:

6.1 Loss-Supported Refinement Problems

Given a copy number tree T, a copy-number profile vector $c = [c_v]_{v \in V(T)}$, a mutation matrix $B = [b_v]_{v \in L(T)}$ and supported loss sets $\mathcal{L}$, the LSR problem asks us to find a refinement T' of a copy-number tree T such that T' is a loss-supported phylogeny.

The authors present a recursive algorithm to solve LSR from leaves to root. SCARLET itself addresses a variant of LSR, called the maximum-likelihood loss-supported refinement problem (ML-LSR), that extends the LSR algorithm. For input to ML-IDP, we define a ternary matrix $\bar{B'_v}$ per vertex. Our ILP computes the maximum likelihood submatrix $B*$ at each of these vertices such that B'_v admits an incomplete directed perfect phylogeny. The linear objective for this problem maximizes likelihood of observing a given subtree, conditioned on our read count matrices X and Y.

$$\text{maximize} \sum_{w,a} b'_{w,a} \cdot C_{w,a} \tag{13}$$

$$\text{subject to} \tag{14}$$

$$F_{a,b} + G_{a,b} + H_{a,b} \leq 2 \text{ for all } a, b \tag{15}$$

$$b'_{w,a} + b'_{w,b} - 1 \leq F_{w,a,b} \leq \min(b'_{w,a}, b'_{w,b}) \text{ for all } a, b, w \tag{16}$$

$$\max_w F_{w,a,b} \leq F_{a,b} \leq \sum_w F_{w,a,b} \text{ for all } a, b \tag{17}$$

$$-b'_{w,a} + b'_{w,b} \leq G_{w,a,b} \leq \min(1 - b'_{w,a}, b'_{w,b}) \text{ for all } a, b, w \tag{18}$$

$$\max G_{w,a,b} \leq G_{a,b} \leq \sum_w G_{w,a,b} \text{ for all } a, b \tag{19}$$

$$b'_{w,a} - b'_{w,b} \leq H_{w,a,b} \leq \min(b'_{w,a}, 1 - b'_{w,b}) \text{ for all } a, b, w \tag{20}$$

$$\max H_{w,a,b} \leq H_{a,b} \leq \sum_w H_{w,a,b} \text{ for all } a, b \tag{21}$$

$$\tag{22}$$

...where $C_{w,a} = \log \Pr(x_{w,a} | y_{w,a}, b_{w,a} = 1) - \log \Pr(x_{w,a} | y_{w,a}, b_{w,a} = 0)$

and the variables F, G, and H enforce the three-gametes condition, as described in the **Perfect Phylogeny Theorem**.

ML-LSR is known to be NP-hard; we'll omit the proof here, but it does proceed from a reduction of the Flip problem[2], which is known to be NP-Complete.

7 Parsimonious Clone Tree Reconciliation in Cancer

This paper [13] focuses on solving the clone identification problem in terms of both single-nucleotide variants (SNVs) and copy-number aberrations (CNAs), which are two types of somatic mutations in tumor cells. Previous works are restricted to using either SNVs or CNAs, but not both.

This work tackles this problem under two formulations, i.e., Parsimonious Clone Reconciliation (PCR) and Parsimonious Clone Tree Reconciliation (PCTR). We will give details for PCR and PCTR later in this section. The inputs and outputs to the problem are illustrated in figure 1.

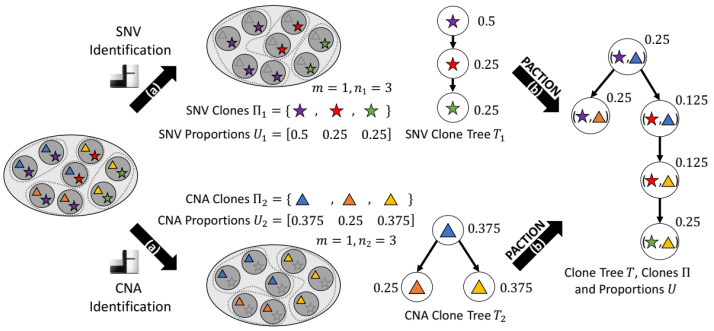

Figure 1: Overview of the PCR and PCTR problem.

7.1 Parsimonious Clone Reconciliation (PCR)

Input Suppose a tumor is composed of a set Π of n clones, which occur in m samples at varying proportions. We represent the proportions of clones in each sample as a $m \times n$ matrix, defined as follows:

$$U \in \mathbb{R}^{m\times n}, s.t. \begin{cases} u_{p,l} \geq 0, \forall p, l \\ \sum_{l=1}^{n} u_{p,l} = 1, \forall p. \end{cases} \tag{23}$$

The proportions of the Π_1-clones (i.e. SNVs) and Π_2-clones (i.e. CNAs) are given by the $m \times n_1$ proportion matrix U_1 and the $m \times n_2$ proportion matrix U_2, respectively.

Target We want to find the unique clones Π given Π_1,Π_2,U_1,U_2 as input. Recall that Π is a partition of all tumor cells induced by the combination of both the two features, whereas Π_1 and Π_2 are partitions induced by each feature in isolation. Therefore we have $\Pi \subseteq \Pi_1 \times \Pi_2$, and our target is to find the Π with smallest size satisfying the following proportions constraints:

$$\text{minimize } |\Pi|, s.t. \begin{cases} u^{(1)}_{p,i} = \sum_{\ell:\pi_1(\ell)=i} u_{p,\ell}, \forall p \in [m], i \in [|\Pi_1|] \\ u^{(2)}_{p,j} = \sum_{\ell:\pi_2(\ell)=j} u_{p,\ell}, \forall p \in [m], i \in [|\Pi_2|] \end{cases} \tag{24}$$

where π_1 and π_2 are projection functions from a clone to its individual features (SNVs or CNAs).

Solution The authors of [13] prove that the PCR problem is NP-hard. Therefore, they propose to use a mixed integer linear programming (MILP) formulation to solve the PCR problem. The MILP is formulated as follows:

Variables (1) binary variables $x_{i,j} \in \{0, 1\}$ for each i in Π_1 and j in Π_2, indicating if the clone configuration (i, j) is in Π; (2) continuous variables $u_{p,i,j} \in [0, 1]$, representing the proportion of clone (i, j) in sample $p \in [m]$.

Constraints

$$\begin{aligned} u_{p,i,j} &\leq x_{i,j} \quad \forall p \in [m], i \in [n_1], j \in [n_2] \\ \sum_{j=1}^{n_2} u_{p,i,j} &= u^{(1)}_{p,i} \quad \forall p \in [m], i \in [n_1], \\ \sum_{i=1}^{n_1} u_{p,i,j} &= u^{(2)}_{p,j} \quad \forall p \in [m], j \in [n_2]. \end{aligned} \tag{25}$$

The first constraint enforces $u_{p,i,j} = 0$ if $(i, j) \notin \Pi$, and the next two constraints are the proportions constraint.

Objective $\min \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} x_{i,j}$

7.2 Parsimonious Clone Tree Reconciliation (PCTR)

Input Compared to the PCR problem, the PCTR problem additionally takes two clone trees T_1 and T_2 corresponding to SNVs and CNAs as input, and outputs a refinement tree T. Both T_1 and T_2 are rooted trees with vertices representing the clones and edges representing the mutation event that changes one clone (parent) to the other (child).

Target The *refinement tree* is defined as follows:

(i) for each edge $(i, i') \in E(T_1)$ there exists exactly one $j \in \Pi_2$ such that $((i, j), (i', j)) \in E(T)$,
(ii) for each edge $(j, j') \in E(T_2)$ there exists exactly one $i \in \Pi_1$ such that $((i, j), (i, j')) \in E(T)$,
(iii) for each $((i, j), (i', j')) \in E(T)$, it holds that $(i, i') \in E(T_1)$ and $j = j'$, or $(j, j') \in E(T_2)$ and $i = i'$.

Note that the proportions constraint may not be satisfied given the above refinement tree constraints. Therefore we minimize the following distance between the observed proportion matrices for SNVs and CNAs and the ones induced by the refinement tree:

$$J(U, U_1, U_2) = \sum_{p=1}^{m} \sum_{i=1}^{n_1} \left| u^{(1)}_{p,i} - \sum_{\ell:\pi_1(\ell)=i} u_{p,\ell} \right| + \sum_{p=1}^{m} \sum_{j=1}^{n_2} \left| u^{(2)}_{p,j} - \sum_{\ell:\pi_2(\ell)=i} u_{p,\ell} \right|. \tag{26}$$

Solution The authors prove that the PCTR problem is also NP-hard. Therefore, they propose another MILP problem to solve the PCTR problem. The MILP problem basically writes the above refinement constraints in the form of linear equations. Due to limited space, we refer the reader to Section 4.2 in [13] for full details.

7.3 Conclusion

The paper proposes solutions to two problems, PCR and PCTR, to solve the clone detection problem using the combination of SNVs and CNAs. The authors prove that both problems are NP-hard and can be approximated by two MILP problems. Experimental results on synthetic and real datasets show that the proposed method achieves better performance compared to existing methods [9] that manually annotate SNV trees with CNA events.

8 Detecting Evolutionary Patterns of Cancers Using Consensus Trees

This paper [3] focuses on detecting the repeated patterns of tumor evolution among **multiple patients**. The problem takes a family $\{T_1, \ldots, T_n\}$ of sets of patient clone trees as input, and aims to simultaneously cluster n patients into k sub-types of evolutionary trajectories $\{R_1, \ldots, R_k\}$ and select a clone tree for each patient. Each T_i is a set that contains multiple possible clone trees for the i-th patient: $T_i = \{S_i^1, \ldots, S_i^k\}$.

8.1 Preliminaries: Distances between Trees

To find a consensus tree that minimizes the average distance to all members in the cluster, one needs to define the distance metric between two trees. The authors use the normalized parent–child distance, which is the parent–child distance divided by twice the size of the vertex set $\Sigma = |V(T) \cup V(T')|$:

$$d_N(T, T') = \frac{|E(T) \Delta E(T')| + |V(T) \Delta V(T')|}{2\Sigma} \tag{27}$$

Where Δ is the set difference operator, and $E(\cdot)$ and $V(\cdot)$ represent the edges and vertices of the graph, respectively.

8.2 Preliminaries: the Single Consensus Tree (SCT) Algorithm

The SCT problem has been tackled and solved in [8]. The SCT algorithm takes a set of trees $S_1, \ldots, S_n$ as input, and outputs a consensus tree R with vertex set R such that the total normalized parent–child distance $\sum_{i=1}^{n} d(S_i, R)$ is minimized

8.3 Problem Definition

The authors define the following *Multiple Choice Consensus Tree* (MCCT) problem:

> Given a family $\{T_1, \ldots, T_n\}$ of sets of patient trees composed of subsets of mutations R and integer $k > 0$, find:
> (i) a single tree $S_i \in T_i$ for each patient $i \in [n]$.
> (ii) a clustering $\sigma(\cdot) : [n] \to [k]$ of patients into k clusters.
> (iii) a consensus tree R_j for each cluster $j \in [k]$ such that the total normalized parent–child distance $\sum_{i=1}^{n} d_N\left(S_i', R_{\sigma(i)}\right)$ is minimized.

8.4 The RECAP Algorithm

The authors first prove that the above MCCT problem is NP-hard. They then propose a solution to the problem using a heuristic algorithm, called Revealing Evolutionary Consensus Across Patients (RECAP).

Algorithm 1 RECAP

Input $T = \{T_1, \ldots, T_n\}$; integer $k > 0$; .
Initialize: $S = \{S_1, \ldots, S_n\} \leftarrow$ randomly select a clone tree for each patient from T_i; $\sigma \leftarrow$ randomly assignment of each patient to one of the k clusters; $R = \{R_1, \ldots, R_n\} \leftarrow$ running the SCT algorithm on each clusters assigned by σ.
while not finished **do**
  for $i = 1$ to n **do**
    $S_i, \sigma(i) \leftarrow \text{argmin}_{T \in \mathcal{T}_i, j \in [k]}\, d\,(T, R_j)$ ▷ Find a min distance tree and cluster
  end for
  for $i = 1$ to k **do**
    $R_i, \sigma(i) \leftarrow$ updating consensus tree by running the SCT algorithm on each cluster
  end for
end while
Output S, R, σ

8.5 Monotonicity of the RECAP Algorithm

While this algorithm is a heuristic, the total parent–child score is monotonically decreasing with each iteration. In step (i) within the while loop, the tree selection and cluster assignment is only changed if such a reassignment decreases the score; in step (ii), the updated consensus tree is guaranteed to be optimal and so can only decrease the score.

In practice, one can run the RECAP algorithm multiple times with different random seeds and output the result with minimum total distance to achieve better performance.

8.6 Conclusion

The authors focus on finding common mutation patterns among multiple patients by clustering patients into sub-types and finding a consensus tree within each of the clusters. Because the problem is NP-hard, the authors propose to solve it with a heuristic algorithm called RECAP, and show that its total objective decreases monotonically. Experimental results demonstrate that RECAP can handle larger-scale problems (i.e. more patients) and achieve better results compared to previous works [1, 12].

9 Conclusion

In this survey, we discussed six recent papers that attempt to solve the cancer evolution problem under different setups. Section 3 takes bulk sequencing data as input described by the frequency matrix and seeks to output a clone tree. Section 4 takes single-cell CNA copy-number profiles as input, and builds a copy-number tree from them. Section 5 uses single-cell SNV data as an input instead, and corrects for loss and error in clinical data. Section 6 proposes to solve the phylogeny inference problem with both CNA and SNV data and achieves better results compared to using a single data modality. Section 7 further extends this objective by using the parsimonious tree constraints in addition to a proportion matrix. Finally, section 8 expands the scope from a single tumor to finding a consensus among multiple patients.

Approximation algorithms such as LP-rounding and heuristic search have been widely applied in this literature, as optimization problems are often NP-hard. Our survey demonstrates the potential of applying approximation algorithms to large-scale, real-world problems such as cancer evolution.

References

[1] Giulio Caravagna, Ylenia Giarratano, Daniele Ramazzotti, Ian Tomlinson, Trevor A Graham, Guido Sanguinetti, and Andrea Sottoriva. Detecting repeated cancer evolution from multi-region tumor sequencing data. *Nature methods*, 15(9):707–714, 2018.

[2] D. Chen, O. Eulenstein, D. Fernandez-Baca, and M Sanderson. Minimum-flip supertrees: complexity and algorithms. *IEEE/ACM Transactions in Computational Biology and Bioinformatics*, 3(1):165–173, 2006.

[3] Sarah Christensen, Juho Kim, Nicholas Chia, Oluwasanmi Koyejo, and Mohammed El-Kebir. Detecting evolutionary patterns of cancers using consensus trees. *Bioinformatics*, 36(Supplement_2):i684–i691, 2020.

[4] Mohammed El-Kebir. Sphyr: tumor phylogeny estimation from single-cell sequencing data under loss and error. *Bioinformatics*, 34(1):671–679, 2018.

[5] Mohammed El-Kebir, Layla Oesper, Hannah Acheson-Field, and Benjamin J Raphael. Reconstruction of clonal trees and tumor composition from multi-sample sequencing data. *Bioinformatics*, 31(12):i62–i70, 2015.

[6] Mohammed El-Kebir, Benjamin J Raphael, Ron Shamir, Roded Sharan, Simone Zaccaria, Meirav Zehavi, and Ron Zeira. Complexity and algorithms for copy-number evolution problems. *Algorithms for Molecular Biology*, 12(1):1–11, 2017.

[7] Les R Foulds and Ronald L Graham. The steiner problem in phylogeny is np-complete. *Advances in Applied mathematics*, 3(1):43–49, 1982.

[8] Kiya Govek, Camden Sikes, and Layla Oesper. A consensus approach to infer tumor evolutionary histories. In *Proceedings of the 2018 Acm international conference on bioinformatics, computational biology, and health informatics*, pages 63–72, 2018.

[9] Gunes Gundem, Peter Van Loo, Barbara Kremeyer, Ludmil B Alexandrov, Jose MC Tubio, Elli Papaemmanuil, Daniel S Brewer, Heini ML Kallio, Gunilla Högnäs, Matti Annala, et al. The evolutionary history of lethal metastatic prostate cancer. *Nature*, 520(7547):353–357, 2015.

[10] Dan Gusfield. Efficient algorithms for inferring evolutionary trees. *Networks*, 21(1):19–28, 1991.

[11] Dan Gusfield. Algorithms on stings, trees, and sequences: Computer science and computational biology. *Acm Sigact News*, 28(4):41–60, 1997.

[12] Sahand Khakabimamaghani, Salem Malikic, Jeffrey Tang, Dujian Ding, Ryan Morin, Leonid Chindelevitch, and Martin Ester. Collaborative intra-tumor heterogeneity detection. *Bioinformatics*, 35(14):i379–i388, 2019.

[13] Palash Sashittal, Simone Zaccaria, and Mohammed El-Kebir. Parsimonious clone tree reconciliation in cancer. In *Leibniz International Proceedings in Informatics, LIPIcs*, volume 201, page 9. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2021.

[14] Gryte Satas, Simone Zaccaria, Geoffrey Mon, and Benjamin J Raphael. Scarlet: Single-cell tumor phylogeny inference with copy-number constrained mutation losses. *Cell Systems*, 10(1):323–332, 2020.