Reconstructing Clonal Trees From Multi-Sample Sequencing Data

Hannah Acheson-Field
Senior Honors Thesis
Sc.B Computational Biology
Computer Genomics Tack
Advisor: Ben Raphael
Second Reader: David Laidlaw

April 15, 2015

Abstract

Cancer is caused by somatic mutations, changes in an individual's genome that occur after conception and are not passed to one's offspring. Somatic mutations can accumulate over an individual's lifetime. The clonal theory of cancer states that each cell within a tumor descends from the same cell. A *clone* is a group of cells that descended from a common ancestor. Certain, rare, advantageous mutations may give that cell certain evolutionary advantages. These mutations may lead to a clonal expansion, a group of cells that are descendants, thus would carry that mutation. The occurrence of cells within the same tumor having different genotypes is called intra-tumor heterogeneity. Recently, high throughput sequencing has allowed one to analyze the intra-tumor heterogeneity of tumor samples.

Similar to species evolution, clonal evolution can also be modeled using phylogenetic trees. This model can be most easily applied to data obtained from single-cell sequencing samples, when all samples observed contain only the types of mutations associated with one cell. However, most cancer sequencing studies use bulk-sequencing, in which data from multiple samples are observed as a mixture, thus inferring the evolutionary history is not trivial.

We formalize and implement an algorithm that infers the evolutionary history of tumors that had been measured using bulk-sequencing. We then implement two approaches, Perfect Phylogeny Mixture Problem (PPM) and Greedy Minimum Split Rows (GMSR) Algorithm to the same problem. After running comparisons with pre-existing approaches, PhyloSub and CITUP, we found that PhyloSub and CITUP outperformed both PPM and GMSR. This preliminary work on inferring clonal evolution with discretized mutation matrices led to work on AncesTree, an algorithm that uses a similar formulation but uses different input and clusters mutations differently.

Contents

1 Introduction

1.1 Biology Background

Deozyribonucleic Acid (DNA) are macro molecules that contain all genetic information. DNA is composed of four bases: Adenine (A), Cytosine (C), Guanine (G), and Thymine (T). In an individual, DNA is arranged in a double helix that lie in structures, called chromosomes. An individual's genome describes all of that individual's DNA. Certain parts of the genome contain *genes*, which are heritable units of information that encode proteins. Proteins are responsible for a wide variety of cell function, activity, and regulation.

Genes are converted to proteins first through transcription, then through translation. During transcription, DNA is converted to *ribonucleic Acid (RNA)*, which is composed of Adenine (A), Cytosine (C), Guanine (G), and Urasil (U). During translation, RNA is converted into proteins. Triplet codons, containing three RNA nucleotides, code for a specific amino acid, which are the building blocks of proteins. Because the amino acid sequence, thus protein function, is dependent upon each nucleotide, changes to a nucleotide sequence will often affect cell behavior. An *allele* is a variant a specific position, or *locus*, of one's genome [9].

During DNA sequencing, the order of nucleotides in DNA is determined. Because the genome contains a huge number of nucleotides, it is impossible to scan the genome nucleotide by nucleotide. Instead, small fragments, or sequencing *reads*, are aligned against reference genome. The sequencing *coverage* refers to how many reads align at a certain position.

Throughout this work, we focus on single nucleotide variants (SNVs), which are changes to an allele at a specific locus. The *variant allele frequency (VAF)* is the fraction of nucleotides that contain a specific variant allele. If there is a high VAF at a specific locus, a mutation likely occurred there. When we say that a variant allele was discretized or that a matrix containing variant allele frequencies was discretized, we mean that above a certain threshold, we replace the frequency with a 1 and assume that a mutation occurred at that position.

1.2 Cancer Background

Cancer is caused by somatic mutations, changes in an individual's genome that occur after conception and are not passed to one's offspring. Somatic mutations can accumulate over an individual's lifetime. There are both passenger mutations, which have no effect on a clone's survival, and driver mutations, which are give that cell certain evolutionary advantages. The clonal theory of cancer states that each cell within a tumor descends from the same cell. A *clone* is a group of cells that descended from a common ancestor. These mutations may lead to a clonal expansion, a group of cells that are descendants, thus would carry that mutation [12]. The occurrence of cells within the same tumor having different genotypes is called intra-tumor heterogeneity. Recently, high throughput sequencing has allowed one to analyze the intra-tumor heterogeneity of tumor

samples [5].

Understanding which somatic mutations occurred early on can create more effective treatment options. For instance, if oncologists target an early driver mutation, they remove a large part of the tumor mass, and if they target founder mutations, they remove the entire tumor mass. Oncologists may also choose to target different clonal subpopulations, thus understanding clonal evolution may help devise these more personalized treatments [1].

1.3 Modeling Clonal Evolution with Phylogenetic Trees

Similar to species evolution, tumor evolution can also be modeled using phylogenetic trees. In the case of species evolution, the root would be the common ancestor, the leaves would be the taxa, and the characters can have two states, either normal or mutated. In the case of tumor evolution, the tree leaves would be tumor samples, the root would be the founder cell, and the characters would be mutations, either normal or mutated. A single nucleotide mutation is represented by a 1, if that position is mutated, or a 0 otherwise. The root, the entirely unmutated, founder cell would therefore contain a 0 at every position.

Gusfield introduces Perfect Phylogeny matrices as a way to model clonal expansions. The clonal expansion can be modeled by Tree T defined by a matrix $M \in \{0,1\}^{m \times n}$, where m is the number of samples and n is the number of loci that have been mutated. If T exhibits a perfect phylogeny, then each of the n mutations must label exactly one edge, each of the m samples must correspond to exactly one leaf, and for any sample s, the path from the root to s must contain each mutation that s contains [4].

Because the genome is large, mutations are rare, and mutations at the same locus are event more rare, the infinite sites assumption, which states that an individual mutation event can only occur once, can be used in the analysis of clonal evolution. The 4-gametes condition follows from the infinite sites assumption and explains that a perfect phylogeny exists *iff* no two columns contain (0,0), (0,1), (1,0), and (1,1) [8]. Because a mutation can only occur once, as we assume in the infinite sites assumption, the 4-gametes condition will always be T in modeling clonal expansions. The Perfect Phylogeny Theorem follows from the 4-gametes condition and states that a binary matrix M is a perfect phylogeny matrix if and only if there is no pair of columns that contain the pairs (0,1), (1,0), and (1,1). Because the root will always contain the pair (0,0), having the other three pairs will always violate the 4-gametes condition. Similarly, two columns are said to be in conflict if they contain the pairs (0,1), (1,0), and (1,1). Figure 1(D) provides an example of two columns that are in conflict.

The second part of the Perfect Phylogeny Problem constructs a Perfect Phylogeny Tree given the input matrix is conflict free. Gusfield provides an algorithm to construct this tree. After sorting columns in the matrix based on the number of 1s, i.e. the most common mutation would be moved to the left-most position. The tree is then constructed by systematically adding each sample to the tree by constructing an edge for each mutation. If a mutation is first seen

in that sample, a new edge is constructed, otherwise that edge becomes part of the path from the root to the sample placed in the tree [4].

1.4 Previous Work

Modeling clonal evolution with phylogenetic trees can be most easily applied to data obtained from unmixed, single cells, when all samples observed contain only the mutations associated with one cell. However, the data obtained in most cancer sequencing studies is obtained from bulk sequencing, thus the data observed from samples that contain a mixture of cells. Reconstructing the history of somatic mutations is more difficult in this case because the data obtained can be harder to deconstruct. For instance, the binary data associated with the mutation matrix, whether a mutation exists in a particular sample at a particular position, often contains conflicts because it is unclear in which sample(s) the mutation exists. Figure 1 provides an example of when conflicts are created through multi-sample sequencing.

There are many different approaches that infer clonal histories based on multi-sample sequencing data. We compare to both PhyloSub and CITUP. PhyloSub uses Bayesian inference to determine phylogenetic trees that are consistent with variant allele frequency data [7] and CITUP enumerates all possible trees, then picks the best tree using Bayesian information criterion (BIC). Malikic et al. also introduce an iterative method in which they use a heuristic based on the Expectation Maximization (EM) Algorithm to infer mutation assignment [10].

Clomial determines tumor frequencies but does not infer the evolutionary relationship. Therefore, in experimentation, the matrices obtained are rarely perfect phylogeny matrices, thus comparing generated solutions to Clomial solutions was usually not feasible [15]. LICHeE finds the set of lineage trees that are consistent with its VAF data but only provides an interface with no access to matrices actually obtained. We could not compare directly to LICHeE [14].

1.5 Contributions

This work began as a continuation of a project by Iman Hajirasouliha and Ben Raphael, titled "Reconstructing mutational history in multiply sampled tumors using perfect phylogeny mixtures," in which they give an algorithm using Split Row operations to infer clonal evolution [6]. We first formalize the problem presented in this paper as the Perfect Phylogeny Mixture Problem (PPM). Second, we give further detail in how to implement their algorithm Minimum Split Rows Algorithm by introducing the Greedy Minimum Split Row (GMSR) Algorithm and Exhaustive Minimum Split Row (EMSR) Algorithm. After implementing these algorithms, we find a counter example in which both the GMSR and EMSR algorithm fail. Third, we provide an ILP formulation to solve the same problem for the Perfect Phylogeny Mixture Problem (PPM). In the Results Section, we show comparisons between PhyloSub, CITUP, and these two methods that use discretized input matrices. Finally, we show how both

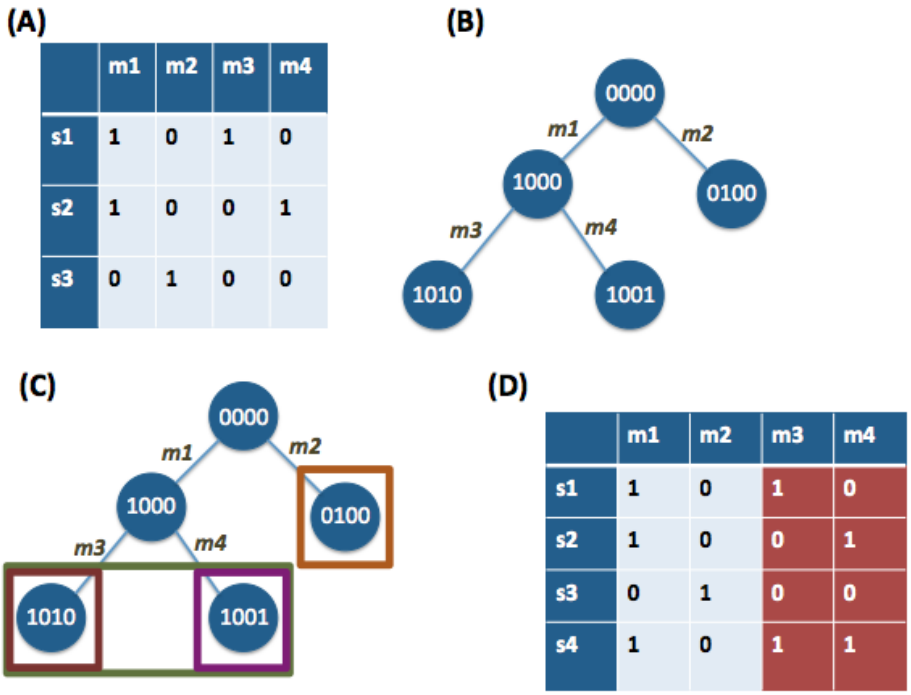


Figure 1: **An Example of Clonal Evolution Modeled By Phylogenetic Trees** (A) An example of a Perfect Phylogeny Matrix with 3 samples and 4 mutations. This matrix is a perfect phylogeny matrix and thus is conflict-free. (B) The Perfect Phylogeny Tree corresponding the Matrix in A. (C) When high throughput sequencing is used, the data generally obtained is in the form of samples that contain a mixture of populations. For example, the green sample (s4) is a mixture of s1 and s2. The data that would be obtained from this sample would be observed as the bitwise or of these two samples. (D) The corresponding matrix containing the mixture of s1 and s2 in s4. The corresponding matrix is not a Perfect Phylogeny Matrix because m3 and m4 contain a conflict because they contain the pairs (1,0), (0,1), and (1,1). This is an example of a matrix that could be obtained from multi-sample sequencing studies. It would be difficult to reconstruct the clonal evolution as it would be hard to determine the actual populations and thus the actual phylogenetic history.

methods using discretized input matrices served as preliminary work and the basis for AncesTree, an algorithm that discerns clonal evolution using a similar equation but uses the original VAF data instead of discretizing variant allele frequencies input.

2 Discretized Methods

2.1 Perfect Phylogeny Mixture Problem (PPM)

The Perfect Phylogeny Mixture Problem (PPM) provides a more formal definition to infer a conflict-free matrix that describes the phylogeny of a tumor sample given a discretized matrix containing samples that contain information from multiple cross sections of a tumor. Work on PPM formalization was done with Mohammed El Kebir and Layla Oesper.

2.1.1 Problem Definition

Given a matrix $\mathbf{A} \in \{0,1\}^{m \times n}$, the i^{th} row of $\mathbf{A}$ will be denoted as $\mathbf{a}_{i\cdot}$ and the j^{th} column of $\mathbf{A}$ as $\mathbf{a}_{\cdot j}$. Two columns $\mathbf{a}_{\cdot j}$ and $\mathbf{a}_{\cdot k}$ are *disjoint* if $a_{ij} \wedge a_{ik} = 0$ for all $i \in \{1, \ldots, n\}$.

Definition 1 *Given matrices* $\mathbf{A} \in \{0,1\}^{m \times n}$ *and* $\mathbf{B} \in \{0,1\}^{n \times p}$, $\mathbf{A} \otimes \mathbf{B} = \mathbf{C}$, *where* $\mathbf{C} \in \{0,1\}^{m \times p}$ *such that* $c_{ij} = \vee_{k=1}^{n}(a_{ik} \wedge b_{kj})$.

Intuitively, $\mathbf{A} \otimes \mathbf{B}$ yields a $\mathbf{C}$ whose rows are the "bitwise or" of a subset of rows in $\mathbf{B}$. The entries in $\mathbf{A}$ determine which rows to use in the "bitwise or" operation. For instance $a_{ik} = 1$ would mean that the i^{th} row of $\mathbf{C}$ is a bitwise or of a set of rows in $\mathbf{B}$, including the k^{th} row of $\mathbf{B}$.

Definition 2 *Given Usage Matrix* $\mathbf{A} \in \{0,1\}^{m \times n}$, *Basis Matrix* $\mathbf{B} \in \{0,1\}^{n \times p}$ *and Input Matrix* $\mathbf{C} \in \{0,1\}^{m \times p}$ *we say that* $(\mathbf{A}, \mathbf{B})$ generates $\mathbf{C}$ *if and only if* $\mathbf{A} \otimes \mathbf{B} = \mathbf{C}$.

Definition 3 *The Mixing Graph* G_A *is a bipartite graph that maps every row* $\mathbf{c}_{i\cdot}$ *with a row* $\mathbf{b}_{j\cdot}$. *We label vertices as* $v_{c_{i\cdot}}$ *and* $v_{b_{j\cdot}}$. *There is an edge between* $v_{c_{i\cdot}}$ *and* $v_{b_{j\cdot}}$ *if and only if* $A_{i,j}$=1.

Definition 4 *Given a binary matrix* $\mathbf{B} \in \{0,1\}^{n \times p}$, *two columns* $\mathbf{b}_{\cdot i}$ *and* $\mathbf{b}_{\cdot j}$ *in* $\mathbf{B}$ *are in* conflict *if and only if there exists three rows* $\mathbf{b}_{r\cdot}, \mathbf{b}_{s\cdot}, \mathbf{b}_{t\cdot}$ *in* $\mathbf{B}$ *such that their* (i, j) *positions are* $(1,1), (0,1), (1,0)$ *respectively.*

Definition 5 *A binary matrix* $\mathbf{B} \in \{0,1\}^{n \times p}$ *is* conflict-free *if and only if it has no pairs of columns in conflict.*

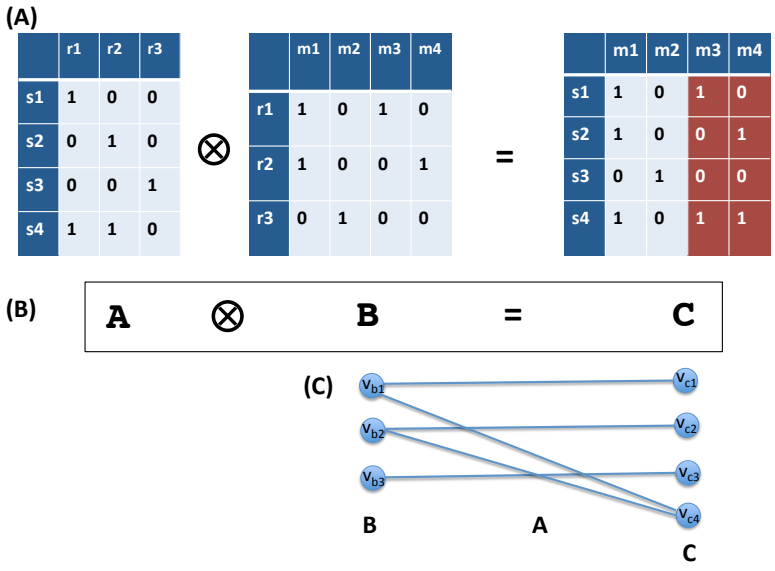


Figure 2: **PPM Problem Definition** (A) Usage matrix A, basis matrix B, and input matrix C. C has two conflicts which are resolved when mixing rows in B. The usage matrix A describes which rows in B to mix for each row in C. (B) The equation that relates A, B, and C. (C) The mixing graph G_A that relates rows in C to the mixture of rows in B that combine to form each row in C. In the ILP Formulation (Section 2.3) we minimize the number of mixings, or the number of edges in the mixing graph.

Minimum Usage Problem Given a binary matrix $\mathbf{C} \in \{0,1\}^{m \times p}$ find a pair $\mathbf{A} \in \{0,1\}^{m \times n}$ and conflict-free $\mathbf{B} \in \{0,1\}^{n \times p}$ such that $(\mathbf{A}, \mathbf{B})$ generates $\mathbf{C}$ and $\sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij}$ is minimal.

In both implementations described below, we assume that every sample in C will contain at least one mutation, or that $\sum_j C_{ij} \geq 1 \; \forall i$. This is because the root node will be unmutated and is implicitly in each clonal history. If two columns in C are identical, we also cluster mutations together by using only one instance of this column.

2.2 Minimum Split Row Algorithm

Hajirasouliha and Raphael formulate the Minimum Split Row Problem as a way to solve the Perfect Phylogeny Mixture problem. They give a graph theoretic algorithm for deconvoluting a discretized input mutation matrix M. Note that M is the same matrix as C as described above, but we will use the same notation

as Hajirasouliha and Raphael did for the remainder of the section. The premise of this Minimum Split Row Algorithm relies on Split-Row Operations, which they define as follows [6].

Definition 6 *Split Row Operation Given a row* $\mathbf{r}$ *of a mutation matrix* M*, a* ***split-row operation*** S_r *on* M *is the following transformation: replace* $\mathbf{r}$ *with* k *new rows* $\hat{r}_1, ..., \hat{r}_k$ *such that if* $\mathbf{r}_i$*=1, then there will be a 1 in at least one of the new, split rows* $\hat{r}_1, ..., \hat{r}_k$ *at position* i*. In other words,* $\mathbf{r}$ *is the bitwise OR of* $\hat{r}_1, ..., \hat{r}_k$*.*

Their graph theoretic algorithm for deconvoluting Mutation Matrix M is based upon splitting rows to resolve conflicts in the graph. They define a conflict graph as follows.

Definition 7 *In* ***Conflict Graph*** $G_{M,\mathbf{r}}$*, a node is added for every entry of* $\mathbf{r}$ *that is a 1, and an edge is added between* v_i *and* v_j *iff* c_i *is in conflict with* c_j*, where* c_i *and* c_j *are columns in* M*. Similarly,* G_M *is the conflict graph for all rows in* M *and an edge is added between* v_i *and* v_j *iff* c_i *is in conflict with* c_j*.*

A graph coloring is a label given to each vertex such that no two adjacent vertices have the same label or "color." Likewise, the chromatic number ($\chi(G)$) lowest possible number of colors needed to color a graph. Figure 3(C-D) provide examples of graph coloring.

In the description of their algorithm, each row is split into each row into k new rows, where k=$\chi(G_{M,r})$. In their approach, they assume that any minimum vertex coloring, a coloring that uses k colors, can be used in their algorithm. However, graph coloring is not a trivial problem, and they do not provide information on how to color their graphs. Therefore, we introduced two forms of the Minimum Split Row Algorithm, both of which use different methods to color each $G_{M,\mathbf{r}}$.

Greedy Minimum Split Row Algorithm

The Greedy Minimum Split Row Algorithm (GMSR) colors $G_{M,\mathbf{r}}$ using a greedy algorithm. To color this graph, assign the lowest available coloring to each column in the graph. Though the GMSR is efficient, oftentimes the GMSR will not find a minimal coloring.

Exhaustive Minimum Split Row Algorithm

The Exhaustive Minimum Split Row Algorithm (EMSR) colors $G_{M,\mathbf{r}}$ exhaustively by obtaining all possible graph colorings for each $G_{m,r}$. This approach is guaranteed to find a minimum vertex coloring. However, this method does not scale well and for large n.

Hajirasouliha and Raphael note that after splitting rows based on the graph coloring, it may be necessary to fix additional conflicts that were created. In order to fix conflicts, they use the containment graph, defined below.

Definition 8 *The **Containment Graph** H_M is a directed acyclic graph (DAG), where there is an edge $i \to j$ if and only if column i contains column j. Note that a column i contains column j if for every row k, $M_{k,i} \geq M_{k,j}$.*

To remove conflicts, after rows are split, traverse every node in H_M to find any pair of vertices, where there is an edge $v_i \to v_j$ and there is a 0 at position i but a 1 at position j. If this occurs, replace the 0 with a 1. In Figure 3 (E), I note this shift as 0¿1.

2.2.1 Efficiency

For GMSR, the conflict graphs can be colored in $O(kn)$, where k is the largest of each $\chi(G_{M,r})$, thus the overall algorithm can still run in polynomial time. For EMSR, the conflict graph can be colored in O(k^n), where k is the largest of each of the $\chi(G_{M,r})$, thus the overall algorithm will be O(mk^n), thus the overall algorithm will run in exponential time. On many instances, the size of n often makes this method infeasible even though a minimal coloring is guaranteed.

2.2.2 Algorithm Failure

After implementation, we realized that there are when both the GMSR and the EMSR algorithm fail to find a conflict-free solution. In the method that Hajirasouliha and Raphael give to fix conflicts, new conflicts are created that cannot be fixed with the method provided. For instance, a new conflict between m4 and m5 exists in the final matrix. On simulated data, the GMSR and EMSR failed on 18 instances, and the EMSR failed on 18 (Table 1). On real data, the GMSR failed on real data, the GMSR failed 7 times and the EMSR failed 8 times (Table 2). Note that when I say the EMSR failed, I mean that either no feasible solution was found or that no feasible coloring was found within a given time because n was too large.

Though splitting rows to avoid conflicts provides a neat formulation, ultimately splitting rows independently is not the best approach to this problem because new conflicts can be created elsewhere. In Section 2.3 we formulate an ILP to solve the PPM Problem instead of split-row operations. In doing so, we can ensure that our solutions will be conflict free, which was proved to not be possible in GMSR and EMSR.

2.3 PPM ILP Formulation

Below we introduce an Integer Linear Programming to solve the PPM Problem. Given K, the number of mixes, we have that $j \in \{1, \ldots, K\}$, $i \in \{1, \ldots, m\}$ and

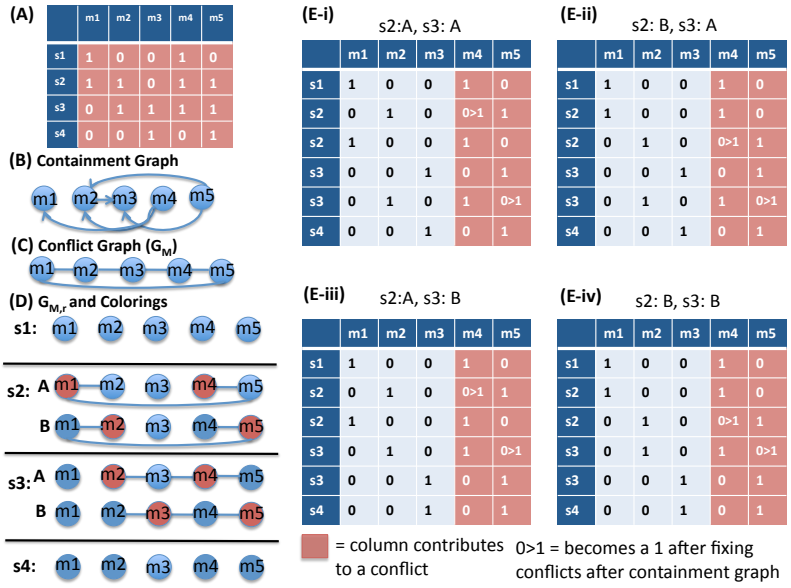


Figure 3: **Counter Example to both the GMSR and EMSR Algorithm** (A) Input mutation matrix M. There are conflicts with column pairs m1-m2, m1-m5, m2-m3, m3-m4, m4-m5. (B) The Containment Graph H_M. (C) The conflict graph G_M that shows all 5 conflicts in M. (D) The Conflict Graphs and colorings for $G_{M,\mathbf{s1}}$, $G_{M,\mathbf{s2}}$, $G_{M,\mathbf{s3}}$, and $G_{M,\mathbf{s4}}$. For s2 and s3 there are two appropriately colored graphs. (E) The split matrix M' that is supposed to be conflict free. E-i to E-iv show the four possible combinations of the EMSR algorithm. E-iv is the example that the GMSR Algorithm would produce. Using the containment graph, two positions are changed from a 0 to 1 at which point they generate new conflicts in m4 and m5.

$k, l \in \{1, \ldots, n\}$.

$$\min \sum_j h_j \tag{1}$$

$$\sum_i \sum_j a_{ij} = K \tag{2}$$

$$d_{kl} \leq b_{jl} - b_{jk} + 1 \qquad \forall j, k \neq l \tag{3}$$

$$f_{kl} \leq 1 - b_{jk} b_{jl} \qquad \forall j, k < l \tag{4}$$

$$d_{kl} + d_{lk} + f_{kl} \geq 1 \qquad \forall k < l \tag{5}$$

$$h_j \geq a_{ij} \qquad \forall i, j \tag{6}$$

$$\sum_{k=1}^{n} 2^k b_{jk} \geq \sum_{k=1}^{n} 2^k b_{(jF)k} + h_j \quad \forall j > 1 \tag{7}$$

$$c_{il} \leq \sum_{l=1}^{K} a_{ij} b_{jl} \qquad \forall i, j \tag{8}$$

$$c_{ij} \geq a_{il} b_{lj} \qquad \forall i, j, l \tag{9}$$

$$a_{ij} \in \{0, 1\} \qquad \forall i, j \tag{10}$$

$$b_{jk} \in \{0, 1\} \qquad \forall j, k \tag{11}$$

$$c_{il} \in \{0, 1\} \qquad \forall i, l \tag{12}$$

$$d_{kl} \in \{0, 1\} \qquad \forall k \neq l \tag{13}$$

$$f_{kl} \in \{0, 1\} \qquad \forall k < l \tag{14}$$

$$h_j \in \{0, 1\} \qquad \forall j \tag{15}$$

$$\tag{16}$$

(1) The objective function is minimizing the number of 'mixes' or the total number of times rows in B are used to generate C. (2-4) These constraints constrain solutions to only show conflict free matrices. d_{kl}=1 if column k is contained in column l and f_{lk}=1 if column l is contained in column k. (5) h represents rows in B that are used to generate C. Constraint 5 ensures that h_j=1 if the j_{th} row in B is used to generate C. (6) This constraint orders rows in B from largest to smallest based on their binary values. (7-8) These constraints ensure that rows in C are the "bitwise or" of rows in B. See Definition 4. Mohammed El Kebir and I implemented the PPM ILP in C++ using CPLEX v12.6.

2.3.1 Enumeration Mode

To run the ILP, we run in enumeration mode, where we begin with setting K as m, the number of samples. We then run the ILP, incrementing K each time, until we find a feasible solution. This ensures that we find the fewest number of mixings. We are guaranteed that the number of mixings can never be less than m, so setting K to m gives us a lower bound on the number of mixes. The

upper bound on the number of mixings will be $||C||_1$. We prove both the lower and upper bounds below.

Lower Bound

Lemma 1 *The minimum number of mixings is at least m.*

Proof. In Section 2.1.1, we assumed that there will be no row that contains all zeros. Therefore, each $\sum_l C_{il} \geq 1 \forall i$. Because $C = A \otimes B$, each row in A must contain at least one 1, or $\sum_j A_{ij} \geq 1, \forall i$. Likewise, the number of mixings, or $\sum_i \sum_j A_{ij} \geq m$.

Upper Bound

Lemma 2 *The maximum number of mixings is at most $||C||_1$.*

Proof. There is always a solution when $B{=}I_n$. If this is the case, every row $C_{i.}$ will be a mixture of $\sum_l C_{il}$ rows of B. Therefore, the total mixings K will be $\sum_i \sum_l C_{il} = ||C||_1$.

3 Results

3.1 Simulated Data

Layla Oesper generated simulated instances using varying values for read coverage, #mutations, #samples, and #clones. These were generated by first partitioning m mutations into n clones. She then built the tree by randomly selecting a root node, then selecting a parent node for each following clone. In finding a usage matrix, she used rejection sampling to ensure that mutations were included in at least two samples. The values for read counts at each sample were distributed with a Poisson distribution, where λ=the read coverage. A Binomial distribution was used to find the number of variant alleles at each position. Variant Allele frequencies are the fraction of read counts that are mutated. Once the variant counts and reference counts were determined, I discretized these frequency matrices by using a threshold of .01. Popic et al. use a the same threshold in LICHeE [14].

3.1.1 Comparisons

I ran both the GMSR and PPM on 90 instances. Table 1 provides a summary of the results of both GMSR and PPM. For each simulated instance, I compared the T solution with the reference solution using the following metrics: (1) The average error between the inferred ($\tilde{A}$) and T (A) usage matrices ($\frac{1}{m_{ref} m_{sol}} ||\tilde{A} - A||_1$), where m_{ref} is the number of samples in C and m_{sol} is the number of rows in B. (2) The accuracy (fraction) of the clustered pairs that were correctly determined. (3) The accuracy (fraction) of ancestral relationships that were

correctly determined. (4) The accuracy (fraction) of incomparable relationships that were correctly determined. Two clones are incomparable if they are on separate branches in they phylogenetic tree T [10]. Figure 4 provides an example of calculating error in usage and the accuracy in ancestral relationships.

Figure 5 shows violin plots of CITUP, PhyloSub, PPM, and GMSR. To generate these plots, I took the best solution for each instance. To find the best solution for a given instance, I first sorted all solutions in a solution set based on each metric separately. I then took the sum of all 4 ranks and took the solution that had the lowest rank. These 90 solutions are those included in the violin plot distributions in Figure 5.

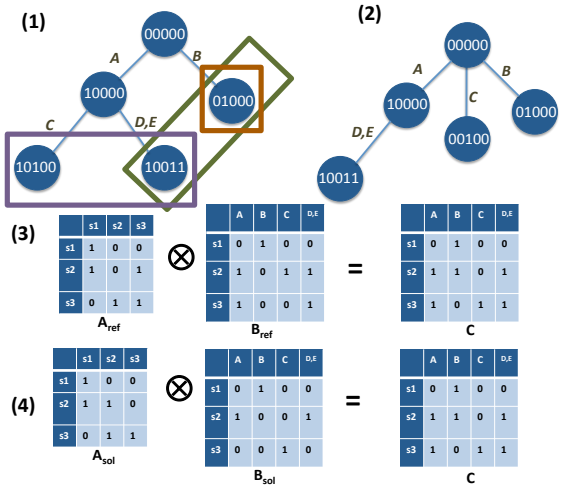
In each metric, PhyloSub and CITUP outperform better than PPM and GMSR, both of which use discretized. Though PhyloSub and CITUP use very different approaches, they both do not use discretized input matrices. In order to better gauge why PPM and GSRM, both of which use discretized input matrices, perform worse, we observe the actual tree topology in the section below.

3.1.2 Visualizing Trees

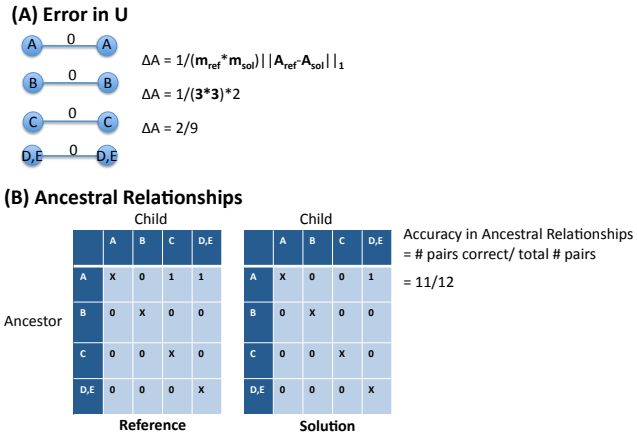
For each simulation, there is only one solution for GMSR. However, for PPM there are multiple solutions for each tumor. Let K_{method} be the #mixings a particular method took. In order to pick an tumor of the tree to analyze further, we selected an tumor such that the K_{ref} was greater than the K_{PPM} and such that the K_{PPM} was greater than K_{GMSR}. There were three patients when this was the case. For the majority of tumors, K_{ref} was greater than K_{PPM} but K_{PPM} was the same as K_{GMSR} (Table 1).

We select by finding the best solution. To do this, we rank each solution based on each of the four metrics. We then use the solution that had the lowest sum of the four ranks. In this tumor, there were six solutions. The median number of solutions for each instance is 4 and generally there are a small number of solutions (Figure 6). For each solution set, there is little variation between solution values (Figure 7). To obtain Figure 7, we took the difference between the largest and smallest values for each of the 90 tumors. We then made the violin plot using to look at the distribution of these 90 values. Because there is little variation, each inferred tree solutions for a given instance would likely present a similar outcome, thus it is likely that each of the solutions for a given tumor will yield a similar tree.

In each instance of the tree, samples from C are connected with dotted lines to leaves of the tree. These leaf nodes are the leaves in B. These doted lines represent mixings. If an internal node is also a row in in B, it is brought down with a second dotted line to be placed in-line with leaf nodes. For example, in Figure 9, there is one parent node, '01001001' that is brought down in-line with the leaves. In this patient (Figures 8, 9, and 10), just by observation, the GMSR and the PPM solution are far more similar to each other than they are to the reference solution. K_{ref} is also much larger than K_{PPM} and K_{GMSR}.

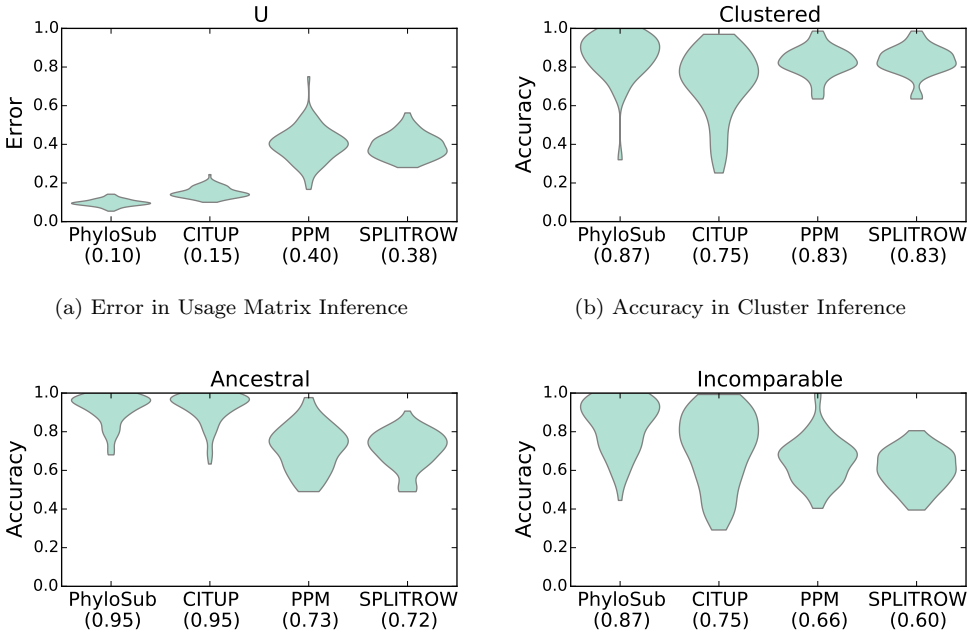


(a) Reference and Inferred Matrices



(b) Calculating ΔU and Ancestral Relationships

Figure 4: **Comparisons Between Reference and Inferred Solution** (a) (1) The original clonal evolution. The colored boxes show how samples are mixed. (2) The clonal evolution in our inferred solution. This is the same as the true solution, except C is mutated earlier. (3) The reference usage and basis matrices. (4) The inferred usage and basis matrices. (b) (5) An example describing how to calculate the error in usage. First, we must find find the difference in all clustered mutations sets. This is simple in this case because the clusters are all the same though when n is larger this is less trivial. Then we must find the minimal one-to-one correspondence and group equate these pairs of mutation clusters together. We then find the error by calculating $\frac{1}{m_{ref}m_{sol}}||\tilde{A} - A||_1$. (6) To find the accuracy in the ancestral relationships, we find the # correct relationships then divide by the total number of pairs. Note: We leave out an example of calculating accuracy in clustering and incomparable relationships because the process is similar to finding the accuracy in ancestral relationships.



(a) Error in Usage Matrix Inference

(b) Accuracy in Cluster Inference

(c) Accuracy in Ancestral Relationship Inference

(d) Accuracy in Incomparable Relationship Inference

Figure 5: **Comparison Between PhyloSub, CITUP, PPM, and GMSR** For each metric, both PhyloSub and CITUP outperform both PPM and GMSR. Though CITUP and PhyloSub use different approaches, they do not use discretized input matrices. PPM and GMSR discretize the input matrices initially, thus information is lost initially.

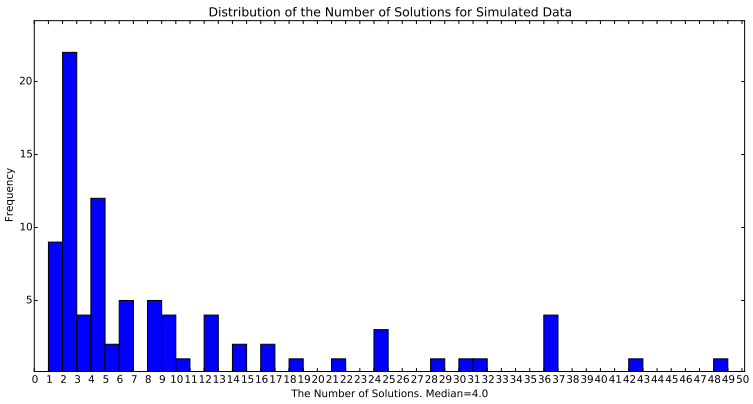


Figure 6: **Distribution of the Number of Solutions for Simulated Data** The median is 4, and the vast majority of simulated instances generate a small number of solutions. Note: There are additional instances at 144, 198, 594 that are not shown.

3.2 Real Data

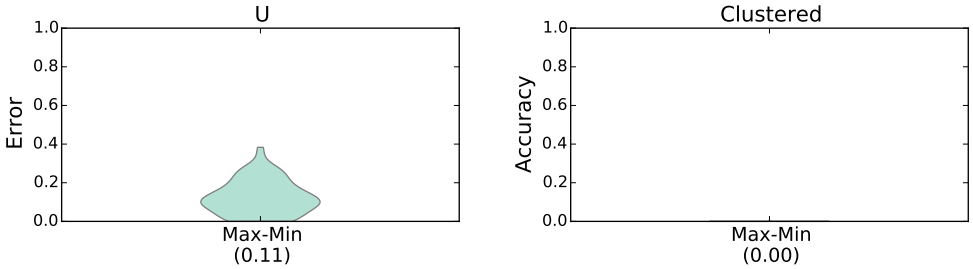
I also ran GMSR and PPM on three other data sets: chronic lymphcytic leukemia (CLL) [13], lung adenocarcinoma [16], and a renal cell carcinoma [3] tumors. While CLL uses data sampled from multiple time points, the other two sets use data in which tumors were sampled from multiple sections. For 14 of the 22 patients, there was both whole-genome or whole-exome sequencing data as well as targeted deep sequencing data available.

Note: PPM did not finish running for patient 4990 because both the deep sequencing and whole-exome sequencing data both took over our time limit,1 day, to run. The number of clones after mutations were clustered, n', was 19 (deep) and 24 (whole), both of which are larger than any other n' of other patients (Table 2).

4 Discussion

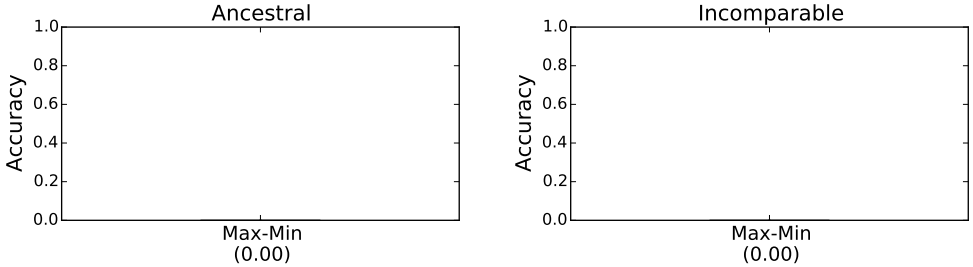
In the generated trees, the tree obtained by both ppm and GMSR look significantly different (Figure 8, 9, 10). One reason there is such a striking distinction between the reference and other two solutions is that the number of mixes in the reference is nearly double that of the other two solutions. Perhaps, minimizing the usage or obtaining a basis matrix with few rows is not the right approach as the reference solution contains samples that contain more portions of the tumor.

When using discretized matrices, there is certain information that cannot be uncovered. For instance, in Figure 12 (C), columns C and D are identical even though they contain different mutations sets. In both methods using discretized



(a) Difference between Max-Min of the Usage Error

(b) Difference between Max-Min of the Accuracy in Cluster Inference



(c) Difference between Max-Min of the Ancestral Relationship Inference

(d) Difference between Max-Min of the Incomparable Relationship Inference

Figure 7: **The Difference Between the Maximum and Minimum Values For Each Set of Solutions.** For each metric, there is relatively little variation between each solution set, especially in b-d, where there is almost no variation.

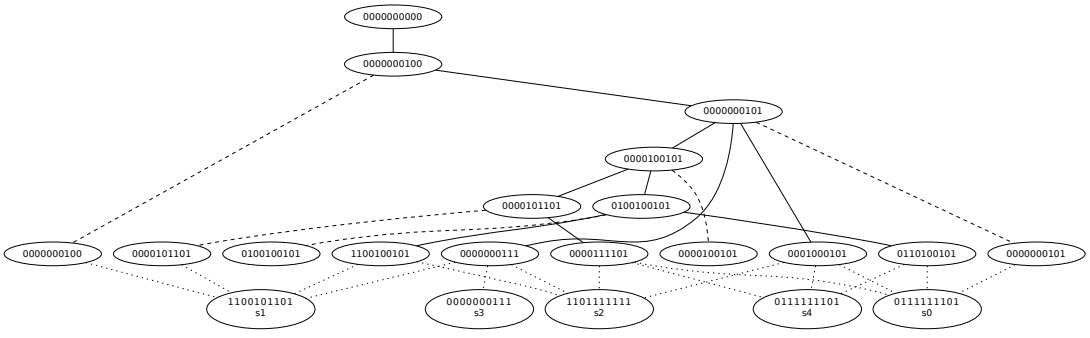


Figure 8: **Reference Solution** The Reference Solution with Coverage=100, Samples=5, Mutations=100, Clones=10, Patient=7. There are 16 mixes.

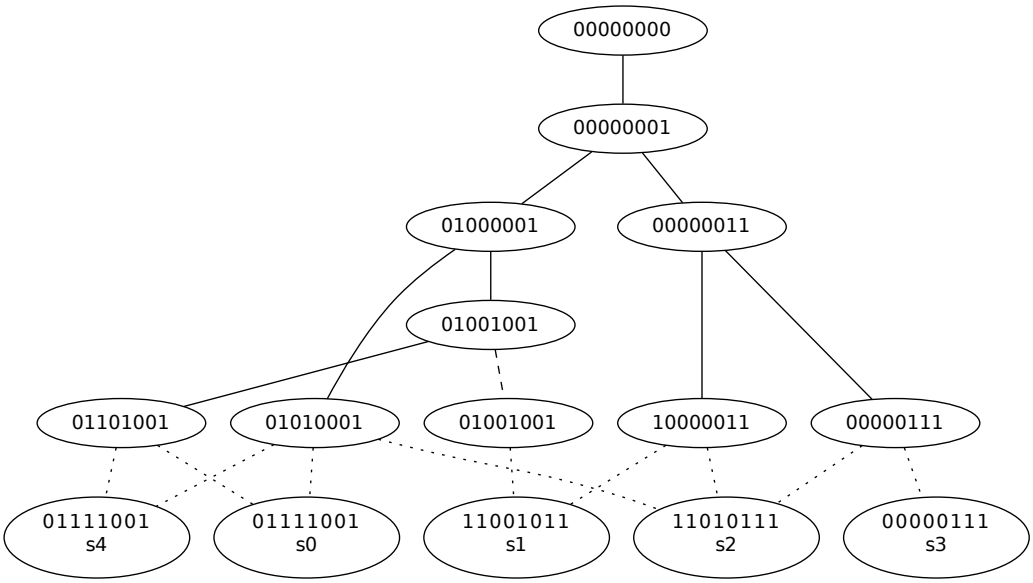


Figure 9: **PPM Solution** The Reference Solution with Coverage=100, Samples=5, Mutations=100, Clones=10, Patient=7. There are 10 mixes.

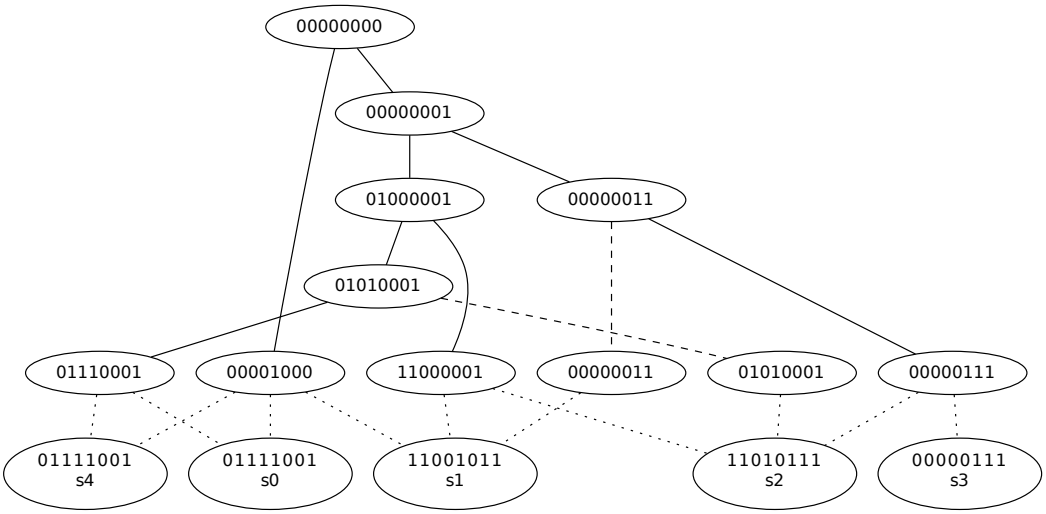
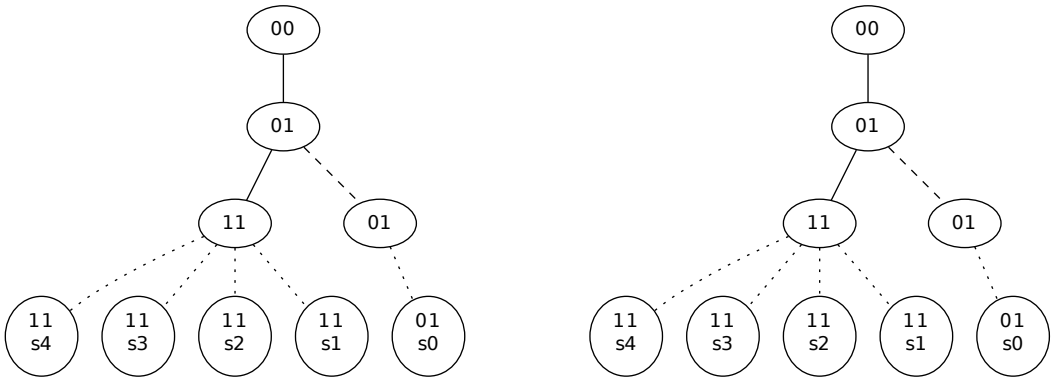


Figure 10: **Split Rows Solution** The Reference Solution with Coverage=100, Samples=5, Mutations=100, Clones=10, Patient=7. There are 11 mixes.



(a) Clonal Evolution of Patient CLL077 (PPM)

(b) Clonal Evolution of Patient CLL077 (GMSR)

Figure 11: **Clonal Trees Inferred for Patient CLL077** There were 16 initial mutations but when clustered there remained 2 clones.

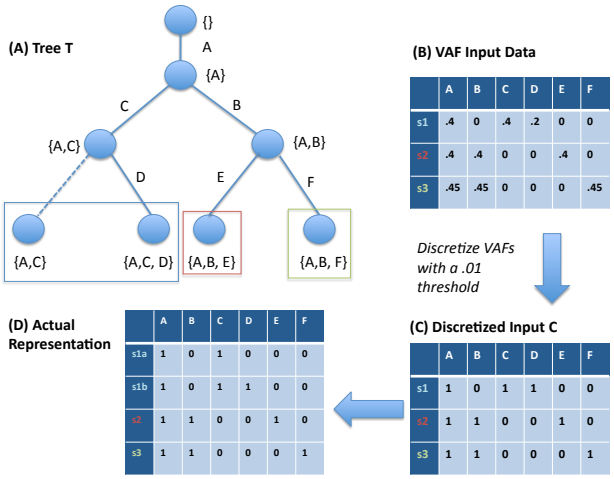


Figure 12: **Information in Discretized Matrices Cannot Always Be Uncovered.** (A) An Example of a Tumor Evolution. (B) The original variant allele frequency data. (C)Though (A,C) and (A,C,D) are on separate branches because the matrix is discretized, columns C and D are identical thus are collapsed onto a single column or mutation cluster. (D) The actual perfect phylogeny matrix representing T. Using PPM, GMSR, or EMSR, the actual representation could never be obtained. However, in looking at the recency data, because mutation C has a higher frequency than D, it would have occurred first.

input matrices, mutations C and D would be clustered into the same mutation set. This occurs when a clone is carried into the next generation as is the case with clone {A,C}. However, in this case, if we use the original frequency data, mutations C and D occur in sample 1 in different frequencies. Because there is a larger frequency of mutation C occurring, it is present in more cells, and thus occurred before mutation D.

4.1 AncesTree

The early work on Perfect Phylogeny Mixtures, the poor performance and counterexamples of GMSR and EMRS, and the loss of information in discretized in put matrices led to the formulation and implementation of of AncesTree. In AncesTree, instead of using a discretized mutation matrix M, we use F, $m \times n$ frequency matrix, where $F = [f_{pi}]$ and f_{pi} is the observed variant allele frequency. We provide a definition for the Variant Allele Frequency Factorization Problem (VAFFP) in which given an observed Variant Allele Frequency (VAF) matrix F, they use a combinatorial approach and an Integer Linear Program to determine a usage matrix U and an n-clonal matrix B that represents a tree T,

where each edge is labeled with exactly one mutation from [n] and no mutation appears more than once.

The crux of the VAFFP is similar to the PPM problem definition (Section 2.1) in that in VAFFP because each sample is a mixture of clones in T with proportions defined in U, the observed frequency matrix can be obtained as: F=$\frac{1}{2}UB$. The coefficient of $\frac{1}{2}$ follows from all mutations being heterozygous because of the infinite sites assumption. This equation is essentially the PPM equation of $A \otimes B = C$, which was a similar factorization problem.

In addition to using a frequency matrix and not a mutation matrix, El Kebir et al. also account for errors that are common in real sequencing data. First, they cluster mutations that likely occurred together. Second, they restrict the ancestry graph, which is a directed acyclic graph (DAG) in which all ancestral relationships between clusters are noted, to contain ancestry edges with high confidence. The VAFFP is unique in that clustering is based on uncertainty in ancestry relationships, not only on similar variant allele frequencies or collapsing identical columns in M. This approach can therefore distinguish variant allele frequencies that may be similar but are actually in different clones. [2]

Figure 13 shows violin plots of comparisons between the inferred and reference solutions of PhyloSub, CITUP, and AncesTree the same simulated instances as described in Section 3.1. AncesTree outperforms CITUP and PhyloSub on all 4 metrics. Though not compared directly with AncesTree because AncesTree only uses data with mutations that occurred with high confidence, AncesTree outperformed both CITUP and PhyloSub, so would outperform GMSR and PPM as well.

In this work, we first formulate the Perfect Phylogeny Mixture Problem (PPM). We then provide implementation details for the GMSR, EMSR, and PPM ILP formulation. After implementing both, we realized that the GMSR and EMSR fail on certain patients and that the GMSR and PPM perform worse that PhyloSub and CITUP. This early work and the formulation of PPM and GMSR as a facotorization problem served as the basis for AncesTree, which uses a similar factorization approach though does not discretize input matrices. AncesTree also accounts for sequencing errors and clusters mutations based on ancestral relationships and can therefore account for mutations that may be in different clones but have similar variant allele frequencies.

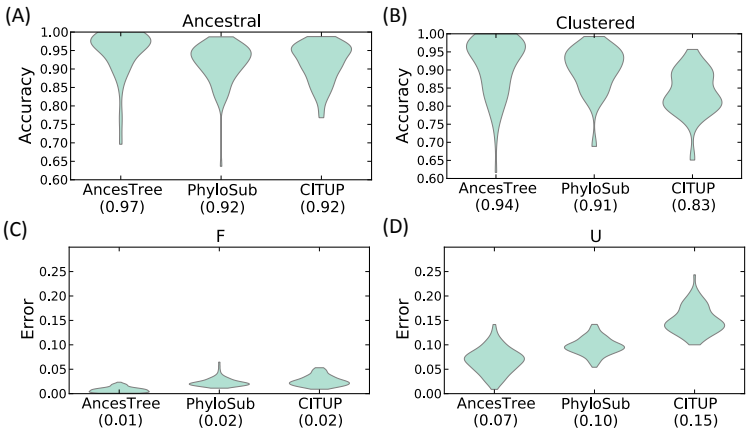


Figure 13: **Comparison of AncesTree, PhyloSub, and CITUP** (A-B,D) The Ancestral, Clustered, and U metrics are described in Section 3.1.1. (C) The Error in the inferred and reference frequency matrix. [2] This is the same violin plot in the published version of AncesTree.

5 Appendix

instance	patient #	m	n	n'	K_{ref}	K_{PPM}	K_{GMSR}	EMSR
Cov_1000_Samples_4	0	4	100	7	13	7	F	R
Cov_1000_Samples_4	1	4	100	8	13	8	8	R
Cov_1000_Samples_4	2	4	100	7	13	10	10	R
Cov_1000_Samples_4	3	4	100	5	12	6	6	R
Cov_1000_Samples_4	4	4	100	7	13	8	8	R
Cov_1000_Samples_4	5	4	100	7	14	6	6	R
Cov_1000_Samples_4	6	4	100	8	13	10	10	R
Cov_1000_Samples_4	7	4	100	7	14	7	7	R
Cov_1000_Samples_4	8	4	100	7	15	8	8	R
Cov_1000_Samples_4	9	4	100	4	13	6	6	R
Cov_1000_Samples_5	0	5	100	8	18	10	10	R
Cov_1000_Samples_5	1	5	100	7	17	12	12	R
Cov_1000_Samples_5	2	5	100	8	13	10	10	R
Cov_1000_Samples_5	3	5	100	8	19	11	11	R
Cov_1000_Samples_5	4	5	100	8	15	9	9	R
Cov_1000_Samples_5	5	5	100	8	17	8	F	R
Cov_1000_Samples_5	6	5	100	10	18	12	12	R
Cov_1000_Samples_5	7	5	100	6	16	7	7	R
Cov_1000_Samples_5	8	5	100	5	15	6	6	R
Cov_1000_Samples_5	9	5	100	8	17	9	9	R
Cov_1000_Samples_6	0	6	100	9	19	14	14	R
Cov_1000_Samples_6	1	6	100	8	18	9	9	R

Cov_1000_Samples_6	2	6	100	8	15	7	7	R
Cov_1000_Samples_6	3	6	100	7	21	13	15	R
Cov_1000_Samples_6	4	6	100	8	19	12	12	R
Cov_1000_Samples_6	5	6	100	7	19	10	10	R
Cov_1000_Samples_6	6	6	100	9	19	14	F	R
Cov_1000_Samples_6	7	6	100	11	19	17	17	F
Cov_1000_Samples_6	8	6	100	8	18	10	10	R
Cov_1000_Samples_6	9	6	100	6	14	9	9	R
Cov_100_Samples_4	0	4	100	7	13	7	7	R
Cov_100_Samples_4	1	4	100	8	13	8	8	R
Cov_100_Samples_4	2	4	100	8	13	8	8	R
Cov_100_Samples_4	3	4	100	9	12	9	F	R
Cov_100_Samples_4	4	4	100	8	13	9	F	R
Cov_100_Samples_4	5	4	100	8	14	8	8	R
Cov_100_Samples_4	6	4	100	10	13	10	10	R
Cov_100_Samples_4	7	4	100	8	14	7	7	R
Cov_100_Samples_4	8	4	100	9	15	10	10	R
Cov_100_Samples_4	9	4	100	4	13	5	5	R
Cov_100_Samples_5	0	5	100	10	18	14	14	F
Cov_100_Samples_5	1	5	100	8	17	12	F	R
Cov_100_Samples_5	2	5	100	10	13	10	10	R
Cov_100_Samples_5	3	5	100	9	19	11	11	R
Cov_100_Samples_5	4	5	100	10	15	11	11	R
Cov_100_Samples_5	5	5	100	11	17	11	11	R
Cov_100_Samples_5	6	5	100	17	18	14	F	F
Cov_100_Samples_5	7	5	100	8	16	10	11	R
Cov_100_Samples_5	8	5	100	8	15	7	7	R
Cov_100_Samples_5	9	5	100	10	17	9	9	R
Cov_100_Samples_6	0	6	100	12	19	14	14	R
Cov_100_Samples_6	1	6	100	8	18	9	9	R
Cov_100_Samples_6	2	6	100	11	15	11	11	R
Cov_100_Samples_6	3	6	100	11	21	17	F	F
Cov_100_Samples_6	4	6	100	8	19	12	12	R
Cov_100_Samples_6	5	6	100	9	19	10	10	R
Cov_100_Samples_6	6	6	100	12	19	16	F	F
Cov_100_Samples_6	7	6	100	14	19	17	17	F
Cov_100_Samples_6	8	6	100	10	18	12	12	F
Cov_100_Samples_6	9	6	100	7	14	9	9	R
Cov_50_Samples_4	0	4	100	8	13	7	7	R
Cov_50_Samples_4	1	4	100	9	13	9	9	R
Cov_50_Samples_4	2	4	100	8	13	8	8	R
Cov_50_Samples_4	3	4	100	11	12	10	F	F
Cov_50_Samples_4	4	4	100	11	13	11	F	F
Cov_50_Samples_4	5	4	100	9	14	8	8	R
Cov_50_Samples_4	6	4	100	11	13	10	10	R

Cov_50_Samples_4	7	4	100	9	14	8	F	R
Cov_50_Samples_4	8	4	100	8	15	8	8	R
Cov_50_Samples_4	9	4	100	6	13	6	6	R
Cov_50_Samples_5	0	5	100	12	18	14	14	F
Cov_50_Samples_5	1	5	100	11	17	13	13	F
Cov_50_Samples_5	2	5	100	10	13	9	9	R
Cov_50_Samples_5	3	5	100	14	19	14	15	F
Cov_50_Samples_5	4	5	100	12	15	14	F	F
Cov_50_Samples_5	5	5	100	12	17	11	F	R
Cov_50_Samples_5	6	5	100	18	18	14	F	F
Cov_50_Samples_5	7	5	100	12	16	13	13	F
Cov_50_Samples_5	8	5	100	7	15	7	7	R
Cov_50_Samples_5	9	5	100	10	17	9	9	R
Cov_50_Samples_6	0	6	100	11	19	14	14	R
Cov_50_Samples_6	1	6	100	10	18	11	12	R
Cov_50_Samples_6	2	6	100	10	15	10	10	R
Cov_50_Samples_6	3	6	100	12	21	20	F	F
Cov_50_Samples_6	4	6	100	11	19	12	12	R
Cov_50_Samples_6	5	6	100	10	19	10	10	R
Cov_50_Samples_6	6	6	100	12	19	16	F	R
Cov_50_Samples_6	7	6	100	14	19	16	F	F
Cov_50_Samples_6	8	6	100	10	18	10	12	R
Cov_50_Samples_6	9	6	100	9	14	9	9	F

Table 1: **Simulated Data Results** This table describes the number of samples, mutations, clustered mutations, and mixes for the reference, PPM, and GMSR solutions. If a solution was not found by GMSR, an F is listed. The EMSR section has two categories, either an R ("runs") if a minimal coloring was found and the algorithm does not fail. If not, an F ("fails") is listed.

patient #	m	n	n'	K_{PPM}	K_{GMSR}	EMSR
EV003	8	40	10	9	9	R
EV005	7	64	9	7	7	R
EV006	9	57	9	9	9	R
EV007	8	56	14	17	17	F
RMH002	5	48	11	9	9	R
RMH004	6	126	15	14	14	F
RMH008	8	71	12	12	12	R
RK26	11	62	15	13	13	R
CLL003_deep	5	20	3	6	6	R
CLL003_whole	5	30	5	8	8	R
CLL006_deep	5	10	1	5	5	R
CLL006_whole	5	16	2	5	5	R
CLL077_deep	5	16	2	5	5	R

CLL077_whole	5	20	6	8	8	R
270_deep	5	396	16	14	F	F
270_whole	5	396	16	13	13	F
283_deep	5	29	5	8	8	R
283_whole	5	29	5	5	5	R
292_deep	3	61	5	4	4	R
292_whole	3	61	5	4	4	R
317_deep	4	3890	10	7	7	R
317_whole	4	3890	12	12	12	F
324_deep	5	438	15	14	15	F
324_whole	5	438	14	13	15	F
330_deep	4	853	12	10	F	R
330_whole	4	853	15	13	F	F
339_deep	4	124	10	8	8	R
339_whole	4	124	10	8	8	R
356_deep	4	112	10	9	9	R
356_whole	4	112	10	9	9	R
472_deep	5	135	14	14	F	F
472_whole	5	135	16	13	F	F
499_deep	4	243	7	5	5	R
499_whole	4	243	9	9	10	R
4990_deep	5	702	19	X	F	F
4990_whole	5	702	24	X	F	F

Table 2: **Real Data Results** The sections first describe the renal cell carcinoma Data, then the chronic lymphcytic leukemia (CLL) data, then the lung adenocarcinoma data. This table describes the number of samples, mutations, clustered mutations, and mixes for the reference, PPM, and GMSR solutions. If a solution was not found by GMSR, an F is listed. The EMSR section has two categories, either an R ("runs") if a minimal coloring was found and the algorithm does not fail. If not, an F ("fails") is listed.

6 Acknowledgements

I would like to thank my Advisor Ben Raphael for his insight, support, and feedback throughout this entire project, Mohammed El Kebir and Layla Oesper for their work with me throughout this project, particularly on the PPM formulation and implementation, Iman Hajirasouliha for his helpful discussions and emails last summer, and David Laidlaw for serving as a second reader.

References

[1] Alderton, Gemma K. "Genomics: One Cell at a Time." Nature Reviews Cancer 11.5 (2011): 312. Web.

[2] El Kebir, M., L. Oesper, H. Acheson-Field, B. Raphael. Reconstruction of clonal trees and tumor composition from multi-sample sequencing data, Conference Preceedings to ISMB, 2015.

[3] Gerlinger, M., Horswell, S., Larkin, J., Rowan, A. J., Salm, M. P., Varela, I., Fisher, R., McGranahan, N., Matthews, N., Santos, C. R., Martinez, P., Phillimore, B., Begum, S., Rabinowitz, A., Spencer-Dene, B., Gulati, S., Bates, P. A., Stamp, G., Pickering, L., Gore, M., Nicol, D. L., Hazell, S., Futreal, P. A., Stewart, A., and Swanton, C. (2014). Genomic architecture and evolution of clear cell renal cell carcinomas defined by multiregion sequencing. Nat Genet, 46(3), 225-233.

[4] Gusfield, Dan. "Trees First." ReCombinatorics: The Algorithmics of Ancestral Recombination Graphs and Explicit Phylogenetic Networks. MIT, 2014. Print. 29-60.

[5] Hajirasouliha, Iman. "A Combinatorial Approach for Analyzing Intra-tumor Heterogeneity from High-throughput Sequencing Data." Bioinformatics 30.12 (2014): 78-86. Web. 20 Feb. 2015.

[6] Hajirasouliha, I., B. Raphael. "Reconstructing mutational history in multiply sampled tumors using perfect phylogeny mixtures." 2014.

[7] Jiao, W., Vembu, S., Deshwar, A. G., Stein, L., and Morris, Q. (2014). Inferring clonal evolution of tumors from single nucleotide somatic mutations. BMC Bioinformatics, 15, 35.

[8] Lam, Fumei, Dan Gusfield, and Srinath Sridhar. "Generalizing the Splits Equivalence Theorem and Four Gamete Condition: Perfect Phylogeny on Three-State Characters." SIAM Journal on Discrete Mathematics 25.3 (2011): 1144. Web.

[9] Karp, Richard M. "Mathematical Challenges from Genomics and Molecular Biology." American Mathematical Society 49.5 (2002): 544-53. Web. 13 Apr. 2015.

[10] Malikic, S. (2014). Clonality Inference in Multiple Tumor Samples using Phylogeny. Master's thesis, Simon Fraser University.

[11] Newburger, D. E., D. Kashef-Haghighi, Z. Weng, R. Salari, R. T. Sweeney, A. L. Brunner, S. X. Zhu, X. Guo, S. Varma, M. L. Troxell, R. B. West, S. Batzoglou, and A. Sidow. "Genome Evolution during Progression to Breast Cancer." Genome Research 23.7 (2013): 1097F08. Web.

[12] Nowell, P. "The Clonal Evolution of Tumor Cell Populations." Science 194.4260 (1976): 23-28. Web.

[13] Schuh, A., Becq, J., Humphray, S., Alexa, A., Burns, A., Clifford, R., Feller, S. M., Grocock, R., Henderson, S., Khrebtukova, I., Kingsbury, Z., Luo, S., McBride, D., Murray, L., Menju, T., Timbs, A., Ross, M., Taylor, J., and Bentley, D. (2012). Monitoring chronic lymphocytic leukemia progression by whole genome sequencing reveals heterogeneous clonal evolution patterns. Blood, 120(20), 4191-4196.

[14] Popic, V., Salari, R., Hajirasouliha, I., Haghighi, D. K., West, R. B., and Batzoglou, S. (2014). Fast and scalable inference of multi-sample cancer lineages. CoRR, abs/1412.8574.

[15] Zare H, Wang J, Hu A, Weber K, Smith J, et al. (2014) Inferring Clonal Composition from Multiple Sections of a Breast Cancer. PLoS Comput Biol 10(7):e1003703. doi: 10.1371/journal.pcbi.1003703

[16] Zhang, J., Fujimoto, J., Zhang, J., Wedge, D. C., Song, X., Zhang, J., Seth, S., Chow, C.-W., Cao, Y., Gumbs, C., Gold, K. A., Kalhor, N., Little, L., Mahadeshwar, H., Moran, C., Protopopov, A., Sun, H., Tang, J., Wu, X., Ye, Y., William, W. N., Lee, J. J., Heymach, J. V., Hong, W. K., Swisher, S., Wistuba, I. I., and Futreal, P. A. (2014). Intratumor heterogeneity in localized lung adenocarcinomas delineated by multiregion sequencing. Science, 346(6206), 256-259.