ALGORITHMS FOR INFECTION AND CANCER GENOMICS

BY

PALASH SASHITTAL

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Computer Science
in the Graduate College of the
University of Illinois Urbana-Champaign, 2021

Urbana, Illinois

Advisor:

Assistant Professor Mohammed El-Kebir

Abstract

Continuous innovations and advances in sequencing technologies have led to the birth and development of several fields of research. In this thesis we propose four methods to address open problems in two such fields, infection genomics and cancer genomics.

The first problem we address is reconstruction of transmission history of an outbreak using genomic and epidemiological data collected from infected hosts. It is challenging to account for all the relevant biological processes that occur during evolution and transmission of the pathogens in the outbreak while also addressing the uncertainty in the most likely solution. Our method, TiTUS, overcomes these challenges by first uniformly sampling from the set of all possible feasible transmission histories of the outbreak under a realistic model of evolution and transmission. Then, a consensus-based solution is generated that summarizes the candidate solutions in a biologically meaningful way. We show that TiTUS efficiently samples the solution space enabling accurate reconstruction of transmission history of an outbreak.

The second method we introduce, Jumper, reconstructs viral transcripts using RNA-sequencing data from infected cells. In this study, we focus our attention on viruses in the *Coronaviridae* family, such as SARS-CoV-2, that express genes by a process of discontinuous transcription mediated by the viral RNA-dependent RNA polymerase. The viral transcriptome provides valuable information with clinical implications such as differential expression of viral genes, the host cell response to viral infection and the viral life cycle. We show that Jumper accurately infers the viral transcripts, outperforming existing transcript assembly methods, and facilitates the study of coronavirus transcriptomes under varying conditions.

The third problem we address is doublet detection in single-cell DNA-sequencing data. Our method, doubletD, is the first stand-alone doublet detection method for single-cell DNA-sequencing data. We use a simple probabilistic model allowing a closed-form maximum likelihood solution that efficiently and accurately detects doublets by identifying characteristic signal in the variant allele frequency (VAF) distribution in the data. On simulations and multiple real datasets, we show that doublet identification and removal using doubletD improves downstream analysis such as genotype calling and phylogeny reconstruction.

Finally, we present a new method, PACTION, which proposes a solution to the tumor phylogeny inference problem in cancer. Due to technological and methodological limitations, existing methods are restricted to identifying tumor clones and phylogenies only based on either small-scale mutations, such as single nucleotide variations (SNVs), or large-scale muta-

tions, such as copy number aberrations (CNAs), preventing a comprehensive characterization of a tumor's clonal composition. To overcome these challenges, we formulate the identification of clones in terms of both SNVs and CNAs as a reconciliation problem. We show that PACTION reliably identifies tumor clones and their evolutionary relationships even in the presence of noise or error in input SNVs and CNAs.

"To my parents and Maitreyee, for their love and support."

Acknowledgments

I would like to begin by thanking my advisor, Professor Mohammed El-Kebir, who guided and supported me through the ups and downs of research. I really appreciate him taking a chance with me, since I had no background in computer science. His positive attitude and encouragement has helped me develop as a researcher and I feel very fortunate that I got the opportunity to work with him.

I would also like to thank Prof. Tandy Warnow, who introduced me to the field of computational biology through her course on Algorithmic Genomic Biology. I really appreciate her support and guidance during my transition into the field of computational biology. I am also grateful to all the other wonderful professors I have had the fortune of learning from and interacting with during my time at University of Illinois, Urbana-Champaign (UIUC).

Next, I acknowledge the support of my fellow research group member – Chuanyi Zhang, Leah Weber, Juho Kim, Yuanyuan Qi and Sarah Christensen. Particularly, I would like thank Chuanyi Zhang for always being available when I needed someone to verify my work and talk about future directions, and Leah Weber for fruitful collaborations and discussions. I have learned a lot from everyone in this group and I feel luck to have been part of it.

Lastly, I would like to thank my family for everything they have done for me. I am grateful to my parents for all the sacrifices they have made to raise me. I would also like to thank Maitreyee for extending support during stressful times and encouraging me to follow my interests.

Table of Contents

Chapter 1: Introduction

Continuous innovations and advances in sequencing technologies have led to the birth and development of several fields of research. In this thesis, we introduce four novel algorithms to solve problems in two such fields, infection genomics and cancer genomics. In the following, we provide an overview of these fields and the problems from these fields that we tackle in this thesis.

1.1 INFECTION GENOMICS

Infection genomics is the study of evolution, infection and resistance to therapy in pathogens (viruses, bacteria and parasites). Genomic sequencing of pathogens in infected hosts provides valuable information to understand transmission and virulence of the disease, host response and pathogen life cycle. We focus on two problems from this field.

The first problem we work on is reconstruction of the transmission history of an outbreak. The transmission history of an outbreak is a crucial tool that improves our understanding of the disease and facilitates public health policy decisions. However, the inference of disease transmission histories remains challenging due to various factors such as genetic diversity of pathogen in infected hosts, known as *within-host diversity* and infection of the host by multiple variants of the pathogen, known as *multi-strain infections*. Moreover, often there are multiple transmission histories that can explain the genetic and epidemiological data equally well. Most current methods for transmission history inference generate just one of the possible solutions leading to biases in downstream analyses. In Chapter 2, we address these challenges by introducing a new method, TiTUS (Transmission Tree Uniform Sampler), which uniformly samples the space of feasible transmission histories under a realistic model that accounts for both within-host diversity and multi-strain infections. We prove the hardness of the decision and counting versions of the transmission tree inference problem. We demonstrate the performance of TiTUS on simulated data and on real data of an HIV outbreak with a known transmission chain [1]. Lastly, we develop a polynomial-time method to summarize the solution space of transmission trees that are consistent with the genetic and epidemiological data. The proposed consensus-based method provides a single transmission tree that summarizes a set of candidate solutions while accounting for the number of distinct strains transmitted in each infection event.

The second problem we tackle is reconstruction of viral transcriptome, also known as *viral transcript assembly*, using RNA-sequencing data of infected cells. The viral transcriptome

has direct influence on the expression levels of viral genes in infected cells and provide valuable information about the host response and viral life cycle. However, transcript assembly remains an open problem, with challenges such as ubiquity of paralogs and unevenness of read coverage. In Chapter 3, we present the first method to reconstruct viral transcripts generated by discontinuous transcription using RNA-seq data of infected cells. Specifically, we focused on viruses in the *Coronaviridae* family, such as SARS-CoV-2, that express genes by a process of discontinuous transcription mediated by the viral RNA-dependent RNA polymerase. Underpinning our approach, JUMPER [2], is the concept of a segment graph, a directed acyclic graph that, distinct from the splice graph used to characterize alternative splicing, has a unique Hamiltonian path. We provide a compact characterization of solutions as subsets of non-overlapping edges in this graph, enabling the formulation of an efficient mixed integer linear program. Applying JUMPER on samples of cells infected by SARS-CoV-1 and SARS-CoV-2, we discovered non-canonical transcripts that are either well supported by long-read data of the same sample or corroborated by multiple independent publicly available SRA samples infected by the same virus. We also found conserved core sequences that possibly explain the generation of some of the inferred non-canonical transcripts. Finally, we demonstrate the use of JUMPER to study viral drug response at the transcript level by analyzing samples with and without treatment prior to infection [3]. In summary, JUMPER enables detailed analyses of coronavirus transcriptomes under varying conditions.

1.2 CANCER GENOMICS

Cancer results from an evolutionary process where somatic mutations accumulate in the genomes of different cells. Computational cancer genomics combines genome sequencing with algorithms to uncover the complexities of cancer. Cancer progression involves proliferation of cells that accumulate new somatic mutations resulting in heterogeneous tumors, composed of different *clones*, each corresponding to a distinct subpopulation of cells with the same set of somatic mutations [4]. The resulting intra-tumor heterogeneity has been clearly linked to critically important cancer phenotypes, including cancer prognosis and the potential of developing resistance to cancer therapy [5, 6]. Therefore, important downstream applications rely on accurate reconstructions of a tumor's clonal architecture, which in turn requires the identification of the different clones, their proportions and their evolutionary history.

One of most recent advances in sequencing technologies to study cancer is single-cell sequencing, in which individual tumor cells are isolated and sequenced independently. This technology holds the potential to facilitate precise reconstruction of a tumor's evolutionary

history. However, the process of isolation of individual tumor cells is error-prone and often result in *doublets* where two or more cells are mistaken for a single cell. Not only do doublets confound downstream analyses, but the increase in doublet rate is also a major bottleneck preventing higher throughput with current single-cell technologies. In Chapter 4, we developed the first stand-alone method for detecting doublets in scDNA-seq data. Our method, DOUBLETD [7], uses a simple probabilistic approach with a closed-form solution and outperforms current methods for downstream analysis of scDNA-seq data that jointly infer the doublets. Underlying our method is the observation that doublets in scDNAseq data have a characteristic variant allele frequency (VAF) distribution. Our novel approach additionally uses allelic dropouts, which are a common source of error in single-cell sequencing methods, as a key signal in identifying doublets. In our work [7], we demonstrated that doublet detection and removal using DOUBLETD improves downstream analyses, such as genotype calling and phylogeny reconstruction, while reducing computational costs. DOUBLETD can be utilized in conjunction with any downstream analysis of choice for scDNA-seq data and therefore obviates the need for downstream methods to individually account for the presence of doublets within their own models.

Lastly, we focus on reconstruction of comprehensive tumor phylogenies in cancer. Tumor phylogenies provide evolutionary relationship between these subpopulations of cells in a cancer tumor and have several clinical applications, such as identifying targets for cancer treatment and understanding the development of metastasis. While cancer cells contain somatic mutations that alter genomes at varying length scales, current tumor phylogeny reconstruction methods only focus on either the small-scale mutations, such as single nucleotide mutations (SNVs), or the large-scale mutations, such as copy number aberrations (CNAs), but not both. In Chapter 5, we investigate whether tumor clonal compositions can be comprehensively reconstructed by reconciliation of the SNV and CNA clone proportions and phylogenies that can be independently and reliably inferred by existing methods for the same cancer tumor. We prove that the proposed reconciliation problem is NP-hard and we introduce PACTION (PArsimonious Clone Tree reconciliatION), an algorithm that solves these problems using two mixed integer linear programming formulations. Using simulations, we find that our approach reliably handles errors in input SNV and CNA proportions and scales to practical instance sizes. On 49 samples from prostate cancer patients [8], we find that our approach more comprehensively reconstructs tumor clonal architectures compared to the manual approach adopted in the previous analysis of the same data.

1.3 PUBLISHED WORK

The work presented in this thesis has resulted in the following peer-reviewed publications. Several chapters of the thesis are reproduced with permission from the following papers. Author order generally follows the convention in biology, where first author carried out majority of the work (joint first authorship is indicated by '$*$').

Palash Sashittal, Chuanyi Zhang, Jian Peng, and Mohammed El-Kebir. Jumper enables discontinuous transcript assembly in Coronaviruses, bioRxiv 2021. (under review)

Palash Sashittal, Simone Zaccaria, and Mohammed El-Kebir. Parsimonious clone tree reconciliation in cancer. *Algorithms for Molecular Biology*, 2021. (in print)

Leah Weber*, Palash Sashittal*, and Mohammed El-Kebir doubletD: detecting doublets in single-cell DNA sequencing data. *Bioinformatics*, 2021. (Special issue for Intelligent Systems for Molecular Biology (ISMB) 2021)

Palash Sashittal, and Mohammed El-Kebir. Sampling and summarizing transmission trees with multi-strain infections. *Bioinformatics*, 2020. (Special issue for Intelligent Systems for Molecular Biology (ISMB) 2020)

Palash Sashittal, and Mohammed El-Kebir SharpTNI: counting and sampling parsimonious transmission networks under a weak bottleneck *RECOMB Comparative Genomics* (RECOMB-CG), Montpellier, France, October 1-4, 2019

Chapter 2: Transmission History Inference

2.1 INTRODUCTION

With the advent of cheaper and more powerful sequencing methods, molecular epidemiology has become an indispensable tool for inference of transmission histories of infectious disease outbreaks. Genomic data of pathogen isolates collected from infected hosts is used to assist with the identification of unknown infection sources and transmission chains. Intensive field work generates crucial epidemiological data that provides addition information such as contact history between patients and exposure times of the patients to sources of infection. Methods that can efficiently use genomic and epidemiological data together for accurate inference of transmission history of outbreaks are the key to real-time outbreak management and devising public health policies and disease control strategies for future outbreaks [9].

There are several challenges that hinder the accurate inference of the transmission history of an outbreak. Phylogeny estimation of the pathogen isolates reveals the evolutionary history of the pathogen during the outbreak. However, due to within-host diversity of many pathogens, branching events in the phylogeny do not correspond to the transmission events during the outbreak [10]. Phylogeny-based methods that assume that the transmission events coincide with the branching events in the phylogeny are therefore only applicable in the context of pathogens with low mutation rates, short incubation times and acute infections [11, 12, 13, 14]. Notably, recent studies of SARS-CoV-2, the virus leading to COVID-19, demonstrate that there are patients that exhibit within-host diversity, *i.e.* the presence of multiple SARS-CoV-2 viral strains in COVID-19 patients [15, 16].

Another factor that makes outbreak transmission history inference challenging is a *weak transmission bottleneck*, where multiple strains of the pathogen are transmitted from a donor to a recipient through a non-negligibly small inoculum. Due to this, the most recent common ancestor of lineages from the same host need not have arisen in that host. A similar phenomenon of co-migration of cancerous cells has been observed in metastatic cancers [17]. Although large inocula have been observed in a number of diseases [18], most of the existing methods for transmission tree inference that account for the within-host diversity do not account for the co-transmission of pathogen strains [19, 20, 21, 22]. That is, these methods assume a *strong transmission bottleneck* where a single strain of the pathogen is transmitted in an infection. A weak transmission bottleneck is considered in SCOTTI [23] and BadTrIP [24], however they make the simplifying assumption that all the transmissions are independent of each other. Our previous work, SharpTNI [25], considers the weak

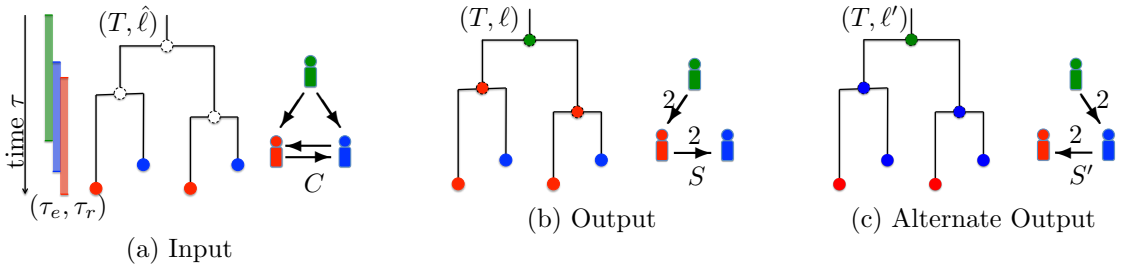


Figure 2.1: **Overview of the Direct Transmission Inference (DTI) problem.** (a) The input of the problem consists of a timed phylogeny T that captures the evolutionary history of the pathogen during the course of the outbreak. Each leaf of T corresponds to a pathogen strain sampled from an infected host and is thus labeled using $\hat{\ell}$ (indicated by colors). Due to within-host diversity, there may exist multiple leaves labeled by the same host. The entry and removal times $[\tau_e(s), \tau_r(s)]$ for each host s is also included in the input. The contact map C is a directed graph between the host set indicating putative transmission pairs. (b) Our aim is to label the internal vertices of T with ℓ such that the resulting transmission edges form a transmission tree S (as shown in Fig. 2.1b). Each edge (s, t) of S is weighted by the number of transmission edges from host s to host t given by the vertex labeling ℓ. (c) An alternative solution to the given DTI instance. It is easy to see that no solution exists under the strong bottleneck constraint whereas under the weak transmission bottleneck there are multiple solutions. All the feasible vertex labelings are shown in Fig. E.1.

transmission bottleneck without this assumption, under a parsimony based framework for a known phylogeny. However, SharpTNI may yield transmission histories that cannot be represented by a tree due to multiple infections of a single host from distinct donors. Such superinfections are unlikely for pathogens where infected hosts acquire immunity towards further infections of the pathogen [26, 27].

Here, we extend our previous work on transmission network inference [25] in the following three ways. First, we consider the problem of counting and sampling uniformly from the set of possible transmission trees for a known phylogeny and epidemiological data. As mentioned, the constraint of tree-like transmissions between hosts is not enforced by SharpTNI [25]. This constraint is enforced by [28] where the order of infections during the outbreak is completely known, and by [29] under the strong transmission bottleneck constraint. In this work, we introduce TiTUS to approximately count and almost uniformly sample the transmission trees under a weak transmission bottleneck for a given timed phylogeny (Fig. 2.1). We prove the hardness of the decision and counting versions of this problem and demonstrate the efficiency and accuracy of TiTUS on simulated data. Second, we present a robust criteria for ranking or prioritizing the uniformly sampled candidate transmission trees. In addition to the simulated data, we demonstrate the performance of the selection criteria on an HIV outbreak with a known transmission chain [1]. Third, in practice, the underlying phylogeny

has some uncertainty and there can be multiple candidates for the transmission tree for a given phylogeny. It is therefore desirable to have an efficient method to summarize the solution space of transmission trees that are consistent with the genetic and epidemiological data. To this end, we propose a consensus-based method that summarizes a set of candidate solutions while accounting for the number of distinct strains transmitted in each infection event.

2.2 PRELIMINARIES

To state the problems we consider in this manuscript, we start by introducing the required concepts and notation. Let T be a rooted tree with vertex set $V(T)$ and edge set $E(T)$. The set of leaves of the tree is given by $L(T)$. The root of the tree is denoted by $r(T)$. We denote the children of a vertex u by $\delta_T(u)$. We write $u \preceq_T v$ if vertex u is ancestral to vertex v, *i.e.* vertex u is present on the unique path from $r(T)$ to vertex v. Note that $\preceq_T$ is reflexive, *i.e.* it holds that $u \preceq_T u$ for all vertices u. We denote the set of m distinct hosts in the outbreak by Σ. In a phylogeographical setting, the set Σ corresponds to m distinct geographical locations.

The evolution of all strains of a pathogen in an outbreak is modeled by a timed phylogeny, which we define as follows.

Definition 2.1. A *timed phylogeny* T is a rooted tree whose vertices are labeled by time-stamps $\tau : V(T) \to \mathbb{R}^{\geq 0}$ such that $\tau(u) \leq \tau(v)$ for all pairs u, v of vertices where $u \preceq_T v$.

Thus, as we can see in the above definition, time moves forward when traversing down a timed phylogeny T starting from the root $r(T)$. It is important to note that the leaves of a timed phylogeny T may occur at distinct time-stamps, *i.e.* T is not necessarily ultrametric.

Each leaf of a timed phylogeny T corresponds to a strain of pathogen that was collected during the outbreak. As such, we know the host from which each individual strain was isolated. This is captured by a leaf labeling, *i.e.* a labeling of the leaves of T by hosts Σ.

Definition 2.2. A *leaf labeling* of a timed phylogeny T is a function $\hat{\ell} : L(T) \to \Sigma$, assigning a host $\hat{\ell}(u) \in \Sigma$ to each leaf vertex $u \in L(T)$.

While we know the host $\hat{\ell}(u)$ from which each individual leaf u of T was sampled, we do not know the hosts of the internal vertices, which correspond to unsampled, ancestral strains. Here, our goal is to determine the hosts in which these ancestral strains reside.Mathematically, we wish to construct a *vertex labeling* $\ell : V(T) \to \Sigma$, such $\ell(u) = \hat{\ell}(u)$

for all leaves $u \in L(T)$. Given a vertex labeling ℓ, each internal vertex u of T thus corresponds to a strain residing within host $\ell(u)$ at time $\tau(u)$.

In addition to the evolutionary history of all strains in the outbreak, a timed phylogeny T combined with a vertex labeling ℓ gives us information about the transmission history of the outbreak. Transmissions of strains from one host to another correspond to edges (u, v) of T labeled by distinct hosts $\ell(u) \neq \ell(v)$. Formally, we define a *transmission edge* as follows.

Definition 2.3. Given a timed phylogeny T and vertex labeling ℓ, an edge (u, v) of T is a *transmission edge* if $\ell(u) \neq \ell(v)$.

The vertex labeling that we construct for a given timed phylogeny T and leaf labeling $\hat{\ell}$, must follow certain constraints for a realistic reconstruction of the transmission history of the pathogen. We will now define these epidemiological constraints.

The first constraint that we introduce is called the *direct transmission constraint*, which imposes the following two restrictions. First, the outbreak begins with a single infected host. We call this initial host the *root host* and it labels the root node $r(T)$ of the timed phylogeny. The *root host* is not infected by any other host and therefore if s is the root host, there cannot exist a transmission edge (u, v) such that $\ell(u) \neq s$ and $\ell(v) = s$. Second, the remaining hosts have a unique infector and are thus infected only once in the course of the outbreak. A possible explanation for this phenomenon is diseases where infected hosts acquire immunity towards further infections of the pathogen [26, 27]. Consequently, there cannot exist two distinct transmission edges (u, v) and (u', v') such that $\ell(v) = \ell(v')$ and $\ell(u) \neq \ell(u')$. However, an infection between any two hosts $s, t \in \Sigma$ may involve the transmission of multiple strains at the same time. This is known as a *weak transmission bottleneck*. Since the transmission of strains must occur concurrently, the time intervals corresponding to any two transmission edges between the same pair (s, t) of hosts must have an non-empty intersection. More formally, we state the *direct transmission* constraint as follows,

Definition 2.4. A vertex labeling ℓ of a timed phylogeny T satisfies the *direct transmission constraint* if (i) there does not exist a transmission edge (u, v) such that $\ell(v) = \ell(r(T))$, (ii) for any two distinct transmission edges (u, v) and (u', v') with $\ell(v) = \ell(v')$, we have $\ell(u) = \ell(u')$ and (iii) we have $[\tau(u), \tau(v)] \cap [\tau(u'), \tau(v')] \neq \emptyset$ for any two distinct transmission edges (u, v) and (u', v') where $\ell(u) = \ell(u')$ and $\ell(v) = \ell(v')$.

Under the *direct transmission* constraint, the set of transmission edges induced by the vertex labeling ℓ uniquely determines the *transmission tree* S. More formally, the vertex set $V(S)$ of a transmission tree S is the host set Σ, and there is a directed edge from $s \in \Sigma$

to $t \in \Sigma$ if and only if there exists at least one edge $(u, v) \in E(T)$ such that (i) $s \neq t$, (ii) $\ell(u) = s$ and (iii) $\ell(v) = t$. Since every host except the *root host* has a unique infector, the directed edges necessarily form a tree. Each directed edge $(s, t) \in E(S)$ is given a weight $w : E(S) \to \mathbb{N}$ such that $w(s, t)$ equals the number of transmission edges in T from host s to t. If $w(s, t) = 1$ for all edges $(s, t) \in E(S)$ then each host is infected due to the transmission of a single pathogen strain. This phenomenon is known as a *strong transmission bottleneck*.

Epidemiological data provide two additional types of information. First, for each host s we are given an interval $[\tau_e(s), \tau_r(s)]$ during which the host was present in the outbreak and susceptible for infection. Specifically, $\tau_e(s) \in \mathbb{R}^{\geq 0}$ is the entry time at which host s became susceptible for infection, whereas $\tau_r(s) \in \mathbb{R}^{\geq 0}$ is the *removal time* at which the host was removed from the susceptible and infected populations and placed in treatment or recovering.

Second, there can also be documented geographical constraints that prevent transmissions between any given pair of hosts. We account for all such constraints using a *contact map*. A *contact map* C is a directed graph whose vertex set equals the set Σ of hosts. A directed edge (s, t) represents a possible infection event from host s to host t. If any two hosts are not connected in C then there can be no infection event between that pair of hosts. It can clearly be seen that (i) the contact map C is a subgraph of the interval graph induced by the intervals $[\tau_e(s), \tau_r(s)], \forall s \in \Sigma$ and (ii) the transmission tree S is a spanning arborescence of the contact map C. Thus, even in the absence of documented contacts between hosts, a contact map is induced by the entry and removal times of the hosts.

2.3 PROBLEM STATEMENT

We focus on inferring the transmission history of an outbreak for a known pathogen phylogeny T. In addition, we are given epidemiological data, which include the contact map C, entry and removal times $[\tau_e(s), \tau_r(s)]$ for each host $s \in \Sigma$ and assume a direct transmission constraint under a weak transmission bottleneck. This leads to the following decision problem.

Problem 2.1 (Direct Transmission Inference (DTI))**.** Given a timed phylogeny T with time-stamps $\tau : V(T) \to \mathbb{R}^{\geq 0}$, a leaf labeling $\hat{\ell} : L(T) \to \Sigma$, a contact map C and entry $\tau_e : \Sigma \to \mathbb{R}^{\geq 0}$ and removal times $\tau_r : \Sigma \to \mathbb{R}^{\geq 0}$, find a vertex labeling ℓ that induces a transmission tree S that is a spanning arborescence of C and $\tau(u) \in [\tau_e(s), \tau_r(s)]$ for all hosts s and vertices u where $\ell(u) = s$.

An instance of the DTI problem is shown in Fig. 2.1a shows an instance of the DTI

problem along a with a solution vertex labeling ℓ and induced transmission tree S, where the three hosts are inducated using three colors. Importantly, a DTI problem instance may admit multiple solutions, as shown in Fig. 2.1b and Fig. 2.1c. These solutions provide alternative reconstructions of the transmission history, and thus must be taken into consideration in any downstream analysis of the outbreak to devise policy to better manage/prevent future outbreaks. To quantify the number of alternative reconstructions, we formulate the following counting problem.

Problem 2.2 (# Direct Transmission Inference (#DTI))**.** Given a timed phylogeny T with time-stamps $\tau : V(T) \to \mathbb{R}^{\geq 0}$, a leaf labeling $\hat{\ell} : L(T) \to \Sigma$, a contact map C and entry $\tau_e : \Sigma \to \mathbb{R}^{\geq 0}$ and removal times $\tau_r : \Sigma \to \mathbb{R}^{\geq 0}$, count the number of vertex labelings ℓ that induce a transmission tree S that is a spanning arborescence of C and $\tau(u) \in [\tau_e(s), \tau_r(s)]$ for all hosts s and vertices u where $\ell(u) = s$.

Let $\mathcal{L}$ be the set of all solutions to a given DTI problem instance. Ideally, we would exhaustively enumerate all solutions to the problem instance. However, worst case, the number of solutions scales exponentially with our input. Thus, to obtain a good overview of the solution space $\mathcal{L}$, we need to consider the sampling version of #DTI problem where we wish to uniformly sample the solution space.

In summary, we defined three versions of the DTI problem: a decision, counting and sampling version. In the following, we will consider a previously defined constrained version of the DTI problem as well as a generalization.

2.3.1 Related Transmission Tree Inference Problems

We start by considering a version of the DTI problem with one additional constraint. This additional constraint requires that only one pathogen strain is transmitted to a new host in a transmission event, and is known as a *strong transmission bottleneck*. We refer to this problem as Directed Transmission Inference under Strong Bottleneck (DTI-SB), and denote the space of solutions by $\mathcal{L}SB$. This problem was posed by [21].In subsequent work, [29] introduced a polynomial time algorithm to enumerate and uniformly sample from the set $\mathcal{L}_{\text{SB}}$. Since the DTI-SB only has one additional constraint over the original DTI problem, the solution space of DTI-SB is a proper subset of the solution space of DTI for the same timed phylogeny T, leaf labeling $\hat{\ell}$ and epidemiological data. More formally, we have $\mathcal{L}_{SB} \subseteq \mathcal{L}$.

The second problem we consider is a relaxed version of DTI. Specifically, we relax the *direct transmission* constraint for a given instance of DTI. We refer to this problem as

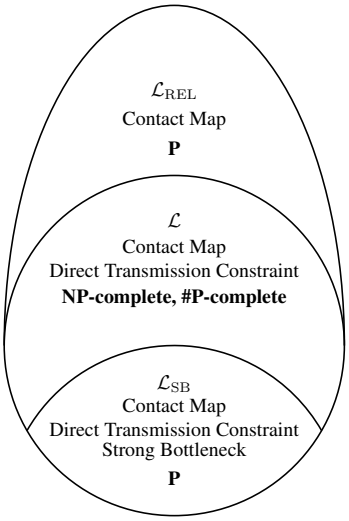


Figure 2.2: **Schematic of the solution spaces of transmission trees under different constraints for a known timed phylogeny.** We have $\mathcal{L}_{\text{SB}} \subseteq \mathcal{L} \subseteq \mathcal{L}_{\text{REL}}$. $\mathcal{L}_{\text{SB}}$ is the solution space of transmission trees with a strong bottleneck that is considered in the work of [29] where they show that counting the solutions and sampling from this solution space can be performed in polynomial time. $\mathcal{L}$ is the solution space of DTI which we show to be both NP-complete and #P-complete. Finally, $\mathcal{L}_{\text{REL}}$ is the relaxed solution space that is used to construct a polynomial time rejection based naive sampling and counting algorithm in Section 2.5.2.1.

rel-DTI and the space of feasible solutions for a given instance by $\mathcal{L}_{\text{REL}}$. Section 2.5.2.1 introduces a polynomial time dynamic programming algorithm that enumerates, counts and uniformly samples from the set $\mathcal{L}_{\text{REL}}$. Since the rel-DTI problem is a relaxation of the DTI problem, we can use the algorithm introduced in Section 2.5.2.1 to uniformly sample from the solution space of the DTI problem ($\mathcal{L}$). Fig. 2.2 shows the relation between the solution spaces of the three transmission tree inference problems.

2.3.2 Consensus Tree Problem

For the DTI problem described in the previous section, we start with a given pathogen phylogeny T. However, in practice the phylogeny needs to be inferred from genomic sequences of the strains collected from individual hosts Σ. Several methods of phylogeny inference generate either multiple candidates for the phylogeny or a posterior on the solution phylogeny space [30, 31]. Moreover, for each given timed phylogeny, we can get multiple solutions to the DTI problem as shown for a representative instance in Fig. 2.1. Therefore, there is a need for an efficient method to summarize the candidate transmission trees that explain the disease outbreak.

A common method to summarize the solution space of transmission trees is to aggregate the information from the candidate transmission trees to generate a single graph where each edge is weighted by the number of candidate trees that support that edge [20, 23, 32]. This graph rarely represents a single coherent transmission tree among the set of all hosts in the dataset. For this reason, the resulting graph is called a *relationship graph* [32] and does not provide crucial information about co-occurrence and mutual exclusivity among edges of the candidate transmission trees.

Another line of method summarizes the set of candidate solutions using one or more consensus trees that best represent the solution space [33, 34]. For instance, [33] apply pairwise distance metrics on the space $\mathcal{S}$ of transmission trees, not taking into account the number $w(s,t)$ of transmitted strains between pairs of host (s,t). The resulting distance matrix is subsequently embedded into lower dimensional space that the authors then cluster. Finally, each cluster is then assigned a single transmission tree in $\mathcal{S}$ as its representative [29]. [34] follow a similar embedding approach, again not taking the number $w(s,t)$ of transmission into account. Thus neither method supports a weak transmission bottleneck. To address this limitation, we define the weighted parent-child distance (WPCD) $d(S_1, S_2)$ between any two transmission trees S_1 and S_2 as follows.

Definition 2.5. Let $S_1 = (\Sigma, E_1)$ with edge labeling w_1 and $S_2 = (\Sigma, E_2)$ with edge labelings w_2 be two transmission tree on the same vertex set Σ. The *weighted parent-child distance* between the two graphs denoted by $d(S_1, S_2)$ is

$$d(S_1, S_2) = \sum_{(s,t)\in E_1} w_1(s,t) + \sum_{(s,t)\in E_2} w_2(s,t) - 2 \sum_{(s,t)\in E_1\cap E_2} \min\{w_1(s,t), w_2(s,t)\}. \tag{2.1}$$

In Appendix C.1 we show that this distance function induces a metric in the space $\mathcal{S}$ of transmission trees. Note that transmission trees S and S' that have the same topology but different edge weights w and w' will have $d(S, S') > 0$. As a result, WPCD can be used to produce a consensus transmission tree while taking an incomplete transmission bottleneck into account. Under the *strong transmission bottleneck* the *weighted parent-child distance* simplifies to the size of the symmetric difference between the edge sets of the two transmission trees, *i.e.* $d(S, S') = |E'\backslash E| + |E\backslash E'|$. This distance is known as the parent-child distance, and has been used to compare tumor phylogenies [35, 36]. Using WPCD, we define the following consensus tree problem.

Problem 2.3 (Single Consensus Transmission Tree (SCTT))**.** Given k distinct transmission

trees $\mathcal{S} = \{S_1, \cdots, S_k\}$ with edge labelings $\{w_1, \cdots, w_k\}$ find a consensus transmission tree R that minimizes $d(\mathcal{S}, R) = \sum_{i=1}^{k} d(S_i, R)$.

2.4 COMPLEXITY

This section establishes hardness results for the decision and counting versions of the DTI problem.

Theorem 2.1. DTI is NP-complete.

We show the hardness of DTI by reduction from the 1-in-3SAT problem, which is a known NP-complete problem [37]. Details are in Appendix B.1.

It is known that the #1-in-3SAT is a #P-complete problem [38]. In order to show that the #DTI is also #P-complete, it suffices to show that there exists a polynomial-time reduction from #1-in-3SAT such that the number of solutions is preserved, which we do in Appendix B.1.

Theorem 2.2. #DTI is #P-complete.

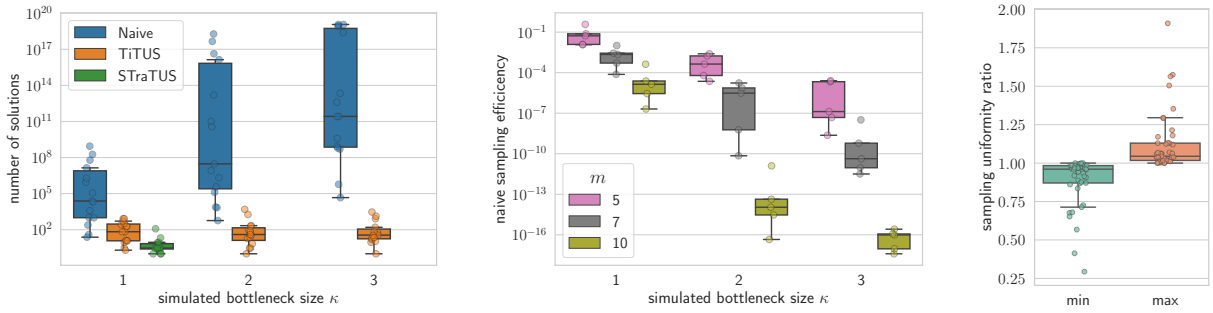
Since the decision problem DTI is NP-complete, there does not exist a fully polynomial randomized approximate scheme (FPRAS) for the counting version of DTI unless NP=RP [39, 40].

2.5 METHODS

This sections describes the methods developed to solve the decision, counting and sampling versions of the DTI problem.

2.5.1 Decision Problem

Since the DTI is NP-complete, we propose to use SATISFIABILITY to solve the decision problem. As such, we construct a Boolean formula ϕ for a given DTI instance $(T, \hat{\ell}, \tau_e, \tau_r, C)$, such that there is a bijection between the solutions of the DTI instance and the corresponding SAT instance ϕ. Solving the SAT instance will then be equivalent to solving the corresponding DTI problem.



(a) (b) (c)

Figure 2.3: **TiTUS accurately samples solutions to the DTI problem.** (a) The number of solution to the rel-DTI ($|\mathcal{L}_{\text{REL}}|$), the DTI ($|\mathcal{L}|$), and the DTI-SB ($|\mathcal{L}_{\text{SB}}|$) problems computed using the Naive rejection sampling, TiTUS, and STraTUS respectively. The number of solutions to the rel-DTI problem grows rapidly for increasing values of the simulated bottleneck size κ, while STraTUS fails to provide any solution when κ is greater than 1. (b) The sampling efficiency, defined as the ratio $|\mathcal{L}|$ and $|\mathcal{L}_{\text{REL}}|$ for increasing values of simulated number of hosts m and bottleneck size κ. (c) The ratio between the minimum and maximum observed sampling frequency using TiTUS with the true uniform sampling frequency .

Vertex labeling: Decision variables $\mathbf{x} \in \{0,1\}^{n \times m}$ encode a vertex labeling, *i.e.* $x_{i,s} = 1$ if and only if the node $\ell(v_i) = s$ and $x_{i,s} = 0$ otherwise. We encode uniqueness of the label of each vertex with the following formula.

$$\text{onehot}(\{x_{i,1}, \cdots, x_{i,m}\}), \quad \forall v_i \in V(T). \tag{2.2}$$

The function onehot(X) encodes that exactly one binary variable $x \in X$ is true, which can be accomplished by the following constraint.

$$\left[\bigvee_{x \in X} x\right] \wedge \left[\bigwedge_{x,y \in X} (\neg x \vee \neg y)\right]. \tag{2.3}$$

Transmission edges: We encode the transmission edges using variables $c_{s,t}$ with $s, t \in \Sigma$ and $s \neq t$. We enforce that $c_{s,t} = 1$ if and only if the host t is infected by host s and $c_{s,t} = 0$, *i.e.*

$$(x_{i,s} \wedge x_{j,t}) \implies c_{s,t}, \quad \forall (v_i, v_j) \in E(T) \text{ and } s, t \in \Sigma. \tag{2.4}$$

Root host: To enforce that the host which labels $r(T)$ is not infected by any other host, we have

$$x_{i,t} \implies \neg c_{s,t}, \quad \forall s, t \in \Sigma, s \neq t, \tag{2.5}$$

where $v_i = r(T)$.

Direct transmission constraint: We enforce that any host cannot be infected by more than one other host. For each host $t \in \Sigma$, we have

$$\neg(c_{s,t} \wedge c_{s',t}), \quad \forall s, s' \in \Sigma \text{ and } s \neq s'. \tag{2.6}$$

We require that all transmission edges from host s to host t must have time intervals that overlap. For all edge pairs $(v_i, v_j), (v_k, v_l)$ that do not have overlapping time intervals, *i.e.* $[\tau(v_i), \tau(v_j)] \cap [\tau(v_k), \tau(v_l)] = \emptyset$, we impose

$$\neg(x_{i,s} \wedge x_{j,t} \wedge x_{k,s} \wedge x_{l,t}), \quad \forall s, t \in \Sigma, s \neq t. \tag{2.7}$$

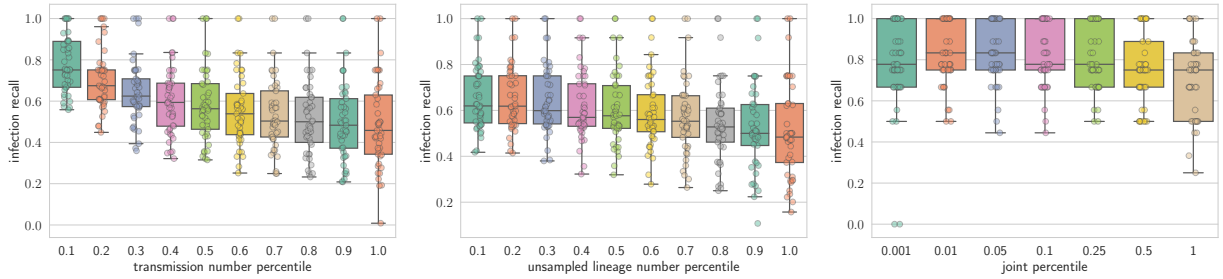
2.5.2 Counting and Sampling Problem

2.5.2.1 Naive Rejection based Method

For a naive rejection sampling algorithm, we relax the *direct transmission constraint* and uniformly sample vertex labelings for the timed phylogeny T such that for all transmission edges (u, v) we have $(\ell(u), \ell(v)) \in E(C)$. As described in Section 2.3.1, we refer to this as the rel-DTI problem. Let the set of such vertex labelings be $\mathcal{L}_{\text{REL}}$. Drawing a vertex labeling labeling $\ell \in \mathcal{L}_{\text{REL}}$ uniformly at random from the set $\mathcal{L}_{\text{REL}}$ can be done in polynomial time, as we describe in Appendix A.1. The sampled vertex labeling labeling ℓ is rejected unless it satisfies the *direct transmission constraint*, which can be verified in polynomial time. The probability of success for this rejection based sampling algorithm is $1 - (|\mathcal{L}|/|\mathcal{L}_{\text{REL}}|)^K$ after K repetitions.

2.5.2.2 Approximate Counting and Sampling using SAT

Using the SAT formulation shown in Section 2.5.1, we use ApproxMC [41, 42] to approximate $|\mathcal{L}|$ and UniGen [43, 44] to sample almost uniformly from $\mathcal{L}$. We call the resulting



(a) (b) (c)

Figure 2.4: The transmission number and number of unsampled lineages of the solutions to the DTI problem are negatively correlated to the infection recall. (a) The infection recall for the uniformly sampled solution within different percentile based on the transmission number. (b) The infection recall for the uniformly sampled solution within different percentile based on the number of unsampled lineages. (c) The infection recall of the consensus transmission trees within different percentiles of both the transmission number and the number of unsampled lineages simultaneously.

method Transmission Tree Uniform Sampler (TiTUS). This method is available, together with our previous method SharpTNI [25], at `https://github.com/elkebir-group/TiTUS`.

2.5.3 Consensus Problem

This section introduces a polynomial time algorithm to solve the SCTT problem. The algorithm and the proof for correctness follow the work of [36]. Let $\mathcal{S} = \{S_1, \cdots, S_k\}$ be a set of k transmission trees with edge weights $\{w_1, \cdots, w_k\}$. Our goal is to find a consensus tree R that minimizes $d(\mathcal{S}, R)$ where $d(\cdot, \cdot)$ is the *weighted parent-child distance*. We start by considering a simpler problem, given a rooted tree R on the set Σ of hosts, find nonnegative weights w^* of the edges of R so as to minimize the WPCD to $\mathcal{S}$. To solve this problem, we augment the given edge weights w_i of trees $S_i \in \mathcal{S}$ to include non-edges, yielding the function $q_i : \Sigma \times \Sigma \to \mathbb{N}$, where

$$q_i(s,t) = \begin{cases} w_i(s,t), & \text{if } (s,t) \in E(S_i), \\ 0, & \text{otherwise.} \end{cases} \tag{2.8}$$

Observe that the parent-child distance between two transmission trees S_i and S_j can be re-written as

$$d(S_i, S_j) = \sum_{(s,t)\in\Sigma\times\Sigma} |q_i(s,t) - q_j(s,t)|. \tag{2.9}$$

To get the optimal weights for the given tree R, for any edge $(s,t) \in E(R)$, we define

$$w^*(s,t) = \arg\min_{z>0} \sum_{S_i\in\mathcal{S}} |q_i(s,t) - z|. \tag{2.10}$$

Intuitively, without the $z > 0$ constraint, the median will minimize this cost. Therefore, $w^*(s,t)$ for every pair of hosts (s,t) is given by $\max\{\text{MED}, 1\}$ where MED is the median of the set $\{q_1(s,t), \cdots, q_k(s,t)\}$. For the case where k is even, we define MED as the smaller of the two middle values. Thus, we have the following proposition.

Lemma 2.1. Given a set $\mathcal{S} = \{S_1, \cdots, S_k\}$ of k transmission trees with edge weights $w_1, \cdots, w_k$ and a transmission tree R, weights $w^*(s,t)$ for $(s,t) \in E(R)$ will minimize the WPCD of $\mathcal{S}$ and R.

To identify a consensus tree R with minimum WPCD, we define the *weighted parent-child graph* P as a complete graph with nodes given by the set Σ and a weight function

$$w_p(s,t) = \sum_{S_i\in\mathcal{S}} (|q_i(s,t) - w^*(s,t)| - |q_i(s,t)|) \tag{2.11}$$

Observe that the weights of the edges of P can be negative.

Theorem 2.3. Given a set $\mathcal{S} = \{S_1, \cdots, S_k\}$ of k transmission trees with edge weights $w_1, \cdots, w_k$, a minimum weight spanning arborescence of the corresponding weighted parent-child graph P defines a tree R that is a solution to the SCTT problem with the distance measure used is weighted parent-child distance.

Proof. Provided in Appendix C.2. QED.

Although edge weights w_p of P can be negative, the requirement of R to be a spanning arborescence of G means that we can solve this problem in polynomial time with standard minimum weight spanning arborescence algorithms.

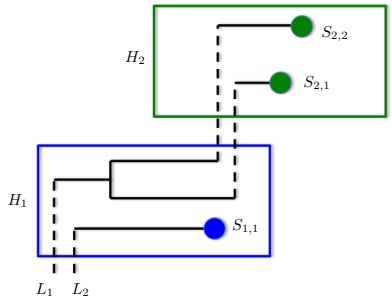


Figure 2.5: **Schematic representation of unsampled lineages in outbreaks**. Different hosts H_1 and H_2 are represented by rectangular boxes and the samples taken from the hosts are indicated by blue or green circles inside the boxes respectively. Black lines represent the evolution of pathogen lineages. Solid lines correspond to within-host evolution of the pathogen whereas dashed lines represent the transmission of strains during infection. Two lineages L_1 and L_2 entering host H_1 are shown. Lineage L_1 is an *unsampled lineage* because even though two strains of L_1 are transmitted to host H_2, none of the samples of H_1 belong to the lineage L_1.

2.6 RESULTS

This section presents the results obtained by applying TiTUS to simulated as well as a real dataset.

2.6.1 Simulations

We employ a two-stage approach to simulate an outbreak, generalizing [20]'s simulation framework that uses a strong transmission bottleneck to support a weak transmission bottleneck. First, we simulate the transmission process between the m hosts using the SIR epidemic model [45]. The epidemiological model takes the transmission bottleneck size κ and minimum number n_s of strains/leaves for each host s as input. Given this input, the model generates a transmission tree S with entry $\tau_e(s)$ and removal times $\tau_r(s)$ for each host s as well as the number of transmissions $w(s,t) = \kappa$ between each pair $(s,t) \in E(S)$ of hosts. Given S and w, we then simulate the evolution of the pathogens within each infected host using a simple coalescence model with constant population size [46]. This process yields a forest of timed phylogenies for each individual host s. We construct a single timed phylogeny of all hosts by stitching together individual timed phylogenies using the transmission tree S. We sample all the pathogen strains present in each infected host. This results in more samples from hosts that have higher within-host diversity. For each combination of number $m \in \{5, 7, 10\}$ of hosts and bottleneck size $\kappa \in \{1, 2, 3\}$ we generate five instances, amounting to a total of 45 simulated instances. The cases with $\kappa = 1$ correspond to outbreaks

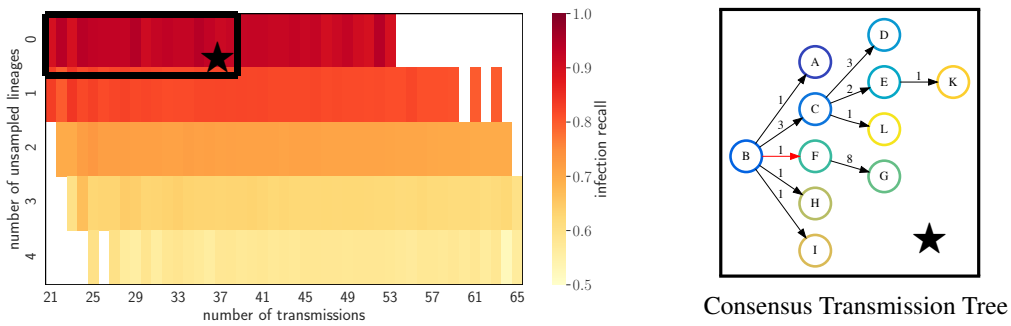


Figure 2.6: **Consensus transmission tree computed for the solutions selected using the proposed criteria infers almost the entire transmission chain for the HIV outbreak.** The figure on the left shows the infection recall of the solutions with different transmission numbers and number of unsampled lineages, uniformly sampled using TiTUS. The black box encompasses the solutions selected for the percentile threshold of $\alpha = 0.01$. The figure on the right shows the consensus transmission tree for the selected solutions. Each edge is labeled by the number of strains transmitted from the donor to the recipient host. The incorrectly inferred transmission B→F is highlighted in red.

with a strong transmission bottleneck. In order to mimic the uncertainty in epidemiological data seen in practice, we increase the length of the entry and removal time interval $[\tau_e(s) - \Delta, \tau_r(s) + \Delta]$ for each host s, where Δ equals 10% of the total outbreak duration.

We find that increasing the number of hosts and bottleneck size in the simulations leads to an increase in the number of vertices n in the phylogenetic trees (Fig. E.2). This leads to a sharp increase in the number of feasible solutions to the rel-DTI (Fig. 2.3a). The number of solutions to DTI, on the other hand, stays relatively constant for increasing bottleneck size. As a consequence of this, the sampling efficiency of the naive rejection sampling method, defined by the ratio $\mathcal{L}/|\mathcal{L}_{\text{REL}}|$, precipitates with increasing number m of hosts and bottleneck size κ proving it unsuitable for any real applications.

For cases with simulated bottleneck size $\kappa > 1$, StraTUS fails to provide any solutions (Fig. 2.3a). This shows that when multi-strain infections occur, transmission history inference with a strong bottleneck assumption will fail to provide the true transmission tree topology. Finally, we assess the sampling accuracy of TiTUS by comparing the sampling frequency with $1/|\mathcal{L}|$ where $|\mathcal{L}|$ is computed with sharpSAT [47]. For each unique solution that is sampled, the expected sampling frequency $1/|\mathcal{L}|$ is the same. Fig. 2.3c shows that the ratio between both the minimum and maximum values of the observed sampling frequencies with their expected values is close to 1.

We evaluate the performance of TiTUS against SharpTNI [25] on simulations with partially sampled outbreaks. That is, we only collect a fixed number of samples per host (equal to the bottleneck size κ), regardless of the within-host diversity. Partial sampling during

an outbreak is common for ongoing and large-scale epidemics, such as the current COVID-19 pandemic. We ran simulations of partially sampled outbreaks, with number of hosts $m \in \{5, 7\}$ and bottleneck size $\kappa \in \{2, 3, 4, 5\}$, where the transmission history is a tree. We generated five instances for each combination of m and κ, resulting in a total of 40 simulated instances. We find that in 26/40 of the instances, SharpTNI fails to produce a transmission tree while TiTUS is able to sample transmission trees in all the cases (Fig. E.3).

In summary, our simulations show that methods that assume a strong transmission bottleneck cannot be applied to outbreaks with a weak bottleneck. Similarly, methods that do not enforce direct transmission, such as SharpTNI, might return transmission histories that include complex transmission pattern such as superinfection. Moreover, the exponentially increasing gap between the size of the solution space of rel-DTI compared to DTI renders the rejection-based sampling impractical. In contrast, TiTUS almost uniformly samples from the complex solution space of DTI.

2.6.1.1 Criteria to Prioritize Candidate Transmission Trees

We propose several criteria for ranking the vertex labelings for a given timed phylogeny uniformly sampled by TiTUS. The first criterion is the *number of transmission edges* in the vertex labeling. Based on the parsimony principle, which has been used in previous works for both phylogeny inference [48] as well as transmission tree inference [25, 32, 49], we expect vertex labelings that have few transmission edges to be closer to the ground truth.

The second criterion is the *number of unsampled lineages*, which is the number of transmission edges (u, v) for which there does not exist a descendant leaf v' (*i.e.* $v \preceq_T v'$) labeled by $\ell(v)$. Unsampled lineages are a consequence of multi-strain infections and we expect to see fewer unsampled lineages when the within-host diversity of the infected hosts is adequately sampled. Fig. 2.5 illustrates this concept.

To assess these criteria, we compare the sampled transmission trees with the ground truth by computing the *infection recall*, defined as the fraction of transmission events between pairs of hosts that are correctly inferred. Fig. 2.4a shows the value of the *infection recall* for candidate solutions in different percentiles based on the number of transmission edges. Clearly, as we look at solutions with larger transmission numbers, the infection recalls decreases. Fig. 2.4b show a similar negative correlation between the infection recall and the number of unsampled lineages. We use both the transmission number and the number of unsampled lineages to prioritize the uniformly sampled candidate solutions. Specifically, for any given percentile threshold α we include all the vertex labelings whose percentile is at most α for both the transmission number and the number of unsampled lineages. (Thus,

setting $\alpha = 1$ will include all sampled vertex labelings.) The selected vertex labelings are then used to compute the consensus transmissions tree. Fig. 2.4c shows the infection recall of the consensus transmission trees for increasing value of the percentile threshold α. We see that a value of α that is either too small or too large results in a decrease in the *infection recall*. Based on the simulated data, we see that $\alpha^* = 0.01$ yields accurate consensus transmission tree solutions. Hence, the two criteria enable accurate prioritization of sampled vertex labelings.

2.6.2 HIV Outbreak with a Known Transmission Chain

We apply our method TiTUS to infer the transmission history of an HIV-1 outbreak involving 11 patients with a known transmission chain [1, 50]. The data consists of 212 samples collected over the span of 18 years from the 11 patients. The direction of transmissions and a relatively narrow time interval for each transmission event were inferred from epidemiological information obtained by patient interviews, clinical data and treatment histories of the patients.

The DTI problem for this HIV dataset is set up as follows. For the timed phylogeny, we use the Maximum Clade Credibility (MCC) tree obtained from the partially sequenced *env* regions presented by [1] in their publication. Table. E.1 shows the sampling times and transmission windows provided in the epidemiological data for each of the hosts. The transmission window of a host is the time interval inside of which the host is expected to have been infected. Transmission windows for host A and host D are incongruent with the given timed phylogeny. By this we mean there is no vertex labeling on the given MCC phylogeny that allows for the known transmissions to host A and host D. We exclude these time windows, while the transmission windows for the remaining hosts are used to constraint the possible vertex labelings of the MCC tree. We restrict the infection for each host to take place in within the transmission window provided in the epidemiological data. Note that while using the time window constraints, we only restrict the time of infection and do not utilize information about the known infectors for each infected host. Finally, for each host the entry time is taken as the beginning of its time window of transmission and the removal time is the latest date of sampling (Table E.1). We find that STraTUS fails to provide a solution on this dataset. Indeed, a weak transmission bottleneck needs to be considered in order to infer the transmission history.

For this DTI instance, using sharpSAT [47] we find that there are exactly 30,901,500 feasible vertex labelings. We generate 100,000 samples from this solution space and compute the *infection recall* when compared to the known transmission chain. Fig. 2.6 shows the

values the *infection recall* for solutions with different number of transmission edges and number of unsampled lineages. The infection recall is close to 1 for the solutions that have no unsampled lineages. The number of transmission edges also has a negative, albeit weaker correlation with the infection recall.

For any given percentile threshold α we include all vertex labelings whose percentile is at most α for both the transmission number and the number of unsampled lineages. Based on the simulations, we focus on percentile threshold $\alpha^* = 0.01$. For this threshold value, Fig. 2.6 shows the consensus transmission tree inferred by TiTUS. The infection recall for this tree is 0.9, *i.e.* we correctly infer 9/10 transmission from the known transmission chain. We incorrectly infer the transmission B→F while the known transmission to F based on epidemiological data is A→F. Fig. E.5 shows similar behavior of the infection recall as a function of α as observed in our simulations. Moreover, this figure shows that our method is robust around $\alpha^* = 0.01$.

2.7 DISCUSSION

In this paper, we formulated the Direct Transmission Inference (DTI) problem of inferring transmission trees for a given timed phylogeny and epidemiological data while supporting a weak transmission bottleneck. Weak transmission bottlenecks are common in the spread of diseases due to pathogens with large inoculum sizes, high mutation rates, long incubation times and chronic infections [18]. Previous studies of counting and sampling transmission trees for a given timed phylogeny assume a strong transmission bottleneck [28, 29], and are not applicable to outbreaks of pathogens with a weak transmission bottleneck, often failing to return any solution.

We proved that the decision version of the DTI problem is NP-complete and the counting version #DTI is #P-complete. Leveraging recent advances made in approximate counting and sampling of solutions to SATISFIABILITY [41, 43, 44, 51], TiTUS, which uses a SATISFIABILITY oracle to almost uniformly sample from the solution space of DTI. In most cases, uniformly sampled candidate solutions from the transmission tree space will deviate considerably from the ground truth. To address this issue, we proposed two criteria that can be used to prioritize the uniformly sampled transmission trees. We demonstrated the performance and robustness of our selection criteria on both simulated data and a real dataset of an HIV outbreak [1].

Further, we also considered the problem of summarizing a given set of candidate transmission tree solutions of a disease outbreak. We defined a new distance metric *weighted parent-child distance* (WPCD) on the space of transmission multi-trees that capture the

transmission of multiple strains between hosts during an outbreak. This distance is an extension of the parent-child distance which is used in previous works to summarize cancer phylogenies [35, 36]. We presented a polynomial time algorithm for finding the consensus transmission tree with minimum total WPCD from the candidate solutions. The performance of the consensus transmission tree of recalling the transmissions that occurred during the outbreak is demonstrated both on simulated and real datasets.

There are several avenues for future research. First, the decision version of the DTI problem can be used to prioritize a posterior distribution of phylogenies, by checking if each phylogeny admits a vertex labeling that induces a transmission tree that is compatible with the given epidemiological data. A similar approach is employed by [52] where they prioritize statistically likely timed phylogenies that admit vertex labelings with fewer transmission edges. By including biological relevant constraints such as a contact map and direct transmission constraints, we expect to obtain high-fidelity phylogenetic and transmission history reconstructions. Second, one limitation of the proposed method is that it assumes that all the infected hosts in the outbreak are sampled. This assumption is only applicable for small outbreaks in regions with perfect surveillance and reporting system in place. An extension of this method to include unsampled hosts would be a useful. Third, akin to [33], we plan to extend the SCTT to simultaneously cluster the set $\mathcal{S}$ of transmission trees and infer a representative consensus transmission tree for each cluster. Fourth, we plan to directly include the identified prioritization criteria as constraints in the DTI problem. Finally, we plan to apply this methodology to study the origins of observed within-host diversity in COVID-19 patients [15, 16].

Chapter 3: Viral Transcript Assembly

3.1 BACKGROUND

Coronaviruses, and more generally viruses in the taxonomic order of *Nidovirales*, are enveloped viruses containing a positive-sense, single-stranded RNA genome that encodes for non-structural proteins near the 5' end as well as structural and accessory proteins near the 3' end [53]. Since the host ribosome processes mRNA starting at the 5' end, translation of the viral genome only generates the non-structural proteins. Expression of the remaining genes is achieved by *discontinuous transcription* performed by the viral RNA-dependent RNA polymerase (RdRp) [54], a protein that is encoded in the non-structural part of the viral genome. Specifically, RdRp can skip over contiguous genomic regions, or *segments*, in the viral RNA template, resulting in a repertoire of *discontinuous transcripts* that correspond to distinct subsequences of segments ordered as in the reference genome (Figure 3.1a). Several recent studies have analyzed SARS-CoV-2 sequencing samples, identifying 'split reads' — *i.e.* single reads that span non-contiguous parts of the viral geneome — that provide evidence for canonical discontinuous transcription events that produce an intact 3' open reading frame (ORF) as well as non-canonical discontinuous transcription events whose role is unclear [55, 56, 57]. However, to the best of our knowledge, no study has attempted to assemble coronavirus transcriptomes, which could provide important clues about the viral life cycle under various conditions such as drug treatment.

Current methods for transcript assembly are mainly designed for eukaryotes and fall under two broad categories: (i) reference-based methods and (ii) *de novo* assembly methods. The main distinction is that the former require the reference genome as input while the latter have no such requirement. As such, *de novo* assembly methods [58, 59, 60, 61, 62] are useful when the reference genome is unavailable or when the diversity of different species in the sample is too large. On the other hand, reference-based methods [63, 64, 65, 66, 67] generally achieve higher accuracy as they use the reference genome as a scaffold on which to align sequencing reads. Specifically, given an alignment $\mathcal{R}$, reference-based methods seek the set $\mathcal{T}$ of transcripts that comprise the transcriptome, enabling the subsequent quantification of their abundances $\mathbf{c}$ using separate tools [68, 69].

While in this work we similarly seek to reconstruct transcripts $\mathcal{T}$ and their abundances $\mathbf{c}$ from an alignment $\mathcal{R}$ of coronavirus sequencing samples, there are critical differences between the processes of transcription in eukaryotes and coronaviruses. In eukaryotes, a gene may express multiple transcripts that differ in their composition due to *alternative splicing*,

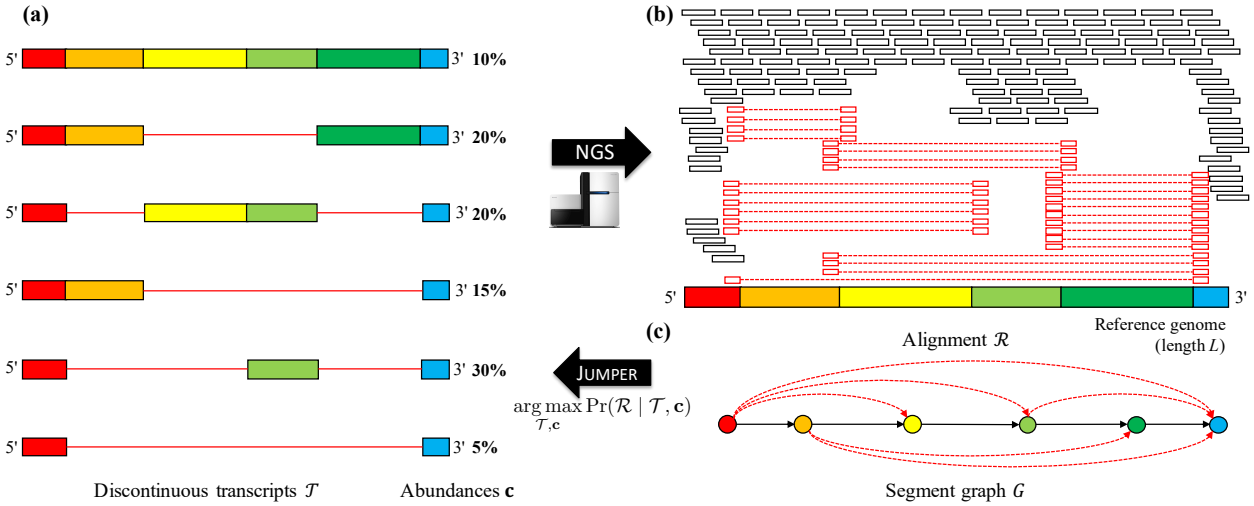


Figure 3.1: (a) Coronaviruses generate a set $\mathcal{T}$ of discontinuous transcripts with varying abundances $\mathbf{c}$ during infection. (b) Next generation sequencing will produce an alignment $\mathcal{R}$ with two types of aligned reads: unphased reads that map to a contiguous genomic region (black) and phased reads that map to distinct genomic regions (red). (c) From $\mathcal{R}$ we obtain the segment graph G, a directed acyclic graph with a unique Hamiltonian path. JUMPER solves the DISCONTINUOUS TRANSCRIPT ASSEMBLY to infer $\mathcal{T}$ and $\mathbf{c}$ with maximum likelihood. While this figure shows single end reads, our problem statement and method make use of the additional information provided by paired-end reads.

which is predominantly mediated by the spliceosome and results in the generation of multiple mRNAs with differentially joined or skipped exons from the same gene. By contrast, transcripts in coronaviruses result from discontinuous transcription, which is mediated by viral RdRp and results in the removal of contiguous segments due to jumps of the RdRp. While conceptually the resulting discontinuous transcripts can be viewed as the result of alternative splicing of a single gene that corresponds to the complete viral genome, there are four key differences and constraints (Figure 3.1a). First, the genomes of coronaviruses are much smaller ($\sim$ 30kb) than eukaryotic genomes. Second, while alternative splicing sometimes involves shuffling of exons, this phenomenon is not observed in discontinuous transcription where the order of segments is fixed. Third, discontinuous transcripts have matching segments at the 5' and 3' ends, which is not necessarily true for eukaryotic transcripts of the same gene. Fourth, the complete viral genome, without any jumps, is always part of the transcriptome. Current transcript assembly methods are not optimized to leverage these four constraints that characterize coronavirus transcriptomes.

In this study, we introduce the DISCONTINUOUS TRANSCRIPT ASSEMBLY (DTA) problem of finding discontinuous transcripts $\mathcal{T}$ and their abundances $\mathbf{c}$ (Figure 3.1a) given an alignment $\mathcal{R}$ of paired end reads (Figure 3.1b). Underpinning our approach is the concept

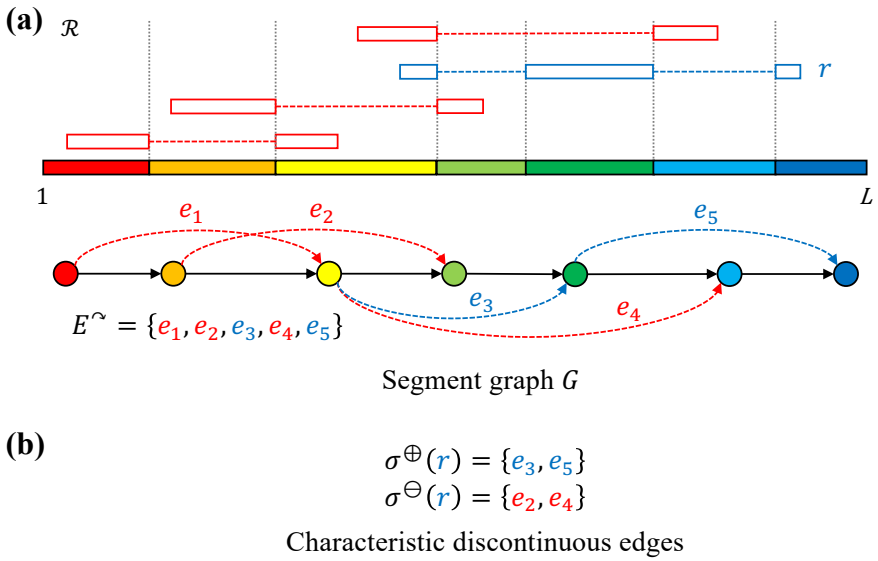


Figure 3.2: (a) Phasing reads in an alignment $\mathcal{R}$ define a set of junctions, which in turn define the segment graph G. (b) Each phasing read has characteristic discontinuous edges indicating the set $\sigma^{\oplus}$ of discontinuous edges present in the read as well as conflicting/overlapping discontinuous edges $\sigma^{\ominus}$. Here, phasing read r (blue), has $\sigma^{\oplus}(r) = \{e_3, e_5\}$ and $\sigma^{\ominus}(r) = \{e_2, e_4\}$. Note that e_1 is not included in $\sigma^{\ominus}(r)$ as it does not overlap with $\pi(r) = \{e_3, e_5\}$.

of a segment graph (Figure 3.1c), a directed acyclic graph that, distinct from the splice graph used to characterize alternative splicing, has a unique Hamiltonian path due to the aforementioned constraints. This enables us to characterize discontinuous transcripts $\mathcal{T}$ as small subsets of non-overlapping edges in this graph. Our method, JUMPER, uses this compact representation to solve the DTA at scale via a progressive heuristic that incorporates a mixed integer linear program. Using simulations, we show that JUMPER drastically outperforms SCALLOP [63] and STRINGTIE [64], existing methods for reference-based transcript assembly in the general case. In real data [55], we run JUMPER on paired-end short-read data of virus infected Vero cells and use long-read data of the same sample for validation. We find that JUMPER not only identifies canonical transcripts that are part of the reference transcriptome, but also predicts expression of non-canonical transcripts that are well supported by long-read data. Similarly, JUMPER identifies canonical and non-canonical transcripts in SARS-CoV-1 and MERS-CoV samples [70]. Finally, we demonstrate the use of JUMPER to study viral drug response at the transcript level by analyzing samples with and without treatment prior to infection [3]. In summary, JUMPER enables detailed analyses of coronavirus transcriptomes under varying conditions.

3.2 DISCONTINUOUS TRANSCRIPTION PROBLEM STATEMENT

To formulate the Discontinuous Transcript Assembly problem, we begin by defining discontinuous transcripts as follows.

Definition 3.1. Given a reference genome, a *discontinuous transcript* T is a sequence $\mathbf{v}_1, \ldots, \mathbf{v}_{|T|}$ of segments where (i) each segment corresponds to a contiguous region in the reference genome, (ii) segment $\mathbf{v}_i$ precedes segment $\mathbf{v}_{i+1}$ in the reference genome for all $i \in \{1, \ldots, |T| - 1\}$, (iii) segment $\mathbf{v}_1$ contains the 5' end of the reference genome and (iv) segment $\mathbf{v}_{|T|}$ contains the 3' end of the reference genome.

In the literature, discontinuous transcripts that differ from the genomic transcript T_0 are called *subgenomic transcripts*, which correspond to subgenomic RNAs (sgRNAs) [55]. Transcripts $\mathcal{T} = \{T_i\}$ occur in *abundances* $\mathbf{c} = [c_i]$ where $c_i \geq 0$ is the relative abundance of transcript T_i such that $\sum_{i=1}^{|\mathcal{T}|} c_i = 1$. While next-generation sequencing technologies provide high coverage of the viral genome of length L of about 10 to 30 Kbp, they are limited to short reads with fixed length ℓ ranging from 100 to 400 bp. For ease of exposition, we describe the formulation in context of single-end reads, but in practice we use the paired-end information if it is available.

As $\ell \ll L$, the identity of the transcript of origin for a given read is ambiguous. Therefore we need to use computational methods to reconstruct the transcripts and their abundances from the sequencing reads. Specifically, given a coronavirus reference genome of length L and reads of a fixed length ℓ, we use a splice-aware aligner such as `STAR` [71] to obtain an alignment $\mathcal{R}$. This alignment provides information about the abundance $\mathbf{c}$ and composition of the underlying transcripts $\mathcal{T}$ in the following two ways. First, the *depth*, or the number of reads along the genome is informative for quantifying the abundance $\mathbf{c}$ of the transcripts. Second, the composition $\mathcal{T}$ of the transcripts is embedded in *phasing reads*, which are reads that align to multiple distinct regions in the reference genome (Figure 3.1b).

To make the relationship between $\mathcal{T}$, $\mathbf{c}$ and $\mathcal{R}$ clear, we introduce the *segment graph* G, which is obtained from the phasing reads in a alignment $\mathcal{R}$. As mentioned, each phasing read $r \in \mathcal{R}$ maps to $q \geq 2$ distinct regions in the reference genome. Each pair of regions that are adjacent in the phasing read are separated by two positions v, w (where $w - v \geq 2$) in the reference genome called *junctions*. Thus, each phasing read contributes $2q-2$ junctions. The collective set of junctions contributed by all phasing reads in $\mathcal{R}$ in combination with positions $\{1, L\}$ induces a partition of the reference genome into closed intervals $[v^-, v^+]$ of junctions that are consecutive in the reference genome (*i.e.* there exists no other junction that occurs in between v^- and v^+). The resulting set of segments equals the node set V of segment

graph G (Figure 3.2a). The edge set E of segment graph G is composed of continuous edges $E^{\rightarrow}$ and discontinuous edges $E^{\curvearrowright}$. Continuous edges $E^{\rightarrow}$ are composed of ordered pairs $(\mathbf{v} = [v^-, v^+], \mathbf{w} = [w^-, w^+])$ of nodes that correspond to segments that are adjacent in the reference genome, *i.e.* where $v^+ = w^-$. On the other hand, discontinuous edges $E^{\curvearrowright}$ are composed of ordered pairs $(\mathbf{v} = [v^-, v^+], \mathbf{w} = [w^-, w^+])$ of nodes that corresponds to segments that are adjacent in at least one phasing read in $\mathcal{R}$ but not adjacent in the reference genome (*i.e.* $w^- - v^+ \geq 2$). Figure 3.1c shows an example of a segment graph.

Definition 3.2. Given an alignment $\mathcal{R}$, the corresponding *segment graph* $G = (V, E^{\rightarrow} \cup E^{\curvearrowright})$ is a directed graph whose node set V equals the set of segments induced by the junctions of phasing reads in $\mathcal{R}$ and whose edge set $E = E^{\rightarrow} \cup E^{\curvearrowright}$ is composed of edges $(\mathbf{v} = [v^-, v^+], \mathbf{w} = [w^-, w^+])$ that are either continuous, *i.e.* $v^+ = w^-$, or discontinuous, *i.e.* $w^- - v^+ \geq 2$ and there exists a phasing read where junctions v^+ and w^- are adjacent.

We note that the segment graph G is closely related to the *splice graph* used in regular transcript assembly where transcripts correspond to varying sequences of exons due to alternative splicing. The key difference, however, is that an alignment $\mathcal{R}$ generated from reads obtained from discontinuous transcripts induces a segment graph G that is a directed acylic graph (DAG) with a unique Hamiltonian path. This is because, as stated in Definition 3.1, discontinuous transcripts $\mathcal{T}$ have matching 5' and 3' ends, and, although their comprising segments may vary, their order follows the reference genome.

Observation 3.1. Segment graph G is a directed acyclic graph with a unique Hamiltonian path.

The unique Hamiltonian path of G corresponds to the sequence of continuous edges $E^{\rightarrow}$. This path corresponds to the whole viral genome which is generated by the RdRp during the replication step [54]. Moreover, by the above observation, G has a unique source node $\mathbf{s}$ and sink node $\mathbf{t}$. Importantly, each transcript $T \in \mathcal{T}$ that is compatible with an alignment $\mathcal{R}$ corresponds to an $\mathbf{s} - \mathbf{t}$ path $\pi(T)$ in G. Here, a *path* π is a subset of edges E that can be ordered $(\mathbf{v}_1, \mathbf{w}_1), \ldots, (\mathbf{v}_{|\pi|}, \mathbf{w}_{|\pi|})$ such that $\mathbf{w}_i = \mathbf{v}_{i+1}$ for all $i \in [|\pi| - 1] = \{1, \ldots, |\pi| - 1\}$. While splice graphs are DAGs and typically have a unique source and sink node as well, they do not necessarily contain a Hamiltonian path [63, 72, 73, 74].

Our goal is to find a set $\mathcal{T}$ of transcripts and their abundances $\mathbf{c}$ that maximize the posterior probability

$$\Pr(\mathcal{T}, \mathbf{c} \mid \mathcal{R}) \propto \Pr(\mathcal{R} \mid \mathcal{T}, \mathbf{c}) \Pr(\mathcal{T}, \mathbf{c}). \tag{3.1}$$

Under an uninformative, flat prior, this is equivalent to maximizing the probability $\Pr(\mathcal{R} \mid \mathcal{T}, \mathbf{c})$. We use the segment graph G to compute the probability $\Pr(\mathcal{R} \mid \mathcal{T}, \mathbf{c})$ of observing an alignment $\mathcal{R}$ given transcripts $\mathcal{T}$ and abundances $\mathbf{c}$. We follow the generative model which has been extensively used for transcription quantification [68, 69, 75]. The notations used in this paper best resemble the formulation described in [74]. Let $\mathcal{R}$ be composed of reads be $\{r_1, \ldots, r_n\}$ and the set $\mathcal{T}$ of transcripts be $\mathcal{T} = \{T_1, \ldots, T_k\}$ with lengths $L_1, \ldots, L_k$ and abundances $\mathbf{c} = [c_1, \ldots, c_k]$. In line with current literature, reads $\mathcal{R}$ are generated independently from transcripts $\mathcal{T}$ with abundances $\mathbf{c}$. Further, we must marginalize over the set of transcripts $\mathcal{T}$ as the transcript of origin of any given read is typically unknown, since $\ell \ll L$. Moreover, we assume that the fixed read length ℓ is much smaller than the length L_i of any transcript T_i. As such, we that $\Pr(\mathcal{R} \mid \mathcal{T}, \mathbf{c})$ equals

$$\begin{aligned}\Pr(\mathcal{R} \mid \mathcal{T}, \mathbf{c}) &= \prod_{j=1}^{n} \Pr(r_j \mid \mathcal{T}, \mathbf{c}) \\ &= \prod_{j=1}^{n} \frac{1}{\sum_{b=1}^{k} c_b L_b} \sum_{i:\pi(T_i) \supseteq \pi(r_j)} c_i, \end{aligned} \tag{3.2}$$

where $\pi(T) \subseteq E$ is the $\mathbf{s} - \mathbf{t}$ path corresponding to transcript T and $\pi(r) \subseteq E$ is the path induced by the ordered sequence of segments (or nodes of G) spanned by read r. By construction, $\pi(T) \supseteq \pi(r)$ is a necessary condition for transcript T to be a candidate transcript of origin of read r. Appendix C.3 gives the derivation of the above equation (Eq. (3.2)). Our goal is to find $\arg\max_{\mathcal{T},\mathbf{c}} \Pr(\mathcal{R} \mid \mathcal{T}, \mathbf{c})$, leading to the following problem.

Problem 3.1 (Discontinuous Transcript Assembly (DTA))**.** Given alignment $\mathcal{R}$ and integer k, find discontinuous transcripts $\mathcal{T} = \{T_1, \ldots, T_k\}$ and abundances $\mathbf{c} = [c_1, \ldots, c_k]$ such that (i) each transcript $T_i \in \mathcal{T}$ is an $\mathbf{s}-\mathbf{t}$ path in segment graph G, and (ii) $\Pr(\mathcal{R} \mid \mathcal{T}, \mathbf{c})$ is maximum.

The probability $P(\mathcal{R} \mid \mathcal{T}, \mathbf{c})$, in Eq. (3.2), is expressed in terms of the observed reads and their induced paths $\pi(r) \subseteq E(G)$ in the segment graph G. In the Methods section, we describe a more concise way of expressing the probability $P(\mathcal{R} \mid \mathcal{T}, \mathbf{c})$ using the fact that the segment graph G is a DAG with a unique Hamiltonian path. This concise characterization enables us to design a progressive heuristic that incorporates an efficient mixed linear integer program (MILP) to solve the DTA problem (details are in the Methods section). Our resulting method, Jumper, is implemented in Python 3 using `Gurobi` [76] (version 9.0.3) to solve the MILP and `pysam` [77] for reading and processing the input BAM file. Jumper is available at `https://github.com/elkebir-group/Jumper`.

3.3 COMBINATORIAL CHARACTERIZATION OF SOLUTIONS

Eq. (3.2) defines the probability $\Pr(\mathcal{R} \mid \mathcal{T}, \mathbf{c})$ in terms of the observed reads r and their induced paths $\pi(r) \subseteq E(G)$ in the segment graph G. The authors in [74] use this characterization of reads as paths in a general *splice* graph to account for ambiguity in the transcript of origin for the reads. For a general splice graph, such a characterization is required to capture all the possible observed reads. However, in our setting, where the segment graph G is a DAG with a unique Hamiltonian path, it is possible to describe each read and each transcript *uniquely* in a more concise form. Each path in the segment graph is characterized by a set of *non-overlapping* discontinuous edges. To describe this, we introduce the following definition.

Definition 3.3. Two edges $(\mathbf{v} = [v^-, v^+], \mathbf{w} = [w^-, w^+])$ and $(\mathbf{x} = [x^-, x^+], \mathbf{y} = [y^-, y^+])$ of G *overlap* if the open intervals (v^+, w^-) and (x^+, y^-) intersect, *i.e.* $(v^+, w^-) \cap (x^+, y^-) \neq \emptyset$.

For any transcript T corresponding to an $\mathbf{s} - \mathbf{t}$ path in G, for which we are only given its discontinuous edges $\sigma(T)$, the continuous edges of T are uniquely determined by G and $\sigma(T)$. That is, the continuous edges of T equal precisely the subset of continuous edges $E^{\rightarrow}$ that do *not* overlap with any of the discontinuous edges in $\sigma(T)$. Conversely, given an $\mathbf{s} - \mathbf{t}$ path $\pi(T)$ of G the corresponding set of discontinuous edges is given by $\sigma(T) = \pi(T) \cap E^{\curvearrowright}$. Thus, we have the following proposition with the proof in Appendix C.4.

Proposition 3.1. There is a bijection between subsets of discontinuous edges that are pairwise non-overlapping and $\mathbf{s} - \mathbf{t}$ paths in G.

In a similar vein, rather than characterizing a read r by its induced path $\pi(r) \subseteq E$ in the segment graph, we characterize a read r by a pair $(\sigma^{\oplus}(r), \sigma^{\ominus}(r))$ of *characteristic discontinuous edges*. Here, $\sigma^{\oplus}(r)$ is the set of discontinuous edges that must be present in any transcript that could generate read r, *i.e.* $\sigma^{\oplus}(r) = \pi(r) \cap E^{\curvearrowright}$. Conversely, $\sigma^{\ominus}(r)$ is the set of discontinuous edges that must be absent in any transcript that could generate read r due to the unidirectional nature of RdRp transcription. Thus, the set $\sigma^{\ominus}(r)$ consists of discontinuous edges $E^{\curvearrowright} \setminus \sigma^{\oplus}$ that overlap with any edge in $\pi(r)$. Clearly, while $\sigma^{\oplus}(r) \cap \sigma^{\ominus}(r) = \emptyset$, it need not hold that $\sigma^{\oplus}(r) \cup \sigma^{\ominus}(r)$ equals $E^{\curvearrowright}$ (see Figure 3.2b). Formally, we define $(\sigma^{\oplus}(r), \sigma^{\ominus}(r))$ as follows.

Definition 3.4. The *characteristic discontinuous edges* of a read r are a pair $(\sigma^{\oplus}(r), \sigma^{\ominus}(r))$ where $\sigma^{\oplus}(r)$ is the set of discontinuous edges present in read r, *i.e.* $\sigma^{\oplus}(r) = \pi(r) \cap E^{\curvearrowright}$, and $\sigma_i^{\ominus}$ is the set of discontinuous edges $(\mathbf{v} = [v^-, v^+], \mathbf{w} = [w^-, w^+]) \in E^{\curvearrowright} \setminus \sigma^{\oplus}(r)$ that *overlaps* with an edge $(\mathbf{x} = [x^-, x^+], \mathbf{y} = [y^-, y^+])$ in $\pi(r)$.

We have the following result with the proof given in Appendix C.4.

Proposition 3.2. Let G be a segment graph, T be a transcript and r be a read. Then, $\pi(T) \supseteq \pi(r)$ if and only if $\sigma(T) \supseteq \sigma^{\oplus}(r)$ and $\sigma(T) \cap \sigma^{\ominus}(r) = \emptyset$.

Hence, we may rewrite the likelihood $\Pr(\mathcal{R} \mid \mathcal{T}, \mathbf{c})$ as

$$\prod_{j=1}^{n} \frac{1}{\sum_{b=1}^{k} c_b L_b} \sum_{i \in X(\mathcal{T}, \sigma_j^{\oplus}, \sigma_j^{\ominus})} c_i. \tag{3.3}$$

where $X(\mathcal{T}, \sigma_j^{\oplus}, \sigma_j^{\ominus})$ be the subset of indices i corresponding to transcripts $T_i \in \mathcal{T}$ where $\sigma(T_i) \supseteq \sigma_j^{\oplus}$ and $\sigma(T_i) \cap \sigma_j^{\ominus} = \emptyset$. Note that the only difference between Eq. (3.3) and the formulation in Eq. (3.2) is the way that the candidate transcripts of origin for a given read are described. In Eq. (3.2), they are described as paths in the splice graph wheres in Eq. (3.3), they are described by sets of pairwise non-overlapping discontinuous edges in the segment graph. This leads to the following theorem.

Theorem 3.1. For any alignment $\mathcal{R}$, transcripts $\mathcal{T}$ and abundances $\mathbf{c}$, Equations (3.2) and (3.3) are identical.

Although we have described the formulation for single-end reads, this characterization is applicable to paired-end and even synthetic long reads. Moreover, our implementation provides support for both single-end and paired-end read samples with a fixed read length. The above characterization using discontinuous edges allows us to reduce the number of terms in the likelihood function since multiple reads can be characterized by the same characteristic discontinuous edges. We describe this in detail in the next section.

3.4 METHODS

To solve the DTA problem, we use the results of the section on 3.3 to write a more concise form of the likelihood. Specifically, let $\mathcal{S} = \{(\sigma_1^{\oplus}, \sigma_1^{\ominus}), \ldots, (\sigma_m^{\oplus}, \sigma_m^{\ominus})\}$ be the set of characteristic discontinuous edges generated by the reads in alignment $\mathcal{R}$. Let $\mathbf{d} = \{d_1, \cdots, d_m\}$ be the number of reads that map to each pair in $\mathcal{S}$. Using that reads r with identical characteristic discontinuous edges $(\sigma^{\oplus}(r), \sigma^{\ominus}(r))$ have identical probabilities $\Pr(r \mid \mathcal{T}, \mathbf{c})$, we obtain the following mathematical program for the log-likelihood $\log \Pr(\mathcal{R} \mid \mathcal{T}, \mathbf{c})$ (see Appendix C.3 for derivation).

$$\max_{\mathcal{T},\mathbf{c}} \sum_{j=1}^{m} d_j \log \sum_{i \in X(\mathcal{T},\sigma_j^{\oplus},\sigma_j^{\ominus})} c_i - n \log \sum_{b=1}^{k} c_b L_b \tag{3.4}$$

$$\text{s.t. } \pi(T_i) \text{ is an } \mathbf{s}-\mathbf{t} \text{ path} \tag{3.5}$$

$$\text{in the segment graph } G, \forall i \in [k],$$

$$\sum_{i=1}^{k} c_i = 1, \tag{3.6}$$

$$c_i \geq 0, \quad \forall i \in [k]. \tag{3.7}$$

Observe that the first sum (over reads) is concave and the second sum (over transcripts) is convex. Since we are maximizing, our objective function would ideally be concave. In Appendix C.4, we prove the following lemma, which enables us to remove the second term using a scaling factor for the relative abundances $\mathbf{c}$ that does not alter the solution space.

Lemma 3.1. Let $D > 0$ be a constant, $\overline{c}_i(\mathbf{c}) = c_i D / \sum_{j=1}^{k} c_j L_j$ and $c_i(\overline{\mathbf{c}}) = \overline{c}_i / \sum_{j=1}^{k} \overline{c}_j$ for all $i \in [k]$. Then, $(\mathcal{T}, \mathbf{c} = [c_1(\overline{\mathbf{c}}), \ldots, c_k(\overline{\mathbf{c}})])$ is an optimal solution for (3.4)-(3.7) if and only if $(\mathcal{T}, \overline{\mathbf{c}} = [\overline{c}_1(\mathbf{c}), \ldots, \overline{c}_k(\mathbf{c})])$ is an optimal solution for

$$\max_{\mathcal{T},\overline{\mathbf{c}}} \sum_{j=1}^{m} d_j \log \sum_{i \in X(\mathcal{T},\sigma_j^{\oplus},\sigma_j^{\ominus})} \overline{c}_i \tag{3.8}$$

$$\text{s.t. } \pi(T_i) \text{ is an } \mathbf{s}-\mathbf{t} \text{ path} \tag{3.9}$$

$$\text{in the segment graph } G, \forall i \in [k],$$

$$\sum_{i=1}^{k} \overline{c}_i L_i = D, \tag{3.10}$$

$$\overline{c}_i \geq 0, \quad \forall i \in [k]. \tag{3.11}$$

We formulate the mathematical program given in Lemma 3.1 as a mixed integer linear program. More specifically, we encode (i) the composition of each transcript T_i as a set $\sigma(T_i)$ of non-overlapping discontinuous edges, (ii) the abundance c_i and length L_i of each transcript T_i, (iii) the total abundance $\sum_{i \in X(\mathcal{T},\sigma_j^{\oplus},\sigma_j^{\ominus})} c_i$ of transcripts supported by characteristic discontinuous edges $(\sigma_j^{\oplus}, \sigma_j^{\ominus})$, and (iv) a piecewise linear approximation of the log function using a user-specified number h of breakpoints. We will describe (i) and (ii) in the following and refer to Appendix A.2 for (iii) and (iv).

Transcript composition. We begin modeling (3.9), which states that each transcript T_i must correspond to an $\mathbf{s}-\mathbf{t}$ path in the segment graph G. Using Proposition 3.1, we introduce binary variables $\mathbf{x} \in \{0,1\}^{|E^{\curvearrowright}|\times k}$ to encode the presence of discontinuous edges in each of the k $\mathbf{s}-\mathbf{t}$ paths corresponding to the k transcripts in $\mathcal{T}$. For any discontinuous edge $e = (\mathbf{v} = [v^-, v^+], \mathbf{w} = [w^-, w^+])$, let $I(e)$ denote the open interval (v^+, w^-) between the two segments $\mathbf{v}$ and $\mathbf{w}$. By Proposition 3.1, it must hold that $I(e) \cap I(e') = \emptyset$ for any two distinct discontinuous edges e and e' assigned to the same transcript. To encode this, we impose

$$x_{e,i} + x_{e',i} \leq 1, \quad \forall i \in [k], e, e' \in E^{\curvearrowright} \tag{3.12}$$

$$\text{s.t. } e \neq e', I(e) \cap I(e') \neq \emptyset. \tag{3.13}$$

Transcript abundance and length. We introduce non-negative continuous variables $\mathbf{c} = [c_1, \ldots, c_k]$ that encode the abundance of the k transcripts. The scale of these abundances depends on the choice of D. We choose $D = \ell^*$ where ℓ^* is the length of the shortest $\mathbf{s}-\mathbf{t}$ path in the segment graph G. Substituting $D = \ell^*$ into (3.10) yields $\sum_{i=1}^{k} c_i L_i = \ell^*$.

Since $c_i L_i \leq \sum_{j=1}^{k} c_j L_j = \ell^*$ and $L_i \geq \ell^*$, we have that $c_i \leq 1$. To model the product $c_i L_i$ of the length L_i of a transcript T_i and its abundance c_i, we focus on individual discontinuous edges e. For any discontinuous edge $e = (\mathbf{v} = [v^-, v^+], \mathbf{w} = [w^-, w^+])$, let $L(e) = w^- - v^+$ be the length of the interval. Observe that

$$c_i L_i = c_i L - c_i \sum_{e \in \sigma(T_i)} L(e) = c_i L - \sum_{e \in E^{\curvearrowright}} c_i x_{e,i} L(e). \tag{3.14}$$

We introduce continuous variables $z_e \in [0,1]^k$ and encode the product $z_{e,i} = c_i x_{e,i}$ for all $e \in E^{\curvearrowright}$ as

$$z_{e,i} \leq c_i, \quad \forall i \in [k], \tag{3.15}$$

$$z_{e,i} \leq x_{e,i}, \quad \forall e \in E^{\curvearrowright}, i \in [k], \tag{3.16}$$

$$z_{e,i} \geq c_i + x_{e,i} - 1, \quad \forall e \in E^{\curvearrowright}, i \in [k]. \tag{3.17}$$

Therefore, we may represent $\sum_{i=1}^{k} c_i L_i = \ell^*$ as

$$\sum_{i=1}^{k} c_i L - \sum_{i=1}^{k} \sum_{e \in E^{\curvearrowright}} z_{e,i} L(e) = \ell^*. \tag{3.18}$$

The resulting formulation has $O(|E^{\curvearrowright}|k + |E^{\curvearrowright}|m + mh)$ variables, where h is the user-

specified number of breakpoints used in the piecewise linear approximation of the log function. This number includes $|E^{\curvearrowright}|k$ binary variables. The number of constraints is $O(k|E|^2 + |E|km)$.

Progressive heuristic. In practice, the number of discontinuous edges in the segment graph is inflated due to ambiguity in the exact location at which the RdRp jumps as well as sequencing and alignment errors. This leads to large number of binary variables in our MILP (we have $k \cdot |E^{\curvearrowright}|$ binary variables) which can make the MILP intractable. In order to approximately solve the problem with large values of k, we implement a progressive heuristic. Our heuristic takes as input the alignment $\mathcal{R}$ and an integer k, which is the maximum number of transcripts in the solution. At each iteration $p \leq k$, we are given a set $\mathcal{T}$ of $p-1$ previously computed transcripts and seek a new transcript T' by solving the MILP (see Appendix A.3 for details) using function SolveILP with additional constraints to fix the values of the variables that encode the presence/absence of discontinuous edges for the transcripts in $\mathcal{T}$. The resulting reduction in number of binary variables from $|E^{\curvearrowright}|k$ to $|E^{\curvearrowright}|$ improves the running time of the MILP. As an additional optimization, we re-estimate the abundances of a new set $\mathcal{T}'$ of transcripts. This set contains all transcripts in $\mathcal{T}$ as well additional transcripts corresponding to all possible subsets of discontinuous edges $\sigma(T')$ of the newly identified transcript T', identified by the function Expand. We solve a linear program (see Appendix A.3 for details) with function SolveLP to re-estimate the abundances $\mathbf{c}'$ of $\mathcal{T}'$, retaining only the top p transcripts T_i from $\mathcal{T}'$ with the largest abundances $c_i L_i$. We terminate upon convergence, *i.e.* if $\mathcal{T} = \mathcal{T}'$, or if the number p of iterations reaches the number k. Algorithm 3.1 provides the pseudo code of the progressive heuristic implemented in Jumper. The details of the subproblems SolveILP and SolveLP are given in Appendix A.3.

Implementation details. Matching core sequences that mediate the discontinuous transcription by RdRp lead to ambiguity in precise location of breakpoint during alignment of spliced reads. Therefore, in practice we observe multiple discontinuous edges with closely spaced 5' and 3' breakpoints. Moreover, false positive discontinuous edges are introduced due to sequencing and alignment errors. We use a threshold on the number of spliced reads supporting a discontinuous edge to filter false positive edges with low support. This parameter can also be used to reduce computational burden and focus on the highly expressed transcripts in the sample. A discussion on the choice of the thresholding parameter Λ is provided in Appendix A.4.

Algorithm 3.1: JUMPER($\mathcal{R}$, k)

$(\mathcal{T}, \mathbf{c}) \leftarrow (\emptyset, [])$
for $p \leftarrow 1$ **to** k **do**
  $T' \leftarrow$ SOLVEILP($\mathcal{T}$)
  $\mathcal{T}' \leftarrow \mathcal{T} \cup$ EXPAND(T')
  $\mathbf{c}' \leftarrow$ SOLVELP($\mathcal{T}'$)
  Sort $(\mathcal{T}', \mathbf{c}')$ s.t. $L_i c'_i \geq L_{i+1} c'_{i+1}$ for all $i \in \{1, \ldots, |\mathcal{T}'| - 1\}$
  $(\mathcal{T}', \mathbf{c}') \leftarrow (\{T_1, \ldots, T_p\}, [c'_1, \ldots, c'_p])$
  if $\mathcal{T}' \neq \mathcal{T}$ **then**
    $(\mathcal{T}, \mathbf{c}) \leftarrow (\mathcal{T}', \mathbf{c}')$
  end
  else
    return $(\mathcal{T}, \mathbf{c})$
  end
end
return $(\mathcal{T}, \mathbf{c})$

3.5 RESULTS

We begin by establishing terminology that will be used throughout this section. A discontinuous edge ($\mathbf{v} = [v^-, v^+], \mathbf{w} = [w^-, w^+]$) is *canonical* provided its 5' junction v^+ occurs in the transcription regulating leader sequence (TRS-L), *i.e.* between positions 50 and 85[1], and the first occurrence of 'AUG' downstream of the 3' junction w^- position coincides with the start codon of a known ORF, otherwise the discontinuous edge is called *non-canonical*. In a similar vein, a transcript is *canonical* if it contains at most one canonical and no non-canonical discontinuous edges, otherwise the transcript is *non-canonical*. We ran all experiments on a server with two 2.6 GHz CPUs and 512 GB of RAM.

3.5.1 Simulations

We generated our simulation instances using a segment graph G obtained from a short-read sample (SRR11409417). Following Kim *et al.* [55], we used `fastp` to trim short reads (trimming parameter set to 10 nucleotides), which were input to `STAR` run in two-pass mode yielding an alignment $\mathcal{R}$. Figure 3a shows the sashimi plot of the *canonical* and the *non-canonical* discontinuous edges (mappings) supported by the reads in the sample. From $\mathcal{R}$, we obtained G by only including discontinuous edges supported by at least 20 reads. The segment graph G has $|V| = 39$ nodes and $|E| = 67$ edges, which include $|E^{\curvearrowright}| = 29$ dis-

[1]This range contains the TRS-L regions of the SARS-CoV-1 [78], SARS-CoV-2 [55] and MERS-CoV [78] genomes analyzed in this paper.

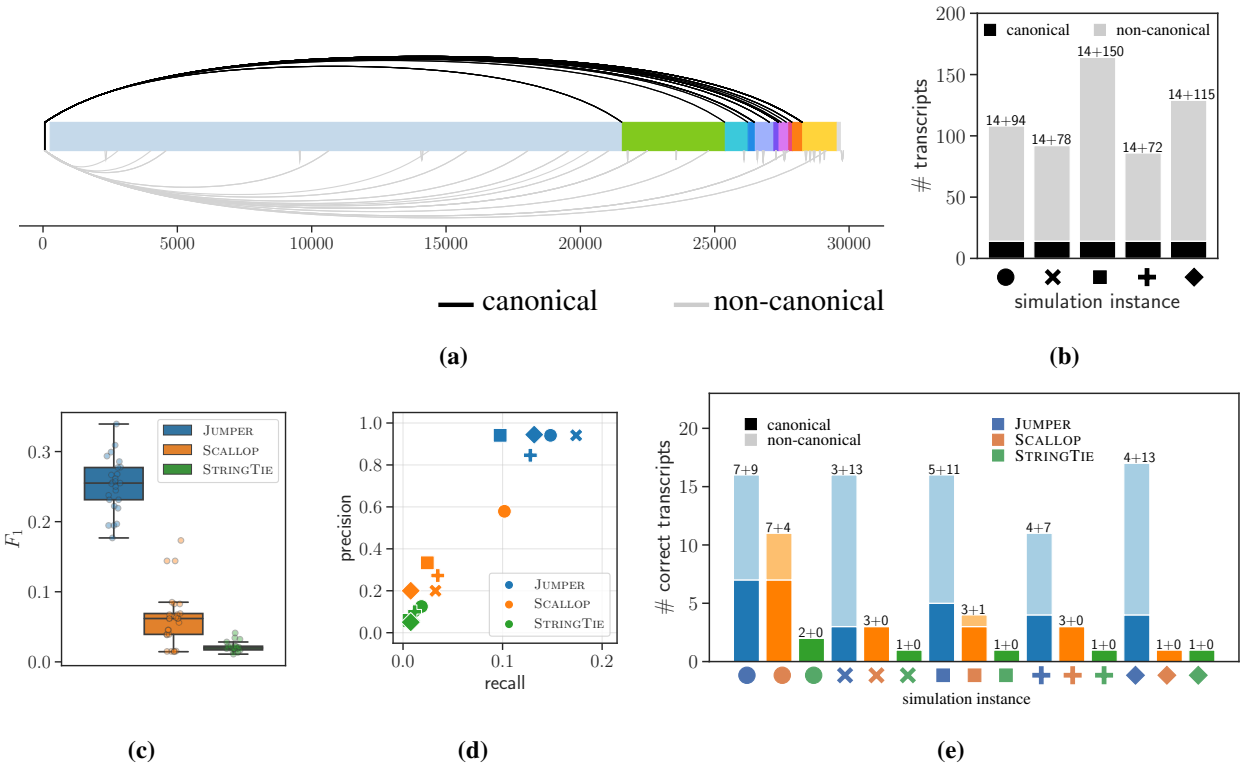


Figure 3.3: JUMPER consistently outperforms SCALLOP [63] and STRINGTIE [64] in reconstruction of viral transcripts from simulated SARS-CoV-2 sequencing data. (a) Sashimi plot showing the canonical (black) and non-canonical (gray) discontinuous mappings supported by reads in short-read sample SRR11409417. (b) Number of canonical and non-canonical transcripts for 5 simulation instances of $(\mathcal{T}, \mathbf{c})$ generated under the negative-sense discontinuous transcription model. (c) F_1 score of the three methods (JUMPER, SCALLOP and STRINGTIE) for all the 25 simulated instances under the negative-sense discontinuous transcription model. (d) Precision and recall values of the three methods with one of sequencing experiment for each simulated instance of $(\mathcal{T}, \mathbf{c})$ under the negative-sense discontinuous transcription model as input. (e) Total number of canonical and non-canonical transcripts recalled by the three methods for the simulated instances shown in panel (d).

continuous edges and $|E^{\rightarrow}| = 38$ continuous edges. The discontinuous edges are subdivided into 14 canonical discontinuous edges that produce a known ORF and 15 non-canonical discontinuous edges. Next, we generated transcripts $\mathcal{T}$ and their abundances $\mathbf{c}$ from G using the negative-sense discontinuous transcription model (described in Appendix D.2). Upon generating the transcripts, we simulated the generation and sequencing of RNA-seq data, and aligned the simulated reads using `STAR` [71]. We generated 5 independent pairs $(\mathcal{T}, \mathbf{c})$ of transcripts and abundances (Figure 3b). For each pair $(\mathcal{T}, \mathbf{c})$ we generated 5 paired-end short read sequencing simulations using `polyester` [79]. Thus, in total we generated $5 \times 5 = 25$ simulation instances.

We compare the performance of our method JUMPER with two other reference-based transcript assembly methods, SCALLOP and STRINGTIE. Note that our method, JUMPER, does *not* use prior knowledge about the underlying negative-sense discontinuous transcription model to infer the viral transcripts from the simulated data. To avoid including false-positive discontinuous edges, we set $\Lambda = 100$ so that JUMPER discards discontinuous edges with fewer than 100 supporting reads. For SCALLOP and STRINGTIE, we performed a sweep on their input parameters and report the best results here. We begin by comparing the transcripts predicted by the three methods to the ground truth transcripts. Specifically, a predicted transcript is *correct* if there exists a transcript in the ground truth whose junction positions match the predicted junctions positions within a tolerance of 10 nucleotides.

Figure 3c shows the F_1 score (harmonic mean of recall and precision) of the three methods for all the simulation instances, showing that JUMPER achieves a higher F_1 score (median of 0.255 and range $[0.176, 0.339]$) compared to SCALLOP (median of 0.062 and range $[0.0145, 0.173]$) and STRINGTIE (median of 0.019 and range $[0.0114, 0.0412]$). Fig. E.9 shows that JUMPER's improved performance holds for both the recall and the precision with running times comparable to the SCALLOP and STRINGTIE. To investigate the effect of threshold parameter Λ on the performance of JUMPER, we ran our method on the simulated instances with $\Lambda \in \{10, 50, 100, 200\}$. Fig. E.10 shows that JUMPER outperforms SCALLOP and STRINGTIE for all values of Λ, although it incurs significantly more runtime for $\Lambda = 10$.

To better understand the tradeoff between precision and recall, we zoom in on five simulation instances with distinct pairs $(\mathcal{T}, \mathbf{c})$. Figure 3d shows the precision and recall achieved by each method for each of these five simulation instances, demonstrating that JUMPER consistently outperforms both SCALLOP and STRINGTIE. On average, JUMPER recalls 5 times more transcripts than SCALLOP and 11 times more transcripts than STRINGTIE while also having higher precision in all simulated cases. Fig. E.11 shows that all three methods produce similar precision and recall values for different sequencing replicates of the same simulated instance of $(\mathcal{T}, \mathbf{c})$, demonstrating consistency in results. Finally, Figure 3e shows the number of canonical and non-canonical transcripts generated by the three methods that match the ground truth for each simulated instance, with JUMPER consistently recalling a larger number of ground-truth canonical and non-canonical transcripts.

In summary, we found that JUMPER correctly predicts higher number of both canonical and non-canonical transcripts compared to SCALLOP and STRINGTIE for all the simulated instances (summarized in Table E.4). We observe similar trends on simulated instances of a human gene (see Appendix E.4).

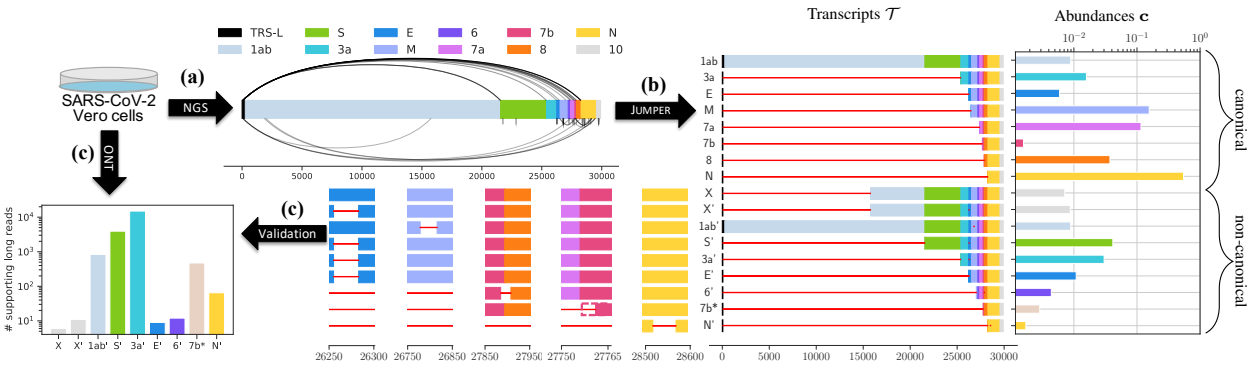


Figure 3.4: Using short-read data of SARS-CoV-2 infected Vero cells [55], JUMPER identifies canonical and non-canonical transcripts that are well supported by long-read sequences of the same sample. (a) The segment graph for the short-read data contains both canonical (above) and non-canonical (below) edges. (b) JUMPER assembles 8 canonical transcripts and 9 non-canonical transcripts and estimates their abundances with zoomed-in view of the non-canonical transcripts X, X', 1ab', S', 3a', E', 6', 7b* and N'. (c) All non-canonical transcripts predicted by JUMPER are well supported by long-read data. (NGS: Next Generation Sequencing; ONT: Oxford Nanopore Technologies)

3.5.2 Viral Transcript Assembly in SARS-CoV-2 Infected Vero Cells

Recently, Kim *et al.* [55] explored the transcriptomic architecture of SARS-CoV-2 by performing short-read as well as long-read sequencing of Vero cells infected by the virus. The authors used oligo(dT) amplification, which targets the poly(A) tail at the 3' end of messenger RNAs, thus limiting positional bias that would occur when using SARS-CoV-2 specific primers [80, 81]. Subsequently, the authors aligned the resulting reads using splice-aware aligners, `STAR` [71] for the short-read sample (median depth of 1763) and `minimap2` [82] for the long-read sample (median depth of 6707 and mean length of 2875 bp). For both complementary sequencing techniques, the authors observed phasing reads that were indicative of canonical as well as non-canonical transcription events. While this previous work quantified the fraction of phasing reads supporting each discontinuous transcription event, it did not attempt to assemble complete viral transcripts.

We used JUMPER to reconstruct the SARS-CoV-2 transcriptome of the short-read sequencing sample using the BAM file obtained by running Kim *et al.*'s pipeline [55]. This was followed by running SALMON to identify precise transcript abundances. We note that running SCALLOP on the short-read data resulted in only a single, complete canonical transcript (corresponding to 'N') but required subsampling of the BAM file (to 20%) due to memory constraints, whereas STRINGTIE produced two incomplete transcripts ('ORF3a' and a non-canonical transcript with low support). On a segment graph with $|V| = 59$ nodes and $|E| = 93$ edges comprised of $|E^{\curvearrowright}| = 35$ most abundant discontinuous edges, 18 of which

canonical and 17 non-canonical (Figure 3.4a), JUMPER identified 33 transcripts, 17 of which have an abundance of at least 0.001 as determined by SALMON (Figure 3.4b). A subset of 8 transcripts are canonical, containing at most one discontinuous edge with the 5' junction in TRS-L and the first ATG downstream of the 3' junction coinciding with the start codon of a known ORF. These canonical transcripts correspond to ORF1ab, ORF3a, E, M, ORF7a, ORF7b, ORF8, N. In particular, ORF1ab (abundance of 0.008) corresponds to the complete viral genome, necessary for viral replication. Notably, ORF10 is the only missing ORF in the identified transcriptome, which is in line with previous studies [55, 57] that did not find evidence for active transcription of ORF10.

As mentioned, JUMPER inferred 9 non-canonical transcripts, denoted as X, X', 1ab', S', 3a', 6', E', 7b* and N'. Among these, transcripts 1ab', S', 3a' and 6' encode for the 1ab polypeptide, spike protein S, accessory protein 3a and accessory protein 6, respectively. Transcripts X and X' both contain the discontinuous edge going from position 68 to 15774, with the latter containing an additional discontinuous edge from position 26256 to 26284. The 5' end of the common discontinuous edge occurs within TRS-L, whereas the 3' end occurs in the middle of ORF1b but is out of frame with respect to the starting position of ORF1b (13468). Specifically, the start codon 'ATG' downstream of the 3' end is located at position 15812 and occurs within nsp12 (RdRp) and the first stop codon is located at position 15896, encoding for a peptide sequence of 28 amino acids. Interestingly, when we examined the reference genome, we observed matching sequences "GAACTTTAA" near the 5' and 3' junctions of the discontinuous edge common to X and X', possibly explaining why the viral RdRp generated this jump (Fig. E.12a,b). Strikingly, both matching sequences are conserved within the *Sarbecovirus* subgenus but not in other subgenera of the *Betacoronavirus* genus (Fig. E.12a,c). To further corroborate this transcript, we examined short and long-read SARS-CoV-2 sequencing samples from the NCBI Sequence Read Archive (SRA). Specifically, we looked for the presence of reads potentially originating from transcript X focusing on high-quality samples with 100 or more leader-spanning reads (reads whose 5' end maps to the TRS-L region). We say a read r *supports* a transcript T if the discontinuous edges of r exactly match those of T, *i.e.* $\pi(r) \subseteq \pi(T)$ and $|\sigma^{\oplus}(r)| = |\sigma(T)|$ (Fig. E.13). We found ample support for transcript X in both short and long-read samples on SRA, with 100 out of 351 short-read samples and 81 out of 653 long-read samples having more than 0.1% of leader-spanning reads supporting transcript X (Fig. E.14). We note that although this discontinuous transcription event was also observed in [57], the authors found no evidence of this transcript leading to protein product in the ribo-seq data. Further research into a potentially regulatory function of this transcript is required.

As stated, the difference between transcripts X and X' is that the latter includes an ad-

ditional discontinuous edge, corresponding to a short jump of $\sim$ 27 nucleotides between positions 26256 and 26284. This is an in frame deletion inside ORF E, resulting in the loss of 9 amino acids that span the N-terminal domain (4 amino acids) and the transmembrane domain (5 amino acids) of the E protein [83]. A similar in-frame deletion of 24 nucleotides (from position 26259 to 26284) was observed by Finkel *et al.* [57] that resulted in the loss of a subset of 8 out of the 9 amino acids in the deletion that we observed. Furthermore, it is possible that this common deletion is being selected for during passage in Vero E6 cells, which were used by both Kim *et al.* [55] and Finkel *et al.* [57]. Non-canonical transcripts S', 3a' and E' also contain the same discontinuous edge from position 26256 to 26284. While transcript E' produces a version of protein E with 9 missing amino acids, transcripts S' and 3a' produce complete viral proteins S and 3a, respectively. Non-canonical transcript 6' differs from the canonical transcript 6, containing a jump from position 27886 to 27909. This jump is downstream of ORF6 and therefore does not disrupt the translation of accessory protein 6. Similarly, transcript 1ab' has a single jump from position 26779 to 26817, which is downstream of the ORF1ab gene and therefore will yield the complete polypeptide 1ab. Transcript 7b*, on the other hand, has a single discontinuous edge from position 71 to 27762. The start codon 'ATG' downstream of the 3' end occurs at position 27825, maintaining the frame of 7b, and thus leading to an N-terminal truncation [55] of 23 amino acids. Interestingly, transcript 7b and transcript 7b* appear with similar abundances in our solution. Finally, transcript N' has one canonical discontinuous edge from TRS-L (position 65) to the transcription regulating body sequence (TRS-B) region corresponding to ORF N (position 28255) and an additional jump from position 28525 to 28577, which leads to an in-frame deletion of 17 amino acids in the N-terminal RNA-binding domain [84, 85] of ORF N. Thus, with the exception of transcripts X and X', the non-canonical transcripts identified by JUMPER either produce complete viral proteins (1ab', S', 3a', 6'), contain in-frame deletions in the middle of known proteins (E', N') or produce N-terminally truncated proteins (7b*).

One of the major findings of the Kim *et al.* paper [55] is that the SARS-CoV-2 transcriptome is highly complex owing to numerous non-canonical discontinuous transcription events. Strikingly, our results show that these non-canonical transcription events do not significantly change the resulting proteins. Indeed, we find that 4 out of the 9 non-canonical transcripts produce a complete known viral protein and the total abundance of the predicted transcripts that produce a complete known viral protein is 0.968. Moreover, these predicted transcripts account for more than 90% of the reads in the sample according to the estimates provided by SALMON.

Typically, reads from short-read sequencing samples are not long enough to contain more

than one discontinuous edge. As a result, short-read data can only provide direct evidence for transcripts with closely spaced discontinuous edges. For instance, we observed ample support (63485 short reads) for the predicted non-canonical transcript E', which has two discontinuous edges $(69, 26237)$ and $(26256, 26284)$, in short-read data due to the close proximity of the two discontinuous edges (*i.e.* the discontinuous edges are only $26256-26237 = 19$ nucleotides apart). The other non-canonical transcripts with multiple discontinuous edges, *i.e.* X', S', 3a', 6' and N', have edges that are too far apart to be spanned by a single short read. Using the long-read sequencing data of this sample, we detected supporting long reads that span the exact set of discontinuous edges of all 9 non-canonical transcripts (Figure 3.4c). Moreover, we found support for the canonical transcripts as well (Fig. E.15). Thus, all transcripts identified by JUMPER from the short-read data are supported by direct evidence in the long-read data.

In summary, using JUMPER we reconstructed a detailed picture of the transcriptome of a short-read sequencing sample of Vero cells infected by SARS-CoV-2. While existing methods failed to recall even the reference transcriptome, JUMPER identified transcripts encoding for all known viral protein products. In addition, our method predicted non-canonical transcripts and their abundances, whose presence we subsequently validated on a long-read sequencing sample of the same cells.

3.5.3 Viral Transcript Assembly in SARS-CoV-2 Infected A549 Cells with and without Treatment

To demonstrate that JUMPER can be used to understand the effect of drugs on the viral transcriptome, we analyzed a recent dataset by Blanco et al. [3] who studied the host transcriptional response to SARS-CoV-2 and other viral infections using various cell lines. We focused on A549 lung alveolar cell line samples that were sequenced after 24 hours of SARS-CoV-2 infection. There are a total of eight samples, four of which were pre-treated with ruxolitinib for 1 hour before the infection and the remaining four were untreated. Ruxolitinib is a JAK1 and 2 kinase inhibitor, which blocks type-I interferon (IFN-I) signaling necessary to engage cellular antiviral defenses [86, 87]. Specifically, the four samples without treatment are SRR11573904 (median depth of 86), SRR11573905 (median depth of 85), SRR11573906 (median depth of 89) and SRR11573907 (median depth of 89), and the four samples treated with ruxolitinib are SRR11573924 (median depth of 90), SRR11573925 (median depth of 91), SRR11573926 (median depth of 91) and SRR11573927 (median depth of 92). We used `fastp` to trim the short reads (trimming parameter set to 10 nucleotides) and we aligned the resulting reads using `STAR` in two-pass mode. We ran JUMPER with the 35

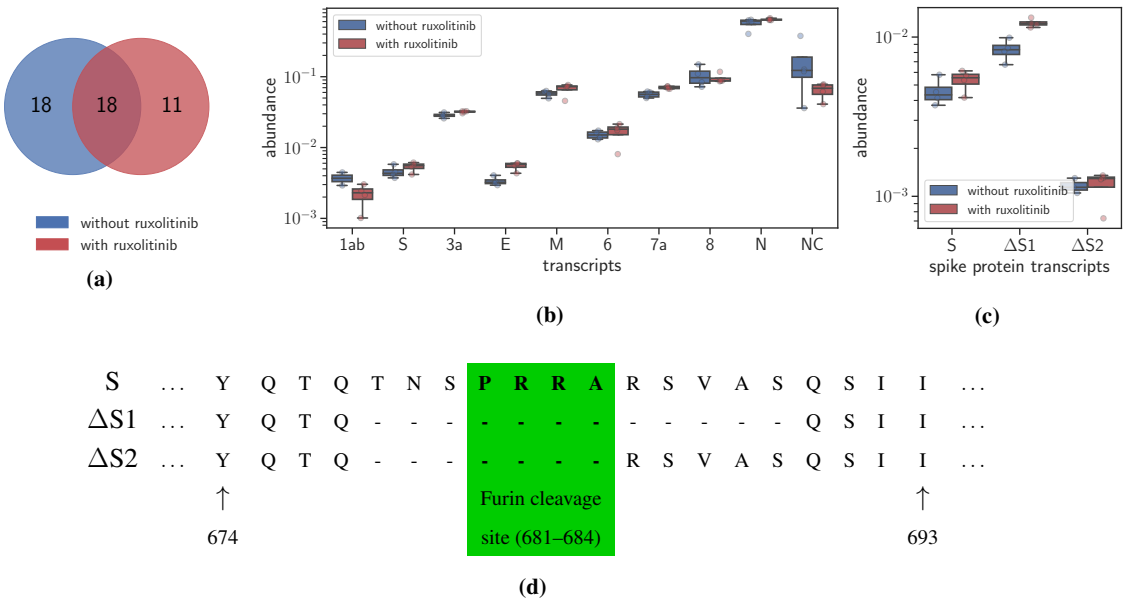


Figure 3.5: JUMPER enables analysis of drug response in SARS-CoV-2 infected cells [3] at the transcript level. (a) A Venn diagram shows in the number of transcripts reconstructed from samples with and without treatment with ruxolitinib. Fig. E.16 shows the distribution of the 18 transcripts that are common between samples with and without treatment while Table E.3 describes these transcripts. (b) Abundance of the transcripts yielding canonical proteins in the samples along with 'NC' depicting the abundance of the non-canonical transcripts. (c) Abundance of the transcripts yielding the spike protein (S) and its variants ΔS1 and ΔS2 whose structure is described in (d).

most abundant discontinuous edges in the segment graph. Similarly to the previous analysis, we restricted our attention to transcripts identified by JUMPER that have more than 0.001 abundance as estimated by SALMON [69].

SCALLOP, run with default parameters, identified at most two transcripts for each sample encoding for different variants of ORF N. JUMPER identified a total of 47 transcripts across the eight samples, with 18 of these transcripts present in both ruxolitinib treated and untreated samples (Fig. E.16a,E.16c). We observed that samples with pre-treatment of ruxolitinib cumulatively have fewer transcripts compared to the number of transcripts from samples without any treatment (29 vs. 36 transcripts, Fig. 5a). Strikingly, all the transcripts that are present in two or more samples were also present across the two groups of samples (treated and untreated). Focusing on the 18 common transcripts, Figure C11d in Appendix shows the total number of samples that contain each of these 18 transcripts. A subset of 13 out of these 18 transcripts produce all known canonical viral proteins except 7b. Fig. 5b shows the abundance of the transcripts yielding functional proteins in the samples along with 'NC' depicting the abundance of transcripts producing either non-canonical or

non-functional viral proteins. The abundance of the canonical transcripts, except 1ab, is slightly higher in samples with treatment compared to the samples without treatment. Consequentially, the abundance of non-canonical transcripts is lower in samples with treatment compared to samples without treatment.

There are five non-canonical transcripts, including ∇M, NC1 and NC2, which do not encode for known SARS-CoV-2 proteins but are explained by matching motifs near the 5' and 3' ends of the non-canonical discontinuous edges, described in Table E.3, potentially mediating the jump made by the RdRp to generate these transcripts. Specifically, while transcript ∇M contains a canonical discontinuous edge from the leader to the known TRS-B region of M, it also contains an out-of-frame deletion such that the transcript yields a 116 amino acids long protein which matches the M protein for the first 87 amino acids (total length of protein M is 222 amino acids). Both transcripts NC1 and NC2 contain only one jump with the 5' end within ORF1a. The 3' end of the jump lies within ORF7b and ORF N for transcript NC1 and transcript NC2, respectively. The remaining two non-canonical transcripts, ΔS1 and ΔS2, have in-frame deletions in the region that encodes for the spike protein.

ΔS1 contains an in-frame jump from position 23593 to 23630 resulting in a 12 amino-acid in-frame deletion, while ΔS2 contains a jump from position 23593 to 23615, which results in a 7 amino-acid in-frame deletion in the spike protein (Fig. 5d). Both these deletions overlap with the furin cleavage site (FCS), highlighted in Fig. 5d, which has been the focus of several recent studies [56, 88, 89]. The authors of [56] deduced that the deletion of the FCS enhances the ability of the virus to enter Vero cells and is selected for during passage in Vero E6 cells, a cell line that lacks a working type-I interferon response. The observation of ΔS1 and ΔS2 in infected A549 cell samples can be explained by the fact that Blanco et al. [3] propagated SARS-CoV-2 in Vero E6 cells prior to the infection of the A549 cells. Fig. 5c shows that pre-treatment with ruxolitinib leads to an increase in the abundance of the three transcripts, S (median increase from 0.004 to 0.005), ΔS1 and ΔS2 (median increase from 0.0011 to 0.0012), with the increase being most significant for ΔS1 (median increase from 0.008 to 0.012) with a p-value of 0.015 with the Mann-Whitney u-test. This shows that the response of different variants of the virus to treatment of drugs can differ significantly. In summary, we find that JUMPER enables transcript-level analysis of the viral response to drug treatments.

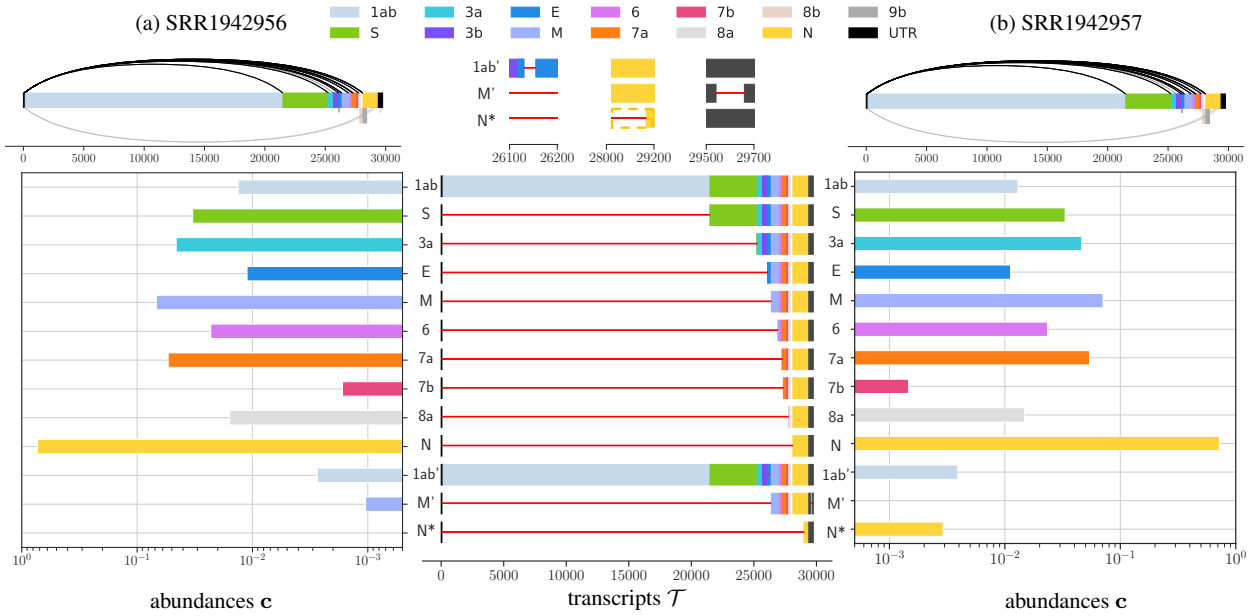


Figure 3.6: JUMPER identifies canonical and non-canonical transcripts that recur in two short-read sequencing samples of SARS-CoV-1 infected Calu-3 cells [70]. For both the samples, we show the segment graph, with canonical (above) and non-canonical (below) discontinuous edges. We also show the predicted transcripts and their abundances in the two samples with a zoomed-in view of the non-canonical transcripts 1ab', M' and N*. UTR: untranslated region.

3.5.4 Viral Transcript Assembly in SARS-CoV-1 and MERS-CoV Infected Cells

To show the generalizability of our method, we considered two other coronaviruses, SARS-CoV-1 and MERS-CoV. We describe the results for two SARS-CoV-1 infected cell samples here and the analysis of three MERS-CoV infected cell samples is described in Appendix E.5.

We analyzed two published samples of human Calu-3 cells infected with SARS-CoV-1 [70], SRR1942956 and SRR1942957, with a median depth of 21,358 and 20,991, respectively. These two samples originate from the same SRA project ('PRJNA279442') whose metadata states that both samples were sequenced 24 hours after infection. We used `fastp` to trim the short reads (trimming parameter set to 10 nucleotides) and we aligned the resulting reads using `STAR` in two-pass mode. We ran JUMPER with the 35 most abundant discontinuous edges in the segment graph. As observed previously, SCALLOP only identified a single transcript corresponding to ORF N in both the samples. By contrast, JUMPER reconstructed 25 transcripts in sample SRR1942956 and 26 transcripts for sample SRR1942957. Similarly to the previous analysis, we discuss the transcripts identified by JUMPER that have more than 0.001 abundance as estimated by SALMON. There are 13 such transcripts for sample SRR1942956 and 13 such transcripts for sample SRR1942957 (Figure 3.6).

SARS-CoV-1 has a genome of length 29751 bp, and consists of 13 ORFs (1ab, S, 3a, 3b, E, M, 6, 7a, 7b, 8a, 8b, N and 9b), two more than SARS-CoV-2. For both samples, JUMPER identified canonical transcripts corresponding to all the ORFs of SARS-CoV-1 except ORF3b, ORF8b and ORF9b (Figure 3.6). Notably, ORF8b and ORF9b share transcription regulating body sequences (TRS-B) with ORF8a and ORF N respectively [90]. More specifically, ORF9b (from position 28130 to 28426) is nested within ORF N (from position 28120 to 29388) with start codons only 10 nucleotides apart and consequently shares the same TRS-B as ORF N. ORF8b (from position 27864 to 28118) intersects with ORF8a (from 27779 to 27898) and previous studies have failed to validate a TRS-B region for ORF8b [90]. One possible way that these ORFs are translated is due to ribosome leaky scanning, which was also hypothesized to lead to ORF7b translation in SARS-CoV-2 [57]. This explains why JUMPER was unable to identify transcripts that directly encode for 8b and 9b. Regarding ORF3b, JUMPER did identify a canonical transcript corresponding to 3b in both samples, but the SALMON estimated abundances (0.00044 for SRR1942956 and 0.0005 for SRR1942957) for these transcripts were below the cut-off value of 0.01. Finally, we note that the relative abundances of the canonical transcripts are consistent for the two samples (Figure 3.6) and ranked in the same order (Fig. E.17), with ORF7b being the least abundant and ORF N having the largest abundance, in line with the observations in SARS-CoV-2 infected cells described in the previous sections.

Figure 3.6 shows the three non-canonical transcripts predicted by JUMPER in the two SARS-CoV-1 samples, designated as 1ab', M' and N*. Since these non-canonical transcripts are in very low abundance, we see some discrepancy in the prediction between the two samples. The first non-canonical transcript 1ab' with a single short discontinuous edge from position 26131 to 26156 is detected in both samples and has a very low abundance compared to the canonical transcript 1ab (0.0133 for 1ab vs 0.002 for 1ab' in SRR1942956, and 0.013 for 1ab vs 0.0039 for 1ab' in SRR1942956). Since the discontinuous edge occurs downstream of the stop codon of 1ab (position 21492), the 1ab' transcript encodes for the complete polypeptide 1ab. The second non-canonical transcript M' has two discontinuous edges: a canonical discontinuous edge from TRS-L (position 65) to TRS-B of ORF M (position 26351) and a non-canonical discontinuous edge from 29542 to 29661 in the 3' untranslated region (UTR). As such, this transcript encodes for the complete M protein. This transcript is detected in SRR1942956 with a very low abundance of 0.001 and is detected at an even lower abundance of 0.0008 in SRR1942957, which is below the cut-off threshold of 0.001. The third non-canonical transcript, denoted by N*, has a single discontinuous edge from position 65 to 29003. While JUMPER and SALMON detected this transcript only in sample SRR1942957 with a low abundance of 0.003, we do observe 119 reads in SRR1942956 (com-

pared to 151 reads in SRR1942957) that support this edge, suggesting that N* might be present in the latter sample at too small of an abundance to be detected. Transcript N* is interesting because the first 'ATG' downstream of the 3' end of its discontinuous edge occurs at position 29071 maintaining the frame of N (which starts at position 28120). Thus transcript N* encodes for an N-terminally truncated version of protein N with 105 amino acids (while protein N is composed of 422 amino acids) and only contain part of the C-terminal dimerization domain [84] of protein N. This is similar to transcript 7b* in the SARS-CoV-2 infected Vero cell sample, which yields a N-terminal truncated version of protein 7b. Detection of non-canonical transcripts such as E' and 7b* in SARS-CoV-2 and N' in SARS-CoV-1 suggests that generation of N-terminally truncated proteins might be a common feature in coronaviruses.

In summary, JUMPER can used to to reconstruct the transcriptome of all viruses in and lead to discovery of novel viral transcripts and corresponding viral proteins. While this section focused on SARS-CoV-1, we observed similar results for MERS-CoV samples, where JUMPER reconstructed transcripts corresponding to all the ORFs with well-supported TRS-B sites along with consistent abundances across the three samples (see Appendix E.5).

3.6 DISCUSSION

In this paper, we formulated the DISCONTINUOUS TRANSCRIPT ASSEMBLY (DTA) problem of reconstructing viral transcripts from short-read RNA-seq data of coronaviruses. The discontinuous transcription process exhibited by the viral RNA-dependent RNA polymerase (RdRp) is distinct from alternative splicing observed in eukaryotes. Our proposed method, JUMPER, is specifically designed to reconstruct the viral transcripts generated by discontinuous transcription and is therefore able to outperform existing transcript assembly methods such as SCALLOP and STRINGTIE, as we have shown in both simulated and real data.

For real-data analysis, we used publicly available short-read and long-read sequencing data of the same sample of SARS-CoV-2 infected Vero cells [55]. We performed transcript assembly using the short-read sequencing data and used the long-read data for validation. JUMPER was able to identify transcripts encoding for all known viral proteins except ORF10, which has been shown to have little support of active transcription in previous studies [55, 57]. Moreover, we predicted 9 non-canonical transcripts that are well supported by long-read sequencing data.

Furthermore, we demonstrated that JUMPER enables transcript-level quantitative analysis of viral response to treatment with drugs. More specifically, we analyzed 8 samples of A549 lung alveolar cells infected by SARS-CoV-2, four of which were pre-treated with ruxolitinib

for 1 hour before infection [3]. JUMPER identified one variant of the spike protein, with a 12 amino acid deletion overlapping with the furin cleavage site, that showed statistically significant increase in expression in samples that were pre-treated with ruxolitinib. We also showed the versatility of JUMPER by considering two additional coronaviruses, SARS-CoV-1 and MERS-CoV. For two samples of Calu-3 cells infected by SARS-CoV-1 and three samples of Calu-3 cells infected by MERS-CoV [70], JUMPER reconstructed all the canonical transcripts with distinct TRS-B regions and additionally predicted the presence of non-canonical transcripts encoding for either complete or truncated versions of known viral proteins.

There are several avenues for future work. First, JUMPER currently is only applicable to data obtained using technologies that limit positional bias such as oligo(dT) amplification, which targets the poly(A) tail at the 3' end of messenger RNAs. We plan to extend our current model to account for positional and sequencing biases in the data. Doing so will enable us to assemble transcriptomes from sequencing samples that used SARS-CoV-2-specific primers, which form the majority of currently available data. Second, we currently make the assumption of a fixed read length that is much smaller than the length of viral transcripts. We will relax this assumption in order to support long-read sequencing data that have variable read lengths. Third, we plan to study the effect of mutations (including single-nucleotide variants as well as indels) on the transcriptome. Along the same lines, there is evidence of within-host diversity in COVID-19 patients [15, 16, 91, 92, 93, 94]. It will be interesting to study whether this diversity translates to distinct sets of transcripts and abundances within the same host. Fourth, there are possibly multiple optimal solutions to the DTA problem that present equally likely viral transcripts with different relative abundances in the sample. A useful direction of future work is to explore the space of optimal solutions similar to the work done in [74]. Finally, the approach presented in this paper can extended to the general transcript assembly problem. Although JUMPER can be used for transcript assembly of individual eukaryotic genes (see Appendix E.4), it does not currently support assembly across multiple genes. The extension of the current approach can be facilitated by using the topological ordering of the nodes in a general splice graph that does not have a unique Hamiltonian path, unlike the segment graph considered in the DTA problem. We envision this will facilitate efficient use of combinatorial optimization tools such as integer linear programming to transcript assembly problems.

Chapter 4: Doublet Detection in Single-Cell DNA-sequencing data

4.1 INTRODUCTION

The increased use of single-cell sequencing for cancer research is providing a wealth of new insights regarding intra-tumor heterogeneity, metastasis and the landscape of the tumor microenvironment [95, 96, 97, 98]. In particular, the ongoing improvement in single-cell DNA sequencing (scDNA-seq) assays is rapidly advancing methods for reconstructing the evolutionary history of a tumor [99, 100, 101, 102, 103, 104]. While scDNA-seq is more labor intensive and error-prone than traditional bulk DNA sequencing [105], scDNA-seq permits the observation of mutation co-occurrence patterns within a single cell, yielding both higher fidelity tumor phylogeny reconstructions and more accurate identification of a set of distinct tumor clones or genotypes.

The smaller amount of DNA material within a cell compared to RNA poses additional sequencing challenges than those faced in single-cell RNA sequencing (scRNA-seq) [106]. Medium to high coverage scDNA-seq technology, suitable for detecting single-nucleotide variants, suffers from elevated rates of technical errors due to whole-genome amplification that may impact downstream analyses, including allelic dropout (ADO), copying mistakes in the amplification reaction, unbalanced amplification and doublets. Specifically, when ADO occurs, one or more of the alleles may fail to be amplified during the early stages of the process and thus the allele is said to "drop out" prior to sequencing. While technological advances have decreased the frequency of these errors, one remaining technical challenge is when multiple cells, or *multiplets*, are captured within a droplet and linked to a single barcode making all subsequent reads appearing as if they originated from one cell. To mitigate this effect, practitioners utilize a Poisson distribution to estimate the probability that a droplet contains a specified number of cells. The rate parameter of the Poisson distribution is then determined by a function of the cell solution concentration and droplet volume to obtain the desired probability of multiplets [107]. This results in the majority of droplets containing zero cells and multiplets with more than two cells are rare. However, *doublets*, which are droplets containing two cells, occur frequently and are therefore the focus of this work [108, 109, 110].

Adapting terminology from the scRNA-seq literature [111], we introduce three categories for doublets in scDNA-seq: (i) selflet, (ii) nested and (iii) neotypic (Fig. 4.1a). *Selflets* are comprised of cells with identical genotypes. *Nested* doublets occur when the set of mutations in one cell is a proper subset of the mutations in the other cell. A *neotypic* doublet is a doublet that is not nested or a selflet and implies the existence of a novel genotype not

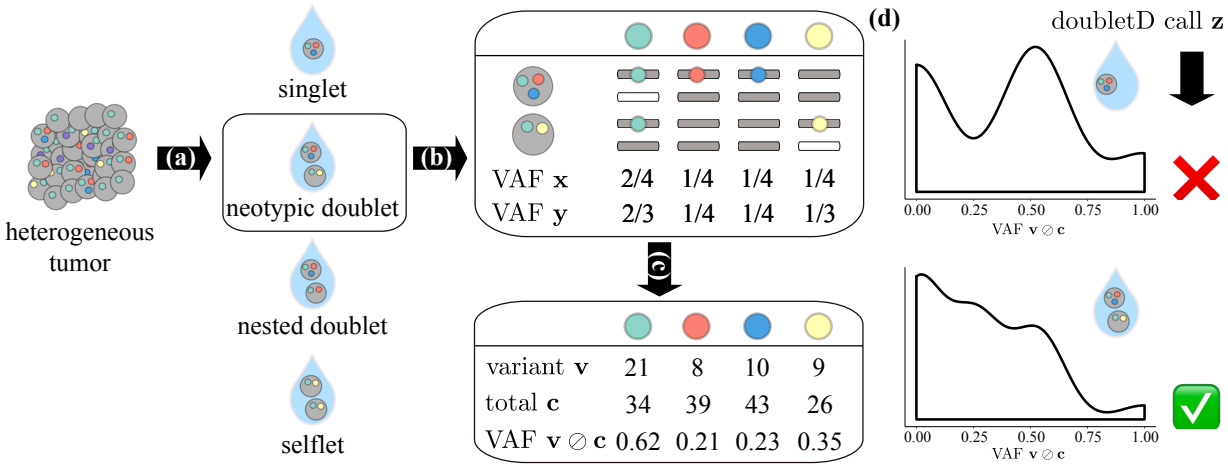


Figure 4.1: **doubletD calls doublets in medium to high coverage scDNA-seq data.** (a) The first step of most single-cell sequencing technologies involves cell capture where the goal is to encapsulate single cells into droplets, known as *singlets*. However, errors in this process (details in Section 4.3) can lead to three kind of doublets – *neotypic* doublets, *nested* doublets and *selflets*. (b) The cells in each isolated droplet i undergo whole-genome amplification and sequencing independently. These processes introduce errors such as allelic dropouts and imbalance in amplification. (c) The resulting aligned reads are used for variant calling yielding alternate $v_{i,j}$ and total $c_{i,j}$ read counts at each locus of interest j. (d) DOUBLETD uses the observed variant allele frequencies $v_{i,j}/c_{i,j}$ as the key signal, while accounting for sequencing and amplification errors to detect doublets in the sample. The symbol $\oslash$ denotes element-wise division.

present in the sample. Neotypic doublets thus distort the signal of mutation co-occurrence patterns and makes it challenging to distinguish the presence of rare clones, that may be resistant to certain treatments, from a neotypic doublet [105]. Although nested doublets and selflets will not impact the analysis of mutation co-occurrence or mutual exclusivity patterns, they may impact the estimation of clonal abundances, which are used to model both the evolutionary trajectory and the fitness landscape of a tumor [97, 112].

While there are downstream analysis methods, such as genotype and/or phylogeny inference methods, that account for the presence of doublets, to the best of our knowledge, there exists no standalone method for doublet detection in scDNA-seq data. There are a number of drawbacks to methods that jointly infer the doublets during any downstream analysis. First, methods like ∞SCITE [113], SCG [103] and SiCloneFit [102] utilize Bayesian inference in the form of Markov chain Monte Carlo (MCMC) or variational inference, which scale poorly with the inclusion of doublets and size of the input [102, 103, 113]. Second, methods, such as ScisTree [114], are able to identify doublets only under the infinite sites model of evolution. Third, most methods require a binarized or discretized experiment by loci matrix

input as opposed to positional variant and reference allele read counts. This results in the loss of useful information for doublet identification. Lastly, as a result of the discrete input and/or utilizing the infinite sites assumption, methods that do identify doublets are at best only able to identify neotypic doublets.

In contrast, there exist a number of standalone methods for detecting doublets in single-cell RNA sequencing data. [111, 115, 116]. See [117] for an excellent overview and benchmarking of scRNA-seq doublet detection methods. Doublets in single-cell RNA sequencing (scRNA-seq) result in the observation of neotypic gene expression profiles, which impacts cell clustering and the identification of cell-state trajectories ([117]). In general, these methods follow a four step process. First, simulated doublets are created by mixing observed gene expression profiles. Second, the observed and simulated data are embedded into a latent space using dimensionality reduction. Third, machine learning methods are used to estimate the probability that a droplet is a doublet. Finally, a threshold scheme is enacted based on knowledge of the experimental doublet rate to classify experiments as either a singlet or doublet. The main variation within these methods is the choice of embedding/dimension reduction and classifier. Additionally, these methods are designed to capture neotypic doublets and struggle to identify embedded doublets, which are often located within clusters of singlets in the embedded space. While it is possible to directly apply scRNA-seq doublet detection methods on DNA variant read counts, such methods do not properly account for the distinct error profile of scDNA-seq data.

As a first step in addressing the need for a fast, standalone method for scDNA-seq doublet detection, we introduce doubletD, which performs doublet detection in medium to high coverage scDNA-seq data. Critically, doubletD does not make any assumptions about the model of evolution, the number of distinct clones or assume a threshold on the minimum clonal abundance in the sample. doubletD operates directly on variant and reference allele counts without the need to discretize the input, thus retaining a critical signal for doublet detection in the form of the variant allele frequency (VAF) (Fig. 4.1c). Specifically, underlying doubletD is the observation that doublets in scDNA-seq data have a characteristic VAF spectrum due to increased number of copies and/or allelic dropout (Fig. 4.1d). Others have noted the presence of some of these characteristics in a *post hoc* analysis of either single-nucleotide variant [118] or copy-number aberration calling [119]. doubletD considers each droplet independently but borrows strength from the entire dataset while using a maximum likelihood approach in order to rapidly classify an experiment as either a doublet or singlet prior to downstream analyses. We demonstrate on both simulated and real datasets that these design choices allow doubletD to be utilized in conjunction with any downstream analysis of choice and therefore obviates the need for more complex downstream methods to

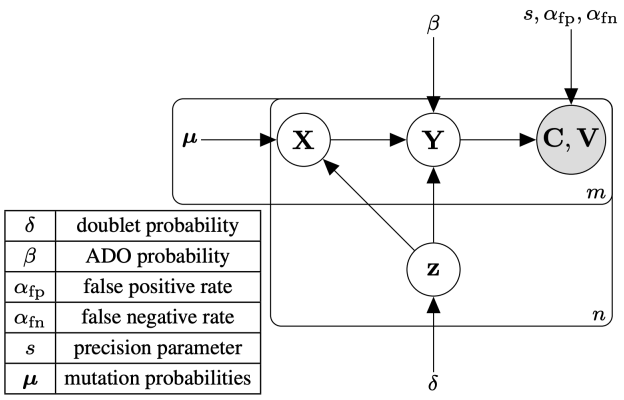


Figure 4.2: **Plate diagram of the doubletD's graphical model.** Observed total and variant read counts $(\mathbf{C}, \mathbf{V})$ of m loci in n droplets are affected by doublet status $\mathbf{z}$, allelic dropout and additional errors during sequencing.

individually account for the presence of doublets within their own models.

4.2 METHODS

4.2.1 Generative Model

Similarly to scRNA-seq, there are two main types of high-throughput cell capture strategies in scDNA-seq: microfluidics and well-based protocols, which, respectively, distribute a cell suspension into either droplets or wells [120, 121]. Here, we use the term 'droplet' independent of the used technology. Consider a scDNA-seq experiment with n droplets and m mutation loci that were identified after read alignment and variant calling. Each mutation locus has two alleles: a reference allele and a variant allele. Thus, we are given $\mathbf{C} = [c_{i,j}] \in \mathbb{N}^{n \times m}$ total read counts and $\mathbf{V} = [v_{i,j}] \in \mathbb{N}^{n \times m}$ variant counts, which are independent across droplets and loci. Read counts $v_{i,j}$ and $c_{i,j}$ of mutation locus j in droplet i are affected by (i) whether droplet i is a doublet (Section 4.2.1.1), (ii) the genotype(s) at locus j in the droplet (Section 4.2.1.2), and errors during sequencing including (iii) allelic dropout (Section 4.2.1.3) and (iv) amplification bias and sequencing errors (Section 4.2.1.4). We make these relationships explicit in a generative model for $\mathbf{C}$ and $\mathbf{V}$ (Fig. 4.2).

4.2.1.1 Doublet Model

In the following we will define random variables $\mathbf{z} \in \{0, 1\}^n$, where z_i indicates whether droplet i is a doublet (i.e. $z_i = 1$) or a singlet (i.e. $z_i = 0$). During the capture step, cells

are released into a nozzle with a constant rate r and there is a fixed time-interval t in which a droplet is formed. The number of cells in a droplet is given by the number of cells that enter the nozzle in the time-interval during which the droplet is formed. Therefore, the prior on the doublet probability is a Poisson distribution with mean $\lambda = rt$. Moreover, only non-empty droplets will yield sequence reads. This combined with the fact that doublets are composed of two cells, we have that $z_i = 1$, i.e. the event of droplet i being a doublet, equals

$$P(z_i = 1) = \frac{\Lambda(2;\lambda)}{\sum_{k=1}^{\infty}\Lambda(k;\lambda)} = \frac{\Lambda(2;\lambda)}{1-\Lambda(0;\lambda)}, \tag{4.1}$$

where $\Lambda(k;\lambda)$ is the probability of $k \in \mathbb{N}$ occurrences (here cells) under a Poisson distribution with mean λ. In practice rt is very small (i.e. $\lambda \ll 1$), and thus the mass of the Poisson distribution $\Lambda(k;\lambda)$ is concentrated around two outcomes $k \in \{1,2\}$. Therefore, z_i can be approximately modeled by a Bernoulli distribution with probability of success $\delta = \Lambda(2;\lambda)/(\Lambda(1;\lambda)+\Lambda(2;\lambda))$ so that

$$P(z_i = 1) = \delta. \tag{4.2}$$

Considering independence between distinct droplets, we get

$$P(\mathbf{z}) = \prod_{i=1}^{n} \delta^{z_i}(1-\delta)^{(1-z_i)}. \tag{4.3}$$

4.2.1.2 Genotype Model

We make the simplifying assumption that each mutation locus has copy number 2 in a single cell — we show robustness of violations to this assumption in Section 4.3.1. Thus the genotype of a locus j in a single cell can be in one of three states: (i) wild type (wt) where both copies have the reference allele, (ii) heterozygous (het) with one variant and one reference copy, and (iii) homozygous (hom) where both copies have the variant allele. Let $\mu_{\text{wt},j}$, $\mu_{\text{het},j}$ and $\mu_{\text{hom},j}$ be the mutation probabilities at locus j of the three types, respectively, such that $\mu_{\text{wt},j} + \mu_{\text{het},j} + \mu_{\text{hom},j} = 1$. Let $x_{i,j}$ indicate the variant allele frequency (VAF) at locus j in droplet i. In case i is a singlet, we have that $x_{i,j} \in \Sigma_{\text{singlet}}$ where $\Sigma_{\text{singlet}} = \{0, 1/2, 1\}$ for any locus j. On the other hand, if i is a doublet, we have that $x_{i,j} \in \Sigma_{\text{doublet}}$ where $\Sigma_{\text{doublet}} = \{0, 1/4, 1/2, 3/4, 1\}$ for any locus j. For a droplet i comprising of a single cell

$(z_i = 0)$, the probability $P(x_{i,j} \mid z_i = 0)$ equals

$$P(x_{i,j} \mid z_i = 0) = \begin{cases} \mu_{\text{wt},j}, & \text{if } x_{i,j} = 0, \\ \mu_{\text{het},j}, & \text{if } x_{i,j} = 1/2, \\ \mu_{\text{hom},j}, & \text{if } x_{i,j} = 1, \\ 0, & \text{otherwise.} \end{cases} \tag{4.4}$$

Following current single-cell literature [122, 123], we assume that a doublet contains two cells with independent genotypes. Therefore, we may define $P(x_{i,j} \mid z_i = 1)$ using probabilities $P(x_{i,j} \mid z_i = 0)$ as

$$\frac{\sum_{g,h \in S(f)} P(x_{i,j} = g \mid z_i = 0) P(x_{i,j} = h \mid z_i = 0)}{\sum_{g,h \in \Sigma_{\text{singlet}} \times \Sigma_{\text{singlet}}} P(x_{i,j} = g \mid z_i = 0) P(x_{i,j} = h \mid z_i = 0)}, \tag{4.5}$$

where $S(f) = \{(g,h) \in \Sigma_{\text{singlet}} \times \Sigma_{\text{singlet}} \mid 2g + 2h = 4f\}$ gives all pairs (g,h) of VAFs in Σ_{singlet} that result in the doublet VAF f. For example, a doublet VAF $f = 1/2$ results from two cells with pairs (g,h) of VAFs in the set $S(1/2) = \{(1/2, 1/2), (1,0), (0,1)\}$.

4.2.1.3 Allelic Dropout Model

We follow the work in [123, 124] to model the shift in variant allele frequency due to allelic dropouts (ADO). In this model, ADO is introduced by deciding for each cell whether a given allele is amplified or not according to a specific probability β known as the ADO rate. Dropout of distinct alleles is assumed to be independent and the ADO rate β is assumed to be constant for all cells and all loci. Although this could be easily extended to account for site-specific ADO as considered in other work [125], here we opt for a global allelic dropout rate to reduce the number of parameters. The VAF $y_{i,j}$ at locus j in droplet i after the dropout event depends on the VAF $x_{i,j}$ and doublet indicator z_i (Fig. 4.2). Specifically, each possible pair $(x_{i,j}, z_i)$, where $x_{i,j} \in \Sigma_{\text{singlet}}$ when $z_i = 0$ and $x_{i,j} \in \Sigma_{\text{doublet}}$ when $z_i = 1$, can yield varying $y_{i,j}$ with probabilities that depend on the number of alleles that are dropped during amplification. Using that each mutation locus has copy number 2 in a single cell and allowing any number of copies to drop out, we have $y_{i,j} \in \Theta_{\text{singlet}}$ where $\Theta_{\text{singlet}} = \{0, 1/2, 1\}$ if droplet i is a singlet. Conversely, if i is a doublet, we have $y_{i,j} \in \Theta_{\text{doublet}}$ where $\Theta_{\text{doublet}} = \{0, 1/4, 1/3, 1/2, 2/3, 3/4, 1\}$. Table A.1 in Appendix A.5 lists all values of $P(y_{i,j} \mid x_{i,j}, z_i)$ for varying $(x_{i,j}, z_i)$ in terms of the ADO rate β.

4.2.1.4 Read Count Model

Beyond ADO, there are two types of additional errors that affect read counts $(c_{i,j}, v_{i,j})$ and lead to an observed VAF $v_{i,j}/c_{i,j}$ that differs from the latent VAF $y_{i,j}$ after ADO: (i) copy errors, which occur early during PCR and lead to a propagation of incorrect nucleotides, and (ii) allelic imbalance, where amplification is biased towards one of the alleles [106]. We model the resulting overdispersion with a beta-binomial as is standard in the field [122, 123, 125]. We use an uninformative prior on total read counts $c_{i,j}$ yielding

$$P(c_{i,j}, v_{i,j} \mid y_{i,j}) = P(v_{i,j} \mid c_{i,j}, y_{i,j})P(c_{i,j}) \propto P(v_{i,j} \mid c_{i,j}, y_{i,j}). \tag{4.6}$$

While copy errors and uneven amplification errors happen simultaneously during the amplification stage, here, following [125], we employ a simpler model that assumes that the copy errors precede the allelic imbalance during amplification. We capture copy errors using a specified false positive rate α_{fp}, which is the probability of generating an alternate allele in the copy when the template has the reference allele, and false negative rate α_{fn}, which is the probability of generating a reference allele in the copy when the template has the alternate allele. Specifically, the probability $p_{i,j}$ of producing a copy with the alternate allele at locus j in experiment i is given by

$$p_{i,j} = y_{i,j}(1 - \alpha_{\text{fn}}) + (1 - y_{i,j})\alpha_{\text{fp}} = \alpha_{\text{fp}} + (1 - \alpha_{\text{fp}} - \alpha_{\text{fn}})y_{i,j}. \tag{4.7}$$

The number $v_{i,j}$ of variant reads resulting after amplification in presence of allelic imbalance is modeled by the following beta-binomial distribution.

$$\pi_{i,j} \sim \text{beta}(p_{i,j}, s), \tag{4.8}$$

$$v_{i,j} \mid c_{i,j}, \pi_{i,j} \sim \text{Binom}(c_{i,j}, \pi_{i,j}), \tag{4.9}$$

where s is the precision parameter that quantifies allelic imbalance error. A low precision s signifies high unevenness in amplification.

4.2.2 Posterior Probability

To determine which droplets are doublets, we are interested in the posterior probability of $\mathbf{z}$ for the given single-cell sequencing data $(\mathbf{C}, \mathbf{V})$, which is defined as

$$P(\mathbf{z} \mid \mathbf{C}, \mathbf{V}) = \frac{P(\mathbf{C}, \mathbf{V} \mid \mathbf{z})P(\mathbf{z})}{P(\mathbf{C}, \mathbf{V})} \propto P(\mathbf{C}, \mathbf{V} \mid \mathbf{z})P(\mathbf{z}). \tag{4.10}$$

In line with current methods [102, 123], we use independence of read counts across mutation loci and droplets and obtain

$$P(\mathbf{C}, \mathbf{V} \mid \mathbf{z}) = \prod_{i=1}^{n} \prod_{j=1}^{m} P(c_{i,j}, v_{i,j} \mid z_i). \tag{4.11}$$

We now express $P(c_{i,j}, v_{i,j} \mid z_i)$ in terms of $P(x_{i,j} \mid z_i)$ (described in Section 4.2.1.2), $P(y_{i,j} \mid x_{i,j}, z_i)$ (described in Section 4.2.1.3) and $P(c_{i,j}, v_{i,j} \mid y_{i,j})$ (described in Section 4.2.1.4). Marginalizing over $x_{i,j}$ and $y_{i,j}$ yields,

$$P(c_{i,j}, v_{i,j} \mid z_i) = \sum_{x_{i,j} \in \Sigma_i} \sum_{y_{i,j} \in \Theta_i} P(c_{i,j}, v_{i,j}, x_{i,j}, y_{i,j} \mid z_i) \tag{4.12}$$

$$= \sum_{x_{i,j} \in \Sigma_i} \sum_{y_{i,j} \in \Theta_i} P(c_{i,j}, v_{i,j} \mid x_{i,j}, y_{i,j}, z_i) P(x_{i,j}, y_{i,j} \mid z_i) \tag{4.13}$$

$$= \sum_{x_{i,j} \in \Sigma_i} \sum_{y_{i,j} \in \Theta_i} P(c_{i,j}, v_{i,j} \mid y_{i,j}) P(y_{i,j} \mid x_{i,j}, z_i) P(x_{i,j} \mid z_i) \tag{4.14}$$

where

$$\Sigma_i = \begin{cases} \Sigma_{\text{singlet}}, & \text{if } z_i = 0, \\ \Sigma_{\text{doublet}}, & \text{otherwise.} \end{cases} \quad \text{and} \quad \Theta_i = \begin{cases} \Theta_{\text{singlet}}, & \text{if } z_i = 0, \\ \Theta_{\text{doublet}}, & \text{otherwise.} \end{cases} \tag{4.15}$$

4.2.3 DoubletD

Our goal is to find $\mathbf{z} \in \{0, 1\}^n$ such that the likelihood function (Equation (4.10)) is maximized. Substituting the doublet prior from Equation (4.3) in Equation (4.10) and taking log, we get

$$\log P(\mathbf{z} \mid \mathbf{C}, \mathbf{V}) = \sum_{i=1}^{n} \sum_{j=1}^{m} \log P(c_{i,j}, v_{i,j} \mid z_i) + \sum_{i=1}^{n} \log P(z_i) + K \tag{4.16}$$

where K is the constant of proportionality. Since z_i is an indicator variable (i.e. $z_i \in \{0, 1\}$), we linearize the above equation in terms of $\mathbf{z}$ using,

$$\log P(c_{i,j}, v_{i,j} \mid z_i) = \log P(c_{i,j}, v_{i,j} \mid z_i = 0) + z_i \Omega_{i,j} \tag{4.17}$$

where

$$\Omega_{i,j} = \log\left(\frac{P(c_{i,j}, v_{i,j} \mid z_i = 1)}{P(c_{i,j}, v_{i,j} \mid z_i = 0)}\right) \tag{4.18}$$

and

$$\log P(z_i) = \log P(z_i = 0) + z_i \left(\frac{\log P(z_i = 1)}{\log P(z_i = 0)}\right) \tag{4.19}$$

$$= \log P(z_i = 0) + z_i \log\left(\frac{\delta}{1-\delta}\right). \tag{4.20}$$

where the last equality uses doublet prior model (Equation (4.3)). Note that, since the read counts $(c_{i,j}, v_{i,j})$ are observed, the matrix $\mathbf{\Omega} = [\Omega_{i,j}] \in \mathbb{R}^{n \times m}$ is constant. Ignoring the constant of proportionality K, which is independent of $\mathbf{z}$, and using linearization of $\log P(c_{i,j}, v_{i,j} \mid z_i)$ and $\log P(z_i)$ in Equation (4.16), we get the following linear objective function

$$J(\mathbf{z}) = \Phi + \sum_{i=1}^{n} z_i \left(\sum_{j=1}^{m} \Omega_{i,j} + \log\left(\frac{\delta}{1-\delta}\right)\right) \tag{4.21}$$

where Φ is a constant defined as follows,

$$\Phi = \sum_{i=1}^{n}\sum_{j=1}^{m} \log P(c_{i,j}, v_{i,j} \mid z_i = 0) + \sum_{i=1}^{n} \log P(z_i = 0). \tag{4.22}$$

Since $J(\mathbf{z})$ is linear, we have the following closed form solution maximizing $J(\mathbf{z})$,

$$z_i = \begin{cases} 1, & \text{if } \sum_{j=1}^{m} \Omega_{i,j} + \log\left(\frac{\delta}{1-\delta}\right) > 0, \\ 0, & \text{otherwise.} \end{cases} \tag{4.23}$$

4.2.3.1 Implementation Details

Our resulting method, DOUBLETD, identifies $\mathbf{z} \in \{0,1\}^n$ given total and variant read counts $(\mathbf{C}, \mathbf{V})$ with maximum posterior probability $P(\mathbf{z} \mid \mathbf{C}, \mathbf{V})$. To do so, DOUBLETD requires input mutation probabilities $\mu_{\text{wt}}, \mu_{\text{het}}$ and μ_{hom} at each locus j used in the genotype model (Section 4.2.1.2), and the precision parameter s used in the read count model (Section 4.2.1.4). Appendix A.6 describes a data-driven approach to estimate these param-

eters. Moreover, the doublet prior probability δ can either be taken as input or estimated by maximizing the posterior probability. DOUBLETD is implemented in Python 3, is open source (BSD-3-Clause license), and is available at `https://github.com/elkebir-group/doubletD`.

4.3 RESULTS

We evaluated the performance of doubletD via *in silico* experiments with known ground truth doublets (Section 4.3.1) as well as two real datasets: (i) a two cell line mixture (Section 4.3.2) and (ii) six patients with acute lymphoblasic leukemia [95] (Section 4.3.3).

4.3.1 *In silico* Experiments

We aim to answer the following questions: (i) Is doubletD agnostic to the choice of scDNA-seq assay and experimental design? (ii) How robust is doubletD to the presence of copy number aberrations? (iii) Will the removal of doublets improve downstream analyses? To this end, we simulated scDNA-seq data of 10 genotypes under an evolutionary model that incorporates copy number abberations (CNAs) and SNVs, varying the number of SNVs $m \in \{10, 50, 100\}$, the doublet probability $\delta \in \{0.1, 0.2, 0.4\}$, the mean sequencing coverage $c \in \{10\times, 50\times, 100\times\}$ and ADO probability $\beta \in \{0.0, 0.05, 0.25\}$. Each combination of simulation parameters was replicated with five different random number generator seeds, amounting to a total of 405 experiments. In each experiment, we simulated 500 *in silico droplet*. We benchmarked our method against SCG [103], a genotyping method for scDNA-seq data whose model optionally incorporates doublet detection, which we refer to as SCG:doublet, and Scrublet [111], a standalone doublet detection method designed for scRNA-seq data. We were not able to benchmark against SiCloneFit [102] and ∞SCITE [113], which are tree inference methods that also incorporate doublets, due to their prohibitive runtimes when run in doublet mode. Appendix D.3 further details the simulation design, evolutionary model and method arguments. In particular, for SCG we performed 25 restarts unless specified otherwise, using the restart with the maximum evidence lower bound (ELBO).

Assay and design agnosticism: We focus on simulations with a mean coverage of $c = 50\times$ and simulated doublet probability of $\delta = 0.2$. We refer to Appendix E.7 for other simulation regimes. While all three methods show increasing F_1 scores (the harmonic mean between precision and recall) with increasing number m of mutations, doubletD achieves the highest F_1 score (median: 0.88) compared to SCG:doublet (median: 0.76) and Scrublet

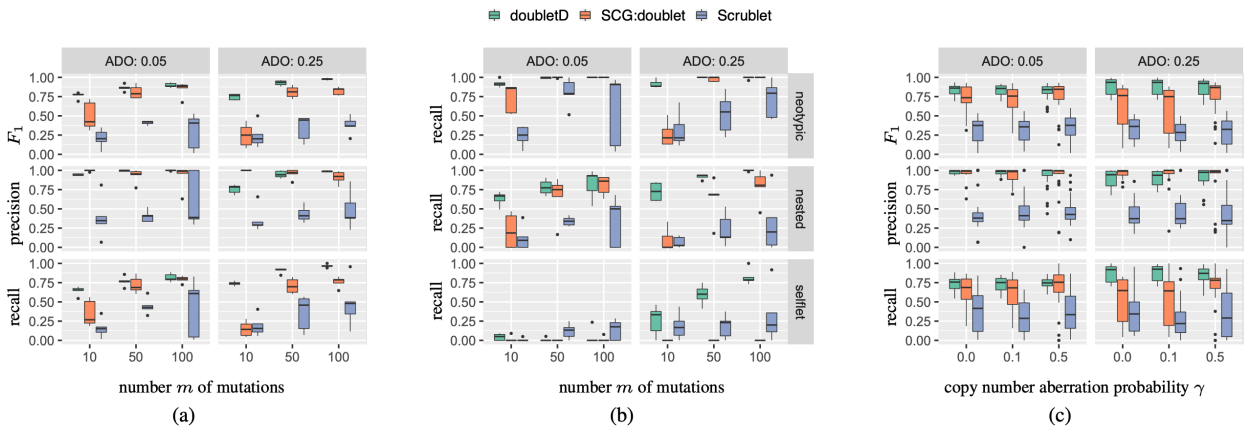


Figure 4.3: **Simulations show that doubletD has high recall and precision in doublet detection, outperforming SCG and Scrublet across various experimental regimes and improving performance in downstream genotyping.** (a) F_1 score, precision and recall of doublet detection for the three competing methods (DOUBLETD, SCG:doublet and SCRUBLET) in simulations with varying ADO rate β and number of mutations m in the absence of CNAs ($\gamma = 0$). (b) Recall of the three kind of doublets, *i.e.* neotypic, nested and selflet. (c) F_1 score, precision and recall by method in the presence of CNAs ($\gamma \in \{0, 0.1, 0.5\}$) and varying ADO rate β. All results are for simulations with doublet probability $\delta = 0.2$, mean read depth $c = 50\times$ and precision parameter $s = 15$.

(median: 0.37) (Fig. 4.3a).Specifically, we find that Scrublet has the worst performance in terms of both recall (median: 0.35) and precision (median: 0.38), demonstrating that doublet detection methods developed for scRNA-seq data *cannot* be directly applied to scDNA-seq data. While both doubletD and SCG:doublet have equivalently high precision (SCG:doublet median: 0.99 vs doubletD median: 0.98), doubletD has superior recall (median: 0.78) among all methods (median recall of 0.67 for SCG:doublet and 0.35 for Scrublet). Strikingly, SCG:doublet performs poorly in the regime of a small number $m = 10$ of mutations, with a median recall and precision of 0.21 and 1.00, compared to 0.70 and 0.87 for doubletD, respectively. Such small number of mutations do occur in practice — e.g. the ALL data analyzed in Section 4.3.3.

Zooming in on doublet type in Fig. 4.3b, we find that all methods have the highest recall for neotypic doublets (median: 1.00 for doubletD, 1.00 for SCG:doublet and 0.50 for Scrublet), and that the recall increases for both nested and neoytpic doublets with increasing number of mutations and increasing ADO. Notably, doubletD has the highest recall for nested doublets (median: 0.85) compared to SCG:doublet (median: 0.57) and Scrublet (median: 0.15). As expected, doubletD and SCG:doublet are unable to detect selflets for ADO rate 0.05 while Scrublet does detect a small proportion of selflets (median: 0.05). However, when ADO rate is 0.25, doubletD has significantly higher recall (median: 0.6) as compared with SCG:doublet

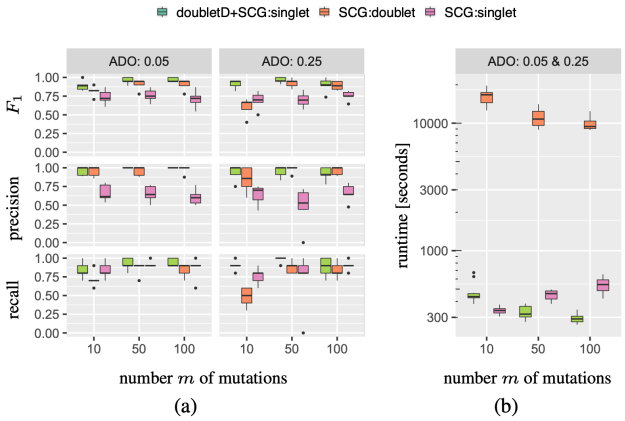


Figure 4.4: **Simulations show that removal of doublets using doubletD improves downstream genotype calling with reduced runtime.** (a) F_1 score, precision and recall of genotypes for doubletD+SCG:singlet, SCG:doublet and SCG:singlet for varying number of mutation m and ADO rate β and without CNAs ($\gamma = 0$). (b) Running time for genotype calling using doubletD+SCG:singlet, SCG:doublet and SCG:singlet for simulations with varying number of mutations m without CNAs ($\gamma = 0$). All results are for simulations with doublet probability $\delta = 0.2$, mean read depth $c = 50\times$ and precision parameter $s = 15$.

(median: 0) and Scrublet (median: 0.2). Note that SCG:doublet is unable to detect selflets due to VAF discretization. Further, both SCG:doublet (IQR: 0.34-0.80) and Scrublet (IQR: 0.13-0.50) show large variance in recall rates as opposed to doubletD (IQR: 0.73-0.92).

Additionally, we find that our method maintains its good performance in simulations when varying coverage and doublet probabilities (Fig. E.18). The lower bound of coverage for the *in silico* experiments was $10\times$. Even at such a low coverage, DOUBLETD maintains its good performance (median precision: 0.83 and median recall 0.78, see Fig. E.18a). It is also important to note that doubletD's improved performance does not come at the expense of running time (Fig. E.20a, median: 14.9 s vs. 11,000.0 s for SCG:doublet and 4.1 s for Scrublet). Finally, DOUBLETD is robust to the choice of user-specified parameters such as the precision s (Appendix E.7.1, Fig. E.21, Fig. E.22, Fig. E.23 and Fig. E.24). In summary, we find that doubletD is robust to many variations in experimental assays and design, outperforming SCG:doublet and Scrublet.

Robustness with respect to CNAs: In order to evaluate the robustness of DOUBLETD to the presence of copy number aberrations (CNAs), we generated simulations with varying probability of CNAs $\gamma \in \{0, 0.1, 0.5\}$, where $\gamma = 0$ represents simulations with no CNAs. More specifically, for each locus that undergoes a copy number aberration (with probability γ), we introduced a loss with probability $\ell \in \{0.1, 0.5\}$ and a gain otherwise. We

ran SCG:doublet with 5 restarts due to increased runtimes compared to the copy-neutral simulations.

Although DOUBLETD does not explicitly account for CNAs, Fig. 4.3c shows that DOUBLETD is robust to varying copy number aberration probability γ, outperforming SCG:doublet and SCRUBLET in most regimes. Specifically, DOUBLETD yields the highest recall (median: 0.79) with good precision (median: 0.98) resulting in the highest F_1 score (median: 0.87) compared to SCG:doublet (median: 0.80) and SCRUBLET (0.36). While SCG:doublet has the same precision as DOUBLETD (median: 0.98), this comes at the cost of lower recall (median: 0.73) compared to DOUBLETD (median: 0.79).

The robustness of DOUBLETD can be explained by the observation that losses (deletions) introduced by CNAs behave similarly to allelic dropouts, which is a key signal used by DOUBLETD to detect doublets. We demonstrate the vulnerability of DOUBLETD to copy number gains on simulations with highest possible copy number aberration probability $\gamma = 1$ and lowest possible loss probability $\ell = 0$ (Fig. E.19). Note that this kind of extreme presentation of CNAs is not observed in practice and that copy number losses including loss of heterozygosity events are common in cancer [100, 104, 126].

In summary, we find that DOUBLETD is robust to the presence of CNAs and outperforms both SCG:doublet and SCRUBLET in doublet detection.

Improving downstream genotype calling: SCG is a genotyping method for scDNA-seq data of tumors that includes doublet detection. It has two modes: in *singlet mode* (SCG:singlet) all droplets are considered singlets, whereas in *doublet mode* (SCG:doublet) genotypes and doublets in the sample are jointly inferred. Here, we assess whether the sequential use of DOUBLETD followed by SCG:singlet (doubletD+SCG:singlet) performs better than SCG:singlet and SCG:doublet. In each of these settings, SCG is run with 25 restarts.

Recall that each of our simulated instances contain 10 genotypes. To assess the performance of the three methods, we compute recall, precision and F_1 score with respect to these ground-truth genotypes, considering a genotype as correctly inferred (i.e. a true positive) if it precisely matches a ground-truth genotype. Thus, if a method infers the exact set of 10 ground-truth genotypes, its recall, precision and F_1 score will be 1. We find that doubletD+SCG:singlet has the highest F_1 score (median: 0.95) compared to SCG:singlet (median: 0.73) and SCG:doublet (median 0.89) across all experimental regimes (Fig. 4.4a). SCG:singlet has good genotype recall (median: 0.9) but reduced precision (median: 0.64) since it misidentifies doublets as cells with distinct genotypes. SCG:doublet, on the other hand, has better precision (median: 1.0) but filters out rare genotypes misidentified as dou-

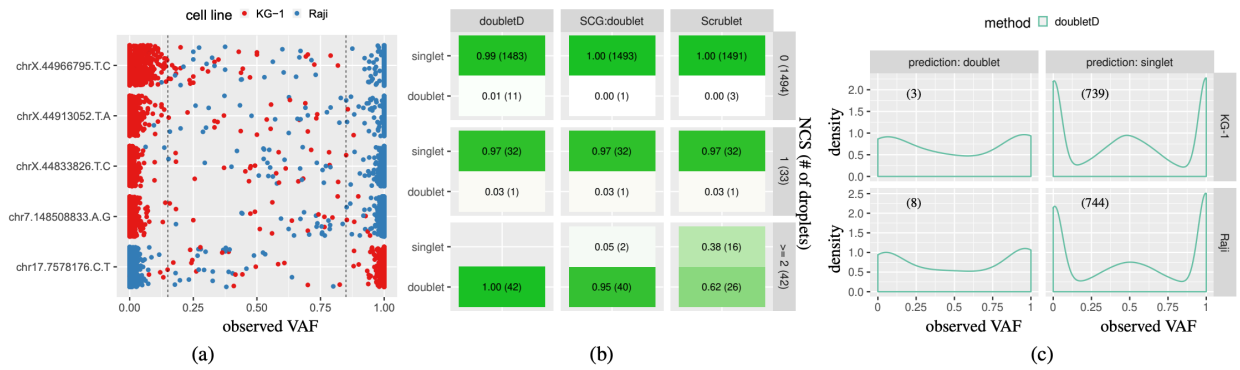


Figure 4.5: **doubletD successfully recalls all 42 orthogonally validated high confidence neotypic doublets and identifies 11 putative selflets in a two cell line mixture dataset.** (a) The VAF for each droplet at each of the five validation loci. Droplets are assigned a neotypic doublet confidence score (NCS), which is the number of validation loci whose VAF was in the range $[0.15, 0.85]$ (dotted lines). (b) The resulting proportion (total) of droplet calls by method (doubletD, SCG:doublet, and Scrublet) by prediction (singlet, doublet) and NCS. (c) The aggregated observed VAF distribution by doubletD prediction and cell line for droplets with NCS $= 0$. The number of droplets in the aggregate are shown in the parenthesis.

blets resulting in reduced recall (median: 0.80). doubletD+SCG:singlet yields the highest recall (median: 0.90) and precision (median: 1.0). In general, SCG:singlet calls more genotypes (median: 14) while SCG:doublet calls fewer genotypes (median: 8.5) compared to the ground truth of 10 genotypes (Fig. E.25). On the other hand, doubletD+SCG:singlet is closer to ground truth with a median of 9.5 distinct genotypes. Furthermore, Fig. 4.4b shows that doubletD+SCG:singlet takes orders of magnitude less time compared to SCG:doublet. While SCG:singlet takes the least time to run, it also yields the lowest F_1 score (Fig. 4.4a).

In summary, we find that the use of DOUBLETD improves genotype calling of SCG while incurring runtimes comparable to SCG in singlet mode. This suggests that doublet removal using DOUBLETD is a useful pre-processing step for downstream analyses of scDNA-seq data of tumors.

4.3.2 Mixture of Two Cell Lines

We validated DOUBLETD on a dataset of $n = 1{,}569$ droplets comprised of a 50%-50% mix of KG-1 and Raji cell lines (with $m = 26$ loci) captured by Mission Bio's Tapestri platform and sequenced by Illumina NextSeq[1]. Appendix E.7.2 details the data preparation, including the exclusion of 23 cells that had a genotype distinct from the two cell lines. KG-1 had

[1] `https://portal.missionbio.com/datasets/KG-1-Raji-50-50-Myeloid`

12 heterozygous (het), 7 wild type and 7 homozygous loci, while Raji had 11 heterozygous, 7 wild type and 8 homozygous loci (Fig. E.26). The mean sequencing coverage c was 110×. Following the procedure outlined in Appendix A.6, we fit beta-binomial precision $s = 10.5$, $\alpha_{\text{fp}} = 0.015$, $\alpha_{\text{fn}} = 0.0073$ and locus-specific mutation probabilities $\boldsymbol{\mu}$ to the observed variant $\mathbf{V}$ and total read counts $\mathbf{C}$. We used the experimental ADO rate ($\beta = 0.06$) previously estimated by [96] on a large patient cohort using Mission Bio's Tapestri platform.

There are two unique characteristics of this dataset that permit identification of neotypic doublets for orthogonal validation: (i) the droplets are easily clustered into two clones by the cell line of origin (Fig. E.26) and (ii) the droplets are comprised of distinct cell lines with distinct evolutionary histories. These characteristics are uncommon in regular datasets where the number of clones and associated genotypes is unknown *a priori* and droplets originate from a single tumor whose clones have a shared evolutionary history. As such, we conclude that doublets will be either neotypic (one cell from each cell line), or selflets (two cells from one cell line).

Using the property that the two cell lines have independent origins and relaxing Mission Bio's standard filtering criteria, we identified an additional set of 5 *validation loci* with distinct wild type/homozygous states among the two cell lines, i.e. each validation locus has state wt (hom) in one cell line and hom (wt) in the other (Fig. 4.5a). Recall that a singlet i will have an observed VAF $v_{i,j}/c_{i,j}$ of approximately 1 if locus j is homozygous and VAF 0 if locus j is wild type. As such, any droplets with observed VAF not close to either 0 or 1 (Fig. 4.5a) indicate that the droplet may be a neotypic doublet comprised of a cell from each cell line. We therefore assign a *neotypic doublet confidence score* (NCS) to each droplet, counting the number of validation loci with VAF between 0.15 and 0.85. This approach yielded 1,494 droplets with NCS = 0, 33 droplets with NCS = 1 and 42 droplets with NCS $\geq$ 2. Note that the NCS is specifically designed to express confidence that a doublet is neotypic but does not capture selfets. Fig. E.26 shows a comparison of the observed variant allele frequency (VAF) of droplets categorized by cell line droplets with a neotypic doublet confidence score (NCS $\geq$ 2).

We ran DOUBLETD, SCG:doublet (with 5 restarts), and SCRUBLET. Since we did not know the true doublet probability δ, we used the maximum likelihood criterion to establish the estimate the doublet probability for DOUBLETD as $\delta = 0.05$ (Fig. E.27a). However, we provided SCG and SCRUBLET with the doublet probability $\delta = 0.09$ as estimated by Mission Bio in similar cell line experiments [127]. For each method and NCS, we calculated the proportion of predicted singlets and doublets (Fig. 4.5b). DOUBLETD identified the most droplets as doublets (54), followed by SCG:doublet (42) and Scrublet 30. DOUBLETD predicted 100% of doublets with NCS $\geq$ 2 whereas SCG:doublet identifies 95.2% of these

droplets with similarly high NCS. Scrublet is the worst performing, identifying only 61.9% of such droplets (Fig.4.5b). In terms of running time, SCG:doublet took 16,259.7 s, doubletD took 24.1 s and Scrublet took 2.4 s.

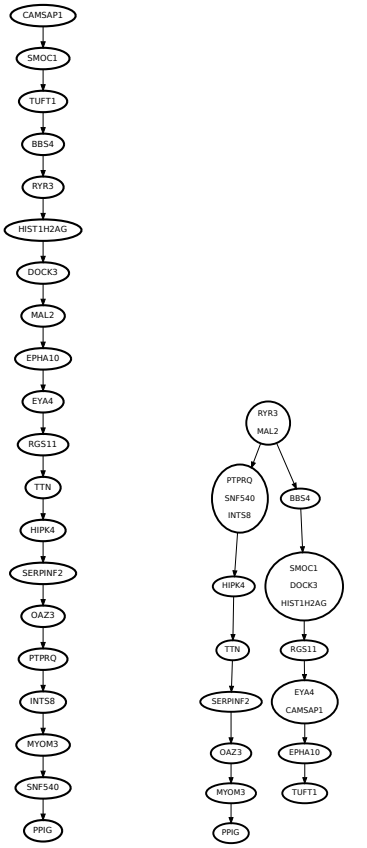
All three methods designated the same droplet at NCS = 1 as a doublet. This suggests that for the remaining 32 droplets at NCS = 1 the observed VAF in $[0.15, 0.85]$ at one of these 5 validation loci is likely attributable to amplification and sequencing error. The one doublet identified by all methods does appear to be neotypic as evidenced by an observed VAF of 0.39 for the validation locus on chromosome 17, which is far from the cut off criterion of 0.15 and is hard to explain by other errors. Furthermore, the VAF distribution across the 26 inference loci for this droplet has a peak at 0.25 and is strikingly different from the distribution of the other Raji droplets with NCS equal to 1 (Fig. E.27b). Lastly, doubletD identifies 11 (proportion: 0.007) putative selflets at NCS = 0, 3 of which are KG-1 and 8 are Raji. SCG calls 1, which was also called by doubletD, and Scrublet calls 3 such droplets with only one called by doubletD. Corroborating this, we note a visual difference in the aggregated VAF distribution across the inference 26 loci between doubletD predicted singlets and doublets with NCS = 0 (Fig. 4.5c). A Venn diagram of the droplets with different NCS score that were predicted as doublets by the three competing methods is shown in Fig. E.28.

In summary, doubletD is able to recall all orthogonally validated high confidence neotypic doublets (with NCS ≥ 2) as well as successfully distinguish the VAF signal of neotypic doublets from sequencing-related error. In addition, we suspect that doubletD is able to recall a small number of selflets even in the presence of low ADO rates ($\beta = 0.05$).

4.3.3 Phylogeny Inference of an Acute Lymphoblastic Leukemia Patient

As discussed in Section 4.1, while nested doublets and selflets do not yield new genotypes, neotypic doublets can be mistaken as an additional clone with a unique genotype [108]. In the extreme case of a phylogeny with only two branches, neotypic doublets that correspond to the two leaves of this tree will include all mutations. Consequently, phylogeny inference under the infinite sites assumption will yield a linear phylogeny. Here, we investigate the impact of doublets on phylogeny inference for a patient (Patient 1) in an acute lymphoblastic leukemia cohort previously suspected to contain doublet droplets [95] — we refer to Table E.5 for doubletD results of the other patients.

[95] sequenced 243 droplets and identified 20 mutations for Patient 1. We analyzed this patient using PhISCS-B [128], which is a phylogeny inference method that seeks to identify a tree constrained by the infinite sites assumption. Since it does not account for doublets, PhISCS-B requires doublets be removed in a pre-processing step. While SCG:doublet was



(a) With doublets (b) Excluding doublets

Figure 4.6: **Doublets lead to incorrect phylogeny inference in acute lymphoblastic leukemia patient 1.** (a) PhISCS-B returns an linear phylogeny with mean log likelihood of $-2806.49/243 = -11.55$ if the 50 doublets detected by doubletD are retained. (b) PhISCS-B returns a branching phylogeny with higher mean likelihood of $-1157.39/193 = -6.00$.

unable to identify any doublets, DOUBLETD identified 50 doublets for this patient. Fig. E.29 corroborates these doublets, showing distinct VAF distributions between singlets and doublets for an orthogonal set of holdout loci. We ran PhISCS-B in single-cell data mode on the complete set of droplets (including doublets) as well as the set of droplets without doublets (details in Appendix E.7.3.3). Fig. 4.6 shows that doublet removal in this patient results in a branching phylogeny with a higher mean likelihood ($-1157.39/193 = -6.00$) compared to a linear phylogeny ($-2806.49/243 = -11.55$) on the complete set of droplets. Furthermore, the branching pattern observed in the inferred phylogeny after doublet removal is in agreement with several other trees published for Patient 1, with identical grouping of the mutations across the two branches [95, 113, 129].

Thus, phylogeny inference is an additional example of a downstream analysis where the

inclusion of doublets may yield incorrect conclusions.

4.4 DISCUSSION

In this work we introduced doubletD, the first standalone method for detecting doublets in single-cell DNA sequencing (scDNA-seq) data with medium to high-coverage ($\geq 5\times$) suitable for single-nucleotide variants. Our method operates directly on variant and total read counts of mutation loci. Underlying our method is the observation that doublets in scDNA-seq data have a characteristic VAF distribution. An additional signal that we exploit is the shift in VAFs due to allelic dropout. This unique approach enables DOUBLETD to capitalize on a major downside of single-cell sequencing in order to identify selflets and nested doublets, that are notoriously hard to detect by current methods. doubletD utilizes a probabilistic model that specifically accounts for allelic imbalance and dropout during whole-genome amplification in scDNA-seq as well as sequencing errors. We introduced a closed-form solution for the inference problem. We demonstrated that our method outperforms current methods for downstream analysis of scDNA-seq data that jointly infer doublets and genotypes [103] as well as standalone approaches for doublet detection in scRNA-seq data [111]. Moreover, we showed that removing doublets using DOUBLETD improves the accuracy and efficiency of downstream analyses such as genotype calling and phylogeny inference.

There are several opportunities for future work. First, while this paper focused on cancer, doubletD can be applied to normal samples as well using heterozygous germline SNPs. Moreover, the same characteristic signal used by our method to detect doublets can be used to detect cells that have undergone whole-genome duplication or are in S-phase with actively replicating DNA. Second, our approach can be extended to support low ($0.1 - 0.5\times$) to ultra-low ($< 0.05\times$) coverage scDNA-seq samples, suitable for copy-number aberrations, by pooling heterozygous germline SNPs located within haplotype blocks. Third, our current formulation assumes that normal cells are diploid. As noted in our simulations, performance slightly decreased in the presence of copy-number aberrations. We plan to extend our probabilistic model to account for copy number. Finally, we envision that DOUBLETD will improve downstream analysis of current and future methods, making doublet detection and removal a standard practice in scDNA-seq analysis.

Chapter 5: Parsimonious Clone Tree Reconciliation

5.1 INTRODUCTION

Cancer results from an evolutionary process where somatic mutations accumulate in the genomes of different cells. This process yields highly heterogeneous tumors composed of different *clones*, each corresponding to a distinct subpopulation of cells with the same complement of somatic mutations [4]. The resulting intra-tumor heterogeneity has been clearly linked to critically important cancer phenotypes, including cancer prognosis and the potential of developing resistance to cancer therapy [5, 6]. Therefore, important downstream applications rely on accurate reconstructions of a tumor's clonal architecture, which in turn requires the identification of the different clones, their proportions and their evolutionary history. However, the presence of different types of somatic mutations in the same clones renders these tasks particularly challenging. In particular, the following two types of somatic mutations are frequent in cancer [130, 131, 132]: (1) single nucleotide variants (SNVs), which are substitutions of individual DNA nucleotides, and (2) copy number alterations (CNAs), which are amplifications and deletions of large genomic regions.

Most cancer sequencing studies use bulk DNA sequencing technology, where one does not directly measure the co-occurrence of different mutations in the same clone because the generated DNA sequencing reads originate from unknown mixtures of millions of different cells in a bulk tumor sample. To identify distinct clones from such data, one thus needs to deconvolve the mixed sequencing data into the different clonal components [133]. Several computational methods have been introduced to perform this task. However, the majority of existing methods only focus on either SNVs [134, 135, 136, 137, 138] or CNAs [139, 140, 141, 142, 143, 144, 145], but rarely on both. Methods that attempt to identify clones in terms of both SNVs and CNAs do not not scale to the numbers of current cancer sequencing datasets (e.g., number of samples, mutations, clones, etc.) and often require heuristics to reduce the size of input instances [146, 147, 148]. As a result, current cancer evolutionary analyses [8, 149] do not apply such proposed methods but rather perform a *post hoc* analysis, manually assigning CNAs to a tree inferred from SNVs. Furthermore, we note that similar issues arise with some single-cell DNA sequencing technologies, since the different features of these technologies only allow the reliable measurement of either SNVs or CNAs [150]. For example, targeted MDA single-cell sequencing technologies are more suited for the idenification of SNVs whereas whole-exome/genome DOP-PCR single-cell technologies are more suited for the identification of CNAs, and both these technologies have been used in parallel on the

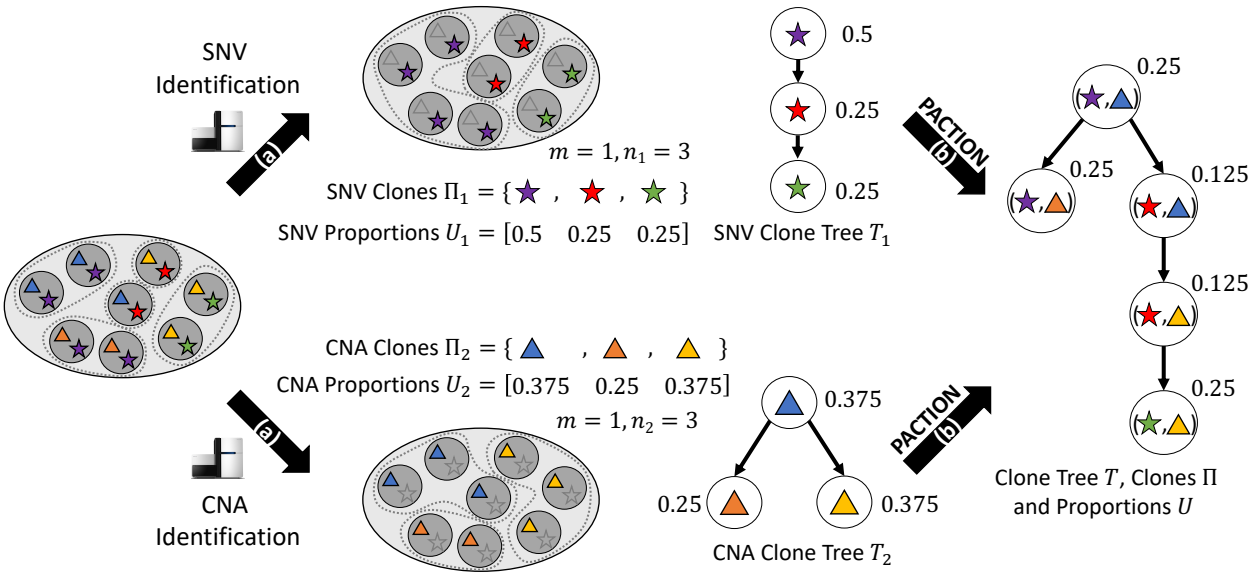


Figure 5.1: **Overview.** A tumor is composed of multiple subpopulations of cells, or clones, with distinct somatic mutations, which can be measured using DNA sequencing. (a) Due to limitations in inference algorithms and/or sequencing technologies, we are limited to characterizing tumor clones in terms of either single-nucleotide variants (SNVs, stars) or copy-number aberrations (CNAs, triangles). That is, we infer clones Π_1, proportions U_1 and a clone tree T_1 for the SNVs. Similarly, we infer clones Π_2, proportions U_2 and a clone tree T_2 for the CNAs. (b) PACTION solves the Parsimonious Clone Tree Reconciliation problem of inferring clones $\Pi \subseteq \Pi_1 \times \Pi_2$, a clone tree T and proportions U that characterize the clones of the tumor in terms of both SNVs and CNAs.

same tumor sample [151].

In this study, we investigate whether tumor clonal compositions can be comprehensively reconstructed by an alternative simpler and automated approach. Leveraging the SNV and CNA clone proportions that can be independently and reliably inferred by existing methods, we introduce the Parsimonious Clone Reconciliation (PCR) and Parsimonious Clone Tree Reconciliation (PCTR) problems to infer clones in terms of both SNVs and CNAs, their proportions and, additionally for the PCTR problem, their evolutionary relationships (Figure 5.1). We prove that the proposed problems are NP-hard and we introduce PACTION (PArsimonious Clone Tree reconciliatION), an algorithm that solves these problems using two mixed integer linear programming formulations. Using simulations, we find that our approach reliably handles errors in input SNV and CNA proportions and scales to practical instance sizes. On 49 samples from prostate cancer patients [8], we find that our approach more comprehensively reconstructs tumor clonal architectures compared to the manual approach adopted in the previous analysis of the same data.

5.2 PROBLEM STATEMENTS

We introduce two reconciliation problem formulations to reconstruct tumor clonal composition from inferred SNV and CNA clone proportions[1]. The first problem aims at inferring tumor clones and related proportions with both SNVs and CNAs given the clone proportions of SNVs and CNAs independently (Section 5.2.1). The second problem additionally considers phylogenetic trees describing the evolution of tumor clones with either different SNVs or CNAs (Section 5.2.2).

5.2.1 Parsimonious Clone Reconciliation

Suppose a tumor is composed of a set Π of $n = |\Pi|$ clones, which are characterised by unique complements of two different features (e.g., SNVs and CNAs). These clones occur in m samples at varying proportions, defined as follows.

Definition 5.1. An $m \times n$ matrix $U = [u_{p,\ell}]$ is a *proportion matrix for n clones* Π provided (i) $u_{p,\ell} \geq 0$ for all samples $p \in [m]$ and clones $\ell \in [n]$, and (ii) $\sum_{\ell=1}^{n} u_{p,\ell} = 1$ for all samples $p \in [m]$.

Due to limitations in inference algorithms and/or sequencing technologies, we only infer clones and their proportions for one feature in isolation. These two features lead to two distinct partitions of all tumor cells: a set $\Pi_1 = [n_1]$ of clones induced by the first feature (e.g., SNVs) and a set $\Pi_2 = [n_2]$ of clones induced by the second feature (e.g., CNAs). We refer to the original clones as Π-clones and the clones induced by the first and the second features as Π_1-clones and Π_2-clones, respectively. The proportions of the Π_1-clones and Π_2-clones are given by the $m \times n_1$ proportion matrix $U_1 = [u^{(1)}_{p,i}]$ and the $m \times n_2$ proportions matrix $U_2 = [u^{(2)}_{p,j}]$, respectively. How are the proportions U_1 for Π_1-clones and the proportions U_2 for Π_2-clones related to the proportions U of the Π-clones?

To answer this question, 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 (Figure 5.2a). As such, we have that the partition Π is a refinement of partitions Π_1 and Π_2. Thus, each Π-clone ℓ corresponds to a unique Π_1-clone i and a unique Π_2-clone j. In other words, we may view the set Π as a binary relation of sets Π_1 and Π_2 of clones composed of pairs $\ell = (i, j)$ of clones, i.e., $\Pi \subseteq \Pi_1 \times \Pi_2$. This relation is captured by the projection functions $\pi_1 : \Pi \to \Pi_1$ and $\pi_2 : \Pi \to \Pi_2$ such that $\pi_1((i, j)) = i$

[1]While reconciliation is used in species phylogenetics, particularly in the context of gene-tree species-tree reconciliation, here we will use this term to indicate the process of obtaining a comprehensive evolutionary tree of tumor clones given input trees that each focus on a distinct genomic feature.

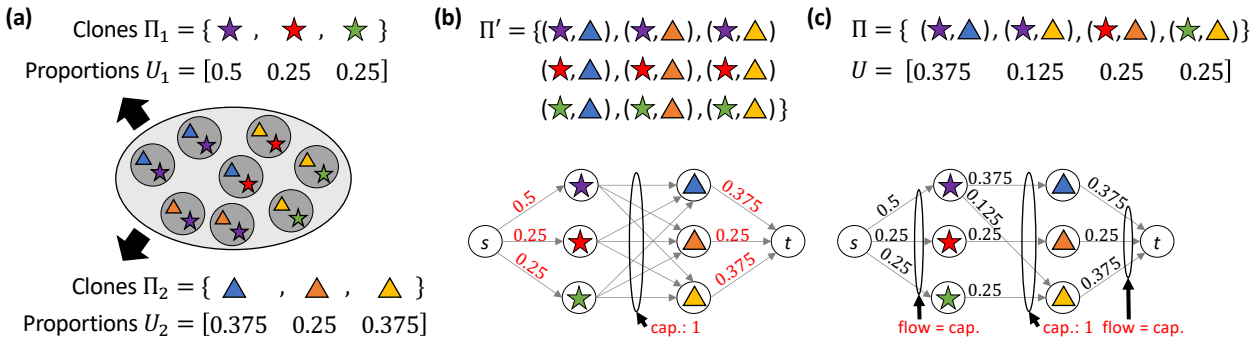


Figure 5.2: **The Parsimonious Clone Reconciliation (PCR) problem.** (a) Given clones Π_1 and Π_2 and corresponding proportions U_1 and U_2, we seek clones $\Pi \subseteq \Pi_1 \times \Pi_2$ and corresponding proportions U consistent with U_1 and U_2. (b) There always exists a consistent proportion matrix U' for the trivial solution $\Pi' = \Pi_1 \times \Pi_2$, which can be identified by solving a maximum flow problem. (c) We seek the solution Π with minimum number $|\Pi|$ of clones. Here, $|\Pi| = 4$, which is smaller than ground truth (see panel (a)). The corresponding matrix U follows from solving the illustrated maximum flow problem. However, incorporating tree constraints, as in the PCTR problem, will lead to ground truth (Figure 5.1).

and $\pi_2((i,j)) = j$ for all $(i,j) \in \Pi$. We relate the proportion matrix U for clones Π to the proportion matrix U_1 for clones Π_1 and the proportion matrix U_2 for clones Π_2 as follows.

Definition 5.2. Given projection functions $\pi_1 : \Pi \to \Pi_1$ and $\pi_2 : \Pi \to \Pi_2$ induced by the set $\Pi \subseteq \Pi_1 \times \Pi_2$ of clones, the proportion matrix $U = [u_{p,\ell}]$ for clones Π is *consistent* with a proportion matrix $U_1 = [u^{(1)}_{p,i}]$ for clones $\Pi_1 = [n_1]$ and proportion matrix $U_2 = [u^{(2)}_{p,j}]$ for clones $\Pi_2 = [n_2]$ provided (i) $u^{(1)}_{p,i} = \sum_{\ell:\pi_1(\ell)=i} u_{p,\ell}$ for all samples $p \in [m]$ and clones $i \in [n_1]$, and (ii) $u^{(2)}_{p,j} = \sum_{\ell:\pi_2(\ell)=j} u_{p,\ell}$ for all samples $p \in [m]$ and clones $j \in [n_2]$.

The above definition formalizes the intuition that clones Π of the tumor are a refinement of the input clones Π_1 and Π_2, and therefore their proportions U must be consistent with the input proportions U_1 and U_2. Our goal is to recover the set $\Pi \subseteq \Pi_1 \times \Pi_2$ of clones and their proportions U from the proportion matrices U_1 and U_2 for clones Π_1 and Π_2, respectively. While there always exist trivial solutions given by the full set $\Pi' = \Pi_1 \times \Pi_2$ of $n = n_1 \cdot n_2$ clones (Figure 5.2b), we seek a solution Π with the smallest number n of clones under the principle of parsimony (Figure 5.2c).

Problem 5.1 (Parsimonious Clone Reconciliation (PCR))**.** Given proportions U_1 for clones $\Pi_1 = [n_1]$ and proportions U_2 for clones $\Pi_2 = [n_2]$, find (i) the smallest set $\Pi \subseteq \Pi_1 \times \Pi_2$ of clones and (ii) proportions U for Π such that U is consistent with U_1 and U_2.

5.2.2 Parsimonious Clone Tree Reconciliation

In practice, proportions U_1 and U_2 are not measured exactly but are affected by potential measurement errors. As such, accurate recovery of the original clones Π and their proportions U requires correcting U_1 and U_2. To accomplish this, we require additional information and constraints. In this work, we propose to use the evolutionary relationships among the clones Π_1 and Π_2 that can be inferred by existing methods in the form of clone trees [134, 135, 152, 153, 154, 155]. Specifically, a rooted tree T is a *clone tree* for clones Π provided the vertex set $V(T)$ equals Π. Moreover, the root vertex $r(T)$ of a clone tree T corresponds to the normal clone while each edge $(u, v) \in E(T)$ represents a mutation event that altered one of the features of clone u and led to the formation of the clone v.

Similarly to the PCR problem, we are given two clone trees, one for each feature in isolation. In the specific example of two features (e.g., SNVs and CNAs), let clone tree T_1 describe the evolution of clones Π_1 (e.g., SNVs) and clone tree T_2 describe the evolution of clones Π_2 (e.g., CNAs). These trees are inferred using standard algorithms in the field [134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145]. Since all clones share a common evolutionary history the original clone tree T is a *refinement* [137, 156] of the clone trees T_1 and T_2, which is defined as follows.

Definition 5.3. Clone tree T for clones Π is a *refinement* of clone trees T_1 for clones Π_1 and clone tree T_2 for clones Π_2 provided

(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 $i = i'$.

Intuitively, the above definition states that when collapsing vertices of T corresponding to identical Π_1-clones one obtains T_1, and, similarly, T_2 is obtained by collapsing vertices of T corresponding to identical Π_2-clones.

Under a principle of parsimony and given clone trees T_1, T_2 with related proportions U_1, U_2, our goal is to find a set $\Pi \subseteq \Pi_1 \times \Pi_2$ of clones, a clone proportion matrix U, and a T_1, T_2-refined clone tree T that require the smallest correction in U_1 and U_2. This motivates the following problem statement.

Problem 5.2 (Parsimonious Clone Tree Reconciliation (PCTR))**.** Given proportions U_1 and tree T_1 for clones $\Pi_1 = [n_1]$ and proportions U_2 and tree T_2 for clones $\Pi_2 = [n_2]$, find (i) the set Π of clones, (ii) clone tree T and (iii) proportions U for Π such that the clone tree T is a refinement of T_1 and T_2 and minimizes the total error $J(U, U_1, U_2)$ such that

$$J(U, U_1, U_2) = \sum_{p=1}^{m} \sum_{i=1}^{n_1} |u_{p,i}^{(1)} - \sum_{\ell:\pi_1(\ell)=i} u_{p,\ell}| + \sum_{p=1}^{m} \sum_{j=1}^{n_2} |u_{p,j}^{(2)} - \sum_{\ell:\pi_2(\ell)=i} u_{p,\ell}|. \tag{5.1}$$

Note that $J(U, U_1, U_2) = 0$ if and only if U is consistent with U_1 and U_2. The clone trees T, T_1 and T_2 do not appear in the objective function $J(U, U_1, U_2)$ and only provides constraints to the optimization problem. Due to these constraints, unlike the previous PCR problem, PCTR does not always admit a trivial solution with $J(U, U_1, U_2) = 0$ (as we further discuss in Section 5.3.2).

5.3 COMBINATORIAL CHARACTERIZATION AND COMPUTATIONAL COMPLEXITY

We investigate the combinatorial structure and computational complexity of the two proposed PCR and PCTR problems in the following two sections, respectively.

5.3.1 Parsimonious Clone Reconciliation

We characterize the combinatorial structure of feasible and optimal solutions (Π, U) for the PCR problem. We first observe that the PCR problem always has a trivial solution. Specifically, given a set Π_1 of $n_1 = |\Pi_1|$ clones and a set Π_2 of $n_2 = |\Pi_2|$ clones and corresponding proportions $U_1 \in [0,1]^{m \times n_1}$ and $U_2 \in [0,1]^{m \times n_2}$, a trivial feasible solution is composed of $n = n_1 n_2$ clones $\Pi = \Pi_1 \times \Pi_2$, which may have many possible corresponding proportions U (Figure 5.2b). For example, proportions $U = [u_{p,(i,j)}]$ can be computed greedily by considering the n clones in any arbitrary order, and assigning each clone $(i,j) \in \Pi$ a proportion of $u_{p,(i,j)} = \min(u_{p,i}^{(1)}, u_{p,j}^{(2)})$ followed by subsequently updating $u_{p,i}^{(1)} := u_{p,i}^{(1)} - u_{p,(i,j)}$ and $u_{p,j}^{(2)} := u_{p,j}^{(2)} - u_{p,(i,j)}$ for each sample $p \in [m]$. Thus, $n = n_1 n_2$ is an upper bound on the number of clones needed. Can we similarly identify a lower bound on n?

To answer this question, let the *support* $S(U)$ of an $m \times n$ proportion matrix U be defined as the number of non-zero entries in the vector $U\mathbf{1}_m$ where $\mathbf{1}_m$ is a $m \times 1$ vector with all entries equal to one. That is, the support $S(U)$ of a proportion matrix U of clones Π signifies the number of clones with non-zero proportion in at least one of the samples $p \in [m]$. Any

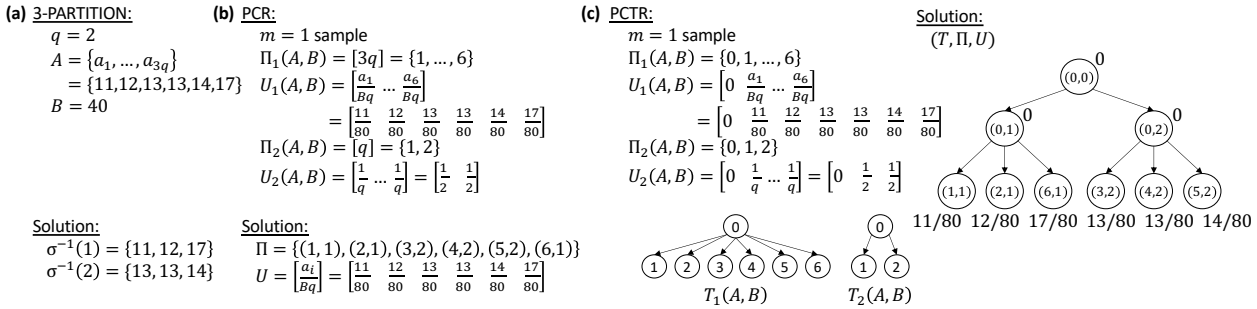


Figure 5.3: **Reduction from 3-PARTITION.** (a) Example instance of 3-PARTITION with a multiset A of 6 elements and target sum $B = 40$. (b) Corresponding PCR instance (Π_1, U_1, Π_2, U_2) and solution (Π, U). (c) Corresponding PCTR instance $(T_1, \Pi_1, U_1, T_2, \Pi_2, U_2)$ and solution (T, Π, U).

such clone must be part of at least one clone $\ell \in \Pi$ in the solution to the PCR problem to ensure consistency of the proportion matrices. This leads to the following observation.

Observation 5.1. Given an instance (Π_1, U_1, Π_2, U_2) of the PCR problem with solution Π we have $n \geq \max(S(U_1), S(U_2))$ where $n = |\Pi|$.

Given any set $\Pi \subseteq \Pi_1 \times \Pi_2$ of clones, deciding whether there exists a proportion matrix U that is consistent with given proportion matrix U_1 for clones Π_1 and U_2 for clones Π_2, and constructing such a matrix is equivalent to solving a maximum flow problem, which takes polynomial time [157]. Figure 5.2 illustrates the construction such that there exists a consistent proportion matrix if and only the value of the flow is 1. Note that for $m > 1$ samples, we need to solve a multi-commodity rather than a single-commodity flow problem. However, the PCR problem, where we simultaneously seek Π and U, is NP-hard and the hardness comes from having to identify the smallest set Π of clones.

Theorem 5.1. The PCR problem is NP-hard even for number $m = 1$ of samples.

This follows by reduction from the 3-PARTITION problem, a known NP-complete problem [158, 159] stated as follows.

Problem 5.3 (3-PARTITION)**.** Given an integer $B \in \mathbb{N}^{>0}$, a multiset $A = \{a_1, \cdots, a_{3q}\}$ of $3q$ positive integers such that $a_i \in (B/4, B/2)$ for all $i \in [3q]$, and $\sum_{i=1}^{3q} a_i = Bq$, does there exist a partition of A into q disjoint subsets such that the sum of the integers in each subset equals B?

Note that since each a_i occurs within the open interval $(B/4, B/2)$ and the elements in each subset of the desired partition sum to B, it holds that each subset must be composed of exactly three elements from the multiset A — hence the name of the problem.

We represent the solution to an instance (A, B) of the 3-PARTITION problem as a function $\sigma : [3q] \to [q]$, which encodes the division of the elements of $A = \{a_1, \ldots, a_{3q}\}$ into q disjoint subsets. The inverse of this function specifies the subset corresponding to each $j \in [q]$ as $\sigma^{-1}(j) = \{i \in [3q] : \ \sigma(i) = j\}$. Note that any solution $\sigma : [3q] \to [q]$ of the 3-PARTITION problem satisfies the following constraint.

$$\sum_{i \in \sigma^{-1}(j)} a_i = B, \quad \forall j \in [q]. \tag{5.2}$$

Figure 5.3a provides an example 3-PARTITION instance and solution.

Given a 3-PARTITION problem instance (A, B), we construct an instance of the PCR problem with number $m = 1$ of samples as follows. The set $\Pi_1(A, B)$ of clones is given by the set $[3q]$. The corresponding proportions are given by the $1 \times 3q$ proportion matrix $U_1(A, B) = [u^{(1)}_{1,i}]$ where $u^{(1)}_{1,i} = a_i/Bq$ for all $i \in [3q]$. Clearly, $U_1(A, B) = [u^{(1)}_{1,i}]$ is a proportion matrix for $\Pi_1(A, B)$ as, by construction, we have that $\sum_{i=1}^{3q} u^{(1)}_{1,i} = 1$ and $u^{(1)}_{1,i} \geq 0$ for all $i \in [3q]$. The second set $\Pi_2(A, B)$ of clones is given by $[q]$. The corresponding proportions are given by the $1 \times q$ proportion matrix $U_2(A, B) = [u^{(2)}_{1,j}]$ where $u^{(2)}_{1,j} = 1/q$ for all $j \in [q]$. It is easy to verify that $U_2(A, B)$ is a proportion matrix for $\Pi_2(A, B)$. Clearly, this construction takes polynomial time. Figure 5.3b shows an example. The hardness follows from the following lemma whose proof is in Appendix B.2.

Lemma 5.1. Given proportions $U_1(A, B)$ for clones $\Pi_1(A, B) = [3q]$ and proportions $U_2(A, B)$ for clones $\Pi_2(A, B) = [q]$, there exists a set Π of clones of size $n = |\Pi| \leq 3q$ with proportions U that are consistent with $U_1(A, B)$ and $U_2(A, B)$ if and only if there exists a solution to the 3-PARTITION instance (A, B).

5.3.2 Parsimonious Clone Tree Reconciliation

We now characterize the combinatorial structure of feasible and optimal solutions (Π, U, T) for the PCTR problem. Let T_1 be the first input clone tree for the input set Π_1 of $n_1 = |\Pi_1|$ clones. Similarly, let T_2 be the second input clone tree for the input set Π_2 of $n_2 = |\Pi_2|$ clones. Let T be a solution clone tree that is a refinement of both T_1 and T_2. First, we observe that the clones that label the root vertices $r(T_1)$ and $r(T_2)$ of the two input trees together label the root vertex $r(T)$ of the output tree T, i.e., $r(T) = (r(T_1), r(T_2))$.

Observation 5.2. If clones Π, clone tree T and proportion matrix U form a solution to the PCTR instance $(\Pi_1, T_1, U_1, \Pi_2, T_2, U_2)$, then $(r(T_1), r(T_2)) \in \Pi$ and $r(T) = (r(T_1), r(T_2))$.

Next, from Definition 5.3 it follows that in the output clone tree T it must hold that along each edge there is either a change in corresponding Π_1-clones or Π_2-clones but not both.

Observation 5.3. For each $(i, j) \in V(T) \setminus \{r(T)\}$ it holds that either $((i', j), (i, j)) \in E(T)$ or $((i, j'), (i, j)) \in E(T)$ where $(i', i) \in E(T_1)$ and $(j', j) \in E(T_2)$.

Combining these observations, we get that the number of vertices/clones in T equals $n = n_1 + n_2 - 1$.

Observation 5.4. The number of clones $V(T)$ equals $n = n_1 + n_2 - 1$.

We note that T is a multi-state perfect phylogeny with two characters, i.e. each character state labels at most one edge of T, whose two sets of states correspond to Π_1 and Π_2. Moreover, T_1 and T_2 impose an ordering of two sets of states to which T must adhere – i.e., the two characters are cladistic [160]. The problem of deciding whether there exists an error-free solution of PCTR with $J(U, U_1, U_2) = 0$ is equivalent to a special case of the Cladistic Multi-state Perfect Phylogeny Deconvolution problem [147]. Details and precise definitions of these concepts are omitted due to space constraints. Although the tree constraints alter the solution space of PCTR problem compared to the PCR problem (see Figure 5.1 and Figure 5.2c), PCTR remains NP-hard, as we will show in the following.

Theorem 5.2. The PCTR problem is NP-hard even for number $m = 1$ of samples.

For a given instance (A, B) of the 3-PARTITION problem, we construct an instance of the PCTR problem as follows. The first set $\Pi_1(A, B)$ of clones equals $\{0\} \cup [3q]$ with corresponding $1 \times (3q + 1)$ proportion matrix $U_1(A, B) = [u^{(1)}_{1,i}]$ where $u^{(1)}_{1,i} = a_i/(Bq)$ for all $i \in [3q]$, and $u^{(1)}_{1,0} = 0$. The second set $\Pi_2(A, B)$ of clones equals $\{0\} \cup [q]$ with corresponding $1 \times (q + 1)$ proportion matrix $U_2(A, B) = [u^{(2)}_{1,j}]$ where $u^{(2)}_{1,j} = 1/q$ for all $j \in [q]$, and $u^{(2)}_{1,0} = 0$. The clone tree $T_1(A, B)$ is a star phylogeny rooted at Π_1-clone $i = 0$ with outgoing edges to each of the remaining Π_1-clones. Similarly, clone tree $T_2(A, B)$ is also a *star* phylogeny rooted at Π_2-clone $j = 0$ with outgoing edges to each of the remaining Π_2-clones. It is easy to verify that $U_1(A, B)$ and $U_2(A, B)$ are proportion matrices for $\Pi_1(A, B)$ and $\Pi_2(A, B)$, respectively. Clearly, this construction takes polynomial time. Figure 5.3c shows an example. The hardness follows from the following lemma whose proof is in Appendix B.2.

Lemma 5.2. Given proportions $U_1(A, B)$ and clone tree T_1 for clones $\Pi_1(A, B) = \{0\} \cup [3q]$ and proportions $U_2(A, B)$ and clone tree T_2 for clones $\Pi_2(A, B) = \{0\} \cup [q]$, there exists a set Π of clones of size $n = |\Pi| = 4q + 1$, clone tree T and proportion matrix U such that T is a refinement of T_1 and T_2 and $J(U, U_1, U_2) = 0$ if and only if there exists a solution of the 3-PARTITION instance (A, B).

5.4 METHODS

We introduce two mixed integer linear programming (MILP) formulations to solve the PCR (Section 5.4.1) and the PCTR problems (Section 5.4.2). We implement these two formulations within the algorithm PACTION (PArsimonious Clone Tree reconciliatION), which uses the MILP-solver Gurobi version 9.1. PACTION is available at `https://github.com/elkebir-group/paction`.

5.4.1 Parsimonious Clone Reconciliation

To solve the PCR problem, we introduce an MILP formulation composed of $\mathcal{O}(n_1 n_2 m)$ variables (including $O(n_1 n_2)$ binary variables) and $\mathcal{O}(n_1 n_2 m)$ constraints. We introduce binary variables $x_{i,j} \in \{0,1\}$ for each Π_1-clone $i \in [n_1]$ and Π_2-clone $j \in [n_2]$ that indicate if clone (i,j) belongs to Π. As such, the corresponding proportion of clone (i,j) in sample $p \in [m]$ is denoted by the continuous variable $u_{p,i,j} \in [0,1]$. In the following we define the constraints on these variables by first describing the constraints for consistency and next those for encoding the objective function.

Consistency constraints. This first set of constraints ensure that proportion matrix U is consistent with proportion matrices U_1 and U_2. We begin by forcing $u_{p,i,j}$ to 0 if (i,j) is not a clone in the solution Π.

$$u_{p,i,j} \leq x_{i,j} \qquad \forall p \in [m], i \in [n_1], j \in [n_2]. \tag{5.3}$$

These above constraints allow us to model consistency of the solution U with input proportions $U_1 = [u^{(1)}_{p,i}]$ and $U_2 = [u^{(2)}_{p,j}]$ as follows.

$$\sum_{j=1}^{n_2} u_{p,i,j} = u^{(1)}_{p,i} \qquad \forall p \in [m], i \in [n_1], \tag{5.4}$$

$$\sum_{i=1}^{n_1} u_{p,i,j} = u^{(2)}_{p,j} \qquad \forall p \in [m], j \in [n_2]. \tag{5.5}$$

Note that these two sets of constraints imply that $\sum_{i=1}^{n_1}\sum_{j=1}^{n_2} u_{p,i,j} = 1$ for all $p \in [m]$.

Objective function. We minimize the total number of clones in the set Π by minimizing the following objective function.

$$\min \sum_{i=1}^{n_1}\sum_{j=1}^{n_2} x_{i,j}. \tag{5.6}$$

5.4.2 Parsimonious Clone Tree Reconciliation

To solve the PCTR problem, we introduce an MILP formulation composed of $\mathcal{O}(n_1 n_2 m)$ variables (including $O(n_1 n_2)$ binary variables) and $\mathcal{O}(n_1 n_2 m)$ constraints. Similarly to the PCR MILP, we introduce binary variables $x_{i,j} \in \{0,1\}$ for $i \in [n_1]$ and $j \in [n_2]$ that indicate if clone (i,j) belongs to Π. As such, the corresponding proportion of clone (i,j) in sample $p \in [m]$ is denoted by the continuous variable $u_{p,i,j} \in [0,1]$. We introduce constraints to model the error $J(U, U_1, U_2)$ used in the objective function, as well constraints to enforce that U is a proportion matrix, and finally constraints to enforce that T is a refinement of T_1 and T_2.

Correction constraints. Unlike the PCR problem, the proportion matrix U need not be consistent with proportion matrices U_1 and U_2. We introduce continuous variables $c^{(1)}_{p,i} \in [0,1]$ for $p \in [m], i \in [n_1]$ and $c^{(2)}_{p,j} \in [0,1]$ for $p \in [m], j \in [n_2]$ to model the entry-wise absolute differences, i.e., $c^{(1)}_{p,i} = |\sum_{j=1}^{n_2} u_{p,i,j} - u^{(1)}_{p,i}|$ and $c^{(2)}_{p,j} = |\sum_{j=1}^{n_2} u_{p,i,j} - u^{(2)}_{p,j}|$. We do so with the following constraints.

$$c^{(1)}_{p,i} \geq \sum_{j=1}^{n_2} u_{p,i,j} - u^{(1)}_{p,i} \qquad \forall p \in [m], i \in [n_1], \tag{5.7}$$

$$c^{(1)}_{p,i} \geq u^{(1)}_{p,i} - \sum_{j=1}^{n_2} u_{p,i,j} \qquad \forall p \in [m], i \in [n_1], \tag{5.8}$$

$$c^{(2)}_{p,j} \geq \sum_{i=1}^{n_1} u_{p,i,j} - u^{(2)}_{p,j} \qquad \forall p \in [m], j \in [n_2], \tag{5.9}$$

$$c^{(2)}_{p,j} \geq u^{(2)}_{p,j} - \sum_{i=1}^{n_1} u_{p,i,j} \qquad \forall p \in [m], j \in [n_2]. \tag{5.10}$$

Proportion matrix constraints. To model that our output matrix U is a proportion matrix, we begin by ensuring that $u_{p,i,j} = 0$ with $x_{i,j} = 0$, i.e., the proportion of clone (i,j) is zero when it is not part of the solution Π with the following constraints.

$$u_{p.i,j} \leq x_{i,j} \qquad \forall p \in [m], i \in [n_1], j \in [n_2]. \tag{5.11}$$

Next, we ensure that matrix U is a valid proportion matrix by enforcing that the proportions of the clones in each sample sum to 1.

$$\sum_{i=1}^{n_1} \sum_{j=1}^{n_2} u_{p,i,j} = 1 \qquad \forall p \in [m]. \tag{5.12}$$

Refinement constraints. We introduce constraints that ensure that the clone tree T is a refinement of the clone trees T_1 and T_2. Following condition (iii) in Definition 5.3, we require that for each clone $(i,j) \neq (r(T_1), r(T_2))$ there only two possible parents, i.e., either (i',j) or (i,j') where $(i',i) \in E(T_1)$ and $(j',j) \in E(T_2)$. We model the first case with continuous variables $z^{(1)}_{(i,i'),j} \in [0,1]$ and the second case with continuous variables $z^{(2)}_{i,(j,j')}$. More specifically, we model the products $z^{(1)}_{(i,i'),j} = x_{i,j}x_{i',j}$ and $z^{(2)}_{i,(j,j')} = x_{i,j}x_{i,j'}$ with the following constraints.

$$z^{(1)}_{(i,i'),j} \leq x_{i,j} \qquad \forall (i,i') \in E(T_1), j \in [n_2], \tag{5.13}$$

$$z^{(1)}_{(i,i'),j} \leq x_{i',j} \qquad \forall (i,i') \in E(T_1), j \in [n_2], \tag{5.14}$$

$$z^{(1)}_{(i,i'),j} \geq x_{i,j} + x_{i',j} - 1 \qquad \forall (i,i') \in E(T_1), j \in [n_2]. \tag{5.15}$$

$$z^{(2)}_{i,(j,j')} \leq x_{i,j} \qquad \forall i \in [n_1], (j,j') \in E(T_2), \tag{5.16}$$

$$z^{(2)}_{i,(j,j')} \leq x_{i,j'} \qquad \forall i \in [n_1], (j,j') \in E(T_2), \tag{5.17}$$

$$z^{(2)}_{i,(j,j')} \geq x_{i,j} + x_{i,j'} - 1 \qquad \forall i \in [n_1], (j,j') \in E(T_2). \tag{5.18}$$

We now enforce conditions (i) and (ii) in Definition 5.3 as follows.

$$\sum_{j=1}^{n_2} z^{(1)}_{(i,i'),j} = 1 \qquad \forall (i,i') \in E(T_1), \tag{5.19}$$

$$\sum_{i=1}^{n_1} z^{(2)}_{i,(j,j')} = 1 \qquad \forall (j,j') \in E(T_2). \tag{5.20}$$

Objective function. Our goal is to minimize the difference between projections of proportion matrix U with U_1 and U_2. To that end, we minimize the following objective function

$$\min \sum_{p=1}^{m} \sum_{i=1}^{n_1} c^{(1)}_{p,i} + \sum_{p=1}^{m} \sum_{j=1}^{n_2} c^{(2)}_{p,j}. \tag{5.21}$$

We provide the full MILP for reference in Appendix A.7.

5.5 RESULTS

5.5.1 Simulations

We perform simulations to investigate the performance of PACTION when solving the PCR and PCTR problems under different simulation regimes.

Setup. Given numbers n_1, n_2 of clones, number m of samples and noise parameter $h \in [0, 1]$, we use a three-step procedure to simulate a set Π of $n = n_1 + n_2$ clones whose SNV and CNA evolution is described by a clone tree T and with clone proportions U on m samples. From T and U, we obtain input trees T_1 and T_2 as well as input proportion matrices U_1 and U_2 subject to additional noise h. We detail the three steps in the following.

First, we use an approach based on growing random networks [161] to simulate T: starting from the root vertex (representing the normal clone $(1, 1)$) T's topology is built by iteratively adding descendant vertices, choosing each parent uniformly at random. Specifically, we label each edge with a single event from either the first set $\{2, \ldots, n_1\}$ or second set $\{2, \ldots, n_2\}$ of features. Thus, the overall clones Π are obtained by labeling all vertices with a depth-first traversal. Second, we obtain the clone trees T_1 and T_2 by collapsing vertices of T corresponding to identical Π_1-clones and collapsing vertices of T corresponding to identical Π_2-clones, respectively. Third, the proportions U of the Π-clones in each sample are simulated by using a Dirichlet distribution with all concentration parameters equal to 1, similarly to previous methods [135, 152]. Proportions U_1 and U_2 are thus obtained following the consistency condition (Definition 5.2). Furthermore, we introduce noise in these two proportion matrices by mixing in a second draw from the same Dirichlet distribution using the parameter $h \in [0, 1]$ — a value of $h = 0$ indicates the absence of noise. Details are in Appendix D.4.

We ran PACTION in both PCR and PCTR mode on 360 simulated instances that we obtained by generating 10 instances for each combination of varying parameters. Matching numbers observed in recent cancer genomics studies [8, 141, 149], we varied the numbers $n_1 \in \{3, 5, 8\}$ and $n_2 \in \{3, 5, 8\}$ of clones, the number $m \in \{1, 2, 5\}$ of samples and noise level $h \in \{0, 0.05, 0.1, 0.15\}$. Note that both proportions U_1, U_2 and the simulated trees T_1, T_2 are taken in input in PCTR mode, while only proportions U_1, U_2 are considered in PCR mode.

Results. We measure the performance of PACTION based on recall, which is the fraction of ground truth clones that are predicted by our method, i.e., the *clone recall* equals $|\Pi \cap \Pi^*|/|\Pi^*|$ where Π is the set of clones inferred by PACTION and Π^* are the ground truth clones. As expected, PACTION in PCTR mode leverages additional information from the clone trees T_1 and T_2 and thus resulted in higher recall compared to PCR mode (Figure 5.4a). Interestingly, recall increased with increasing number m of samples, as each additional samples provides additional constraints regarding consistency of the output clone proportions. Breaking down the clone recall by noise level h, we found that performance decreased with increasing noise levels in both PCR mode (Figure 5.4b) as well as PCTR mode (Figure 5.4c). However, we found that the PCTR solver better handles increasing

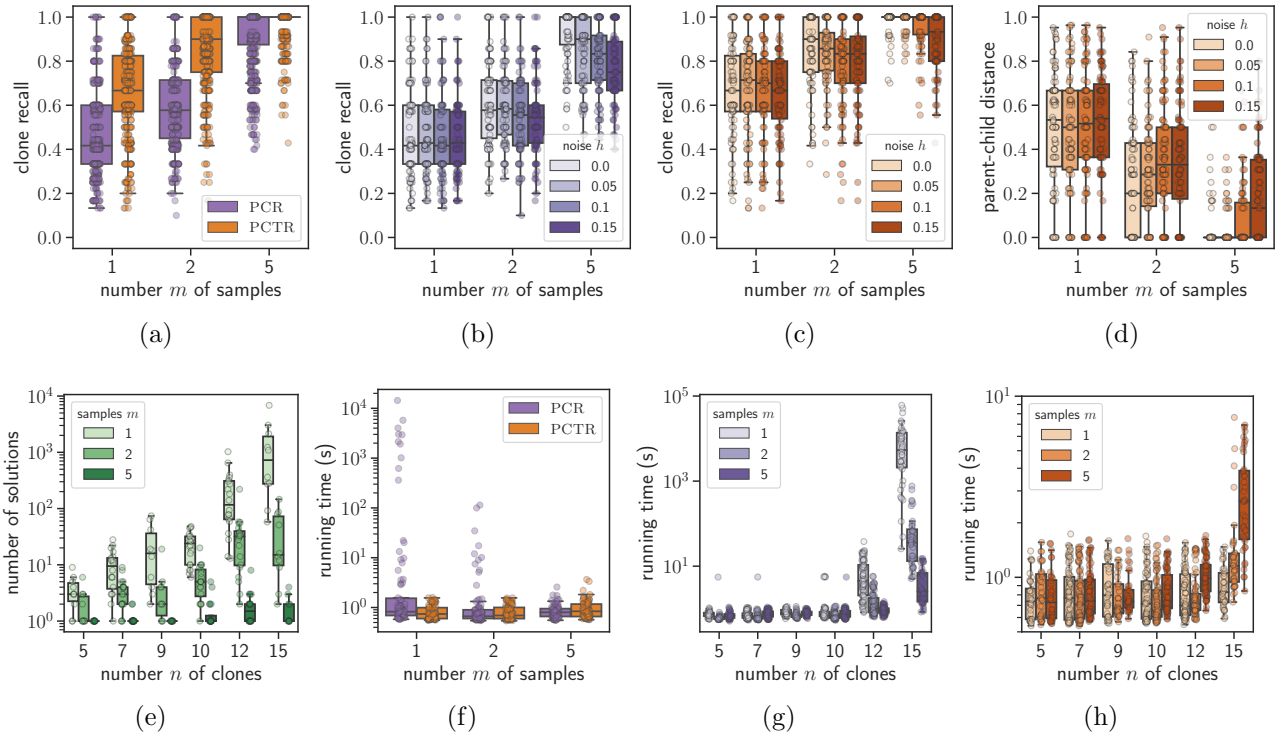


Figure 5.4: **Simulations show that PACTION quickly and accurately reconstructs comprehensive clonal architectures.** (a) Clone recall of PACTION in the PCR and PCTR mode for simulation instances with increasing number m of samples. Clone recall of PACTION in the (b) PCR mode and (c) PCTR mode for different noise levels h and number m of samples. (d) Parent-child distance between the clone tree in the ground truth and the solution of PACTION in the PCTR mode for simulation instances with increasing number m of samples. (e) Number of solutions to the error-free version of the PCTR problem (with additional constraint of $J(U, U_1, U_2) = 0$) by SPRUCE [147] for increasing number n of clones. (f) Running time of PACTION in the PCR and PCTR modes for simulation instances with increasing number m of samples. Running time of PACTION in the (g) PCR mode and the (h) PCTR mode for simulation instances with increasing number n of clones and number m of samples.

noise levels h, with a medial clone recall of 1 for noise level $h = 0$ as well as $h = 0.05$ when number m of samples is 5 (Figure 5.4c and Figure E.30).

Next, we investigated how well PACTION in PCTR mode infers ground truth clone trees T^*. To that end, we computed the parent-child distance [36] between the predicted clone tree T and the clone tree T^* in the ground truth. Specifically, the *parent-child distance* equals the ratio between the size $|E(T) \triangle E(T^*)|$ of the symmetric difference of the edge sets by the size $|E(T) \cup E(T^*)|$ of the union of edge sets. We observed that the clone tree distance is inversely correlated with the clone recall and when the clone recall is 1, the

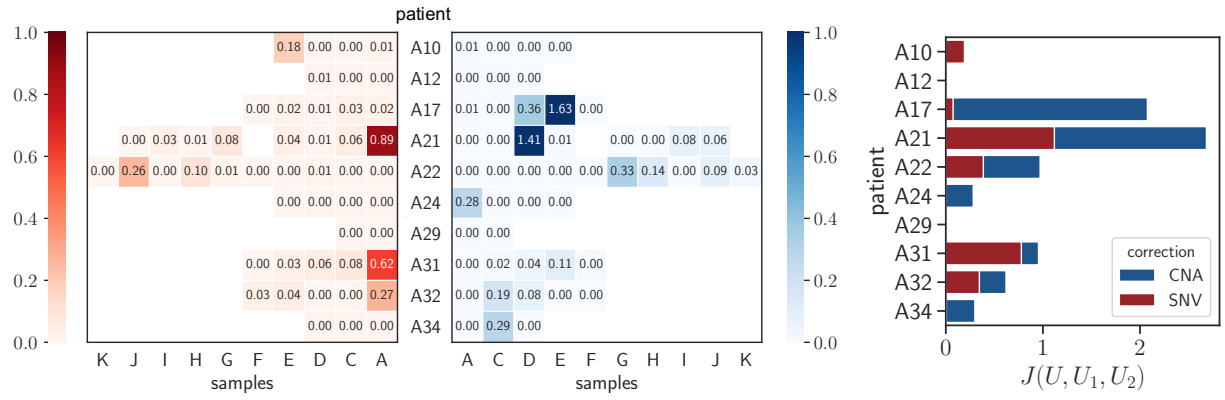
predicted clone tree matches the ground truth perfectly (Figure 5.4d). Indeed, we observed that performance increases with increasing number m of samples, e.g., for $m = 5$ samples the median parent-child distance is 0 for noise levels $h \in \{0, 0.05, 0.1\}$ indicating that in the majority of these instances PACTION perfectly inferred ground truth trees. The reason why performance drops for decreasing number of samples is because the number of solutions increases with decreasing number of samples (Figure 5.4e). We used the correspondence between the PCTR problem (subject to the constraint that $J(U, U_1, U_2) = 0$, i.e., the proportions are error-free) and the perfect phylogeny mixture problem solved by SPRUCE [147] to enumerate all solutions for $h = 0$ instances. For instances with a large number of optimal solutions, the PCTR problem and consequently the MILP lacks additional constraints to disambiguate between solutions, thus sometimes reporting solutions that do not match the ground truth.

Finally, we investigated the running times of PACTION in PCR and PCTR modes. Overall, the running times in PCR mode (median of 0.79 s and mean of 385.52 s) were larger than PCTR mode (median of 0.77 s and mean of 0.95 s), likely due to the tree constraints providing more guidance for the MILP solver. Interestingly, while running time decreased with increasing number m of samples in PCR mode, the opposite is true in PCTR mode. The reason is that in PCTR mode the MILP is often solved in the first iteration prior to branching, where the running time of solving the linear programming relaxation will depend on the size of the formulation, which in turn depends on m. However, in PCR mode, the solver requires branching, and here additional constraints due to more samples will provide stronger bounds that will lead to more pruning and reduction in overall running time.

In summary, our simulations demonstrate that PACTION is able to quickly and accurately reconstruct ground truth clonal architectures under varying noise levels h, especially when the number m is large and when run in PCTR mode.

5.5.2 Metastatic Prostate Cancer

In this study, we analyze whole-genome sequencing data from 49 tumor samples from 10 metastatic prostate cancer patients [8]. In a previous analysis of this data, Gundem *et al.* [8] identified SNV clones and reconstructed the SNV clone tree for each of the 10 patients. To further investigate the role of CNAs on tumor evolution, the authors annotated the SNV clone trees with CNA events in a *post hoc* analysis by manually comparing and matching frequencies of SNVs and CNAs. However, this approach does not allow us to identify tumor clones that are only distinguished by different CNAs and have the same SNVs. Therefore,



(a) (b)

Figure 5.5: **Overview of PACTION results on samples from 10 metastatic prostate cancer patients [8].** (a) The corrections made by PACTION to the SNV and CNA clone proportions in the samples from each of the 10 patients. (b) The total correction made to clone proportions $J(U, U_1, U_2)$ in samples from each patient.

there is no information about CNA-only driven tumor clones nor information about the ordering of the CNA events and the SNV events on the same edge of the tree. Such information is crucial to understand cancer progression [162] and is the subject of numerous studies [163, 164, 165]. Therefore, we investigated whether we can use PACTION to provide a more comprehensive analysis of these tumor clonal compositions by jointly considering SNVs and CNAs.

We applied PACTIONto previously inferred SNV and CNA clone proportions. First, we used the SNV clone proportions as well as the SNV clone tree T_1 inferred for each patient by Gundem *et al.* [8]. Note that each edge of the SNV tree represents a cluster of SNV mutations. As such, we computed the SNV clone proportions U_1 using the published cancer cell fractions of SNVs (details in Appendix E.8). Second, we used the CNA clones obtained from a previous copy-number analysis [141] of the same patients. Since this previous analysis does not provide CNA clone trees, we enumerated all possible binary trees [166] with the CNA clones as the leaves and independently ran PACTION in PCTR mode with each tree as input. We then selected the CNA clone tree with the smallest correction $J(U, U_1, U_2)$, which for each patient was unique. Overall, we ultimately obtained SNV trees with $n_1 \in \{5, \ldots, 16\}$ clones and CNA trees with $n_2 \in \{4, \ldots, 8\}$ clones across $m \in \{2, \ldots, 10\}$ samples (Table E.6).

In all patients but A29, we found that one cannot reconcile independently-inferred SNV and CNA clone trees without additional corrections to the clone proportions. Importantly, this observation highlights that the clone proportions inferred by existing methods are generally characterized by errors (Figure 5.5a). As previously demonstrated in our simulation

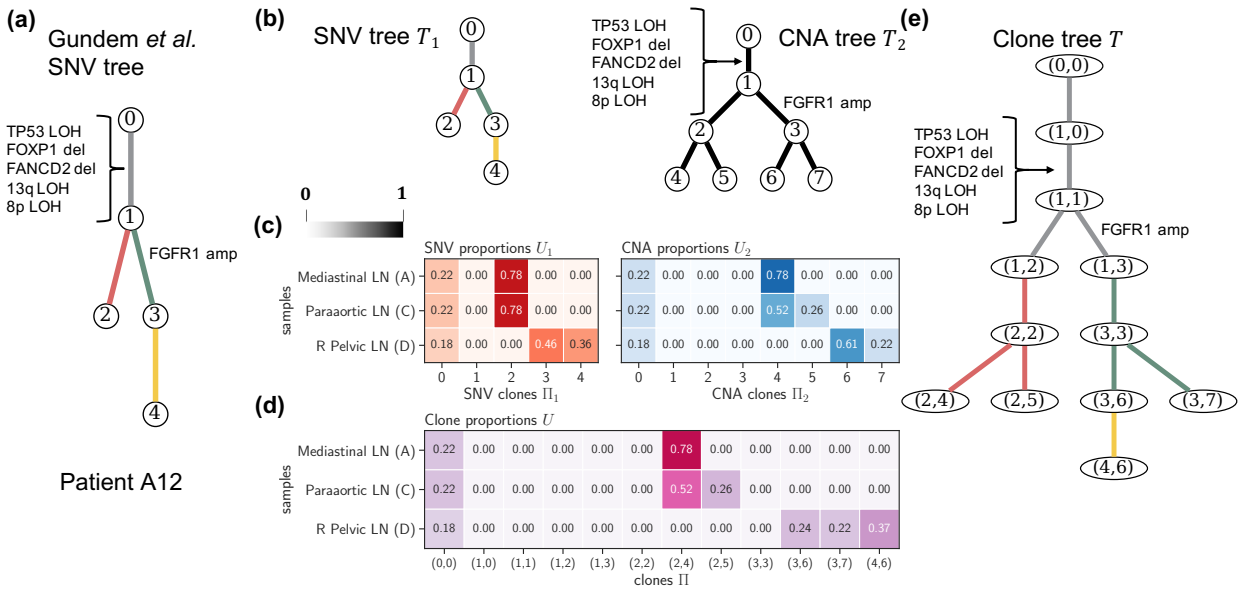


Figure 5.6: **PACTION results for patient A12**. (a) The SNV clone tree reported by Gundem *et al.* [8] where the authors manually annotated edges with CNA events. (b) SNV clone tree T_1 and CNA clone tree T_2 describing the evolution of the SNV clones Π_1 and CNA clones Π_2 in the tumor samples of patient A12, respectively. (c) Proportions U_1 of SNV clones Π_1 and proportions U_2 of CNA clones Π_2 in the four samples of patient A12. (d) Proportions U of tumor clones Π in the four samples of patient A12 inferred by PACTION. (e) Reconciled clone tree T inferred by PACTION. amp: amplification, del: deletion, LOH: loss of heterozygosity.

study, PACTION, however, reliably handles the presence of noise, enabling the inference of the complete clonal composition and tumor evolution with limited corrections for all patients. Specifically, the corrections applied by PACTION were limited to only a few samples per patient, potentially indicating sample-specific errors in previous analysis or samples with higher levels of noise. Importantly, we also observed that corrections were uniformly needed for both SNV and CNA clone proportions (Figure 5.5). This important observation highlights that both features are generally characterized by errors and, therefore, one cannot simply leave one feature fixed and use it to reconcile the other feature, as done previously [8].

Notably, we found that the reconciled clone trees inferred by PACTION reveal additional branching events that were previously missed. As an example, in patient A12, Gundem *et al.* [8] inferred an SNV clone tree with five clones and annotated this tree with five clonal CNA events, including loss-of-heterozygosity (LOH) of gene TP53 and chromosomes 8p and 13q, as well as deletions of genes FOXP1 and FANCD2 (gray edge in Figure 5.6a). The tree also contains a single subclonal CNA event, amplification of gene FGFR1 (green edge in Figure 5.6a). When using PACTION to analyze the previously-inferred SNV and CNA

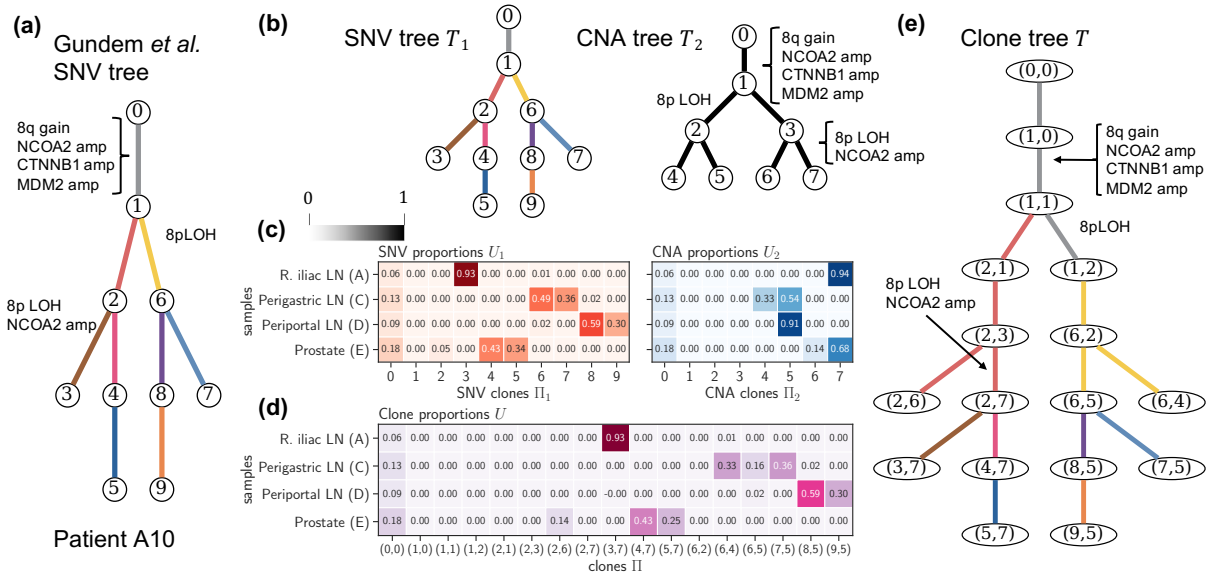


Figure 5.7: **PACTION results for patient A10**. (a) The SNV clone tree reported by Gundem *et al.* [8] where the authors manually annotated edges with CNA events. (b) SNV clone tree T_1 and CNA clone tree T_2 describing the evolution of the SNV clones Π_1 and CNA clones Π_2 in the tumor samples of patient A12, respectively. (c) Proportions U_1 of SNV clones Π_1 and proportions U_2 of CNA clones Π_2 in the four samples of patient A10. (d) Proportions U of tumor clones Π in the four samples of patient A10 inferred by PACTION. (e) Reconciled clone tree T inferred by PACTION. amp: amplification, LOH: loss of heterozygosity.

clone proportions, we reconstructed a reconciled clone tree with higher resolution. In fact, PACTION reconstructed a more refined clone tree with 12 clones while only applying modest corrections to the input clone proportions (Figure 5.5a). Similarly to the published tree, PACTION's inferred clone tree contains a trunk with the same four clonal CNA events. However, PACTION's tree contains additional branching events that are absent in the published SNV tree. Specifically, we observed that two SNV clones in the published tree (i.e., 2 and 3) were split into multiple clones in PACTION's refined tree (i.e., (2, 2), (2, 4), and (2, 5) for SNV clone 2, and (3, 3), (3, 6), and (3, 7) for SNV clone 3). Importantly, a subset of these refined clones are present at large proportions in the sequenced samples (Figure 5.6d), thus showing that PACTION enables a more fine-grained analysis of current sequencing data.

Finally, we found that the more refined clone trees inferred by PACTION also reveal novel insights about the relative temporal ordering of SNVs and CNAs. This phenomenon is particularly interesting in patient A10 (Figure 5.7a), for which PACTION inferred a clone tree with 17 clones and relatively high corrections to the previous SNV clone proportions (Figure 5.7b-d). PACTION's tree recapitulates the same four clonal CNAs identified in

the previous tree, including gain of chromosome 8q and amplifications of genes NCOA2, CTNNB1 and MDM2 (gray edge in Figure 5.7a). Importantly, PACTION's tree also recapitulates subclonal CNA events as in the previous tree but further revealed that these CNA events precede the SNV events placed on the same edges in the published SNV clone tree (Figure 5.7e). More specifically, PACTION revealed that LOH of chromosome 8p and amplification of gene NCOA2 occur on the edge from clone $(2, 3)$ to $(2, 7)$ which precedes the SNV cluster represented by the edge from clone $(2, 7)$ to $(3, 7)$. Similarly, PATION revealed that LOH of chromosome 8p occurs on the edge from clone $(1, 1)$ to $(1, 2)$ which precedes the SNV cluster represented by the edge from clone $(1, 2)$ to $(6, 2)$.

In summary, we demonstrated on metastatic prostate cancer patients that PACTION is able to resolve the temporal ordering of mutations and reveal branching events that are either unclear or hidden when the SNV tree or the CNA tree are considered in isolation.

5.6 DISCUSSION

In this paper, we introduced PACTION, a new algorithm that infers comprehensive tumor clonal compositions by reconciling the clones proportions of both SNVs and CNAs that are inferred by existing methods. Our algorithm can additionally leverage SNV and CNA clone trees reconstructed by existing methods to obtain a refined tumor clone tree and correct potential errors in the input proportions. We formulated two problems, the PCR problem to infer the clones and their proportions, and the PCTR problem to additionally infer tumor clone trees with both SNVs and CNAs. We showed that both problems are NP-hard and can be solved exactly by PACTION using two mixed inter linear programming formulations. We demonstrated the performance of PACTION on simulations, showing that our method accurately reconciles clone trees, reliably handles errors in clone proportions, and scales to practical input sizes. Finally, we applied our method to whole-genome sequencing data from 10 metastatic prostate cancer patients [8], obtaining a higher resolution view of tumor evolution than previously reported.

In addition to the contributions of this study, we foresee four major avenues for future research. First, building upon the established relationship of the error-free PCTR and the cladistic multi-state perfect phylogeny deconvolution problems, we can adapt the existing method SPRUCE [147] to enumerate all possible solution of the PCTR problem in the presence of errors in the input proportions. Second, PACTION can be extended to account for uncertainty in the input clone trees and quantify its effect on the solution space. One way of incorporating the uncertainty in the input clone trees, is to consider a set of possible clone trees for each feature instead of a single input tree, choosing the best tree that leads to the

most parsimonious solution. Moreover, we plan to adapt the PCR and PCTR to incorporate probabilistic models that account for uncertainty in the estimated clone proportions. Third, the PCR and PCTR problems can be generalized to reconcile more than two features. For instance, in addition to SNVs and CNAs, tumor cells may be partitioned into clones based on RNA expression or DNA methylation profiles. Finally, a likelihood-based objective function could be used to incorporate a joint evolutionary model for SNVs and CNAs [104].

Chapter 6: Discussion

In this dissertation, we introduced four novel methods solving problems from infection and cancer genomics. The research presented in this thesis broadly spans three themes:

- Addressing uncertainty in the solution space.
- Employing a comprehensive model that incorporates all the relevant biological processes.
- Developing algorithms that can scale to large datasets.

For each biological question addressed in this thesis, we pose an optimization problem. This is followed by characterization of the solution space and determining the hardness of the problem. Finally, we propose an algorithm which is benchmarked against existing methods on both simulated and real data.

In the context of infection genomics, we introduce two new methods called TiTUS and Jumper. TiTUS reconstructs the transmission history of an outbreak using genetic and epidemiological data collected from infected hosts. Our method accounts for biologically relevant processes such as within-host evolution and multi-strain infections while also accounting for uncertainty in the solution space. Specifically, TiTUS uniformly samples from the solution space of feasible transmission histories for a given timed phylogeny of the pathogen and epidemiological data of the outbreak. The candidate solutions are then summarized the sampled candidate solutions using an efficient consensus-based method in a biologically meaningful way. Jumper, on the other hand, reconstructs viral transcriptome using RNA-sequencing data from infected cells. Our method focuses on viruses in the *Coronaviridae* family, such as SARS-CoV-2, that express genes by a process of discontinuous transcription mediated by the viral RNA-dependent RNA polymerase. Jumper uses an novel combinatorial characterization of the viral transcripts that enables the formulation of an efficient mixed integer linear program. Our results show that Jumper accurately infers the viral transcripts, outperforming existing transcript assembly methods, and enables the study of coronavirus transcriptomes under varying conditions.

In the context of cancer genomics, we develop two novel methods, doubletD and PACTION. doubletD is the first stand-alone doublet detection method for single-cell DNA-sequencing data. Underpinning the doubletD algorithm is the observation that doublets exhibit a characteristic variant allele frequency (VAF) distribution which is distinct from a single-cell. We also observe that this signal is bolstered by allelic dropouts, which are a common source of error in single-cell sequencing. Using a simple probabilistic model with

closed-form maximum likelihood solution, DOUBLETD is able to accurately detect doublets while scaling to large datasets. We show, on multiple real datasets, that doublet identification and removal using DOUBLETD improves downstream analysis such as genotype calling and phylogeny reconstruction. The second method, PACTION, generates a comprehensive tumor phylogeny consisting of both small-scale somatic mutations, specifically single nucleotide variations (SNVs) and large-scale somatic mutations, specifically copy number aberrations. We leverage the power of existing tumor phylogeny reconstruction method, that only focus on either SNVs or CNAs, but not both. Specifically, PACTION reconciles SNV and CNA clone proportions and phylogenies inferred by existing methods for the same cancer tumor. Using simulations, we show that PACTION reliably handles errors in input SNV and CNA proportions and scales to practical instance sizes. On real datasets, PACTION reconstructs tumor clonal architectures that are more reliable and comprehensive than previous studies that employed a manual approach

In the future, the techniques used in the four methods proposed in this thesis, can be extended to other applications as well. For instance, uncertainty in the solution space exists in transcript assembly problems. None of the existing methods completely address the possibility of multiple equally likely reconstructions of the transcriptome. The approach employed in TITUS can also be extended to transcript assembly problems. Another interesting direction of future work is to employ the reconciliation approach of PACTION to multi-omic and epigenomic data. While the current implementation of PACTION focuses on SNVs and CNAs, the underlying principle of integration of multiple modalities and reconciliation of the phylogenetic trees is extremely general.

In conclusion, while ongoing innovations genomic sequencing technologies are opening new ways of seeing the biological world, each technological advancement raises the need for computational methods that can leverage the new modes of data.

Appendix A: Algorithmic details

A.1 NAIVE REJECTION SAMPLING ALGORITHM

Here we describe the naive rejection sampling algorithm introduced in Section 2.5.2.1. Let $h[v, s]$ denote the number of vertex labelings $\ell \in \mathcal{L}_{\text{REL}}$ in the subtree T_v of T rooted at vertex v when $\ell(v) = s$. We define $h[v, s]$ recursively as

$$\begin{cases} 1, & \text{if } v \in L(T),\ \hat{\ell}(v) = s, \\ 0, & \text{if } v \in L(T),\ \hat{\ell}(v) \neq s, \\ 0, & \text{if } v \notin L(T), \tau(v) \notin I(s), \\ \displaystyle\prod_{w \in \delta_T(v)} \sum_{t \in \Gamma_C(s)} h[w, t], & \text{if } v \notin L(T), \tau(v) \in I(s), \end{cases} \tag{A.1}$$

where $I(s) = [\tau_e(s), \tau_r(s)]$ and $\Gamma_C(s) = \{s, \delta_C(s)\}$. Let $\Sigma^* = \{s_1, \ldots, s_k\}$ be the set of possible labels for the root vertex $r(T)$, *i.e.* $\Sigma^* = \{s \in \Sigma \mid \tau(r(T)) \in I(s)\}$. The number of vertex labelings $|\mathcal{L}_{\text{REL}}|$ is given by $\sum_{s' \in \Sigma^*} h[r(T), s']$.

Using the count matrix $h[u, s]$, we introduce a subroutine that takes a vertex v and host s as input, and uniformly samples a vertex labeling ℓ_u of subtree T_u rooted at u subject to the restriction that $\ell_u(u) = s$ (Algorithm A.3). The fraction p_s of the vertex labelings ℓ where $\ell(r(T)) = s$ equals $h[r(T), s] / \sum_{s' \in \Sigma^*} h[r(T), s']$. Thus, to sample *all* vertex labelings uniformly at random, we draw a $s \in \Sigma^*$ according to the categorical probability distribution defined by $(p_1, \ldots, p_k)$. Algorithm A.4 is then used on T with $\ell(r(T)) = s$ to sample minimum transmission host labeling ℓ of T uniformly at random. This takes $O(nm)$ time per sample.

For a given phylogeny and vertex labeling (T, ℓ), it is possible to find the minimum number of transmission events in polynomial time [25]. The *direct transmission constraint* is satisfied by the vertex labeling when the number of transmission events is $m - 1$, where each transmission event corresponds to an edge of the transmission tree. We can therefore draw vertex labelings from $\mathcal{L}_{\text{REL}}$ and only retain the solutions that belong to $\mathcal{L}$ in polynomial time. Since we are uniformly sampling from $\mathcal{L}_{\text{REL}}$, the retained solutions will also be uniformly sampled from $\mathcal{L}$. For the counting problem we estimate the number of vertex labelings in $\mathcal{L}$ by the success rate of the sampling algorithm. Say after K draws of samples from $\mathcal{L}_{\text{REL}}$, we retain K' vertex labelings that belongs to $\mathcal{L}$. In that case the estimate of the size of $\mathcal{L}$,

denote by $\langle|\mathcal{L}|\rangle$, is given by

$$\langle|\mathcal{L}|\rangle = \left(1 - \frac{K'}{K}\right)^{1/K} \tag{A.2}$$

From the law of large numbers, as $K \to \infty$ we have $\langle|\mathcal{L}|\rangle \to |\mathcal{L}|$. We now present the algorithms for naive rejection based sampling.

Algorithm A.1: EnumRelDTI$(T, \hat{\ell}, u, s)$

Output: Set $\mathcal{L}_u$ of vertex labelings ℓ of T_u where $\ell(u) = s$

if $u \in L(T)$ **then**
  Let s be the unique host where $\hat{\ell}(u) = s$
  return $\{\{(u, s)\}\}$
else
  Let $v_1, \ldots, v_k$ be the children of v
  $\mathcal{L}_1, \ldots, \mathcal{L}_k \leftarrow \emptyset, \ldots, \emptyset$
  for $v \in \{v_1, \ldots, v_k\}$ **do**
    for $t \in \Gamma((u, v), s)$ **do**
      $\mathcal{L}_v \leftarrow \mathcal{L}_v \cup$ EnumRelDTI(T, g, v, t)
    end for
  end for
  $\mathcal{L}_u \leftarrow \emptyset$
  for $\ell_1, \ldots, \ell_k \in \mathcal{L}_1 \times \ldots \times \mathcal{L}_k$ **do**
    $\mathcal{L}_u \leftarrow \mathcal{L}_u \cup \{\ell_1 \cup \ldots \cup \ell_k \cup \{(u, s)\}\}$
  end for
  return $\mathcal{L}_u$
end if

Algorithm A.2: EnumRelDTI(T, g)

Output: Set $\mathcal{L}$ of optimal host labelings ℓ of T

Let Σ^* be the set of hosts s where $\tau(r(T)) \in I(s)$
$\mathcal{L} \leftarrow \emptyset$
for $s \in \Sigma^*$ **do**
  $\mathcal{L} \leftarrow \mathcal{L} \cup$ EnumRelDTI$(T, \hat{\ell}, r(T), s)$
end for
return $\mathcal{L}$

Algorithm A.3: SAMPLERELDTI(T, h, u, s)

Output: Random, optimal host labeling ℓ of T_u where $\ell(u) = s$

Let $\delta_T(u) = \{v_1, \ldots, v_k\}$ be the children of u
for $v \in \{v_1, \ldots, v_k\}$ **do**
  $K \leftarrow \sum_{t \in \Gamma_C(s)} h[v, t]$
  for $t \in \Sigma = \{1, \ldots, m\}$ **do**
    if $t \in \Gamma_C(s)$ **then**
      $p(t) \leftarrow h[v, t]/K$
    else
      $p(t) \leftarrow 0$
    end if
  end for
  Draw host $t^* \in \Sigma$ randomly according to $(p_1, \ldots, p_m)$
  $\ell_v \leftarrow$ SAMPLERELDTI(T, g, h, v, t^*)
  for $w \in V(T_v)$ **do**
    $\ell(w) \leftarrow \ell_v(w)$
  end for
end for
$\ell(u) \leftarrow s$
return ℓ

Algorithm A.4: SAMPLERELDTI(T, h)

Output: Random, optimal host labeling ℓ of T

Let Σ^* be the set of hosts s where $\tau(r(T)) \in I(s)$
$K \leftarrow \sum_{s \in \Sigma^*} h[r(T), s]$
for $s \in \Sigma$ **do**
  if $s \in \Sigma^*$ **then**
    $p_s \leftarrow h[r(T), s]/K$
  else
    $p_s \leftarrow 0$
  end if
end for
Draw $s^* \in \Sigma$ according to probabilities $p_1, \ldots, p_m$
return SAMPLERELDTI$(T, h, r(T), s^*)$

A.2 MIXED INTEGER LINEAR PROGRAM FOR DTA PROBLEM

In the following, we introduce variables and constraints to encode the following.

(i) The composition of each transcript T_i as a set $\sigma(T_i)$ of non-overlapping discontinuous edges.

(ii) The abundance c_i and length L_i of each transcript T_i.

(iii) The total abundance $\sum_{i \in X(\mathcal{T}, \sigma_j^{\oplus}, \sigma_j^{\ominus})} c_i$ of transcripts supported by characteristic discontinuous edges $(\sigma_j^{\oplus}, \sigma_j^{\ominus})$.

(iv) A piecewise linear approximation of the log function.

We describe (iii) and (iv) in the following and refer to Section 3.4 for (i) and (ii).

Contribution of transcripts to each pair of characteristic discontinuous edges. The objective function has m terms, one corresponding to each pair $(\sigma_j^{\oplus}, \sigma_j^{\ominus}) \in \mathcal{S}$ of characteristic discontinuous edges (see Eq. (3.4)). Specifically, each term j equals $d_j \log \sum_{i \in X(\mathcal{T}, \sigma_j^{\oplus}, \sigma_j^{\ominus})} c_i$ where d_j is a constant, for all $j \in [m]$. We introduce non-negative continuous variables $\mathbf{q} = \{q_1, \ldots, q_m\}$ such that

$$q_j = \sum_{i \in X(\mathcal{T}, \sigma_j^{\oplus}, \sigma_j^{\ominus})} c_i = \sum_{i=1}^{k} \left(c_i \prod_{e \in \sigma_j^{\oplus}} x_{e,i} \prod_{e' \in \sigma_j^{\ominus}} x_{e',i} \right), \tag{A.3}$$

where the last equality uses the characterization of candidate transcripts of origin for a given read described in Proposition 3.2. We introduce continuous variables $\mathbf{y}_j \in [0,1]^k$ that encode the product $y_{j,i} = c_i \prod_{e \in \sigma_j^{\oplus}} x_{e,i} \prod_{e' \in \sigma_j^{\ominus}} x_{e',i}$. Intuitively, each variable $y_{j,i}$ encodes the contribution of a transcript T_i for the given characteristic discontinuous edge sets $(\sigma_j^{\oplus}, \sigma_j^{\ominus})$. We linearize the product $c_i \prod_{e \in \sigma_j^{\oplus}} x_{e,i} \prod_{e' \in \sigma_j^{\ominus}} x_{e',i}$ as follows.

$$y_{j,i} \leq c_i, \quad \forall i \in [k], j \in [m], \tag{A.4}$$

$$y_{j,i} \leq x_{e,i}, \quad \forall e \in \sigma_i^{\oplus}, i \in [k], j \in [m], \tag{A.5}$$

$$y_{j,i} \leq 1 - x_{e,i}, \quad \forall e \in \sigma_i^{\ominus}, i \in [k], j \in [m], \tag{A.6}$$

$$y_{j,i} \geq c_i + \sum_{e \in \sigma_j^{\oplus}} x_{e,i} + \sum_{e \in \sigma_j^{\ominus}} (1 - x_{e,i}) - |\sigma_j^{\oplus}| - |\sigma_j^{\ominus}|, \quad \forall i \in [k], j \in [m]. \tag{A.7}$$

Hence, we have

$$q_j = \sum_{i=1}^{k} y_{j,i}. \tag{A.8}$$

Objective function. The objective function (see Eq. (3.4)) can be written in terms of continuous variables $\mathbf{q}$ as

$$J(\mathbf{q}) = \sum_{j=1}^{m} d_j \log q_j, \tag{A.9}$$

where d_j is a constant and $\mathbf{q}$ is as in (A.8). We use the lambda method to approximate our objective method using a piecewise linear function [167]. Following the method described in [167], we partition the domain $(0, 1]$ with h breakpoints $b_1 \leq b_2 \leq \ldots \leq b_h$. We introduce continuous variables $\boldsymbol{\lambda}_j \in [0, 1]^h$ with the constraints

$$\sum_{o=1}^{h} \lambda_{j,o} = 1, \quad \forall j \in [m], \tag{A.10}$$

$$\sum_{o=1}^{h} b_o \lambda_{j,o} = q_j, \quad \forall j \in [m]. \tag{A.11}$$

Note that b_o for $o \in [h]$ are constants. Since each of the m terms in the objective function are individually concave and we are maximizing, the adjacency condition of breakpoints does not need to be enforced. For each $j \in [m]$, the log function is then approximated as

$$\log(q_j) \approx \sum_{o=1}^{h} \lambda_{j,o} \log(b_o), \tag{A.12}$$

where $\log(b_o)$ is a constant for each $o \in [h]$. Therefore the objective function we wish to maximize is

$$\sum_{j=1}^{m} d_j \sum_{o=1}^{h} \lambda_{j,o} \log(b_o). \tag{A.13}$$

Note that since we have a log-likelihood objective function, feasibility of the solution requires that $q_j > 0$ for $j \in [m]$. This means that for each characteristic discontinuous edge sets $(\sigma_j^{\oplus}, \sigma_j^{\ominus})$, there must be at least one candidate transcript of origin T_i with non-zero abundance $c_i > 0$. This leads to the solution containing a large number of transcripts and

making the problem intractable while also preventing us from finding parsimonious sets of transcripts that support most but not all of the observed reads in the sample. Finding such parsimonious solutions is often desirable since they provide a reasonable explanation of the observed reads while keeping the problem computationally tractable. In order to allow us to generate solutions that can partially explain the observed reads, we slightly modify our objective function. We introduce a new breakpoint $b_0 = 0$ and associated continuous variables $\lambda_{j,0} \in [0, 1]$ for $j \in [m]$ so that

$$\sum_{o=0}^{h} \lambda_{j,o} = 1, \quad \forall j \in [m], \tag{A.14}$$

$$\sum_{o=0}^{h} b_o \lambda_{j,o} = q_j, \quad \forall j \in [m]. \tag{A.15}$$

The objective function we maximize is

$$\sum_{j=1}^{m} d_j \left(\lambda_{j,0} \log(\delta) + \sum_{o=1}^{h} \lambda_{j,o} \log(b_o) \right), \tag{A.16}$$

where $\delta > 0$ is a small constant. Note that instead of evaluating the log function at b_0, we include $\log(\delta)$ which is well defined since $\delta > 0$. In this study, we choose $\delta = b_1/100 = 1/(2^{h-1} \times 100)$ while h is left as the user's choice with default value of 16.

Moreover, the choice of breakpoints to approximate the objective function can have a significant impact on the accuracy of the MILP solver. As a result, there has been research in efficient methods for choosing optimal breakpoint locations for convex functions, such as recursive descent algorithms [168]. In this work we take a simpler approach, by choosing breakpoints such that their spacing around a given breakpoint is proportional to the local gradient of the objective function. For the log function, this is equivalent to choosing breakpoints such that $b_i = 2^{i-1}/2^{h-1}$. Note that $b_0 = 1/2^{h-1}$ while $b_h = 1$.

Number of variables and constraints. The total number of binary variables $\mathbf{x}$ is $|E^{\curvearrowright}|k$. Note that $\mathbf{q}$ are auxiliary (intermediate) variables that are uniquely determined by $\mathbf{c}, \mathbf{y}$, $\mathbf{z}$ and $\boldsymbol{\lambda}$. Therefore, the total number of required continuous variables (*i.e.* $\mathbf{c}$, $\mathbf{y}$, $\mathbf{z}$ and $\boldsymbol{\lambda}$) is $k + mk + |E^{\curvearrowright}|k + mh$. The number of constraints is $O(k|E|^2 + |E|km)$. We provide the full MILP for reference.

$$\max \sum_{j=1}^{m} d_j \sum_{o=1}^{h} \lambda_{j,o} \log(b_o) \tag{A.17}$$

$$\text{s.t. } x_{e,i} + x_{e',i} \leq 1, \qquad \forall i \in [k] \text{ and } e, e' \in E^{\curvearrowright}, \tag{A.18}$$

$$\text{s.t. } I(e) \cap I(e') \neq \emptyset, \tag{A.19}$$

$$y_{j,i} \leq c_i, \qquad \forall i \in [k], j \in [m], \tag{A.20}$$

$$y_{j,i} \leq x_{e,i}, \qquad \forall e \in \sigma_j^{\oplus}, i \in [k], j \in [m], \tag{A.21}$$

$$y_{j,i} \leq 1 - x_{e,i}, \qquad \forall e \in \sigma_j^{\ominus}, i \in [k], j \in [m], \tag{A.22}$$

$$y_{j,i} \geq c_i + \sum_{e \in \sigma_j^{\oplus}} x_{e,i} + \sum_{e \in \sigma_j^{\ominus}} (1 - x_{e,i}) - |\sigma_j^{\oplus}| - |\sigma_j^{\ominus}|, \qquad \forall i \in [k], j \in [m], \tag{A.23}$$

$$z_{e,i} \leq c_i, \qquad \forall i \in [k], \tag{A.24}$$

$$z_{e,i} \leq x_{e,i}, \qquad \forall e \in E^{\curvearrowright}, i \in [k], \tag{A.25}$$

$$z_{e,i} \geq c_i + x_{e,i} - 1, \qquad \forall e \in E^{\curvearrowright}, i \in [k], \tag{A.26}$$

$$\sum_{i=1}^{k} c_i L - \sum_{i=1}^{k} \sum_{e \in E^{\curvearrowright}} z_{e,i} L(e) = \ell^*, \tag{A.27}$$

$$\sum_{o=1}^{h} \lambda_{j,o} = 1, \qquad \forall j \in [m], \tag{A.28}$$

$$\sum_{o=1}^{h} b_o \lambda_{j,o} = \sum_{i=1}^{k} y_{j,i}, \qquad \forall j \in [m], \tag{A.29}$$

$$x_{e,i} \in \{0, 1\}, \qquad \forall i \in [k], e \in E^{\curvearrowright}, \tag{A.30}$$

$$c_i \geq 0, \qquad \forall i \in [k], \tag{A.31}$$

$$y_{j,i} \geq 0, \qquad \forall j \in [m], i \in [k], \tag{A.32}$$

$$z_{e,i} \geq 0 \qquad \forall e \in E^{\curvearrowright}, i \in [k], \tag{A.33}$$

$$\lambda_{j,o} \geq 0, \qquad \forall j \in [m], o \in [h]. \tag{A.34}$$

A.3 PROGRESSIVE HEURISTIC FOR THE DTA PROBLEM

Here we describe the subproblems that are solved at each iteration of the greedy heuristic. For a given set of transcripts $\mathcal{T}$ and characteristic discontinuous edge sets $\mathcal{S}$, consider the

optimization problem which we denote by P_1,

$$\max_{T',\mathbf{c},c'} \sum_{j=1}^{m} d_j \log \left(\sum_{i \in X(\mathcal{T},\sigma_j^{\oplus},\sigma_j^{\ominus})} c_i + \mathbf{1}(X(\{T'\},\sigma_j^{\oplus},\sigma_j^{\ominus})) \neq \emptyset) c' \right) \tag{A.35}$$

$$\text{s.t. } \pi(T') \text{ is an } \mathbf{s}-\mathbf{t} \text{ path in the segment graph } G \tag{A.36}$$

$$\sum_{i=1}^{|\mathcal{T}|} \overline{c}_i L_i + c' L' = D, \tag{A.37}$$

$$c_i \geq 0 \qquad \forall i \in [|\mathcal{T}|] \tag{A.38}$$

$$c' \geq 0 \qquad . \tag{A.39}$$

and the following optimization problem denoted by P_2,

$$\max_{\mathbf{c}} \sum_{j=1}^{m} d_j \log \sum_{i \in X(\mathcal{T},\sigma_j^{\oplus},\sigma_j^{\ominus})} \overline{c}_i \tag{A.40}$$

$$\sum_{i=1}^{|T|} c_i L_i = D, \tag{A.41}$$

$$c_i \geq 0 \qquad \forall i \in [|\mathcal{T}|]. \tag{A.42}$$

Solution to P_1 We obtain the solution of P_1 by solving the optimization problem given in Eq. (3.4) to (3.7) with additional constraints to fix the values of the variables that encode the presence/absence of discontinuous edges for the transcripts in $\mathcal{T}$. More specifically, for each transcript $T_i \in \mathcal{T}$, we enforce $x_{e,i} = 1$ for each edge $e \in \sigma(T_i)$ and $x_{e,i} = 0$ otherwise. Note that c_i for $T_i \in \mathcal{T}$ are still variables and are solved for in the optimization problem. By doing so, we only solve for the structure of the transcript T' while solving for the abundance of all transcripts.

Solution to P_2 Similar to the approach taken to solve P_1, we fix the values of the variables that encode the presence/absence of discontinuous edges in the transcripts. This results in all the binary variables in the MILP with fixed values rendering the resulting optimization problem a simpler linear program.

Heuristic Algorithm The Algorithm 3.1 is re-written here in form of an itemized list.

1. Initialize $\mathcal{T} = \{\}$, $i = 1$

2. Solve P_1 with $\mathcal{T}$ to get a new transcript T' with abundance c'

3. Generate a new set of transcripts $\mathcal{T} \leftarrow \mathcal{T} \cup \text{EXPAND}(T')$ where $\text{EXPAND}(T') = \{T : \sigma(T) \in 2^{\sigma(T')}\}$.

4. Solve P_2 with $\mathcal{T}$ as input

5. Select i transcripts from $\mathcal{T}$. If $i < k$ go to step (2) else return $(\mathcal{T}, \mathbf{c})$

A.4 FILTERING FALSE POSITIVE DISCONTINUOUS EDGES

In practice, we see spurious discontinuous edges in the resulting segment graph due to sequencing and alignment errors. We filter these edges by requiring a minimum number Λ of spliced reads to support each discontinuous edge in the segment graph. The higher the value of Λ, fewer will be the number of edges and nodes in the resulting segment graph.

It is not trivial to infer the optimal value of Λ to remove all false positive discontinuous edges. Several heuristics are used in existing methods to remove spurious splicing events. Scallop removes an edge e from its splice graph if the coverage of the exons of either end of the edge is more than $2w(e)^2+18$, where $w(e)$ is the number of spliced reads that support the edge e. StringTie on the other hand, terminates its algorithm of assembling transcripts when the coverage of all the paths in the splice graph build from the un-assigned reads drops below a threshold, set by default to 2.5 reads per base-pair. By default, Jumper requires a support of 100 reads for a discontinuous edge to be included in the segment graph.

Another parameter that can be used to filter false-positive splicing events is the number of discontinuous edges allowed in the segment graph. From tests on simulated instances emulating SARS-CoV-2 samples, we found that focusing on the 35 most abundant discontinuous edges is sufficient to get a summary of the transcriptome and highly expressed canonical and non-canonical transcripts in the sample. A higher value can be used to capture more complexity of the transcriptome. By default, we set this parameter to 35.

A.5 ALLELIC DROPOUT MODEL

Table A.1 shows the value of $P(y_{i,j} \mid x_{i,j}, z_i)$ for all possible combinations of post-ADO VAF $y_{i,j}$, pre-ADO VAF $x_{i,j}$ and doublet status z_i.

$x_{i,j}$	z_i \ $y_{i,j}$	0	1/4	1/3	1/2	2/3	3/4	1	NaN
0	0	$1-\beta^2$	0	0	0	0	0	0	β^2
1/2	0	$\beta(1-\beta)$	0	0	$(1-\beta)^2$	0	0	$\beta(1-\beta)$	β^2
1	0	0	0	0	0	0	0	$1-\beta^2$	β^2
0	1	$1-\beta^4$	0	0	0	0	0	0	β^4
1/4	1	$\beta(1-\beta)^3+3\beta^2(1-\beta)^2+3\beta^3(1-\beta)$	$(1-\beta)^4$	$3\beta(1-\beta)^3$	$3\beta^2(1-\beta)^2$	0	0	$\beta^3(1-\beta)$	β^4
1/2	1	$\beta^2(1-\beta)^2+2\beta^3(1-\beta)$	0	$2\beta(1-\beta)^3$	$(1-\beta)^4+4\beta^2(1-\beta)^2$	$2\beta(1-\beta)^3$	0	$\beta^2(1-\beta)^2+2\beta^3(1-\beta)$	β^4
3/4	1	$\beta^3(1-\beta)$	0	0	$3\beta^2(1-\beta)^2$	$3\beta(1-\beta)^3$	$(1-\beta)^4$	$\beta(1-\beta)^3+3\beta^2(1-\beta)^2+3\beta^3(1-\beta)$	β^4
1	1	0	0	0	0	0	0	$1-\beta^4$	β^4

Table A.1: This table shows the value of $P(y_{i,j} \mid x_{i,j}, z_i)$, i.e. the probability of having VAF $y_{i,j}$ at locus j in droplet i after allelic dropout (ADO) given pre-ADO VAF $x_{i,j}$ and doublet status z_i. The last column 'NaN' represents the case when all the alleles are dropped and, as a result, no reads span locus j in droplet i. The values in each row sum to 1.

A.6 PARAMETER ESTIMATION IN DOUBLETD

DOUBLETD requires the user to input mutation probabilities $\mu_{\text{wt}}, \mu_{\text{het}}$ and μ_{hom} at each locus j used in the genotype model (Section 2.1.2), and the precision parameter s used in the read count model (Section 2.1.4). In this section we describe a data-driven approach to estimate these parameters.

Due to the evolutionary pressures on the cells in the sample, the rate of mutations can change significantly across loci. We therefore use the data to get the mutation probabilities $\mu_{\text{wt},j}, \mu_{\text{het},j}$ and $\mu_{\text{hom},j}$ for each locus j which serve as the input parameters for the genotype model (Section 2.1.2). The observed VAFs $v_{i,j}/c_{i,j}$ for each droplet i at locus j are mapped to the closest value in Σ_{singlet}. For $x \in \Sigma_{\text{singlet}}$, let $d_j(x)$ be the number of cells in which the VAF at site j was mapped to x. We estimate the mutation rate $\mu_{\text{wt},j}$ as follows,

$$\mu_{\text{wt},j} = \frac{d_j(0)}{d_j(0) + d_j(1/2) + d_j(1)}. \tag{A.43}$$

Similarly, the mutation rates $\mu_{\text{het},j}$ and $\mu_{\text{hom},j}$ are estimating as follows.

$$\mu_{\text{het},j} = \frac{d_j(1/2)}{d_j(0) + d_j(1/2) + d_j(1)}, \tag{A.44}$$

$$\mu_{\text{hom},j} = \frac{d_j(1)}{d_j(0) + d_j(1/2) + d_j(1)}. \tag{A.45}$$

Fig. E.24a shows that this method gives reliable estimates of the mutation probabilities in simulations.

To estimate the precision parameter s for the beta-binomial distribution, the observed VAFs $v_{i,j}/c_{i,j}$ for each droplet i at locus j are mapped to a value in Σ_{singlet}. For each $x \in \Sigma_{\text{singlet}}$, let ω_x be the the set of observed VAFs mapped to x. We first fit shape parameters $\hat{\alpha}_x$, $\hat{\beta}_x$ for each $x \in \Sigma_{\text{singlet}}$ set ω_x, utilizing method of moments estimation [169] and obtain observed precision $\hat{s}_x = \hat{\alpha}_x + \hat{\beta}_x$. Since our method utilizes a global precision parameter for all droplets and loci, we set the precision parameter to the median of the set $\{\hat{s}_x \mid x \in \Sigma_{\text{singlet}}\}$. The estimation of this parameter can be supplemented from non-variant loci or SNP positions in addition to the observed VAFs. Fig. E.24b shows that we recover reliable estimates of the precision parameter s in simulations. See [170] for an alternative estimation procedure and MDA specific shape parameters that scale linearly with sequencing coverage.

A.7 MILP FORMULATION FOR THE PCTR PROBLEM

$$\min \sum_{p=1}^{m}\sum_{i=1}^{n_1} c_{p,i}^{(1)} + \sum_{p=1}^{m}\sum_{j=1}^{n_2} c_{p,j}^{(2)} \tag{A.46}$$

$$\text{s.t. } c_{p,i}^{(1)} \geq \sum_{j=1}^{n_2} u_{p,i,j} - u_{p,i}^{(1)} \qquad \forall p \in [m], i \in [n_1], \tag{A.47}$$

$$c_{p,i}^{(1)} \geq u_{p,i}^{(1)} - \sum_{j=1}^{n_2} u_{p,i,j} \qquad \forall p \in [m], i \in [n_1], \tag{A.48}$$

$$c_{p,j}^{(2)} \geq \sum_{i=1}^{n_1} u_{p,i,j} - u_{p,j}^{(2)} \qquad \forall p \in [m], j \in [n_2], \tag{A.49}$$

$$c_{p,j}^{(2)} \geq u_{p,j}^{(2)} - \sum_{i=1}^{n_1} u_{p,i,j} \qquad \forall p \in [m], j \in [n_2], \tag{A.50}$$

$$u_{p.i,j} \leq x_{i,j} \qquad \forall p \in [m], i \in [n_1], j \in [n_2], \tag{A.51}$$

$$\sum_{i=1}^{n_1}\sum_{j=1}^{n_2} u_{p,i,j} = 1 \qquad \forall p \in [m], \tag{A.52}$$

$$z_{(i,i'),j}^{(1)} \leq x_{i,j} \qquad \forall (i,i') \in E(T_1), j \in [n_2], \tag{A.53}$$

$$z_{(i,i'),j}^{(1)} \leq x_{i',j} \qquad \forall (i,i') \in E(T_1), j \in [n_2], \tag{A.54}$$

$$z_{(i,i'),j}^{(1)} \geq x_{i,j} + x_{i',j} - 1 \qquad \forall (i,i') \in E(T_1), j \in [n_2]. \tag{A.55}$$

$$z_{i,(j,j')}^{(2)} \leq x_{i,j} \qquad \forall i \in [n_1], (j,j') \in E(T_2), \tag{A.56}$$

$$z_{i,(j,j')}^{(2)} \leq x_{i,j'} \qquad \forall i \in [n_1], (j,j') \in E(T_2), \tag{A.57}$$

$$z_{i,(j,j')}^{(2)} \geq x_{i,j} + x_{i,j'} - 1 \qquad \forall i \in [n_1], (j,j') \in E(T_2), \tag{A.58}$$

$$\sum_{j=1}^{n_2} z_{(i,i'),j}^{(1)} = 1 \qquad \forall (i,i') \in E(T_1), \tag{A.59}$$

$$\sum_{i=1}^{n_1} z_{i,(j,j')}^{(2)} = 1 \qquad \forall (j,j') \in E(T_2), \tag{A.60}$$

$$x_{i,j} \in \{0,1\}, \qquad \forall i \in [n_1], j \in [n_2], \tag{A.61}$$

$$u_{p,i}^{(1)} \in [0,1], \qquad \forall p \in [m], i \in [n_1], \tag{A.62}$$

$$u_{p,j}^{(2)} \in [0,1], \qquad \forall p \in [m], j \in [n_2], \tag{A.63}$$

$$c_{p,i}^{(1)} \in [0,1], \qquad \forall p \in [m], i \in [n_1], \tag{A.64}$$

$$c_{p,j}^{(2)} \in [0,1], \qquad \forall p \in [m], j \in [n_2], \tag{A.65}$$

$$z_{(i,i'),j}^{(1)} \geq 0 \qquad \forall (i,i') \in E(T_1), j \in [n_2], \tag{A.66}$$

$$z_{i,(j,j')}^{(2)} \geq 0 \qquad \forall i \in [n_1], (j,j') \in E(T_2). \tag{A.67}$$

Appendix B: Complexity Proofs

B.1 COMPLEXITY OF DIRECT TRANSMISSION INFERENCE PROBLEM

This section shows the hardness of the decision and the counting versions of the DTI problem by reduction from the one-in-three SAT (1-in-3 SAT).

Problem B.1 (1-in-3SAT)**.** Given a Boolean formula $\phi = \bigwedge_{i=1}^{k}(y_{i,1} \vee y_{i,2} \vee y_{i,3})$ in 3-conjunctive normal form (3-CNF) with n variables and k clauses, decide whether there exists a truth assignment $\theta : [n] \to \{0, 1\}$ so that each clause has *exactly* one true literal (and thus exactly two false literals).

B.1.1 Decision Problem

To relate literals to variables, we use the function $\nu : [k] \times \{1, 2, 3\} \to [n]$ such that $\nu(i, j)$ is the variable corresponding to literal $y_{i,j}$. We define $\sigma(i, j)$ to be 1 if $y_{i,j}$ is a positive literal (*i.e.* $y_{i,j} = x_{\nu(i,j)}$), otherwise $\sigma(i, j) = 0$ if $y_{i,j}$ is a negative literal (*i.e.* $y_{i,j} = \neg x_{\nu(i,j)}$). A truth assignment θ satisfies ϕ if for each clause $i \in [k]$ there exists a $j \in \{1, 2, 3\}$ such that $\sigma(i, j) = \theta(\nu(i, j))$.

Given ϕ, we construct a timed phylogeny $T(\phi)$ with leaf labeling $\hat{\ell}$, a contact map $C(\phi)$ and time-stamps τ, τ_e, τ_r, as depicted in Fig. B.1 and detailed below. We set $\Sigma = \{\perp, x_1, \ldots, x_n, \neg x_1, \ldots, \neg x_n\, c_1, \ldots, c_k\}$. Let $\varepsilon > 0$ be a small positive constant. As for entry and removal time-stamps, we set $\tau_e(\perp) = 0, \tau_r(\perp) = \varepsilon$, and $\tau_e(x_i) = \tau_e(\neg x_i) = \varepsilon$ and $\tau_r(x_i) = \tau_r(\neg x_i) = 3\varepsilon$ for each variable $i \in [n]$. For each clause $c_i,\ i \in [k]$ we set $\tau_e(c_i) = \tau_r(c_i) = 3\varepsilon$. Timed phylogeny $T(\phi)$ is composed of $3k$ clause gadgets and n variable gadgets, each corresponding to a subtree that is directly attached to the root $r(T(\phi))$. The root vertex has time-stamp $\tau(r(T(\phi)) = 0$. The leaves of T have identical time-stamps 3ε. For each variable $i \in [n]$, we have a subtree $T[x_i]$ whose root has time-stamp $\tau(r(T[x_i])) = 2\varepsilon$. The two children of $r(T[x_i])$ have identical time-stamps 3ε, with one child leading to two leaves labeled by positive literal x_i and the other child leading to two leaves labeled by negative literals $\neg x_i$. Similarly, for each clause $c_i,\ i \in [k]$, we have 3 subtrees $T[y_{i,1}], T[y_{i,2}]$ and $T[y_{i,3}]$. The root of the subtree $T[y_{i,j}]$ has time-stamp ε and two children, one of which is the leaf labeled by $x_{\nu(i,j)}$ if $y_{i,j} = \neg x_{\nu(i,j)}$ and $\neg x_{\nu(i,j)}$ if $y_{i,j} = x_{\nu(i,j)}$. The other child node, denoted as $v_{i,j}$, has time-stamp $\tau(v_{i,j}) = 2\varepsilon$ and has only one child which is a leaf labeled by c_i. The contact map $C(\phi)$ is constructed as follows. The vertex set for the contact map is given by Σ. We have a directed edge from $\perp$ to each of the variables $\{x_1, \cdots, x_n, \neg x_1, \cdots, \neg x_n\}$.

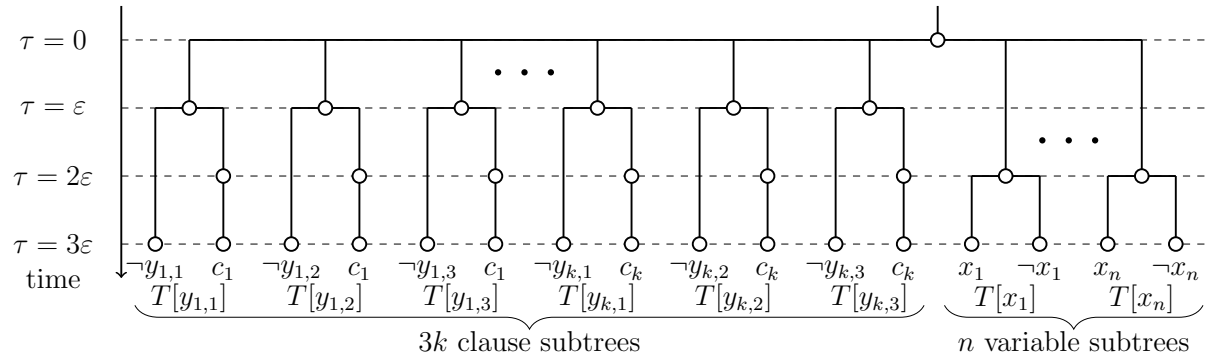


Figure B.1: **Construction of $T(\phi)$ for reduction from 1-in-3SAT to DTI.** Let ϕ be an 1-in-3SAT formula with k clauses and n variables. $T(\phi)$ is built with a root node $r(T(\phi))$ can is connected to $3k$ clause subtrees $\{T[y_{1,1}], T[y_{1,2}], T[y_{1,3}], \cdots, T[y_{k,1}], T[y_{k,2}], T[y_{k,3}]\}$ and n variable subtrees $\{T[x_1], \cdots, T[x_n]\}$. We set $\tau_e(\perp) = 0, \tau_r(\perp) = \varepsilon$, and $\tau_e(x_i) = \tau_e(\neg x_i) = \varepsilon$ and $\tau_r(x_i) = \tau_r(\neg x_i) = 3\varepsilon$ for each variable $i \in [n]$. For each clause $c_i,\ i \in [k]$ we set $\tau_e(c_i) = \tau_r(c_i) = 3\varepsilon$. We prove that there exits a truth assignment so that each clause of ϕ has exactly one true literal if and only if there exists a vertex labeling for $T(\phi)$ that results in a transmission tree that is a spanning arborescence of the contact map $C(\phi)$ (Fig. B.2).

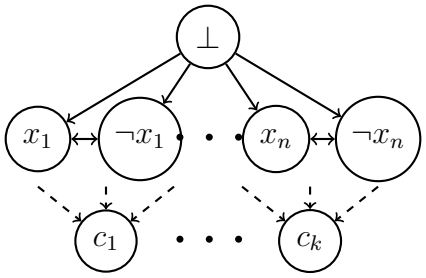


Figure B.2: **Construction of $C(\phi)$ for reduction from 1-in-3SAT to DTI.** Let ϕ be an 1-in-3SAT formula with k clauses and n variables. The host set is $\Sigma = \{\perp, x_1, \cdots, x_n, \neg x_1, \cdots, \neg x_n, c_1, \cdots, c_k\}$. We have a directed edge from $\perp$ to each of the variables $\{x_1, \cdots, x_n, \neg x_1, \cdots, \neg x_n\}$. Each each $i \in [n]$, variable x_i has an outgoing edge to $\neg x_i$ and similarly variable $\neg x_i$ has an outgoing edge to x_i. Finally, each clause c_i has three incoming edges, one from each of the literals that form the clause, i.e. $y_{i,1}, y_{i,2}$ and $y_{i,3}$.

For $i \in [n]$, each variable x_i has an outgoing edge to $\neg x_i$ and similarly variable $\neg x_i$ has an outgoing edge to x_i. Finally, each clause c_i has three incoming edges, one from each of the literals that form the clause, i.e. $y_{i,1}, y_{i,2}$ and $y_{i,3}$. For instance, if $c_1 \coloneqq (x_1 \vee x_2 \vee \neg x_3)$, then we have the directed edges $(x_1, c_1), (x_2, c_1)$ and $\neg x_3, C_1$. Clearly, $T(\phi)$ and $C(\phi)$ can be obtained in polynomial time from ϕ. An example of this reduction is shown in Fig. B.3.

Lemma B.1. For any vertex labeling ℓ of $T(\phi)$, $\perp$ is the *root host*.

Proof. Under the direct transmission constraint, *root host* is given by the host that labels the root node of the timed phylogeny. The time stamp of the root node of $T(\phi)$ is $\tau(r(T(\phi))) = 0$.

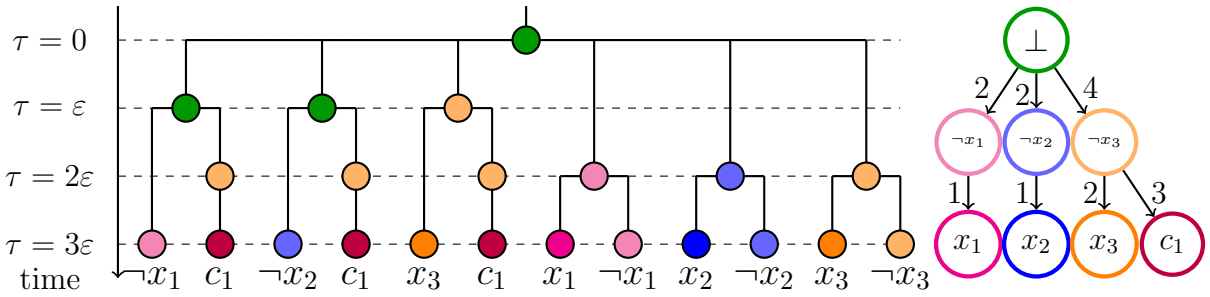


Figure B.3: **Example of reduction.** Consider the 1-in-3SAT Boolean formula $\phi = (x_1 \vee x_2 \vee \neg x_3)$. ϕ is satisfiable with truth assignment $\theta(1) = 0, \theta(2) = 0$ and $\theta(3) = 0$. Figures (on the left) shows a vertex labeling ℓ corresponding to θ. Since the vertex labeling admits a transmission tree (one the right), ϕ is Exactly-1 satisfied with truth assignment θ.

The only host that has entry time before $\tau_e \leq 0$ is $\perp$. Therefore, for any vertex labeling we have $\ell(r(T(\phi))) = \perp$, which makes $\perp$ the *root host*. QED.

Lemma B.2. For any variable x, either $\{(\perp, x), (x, \neg x)\} \subseteq E(S)$ or $\{(\perp, \neg x), (\neg x, x)\} \subseteq E(S)$.

Proof. For any variable x, consider the subtree $T[x]$. By construction we have, $\tau(r(T[x])) = 2\varepsilon$ and the node only has two children labeled by x and $\neg x$. From the contact map we know that the only possible infectors for x has $\perp$ and $\neg x$ and similarly for $\neg x$ are $\perp$ and x. Given that $\tau_r(\perp) < \tau(r(T[x]))$, the only remaining choices for $\ell(r(T[x]))$ are x and $\neg x$.

If $\ell(r(T[x])) = x$ then we have $\{(\perp, x), (x, \neg x)\} \subseteq E(S)$ and if $\ell(r(T[x])) = \neg x$ we have $\{(\perp, \neg x), (\neg x, x)\} \subseteq E(S)$. QED.

Lemma B.3. For any clause $c_i = (y_{i,1} \vee y_{i,2} \vee y_{i,3})$, if $(y_{i,j}, c_i) \in E(S)$ then $\ell(r(T[y_{i,j}])) = y_{i,j}$ and $\ell(r(T[y_{i,j'})) = \perp$ for $j' \neq j$.

Proof. Consider the subtree $T[y_{i,j}]$. Let us denote the node that is child of $r(T[y_{i,j}])$ and parent of the leaf of $T[y_{i,j}]$ labeled with c_i as v_j.

Since S is a spanning arborescence of $C(\phi)$ we have either $(y_{i,1}, c_i), (y_{i,2}, c_i)$ or $(y_{i,3}, c_i)$ in $E(S)$. Without loss of generality, let us assume that $(y_{i,1}, c_i) \in E(S)$.

The edges $(v_1, \delta_T(v_1)), (v_2, \delta_T(v_2))$ and $(v_3, \delta_T(v_3))$ need to be transmission edges since $\tau(v_1) = \tau(v_2) = \tau(v_3) < \tau_e(c_i)$. Since $(y_{i,1}, c_i) \in E(S)$, we require $\ell(v_1) = \ell(v_2) = \ell(v_3) = y_{i,1}$. Looking at $r(T[y_{i,2}])$ and $r(T[y_{i,3}])$, since each clause consists of distinct variables, we can only have $\ell(r(T[y_{i,2}])) = \ell(r(T[y_{i,3}])) = \perp$. Consequently, the transmission edges $(r(T[y_{i,2}]), v_2)$ and $(r(T[y_{i,3}]), v_3)$ results in a edge $(\perp, y_{i,1})$ in $E(S)$. By Lemma B.2, this also means $(y_{i,1}, \neg y_{i,1}) \in E(S)$ and therefore $\ell(r(T[y_{i,1}])) = y_{i,1}$. QED.

Lemma B.4. For any literal $y_{i,j}$ in clause c_i, $(\perp, y_{i,j}) \in E(S)$ if and only if $(y_{i,j}, c_i) \in E(S)$.

Proof. Consider the subtree $T[y_{i,j}]$. Let us denote the node that is child of $r(T[y_{i,j}])$ and parent of the leaf of $T[y_{i,j}]$ labeled with c_i as v.

($\Rightarrow$) If $(\perp, y_{i,j}) \in E(S)$, then by Lemma B.2 we know that $(y_{i,j}, \neg y_{i,j}) \in E(S)$. Therefore, $\ell(r(T[y_{i,j}])) = y_{i,j}$. Given that $\ell(r(T[y_{i,j}])) = y_{i,j}$, $\ell(\delta_T(v)) = c_i$ and $\tau(v) = \varepsilon$, the only feasible label for v is $y_{i,j}$. Therefore $\ell(v) = y_{i,j}$ and $(y_{i,j}, c_i) \in E(s)$.

($\Leftarrow$) If $(y_{i,j}, c_i) \in E(S)$, then since $\tau(v) < \tau_e(c_i)$, we have $\ell(v) = y_{i,j}$. From Lemma B.3 we know that $\ell(r(T[y_{i,j}]))$ is either $\perp$ or $y_{i,j}$. If $\ell(r(T[y_{i,j}])) = \perp$, then we will have $\{(\perp, y_{i,j}), (\perp, \neg y_{i,j})\}$ which is not possible due to Lemma B.2. Therefore $\ell(r(T[y_{i,j}])) = y_{i,j}$ and consequently $(\perp, y_{i,j}) \in E(S)$. QED.

Proposition B.1. There exists a vertex labeling ℓ of $T(\phi)$ under the direct transmission constraint such that the corresponding transmission tree $S(\ell)$ is a spanning arborescence of $C(\phi)$ if and only if ϕ is satisfiable with a truth assignment θ so that each clause has exactly one true literal.

Proof. ($\Rightarrow$) Let ℓ be a vertex labeling of $T(\phi)$ under the direct transmission constraint such that the corresponding transmission tree S is a spanning arborescence of $C(\phi)$. We construct the corresponding truth assignment θ for ϕ as follows. From Lemma B.2 we know that for any variable x, either $(\perp, x) \in E(S)$ or $(\perp, \neg x) \in E(S)$. We set $\theta(i) = 1$ if $(\perp, x_i) \in E(S)$ and $\theta(i) = 0$ if $(\perp, \neg x_i) \in E(S)$. We claim that the this truth assignment satisfies ϕ with exactly one literal for each clause.

We need to show that, for any clause $c_i = (y_{i,1} \vee y_{i,2} \vee y_{i,3})$, exactly one of $(\perp, y_{i,1}), (\perp, y_{i,2})$ and $(\perp, y_{i,3})$ is in $E(S)$. From Lemma B.4 we know that $(\perp, y_{i,j}) \in E(S)$ if and only if $(y_{i,j}, c_i) \in E(S)$. Since S is a spanning arborescence, exactly one of $(y(i,1), c_i), (y_{i,2}, c_i)$ and $(y_{i,3}, c_i)$ is in $E(S)$. Therefore, exactly one of $(\perp, y_{i,1}), (\perp, y_{i,2})$ and $(\perp, y_{i,3})$ is in $E(S)$ which renders the clause c_i satisfied with exactly one literal.

($\Leftarrow$) Consider the truth assignment θ that satisfies ϕ with exactly one literal for each clause in ϕ. We build the vertex labeling ℓ for $T(\phi)$ as follows. From Lemma B.1 it is clear that $\perp$ is the root host and therefore $r(S) = \perp$. We set $\ell(T[x_i]) = x_i$ if $\theta(i) = 1$ and $\ell(T[x_i]) = \neg x_i$ if $\theta(i) = 0$. For any clause c_i in ϕ, if $y_{i,j}$ is true we set $\ell(r(T[y_{i,j}])) = y_{i,j}$ and if $\neg y_{i,j}$ is true we set $\ell(r(T[y_{i,j}])) = \perp$. Finally, we set $\ell(v_{i,j}) = y_{i,j}$ for all $j \in \{1, 2, 3\}$. We need to show that constructed vertex labeling satisfies the direct transmission constraint and that the resulting transmission tree is a spanning arborescence of the contact map $C(\phi)$. We do this by first showing that (i) each variable has a unique infector and (ii) all transmission edges between the same pair of hosts have time intervals that overlap.

Consider all the variables that are assigned true by the truth assignment. The infector for all these variables is $\perp$ since $\ell(r(T(\phi))) = \perp$ and $\ell(T[x_i]) = x_i$ if $\theta = 1$ and $\ell(r(T[y_{i,j}])) = \perp$

if $\neg y_{i,j}$ is true. This agrees with $C(\phi)$. The time intervals of the outgoing edges from $r(T(\phi))$ and $r(T[y_{i,j}]), \forall i \in [k], j \in \{1,2,3\}$ contain $\tau = \varepsilon$. Therefore, all possible transmission edges from $\perp$ overlap at $\tau = \varepsilon$.

Consider the variables that are assigned false by the truth assignment. From Lemma B.2 we know that for any such variable x, they are infected by $\neg x$. This agrees with $C(\phi)$. Moreover, these variables do not label any of the interval vertices of the tree T and all the leaves of T are at the same time-stamp $\tau = 3\varepsilon$. Therefore, all possible transmission edges to any such variable x overlap at $\tau = 3\varepsilon$.

Finally, consider any clause c_i. All the internal vertices $v_{i,j}, j \in \{1,2,3\}$ are labeled by the same variable $y_{i,j}$ that renders the clause c_i satisfied. As a result, $y_{i,j}$ is a unique infector of c_i and $(y_{i,j}, c)$ exists in $E(C(\phi))$ by construction. Also, time-stamp of all vertices $v_{i,j}$ are the same $\tau = 2\varepsilon$ and therefore, the transmission edges overlap at $\tau = 2\varepsilon$. QED.

B.1.2 Counting Problem

This section proves the #P-completeness of the #DTI problem.

Proposition B.2. There exists a parsimonious reduction from #1-in-3SAT to #DTI.

Proof. Consider the reduction shown in Section B.1.1. Here we show that this reduction is parsimonious, i.e. it preserves the number of solutions in the solution spaces of the two problems. We show a bijection between the solution space of a 1-in-3SAT and the solution space of the corresponding DTI instance.

Consider the Boolean formula ϕ. For a given truth assignment θ that satisfies each clause of ϕ with exactly one true literal, we construct the vertex labeling of $T(\phi)$ as following. We let $\ell(T[x_i]) = x_i$ if $\theta(i) = 1$ and $\ell(T[x_i]) = \neg x_i$ if $\theta(i) = 0$. We will show that this unique determines the labeling for the rest of the internal vertices of $T(\phi)$. Consider the clause c_i and the corresponding subtrees $T[y_{i,1}], T[y_{i,2}]$ and $T[y_{i,3}]$. Since the truth assignment satisfies each clause with exactly one literal, without loss generality, assume that $y_{i,1}$ is true. Then using Lemma B.4, since $(\perp, y_{i,j}) \in E(S)$, we have $(y_{i,j}, c_i) \in E(S)$. For the nodes $v_{i,j}$ we have $\tau(v_{i,j}) < \tau_e(c_i)$ and therefore $\ell(v_{i,j}) = y_{i,j}, \forall j \in \{1,2,3\}$. Finally, the vertex labels for the roots of the clause subtrees $\ell(r(T[y_{i,1}])) = \ell(r(T[y_{i,2}])) = \ell(r(T[y_{i,3}])) = y_{i,1}$ due to Lemma B.3. Proof of Proposition B.1 shows that this vertex labeling is a solution of the DTI problem.

From a given vertex labeling ℓ, we construct the truth assignment as follows. We set $\theta(i) = 1$ if $\ell(r(T[x_i])) = x_i$ and $\theta(i) = 0$ if $\ell(r(T[x_i])) = \neg x_i$. Proof of Proposition B.1 shows that this is a truth assignment that satisfies each clause with exactly one true literal.

The construction of θ from ℓ and ℓ from θ are inverses of each other. If we view these constructions as functions then they show a bijection in the solutions spaces of #1-in-3SAT and #DTI. This shows that the number of solutions is preserved. Obviously, the reduction can be performed in polynomial time. Therefore, the reduction is parsimonious. QED.

B.2 COMPLEXITY OF PCR AND PCTR PROBLEMS

The following two Lemmas prove the hardness of the PCR (Section 5.2.1) and the PCTR ((Section 5.2.2)) problems.

Lemma B.5. Given proportions $U_1(A, B)$ for clones $\Pi_1(A, B) = [3q]$ and proportions $U_2(A, B)$ for clones $\Pi_2(A, B) = [q]$, there exists a set Π of clones of size $n = |\Pi| \leq 3q$ with proportions U that are consistent with $U_1(A, B)$ and $U_2(A, B)$ if and only if there exists a solution to the 3-PARTITION instance (A, B).

Proof. ($\Rightarrow$) Let clones Π and proportion matrix U be a solution to the PCR problem instance $(\Pi_1(A, B), U_1(A, B), \Pi_2(A, B), U_2(A, B))$. By the premise, we have that $n = |\Pi| \leq 3q$. Note that since $n_1 = |\Pi_1| = 3q > q = |\Pi_2| = n_2$ and $u^{(1)}_{1,i} > 0$ for all $i \in [3q]$, by Observation 5.1, we have $n = |\Pi| \geq 3q$. Putting this together with the upper bound $n = |\Pi| \leq 3q$, obtained from the premise, we have that the PCR solution has $n = 3q$ clones. Note that since $|\Pi_1| = |\Pi|$, for every $i \in [3q]$, there is a unique $j \in [q]$ such that $(i, j) \in \Pi$. Since U is consistent with U_1 and u_2, we have that

$$u_{1,(i,j)} = u^{(1)}_{1,i}. \tag{B.1}$$

We claim that the solution to the 3-PARTITION problem instance (A, B) is given by the function $\sigma(i) = j$ where $(i, j) \in \Pi$, for each $i \in [3q]$.

We show that σ defined above satisfies Equation (5.2). Recall that $\pi_1((i, j)) = i$ and $\pi_2((i, j)) = j$. For any $j \in [q]$, we have

$$\sum_{i \in \sigma^{-1}(j)} a_i = \sum_{(i,j') \in \Pi : \pi_2((i,j')) = j} a_i \tag{B.2}$$

$$= \sum_{(i,j') \in \Pi : \pi_2((i,j')) = j} Bqu^{(1)}_{1,s} \tag{B.3}$$

$$= \sum_{(i,j') \in \Pi : \pi_2((i,j')) = j} Bqu_{1,(i,j)} \tag{B.4}$$

$$= Bqu^{(2)}_{1,j} = B, \tag{B.5}$$

where the second equality follows from construction with $u^{(1)}_{1,i} = a_i/(Bq)$, the third equality uses Equation (B.1), the fourth equality uses consistency of proportion matrix U with respect to U_2 given projection function π_2 and the fifth equality uses the construction $u^{(2)}_{1,j} = 1/q$.
($\Leftarrow$) Let $\sigma : [3q] \to [q]$ be a solution to the 3-PARTITION problem instance (A, B). We claim that $\Pi = \{(i, \sigma(i)) : \ i \in [3q]\}$ with $n = |\Pi| = 3q$ clones and $1 \times n$ proportion matrix $U = [u_{1,(i,\sigma(i))}]$ where $u_{1,(i,\sigma(i))} = u^{(1)}_{1,i} = a_i/(Bq)$ is a solution to the corresponding PCR problem.

To see why, recall that $\pi_1((i, \sigma(i))) = s$ and $\pi_2((i, \sigma(i))) = \sigma(i)$. Given these projection functions, we need to show that $U$is consistent with $U_1(A, B)$ and $U_2(A, B)$. The consistency with respect to the Π_1-clones is trivial as for each $i \in \Pi_1 = [3q]$ there exists exactly one pair $(i, j) \in \Pi$, i.e., the pair (i, j) where $j = \sigma(i)$, with proportion $u_{1,(i,\sigma(i))} = u^{(1)}_{1,i}$. To see the consistency with respect to the Π_2-clones, consider for any $j \in \Pi_2 = [q]$,

$$\sum_{(i,\sigma(i)):\sigma(i)=j} u_{1,(i,\sigma(i))} = \sum_{i \in \sigma^{-1}(j)} u^{(1)}_{1,i} = \sum_{i \in \sigma^{-1}(j)} \frac{a_i}{Bq} = \frac{1}{Bq} B = \frac{1}{q}, \tag{B.6}$$

where the second to last equality uses Equation (5.2). Since $u^{(2)}_{1,j} = 1/q$ for all $j \in [q]$, we have

$$\sum_{(i,\sigma(i)):\sigma(i)=j} u_{1,(i,\sigma(i))} = u^{(2)}_{1,j}, \quad \forall j \in [q],$$

which is the required condition for consistency. QED.

Lemma B.6. Given proportions $U_1(A, B)$ and clone tree T_1 for clones $\Pi_1(A, B) = \{0\} \cup [3q]$ and proportions $U_2(A, B)$ and clone tree T_2 for clones $\Pi_2(A, B) = \{0\} \cup [q]$, there exists a set Π of clones of size $n = |\Pi| = 4q + 1$, clone tree T and proportion matrix U such that T is a refinement of T_1 and T_2 and $J(U, U_1, U_2) = 0$ if and only if there exists a solution of the 3-PARTITION instance (A, B).

Proof. ($\Rightarrow$) Let clones Π, clone tree T and proportion matrix U be a solution to the PCTR problem instance $(\Pi_1(A, B), T_1(A, B), U_1(A, B), \Pi_2(A, B), T_2(A, B), U_2(A, B))$. By the premise, we have that $J(U, U_1, U_2) = 0$, which implies that U is consistent with U_1 and U_2.

Note that by Observation 5.2, since $r(T_1) = 0$ and $r(T_2) = 0$, we have $(0, 0) \in \Pi$ and $r(T) = (0, 0)$. Recall that $\pi_1 : \Pi \to \Pi_1$ maps each clone in Π to its corresponding Π_1-clone. We claim that this function is a bijection for this construction. That is, $|\pi_1^{-1}(i)| = 1$ for all $i \in [3q]$. Clearly $|\pi_1^{-1}(i)| \geq 1$ by Definition 5.3 condition (i) and Observation 5.2. As for why $|\pi_1^{-1}(i)| \leq 1$, assume for a contradiction, there exists an $i \in [3q]$ such that $\{j, j'\} \subseteq \pi_1^{-1}(i)$,

i.e. $(i,j),(i,j') \in \Pi$ for two distinct $j, j' \in \Pi_2$. Since $(i,j) \in \Pi$, by Observation 5.3 either $((0,j),(i,j)) \in E(T)$ or $((i,0),(i,j)) \in E(T)$. Similarly since $(i,j') \in \Pi$, by Observation 5.3, either $((0,j'),(i,j')) \in E(T)$ or $((i,0),(i,j')) \in E(T)$. Putting these two conditions together, either $\{((0,j),(i,j)),((0,j'),(i,j'))\} \subseteq E(T)$ or $\{((i,0),(i,j)),((i,0),(i,j'))\} \in E(T)$. We analyze these two cases as follows.

- **Case 1**: Consider $\{((0,j),(i,j)),((0,j'),(i,j'))\} \subseteq E(T)$. This violates condition (i) in the Definition 5.3 of T being a refinement of T_1 and T_2.

- **Case 2**: Consider $\{((i,0),(i,j)),((i,0),(i,j'))\} \in E(T)$. Note that the proportion $u^{(1)}_{1,i}$ of each clone $i \in \Pi_1 \setminus \{0\}$ occurs within the open interval $(1/4q, 1/2q)$ and therefore $u^{(1)}_{1,i} < 1/q$. Since the proportion $u^{(2)}_{1,j} = 1/q$ for any clone $j \in \Pi_2 \setminus \{0\}$, we have $|\pi_2^{-1}(j)| > 1$. Let $i' \in \pi_2^{-1}(j)$ such that $i \neq i'$. Since $(i',j) \in \Pi$, by Observation 5.3, either $((i',0),(i',j)) \in E(T)$ or $((0,j),(i',j)) \in E(T)$. If $((i',0),(i',j)) \in E(T)$, since by premise $\{((i,0),(i,j)),((i,0),(i,j'))\} \in E(T)$, it will violate condition (ii) in the Definition 5.3 of T being refinement of T_1 and T_2. Alternatively, $((0,j),(i',j)) \in E(T)$ implies that $(0,j) \in \Pi$, which by condition (iii) in Definition 5.3 entails $((0,0),(0,j)) \in E(T)$. However, since $((i,0),(i,j)) \in E(T)$, this would violate condition (ii) in Definition 5.3.

Therefore, $|\pi_1^{-1}(i)| = 1$ for $i \in [3q]$. Also, since $u^{(2)}_{1,0} = 0$ and since the proportion matrix U is consistent with U_1 and U_2, we have $\pi_1^{-1}(i) = j \in [q]$ for each $i \in [3q]$. We claim that the solution of the 3-PARTITION problem instance (A, B) is $\sigma(i) = j$ for $(i,j) \in \Pi$, for each $i \in [3q]$. We show that σ defined above satisfies Equation (5.2). Recall that $\pi_1((i,j)) = i$ and $\pi_2((i,j)) = j$. For any $j \in [q]$, we have

$$\sum_{i \in \sigma^{-1}(j)} a_i = \sum_{(i,j') \in \Pi : \pi_2((i,j'))=j} a_i \tag{B.7}$$

$$= \sum_{(i,j') \in \Pi : \pi_2((i,j'))=j} Bqu^{(1)}_{1,s} \tag{B.8}$$

$$= \sum_{(i,j') \in \Pi : \pi_2((i,j'))=j} Bqu_{1,(i,j)} \tag{B.9}$$

$$= Bqu^{(2)}_{1,j} \tag{B.10}$$

$$= Bq\frac{1}{q} \tag{B.11}$$

$$= B, \tag{B.12}$$

where the second equality follows from construction with $u^{(1)}_{1,i} = a_i/(Bq)$, the third equality uses Equation (B.1), the fourth equality uses consistency of proportion matrix U with respect

to U_2 given projection function π_2 and the fifth equality uses the construction $u^{(2)}_{1,j} = 1/q$.

($\Leftarrow$) Let $\sigma : [3q] \to [q]$ be a solution to the 3-PARTITION problem instance (A, B). We claim that $\Pi = \{(0, j) : j \in [q] \cup \{0\}\} \cup \{(i, \sigma(i)) : i \in [3q]\}$ with $n = |\Pi| = 4q+1$ clones, clone tree T with edges $E(T) = \{((0, 0), (0, j)) : j \in [q]\} \cup \{((0, j), (i, j)) : i \in \sigma^{-1}(j), j \in [q]\}$ and $1 \times (4q + 1)$ proportion matrix $U = [u_{1,(i,j)}]$ where $u_{1,(i,\sigma(i))} = u^{(1)}_{1,i} = a_i/(Bq)$ when $i \in [3q]$, $u_{1,(0,j)} = 0$ for $j \in [q] \cup \{0\}$ is a solution to the PCTR problem with $J(U, U_1, U_2) = 0$.

We first show that U is consistent with respect to U_1 and U_2 which is equivalent to the condition $J(U, U_1, U_2) = 0$. Recall the projection functions $\pi_1((i, j)) = i$ and $\pi_2((i, j)) = j$. We show consistency with respect to U_1 as follows. For $i = 0$, since $\pi_1^{-1}(0) = [q] \cup \{0\}$, we have

$$\sum_{j \in \pi_1^{-1}(0)} u_{1,(0,j)} = \sum_{j=0} u_{1,(0,j)} = 0 = u^{(1)}_{1,0}. \tag{B.13}$$

For $i \in [3q]$, since $\pi_1^{-1}(i) = \sigma(i)$ we have,

$$\sum_{j \in \pi_1^{-1}(i)} u_{1,(i,j)} = u_{1,(i,\sigma(i))} = a_i/(Bq) = u^{(1)}_{1,i}. \tag{B.14}$$

We show consistency with respect to U_2 as follows. For $j = 0$, since $\pi_2^{-1}(0) = 0$, we have

$$\sum_{i \in \pi_2^{-1}(0)} u_{1,(i,0)} = u_{1,(0,0)} = 0 = u^{(2)}_{1,0}. \tag{B.15}$$

For $j \in [q]$, since $\pi_2^{-1}(j) = \sigma^{-1}(j)$, we have

$$\sum_{i \in \pi_2^{-1}(j)} u_{1,(i,j)} = \sum_{i \in \sigma^{-1}(j)} u_{1,(i,j)} = \sum_{i \in \sigma^{-1}(j)} a_i/(Bq) = B/(Bq) = 1/q = u^{(2)}_{1,j}, \tag{B.16}$$

where the third equality uses the premise that σ is a solution of the 3-PARTITION problem instance (A, B).

Now we show that T is a refinement of T_1 and T_2. We address the three conditions in Definition 5.3 as follows.

- **Condition (i):** Recall that $E(T_1) = \{(0, i) : i \in [3q]\}$. For each edge $(0, i) \in E(T_1)$, we have a unique $j = \sigma(i) \in [q]$ such that $((0, j), (i, j)) \in E(T)$.
- **Condition (ii):** Recall that $E(T_2) = \{(0, j) : j \in [q]\}$. For each edge $(0, j) \in E(T_2)$, we have a unique $i = 0$ such that $((0, 0), (0, j)) \in E(T)$.

- **Condition (iii):** Recall that $E(T) = \{((0,0),(0,j)) : j \in [q]\} \cup \{((0,j),(i,j)) : i \in \sigma^{-1}(j), j \in [q]\}$. For each edge $((0,0),(0,j)) \in E(T), j \in [q]$, we have $(0,j) \in E(T_2)$ and for edge $((0,j),(i,j)) \in E(T), i \in \sigma^{-1}(j), j \in [q]$, we have $(0,i) \in E(T_1)$.

QED.

Appendix C: Other Proofs and Derivations

C.1 TRANSMISSION TREE DISTANCE METRIC

In this section we show that WPCD is a distance metric. To show that WPCD is a distance metric, for any transmission tree S_i, we define the function $q_i : \Sigma \times \Sigma \to \mathbb{N}$ as

$$q_i(s,t) = \begin{cases} w_i(s,t), & (s,t) \in E(S_i), \\ 0, & \text{otherwise.} \end{cases} \tag{C.1}$$

Observe that, by construction, q_i uniquely determines the transmission tree S_i since for any edge $(s,t) \in E(S_i)$ we have $w_i(s,t) > 0$. Further, the WPCD between any two transmission trees S_1 and S_2 can be alternatively written in terms of q_1 and q_2 as follows,

$$d(S_1, S_2) = \sum_{(s,t)\in\Sigma\times\Sigma} |q_1(s,t) - q_2(s,t)|. \tag{C.2}$$

Proposition C.1. WPCD is a distance metric on the space of transmission trees $\mathcal{T}$.

Proof. First, we show that for any two transmission trees S_1 and S_2, $d(S_1, S_2) = 0$ if and only if $S_1 = S_2$. Clearly when $S_1 = S_2$, we have $d(S_1, S_2) = 0$. Now, let us consider the case $d(S_1, S_2) = 0$. For any $(s,t) \in \Sigma \times \Sigma$, $|q_1(s,t) - q_2(s,t)| \geq 0$. Therefore, if $d(S_1, S_2)$ then for all $(s,t) \in \Sigma \times \Sigma$ we have $q_1(s,t) = q_2(s,t)$ implying that $S_1 = S_2$.

By definition, WPCD is always nonnegative and symmetric. We only need to show the triangle inequality, *i.e.* given trees S_1, S_2 and S_3, we must show

$$d(S_1, S_3) \leq d(S_1, S_2) + d(S_2, S_3). \tag{C.3}$$

We show this as follows.

$$\begin{aligned} d(S_1, S_3) &= \sum_{(s,t)\in\Sigma\times\Sigma} |q_1(s,t) - q_3(s,t)| && \text{(C.4)} \\ &= \sum_{(s,t)\in\Sigma\times\Sigma} |q_1(s,t) - q_2(s,t) + q_2(s,t) - q_3(s,t)| && \text{(C.5)} \\ &\leq \sum_{(s,t)\in\Sigma\times\Sigma} (|q_1(s,t) - q_2(s,t)| + |q_2(s,t) - q_3(s,t)|) && \text{(C.6)} \\ &= d(S_1, S_2) + d(S_2, S_3). && \text{(C.7)} \end{aligned}$$

QED.

C.2 CONSENSUS TRANSMISSION TREE ALGORITHM

Theorem C.1. Given a set $\mathcal{S} = \{S_1, \cdots, S_k\}$ of k transmission trees with edge weights $w_{S_1}, \cdots, w_{S_k}$, the minimum weight spanning arborescence of the corresponding weighted parent-child graph P defines a tree R that is a solution to the SCTT problem with the distance measure used is weighted parent-child distance.

Proof. Consider the *weighted parent-child graph* P for the set of transmission trees $\mathcal{S}$. Since P is a complete graph, the optimal consensus tree R is necessarily a spanning arborescence of P. The weights of the edges in R are given by w^* (Proposition 2.1).

$$w^*(s,t) = \arg\min_{z>0} \sum_{S_i \in \mathcal{S}} |q_i(s,t) - z|. \tag{C.8}$$

The total WPCD of R from the set of transmission trees $\mathcal{S}$ is given by $d(R,\mathcal{S}) = \sum_{S_i \in \mathcal{S}} d(R, S_i)$ where

$$d(R,S_i) = \sum_{(s,t)\in E(R)} |q_i(s,t) - w^*(s,t)| + \sum_{(s,t)\notin E(R)} |q_i(s,t)| \tag{C.9}$$

$$= \sum_{(s,t)\in E(R)} (|q_i(s,t) - w^*(s,t)| - |q_i(s,t)|) + \sum_{(s,t)\in \Sigma\times\Sigma} |q_i(s,t)|. \tag{C.10}$$

Consequently,

$$d(R,\mathcal{S}) = \sum_{S_i\in\mathcal{S}} \sum_{(s,t)\in\Sigma\times\Sigma} |q_i(s,t)| + \sum_{(s,t)\in E(R)} w_P(s,t), \tag{C.11}$$

where the first term is a constant with respect to R and minimizing the second term is the sum of the weights of a minimum weight spanning arborescence R of P. QED.

C.3 LIKELIHOOD MODEL FOR DISCONTINUOUS TRANSCRIPTION

We use the segment graph G to compute the probability $\Pr(\mathcal{R} \mid \mathcal{T}, \mathbf{c})$ of observing the alignment $\mathcal{R}$ given transcripts $\mathcal{T}$ and abundances $\mathbf{c}$. We follow the generative model described in [171], which has been extensively used for transcription quantification [68, 69, 75]. Let the set $\mathcal{R}$ of reads be $\{1, \ldots, r_n\}$ and the set $\mathcal{T}$ of transcripts be $\mathcal{T} = \{T_1, \ldots, T_k\}$ with lengths $L_1, \ldots, L_k$ and abundances $\mathbf{c} = [c_1, \ldots, c_k]$. In line with current literature, reads

$\mathcal{R}$ are generated independently from transcripts $\mathcal{T}$ with abundances $\mathbf{c}$. Further, we must marginalize over the set of transcripts $\mathcal{T}$ as the transcript of origin of any given read is typically unknown, due to $\ell \ll L$. Thus,

$$\Pr(\mathcal{R} \mid \mathcal{T}, \mathbf{c}) = \prod_{j=1}^{n} \Pr(r_j \mid \mathcal{T}, \mathbf{c}) \tag{C.12}$$

$$= \prod_{j=1}^{n} \sum_{i=1}^{k} \Pr(r_j, Z_{i,j} \mid \mathcal{T}, \mathbf{c}) \tag{C.13}$$

$$= \prod_{j=1}^{n} \sum_{i=1}^{k} \Pr(r_j \mid Z_{i,j}) \Pr(Z_{i,j} \mid \mathcal{T}, \mathbf{c}), \tag{C.14}$$

where $Z_{i,j}$ is the indicator random variable for the event that T_i is the transcript of origin for read r_j. We denote by $\Pr(r_j \mid Z_{i,j})$ the probability of observing read r_j given that it is generated from transcript T_i and $\Pr(Z_{i,j} \mid \mathcal{T}, \mathbf{c})$ denotes the probability of generating a read from transcript T_i given transcripts $\mathcal{T}$ and abundances $\mathbf{c}$.

Assuming no amplification and sequencing bias, the probability $\Pr(Z_{i,j} \mid \mathcal{T}, \mathbf{c})$ of generating a read from a transcript T_i of length L_i is given by

$$\Pr(Z_{i,j} \mid \mathcal{T}, \mathbf{c}) = \frac{c_i L_i}{\sum_{j=1}^{k} c_j L_j}. \tag{C.15}$$

We now derive the probability $\Pr(r_j \mid Z_{i,j})$ of transcript T_i generating read r_j of fixed length ℓ. We do so using the segment graph $G = (V, E)$. Recall that a transcript T must correspond to an $\mathbf{s}$ to $\mathbf{t}$ path in G. Let $\pi(T) \subseteq E$ denote the path corresponding to transcript T. Similarly, each read r induces a path $\pi(r) \subseteq E$ in G. Read r can only be generated by transcript T if $\pi(r) \subseteq \pi(T)$. Hence, the probability of transcript T_i generating a given read r_j is given by

$$\Pr(r_j \mid Z_{i,j}) = \begin{cases} 1/L'_i, & \text{if } \pi(r_j) \subseteq \pi(T_i), \\ 0, & \text{otherwise,} \end{cases} \tag{C.16}$$

where $L'_i = L_i - \ell$ is the *effective length* of the transcript. We assume that the transcripts

are much longer than the reads and as such $L'_i/L_i \approx 1$. Putting it all together we get

$$\Pr(\mathcal{R} \mid \mathcal{T}, c) = \prod_{j=1}^{n} \sum_{i=1}^{k} \Pr(r_j \mid Z_{i,j}) \Pr(Z_{i,j} \mid \mathcal{T}, \mathbf{c}) \tag{C.17}$$

$$= \prod_{j=1}^{n} \sum_{i=1}^{k} \frac{\mathbf{1}\{\pi(r_j) \subseteq \pi(T_i)\}}{L'_i} \cdot \frac{c_i L_i}{\sum_{b=0}^{k} c_b L_b} \tag{C.18}$$

$$= \prod_{j=1}^{n} \sum_{i:\pi(T_i) \supseteq \pi(r_j)} \frac{1}{L'_i} \cdot \frac{c_i L_i}{\sum_{b=0}^{k} c_b L_b} \tag{C.19}$$

$$= \prod_{j=1}^{n} \frac{1}{\sum_{b=1}^{k} c_b L_b} \sum_{i:\pi(T_i) \supseteq \pi(r_j)} c_i \frac{L_i}{L'_i} \tag{C.20}$$

$$= \prod_{j=1}^{n} \frac{1}{\sum_{b=1}^{k} c_b L_b} \sum_{i:\pi(T_i) \supseteq \pi(r_j)} c_i. \tag{C.21}$$

C.4 RECHARACTERIZATION OF SOLUTIONS USING DISCONTINUOUS EDGES

We prove the following two propositions.

Proposition C.2. There is a bijection between subsets of discontinuous edges that are pairwise non-overlapping and $\mathbf{s} - \mathbf{t}$ paths in G.

Proof. Let Π be the set of $\mathbf{s} - \mathbf{t}$ paths in G. We indicate with Σ the family of subsets of discontinuous edges that are pairwise non-overlapping. Note that $\Sigma \subseteq 2^{E^{\curvearrowright}}$.

For an $\mathbf{s}-\mathbf{t}$ path $\pi \in \Pi$, let $f(\pi)$ be the set of discontinuous edges in π, *i.e.* $f(\pi) = \pi \cap E^{\curvearrowright}$. Since π is an $\mathbf{s}-\mathbf{t}$ path of G, we have that for each edge $(\mathbf{v} = [v^-, v^+], \mathbf{w} = [w^-, w^+]) \in \pi$ it holds that $v^+ \leq w^-$. Therefore, $f(\pi)$ is composed of pairwise non-overlapping disconnected edges.

Now, consider a subset $\sigma \in \Sigma$ of discontinuous edges that are pairwise non-overlapping. We obtain the corresponding $\mathbf{s}-\mathbf{t}$ path $f^{-1}(\sigma)$ by first ordering the edges of σ in ascending order. That is, let $\sigma = \{(\mathbf{v}_1 = [v_1^-, v_1^+], \mathbf{w}_1 = [w_1^-, w_1^+]), \ldots, (\mathbf{v}_{|\sigma|} = [v_{|\sigma|}^-, v_{|\sigma|}^+], \mathbf{w}_{|\sigma|} = [w_{|\sigma|}^-, w_{|\sigma|}^+])\}$ such that $w_i^+ \leq v_{i+1}^-$ for all $i \in \{1, \ldots, |\sigma| - 1\}$. For every two consecutive discontinuous edges $(\mathbf{v}_i = [v_i^-, v_i^+], \mathbf{w}_i = [w_i^-, w_i^+])$ and $(\mathbf{v}_{i+1} = [v_{i+1}^-, v_{i+1}^+], \mathbf{w}_{i+1} = [w_{i+1}^-, w_{i+1}^+])$, we include the corresponding subpath of continuous edges from $\mathbf{w}_i$ to $\mathbf{v}_{i+1}$ into $f^{-1}(\sigma)$. In addition, we include the subpath of continuous edges from node $\mathbf{s}$ to node $\mathbf{v}_1$ as well as the subpath from node $\mathbf{w}_{|\sigma|}$ to $\mathbf{t}$ into $f^{-1}(\sigma)$. By construction, $f^{-1}(\sigma)$ is an $\mathbf{s} - \mathbf{t}$ path. QED.

Proposition C.3. Let G be a segment graph, T be a transcript and r be a read. Then, $\pi(T) \supseteq \pi(r)$ if and only if $\sigma(T) \supseteq \sigma^{\oplus}(r)$ and $\sigma(T) \cap \sigma^{\ominus}(r) = \emptyset$.

Proof. **(⇒)** By the premise, $\pi(T) \supseteq \pi(r)$. By definition, $\sigma(T) = \pi(T) \cap E^{\curvearrowright}$. By Definition 3.4, $\sigma^{\oplus}(r) = \pi(r) \cap E^{\curvearrowright}$. As $\pi(T) \supseteq \pi(r)$, we have that $\sigma(T) = \pi(T) \cap E^{\curvearrowright} \supseteq \pi(r) \cap E^{\curvearrowright} = \sigma^{\oplus}(r)$. By definition, $\sigma^{\ominus}(r)$ is the subset of discontinuous edges in $E^{\curvearrowright} \setminus \sigma^{\oplus}(r)$ that overlaps with an edge in $\pi(r)$. Since $\pi(T) \supseteq \pi(r)$, every edge included in $\sigma^{\ominus}(r)$ because of an overlap with an edge in $\pi(r)$ must also overlap with the same edge in $\pi(T)$. Since $\pi(T)$ is an $\mathbf{s} - \mathbf{t}$ path, and thus does not contain pairwise overlapping edges, we infer that $\sigma^{\ominus}(r) \cap \sigma(T) = \emptyset$. **(⇐)** By the premise, $\sigma(T) \supseteq \sigma^{\oplus}(r)$ and $\sigma(T) \cap \sigma^{\ominus}(r) = \emptyset$. As $\sigma(T) \supseteq \sigma^{\oplus}(r)$, we have that $\pi(T) \cap E^{\curvearrowright} = \sigma(T) \supseteq \sigma^{\oplus}(r) = \pi(r) \cap E^{\curvearrowright}$. Since $\sigma(T) \cap \sigma^{\ominus}(r) = \emptyset$, we have by Definition 3.4, that no discontinuous edge in $\sigma(T)$ overlaps with any edge in $\pi(r)$. Since $\pi(T)$ is an $\mathbf{s} - \mathbf{t}$ path containing the subset $\sigma^{\oplus}(r)$ of discontinuous edges in $\pi(r)$, it holds that $\pi(T) \cap E^{\rightarrow} \supseteq \pi(r) \cap E^{\rightarrow}$. Finally, as $E^{\curvearrowright} \cup E^{\rightarrow} = E$, $\pi(r) \subseteq E$ and $\pi(T) \subseteq E$, we get $\pi(T) \supseteq \pi(r)$. QED.

Using this proposition, we derive a simpler form of the likelihood given in Eq. (3.3). Let $\mathcal{S} = \{(\sigma_1^{\oplus}, \sigma_1^{\ominus}), \ldots, (\sigma_m^{\oplus}, \sigma_m^{\ominus})\}$ be the set of characteristic discontinuous edges generated by the reads in alignment $\mathcal{R}$. Let $\mathbf{d} = \{d_1, \cdots, d_m\}$ be the number of reads that map to each pair in $\mathcal{S}$. Using that distinct reads r_j and $r_{j'}$ with the same characteristic discontinuous edges $(\sigma^{\oplus}(r_j), \sigma^{\ominus}(r_j)) = (\sigma^{\oplus}(r_{j'}), \sigma^{\ominus}(r_{j'}))$ have the same likelihood in terms of Eq. 3.3, we have

$$\Pr(\mathcal{R} \mid \mathcal{T}, \mathbf{c}) = \prod_{j=1}^{n} \frac{1}{\sum_{b=1}^{k} c_b L_b} \sum_{i \in X(\mathcal{T}, \sigma_j^{\oplus}, \sigma_j^{\ominus})} c_i = \prod_{j=1}^{m} \left(\frac{1}{\sum_{b=1}^{k} c_b L_b} \sum_{i \in X(\mathcal{T}, \sigma_j^{\oplus}, \sigma_j^{\ominus})} c_i \right)^{d_j}. \quad \text{(C.22)}$$

Now, taking the logarithm yields

$$\begin{aligned} \log \Pr(\mathcal{R} \mid \mathcal{T}, \mathbf{c}) &= \sum_{j=1}^{m} d_j \left(\log \left(\frac{1}{\sum_{b=1}^{k} c_b L_b} \right) + \log \sum_{i \in X(\mathcal{T}, \sigma_j^{\oplus}, \sigma_j^{\ominus})} c_i \right) \\ &= -\sum_{j=1}^{m} d_j \left(\log \sum_{b=1}^{k} c_b L_b \right) + \sum_{j=1}^{m} \left(d_j \log \sum_{i \in X(\mathcal{T}, \sigma_j^{\oplus}, \sigma_j^{\ominus})} c_i \right) \\ &= \sum_{j=1}^{m} \left(d_j \log \sum_{i \in X(\mathcal{T}, \sigma_j^{\oplus}, \sigma_j^{\ominus})} c_i \right) - n \log \sum_{b=1}^{k} c_b L_b. \end{aligned} \quad \text{(C.23)}$$

The goal is the remove the second sum in the above equation, as it is convex and we are maximizing. In order to do so, we first prove the following lemma.

Lemma C.1. For any given scaling factor $\alpha > 0$, we have that $\log \Pr(\mathcal{R} \mid \mathcal{T}, \mathbf{c}) = \log \Pr(\mathcal{R} \mid \mathcal{T}, \alpha\mathbf{c})$.

Proof.

$$\log \Pr(\mathcal{R} \mid \mathcal{T}, \alpha\mathbf{c}) = \sum_{j=1}^{m} \left(d_j \log \sum_{i \in X(\mathcal{T}, \sigma_j^{\oplus}, \sigma_j^{\ominus})} \alpha c_i \right) - n \log \sum_{b=1}^{k} \alpha c_b L_b \tag{C.24}$$

$$= \sum_{j=1}^{m} \left(d_j \log \left(\alpha \sum_{i \in X(\mathcal{T}, \sigma_j^{\oplus}, \sigma_j^{\ominus})} c_i \right) \right) - n \log \alpha \sum_{b=1}^{k} c_b L_b \tag{C.25}$$

$$= \sum_{j=1}^{m} d_j \log \alpha + \sum_{j=1}^{m} \left(d_j \log \sum_{i \in X(\mathcal{T}, \sigma_j^{\oplus}, \sigma_j^{\ominus})} c_i \right) - n \log \alpha - n \log \sum_{b=1}^{k} c_b L_b \tag{C.26}$$

$$= n \log \alpha + \sum_{j=1}^{m} \left(d_j \log \sum_{i \in X(\mathcal{T}, \sigma_j^{\oplus}, \sigma_j^{\ominus})} c_i \right) - n \log \alpha - n \log \sum_{b=1}^{k} c_b L_b \tag{C.27}$$

$$= \sum_{j=1}^{m} \left(d_j \log \sum_{i \in X(\mathcal{T}, \sigma_j^{\oplus}, \sigma_j^{\ominus})} c_i \right) - n \log \sum_{b=1}^{k} c_b L_b \tag{C.28}$$

$$= \log \Pr(\mathcal{R} \mid \mathcal{T}, \mathbf{c}). \tag{C.29}$$

QED.

This enables us to prove the following lemma.

Lemma C.2. Let $D > 0$ be a constant, $\overline{c}_i(\mathbf{c}) = c_i D / \sum_{j=1}^{k} c_j L_j$ and $c_i(\overline{\mathbf{c}}) = \overline{c}_i / \sum_{j=1}^{k} \overline{c}_j$ for all $i \in [k]$. Then, $(\mathcal{T}, \mathbf{c} = [c_1(\overline{\mathbf{c}}), \ldots, c_k(\overline{\mathbf{c}})])$ is an optimal solution for Eq. (3.4) to (3.7) if

and only if $(\mathcal{T}, \overline{\mathbf{c}} = [\overline{c}_1(\mathbf{c}), \ldots, \overline{c}_k(\mathbf{c})])$ is an optimal solution for

$$\max_{\mathcal{T}, \overline{\mathbf{c}}} \sum_{j=1}^{m} d_j \log \sum_{i \in X(\mathcal{T}, \sigma_j^{\oplus}, \sigma_j^{\ominus})} \overline{c}_i \tag{C.30}$$

$$\text{s.t. } \pi(T_i) \text{ is an } \mathbf{s} - \mathbf{t} \text{ path in the segment graph } G \qquad \forall i \in [k], \tag{C.31}$$

$$\sum_{i=1}^{k} \overline{c}_i L_i = D, \tag{C.32}$$

$$\overline{c}_i \geq 0 \qquad \forall i \in [k]. \tag{C.33}$$

Proof. We will refer to the optimization problem in Eq. (3.4) to (3.7) as P and the optimization problem in Eq. (C.30)-(C.33) as Q. Further, we will refer to the objective function in Eq. (3.4) as $J(\mathcal{T}, \mathbf{c})$ and the objective function in (C.30) as $K(\mathcal{T}, \overline{\mathbf{c}})$. Observe that

$$\begin{aligned} K(\mathcal{T}, \overline{\mathbf{c}}) =& \log \Pr(\mathcal{R} \mid \mathcal{T}, \overline{\mathbf{c}}) + n \log \sum_{b=1}^{k} \overline{c}_b L_b \\ =& J(\mathcal{T}, \overline{\mathbf{c}}) + n \log \sum_{b=1}^{k} \overline{c}_b L_b, \end{aligned} \tag{C.34}$$

where the last equality uses (C.23).

(⇒) Let $(\mathcal{T}, \mathbf{c})$ be an optimal solution to problem P. We begin by showing that $(\mathcal{T}, \overline{\mathbf{c}})$ is a feasible solution to Q where $\overline{\mathbf{c}} = [\overline{c}_1(\mathbf{c}), \ldots, \overline{c}_k(\mathbf{c})]$. By definition of $\overline{c}_i(\mathbf{c})$, constraints Eq. (C.32) are satisfied. Hence, $(\mathcal{T}, \overline{\mathbf{c}})$ is a feasible solution to problem Q.

We now show that if $(\mathcal{T}, \mathbf{c})$ is an optimal solution to problem P, then $(\mathcal{T}, \overline{\mathbf{c}})$ is an optimal solution to problem Q. Let $(\mathcal{T}', \overline{\mathbf{c}}')$ be an optimal solution to problem Q. Then, by optimality of $(\mathcal{T}', \overline{\mathbf{c}}')$, we have

$$K(\mathcal{T}', \overline{\mathbf{c}}') \geq K(\mathcal{T}, \overline{\mathbf{c}}). \tag{C.35}$$

Let $\mathbf{c}' = [c_1(\overline{\mathbf{c}}'), \ldots, c_k(\overline{\mathbf{c}}')]$. Note that $\mathbf{c}'$ satisfies constraints in Eq. (5). Thus $(\mathcal{T}', \mathbf{c}')$ is a feasible solution to problem P. Since $(\mathcal{T}, \mathbf{c})$ is an optimal solution of P, we have

$$J(\mathcal{T}, \mathbf{c}) \geq J(\mathcal{T}', \mathbf{c}'). \tag{C.36}$$

Since $\mathbf{c}'$ and $\overline{\mathbf{c}}'$ only differ by a positive scaling factor $\alpha = 1/\sum_{i=1}^{k} \overline{c}'_i$, we use Lemma C.1 to get $J(\mathcal{T}', \mathbf{c}') = J(\mathcal{T}', \overline{\mathbf{c}}')$. Similar result holds for $\mathbf{c}$ and $\overline{\mathbf{c}}$, *i.e.* $J(\mathcal{T}, \mathbf{c}) = J(\mathcal{T}, \overline{\mathbf{c}})$. Applying

this to (C.36), we get

$$J(\mathcal{T}, \overline{\mathbf{c}}) \geq J(\mathcal{T}', \overline{c}'). \tag{C.37}$$

Using (C.32) and (C.34), we get

$$\begin{aligned}
& J(\mathcal{T}, \overline{\mathbf{c}}) && \geq J(\mathcal{T}', \overline{\mathbf{c}}') \\
\implies & K(\mathcal{T}, \overline{\mathbf{c}}) - n \log \sum_{b=1}^{k} \overline{c}_b L_b && \geq K(\mathcal{T}', \overline{\mathbf{c}}') - n \log \sum_{b=1}^{k} \overline{c}'_b L_b \\
\implies & K(\mathcal{T}, \overline{\mathbf{c}}) - n \log D && \geq K(\mathcal{T}', \overline{\mathbf{c}}') - n \log D \\
\implies & K(\mathcal{T}, \overline{\mathbf{c}}) && \geq K(\mathcal{T}', \overline{\mathbf{c}}').
\end{aligned} \tag{C.38}$$

Finally, using (C.35) and (C.38), we get $K(\mathcal{T}, \overline{\mathbf{c}}) = K(\mathcal{T}', \overline{\mathbf{c}}')$. Hence, $(\mathcal{T}, \overline{\mathbf{c}})$ is an optimal solution of Q.

(⇐) Let $(\mathcal{T}, \overline{\mathbf{c}})$ be an optimal solution to problem Q. We begin by showing that $(\mathcal{T}, \mathbf{c})$ is a feasible solution to P where $\mathbf{c} = [c_1(\overline{\mathbf{c}}), \ldots, c_k(\overline{\mathbf{c}})]$. By definition of $c_i(\overline{\mathbf{c}})$, constraints in Eq. (3.6) are satisfied. Hence, $(\mathcal{T}, \mathbf{c})$ is a feasible solution to problem P.

Next, we need to show that $(\mathcal{T}, \mathbf{c})$ is an optimal solution to problem P. Let $(\mathcal{T}', \mathbf{c}')$ be an optimal solution to problem P.

Then, from the optimality condition, we get

$$J(\mathcal{T}', \mathbf{c}') \geq J(\mathcal{T}, \mathbf{c}). \tag{C.39}$$

Let $\overline{\mathbf{c}}' = [\overline{c}_1(\mathbf{c}'), \ldots, \overline{c}_k(\mathbf{c}')]$. Note that $\overline{\mathbf{c}}'$ satisties constraint (C.32) and thus $(\mathcal{T}', \overline{\mathbf{c}}')$ is a feasible solution to problem Q. Using (C.34) and the fact that $(\mathcal{T}, \overline{\mathbf{c}})$ is an optimal solution of problem $\overline{P}$ we get

$$\begin{aligned}
& K(\mathcal{T}, \overline{\mathbf{c}}) && \geq K(\mathcal{T}', \overline{\mathbf{c}}') \\
\implies & J(\mathcal{T}, \overline{\mathbf{c}}) + n \log \sum_{b=1}^{k} \overline{c}_b L_b && \geq J(\mathcal{T}', \overline{\mathbf{c}}') + n \log \sum_{b=1}^{k} \overline{c}'_b L_b \\
\implies & J(\mathcal{T}, \overline{\mathbf{c}}) + n \log D && \geq J(\mathcal{T}', \overline{\mathbf{c}}') + n \log D \\
\implies & J(\mathcal{T}, \overline{\mathbf{c}}) && \geq J(\mathcal{T}', \overline{\mathbf{c}}').
\end{aligned} \tag{C.40}$$

Observe that $\mathbf{c}'$ and $\overline{\mathbf{c}}'$ only differ by a positive scaling factor $\alpha = D / \sum_{j=1}^{k} c'_j L_j$. Therefore, using Lemma C.1, we have $J(\mathcal{T}', \mathbf{c}') = J(\mathcal{T}', \overline{\mathbf{c}}')$. Similarly, for $\mathbf{c}$ and $\overline{\mathbf{c}}$, we have $J(\mathcal{T}, \mathbf{c}) =$

$J(\mathcal{T}, \overline{\mathbf{c}})$. Using this together with (C.40), we obtain

$$J(\mathcal{T}, \mathbf{c}) \geq J(\mathcal{T}', \mathbf{c}'). \tag{C.41}$$

Moreover, (C.39) and (C.41) simultaneously imply $J(\mathcal{T}, \mathbf{c}) = J(\mathcal{T}', \mathbf{c}')$. Hence, $(\mathcal{T}, \mathbf{c})$ is an optimal solution to problem P. QED.

Appendix D: Simulation Details

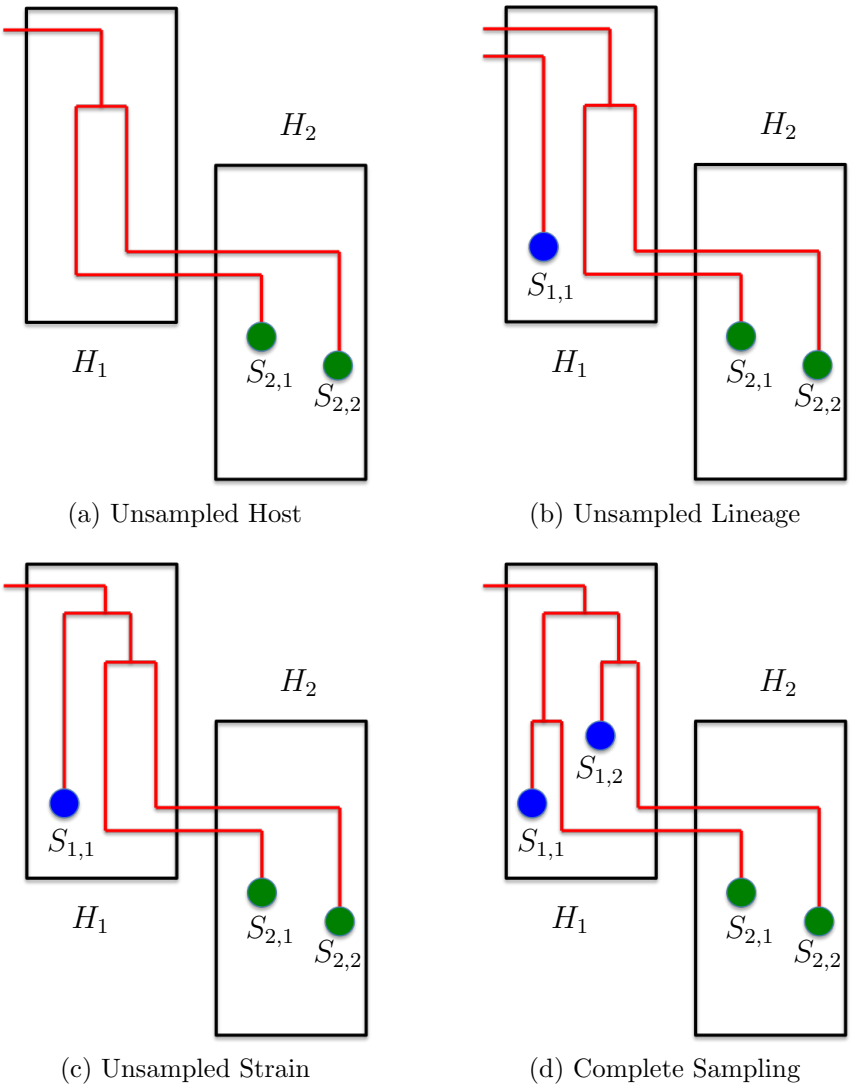
D.1 SAMPLING SCENARIOS IN AN OUTBREAK

The weak transmission bottleneck has some interesting implications for the sampling of the within-host diversity of the infected hosts. Fig. D.1 gives an overview, with schematic representations, of 4 different scenarios that can occur for real outbreaks.

D.2 SIMULATION PIPELINE FOR DISCONTINUOUS TRANSCRIPTION

Our simulations are based on a widely believed model of discontinuous transcription. Briefly, there are two competing models of discontinuous transcription for coronaviruses [172]. Both models agree that the RdRp jump is mediated by matching core-sequences (motifs) present in the TRSs in the viral genome. The only point of difference between the two models is whether discontinuous transcription occurs during the plus-strand synthesis or the minus-strand synthesis. The *negative-sense discontinuous transcription model* [173] proposes that the it is during the minus-strand synthesis that the RdRp performs discontinuous transcription. Transcription is initiated at the 3' end of the plus-strand RNA and the RdRp jumps to the TRS-L region when it reaches a TRS-B region adjacent to a gene, thereby generating a minus-strand subgenomic RNA. The minus-strand subgenomic RNA is then replicated by the RdRp to produce a plus-strand RNA which can be translated to a viral protein. Currently, this model is largely believed to be true due to the considerable experimental support from genetic studies detecting minus-strand subgenomic RNAs [174, 175, 176, 177, 178].

We now describe the procedure to simulate transcripts and their abundances following the negative-sense model of discontinuous transcription for a given segment graph. The model is parameterized by the function $p : E \to [0, 1]$. According to the *negative-sense discontinuous transcription* model, the transcription process is modeled as an $\mathbf{t} - \mathbf{s}$ walk in the reverse graph $\bar{G}$ where the direction of each original edge is reversed. At each node the RdRp randomly chooses an outgoing edge to traverse in the reverse graph $\bar{G}$ (which would be an incoming edge to the node in the original graph G) where the probabilities are given by the function p. Hence, the corresponding constraint on p under the negative-sense discontinuous transcription model is $\sum_{e \in \delta^-(\mathbf{v})} p(e) = 1$. The probabilities are drawn from a Dirichlet distribution with concentration parameter α set to 10 for edges that are present in the path corresponding to any of the canonical transcripts and 1 otherwise. This is done



(a) Unsampled Host

(b) Unsampled Lineage

(c) Unsampled Strain

(d) Complete Sampling

Figure D.1: **Schematic representation of different sampling scenarios during an outbreak.** Different hosts H_1 and H_2 are represented by rectangular boxes and the samples taken from the hosts are indicated by blue or green circles inside the boxes respectively. Red lines represent the evolution of pathogen lineages. Different scenarios described are (a) *Unsampled Host* scenario where host H_1 is not sampled even though it is part of the outbreak and infects H_2 with multiple strains (b) *Unsampled Lineage* where even though host H_1 is sampled with sample $S_{1,1}$, the lineage that passes two strains into host H_2 remains unsampled (c) *Unsampled Strain* scenario where the host H_1 is sampled and the right lineage is also sampled however the two strains that are transmitted to host H_2 are not sampled (d) *Complete Sampling* scenario where there is no incomplete lineage sorting (ILS) and all the strains transmitted from H_1 to H_2 are sampled.

to ensure that canonical transcripts are generated with high enough abundance, making the simulations similar to real data.

The next step of our simulation pipeline is to generate transcripts $\mathcal{T}$ and their abundances $\mathbf{c}$ for the given segment graph. We simulate the transcription process by generating 100,000 $\mathbf{s}-\mathbf{t}$ paths on the segment graph and report the number of unique paths/transcripts $\mathcal{T}$ and their abundances $\mathbf{c}$. We repeat this process to generate 5 independent sets of transcripts and abundances for the positive and the negative model each. Figure 3.3b shows the number of transcripts generated from each simulation using the negative-sense discontinuous transcription model. To contrast, the total number of $\mathbf{s}-\mathbf{t}$ paths in the underlying segment graph is 3440.

Once the transcripts are generated, the next step in our pipeline is to simulate the generation and sequencing of RNA-seq data. We use `polyester` [79] for this step as it allows the user to provide the number of reads generated from each transcript. For a given total number n of reads, the number of reads generated from transcript T_i is given by $nc_iL_i/\sum_{j=1}^{k} c_jL_j$ where L_i is the length of the transcript T_i. We use the default parameters for read length ($\ell = 100$) and fragment length distribution (Gaussian with mean $\mu_r = 250$ and standard deviation $\sigma_r = 25$) to generate 3,000,000 reads. For each set of transcript and abundances generated in the previous step of the pipeline, we simulate 5 replicates of the sequencing experiment.

The final step of the simulation pipeline is to align the generated reads to the reference genome NC_045512.2 using `STAR` [71]. The resulting BAM file serves as the input for the transcription assembly methods. To summarize, we generated 5 independent pairs $(\mathcal{T}, \mathbf{c})$ of transcripts and abundances under the negative-sense discontinuous transcription model. For each pair $(\mathcal{T}, \mathbf{c})$ we run 5 simulated sequencing experiments using `polyester` [79]. Therefore, we generated a total of $5 \times 5 = 25$ simulated instances.

D.3 SIMULATION DETAILS FOR DOUBLETD

D.3.1 Simulation Setup

We generate variant $\mathbf{V}$ and total read counts $\mathbf{C}$ for 500 *in silico* droplets as follows. First, we evolve 10 genotypes under an evolutionary model that incorporates CNAs and SNVs (detailed below) and use a symmetric Dirichlet distribution to obtain clonal abundances (concentration parameter $\alpha = 2$). The minimum allowable clonal abundance for any genotype was 0.02, which we enforce using rejection sampling. We vary the number of SNVs $m \in \{10, 50, 100\}$. Next, we decided for each droplet whether it is a doublet with proba-

bility $\delta \in \{0.1, 0.2, 0.4\}$. Depending on the outcome, we randomly sample one or two cells from the 10 genotypes to comprise the droplet. We also vary the mean sequencing coverage $c \in \{10, 50, 100\}$ and ADO probability $\beta \in \{0.0, 0.05, 0.25\}$. To draw total reads $\mathbf{C}$, we use a negative binomial distribution parameterized by the mean sequencing coverage c and a dispersion of 5. The copy error rate was fixed at $\alpha_{\text{fp}} = \alpha_{\text{fn}} = 0.001$ for all experiments in accordance with [106]. Finally, we set the beta-binomial precision to $s = 15$ — a simulation regime that matches the Tapestri platform by Mission Bio (Section 3.2 as well as MDA-based single-cell DNA sequencing (Section 3.3). Using the drawn total read counts $\mathbf{C}$ and the above parameters, we draw variant read counts $\mathbf{V}$. Each combination of simulation parameters was replicated with five different random number generator seeds, amounting to a total of 405 experiments.

D.3.2 Evolutionary Model

We now describe the model that we used to simulate the evolutionary history of $k = 10$ genotypes comprised of both CNAs and SNVs. Using Prüfer sequences [179], we begin by drawing a labeled tree T comprised of 10 nodes uniformly at random. For each of the m SNVs, we decide whether that mutation will undergo copy-neutral loss of heterozygosity (CN-LOH) with a probability of 0.1 (as described above). Then, we uniformly assign each mutation to a node of T to generate mutation clusters when $m > k$. For each non CN-LOH mutation, we decide with probability γ if that mutation will undergo a CNA event. These define a set of CNA events. For the set of CNA events, we then decide if that event is a loss with probability ℓ or a gain otherwise. For any gain events, we determine the number of copies gained by drawing a number uniformly between 1 and the max number of copies (3 for base simulations and 5 for our extreme CNA scenario). Next, we randomly assign CNA events to nodes of the tree T. We then generate the set of k genotypes by evolving SNV events (het or hom) and CNV events down the tree. We start with a pair (ω_i, ρ_i) representing the number of variant alleles and reference alleles of each SNV locus $i \in [m]$. At each node, we update this genotype in accordance with the events encountered on the path from the root to the node. When applying CNA events, if the current number of mutated alleles is non-zero, we select either the mutated or reference allele to undergo the CNA event. Otherwise, it is applied to the reference allele. If an SNV and CNA are introduced at the same time, we randomly determine which event to apply first.

D.4 SIMULATION DETAILS FOR PARSIMONIOUS CLONE TREE RECONCILIATION

We perturb the proportion matrices U_1 and U_2 by introducing noise following a user-defined level $h \in [0, 1]$. For each sample $p \in [m]$, let $\mathbf{u}_p^{(1)} = [u_{p,i}^{(1)}]$ for $i \in [n_1]$ and $\mathbf{u}_p^{(2)} = [u_{p,j}^{(2)}]$ for $j \in [n_2]$. The perturbed proportions $\bar{\mathbf{u}}_p^{(1)}$ and $\bar{\mathbf{u}}_p^{(2)}$ are drawn from the following distributions

$$\bar{\mathbf{u}}_p^{(1)} \sim (1-h)\mathbf{u}_p^{(1)} + h\text{Dir}(\mathbf{1}_{n_1}), \quad \forall p \in [m], \tag{D.1}$$

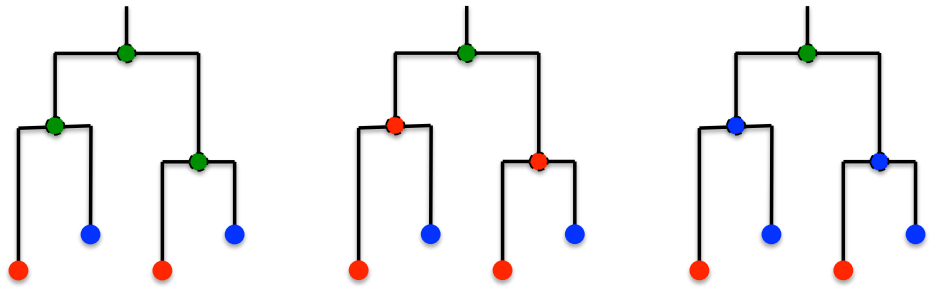
$$\bar{\mathbf{u}}_p^{(2)} \sim (1-h)\mathbf{u}_p^{(2)} + h\text{Dir}(\mathbf{1}_{n_2}), \quad \forall p \in [m]. \tag{D.2}$$

The resulting proportion matrices are $\bar{U}_1 = [\bar{u}_{p,i}^{(1)}]$ for $p \in [m], i \in [n_1]$ and $\bar{U}_2 = [\bar{u}_{p,j}^{(2)}]$ for $p \in [m], j \in [n_2]$. Note that when noise level $h = 0$, we have $\bar{U}_1 = U_1$ and $\bar{U}_2 = U_2$. Also, for any $h \in [0, 1]$, the matrices $\bar{U}_1$ and $\bar{U}_2$ satisfy the conditions laid out in the definition of proportion matrices (Definition 5.2).

Appendix E: Supplementary Results

E.1 MULTIPLE SOLUTIONS TO THE DTI PROBLEM

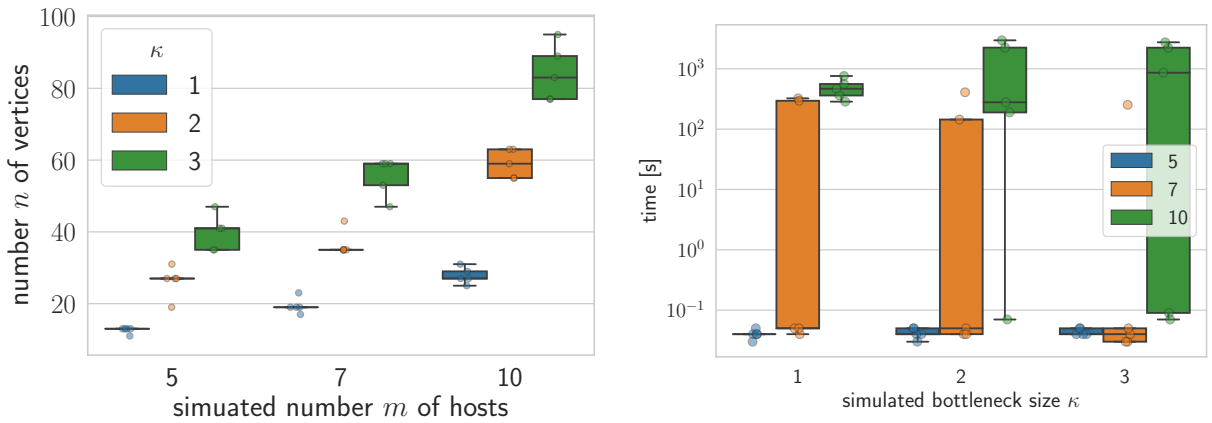
Fig. E.1 shows all the feasible solutions to the representative DTI problem described in the Fig. 2.1.



(a) (b) (c)

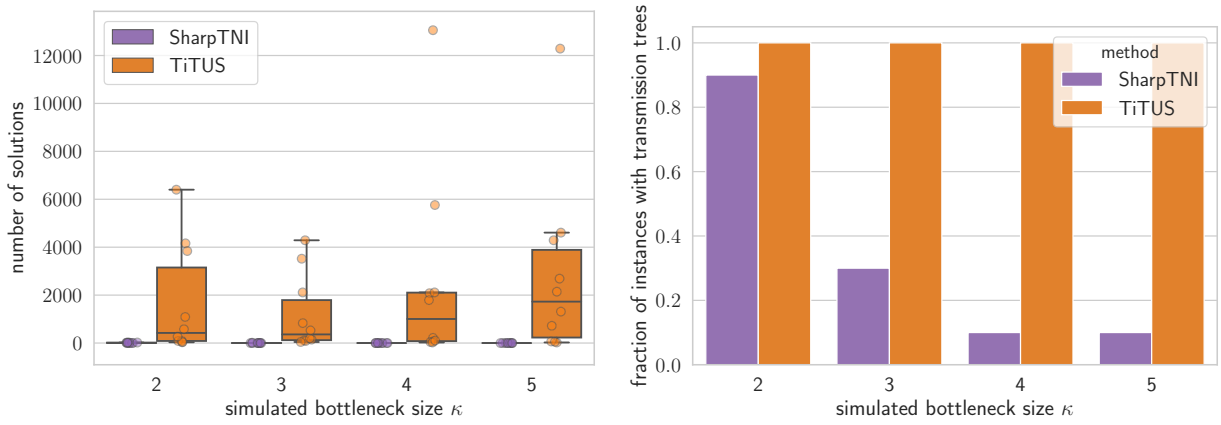
Figure E.1: The timed phylogeny shown in Fig. 2.1 has 3 possible vertex labeling solutions.

E.2 ADDITIONAL SIMULATION RESULTS FOR THE DTI PROBLEM



(a) (b)

Figure E.2: (a) The number of vertices n in the timed phylogeny T for increasing number m of simulated hosts and bottleneck size κ. (b) Time taken to generate 100,000 uniformly sampled solutions to the DTI problem using TiTUS for increasing values of simulated bottleneck size κ.



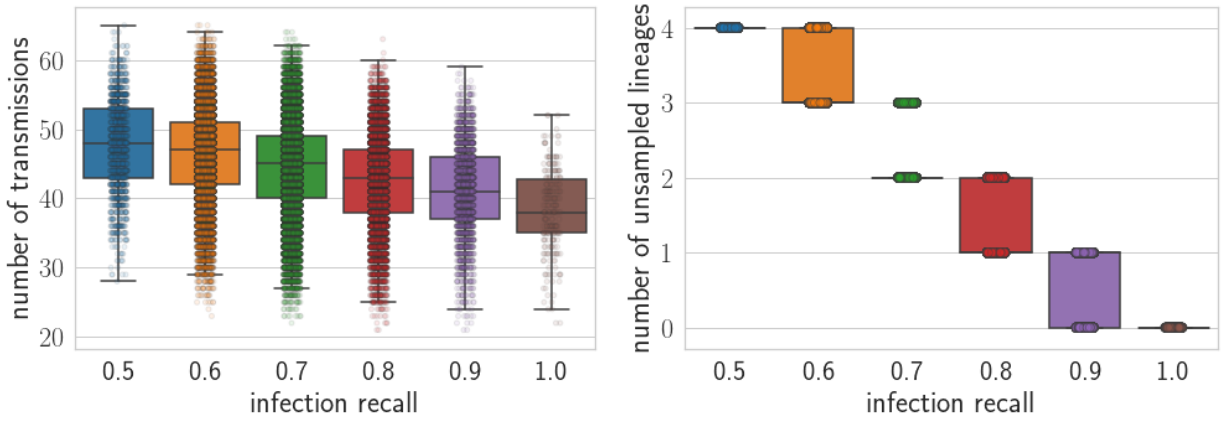
(a) (b)

Figure E.3: Comparison of SharpTNI and TiTUS on simulated, partially sampled outbreaks. Transmissions in simulated instances followed a tree-like pattern (*i.e.* direct transmission) but not all strains within a host were sampled. (a) The number of solutions where the vertex labeling induces a transmission tree (rather than a general graph) for increasing bottleneck size κ. (b) The fraction of simulation instances for which each method identified a transmission tree, for increasing values of simulated bottleneck size κ.

E.3 ADDITIONAL HIV DATA ANALYSIS AND IMPLEMENTATION DETAILS

host	transmission window	known infector	latest sample time	entry time	removal time
A	? - 14/05/90	B	7/11/05	$\tau(r(T))$	7/11/05
F	01/02/95 - 02/08/95	A	19/09/05	01/02/95	19/09/05
G	16/01/02 - 16/04/02	F	16/04/02	16/01/02	16/04/02
H	29/06/95 - 24/07/95	B	25/05/98	29/06/95	25/05/98
I	01/02/93 - 28/04/93	B	06/10/99	01/02/93	06/10/99
C	23/09/93 - 10/01/94	B	15/12/03	23/09/93	15/12/03
D	16/03/95 - 01/07/95	C	24/03/03	16/03/95	24/03/03
L	23/09/93 - 12/03/06	C	24/03/06	23/09/93	24/03/06
E	15/06/00 - 01/02/01	C	22/02/06	15/06/00	22/02/06
K	01/06/04 - 15/09/04	E	30/09/04	01/06/04	30/09/04

Table E.1: This table shows the epidemiological information provided in the HIV dataset [1]. The transmission window of a host is the expected time-interval during which the host was infected.



(a) (b)

Figure E.4: (a) Transmission number and (b) number of unsampled lineages of all the solutions generated using TiTUS on the HIV dataset vs different infection recall values.

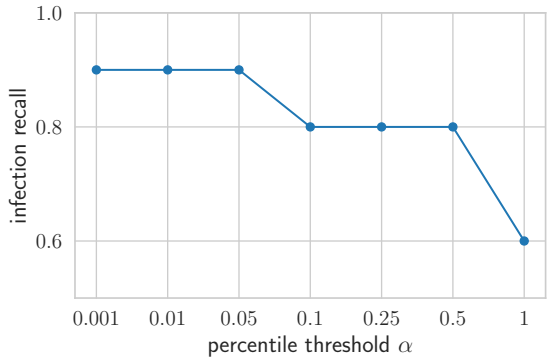


Figure E.5: The infection recall of the consensus transmission tree for solutions sampled using TiTUS on the HIV dataset for increasing values of the percentile threshold α.

E.4 HUMAN GENE TRANSCRIPT ASSEMBLY RESULTS

We evaluate the performance of Jumper, Scallop and StringTie on simulated samples of the human gene FAS as well. This gene is located on the long arm of chromosome 10 in humans and encodes the Fas cell surface receptor which leads to programmed cell death if it binds its ligand (Fas ligand). The FAS gene has 15 exons, yielding the following seven isoforms via alternative splicing (`https://www.uniprot.org/uniprot/P25445`).

1. P25445-1 with length of 335aa
   `https://useast.ensembl.org/Homo_sapiens/Transcript/Summary?db=core;g=ENSG00000026103;r=10:88990731-89014619;t=ENST00000652046`

2. P25445-2 with length of 103aa
   https://uswest.ensembl.org/Homo_sapiens/Transcript/Summary?db=core;g=ENSG00000026103;r=10:88990731-89014619;t=ENST00000484444

3. P25445-3 with length of 86aa
   https://uswest.ensembl.org/Homo_sapiens/Transcript/Summary?db=core;g=ENSG00000026103;r=10:88990731-89014619;t=ENST00000479522

4. P25445-4 with length of 149aa
   https://uswest.ensembl.org/Homo_sapiens/Transcript/Summary?db=core;g=ENSG00000026103;r=10:88990731-89014619;t=ENST00000494410

5. P25445-5 with length of 132aa
   https://uswest.ensembl.org/Homo_sapiens/Transcript/Summary?db=core;g=ENSG00000026103;r=10:88990731-89014619;t=ENST00000492756

6. P25445-6 with length of 314aa
   https://uswest.ensembl.org/Homo_sapiens/Transcript/Summary?db=core;g=ENSG00000026103;r=10:88990731-89014619;t=ENST00000357339

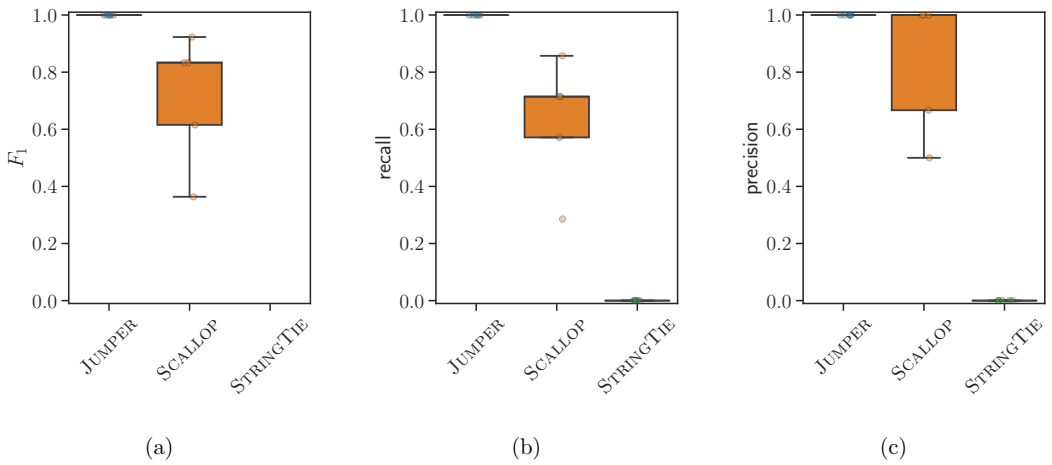
7. P25445-7 with length of 220aa
   https://uswest.ensembl.org/Homo_sapiens/Transcript/Summary?db=core;g=ENSG00000026103;r=10:88990731-89014619;t=ENST00000355279

The region between the first and the last exon span position 5001 to 30255 of the FAS gene. We used this region as the reference genome in our simulations[1].

We include the seven isoforms with equal proportion of 1/7 in the ground truth. We add a poly-A tail of length 85 at the end of the reference genome as well as each of the isoforms to emulate the transcription process. We use polyester [79] to simulate the sequencing of 35,000,000 paired-end reads of the sample with a Gaussian fragment length distribution with mean 250 and standard deviation of 25. We simulate 5 replicates of the sequencing experiment. The simulated reads are aligned to the selected region of the FAS gene using `STAR` [71]. The resulting BAM file serves as the input for the transcription assembly methods We evaluate the recall and precision of the three methods focusing on transcripts with abundance of more than 0.01. Figure E.6 shows that Jumper (median F1 score of 1) outperforms Scallop (median F1 score of 0.83) in terms of both recall and precision, while StringTie is not able to recall any of the 7 transcripts in the ground truth. We run the

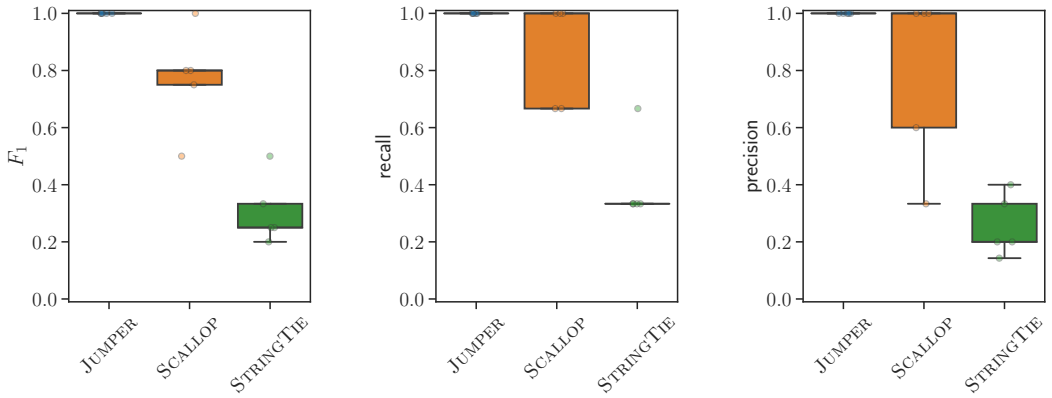
[1]NCBI reference sequence NG_009089.2: https://www.ncbi.nlm.nih.gov/nuccore/NG_009089.2?from=5001&to=30255&report=fasta

simulations again with only 3 of the isoforms, P25445-1, P25445-6 and P25445-7. Figure E.7 shows that StringTie is able to perform better with a median recall of 0.33, but still not as well as either Scallop (median recall of 1) or Jumper (median recall of 1).



(a) (b) (c)

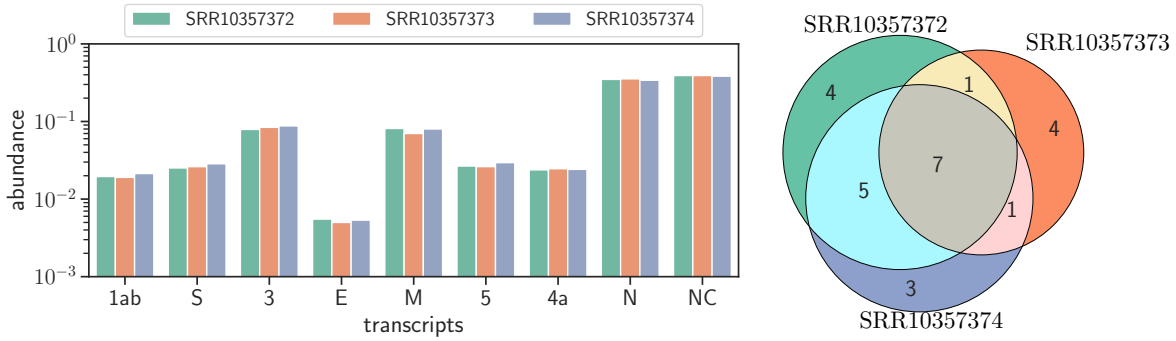
Figure E.6: Jumper outperforms Scallop and StringTie for all simulation instances of the FAS gene (on human chromosome 10) with all 7 isoforms of the gene in terms of F_1 score, recall and precision while maintaining a modest running time. (a) F_1 score (b) recall and (c) precision of the three methods for the simulated instances. The ground truth contained seven isoforms of the FAS gene with uniform relative abundances.



(a) (b) (c)

Figure E.7: Jumper outperforms Scallop and StringTie for all simulation instances of the FAS gene (on human chromosome 10) with only 3 isoforms (P25445-1, P25445-6 and P25445-7) in terms of F_1 score, recall and precision while maintaining a modest running time. (a) F_1 score (b) recall and (c) precision of the three methods for the simulated instances. The ground truth contained three isoforms of the FAS gene with uniform relative abundances.

E.5 TRANSCRIPT ASSEMBLY OF MERS-COV SAMPLES



(a) transcript abundances

(b) non-canonical transcripts

Figure E.8: JUMPER finds all canonical transcripts and some non-canonical transcripts from three MERS-CoV samples. (a) Abundance of the detected transcripts in the three samples, SRR10357372, SRR10357374 and SRR10357375. (b) A Venn diagram of the non-canonical transcripts reconstructed for each sample showing that there are 7 non-canonical transcripts that are present in all the three samples. Table E.2 shows the abundance of the 8 canonical transcripts that are present in all the samples and 14 non-canonical transcripts that are present in more than 1 sample.

MERS-CoV has a genome of length 30119 bp, and consists of 10 ORFs (1ab, S, 3, E, M, 4a, 4b, 5, 8b, N). We ran JUMPER on three published MERS-CoV samples [70], SRR10357372, SRR10357373 and SRR10357374, with a median coverage of 41,999, 36,663 and 45,235 respectively. These samples correspond to MERS-CoV infected Calu-3 cell lines [70]. Similar to previous analyses in this paper, we used `fastp` to trim the short reads (trimming parameter set to 10 nucleotides) and we aligned the resulting reads using `STAR` in two-pass mode. SCALLOP identified at most two canonical transcripts in each of the three samples (transcripts corresponding to ORF3 and ORF M in SRR10357372, ORF5 and ORF3 in SRR10357373, and ORF N in SRR10357374). We ran JUMPER with the 35 most abundant discontinuous edges in the segment graph and restrict our attention to transcripts identified by JUMPER that have more than 0.001 abundance as estimated by SALMON [69].

JUMPER reconstructs transcripts corresponding to all canonical ORFs of MERS-CoV in all the samples, except for ORF4b and ORF8b which are the only canonical ORFs that are not preceded by well supported TRS-B regions [90]. The most abundant transcript corresponds to ORF N (median abundance of 0.348), in line with the observations for SARS-CoV-2, while the least abundant canonical transcript encodes for protein E (median abundance of 0.0053). Figure E.8a shows, for each sample, the relative abundances of each canonical transcript as well as the total abundances of all non-canonical transcripts. Firstly, we observe that the

abundance of each canonical transcript is consistent across the three samples. Secondly, we see that all the three samples have high total abundance of non-canonical transcripts (median total abundance of 0.3908). Figure E.8b shows a Venn diagram for the non-canonical transcripts present in the three samples. We see out of the 25 distinct non-canonical transcripts, 7 are present in all the three samples and 14 are present in at least two of the samples. Table E.2 shows the abundance of the 8 canonical transcripts present in all the samples and the 14 non-canonical transcripts present in at least two samples. We will now describe the most abundant non-canonical transcripts in each sample.

The most abundant non-canonical transcript in samples SRR10357372 and SRR10357373 is 'NC8', which has a single discontinuous edge from position 1317 (5' end) to 29600 (3' end). The abundance of this transcript is 0.1019 in sample SRR10357372 and 0.1639 in sample SRR10357372, which is higher than all the canonical transcripts in both the samples except the transcript corresponding to ORF N. The 5' end of the discontinuous edge is in ORF1ab (nsp2 region) and the 3' end is in ORF N. Interestingly the most abundant non-canonical transcript in the third sample SRR10357374 is 'NC12', which has a single discontinuous edge with the same 3' end of 29600 while the 5' end is at position 1297 (also in the nsp2 region of ORF1ab). This transcript has abundance of 0.1486 in sample SRR10357374, higher than all the canonical transcripts in SRR10357374 except the transcript corresponding to ORF N, and 0.0483 in sample SRR10357372. We were not able to attribute the occurrence of transcripts NC8 and NC12 to matching motifs at the 5' and 3' ends of the discontinuous edges. Given the high abundance of these non-canonical transcripts in the sample, further investigation is required to ascertain their function.

E.6 ADDITIONAL RESULTS ON THE DTA PROBLEM

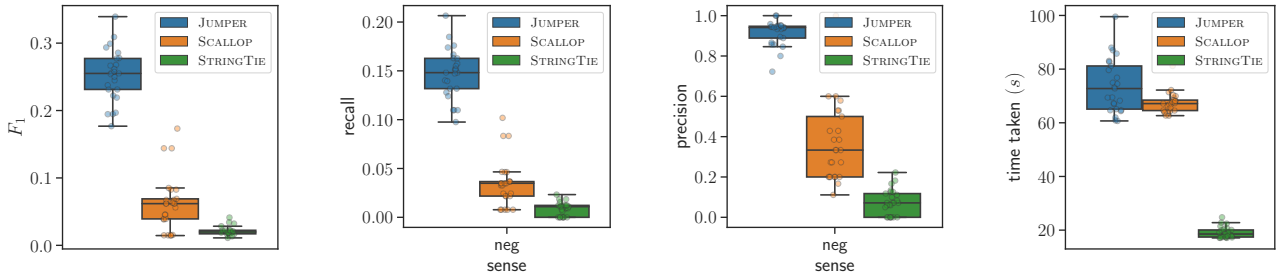
We have the following supplementary figures.

- Figure E.9 shows that Jumper outperforms Scallop and StringTie on simulated instances while maintaining a modest running time.
- Figure E.10 shows that Jumper outperforms Scallop and StringTie for varying values of thresholding parameter Λ.
- Figure E.11 shows that Jumper produces better recall and precision when compared to Scallop and StringTie for every simulation instance $(\mathcal{T}, \mathbf{c})$.
- Figure E.12 shows a core sequence potentially explaining a non-canonical discontinuous transcription that is conserved across *Sarbecovirus* species.

Transcript	Discontinuous Edges	SRR10357372	SRR10357373	SRR10357374	
1ab	-	0.0195	0.0190	0.0213	canonical
S	(59, 21402)	0.0251	0.0261	0.0284	
3	(59, 25518)	0.0789	0.0840	0.0876	
E	(61, 27582)	0.0055	0.0049	0.0053	
M	(58, 27834)	0.0812	0.0699	0.08	
5	(55, 26826)	0.0266	0.0261	0.0294	
4a	(59, 25840)	0.0237	0.0246	0.0241	
N	(53, 28536)	**0.3483**	**0.3542**	**0.34**	
NC1	(62, 28626)	0.0017	0.0016	0.0015	non-canonical
NC2	(65, 29106)	0.0043	0.0029	0.0026	
NC3	(61, 29503)	0.0016	0.0014	0.0015	
NC4	(61, 29582)	0.003	0.0027	0.0029	
NC5	(1727, 28983)	0.016	0.0169	0.0198	
NC6	(2343, 29204)	0.0736	0.1047	0.0575	
NC7	(7120, 24104)	0.0086	0.0088	0.0087	
NC8	(1317, 29600)	**0.1019**	**0.1639**	-	
NC9	(2333, 29203)	0.055	-	0.049	
NC10	(63, 680) (1727, 28983)	0.0019	-	0.0017	
NC11	(59, 21402) (24103, 27938)	0.0011	-	0.0011	
NC12	(1297, 29600)	0.0483	-	**0.1486**	
NC13	(64, 29105)	0.0011	-	0.001	
NC14	(2333, 29150)	-	0.0613	0.0363	

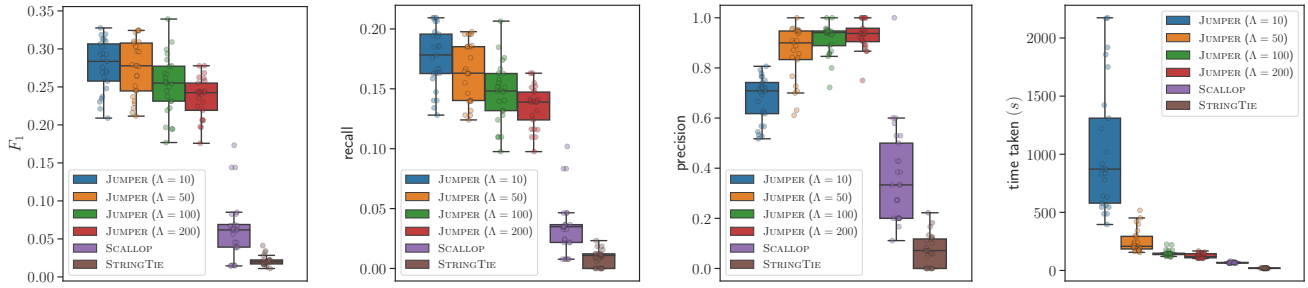
Table E.2: Abundance of 8 canonical transcript present in all three MERS-CoV samples and 14 non-canonical transcript present in more than 1 sample. The canonical and non-canonical transcripts with the highest abundance in each sample are highlighted. Figure E.8b shows the Venn diagram of all the transcripts in the solution.

- Figure E.13 shows an example of a supporting read for a transcript with two discontinuous edges.
- Figure E.14 shows that transcript X is supported in both long-read and short-read samples deposited in SRA.
- Figure E.15 shows the number of *supporting reads* with the 5' end mapping to the leader sequence in the short and long read sequencing data.
- Figure E.17 shows the abundances of the predicted transcripts by JUMPER in two SARS-CoV-1 infected samples.
- Table E.3 describes 18 transcripts (including 9 canonical transcripts) detected from SARS-CoV-2 infected samples with and without pre-treatment of ruxolitinib.
- Table E.4 shows summary of the results from the simulations.



(a) (b) (c) (d)

Figure E.9: Jumper outperforms Scallop and StringTie for all simulation instances in terms of F_1 score, recall and precision while maintaining a modest running time. (a) F_1 score (b) recall, (c) precision and (d) time taken by the three methods for the simulated instances.



(a) (b) (c) (d)

Figure E.10: Jumper outperforms Scallop and StringTie for varying values of thresholding parameter Λ. (a) F_1 score (b) recall, (c) precision and (d) time taken by the Jumper for different values of Λ compared to Scallop and StringTie on the simulated instances. As expected, the recall value drops for increasing Λ while the precision increases. We set the default value of Λ to 100 which incurs runtime comparable to Scallop while producing higher recall and precision solutions.

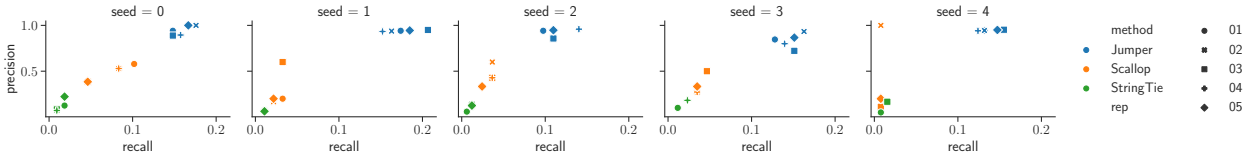
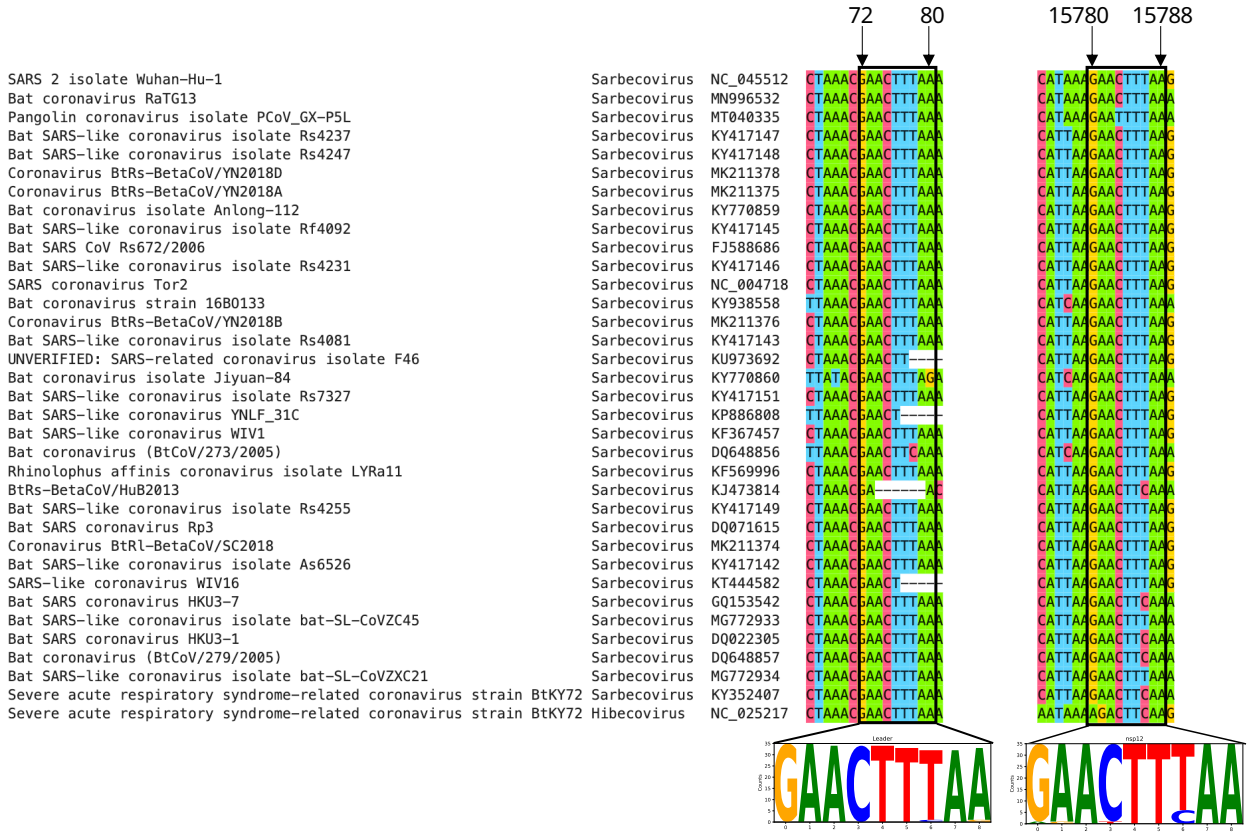
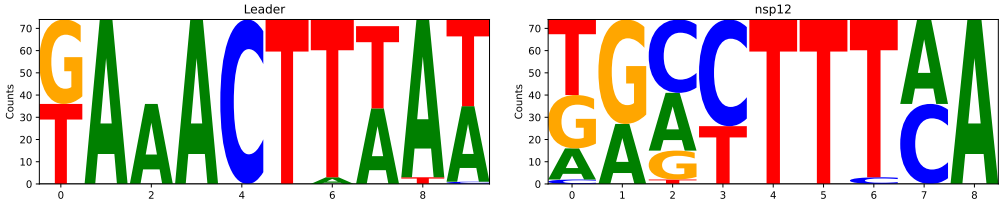


Figure E.11: While all three methods return consistent results when generating technical sequencing replicates, Jumper produces better recall and precision when compared to Scallop and StringTie for every simulation instance $(\mathcal{T}, \mathbf{c})$. Varying simulation instances $(\mathcal{T}, \mathbf{c})$ correspond to distinct panels. Each panel shows the recall and precision of the three methods for 5 sequencing experiments of the same simulated instance $(\mathcal{T}, \mathbf{c})$.

(a) Core sequence



(b) MSA and sequence logo with 34 *Sarbecoviruses* and 1 *Hibecovirus*.



(c) Sequence logo of 11 *Nobecoviruses*, 27 *Merbecoviruses* and 36 *Embecoviruses*

Figure E.12: The core sequence of transcript X is conserved within the *Sarbecovirus* subgenus but not in other subgenera of the *Betacoronavirus* genus. (a) Core sequence for the transcript X and X'. (b) Sequence logo for the positions 15780 to 15788 in SARS-CoV-2 genome built from the multiple sequence alignment of the leader sequence and ORF1ab of 34 *Sarbecovirus* and a *Hibecovirus*. (c) Sequence logo for positions 15780 to 15788 in SARS-CoV-2 genome built from multiple sequence alignment with the remaining subgenera of *Betacoronaviruses*.

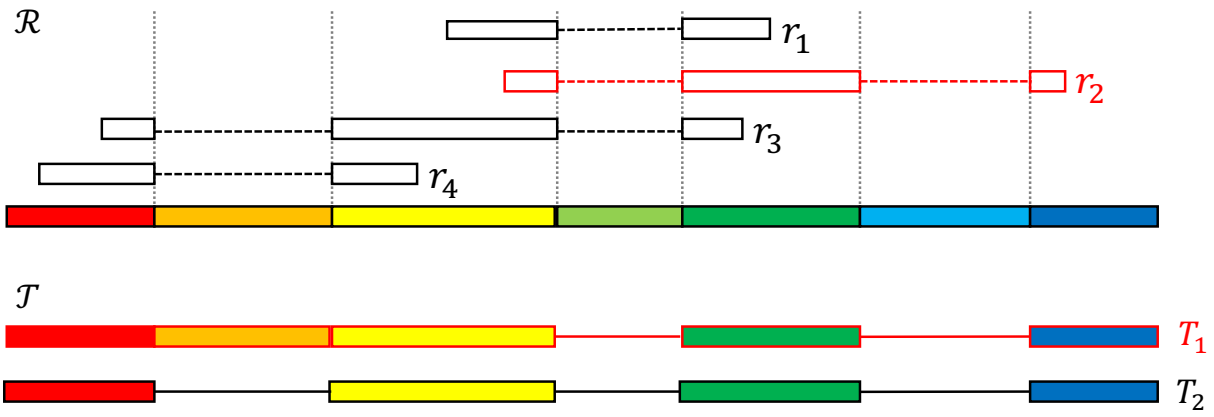
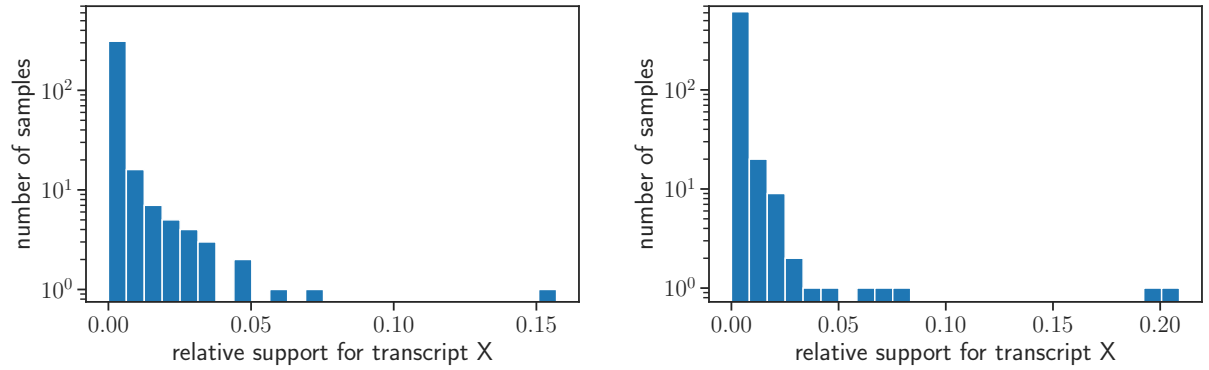


Figure E.13: A schematic showing an example of a supporting read for a transcript T_1 with $\sigma^{\oplus}(T_1) = 2$. Transcript T_1 is supported by r_2 because $\pi(r_2) = \pi(T_1)$ and $|\sigma^{\oplus}(r_1)| = |\sigma^{\oplus}(T_1)| = 2$. Reads r_1, r_3 and r_4 do not support T_1 since $|\sigma^{\oplus}(r_1)| < |\sigma^{\oplus}(T_1)|$ and $\pi(r_3), \pi(r_4) \nsubseteq \pi(T_1)$. No reads support T_2 since $|\sigma^{\oplus}(r_j)| < |\sigma^{\oplus}(T_2)|$ for all reads r_j.



(a) short-read samples

(b) long-read samples

Figure E.14: Transcript X has supporting reads in multiple independent publicly available samples of SARS-CoV-2 infected cells on SRA. Distribution of number of (a) short-read and (b) long-read SRA samples with varying proportion of leader-sequence spanning reads that support transcript X. All the short-read samples were aligned using `STAR` [71] while the long-read samples were aligned using `minimap2` [82]. In this plot we only consider samples with more than 100 reads that map to the leader-sequence (position 55 to 85 in the SARS-CoV-2 reference genome).

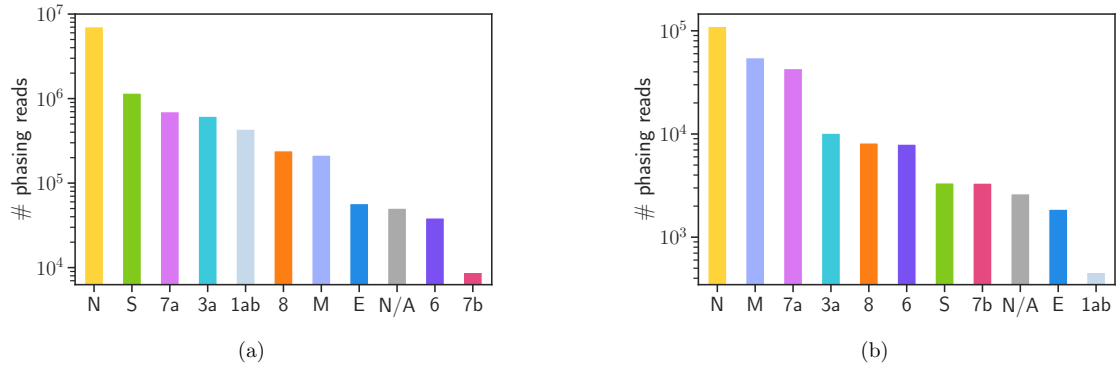


Figure E.15: Supporting phasing reads with 5' end mapping to the leader sequence in short and long-read sequencing samples of SARS-CoV-2 infected Vero cells [55]. Supporting phasing reads in the (a) short-read sequencing sample and (b) long-read sequencing sample.

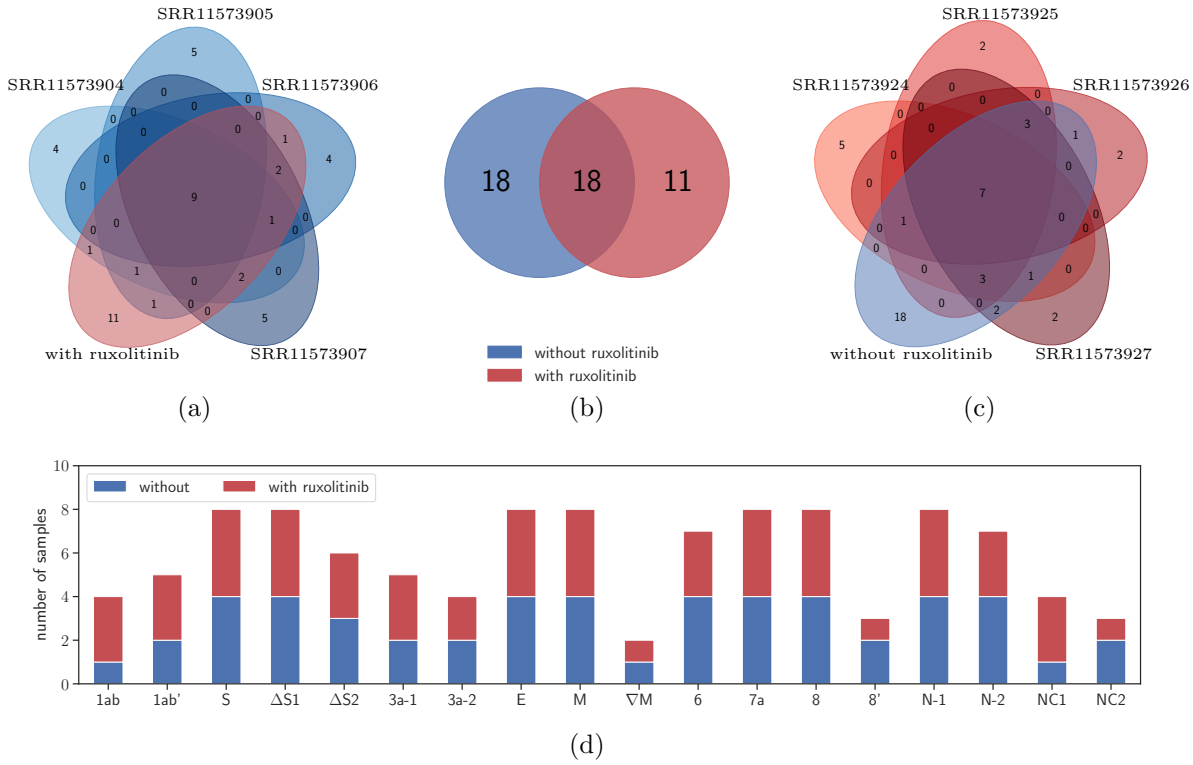


Figure E.16: JUMPER enables analysis of drug response of the virus in infected cells at the transcript level. (a) A Venn diagram of recalled transcripts from sample with and without treatment of ruxolitinib and a bar plots showing the number of samples containing each of the 18 common transcripts. Table E.3 described each of the 18 common transcripts. The transcripts are named based on the protein they yield, with ∇ indicating presence of out of frame deletions and Δ indicating in-frame deletions.

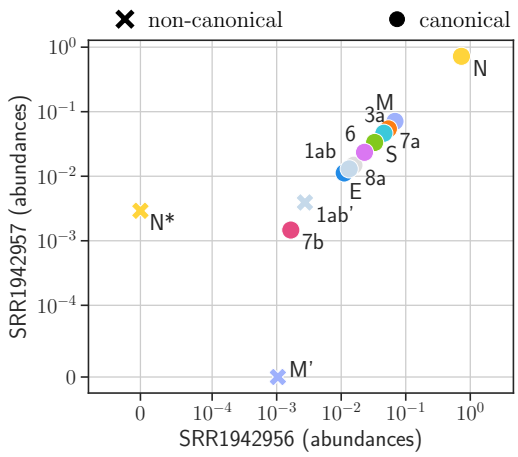


Figure E.17: Abundances of the canonical and non-canonical transcripts predicted by JUMPER are consistent in the two SARS-CoV-1 infected samples (SRR194256 and SRR194257). JUMPER predicts 10 canonical and 3 non-canonical transcripts across the two samples.

Transcript	Discontinuous Edges	Description
1ab	-	canonical transcript with no discontinuous edges
1ab'	(23593, 23630)	single discontinuous edge downstream of ORF1ab
S	(65, 21552)	single discontinuous edge from TRS-L to TRS-B of ORF S
ΔS1	(65, 21552) (23593, 23630)	single discontinuous edge from TRS-L to TRS-B of ORF S and an in-frame 12 amino-acid deletion overlapping the furin cleavage site
ΔS1	(65, 21552) (23593, 23615)	single discontinuous edge from TRS-L to TRS-B of ORF S and an in-frame 7 amino-acid deletion overlapping the furin cleavage site
3a-1	(65, 25381)	single discontinuous edge from TRS-L to TRS-B of ORF3a
3a-2	(66, 27385)	single discontinuous edge from TRS-L to TRS-B of ORF3a
E	(69, 26237)	single discontinuous edge from TRS-L to TRS-B of ORF E
M	(64, 26468)	single discontinuous edge from TRS-L to TRS-B of ORF M
∇M	(64, 26468) (26779, 26817) (28525, 28577)	single discontinuous edge from TRS-L to TRS-B of ORF M with an out of frame deletion with motifs 'CAATGGCTT' to 'CATTGCTT' and another downstream deletion within ORF N
6	(69, 27041)	single discontinuous edge from TRS-L to TRS-B of ORF6
7a	(66, 27385)	single discontinuous edge from TRS-L to TRS-B of ORF7a
8	(65, 27884)	single discontinuous edge from TRS-L to TRS-B of ORF8
8'	(65, 27884) (28270, 28970)	single discontinuous edge from TRS-L to TRS-B of ORF8 with a single deletion downstream of ORF8
N-1	(64, 28255)	single discontinuous edge from TRS-L to TRS-B of ORF N
N-2	(68, 28263)	single discontinuous edge from TRS-L to TRS-B of ORF N
NC1	(6001, 27376)	matching motif 'AGAGCAACCAAT' on the 5' and 3' ends of the jump
NC2	(731, 29307)	matching motif 'ATTTTCAA' to 'AATTTCAA'

Table E.3: 18 transcripts (including 9 canonical transcripts) detected from SARS-CoV-2 infected A549 cell line samples with and without pre-treatment of ruxolitinib. Figure 3.5 shows the abundances of these transcripts in the samples.

Simulation				Jumper			Scallop			StringTie		
seed	rep	can	non-can	TP		FP	TP		FP	TP		FP
				can	non-can		can	non-can		can	non-can	
0	1	14	94	7	9	1	7	4	8	2	0	14
0	2	14	94	8	11	0	7	2	8	1	0	13
0	3	14	94	7	11	2	4	1	8	1	0	11
0	4	14	94	6	9	2	7	2	8	1	0	13
0	5	14	94	7	11	0	4	1	8	1	0	7
1	1	14	78	3	13	1	3	0	12	2	0	13
1	2	14	78	4	11	1	2	0	10	1	0	13
1	3	14	78	3	16	1	3	0	2	1	0	12
1	4	14	78	3	11	1	2	0	8	0	0	16
1	5	14	78	4	13	1	2	0	8	1	0	15
2	1	14	150	5	11	1	3	1	8	1	0	16
2	2	14	150	4	14	3	5	1	4	2	0	15
2	3	14	150	5	13	3	5	1	8	2	0	13
2	4	14	150	7	16	1	5	1	8	2	0	16
2	5	14	150	4	14	1	3	1	8	2	0	14
3	1	14	72	4	7	2	3	0	8	1	0	9
3	2	14	72	6	8	2	3	0	4	0	0	8
3	3	14	72	7	6	4	4	0	8	0	0	20
3	4	14	72	4	8	3	3	0	8	2	0	9
3	5	14	72	4	9	2	3	0	6	0	0	4
4	1	14	115	4	13	1	1	0	4	1	0	19
4	2	14	115	5	12	1	1	0	0	0	0	12
4	3	14	115	6	14	1	1	0	8	2	0	10
4	4	14	115	6	10	1	1	0	4	0	0	16
4	5	14	115	6	13	1	1	0	4	0	0	12

Table E.4: Simulation results for the three methods Jumper, Scallop and StringTie. Each distinct value in the column 'seed' is a unique instance of $(\mathcal{T}, \mathbf{c})$ and each distinct value in the column 'rep' is a unique sequencing experiment for the given $(\mathcal{T}, \mathbf{c})$. (rep: replicate, can: canonical, non-can: non-canonical, TP: true positives, FP: false positives)

E.7 ADDITIONAL RESULTS ON THE DOUBLET DETECTION PROBLEM

E.7.1 Sensitivity of doubletD to Input Parameters

Our method takes several parameters as input (Fig. 2). The ADO probability β and the sequencing error rates $\alpha_{\text{fp}}, \alpha_{\text{fn}}$ are sequencing platform specific and typically known a priori. While the doublet probability δ is also typically known beforehand, we find that maximum likelihood is good criterion for estimating this parameter in case it is unknown. Specifically, varying the doublet probability in $\{0.01, 0.1, 0.2, 0.4, 0.7, 0.9\}$ and selecting value with maximum likelihood, we achieve similar precision and recall values as using the ground

truth doublet rate (Fig. E.21). Further, the likelihood function is maximized at or close to the simulated doublet rate indicating that it can be used as a reliable criteria for estimating the true doublet rate (Fig E.22).

doubletD estimates the beta-binomial precision parameter s and mutation probabilities μ from input data. We find that procedure outlined in Section A.6 for estimation the beta-binomial precision parameter s and mutation probabilities μ from input data results in only minor error with a deviation of 0.08, -0.11, 0.02, and 6.0 for μ_{wt}, μ_{het} and μ_{hom} (Fig. E.24a) and s (Fig. E.24a), respectively. We then fixed the inference doublet rate at the simulated rate and varied the beta precision by inputting half and twice the simulated beta precision into doubletD (Fig. E.23). We found that halving or doubling in inference beta precision parameter with respect to the simulated parameter had no significant impact when the simulated precision was high (1000). When the simulated precision was low (15), we noted that overestimation by utilizing double the simulated precision parameter did result in a small decrease in the median precision of 0.86 to 0.81 (Fig. E.23). Conversely, utilizing half the simulated rate resulted in improved precision without a reduction in recall. Thus, in the presence of uncertainty of this parameter, preference should be given to underestimation or lower values. In conclusion, we found that doubletD is resilient to variations in the user-inputted doublet rate parameter, especially in the range of typical experimental doublet rates (0.1-0.4). As the inference doublet rate increases beyond this range, precision is reduced since the threshold for calling an experiment a doublet is significantly lowered. The likelihood function calculated as a function of the predicted experimental labels is maximized at or close to the simulated doublet rate (Fig E.21).

E.7.2 Two Cell Line Mixture

We followed the procedure outlined in the vignette accompanying this dataset[2]. The one notable exception is that we used relaxed filtering criteria to identify additional loci for orthogonal doublet validation. The filtering criteria were as follows.

```
gt.filter=TRUE, gt.gqc = 30, \
gt.dpc = 10, gt.afc = 20, \
gt.mv = 50, gt.mc = 50, \
gt.mm = 0.5, gt.mask = TRUE
```

We extracted variant and total read counts from the AD and DP layers of the loom file for 1592 droplets. To identify a subset of 26 high-quality inference loci among the total

[2]https://support.missionbio.com/hc/en-us/articles/360045899834-Installation-instructions-for-tapestriR

number of 133 loci, we excluded loci that had a copy number greater than 2 (using the `compute_ploidy` function in the Tapestri R package).

Upon performing a preliminary dimensionality reduction (t-sne) and hierarchical clustering on the 'zygosity' (binning of VAFs into homozygous, heterozygous or wild type) of all 133 loci, we noted the existence of a third cluster composed of 23 droplets with a distinct genotype and mutually exclusive mutations from the other two cell lines. We excluded these droplets from our analyses. Further, we identified a set of 5 validation loci distinct from the 23 inference loci that were homozygous (hom) in one cell line cluster but wild type (wt) in the other. We use these loci to establish the ground truth assignment of droplets to the two cell lines, Raji and KG-1, and to compute the NCS score for each droplet.

E.7.3 Acute Lymphoblastic Leukemia Tumors

E.7.3.1 Preprocessing

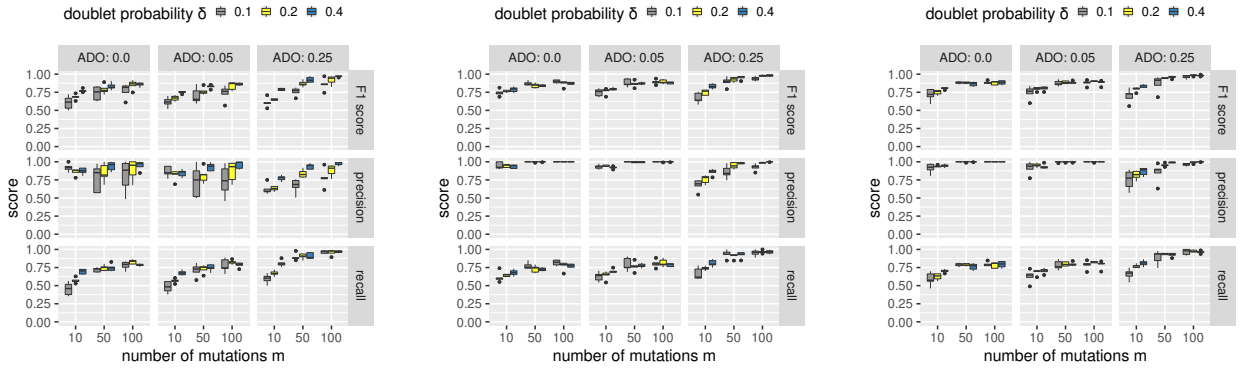
We utilized scDNA-seq data in the form of FASTQ files from the Sequence Read Archive database (accession no. SRP044380). After adaptor trimming (Trimmomatic), read alignment (bwa) to reference genome hg19 and PCR duplicate removal (Picard), we performed a pileup of the variant positions called by [95] to obtain the variant $\mathbf{V}$ and total read counts $\mathbf{C}$.

E.7.3.2 Doublet Detection

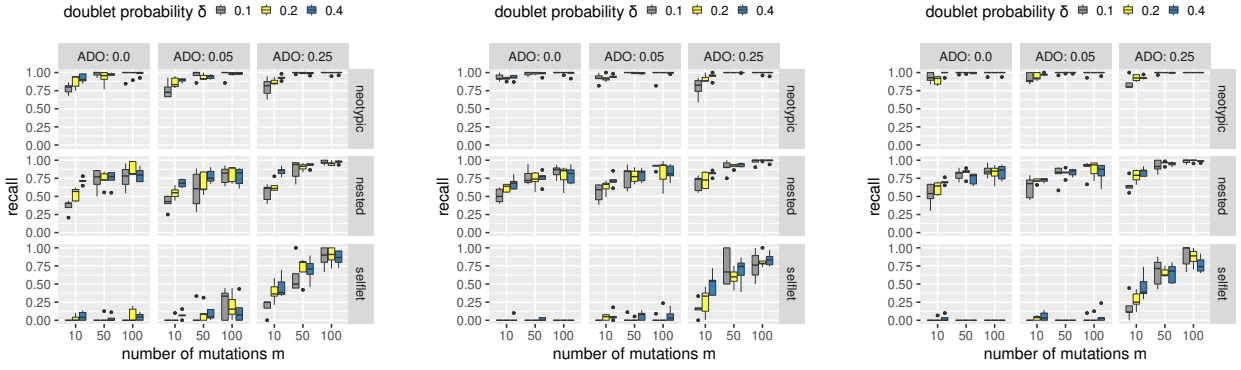
To mitigate the impact of CNAs on doublet detection, we only included loci that were heterozygous and had a median VAF $\in [0.45, 0.55]$. We ran doubletD directly on the total and variant read count data obtained from the pileup utilizing the ADO rates reported by [95]. Since no information was published on the expected doublet rate, we performed a grid search to obtain the maximum likelihood estimate of the prior doublet probability δ. Fig. E.29 shows that the identified doublets have distinct VAF distributions compared to the remaining singlet droplets, both on the set of inference loci (that met the heterozygosity filtering criterion) as well as an orthogonal set of holdout loci (that did not meet the filtering criterion). Table E.5 shows the statistics and results generated by applying doubletD on data from all the patients in the dataset. Section 3.3 shows detailed analysis of Patient 1.

patient	n	m'	m	mean coverage	ADO β	doubletD
1	243	20	14	14.3×	0.20	50 (0.21)
2	256	16	9	15.6×	0.18	22 (0.09)
3	266	48	31	9.1×	0.25	86 (0.32)
4	276	78	50	7.1×	0.24	92 (0.33)
5	225	105	59	7.0×	0.25	55 (0.24)
6	224	10	7	16.2×	0.18	46 (0.21)

Table E.5: Statistics and doublet detection results of an acute lymphoblastic leukemia cohort of six patients. From left to right, the table shows for each patient the number n of droplets, the number m' of loci identified by [95], the number m of loci that meet our filtering criteria, the mean coverage of the samples, the ADO rate reported by [95] and the number (and fraction) of doublets identified by doubletD.



(a) $c = 10\times$ (b) $c = 50\times$ (c) $c = 100\times$

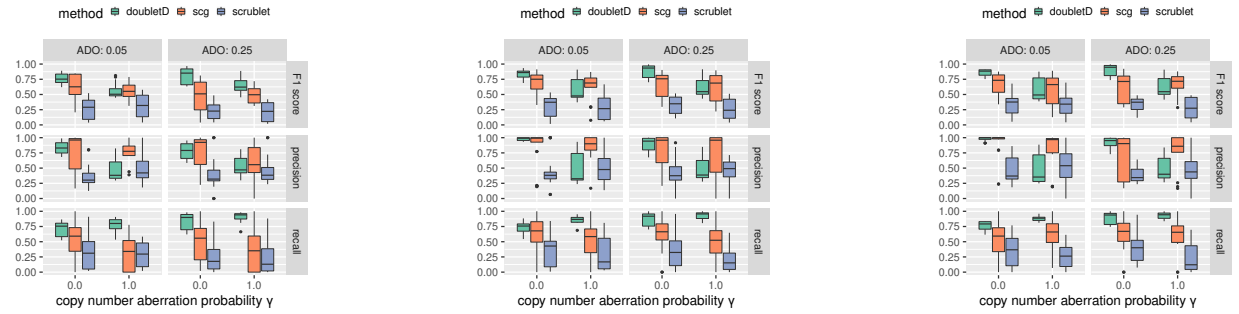


(d) $c = 10\times$ (e) $c = 50\times$ (f) $c = 100\times$

Figure E.18: F_1 score, recall and precision of doublet detection using DOUBLETD for varying mean read depths c, ADO rates β and doublet probabilities δ. All simulations are run without copy number aberrations $\gamma = 0$ and precision parameter $s = 15$.

E.7.3.3 Phylogeny Inference with PhISCS-B

PhISCS-B operates on a discretized input matrix that codes for the presence ('1') or absence ('0') of a mutation in a droplet as well as missing data ('?'). To discretize the input matrix, we used the binomial exact test to determine mutation status for each corresponding entry in the total and variant read count matrices $\mathbf{C}$ and $\mathbf{V}$ with a null error rate of 0.001 and a p-value of 10^{-6}. We provided PhISCS-B with the ADO rate $\beta = 0.2$ reported by [95] and a false positive rate of 0.001 that is typical for multiple displacement amplification (MDA) whole-genome amplification [180]. We imposed a maximum time limit of 3600 s.



(a) (b) (c)

Figure E.19: F_1 score, recall and precision of doublet detection using DOUBLETD on simulations without copy number losses and only gains. Results are shown for copy number aberrations probability $\gamma \in \{0, 1\}$ and ADO rates $\beta \in \{0.05, 0.25\}$. All simulations are run with doublet probability $\delta = 0.2$, mean read depth $c = 50\times$, number of mutations $m \in \{10, 50, 100\}$ and precision parameter $s = 15$.

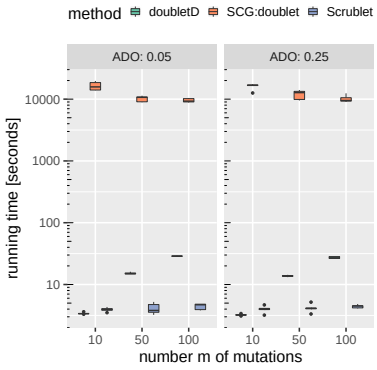
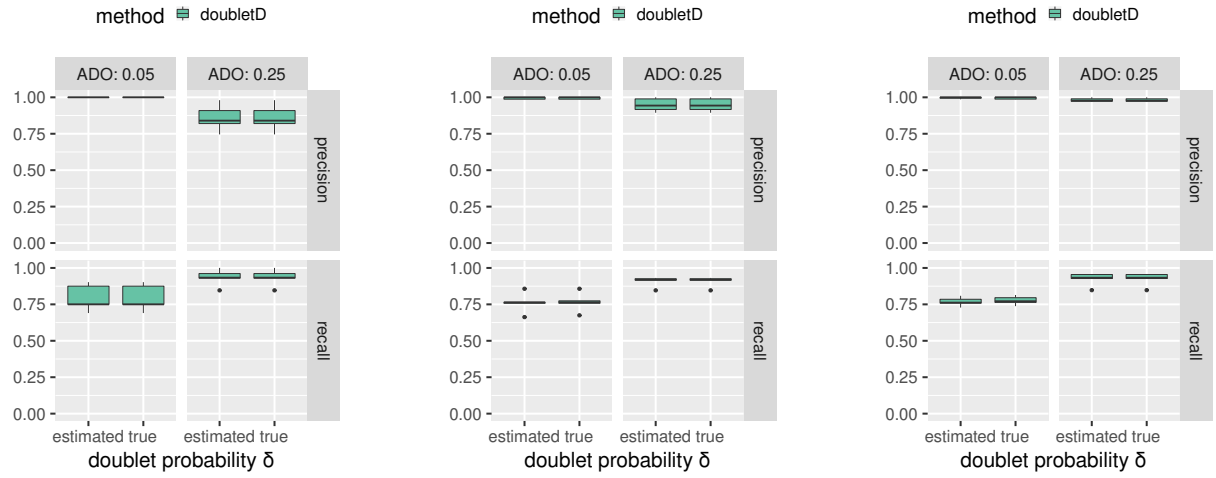
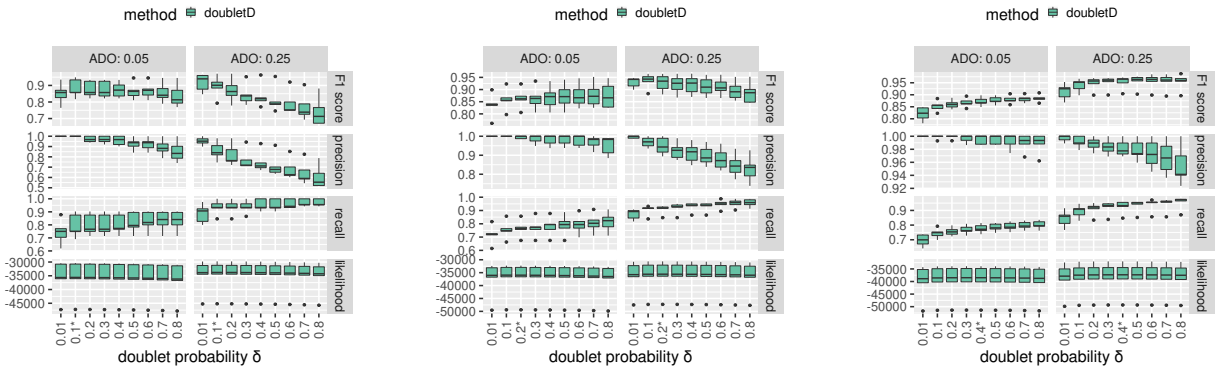


Figure E.20: Running time for doublet detection using DOUBLETD, SCG:doublet and SCRUBLET for simulations without CNAs ($\gamma = 0$) with varying number of mutations m. All simulations have doublet probability $\delta = 0.2$, mean read depth $c = 50\times$ and precision parameter $s = 15$.



(a) $c = 10\times$ (b) $c = 50\times$ (c) $c = 100\times$

Figure E.21: Precision and recall for doublet detection using DOUBLETD with maximum likelihood estimate of the doublet probability δ and the true doublet probability used in the simulations for varying ADO rates β and mean read depth c. Results are shown for simulations with doublet probability $\delta = 0.2$, number of mutations $m \in \{10, 50, 100\}$ and precision parameter $s = 15$.



(a) (b) (c)

Figure E.22: F_1 score, precision, recall and posterior likelihood of doublet detection using DOUBLETD with varying input doublet probability δ. The simulations are run with doublet probability $\delta = \{0.1, 0.2, 0.4\}$, number of mutations $m = 50$, coverage $50\times$ and precision parameter $s = 15$. Copy number aberration probability γ was set to 0.

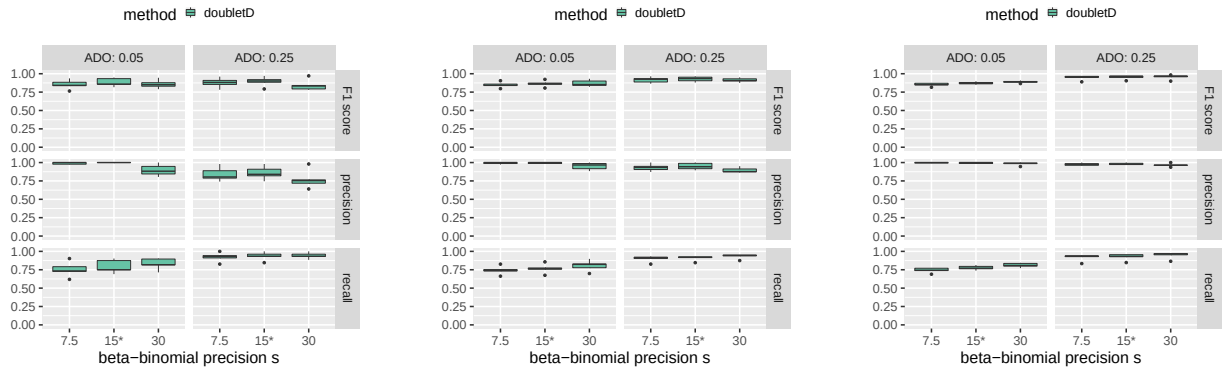


Figure E.23: F_1 score, precision, recall and posterior likelihood of doublet detection using DOUBLETD with varying input precision parameter s. The simulations are run with doublet probability $\delta = 0.2$, number of mutations $m \in \{10, 50, 100\}$ and precision parameter $s = 15$. Copy number aberration probability γ was set to 0.

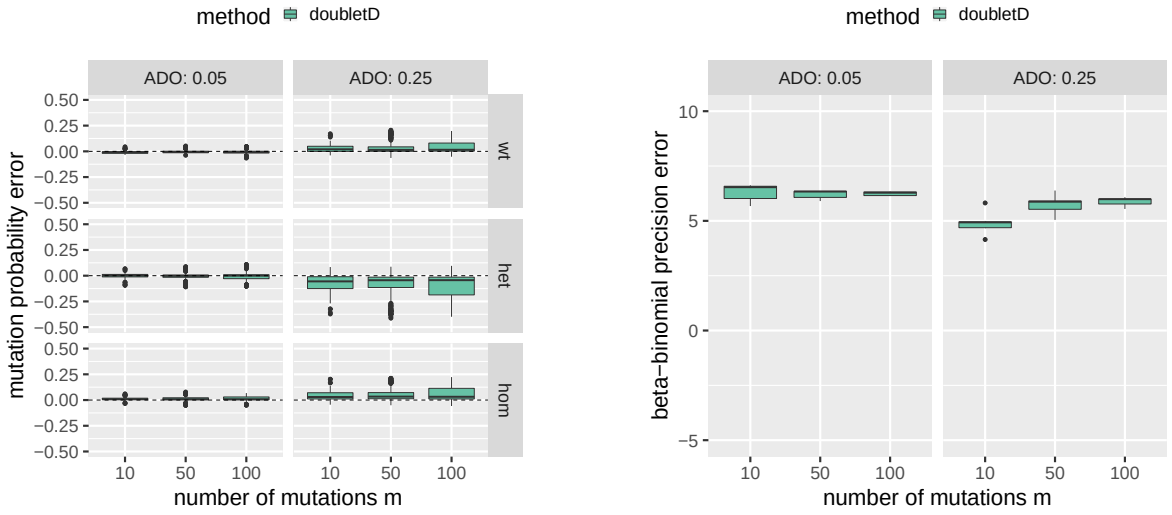


Figure E.24: Error in the estimation of (a) the mutation probabilities $\boldsymbol{\mu} = \{\boldsymbol{\mu}_{\mathbf{wt}}, \boldsymbol{\mu}_{\mathbf{het}}, \boldsymbol{\mu}_{\mathbf{hom}}\}$ and (b) the precision parameter s from simulated data with varying number of mutations m and ADO rates β. All simulations are run with doublet probability $\delta = 0.2$, copy number aberration probability $\gamma = 0$ and precision parameter $s = 15$.

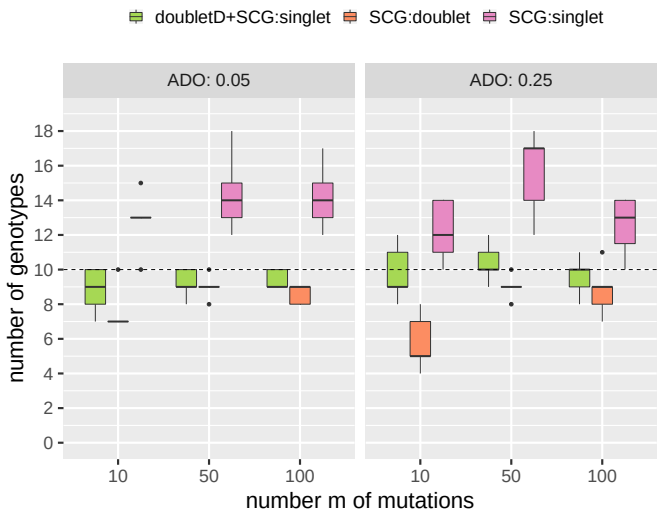


Figure E.25: Number of genotypes called by doubletD+SCG:singlet, SCG:doublet and SCG:singlet on simulations with varying number of muations m and ADO rates β. All simulations have doublet probability $\delta = 0.2$, mean read depth $c = 50\times$, precision parameter $s = 15$ and copy number aberration $\gamma = 0$.

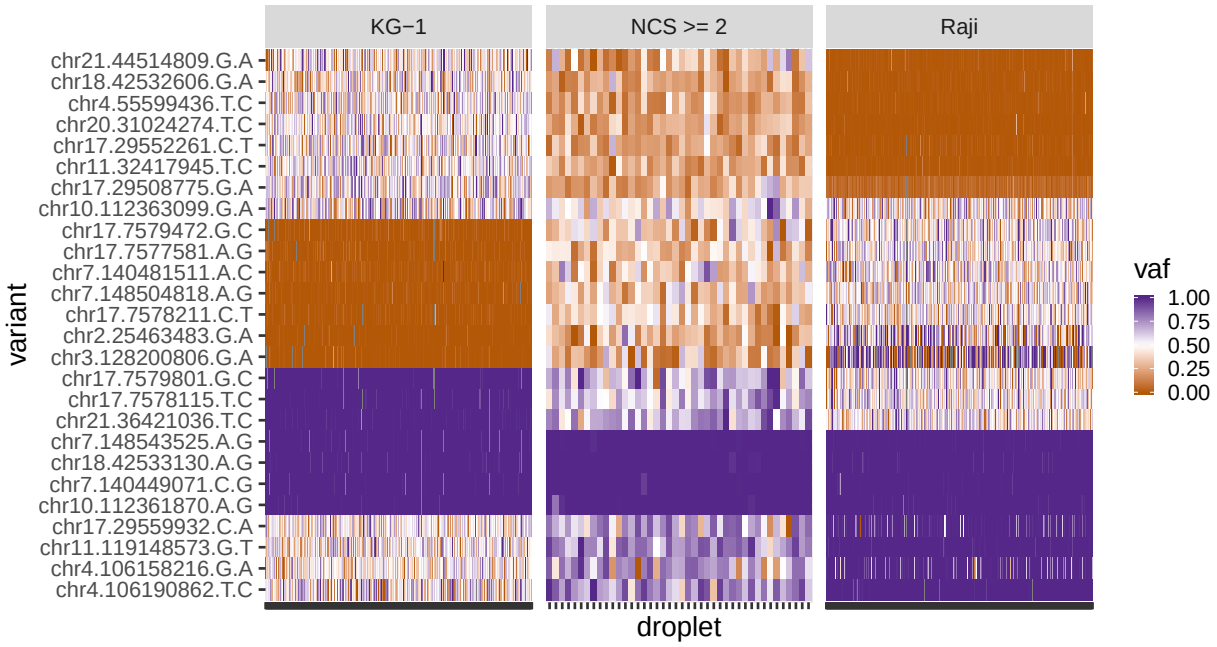
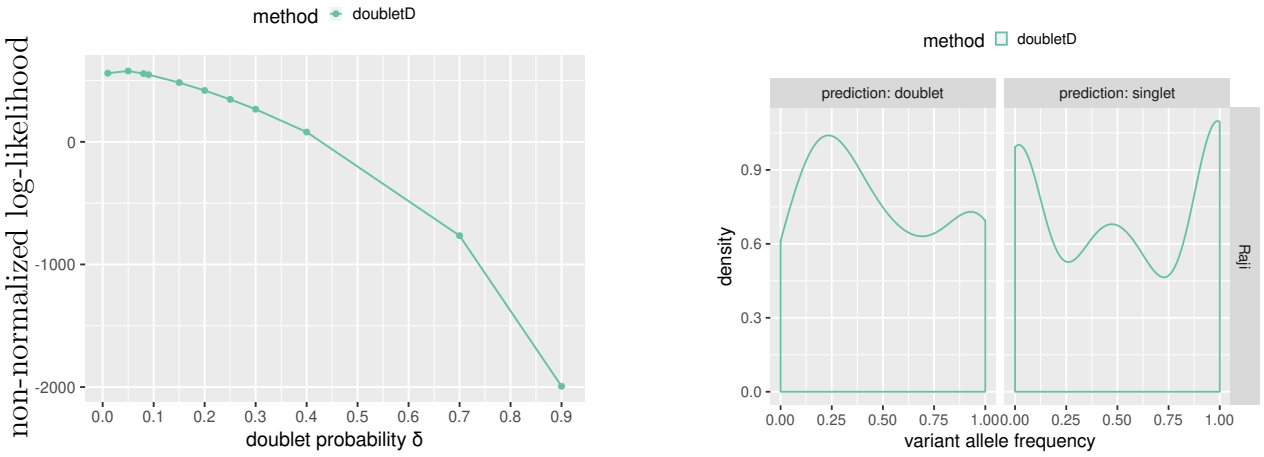
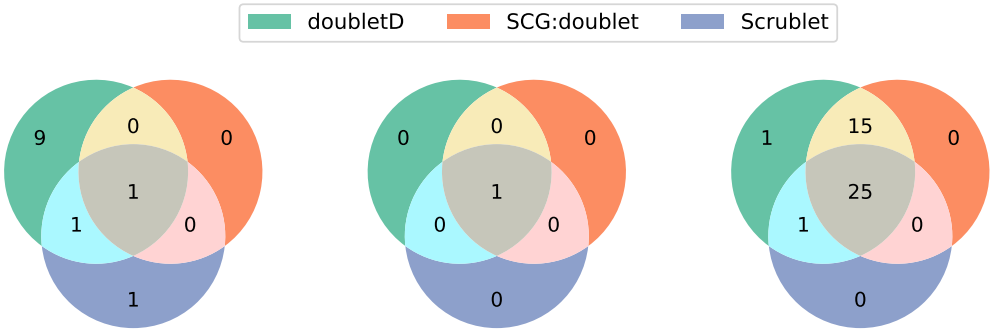


Figure E.26: Heatmap showing the observed variant allele frequency (VAF) of cell line droplets categorized by cell line or droplets with a neotypic doublet confidence score (NCS $\geq$ 2).



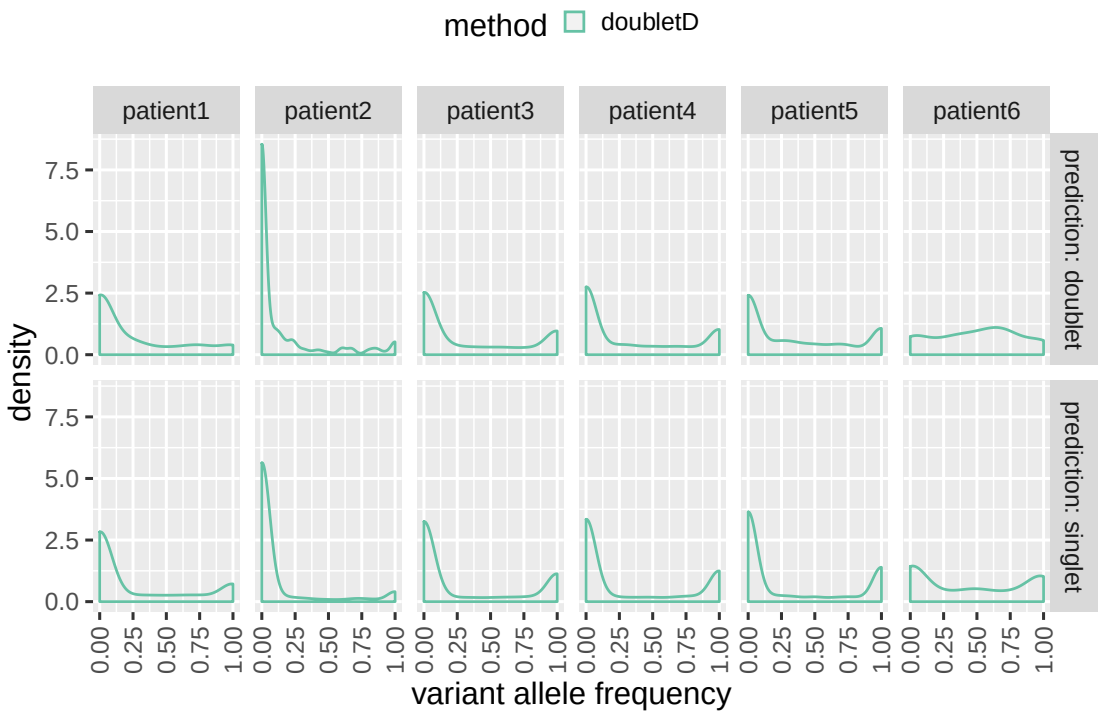
(a) (b)

Figure E.27: DOUBLETD results on cell line data. (a) The posterior likelihood (non-normalized) as a function of input doublet probability δ. Due to non-normalization, the log-likelihood is shifted by a constant explaining the positive values observed in the plot. (b) The observed VAF distributions for doubletD predicted doublets (1) and singlet droplets (8) for Raji droplets with neotypic doublet confidence score NCS = 1.

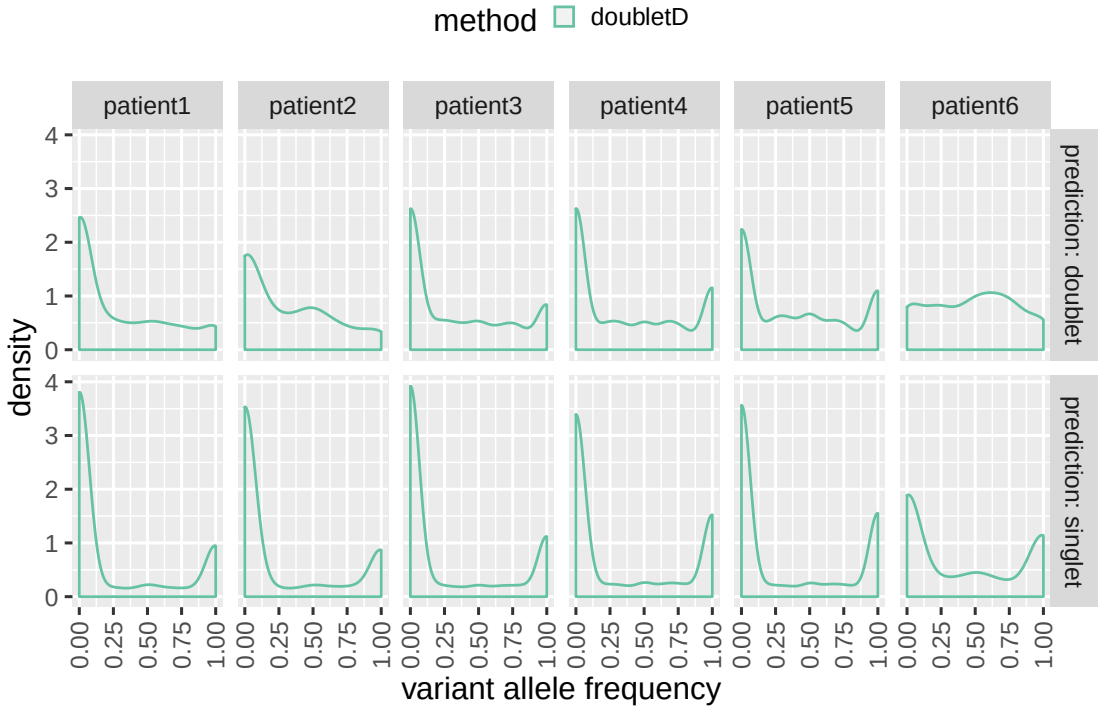


(a) NCS = 0 (b) NCS = 1 (c) NCS $\geq$ 2

Figure E.28: Venn diagram of the droplets with NCS score of (a) 0 (b) 1 and (c) $\geq$ 2 that were predicted as doublets by the three methods, doubletD, SCG:doublet and SCRUBLET.



(a)



(b)

Figure E.29: Aggregated observed variant allele frequency distribution by patient and DOUBLETD prediction for (a) across holdout loci (b) across inference loci.

E.8 COMPUTATION OF SNV CLONE PROPORTIONS

Each edge of the SNV clone tree T_1 reported by Gundem *et al.* [8] represents a set of mutations, also known as mutation clusters. As such, for a SNV clone tree T_1 with n_1 vertices, there are $n_1 - 1$ mutation clusters. The authors have provided the cancer cell fraction (CCF) for each of the mutation clusters in each sample of the ten patients. They used pigeonhole principle (PPH) to construct the SNV clone tree manually. For a given patient, let $F \in [0,1]^{m\times(n_1-1)}$ be the CCF matrix such that $F = [f_{p,k}]$ and $f_{p,k}$ is the CCF of mutation cluster $k \in [n_1 - 1]$ in sample $p \in [m]$. The SNV clone tree T_1, excluding the root vertex which represent the normal cell, is used to construct a perfect phylogeny matrix B [181]. We use the perfect phylogeny matrix B and the CCF matrix F to get the proportion U' of SNV clones, excluding the normal clone, in each sample of the ten patients by solving the following linear program

$$\min |F - BU'|_1, \tag{E.1}$$

$$\text{s.t. } 0 \leq u_{p,i} \leq 1, \quad \forall p \in [m], i \in [n_1 - 1], \tag{E.2}$$

$$\sum_{i=1}^{n_1-1} u_{p,i} = 1, \quad \forall p \in [m], \tag{E.3}$$

where $|\cdot|_1$ is the entry-wise L_1 norm. Finally, we correct the proportion matrix U' for the purity of the tumor samples (also known as tumor cellularity), which is the proportion of cancer cells in the tumor. We use the proportion of normal cells in each sample, inferred by HATCHet [141], to compute the purity of the tumor samples. Let $\boldsymbol{\gamma} \in [0,1]^{m\times 1}$ be a vector such that $\gamma_{p,1}$ is the purity of sample $p \in [m]$ inferred using HATCHet. The proportion matrix $U \in [0,1]^{m\times n_1}$ of the SNV clones is given by

$$U = \begin{bmatrix} \text{Diag}(\boldsymbol{\gamma})U' & \mathbf{1}_m - \boldsymbol{\gamma} \end{bmatrix} \tag{E.4}$$

where $\mathbf{1}_m$ is a $m \times 1$ vector with all entries equal to 1 and $\text{Diag}(\gamma)$ is a $m \times m$ diagonal matrix with the diagonal elements given by the entries of the vector $\boldsymbol{\gamma}$. It is easy to see that the proportion matrix U satisfies the conditions for being a proportion matrix (see Definition 5.1).

E.9 ADDITIONAL RESULTS ON PCTR PROBLEM

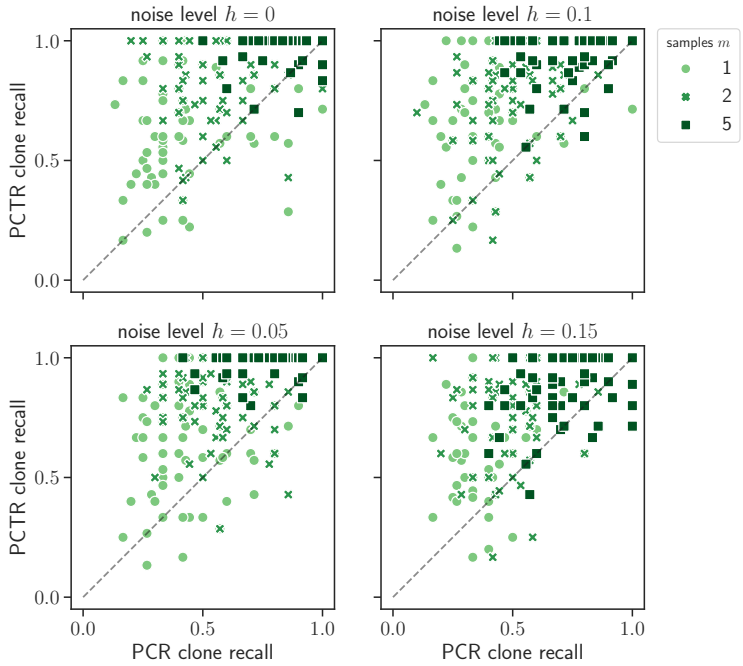


Figure E.30: Clone recall for the two modes of PACTION on the simulated instances. We show the clone recall of PACTION with the PCR and the PCTR mode on the simulated instances for varying noise levels h and number m of samples. For majority of simulated instances, PACTION in the PCTR mode has a higher recall compared to the PCR mode.

patient	number m of samples	number n_1 of SNV clones	number n_2 of CNA clones
A10	4	10	8
A12	3	5	8
A17	5	11	6
A21	8	15	6
A22	10	16	4
A24	4	10	4
A29	2	6	4
A31	5	11	6
A32	5	13	6
A34	3	14	6

Table E.6: Statistics of the metastatic prostate cancer data [8]. Number m of samples, number n_1 of SNV clones and number n_2 of CNA clones for the 10 patients from Gundem *et al.* [8]. The CNA clones were identified using HATCHet [141].

References

[1] B. Vrancken et al., "The genealogical population dynamics of HIV-1 in a large transmission chain: bridging within and among host evolutionary rates," *PLoS Computational Biology*, vol. 10, no. 4, 2014.

[2] P. Sashittal, C. Zhang, J. Peng, and M. El-Kebir, "Jumper enables discontinuous transcript assembly in coronaviruses," *bioRxiv*, 2021.

[3] D. Blanco-Melo, B. E. Nilsson-Payant, W.-C. Liu, S. Uhl, D. Hoagland, R. Møller, T. X. Jordan, K. Oishi, M. Panis, D. Sachs et al., "Imbalanced host response to SARS-CoV-2 drives development of COVID-19," *Cell*, 2020.

[4] P. C. Nowell, "The clonal evolution of tumor cell populations," *Science*, vol. 194, no. 4260, pp. 23–28, 1976.

[5] R. A. Burrell, N. McGranahan, J. Bartek, and C. Swanton, "The causes and consequences of genetic heterogeneity in cancer evolution," *Nature*, vol. 501, no. 7467, pp. 338–345, 2013.

[6] N. McGranahan and C. Swanton, "Biological and therapeutic impact of intratumor heterogeneity in cancer evolution," *Cancer cell*, vol. 27, no. 1, pp. 15–26, 2015.

[7] L. Weber, P. Sashittal, and M. El-Kebir, "doubletD: Detecting doublets in single-cell DNA sequencing data," 2021, (the first two authors contributed equally, in Print for Bioinformatics).

[8] G. Gundem, P. Van Loo, B. Kremeyer, L. B. Alexandrov, J. M. Tubio, E. Papaemmanuil, D. S. Brewer, H. M. Kallio, G. Högnäs, M. Annala et al., "The evolutionary history of lethal metastatic prostate cancer," *Nature*, vol. 520, no. 7547, pp. 353–357, 2015.

[9] S. Dellicour, G. Baele, G. Dudas, N. R. Faria, O. G. Pybus, M. A. Suchard, A. Rambaut, and P. Lemey, "Phylodynamic assessment of intervention strategies for the West African Ebola virus outbreak," *Nature communications*, vol. 9, no. 1, p. 2222, 2018.

[10] E. Romero-Severson et al., "Timing and order of transmission events is not directly reflected in a pathogen phylogeny," *Molecular Biology and Evolution*, vol. 31, no. 9, pp. 2472–2482, 2014.

[11] R. J. Ypma et al., "Unravelling transmission trees of infectious diseases by combining genetic and epidemiological data," *Proceedings of the Royal Society B: Biological Sciences*, vol. 279, no. 1728, pp. 444–450, 2011.

[12] S. R. Harris, E. J. Feil, M. T. Holden, M. A. Quail, E. K. Nickerson, N. Chantratita, S. Gardete, A. Tavares, N. Day, J. A. Lindsay et al., "Evolution of MRSA during hospital transmission and intercontinental spread," *Science*, vol. 327, no. 5964, pp. 469–474, 2010.

[13] T. Leitner et al., "Accurate reconstruction of a known HIV-1 transmission history by phylogenetic tree analysis," *Proceedings of the National Academy of Sciences*, vol. 93, no. 20, pp. 10 864–10 869, 1996.

[14] E. M. Cottam et al., "Integrating genetic and epidemiological data to determine transmission pathways of foot-and-mouth disease virus," *Proceedings of the Royal Society B: Biological Sciences*, vol. 275, no. 1637, pp. 887–895, 2008.

[15] X. Tang, C. Wu, X. Li, Y. Song, X. Yao, X. Wu, Y. Duan, H. Zhang, Y. Wang, Z. Qian et al., "On the origin and continuing evolution of SARS-CoV-2," *National Science Review*, 2020.

[16] Z. Shen, Y. Xiao, L. Kang, W. Ma, L. Shi, L. Zhang, Z. Zhou, J. Yang, J. Zhong, D. Yang et al., "Genomic diversity of SARS-CoV-2 in coronavirus disease 2019 patients," *Clinical Infectious Diseases*, 2020.

[17] M. El-Kebir, G. Satas, and B. J. Raphael, "Inferring parsimonious migration histories for metastatic cancers." *Nature Genetics*, vol. 50, no. 5, pp. 718–726, May 2018.

[18] A. S. Leonard et al., "Transmission bottleneck size estimation from pathogen deep-sequencing data, with an application to human influenza A virus," *Journal of Virology*, vol. 91, no. 14, pp. e00 171–17, 2017.

[19] R. J. Ypma, W. M. van Ballegooijen, and J. Wallinga, "Relating phylogenetic trees to transmission trees of infectious disease outbreaks," *Genetics*, vol. 195, no. 3, pp. 1055–1062, 2013.

[20] X. Didelot, J. Gardy, and C. Colijn, "Bayesian inference of infectious disease transmission from whole-genome sequence data," *Molecular Biology and Evolution*, vol. 31, no. 7, pp. 1869–1879, 2014.

[21] M. Hall et al., "Epidemic reconstruction in a phylogenetics framework: transmission trees as partitions of the node set," *PLoS Computational Biology*, vol. 11, no. 12, p. e1004613, 2015.

[22] X. Didelot, C. Fraser, J. Gardy, and C. Colijn, "Genomic infectious disease epidemiology in partially sampled and ongoing outbreaks," *Molecular Biology and Evolution*, vol. 34, no. 4, pp. 997–1007, 2017.

[23] N. De Maio, C.-H. Wu, and D. J. Wilson, "SCOTTI: efficient reconstruction of transmission within outbreaks with the structured coalescent," *PLoS Computational Biology*, vol. 12, no. 9, p. e1005130, 2016.

[24] N. De Maio, C. J. Worby, D. J. Wilson, and N. Stoesser, "Bayesian reconstruction of transmission within outbreaks using genomic variants," *PLoS Computational Biology*, vol. 14, no. 4, p. e1006117, 2018.

[25] P. Sashittal and M. El-Kebir, "SharpTNI: Counting and sampling parsimonious transmission networks under a weak bottleneck," *bioRxiv*, p. 842237, 2019.

[26] H. C. Whittle et al., "Effect of subclinical infection on maintaining immunity against measles in vaccinated children in west africa," *The Lancet*, vol. 353, no. 9147, pp. 98–102, 1999.

[27] H. J. Wearing and P. Rohani, "Estimating the duration of pertussis immunity using epidemiological signatures," *PLoS pathogens*, vol. 5, no. 10, 2009.

[28] E. Kenah et al., "Molecular infectious disease epidemiology: survival analysis and algorithms linking phylogenies to transmission trees," *PLoS Computational Biology*, vol. 12, no. 4, p. e1004869, 2016.

[29] M. D. Hall and C. Colijn, "Transmission trees on a known pathogen phylogeny: enumeration and sampling," *Molecular Biology and Evolution*, vol. 36, no. 6, pp. 1333–1343, 2019.

[30] R. Bouckaert, T. G. Vaughan, J. Barido-Sottani, S. Duchêne, M. Fourment, A. Gavryushkina, J. Heled, G. Jones, D. Kühnert, N. De Maio et al., "BEAST 2.5: An advanced software platform for Bayesian evolutionary analysis," *PLoS Computational Biology*, vol. 15, no. 4, p. e1006650, 2019.

[31] A. Stamatakis, "RAxML version 8: a tool for phylogenetic analysis and post-analysis of large phylogenies," *Bioinformatics*, vol. 30, no. 9, pp. 1312–1313, 2014.

[32] C. Wymant et al., "Phyloscanner: inferring transmission from within-and between-host pathogen genetic diversity," *Molecular Biology and Evolution*, vol. 35, no. 3, pp. 719–733, 2017.

[33] T. Jombart, M. Kendall, J. Almagro-Garcia, and C. Colijn, "treespace: Statistical exploration of landscapes of phylogenetic trees," *Molecular Ecology Resources*, vol. 17, no. 6, pp. 1385–1392, 2017.

[34] M. Kendall, D. Ayabina, Y. Xu, J. Stimson, C. Colijn et al., "Estimating transmission from genetic and epidemiological data: a metric to compare transmission trees," *Statistical Science*, vol. 33, no. 1, pp. 70–85, 2018.

[35] N. Aguse, Y. Qi, and M. El-Kebir, "Summarizing the solution space in tumor phylogeny inference by multiple consensus trees," *Bioinformatics*, vol. 35, no. 14, pp. i408–i416, 2019.

[36] K. Govek, C. Sikes, and L. Oesper, "A consensus approach to infer tumor evolutionary histories," in *Proceedings of the 2018 Acm international conference on bioinformatics, computational biology, and health informatics*, 2018, pp. 63–72.

[37] R. M. Karp, *Reducibility among Combinatorial Problems*. Springer, 1972, pp. 85–103.

[38] N. Creignou and M. Hermann, "On P completeness of some counting problems," INRIA, Research Report RR-2144, 1993. [Online]. Available: https://hal.inria.fr/inria-00074528

[39] M. Jerrum, *Counting, sampling and integrating: algorithms and complexity*. Springer Science & Business Media, 2003.

[40] I. Miklós, *Computational Complexity of Counting and Sampling*. CRC Press, 2019.

[41] S. Chakraborty, K. S. Meel, and M. Y. Vardi, "A Scalable Approximate Model Counter," in *Principles and Practice of Constraint Programming*. Berlin, Heidelberg: Springer, Berlin, Heidelberg, Sep. 2013, pp. 200–216.

[42] M. Soos et al., "BIRD: Engineering an efficient CNF-XOR SAT solver and its applications to approximate model counting," in *Proceedings of AAAI Conference on Artificial Intelligence (AAAI)(1 2019)*, 2019.

[43] S. Chakraborty et al., "Balancing scalability and uniformity in SAT witness generator," in *Proceedings of the 51st Annual Design Automation Conference*. ACM, 2014, pp. 1–6.

[44] S. Chakraborty, D. J. Fremont, K. S. Meel, S. A. Seshia, and M. Y. Vardi, "On parallel scalable uniform SAT witness generation," in *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*. Springer, 2015, pp. 304–319.

[45] L. J. Allen, "An introduction to stochastic epidemic models," in *Mathematical Epidemiology*. Springer, 2008, pp. 81–130.

[46] J. Kingman, "b the coalescent. stoch," in *Proc. Appl*, vol. 13, 1982, pp. 235–248.

[47] M. Thurley, "sharpSAT–counting models with advanced component caching and implicit bcp," in *International Conference on Theory and Applications of Satisfiability Testing*. Springer, 2006, pp. 424–429.

[48] D. Sankoff, "Minimal mutation trees of sequences," *SIAM Journal on Applied Mathematics*, vol. 28, no. 1, pp. 35–42, 1975.

[49] E. S. Snitkin et al., "Tracking a hospital outbreak of carbapenem-resistant Klebsiella pneumoniae with whole-genome sequencing," *Science Translational Medicine*, vol. 4, no. 148, pp. 148ra116–148ra116, 2012.

[50] P. Lemey, I. Derdelinckx, A. Rambaut, K. Van Laethem, S. Dumont, S. Vermeulen, E. Van Wijngaerden, and A.-M. Vandamme, "Molecular footprint of drug-selective pressure in a human immunodeficiency virus transmission chain," *Journal of Virology*, vol. 79, no. 18, pp. 11 981–11 989, 2005.

[51] M. Soos et al., "Extending SAT solvers to cryptographic problems," in *International Conference on Theory and Applications of Satisfiability Testing*. Springer, 2009, pp. 244–257.

[52] S. Sledzieski, C. Zhang, I. Mandoiu, and M. S. Bansal, "Treefix-tp: Phylogenetic error-correction for infectious disease transmission network inference," *bioRxiv*, p. 813931, 2019.

[53] A. A. De Vries, M. C. Horzinek, P. J. Rottier, and R. J. De Groot, "The genome organization of the Nidovirales: similarities and differences between Arteri-, Toro-, and Coronaviruses," in *Seminars in VIROLOGY*, vol. 8, no. 1. Elsevier, 1997, pp. 33–47.

[54] H. J. Maier, E. Bickerton, P. Britton et al., *Coronaviruses: methods and protocols.* Springer Berlin, 2015.

[55] D. Kim, J.-Y. Lee, J.-S. Yang, J. W. Kim, V. N. Kim, and H. Chang, "The architecture of SARS-CoV-2 transcriptome," *Cell*, 2020.

[56] A. D. Davidson, M. K. Williamson, S. Lewis, D. Shoemark, M. W. Carroll, K. J. Heesom, M. Zambon, J. Ellis, P. A. Lewis, J. A. Hiscox et al., "Characterisation of the transcriptome and proteome of SARS-CoV-2 reveals a cell passage induced in-frame deletion of the furin-like cleavage site from the spike glycoprotein," *Genome medicine*, vol. 12, no. 1, pp. 1–15, 2020.

[57] Y. Finkel, O. Mizrahi, A. Nachshon, S. Weingarten-Gabbay, D. Morgenstern, Y. Yahalom-Ronen, H. Tamir, H. Achdout, D. Stein, O. Israeli et al., "The coding capacity of SARS-CoV-2," *Nature*, vol. 589, no. 7840, pp. 125–130, 2021.

[58] G. Robertson, J. Schein, R. Chiu, R. Corbett, M. Field, S. D. Jackman, K. Mungall, S. Lee, H. M. Okada, J. Q. Qian et al., "De novo assembly and analysis of RNA-seq data," *Nature Methods*, vol. 7, no. 11, pp. 909–912, 2010.

[59] M. G. Grabherr, B. J. Haas, M. Yassour, J. Z. Levin, D. A. Thompson, I. Amit, X. Adiconis, L. Fan, R. Raychowdhury, Q. Zeng et al., "Trinity: reconstructing a full-length transcriptome without a genome from RNA-Seq data," *Nature Biotechnology*, vol. 29, no. 7, p. 644, 2011.

[60] Y. Xie, G. Wu, J. Tang, R. Luo, J. Patterson, S. Liu, W. Huang, G. He, S. Gu, S. Li et al., "SOAPdenovo-Trans: de novo transcriptome assembly with short RNA-Seq reads," *Bioinformatics*, vol. 30, no. 12, pp. 1660–1666, 2014.

[61] Z. Chang, G. Li, J. Liu, Y. Zhang, C. Ashby, D. Liu, C. L. Cramer, and X. Huang, "Bridger: a new framework for de novo transcriptome assembly using RNA-seq data," *Genome Biology*, vol. 16, no. 1, p. 30, 2015.

[62] M. H. Schulz, D. R. Zerbino, M. Vingron, and E. Birney, "Oases: robust de novo RNA-seq assembly across the dynamic range of expression levels," *Bioinformatics*, vol. 28, no. 8, pp. 1086–1092, 2012.

[63] M. Shao and C. Kingsford, "Accurate assembly of transcripts through phase-preserving graph decomposition," *Nature Biotechnology*, vol. 35, no. 12, pp. 1167–1169, 2017.

[64] M. Pertea, G. M. Pertea, C. M. Antonescu, T.-C. Chang, J. T. Mendell, and S. L. Salzberg, "StringTie enables improved reconstruction of a transcriptome from RNA-seq reads," *Nature Biotechnology*, vol. 33, no. 3, pp. 290–295, 2015.

[65] J. Liu, T. Yu, T. Jiang, and G. Li, "TransComb: genome-guided transcriptome assembly via combing junctions in splicing graphs," *Genome Biology*, vol. 17, no. 1, p. 213, 2016.

[66] C. Trapnell, B. A. Williams, G. Pertea, A. Mortazavi, G. Kwan, M. J. Van Baren, S. L. Salzberg, B. J. Wold, and L. Pachter, "Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation," *Nature Biotechnology*, vol. 28, no. 5, pp. 511–515, 2010.

[67] L. Song and L. Florea, "CLASS: constrained transcript assembly of RNA-seq reads," in *BMC Bioinformatics*, vol. 14, no. S5. Springer, 2013, p. S14.

[68] N. L. Bray, H. Pimentel, P. Melsted, and L. Pachter, "Near-optimal probabilistic RNA-seq quantification," *Nature Biotechnology*, vol. 34, no. 5, pp. 525–527, 2016.

[69] R. Patro, G. Duggal, M. I. Love, R. A. Irizarry, and C. Kingsford, "Salmon provides fast and bias-aware quantification of transcript expression," *Nature methods*, vol. 14, no. 4, pp. 417–419, 2017.

[70] X. Zhang, H. Chu, L. Wen, H. Shuai, D. Yang, Y. Wang, Y. Hou, Z. Zhu, S. Yuan, F. Yin et al., "Competing endogenous RNA network profiling reveals novel host dependency factors required for MERS-CoV propagation," *Emerging microbes & infections*, vol. 9, no. 1, pp. 733–746, 2020.

[71] A. Dobin, C. A. Davis, F. Schlesinger, J. Drenkow, C. Zaleski, S. Jha, P. Batut, M. Chaisson, and T. R. Gingeras, "STAR: ultrafast universal RNA-seq aligner," *Bioinformatics*, vol. 29, no. 1, pp. 15–21, 2013.

[72] E. Bernard, L. Jacob, J. Mairal, and J.-P. Vert, "Efficient RNA isoform identification and quantification from RNA-Seq data with network flows," *Bioinformatics*, vol. 30, no. 17, pp. 2447–2455, 2014.

[73] E. Bernard, L. Jacob, J. Mairal, E. Viara, and J.-P. Vert, "A convex formulation for joint RNA isoform detection and quantification from multiple RNA-seq samples," *BMC bioinformatics*, vol. 16, no. 1, pp. 1–10, 2015.

[74] H. Zheng, C. Ma, and C. Kingsford, "Deriving ranges of optimal estimated transcript expression due to non-identifiability," *bioRxiv*, pp. 2019–12, 2021.

[75] B. Li and C. N. Dewey, "RSEM: accurate transcript quantification from RNA-Seq data with or without a reference genome," *BMC bioinformatics*, vol. 12, no. 1, p. 323, 2011.

[76] L. Gurobi Optimization, "Gurobi optimizer reference manual," 2020. [Online]. Available: http://www.gurobi.com

[77] H. Li, B. Handsaker, A. Wysoker, T. Fennell, J. Ruan, N. Homer, G. Marth, G. Abecasis, and R. Durbin, "The sequence alignment/map format and SAMtools," *Bioinformatics*, vol. 25, no. 16, pp. 2078–2079, 2009.

[78] D. Yang and J. L. Leibowitz, "The structure and functions of coronavirus genomic 3' and 5' ends," *Virus research*, vol. 206, pp. 120–133, 2015.

[79] A. C. Frazee, A. E. Jaffe, B. Langmead, and J. T. Leek, "Polyester: simulating RNA-seq datasets with differential transcript expression," *Bioinformatics*, vol. 31, no. 17, pp. 2778–2784, 2015.

[80] D. M. Gohl, J. Garbe, P. Grady, J. Daniel, R. H. Watson, B. Auch, A. Nelson, S. Yohe, and K. B. Beckman, "A rapid, cost-effective tailed amplicon method for sequencing SARS-CoV-2," *BMC genomics*, vol. 21, no. 1, pp. 1–10, 2020.

[81] J. Quick, "nCoV-2019 sequencing protocol v3 (LoCost)," *protocols.io*, 08 2020, https://protocols.io/view/ncov-2019-sequencing-protocol-v3-locost-bh42j8ye.

[82] H. Li, "Minimap2: pairwise alignment for nucleotide sequences," *Bioinformatics*, vol. 34, no. 18, pp. 3094–3100, 2018.

[83] V. S. Mandala, M. J. McKay, A. A. Shcherbakov, A. J. Dregni, A. Kolocouris, and M. Hong, "Structure and drug binding of the SARS-CoV-2 envelope protein transmembrane domain in lipid bilayers," *Nature Structural & Molecular Biology*, vol. 27, no. 12, pp. 1202–1208, Dec. 2020. [Online]. Available: http://www.nature.com/articles/s41594-020-00536-8

[84] S. Kang, M. Yang, Z. Hong, L. Zhang, Z. Huang, X. Chen, S. He, Z. Zhou, Z. Zhou, Q. Chen et al., "Crystal structure of SARS-CoV-2 nucleocapsid protein RNA binding domain reveals potential unique drug targeting sites," *Acta Pharmaceutica Sinica B*, vol. 10, no. 7, pp. 1228–1238, 2020.

[85] Q. Ye, A. M. West, S. Silletti, and K. D. Corbett, "Architecture and self-assembly of the SARS-CoV-2 nucleocapsid protein," *Protein Science*, vol. 29, no. 9, pp. 1890–1901, 2020.

[86] A. Murira and A. Lamarre, "Type-I interferon responses: from friend to foe in the battle against chronic viral infection," *Frontiers in immunology*, vol. 7, p. 609, 2016.

[87] J. S. Lee and E.-C. Shin, "The type I interferon response in COVID-19: implications for treatment," *Nature Reviews Immunology*, vol. 20, no. 10, pp. 585–586, 2020.

[88] S. Xia, Q. Lan, S. Su, X. Wang, W. Xu, Z. Liu, Y. Zhu, Q. Wang, L. Lu, and S. Jiang, "The role of furin cleavage site in SARS-CoV-2 spike protein-mediated membrane fusion in the presence or absence of trypsin," *Signal transduction and targeted therapy*, vol. 5, no. 1, pp. 1–3, 2020.

[89] B. A. Johnson, X. Xie, A. L. Bailey, B. Kalveram, K. G. Lokugamage, A. Muruato, J. Zou, X. Zhang, T. Juelich, J. K. Smith et al., "Loss of furin cleavage site attenuates SARS-CoV-2 pathogenesis," *Nature*, pp. 1–10, 2021.

[90] Y. Yang, W. Yan, A. B. Hall, and X. Jiang, "Characterizing Transcriptional Regulatory Sequences in Coronaviruses and Their Role in Recombination," *Molecular Biology and Evolution*, 11 2020, msaa281. [Online]. Available: https://doi.org/10.1093/molbev/msaa281

[91] P. Sashittal, Y. Luo, J. Peng, and M. El-Kebir, "Characterization of SARS-CoV-2 viral diversity within and across hosts," *bioRxiv*, 2020.

[92] R. Rose, D. J. Nolan, S. Moot, A. Feehan, S. Cross, J. Garcia-Diaz, and S. L. Lamers, "Intra-host site-specific polymorphisms of SARS-CoV-2 is consistent across multiple samples and methodologies," *medRxiv*, 2020.

[93] D. Ramazzotti, F. Angaroni, D. Maspero, C. Gambacorti-Passerini, M. Antoniotti, A. Graudenzi, and R. Piazza, "Characterization of intra-host SARS-CoV-2 variants improves phylogenomic reconstruction and may reveal functionally convergent mutations," *bioRxiv*, 2020.

[94] T. Karamitros, G. Papadopoulou, M. Bousali, A. Mexias, S. Tsiodras, and A. Mentis, "SARS-CoV-2 exhibits intra-host genomic plasticity and low-frequency polymorphic quasispecies," *bioRxiv*, 2020.

[95] C. Gawad et al., "Dissecting the clonal origins of childhood acute lymphoblastic leukemia by single-cell genomics," *Proceedings of the National Academy of Sciences*, vol. 111, no. 50, pp. 17 947–17 952, 2014.

[96] K. Morita et al., "Clonal evolution of acute myeloid leukemia revealed by high-throughput single-cell genomics," *Nature Communications*, vol. 11, no. 1, pp. 1–1, 2020.

[97] L. A. Miles et al., "Single-cell mutation analysis of clonal evolution in myeloid malignancies," *Nature*, vol. 587, no. 7834, pp. 477–482, 2020.

[98] B. Lim et al., "Advancing cancer research and medicine with single-cell genomics," *Cancer Cell*, vol. 37, no. 4, pp. 456–470, 2020.

[99] K. Jahn et al., "Tree inference for single-cell data," *Genome biology*, vol. 17, no. 1, pp. 1–17, 2016.

[100] M. El-Kebir, "SPhyR: tumor phylogeny estimation from single-cell sequencing data under loss and error," *Bioinformatics*, vol. 34, no. 17, pp. i671–i679, 2018.

[101] E. M. Ross and F. Markowetz, "OncoNEM: inferring tumor evolution from single-cell sequencing data," *Genome biology*, vol. 17, no. 1, pp. 1–14, 2016.

[102] H. Zafar et al., "SiCloneFit: Bayesian inference of population structure, genotype, and phylogeny of tumor clones from single-cell genome sequencing data," *Genome research*, vol. 29, no. 11, pp. 1847–1859, 2019.

[103] A. Roth et al., "Clonal genotype and population structure inference from single-cell tumor sequencing," *Nature methods*, vol. 13, no. 7, pp. 573–576, 2016.

[104] G. Satas, S. Zaccaria, G. Mon, and B. J. Raphael, "Scarlet: Single-cell tumor phylogeny inference with copy-number constrained mutation losses," *Cell Systems*, vol. 10, no. 4, pp. 323–332, 2020.

[105] M. Pellegrino et al., "High-throughput single-cell DNA sequencing of acute myeloid leukemia tumors with droplet microfluidics," *Genome research*, vol. 28, no. 9, pp. 1345–1352, 2018.

[106] C. F. De Bourcy et al., "A quantitative comparison of single-cell whole genome amplification methods," *PLOS One*, vol. 9, no. 8, p. e105585, 2014.

[107] H. Liu et al., "Improving single-cell encapsulation efficiency and reliability through neutral buoyancy of suspension," *Micromachines*, vol. 11, no. 1, p. 94, 2020.

[108] N. E. Navin and K. Chen, "Genotyping tumor clones from single-cell data," *Nature methods*, vol. 13, no. 7, pp. 555–556, 2016.

[109] H. Zafar et al., "Computational approaches for inferring tumor evolution from single-cell genomic data," *Current Opinion in Systems Biology*, vol. 7, pp. 16–25, 2018.

[110] J. Kuipers et al., "Advances in understanding tumour evolution through single-cell sequencing," *Biochimica et Biophysica Acta (BBA)-Reviews on Cancer*, vol. 1867, no. 2, pp. 127–138, 2017.

[111] S. L. Wolock et al., "Scrublet: computational identification of cell doublets in single-cell transcriptomic data," *Cell systems*, vol. 8, no. 4, pp. 281–291, 2019.

[112] S. Salehi et al., "Single cell fitness landscapes induced by genetic and pharmacologic perturbations in cancer," *bioRxiv*, 2020.

[113] J. Kuipers et al., "Single-cell sequencing data reveal widespread recurrence and loss of mutational hits in the life histories of tumors," *Genome Research*, vol. 27, no. 11, pp. 1885–1894, Nov. 2017. [Online]. Available: http://genome.cshlp.org/lookup/doi/10.1101/gr.220707.117

[114] Y. Wu, "Accurate and efficient cell lineage tree inference from noisy single cell data: the maximum likelihood perfect phylogeny approach," *Bioinformatics*, vol. 36, no. 3, pp. 742–750, 2020.

[115] C. S. McGinnis et al., "DoubletFinder: doublet detection in single-cell RNA sequencing data using artificial nearest neighbors," *Cell systems*, vol. 8, no. 4, pp. 329–337, 2019.

[116] E. A. DePasquale et al., "DoubletDecon: deconvoluting doublets from single-cell RNA-sequencing data," *Cell reports*, vol. 29, no. 6, pp. 1718–1727, 2019.

[117] N. M. Xi and J. J. Li, "Benchmarking computational doublet-detection methods for single-cell RNA sequencing data," *Cell Systems*, 2020.

[118] L. J. Luquette et al., "Identification of somatic mutations in single cell DNA-seq using a spatial model of allelic imbalance," *Nature Communications*, vol. 10, no. 1, p. 3908, Dec. 2019. [Online]. Available: http://www.nature.com/articles/s41467-019-11857-8

[119] S. Zaccaria and B. J. Raphael, "Characterizing allele- and haplotype-specific copy numbers in single cells with CHISEL," *Nature Biotechnology*, Sep. 2020. [Online]. Available: http://www.nature.com/articles/s41587-020-0661-6

[120] B. Hwang et al., "Single-cell RNA sequencing technologies and bioinformatics pipelines," *Experimental & molecular medicine*, vol. 50, no. 8, pp. 1–14, 2018.

[121] G. Chen et al., "Single-cell RNA-seq technologies and related computational data analysis," *Frontiers in genetics*, vol. 10, p. 317, 2019.

[122] M. Gerstung et al., "Reliable detection of subclonal single-nucleotide variants in tumour cell populations," *Nature communications*, vol. 3, no. 1, pp. 1–8, 2012.

[123] H. Zafar et al., "Monovar: single-nucleotide variant detection in single cells," *Nature methods*, vol. 13, no. 6, pp. 505–507, 2016.

[124] D. Posada, "CellCoal: coalescent simulation of single-cell sequencing samples," *Molecular biology and evolution*, vol. 37, no. 5, pp. 1535–1542, 2020.

[125] D. Lähnemann et al., "Prosolo: Accurate variant calling from single cell dna sequencing data," *bioRxiv*, 2020.

[126] A. McPherson et al., "Divergent modes of clonal spread and intraperitoneal mixing in high-grade serous ovarian cancer," *Nature genetics*, vol. 48, no. 7, p. 758, 2016.

[127] Mission Bio, "Performance of the Tapestri platform for single-cell targeted DNA sequencing," Tech. Rep., 2019. [Online]. Available: https://missionbio.com/resources/technical-notes/platform_white_paper/

[128] S. Malikic et al., "Phiscs: a combinatorial approach for subperfect tumor phylogeny reconstruction via integrative use of single-cell and bulk sequencing data," *Genome research*, vol. 29, no. 11, pp. 1860–1877, 2019.

[129] S. Malikic et al., "Integrative inference of subclonal tumour evolution from single-cell and bulk sequencing data," *Nature communications*, vol. 10, no. 1, pp. 1–12, 2019.

[130] G. Ciriello, M. L. Miller, B. A. Aksoy, Y. Senbabaoglu, N. Schultz, and C. Sander, "Emerging landscape of oncogenic signatures across human cancers," *Nature genetics*, vol. 45, no. 10, pp. 1127–1133, 2013.

[131] I. The, T. P.-C. A. of Whole, G. Consortium et al., "Pan-cancer analysis of whole genomes," *Nature*, vol. 578, no. 7793, p. 82, 2020.

[132] T. B. Watkins, E. L. Lim, M. Petkovic, S. Elizalde, N. J. Birkbak, G. A. Wilson, D. A. Moore, E. Grönroos, A. Rowan, S. M. Dewhurst et al., "Pervasive chromosomal instability and karyotype order in tumour evolution," *Nature*, vol. 587, no. 7832, pp. 126–132, 2020.

[133] M. Tarabichi, A. Salcedo, A. G. Deshwar, M. N. Leathlobhair, J. Wintersinger, D. C. Wedge, P. Van Loo, Q. D. Morris, and P. C. Boutros, "A practical guide to cancer subclonal reconstruction from dna sequencing," *Nature methods*, vol. 18, no. 2, pp. 144–155, 2021.

[134] V. Popic, R. Salari, I. Hajirasouliha, D. Kashef-Haghighi, R. B. West, and S. Batzoglou, "Fast and scalable inference of multi-sample cancer lineages," *Genome biology*, vol. 16, no. 1, pp. 1–17, 2015.

[135] M. El-Kebir, L. Oesper, H. Acheson-Field, and B. J. Raphael, "Reconstruction of clonal trees and tumor composition from multi-sample sequencing data," *Bioinformatics*, vol. 31, no. 12, pp. i62–i70, 2015.

[136] F. Strino, F. Parisi, M. Micsinai, and Y. Kluger, "Trap: a tree approach for fingerprinting subclonal tumor composition," *Nucleic acids research*, vol. 41, no. 17, pp. e165–e165, 2013.

[137] G. Satas and B. J. Raphael, "Tumor phylogeny inference using tree-constrained importance sampling," *Bioinformatics*, vol. 33, no. 14, pp. i152–i160, 2017.

[138] L. K. Sundermann, J. Wintersinger, G. Rätsch, J. Stoye, and Q. Morris, "Reconstructing tumor evolutionary histories and clone trees in polynomial-time with submarine," *PLoS computational biology*, vol. 17, no. 1, p. e1008400, 2021.

[139] L. Oesper, A. Mahmoody, and B. J. Raphael, "Theta: inferring intra-tumor heterogeneity from high-throughput dna sequencing data," *Genome biology*, vol. 14, no. 7, pp. 1–21, 2013.

[140] A. Fischer, I. Vázquez-García, C. J. Illingworth, and V. Mustonen, "High-definition reconstruction of clonal composition in cancer," *Cell reports*, vol. 7, no. 5, pp. 1740–1752, 2014.

[141] S. Zaccaria and B. J. Raphael, "Accurate quantification of copy-number aberrations and whole-genome duplications in multi-sample tumor sequencing data," *Nature communications*, vol. 11, no. 1, pp. 1–13, 2020.

[142] F. Notta, M. Chan-Seng-Yue, M. Lemire, Y. Li, G. W. Wilson, A. A. Connor, R. E. Denroche, S.-B. Liang, A. M. Brown, J. C. Kim et al., "A renewed model of pancreatic cancer evolution based on genomic rearrangement patterns," *Nature*, vol. 538, no. 7625, pp. 378–382, 2016.

[143] S. Zaccaria, M. El-Kebir, G. W. Klau, and B. J. Raphael, "The copy-number tree mixture deconvolution problem and applications to multi-sample bulk sequencing tumor data," in *International Conference on Research in Computational Molecular Biology*. Springer, 2017, pp. 318–335.

[144] A. W. McPherson, A. Roth, G. Ha, C. Chauve, A. Steif, C. P. de Souza, P. Eirew, A. Bouchard-Côté, S. Aparicio, S. C. Sahinalp et al., "Remixt: clone-specific genomic structure estimation in cancer," *Genome biology*, vol. 18, no. 1, pp. 1–14, 2017.

[145] S. Zaccaria, M. El-Kebir, G. W. Klau, and B. J. Raphael, "Phylogenetic copy-number factorization of multiple tumor samples," *Journal of Computational Biology*, vol. 25, no. 7, pp. 689–708, 2018.

[146] A. G. Deshwar, S. Vembu, C. K. Yung, G. H. Jang, L. Stein, and Q. Morris, "Phylowgs: reconstructing subclonal composition and evolution from whole-genome sequencing of tumors," *Genome biology*, vol. 16, no. 1, pp. 1–20, 2015.

[147] M. El-Kebir, G. Satas, L. Oesper, and B. J. Raphael, "Inferring the Mutational History of a Tumor Using Multi-state Perfect Phylogeny Mixtures," *Cell Systems*, vol. 3, no. 1, pp. 43–53, July 2016. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2405471216302216

[148] Y. Jiang, Y. Qiu, A. J. Minn, and N. R. Zhang, "Assessing intratumor heterogeneity and tracking longitudinal and spatial clonal evolutionary history by next-generation sequencing," *Proceedings of the National Academy of Sciences*, vol. 113, no. 37, pp. E5528–E5537, 2016.

[149] M. Jamal-Hanjani, G. A. Wilson, N. McGranahan, N. J. Birkbak, T. B. Watkins, S. Veeriah, S. Shafi, D. H. Johnson, R. Mitter, R. Rosenthal et al., "Tracking the evolution of non–small-cell lung cancer," *New England Journal of Medicine*, vol. 376, no. 22, pp. 2109–2121, 2017.

[150] C. Gawad, W. Koh, and S. R. Quake, "Single-cell genome sequencing: current state of the science," *Nature Reviews Genetics*, vol. 17, no. 3, p. 175, 2016.

[151] M. L. Leung, A. Davis, R. Gao, A. Casasent, Y. Wang, E. Sei, E. Vilar, D. Maru, S. Kopetz, and N. E. Navin, "Single-cell dna sequencing reveals a late-dissemination model in metastatic colorectal cancer," *Genome research*, vol. 27, no. 8, pp. 1287–1299, 2017.

[152] S. Malikic, A. W. McPherson, N. Donmez, and C. S. Sahinalp, "Clonality inference in multiple tumor samples using phylogeny," *Bioinformatics*, vol. 31, no. 9, pp. 1349–1356, 2015.

[153] R. F. Schwarz, A. Trinh, B. Sipos, J. D. Brenton, N. Goldman, and F. Markowetz, "Phylogenetic quantification of intra-tumour heterogeneity," *PLoS Comput Biol*, vol. 10, no. 4, p. e1003535, 2014.

[154] M. El-Kebir, B. J. Raphael, R. Shamir, R. Sharan, S. Zaccaria, M. Zehavi, and R. Zeira, "Copy-number evolution problems: complexity and algorithms," in *International Workshop on Algorithms in Bioinformatics*. Springer, 2016, pp. 137–149.

[155] M. El-Kebir, B. J. Raphael, R. Shamir, R. Sharan, S. Zaccaria, M. Zehavi, and R. Zeira, "Complexity and algorithms for copy-number evolution problems," *Algorithms for Molecular Biology*, vol. 12, no. 1, pp. 1–11, 2017.

[156] T. Wu, V. Moulton, and M. Steel, "Refining phylogenetic trees given additional data: An algorithm based on parsimony," *IEEE/ACM transactions on computational biology and bioinformatics*, vol. 6, no. 1, pp. 118–125, 2008.

[157] R. K. Ahuja, T. L. Magnanti, J. B. Orlin, and K. Weihe, "Network flows: theory, algorithms and applications," *ZOR-methods and models of operations research*, vol. 41, no. 3, pp. 252–254, 1995.

[158] M. R. Garey and D. S. Johnson, "Computers and intractability. a guide to the theory of np-completeness," 1983.

[159] M. R. Garey and D. S. Johnson, "Complexity results for multiprocessor scheduling under resource constraints," *SIAM Journal on Computing*, vol. 4, no. 4, pp. 397–411, 1975.

[160] D. Fernández-Baca, "The perfect phylogeny problem," in *Steiner Trees in Industries*, D. Z. Zu and X. Cheng, Eds. Kluwer Acedemic Publishers, 2000.

[161] P. L. Krapivsky and S. Redner, "Organization of growing random networks," *Physical Review E*, vol. 63, no. 6, p. 066123, 2001.

[162] H. Teimouri and A. B. Kolomeisky, "Temporal order of mutations influences cancer initiation dynamics," *bioRxiv*, 2021.

[163] K. Sprouffske, J. W. Pepper, and C. C. Maley, "Accurate reconstruction of the temporal order of mutations in neoplastic progression," *Cancer prevention research*, vol. 4, no. 7, pp. 1135–1144, 2011.

[164] J. Guo, H. Guo, and Z. Wang, "Inferring the temporal order of cancer gene mutations in individual tumor samples," *PLoS One*, vol. 9, no. 2, p. e89244, 2014.

[165] S. Khakabimamaghani, D. Ding, O. Snow, and M. Ester, "Uncovering the subtype-specific temporal order of cancer pathway dysregulation," *PLoS computational biology*, vol. 15, no. 11, p. e1007451, 2019.

[166] J. Barnett, H. Correia, P. Johnson, M. Laughlin, and K. Wilson, "Darwin meets graph theory on a strange planet: Counting full n-ary trees with labeled leafs," *Alabama Journal of Mathematics*, 2010.

[167] J. Lee and D. Wilson, "Polyhedral methods for piecewise-linear functions I: the lambda method," *Discrete Applied Mathematics*, vol. 108, no. 3, pp. 269–285, 2001.

[168] A. Imamoto and B. Tang, "A recursive descent algorithm for finding the optimal minimax piecewise linear approximation of convex functions," in *Advances in Electrical and Electronics Engineering-IAENG Special Edition of the World Congress on Engineering and Computer Science 2008*. IEEE, 2008, pp. 287–293.

[169] K. Bowman and L. Shenton, "Parameter estimation for the beta distribution," *Journal of statistical computation and simulation*, vol. 43, no. 3-4, pp. 217–228, 1992.

[170] M. A. Lodato et al., "Somatic mutation in single human neurons tracks developmental and transcriptional history," *Science*, vol. 350, no. 6256, pp. 94–98, 2015.

[171] C. Ma, H. Zheng, and C. Kingsford, "Exact transcript quantification over splice graphs," in *20th International Workshop on Algorithms in Bioinformatics (WABI 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020.

[172] A. O. Pasternak, W. J. Spaan, and E. J. Snijder, "Nidovirus transcription: how to make sense...?" *Journal of General Virology*, vol. 87, no. 6, pp. 1403–1421, 2006.

[173] S. G. Sawicki and D. L. Sawicki, "Coronaviruses use discontinuous extension for synthesis of subgenome-length negative strands," in *Corona-and Related Viruses*. Springer, 1995, pp. 499–506.

[174] G. Van Marle, J. C. Dobbe, A. P. Gultyaev, W. Luytjes, W. J. Spaan, and E. J. Snijder, "Arterivirus discontinuous mRNA transcription is guided by base pairing between sense and antisense transcription-regulating sequences," *Proceedings of the National Academy of Sciences*, vol. 96, no. 21, pp. 12 056–12 061, 1999.

[175] S. Zuniga, I. Sola, S. Alonso, and L. Enjuanes, "Sequence motifs involved in the regulation of discontinuous coronavirus subgenomic RNA synthesis," *Journal of Virology*, vol. 78, no. 2, pp. 980–994, 2004.

[176] A. O. Pasternak, E. van den Born, W. J. Spaan, and E. J. Snijder, "Sequence requirements for RNA strand transfer during nidovirus discontinuous subgenomic RNA synthesis," *The EMBO Journal*, vol. 20, no. 24, pp. 7220–7228, 2001.

[177] D. L. Sawicki, T. Wang, and S. G. Sawicki, "The RNA structures engaged in replication and transcription of the A59 strain of mouse hepatitis virus," *Journal of General Virology*, vol. 82, no. 2, pp. 385–396, 2001.

[178] A. A. de Vries, A. L. Glaser, M. J. Raamsman, and P. J. Rottier, "Recombinant equine arteritis virus as an expression vector," *Virology*, vol. 284, no. 2, pp. 259–276, 2001.

[179] H. Prüfer, "Neuer beweis eines satzes uber permutationen," *Arch Math Phys*, vol. 27, pp. 742–4, 1918.

[180] Y. Fu et al., "Uniform and accurate single-cell sequencing based on emulsion whole-genome amplification," *Proceedings of the National Academy of Sciences*, vol. 112, no. 38, pp. 11 923–11 928, Sep. 2015. [Online]. Available: http://www.pnas.org/lookup/doi/10.1073/pnas.1513988112

[181] D. Pradhan and M. El-Kebir, "On the non-uniqueness of solutions to the perfect phylogeny mixture problem," in *RECOMB International conference on Comparative Genomics*. Springer, 2018, pp. 277–293.