



**Politecnico
di Torino**

Doctoral Thesis in Computer & Mathematical Sciences

Single-cell tumor phylogenetics

Probabilistic models and inference algorithms for tumor
evolution from copy-number aberrations

VITTORIO ZAMPINETTI

KTH ROYAL INSTITUTE OF TECHNOLOGY • POLITECNICO DI TORINO





Single-cell tumor phylogenetics

Probabilistic models and inference algorithms for tumor evolution from copy-number aberrations

VITTORIO ZAMPINETTI

Academic Dissertation which, with due permission of the KTH Royal Institute of Technology and the Politecnico di Torino, is submitted for public defence for the Degree of Doctor of Philosophy on September 18, 2026 in Aula Buzano, DISMA, Politecnico di Torino.

Doctoral Thesis in Computer & Mathematical Sciences
KTH Royal Institute of Technology • Politecnico di Torino
Torino, Italy 2026

Joint Degree Partnership

This thesis is submitted in fulfillment of the requirements for a joint degree
between:

- **KTH Royal Institute of Technology**, Stockholm, Sweden
- **Politecnico di Torino**, Turin, Italy

The research leading to this thesis was conducted under a co-tutelle agreement,
fostering international collaboration and interdisciplinary research.

© Vittorio Zampinetti 2026 unless otherwise stated.

This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). You are free to share and adapt it for any purpose, provided that appropriate credit is given. The full license text is available at <https://creativecommons.org/licenses/by/4.0/>.

The license above covers the introductory chapters of this thesis. The appended papers are included with permission from their respective copyright holders and remain subject to their own terms.

Cover image: Painting *No title*, Annamaria Malaguti

TRITA-EECS-AVL-2026:71
ISBN 978-91-8106-670-8

Typeset in L^AT_EX2e (2024-11-01).
Printed by Universitetsservice US-AB, Sweden 2026

Alla mia nonna Paola

Abstract

Tumors are highly heterogeneous populations of cells that evolve dynamically, acquiring mutations as they divide. Reconstructing this evolutionary history is essential for understanding cancer progression, metastasis, and therapy resistance. With the advent of single-cell DNA sequencing (scDNA-seq), we can now profile the genomic landscape of individual cells, which provides an unprecedented window to the formation and evolution of tumor cell populations. Specifically, single-cell whole-genome sequencing allows us to detect structural variations called *copy number aberrations* (CNAs), which are known to play a critical role in cancer development and progression. Inferring the evolutionary trees, i.e. *phylogenies*, from single-cell DNA sequences presents immense computational challenges due to both the sparsity of the data inherent to the current state of sequencing technologies, and the complexity of the underlying mutational processes.

In this thesis, we develop novel probabilistic models and algorithms to infer the evolutionary history of tumors from single-cell DNA sequencing data. The research addresses core methodological bottlenecks in tumor phylogenetics, moving from foundational distance-based methods to comprehensive joint Bayesian inference frameworks.

First, we introduce a method to estimate biologically meaningful evolutionary distances between single cells directly from noisy read counts, employing an original Hidden Markov Model to accommodate the unique noise profile of scDNA-seq and the interdependence of copy number states across the genome. Next, we extend distance-based tree inference from classical phylogenetics by presenting a scalable algorithm specifically designed for rooted trees, leveraging the biological premise that tumor evolution originates from a known healthy diploid ancestor. To enable rigorous uncertainty quantification over tree topologies, we then tackle the problem of sampling directed trees (arborescences). We present a stable, polynomial-time sampling algorithm capable of generating arborescences even on weakly connected graphs, which commonly arise when performing inference from single-cell sequences. Finally, we integrate these advancements into a comprehensive variational inference framework. This framework

efficiently achieves joint inference over clonal tree structures, branch lengths, copy number profiles, and cell-to-clone assignments.

Collectively, this thesis contributes a suite of statistically grounded, highly scalable tools that bridge the gap between noisy single-cell sequencing reads and robust insights into cancer evolution, offering a foundation for future clinical applications and oncological research.

Keywords

copy number, tumor phylogenetics, variational inference

Sammanfattning

Tumörer är mycket heterogena cellpopulationer som utvecklas dynamiskt och förvärvar mutationer när de delar sig. Att rekonstruera denna evolutionära historia är avgörande för att förstå cancerprogression, metastasering och terapiresistens. Med framväxten av enkelcells-DNA-sekvensering (scDNA-seq) kan vi nu kartlägga det genomiska landskapet hos enskilda celler, vilket ger en oöverträffad inblick i hur tumörcellpopulationer bildas och utvecklas. Specifikt gör enkelcells helgenomsekvensering det möjligt att upptäcka strukturella variationer kallade *copy number aberrations* (CNA:er), vilka är kända för att spela en avgörande roll i cancerutveckling och progression.

Att härleda evolutionära träd, dvs. *fylogener*, från enkelcells-DNA-sekvenser innebär betydande beräkningsmässiga utmaningar, både på grund av den gleshet i data som är inneboende i nuvarande sekvenseringsteknologier och på grund av komplexiteten i de underliggande mutationsprocesserna.

I denna avhandling utvecklar vi nya probabilistiska modeller och algoritmer för att härleda tumörens evolutionära historia från enkelcells-DNA-sekvenseringsdata. Forsknigen adresserar centrala metodologiska flaskhalsar inom tumörfyloenetik och rör sig från grundläggande avståndsbaserade metoder till omfattande gemensamma Bayesianiska inferensramverk.

Först introducerar vi en metod för att uppskatta biologiskt meningsfulla evolutionära avstånd mellan enskilda celler direkt från brusiga läsdata, genom att använda en original Hidden Markov Model som tar hänsyn till den unika brusprofilen hos scDNA-seq samt det ömsesidiga beroendet mellan kopietalstillstånd över genomet. Därefter utvidgar vi avståndsbaserad trädinferens från klassisk fyloenetik genom att presentera en skalbar algoritm specifikt utformad för rotade träd, baserad på den biologiska premissen att tumörutveckling utgår från en känd frisk diploid förfader. För att möjliggöra rigorös osäkerhetskvantifiering över trädtopologier tar vi sedan itu med problemet att sampla riktade träd (arborescenser). Vi presenterar en stabil algoritm med polynomisk tidskomplexitet som kan generera arborescenser även på svagt sammanhängande grafer,

vilket ofta uppstår vid inferens från enkelcellssekvenser. Slutligen integrerar vi dessa framsteg i ett omfattande variationalt inferensramverk. Detta ramverk möjliggör effektiv gemensam inferens av klonala trädstrukturer, grenlängder, kopietalsprofiler och tilldelning av celler till kloner.

Sammantaget bidrar denna avhandling med en uppsättning statistiskt välgrundade och mycket skalbara verktyg som överbryggar klyftan mellan brusiga enkelcellssekvenseringsdata och robusta insikter i cancerutveckling, och lägger grunden för framtida kliniska tillämpningar och onkologisk forskning.

Keywords

kopietal, tumörfylogeni, variational inference

Sommario

I tumori sono popolazioni cellulari altamente eterogenee che evolvono dinamicamente, acquisendo mutazioni durante la divisione cellulare. La ricostruzione di questa storia evolutiva è essenziale per comprendere la progressione del cancro, la metastasi e la resistenza alle terapie. Con l'avvento del sequenziamento del DNA a singola cellula (scDNA-seq), possiamo ora analizzare il panorama genomico delle singole cellule, offrendo una finestra senza precedenti sulla formazione e l'evoluzione delle popolazioni cellulari tumorali. In particolare, il sequenziamento dell'intero genoma a singola cellula consente di rilevare variazioni strutturali chiamate *copy number aberrations* (CNA), note per svolgere un ruolo cruciale nello sviluppo e nella progressione del cancro.

L'inferenza degli alberi evolutivi, ovvero le *filogenesi*, a partire da sequenze di DNA a singola cellula presenta notevoli sfide computazionali, dovute sia alla scarsità dei dati intrinseca alle attuali tecnologie di sequenziamento, sia alla complessità dei processi mutazionali sottostanti.

In questa tesi sviluppiamo nuovi modelli probabilistici e algoritmi per inferire la storia evolutiva dei tumori a partire da dati di sequenziamento del DNA a singola cellula. La ricerca affronta colli di bottiglia metodologici fondamentali nella filogenetica tumorale, passando da metodi basati sulla distanza a framework completi di inferenza bayesiana congiunta.

In primo luogo, introduciamo un metodo per stimare distanze evolutive biologicamente significative tra singole cellule direttamente da conteggi di letture rumorosi, utilizzando un modello Hidden Markov originale che tiene conto del profilo di rumore specifico dello scDNA-seq e dell'interdipendenza degli stati di numero di copie lungo il genoma. Successivamente, estendiamo l'inferenza di alberi basata sulla distanza dalla filogenetica classica presentando un algoritmo scalabile specificamente progettato per alberi radicati, sfruttando il presupposto biologico che l'evoluzione tumorale abbia origine da un antenato diploide sano noto. Per consentire una rigorosa quantificazione dell'incertezza sulle topologie degli alberi, affrontiamo quindi il problema del campionamento

di alberi diretti (arborescenze). Presentiamo un algoritmo stabile a tempo polinomiale in grado di generare arborescenze anche su grafi debolmente connessi, che si presentano comunemente durante l'inferenza da sequenze a singola cellula. Infine, integriamo questi avanzamenti in un framework completo di inferenza variazionale. Questo framework consente un'efficiente inferenza congiunta delle strutture degli alberi clonali, delle lunghezze dei rami, dei profili di numero di copie e dell'assegnazione delle cellule ai cloni.

Nel complesso, questa tesi contribuisce con una serie di strumenti altamente scalabili e statisticamente fondati che colmano il divario tra dati di sequenziamento a singola cellula rumorosi e solide intuizioni sull'evoluzione del cancro, offrendo una base per future applicazioni cliniche e la ricerca oncologica.

Keywords

numero di copie, filogenesi tumorale, inferenza variazionale

Acknowledgment

I would like to thank my supervisors, Jens Lagergren, whose ideas and guidance have always inspired me to challenge myself, and Mauro Gasparini, who tutored me in the art of teaching.

Thanks to my colleagues at Scilife Lab who made the lab a less lonely place. Special thanks to all the *dismini*, fellow PhD students (and non) I met at Politecnico, with whom I shared many unforgettable moments.

I will be forever grateful to my wife, Martina, for her love and constant support during the struggles of my PhD. Deep and heartfelt thanks to my parents, my sister and my brother, who made sure I never forgot to dream and to believe in myself.

A handwritten signature in dark ink, reading "Vittorio Zampinetti", is written over a horizontal line. The signature is fluid and cursive, with a long horizontal stroke at the end.

Vittorio Zampinetti
Stockholm, August 12, 2026

List of included papers

Contributions This doctoral dissertation is a compilation thesis. The kappa provides a self-standing synthesis of the research area, the main questions addressed, the gaps filled by the included papers, and the author's own contributions. The second part comprises the published papers that form the scientific basis of the thesis.

- Paper A** **Cellmates: a method for single-cell phylogeny reconstruction with proper evolutionary distances from copy numbers**, Vittorio Zampinetti, Harald Melin, Marzie Abdolhamdi, and Jens Lagergren. Manuscript, 2026
- Paper B** **Scalable and robust phylogenetic tree reconstruction from copy-number data with Sparse Rooted Neighbor Joining**, Vittorio Zampinetti, Harald Melin, Alva Hallin, and Jens Lagergren. In *bioRxiv*, 2026
- Paper C** **Sampling random spanning arborescences in graphs with low conductance**, Vittorio Zampinetti, Harald Melin, and Jens Lagergren. In *Statistics & Probability Letters*, 2025
- Paper D** **VICTree-A Variational Inference Method for Clonal Tree Reconstruction**, Harald Melin, Vittorio Zampinetti, Andrew McPherson, and Jens Lagergren. In *International Conference on Research in Computational Molecular Biology*, 2024

I Statement of own contributions

My contributions to the thesis include the thesis-level synthesis, the integration of the paper narratives, and significant parts of the method development across the four papers. Cellmates (Paper A) and VICTree (Paper D) were developed together with Harald Melin, with whom I share first authorship. We both contributed equally to the conception, design, implementation, and evaluation of these methods. The other two papers, on the SparseRNJ algorithm (Paper B) and the sampling algorithm for arborescences (Paper C) were developed by myself, with feedback from my supervisors and co-authors.

Declaration of use of generative AI

Generative AI was used as an editing and drafting aid for thesis planning, structuring, and language polishing. All scientific claims, interpretations, and final text are the responsibility of the author.

Contents

Abstract	ix
Sammanfattning	xi
Sommario	xiii
Acknowledgment	xv
List of included papers	xvii
I Statement of own contributions	xix
Declaration of use of generative AI	xxi
Contents	xxiii
List of Figures	xxv
Acronyms	xxxi

I Introduction	1
1 Introduction	3
1.1 Single-cell DNA sequencing and copy-number aberrations	5
1.2 Inference for phylogenetics	6
1.3 Research questions	7
1.4 Thesis outline	9
2 Background	11
2.1 Genomics	11
2.2 Probabilistic modeling and inference	18
2.3 Phylogenetics	25
2.4 Tree sampling	31
2.5 Research gaps	34
3 Copy number evolution and distance-based phylogenetics	37
3.1 Evolutionary distances	38
3.2 Calling copy-number profiles from single-cell data	41
3.3 Distance-based tree reconstruction and the missing root	48
3.4 Summary of contributions	50

CONTENTS

4 Joint Bayesian inference in tumor phylogenetics 53

4.1 Bayesian models of clonal evolution 54

4.2 Scalable joint inference from sequencing data 58

4.3 Sampling spanning arborescences 60

4.4 Thesis contributions 62

5 Summary of the papers 65

5.1 Paper A 65

5.2 Paper B 66

5.3 Paper C 67

5.4 Paper D 67

6 Conclusions 69

6.1 Summary of contributions 69

6.2 Limitations and Future work 70

6.3 Concluding remarks 70

Bibliography 71

II Appended papers 85

List of Figures

2.1.1	Illustration of the packing of the DNA molecule into chromosomes. The DNA double helix wraps around histone proteins to form nucleosomes, which further coil and fold to create the compact structure of a chromosome.	13
2.1.2	Types of genetic variation. (<i>Top</i>) A reference DNA sequence illustrating three mutation types: an SNP/SNV (single base substitution, highlighted in red); a CNA amplification (a segment duplicated to three tandem copies); and a segment deletion. (<i>Bottom</i>) Allelic imbalance arising from a copy error during cell division. A normal cell carries one maternal (purple) and one paternal (blue) allele in a 1:1 ratio. After an unequal division, one daughter cell receives two copies of allele A (2:1 imbalance), while the other loses allele B entirely — a loss of heterozygosity (LOH).	15
2.1.3	Read counts and underlying copy-number profile across genomic bins. Grey dots show the raw read count observed in each fixed-size bin for a simulated single cell, drawn from a Negative Binomial distribution whose mean is proportional to the local copy number. The blue step line indicates the true integer copy-number state (0–5). Four copy-number events are highlighted: a two-copy amplification (Amp.), a single-copy gain (Gain), a single-copy loss (Loss), and a homozygous deletion (Hom. del.). The spread of read counts around the expected mean reflects the overdispersion intrinsic to single-cell sequencing data, illustrating why computational methods must explicitly account for this noise when inferring copy-number states from raw read counts.	17
2.2.4	Directed Acyclic Graph (DAG) for a finite mixture model. The shaded node y_{ig} represents the observed data, while the unshaded node z_i is the latent cluster assignment. The parameters π and θ_g govern the categorical distribution and the emission distributions, respectively. The plates denote N independent observations and G genes.	20
2.2.5	Directed Acyclic Graph for a Hidden Markov Model. The latent states z_m form a Markov chain, and each state emits an observation y_m . The initial state is governed by π (omitted), while transitions and emissions are controlled by $\mathbf{A}$ and ϕ respectively.	21
2.2.6	Illustration of overfitting. A flexible, complex model (red dashed line) perfectly hits all data points by minimizing the fitting error, but completely misses the simpler true underlying function (blue line).	22
2.4.7	Visualizing trees and arborescences. (a) A weighted directed graph $\mathcal{G}$ with root node r . (b) A valid spanning tree (ignoring edge directions). (c) A valid spanning arborescence rooted at r , where all arcs point away from the root.	32

LIST OF FIGURES

3.1.1	Figurative representation of copy-number evolution. Each horizontal line depicts one DNA copy within a genomic bin; the integer below each bin states the copy number explicitly. New breakpoints (BP, solid red) mark positions where the child's segmentation differs from the parent's, reflecting a copy-number change event. Inherited BPs (dashed gray) are boundaries shared by both profiles.	40
3.2.2	Example of a generative model (DAG) of the observed read count y_{mn} for cell n at bin b , where the counts are assumed to be drawn from a Negative-Binomial distribution with mean $\mu_n \times C_{mn}$ and overdispersion α . The mean is therefore modeled as a function of the copy number C_{mn} , which is a latent variable representing the true copy-number profile of the cell.	45
3.2.3	Whole-genome doubling inflates copy-number states without proportionally increasing read depth. (A) Genome-wide read depth (grey, normalised to copy-number scale) and inferred copy-number profile (orange) for one representative WGD cell from the SPECTRUM-OV-075 single-cell whole-genome sequencing dataset with DLP+ protocol [92]. The purple line draws the integer part of the inferred copy number divided by two, illustrating the ambiguity introduced by genome doubling. Vertical grey lines demarcate chromosome boundaries. (B) Violin plots of total per-cell read counts stratified by ploidy class (diploid, tetraploid, and octaploid). Although mean ploidy increases twofold at each step, the distributions of total read depth overlap substantially, demonstrating that sequencing depth is not linearly proportional to ploidy, and that read counts alone cannot be used as a proxy for WGD status.	46
3.2.4	Allele-specific copy-number profile for chromosome 13 in a whole-genome doubled cell. Mirrored B-allele frequency (mBAF) is shown across the genomic position (in Mb) for a representative WGD cell from the SPECTRUM-OV-075 dataset. Grey dots represent the empirical mBAF calculated directly from raw allele-specific read counts. Solid lines indicate the expected mBAF corresponding to the inferred allele-specific copy-number states: the orange line shows the expected mBAF under the WGD model fit, while the purple line shows the expected ratio if the cell were strictly diploid.	47
4.2.1	Illustration of ELBO-driven model selection on a breast cancer 10x Genomics scDNA-seq dataset [145]. Each marker denotes the final ELBO from one run at a fixed clone number K ; the solid line shows the mean across runs and the shaded region indicates one standard deviation. The averaged ELBO profile is convex and attains its maximum at $K = 9$, exemplifying how ELBO can identify a well-supported model complexity.	61

- 4.3.2 Example of a weight matrix representing the inferred variational distribution over trees with $K = 7$ clones after running VICTree on a DLP+ dataset. On the right side, the top-three most probable trees are shown, with their corresponding probabilities, which are computed by self-normalizing a sample of 10^4 trees. The matrix exhibits strongly skewed edge weights for which standard sampling algorithms would fail. 62

Acronyms

aCGH array Comparative Genomic Hybridization
ACT Acoustic Cell Tagmentation
BAF B-allele frequency
CAVI Coordinate Ascent Variational Inference
CBS Circular Binary Segmentation
CNA copy-number aberration
DAG directed acyclic graph
DLCA deepest least common ancestor
DLP Direct Library Preparation
ELBO Evidence Lower Bound
HMC Hamiltonian Monte Carlo
HMM Hidden Markov Model
ILP Integer Linear Programming
IS importance sampling
ITH intra-tumor heterogeneity
JC69 Jukes-Cantor 1969 model
KL Kullback-Leibler
LERW Loop-Erased Random Walk
LOH loss of heterozygosity
MAP maximum a posteriori
MCMC Markov Chain Monte Carlo
MH Metropolis-Hastings
MLE maximum likelihood estimation
MST maximum spanning tree
NJ Neighbor Joining
PCR Polymerase Chain Reaction
RF Robinson-Foulds
SBN Subsplit Bayesian Network
scDNA-seq single-cell DNA sequencing
SMC Sequential Monte Carlo
SNJ Sparse Neighbor Joining
SNV single-nucleotide variant
SPR Subtree Pruning and Regrafting
TSSB Tree-Structured Stick-Breaking
VI Variational Inference
VMP Variational Message Passing
WGA Whole-Genome Amplification
WGD whole-genome doubling

Introduction

1 Introduction

Biological cells are the fundamental units of life. Every living organism is made of a number of cells that live, grow, and divide. Despite the enormous diversity of life, all cells share a common molecular basis: DNA. This molecule stores the genetic information that directs cellular activities and inheritance by encoding the instructions for building and maintaining an organism. Its ability to replicate makes this molecular information survive the natural degradation of matter and enables the transmission of traits across generations. When a cell divides, it copies its DNA and passes it on to its daughter cells. This process is subject to errors, which can lead to mutations that may be beneficial, neutral, or harmful. Over time, these mutations accumulate and drive the evolution of species.

Life as we observe it today is the result of billions of years of evolution, shaped by the complex interplay of genetic variation and natural selection. Multicellular organisms, like plants and animals, consist of specialized cells that perform different functions. This specialization is guided by the regulation of gene expression, which is influenced by the DNA sequence and its epigenetic modifications. Through coordinated cellular activity, these organisms perform complex tasks such as movement, reproduction, and response to environmental stimuli, and therefore shape entire ecosystems. Everything in nature, from the environment we live in to the very reason we exist, is therefore linked to the carefully arranged sequence of nucleotides in the DNA molecule.

In humans, the whole genome is made up of more than 3 billion nucleotides, organized into 23 structural units called chromosomes. Each chromosome contains many genes, each of which encodes a specific protein that performs a particular function in the cell. The human genome is remarkably similar across individuals, with 99.9% of the DNA sequence being identical. Moreover, a large portion of the human genome is made up of non-coding regions, which do not encode proteins. The small fraction that varies between individuals is responsible for the diversity of human traits, like eye color and height. These variations are inherited from our parents, but they are not the only source of genetic diversity. As mentioned before, cell division is not perfect, and errors can

happen in various tissues of the body, leading to somatic mutations that fundamentally change the function of the affected cells.

Most of these mutations are harmless because the human body has mechanisms to repair DNA damage and eliminate malfunctioning cells, so that broken information does not propagate and the organism can continue to function properly. However, some mutations can give cells a growth advantage, allowing them to divide more rapidly and evade the body's defenses. Abnormal or damaged cells that grow uncontrollably can form tumors, which may either stay confined to a specific area, without interfering with the normal function of the body, or spread to other parts of the body, causing severe health problems. In the latter case, we say that the tumor is malignant, and it is commonly referred to as cancer.

Cancer can arise in any tissue of the body and therefore encompasses a wide variety of diseases, each with its own characteristics and challenges. Cancer is a leading cause of death worldwide, accounting for nearly 10 million deaths in 2020 alone [5]. Cancer can arise from a combination of inherited susceptibility, environmental exposures, and lifestyle factors. For example, exposure to carcinogens, such as tobacco smoke and ultraviolet radiation, can damage DNA and increase the risk of developing cancer. Additionally, inherited genetic mutations can predispose individuals to certain types of cancer. Although many drivers are known, predicting exactly when and how a tumor will emerge and evolve remains difficult, and research is ongoing to identify the underlying mechanisms that drive this disease.

Since decades, medicine has been advancing our understanding of cancer and developing new treatments, leading to improved survival rates for many types of cancer. Malignant tumors can be treated with a combination of surgery, chemotherapy, radiation therapy, and targeted therapies, depending on the type and stage of the cancer. Understanding the genetic basis of cancer has been crucial for the development of targeted therapies like immunotherapy, which harnesses the body's immune system to fight cancer by targeting specific molecules on the surface of cancer cells.

However, one of the biggest challenges in treating cancer is its heterogeneity, which makes it difficult to predict how a tumor will respond to treatment and to develop effective therapies [6, 7, 8]. Here, *tumor heterogeneity* refers to differences in genetic and phenotypic characteristics among cells within the same tumor, which can lead to variation in their behavior and response to treatment. This heterogeneity can arise from the accumulation of mutations over time, as well as from interactions with the tumor microenvironment. As a result, even within a single tumor, there can be multiple subpopulations of cells with distinct genetic profiles and drug sensitivities, making it difficult to develop effective treatments that target all of the cancer cells.

To address this challenge, researchers have developed various sequencing technologies that allow us to analyze the genetic makeup of tumors at a high resolution. These technologies have revolutionized our understanding of cancer and have provided insights into the mechanisms that drive tumor heterogeneity and evolution. More specifically, they have enabled us to effectively read the DNA sequence of tumors and identify the mutations that are present in different subpopulations of cells. Unfortunately, the data generated by these sequencing technologies are often noisy and complex, making it difficult to interpret and extract meaningful information. Moreover, it is not enough to

simply identify the mutations that are present in a tumor; we also need to understand how these mutations are related to each other and how they have evolved over time. The level and variety of biological processes that govern tumor evolution are far from being fully understood, and even though new technologies provide us with an increasingly detailed and abundant view of the tumor genome, we still need to develop computational methods that can help us make sense of this data.

1.1 Single-cell DNA sequencing and copy-number aberrations

There are two main categories of alterations: single-nucleotide variants (SNVs), which involve changes to a single nucleotide, and copy-number aberrations (CNAs), which describe changes involving larger segments of DNA, such as deletions, duplications, and amplifications. Conceptually, an SNV is a single-base substitution, while a CNA is a deletion or amplification of a longer genomic segment. Both can strongly affect cellular function and can contribute to oncogenesis. Cancers like breast, ovarian, and lung cancer are often characterized by a high frequency of CNAs, which can lead to the loss of genes that suppress tumor growth or the amplification of genes that promote cell division. Bulk sequencing technologies have been widely used to characterize the genetic landscape of tumors by analyzing DNA from a pool of tumor cells. Although computational methods have been developed to deconvolve the data and identify subpopulations of cells from bulk sequencing data, the presence of multiple subpopulations can confound the analysis, masking the true genetic diversity of the tumor. Single-cell DNA sequencing (scDNA-seq) technologies have emerged as a powerful tool to overcome this limitation by allowing us to analyze the DNA of individual cells.

This has provided unprecedented insights into the clonal structure and evolutionary history of tumors, allowing to measure the specific contributions of each single cell to the whole tumor and to identify rare subpopulations that are inevitably missed by bulk sequencing. For example, scDNA-seq has revealed the presence of remarkably diverse subpopulations of cells within tumors, suggesting that tumors progress in bursts of clonal expansion, where a mutated genome from one single cell rapidly expands and accumulates additional mutations [9]. This has challenged the traditional view of tumor evolution as a gradual process (e.g., Nowell's theory of clonal evolution [10]) and has highlighted the need for new models and methods to study cancer evolution. Unfortunately, sequencing single cells individually presents technical challenges, which lead to noisy and sparse data that can be difficult to interpret. The amount of genetic material in a single cell is extremely small, and the process of amplifying the DNA can introduce errors and biases that can affect the accuracy of the data. Without amplifications, the proportion of the genome that is successfully sequenced is significantly low, leading to large gaps in the data. This is particularly problematic when analysing point mutations, which are often missed due to the low coverage. On the other hand, copy-number aberrations, which affect larger segments of DNA, can be more reliably detected even with low coverage, making them particularly informative for reconstructing tumor evolution.

Every human cell normally contains two copies of each chromosome, one inherited from the mother and one from the father; therefore, healthy human somatic cells are normally *diploid*. During replication, a cell copies its DNA, producing two identical copies of each chromosome, which will then be distributed to the daughter cells. However, errors

can occur during this process, leading to changes in the number of copies of certain segments of DNA. For example, after a pair of chromosomes is duplicated, the four copies may be distributed unevenly to the daughter cells, resulting in one cell with three copies of a chromosome and another cell with only one copy. While such abnormal cells are often identified as damaged and eliminated by quality-control mechanisms, including immune surveillance and apoptosis, some may evade this process, survive, and continue to divide, thereby passing on the abnormal copy number to their descendants. Depending on the specific area of the genome that is affected, copy-number aberrations can have different effects on cellular function, most of which are still not fully understood. More generally, it has been observed that the copy number of genes is often correlated with their expression levels, that is, the more copies of a gene there are, the more of its corresponding protein is produced [11]. This is of course only relevant when looking at the coding part of the genome, which is only the 2% of the whole genome, but copy-number aberrations can happen anywhere in the genome. CNAs that affect non-coding regions can still have significant effects on cellular function, but they also constitute a large source of genetic diversity which provides useful information to trace the evolutionary history of subpopulations of cells within a tumor. By analyzing the copy-number profiles emerging from single-cell DNA sequencing data, we can therefore gain insights into the nature of the tumor of a patient, how it originated, the way it has evolved, and how it may respond to treatment.

1.2 Inference for phylogenetics

The history of a tumor is marked by the process of cell division. If we imagine tracing multiple generations of offspring cells in time, starting from a single healthy diploid ancestor cell, we would look at a tree-like structure where every branch draws the lifespan of a cell, and when that cell divides, two new branches form. Occasionally, a mutation may occur, and the cell that carries that mutation either dies, leaving a dangling branch, or survives and continues to divide, passing on the mutation to its descendants, with the potential to acquire additional mutations along the way, which may lead to the formation of a tumor. Over time, the accumulation of mutations and the process of cell division give rise to multiple subpopulations of cells with distinct genetic profiles. This process of evolution is very similar to the evolution of species, where a common ancestor gives rise to multiple descendant species through a process of speciation, driven by genetic variation and natural selection. It is now established, for instance, that human beings share a common ancestor with chimpanzees, and that millions of years ago they diverged from a species of hominids that lived in Africa [12]. This fact is not only revealed by fossil evidence, but also by the analysis of the DNA of different species and the reconstruction of their phylogenetic tree informed by statistical models of evolution.

In classical phylogenetics, the goal is typically to determine the most likely tree that explains the observed DNA sequences of different species. These observations are the leaves of the tree, and the internal nodes represent the common ancestors, which may be extinct as of today. The ancestral sequences, although unknown, are likely to be correlated with the observed sequences descending from them, therefore shared patterns of mutations between the observed sequences can be used to infer the structure of the tree and the ancestral sequences. A probabilistic model of evolution describes

how DNA sequences change over time, providing a statistical framework to solve the problem of phylogenetic inference. Several models of evolution have been developed, such as the well-known Jukes-Cantor substitution model [13], which assumes that the only mutation that can occur is a change from one nucleotide to another, and that all nucleotides are equally likely to mutate, an assumption that is useful analytically but biologically simplified.

The model alone is not enough to reconstruct the tree, and different approaches have been developed to this end. One common approach is to compute evolutionary distances between the observed sequences, and then apply algorithms like neighbor-joining [14] to reconstruct the tree by optimizing some criterion, such as the minimum total branch length. Another approach is to find the tree that maximizes the likelihood of the observed data under the model of evolution; typically, this is done by searching through the space of possible trees by applying local modifications to the tree structure, and testing whether the likelihood of the data increases or decreases after each modification. Despite being more computationally intensive, a maximum likelihood approach is preferred because it directly aims to find the tree that best explains the data according to the evolution model of choice. However, both of these approaches only provide a single tree as the output, which may not be representative of the true evolutionary history, especially once we take into account the noise and uncertainty in the data. Moreover, even the most sophisticated models of evolution are still simplified representations of the true underlying process, and therefore they may introduce biases in the inference. Bayesian inference treats the tree as a random variable and aims to find the probability distribution over trees given the observed data, thereby explicitly quantifying topological uncertainty. The Bayesian framework also allows to incorporate prior knowledge about the tree structure and the model parameters, serving as a regularization mechanism to guide the inference towards more biologically plausible solutions. Ideally, the posterior distribution better captures the uncertainty in the data and the model, because it does not just provide the best explanation for the data, but rather quantifies how likely every possible explanation is, given the data and the model. This is particularly important in the context of tumor evolution, where the data is often noisy and incomplete, and the underlying evolutionary processes are complex and not fully understood. Uncertainty quantification is also crucial for clinical applications, where the inferred tree may be used to guide treatment decisions.

1.3 Research questions

The advent of single-cell DNA sequencing has opened unprecedented opportunities to study tumor evolution at the cellular level. However, translating the raw, noisy data into meaningful evolutionary history requires overcoming significant computational and statistical challenges. Traditional phylogenetic models were largely designed for nucleotide sequences of distinct species, which evolve relatively slowly and uniformly. In contrast, single-cell copy-number evolution within tumors is highly erratic, characterized by episodic structural change and high intratumor heterogeneity. At the same time, the sequencing technology itself introduces pervasive measurement noise, such as allele dropout and uneven read coverage. The combination of these factors highlights a critical research gap: the need for models specifically tailored to handle both the complex biological nature of copy-number evolution and the noisy reality of single-cell

sequencing data.

A key challenge is that the true copy-number states of a cell are not directly observed; rather, they are latent variables that must be inferred from the noisy read counts. Furthermore, copy-number changes are not independent across the genome; a single mutational event can alter the copy number of a large chromosomal segment, creating strong spatial dependencies among neighboring genomic regions. Hidden Markov Models (HMMs) provide a natural mathematical framework to capture these spatial dependencies, modeling the true copy-number profile as a sequence of hidden states. Applying HMMs to single-cell data therefore calls for robust inference procedures to estimate the model parameters while carefully accounting for the noise profile of the sequencing technology through appropriate observation models, such as Poisson or Negative Binomial distributions.

In stepwise pipelines, once latent copy-number profiles are estimated, reconstructing the evolutionary tree of the tumor presents another major hurdle. Classical distance-based methods, such as standard neighbor-joining, often struggle with tumor data because they do not naturally incorporate the biological constraint that all tumor cells descend from a healthy, diploid ancestral cell. Furthermore, the assumption of constant evolutionary rates is frequently violated by the punctuated nature of cancer evolution. This necessitates the development of specialized distance metrics and rooted tree reconstruction algorithms that can handle high heterogeneity and properly anchor the phylogeny to the diploid ancestor.

Currently, many methods in the field split the problem of phylogenetic inference into separate sequential steps: first calling copy-number profiles, then clustering cells into clones, and finally reconstructing a tree. This stepwise approach is often suboptimal because errors or uncertainties in the early steps propagate to subsequent stages and typically cannot be corrected later. Jointly inferring the copy-number states, the clonal structure, and the evolutionary tree would naturally account for this uncertainty. However, joint inference is extremely challenging from both computational and modeling perspectives.

Even with improved reconstruction algorithms, finding a single best tree is often insufficient due to the inherent ambiguity in the noisy single-cell data. The space of possible evolutionary trees is extraordinarily large, and the distribution of tree probabilities is typically highly skewed and multi-modal. Standard tree sampling techniques, such as Markov Chain Monte Carlo (MCMC), often suffer from poor mixing and slow convergence when exploring these complex posterior landscapes. To achieve scalable and accurate posterior estimation, alternative inference strategies must be explored. Variational inference offers a promising avenue by casting the posterior inference problem as an optimization task, approximating the complex posterior with a simpler, tractable distribution.

Driven by these methodological gaps, this thesis addresses the following core research questions:

Core Research Questions

- **Question 1:** How can noisy single-cell DNA sequencing data be translated into biologically meaningful copy-number profiles, accounting for both spatial genomic dependencies and technology-specific measurement noise?
- **Question 2:** How can we properly leverage the biological constraint of a healthy diploid ancestor, alongside specialized distance metrics, to reconstruct accurate rooted phylogenetic trees despite punctuated evolution and high intratumor heterogeneity?
- **Question 3:** How can we move beyond sequential pipeline approaches to jointly and scalably infer clonal structure, copy-number evolution, and tree uncertainty using advanced Bayesian approximation techniques?

1.4 Thesis outline

This thesis is organized into seven chapters that guide the reader from the statistical and biological foundations of the field, through methodological developments, to practical applications in computational oncology.

Chapter 2 reviews the necessary background, introducing the separate biological and computational concepts—such as single-cell sequencing, probabilistic modeling, and phylogenetic inference.

Chapter 3 addresses the first research question. Building on existing literature for copy-number calling, it introduces the mathematical framework of Hidden Markov Models to infer latent copy-number states and details observation models for single-cell noise.

Chapter 4 tackles the second research question. After reviewing existing distance-based phylogenetic methods, it proposes specialized metrics and rooted reconstruction algorithms that properly anchor the tree to a diploid ancestor.

Chapter 5 explores the third research question. It reviews sequential pipeline approaches before presenting advanced techniques for joint inference, comparing the trade-offs of Bayesian inference methodologies.

Chapter 6 provides a synthetic summary of the four papers included in this thesis, detailing how their individual contributions answer the core research questions.

Finally, **Chapter 7** draws the main conclusions from the research, critically discusses its limitations, and outlines promising directions for future work.

2 Background

Tumor phylogenetics is an inherently interdisciplinary field, drawing from genomics, statistics, and computer science. This chapter provides a comprehensive overview of the key concepts and methodological tools that underpin the research presented in this thesis. Like for any data-driven study, it is convenient to start by describing the data, what it represents, the way it is collected, and the natural phenomena that generate it. The first section introduces the key concepts of biology and genomics relevant to understand the characteristics of single-cell sequencing data. The second section establishes the grounds for statistical inference, briefly covering basic probability concepts for the sake of notation and terminology, then expanding on modeling data and perform inference. The third section provides an introduction to phylogenetics, the various approaches for reconstructing trees from genomic data, and the common similarity measures used to quantify distances between them. The fourth section covers the problem of generating trees at random; trees are combinatorial structures whose properties differ fundamentally from those of standard numerical objects such as vectors or scalars, necessitating dedicated sampling and exploration methods. Finally, the last section summarizes the research gaps that motivate the work presented in this thesis, opening the way for a more in-depth analysis of applied methods contained in the next chapters.

2.1 Genomics

All living organisms are made up of cells. They are responsible for carrying out the most essential functions of life, such as converting nutrients into viable energy, a process referred to as metabolism, and dividing to create new cells. Not all cells are equal, but they share several common characteristics. Each cell is enclosed in a membrane, which separates the interior from the outside environment, controlling the flow of substances in (nutrients) and out (waste). Inside, various components carry out specific functions dictated by the cell's type and state. It was not until 1928 [15], that the DNA molecule was first identified as the responsible for encoding the instructions needed to build and

maintain an organism, changing the common understanding of heredity which until then was thought to be mediated by proteins. This discovery was later confirmed in 1951 by an experiment conducted by two scientists who observed that bacteriophages, viruses that infect bacteria, only transferred DNA and not proteins into the host cells, providing strong evidence that DNA is the genetic material [16]. Since then, the interest in understanding the fundamental role played in all life forms by the apparently inert DNA molecule has only increased, leading to the more recent milestone of sequencing the human genome in 2001 [17], which allowed scientists to read the complete set of instructions encoded in our DNA, opening the way for a deeper understanding of human biology. Decoding the human genome is still an active area of research, as we are still trying to understand how the information encoded in the DNA sequence translates into the complex and dynamic processes that govern the functioning of cells, tissues, and ultimately the whole organism. This section contains an overview of the key concepts in genomics that are relevant for understanding the research questions addressed in this thesis. While it is in the author's intention to provide an accurate description of the biological terms and processes involved, the biologist reader will excuse any oversimplifications as the scope of this thesis mainly lies in the computational and mathematical aspects of tumor phylogenetics.

2.1.1 The DNA molecule

The deoxyribonucleic acid (DNA) molecule is a long, double-stranded helix composed of two complementary strands that are held together by hydrogen bonds between pairs of nitrogenous bases. The bases are of four types: adenine (A), thymine (T), cytosine (C), and guanine (G). The order in which these bases are arranged along the DNA forms the *genome*, the genetic code that contains all the instructions required by organisms to live and reproduce. It should not be surprising that such a vast amount of information requires the molecule to be extremely long. In fact, if we were to stretch the DNA from a human single cell out in a straight line, it would be about 2 meters long, which is around 10^7 times longer than the cell's diameter. To fit inside the cell, the DNA is tightly packed into structures called *chromosomes*. These are formed by the DNA molecule wrapping around proteins called histones, creating a complex called chromatin (see Figure 2.1.1).

Each chromosome can be further divided into functional units called *genes*, which are segments of DNA that encode for specific proteins, the workhorses of a cell. However, not all the DNA sequence encodes for proteins; in fact, only about 1-2% of the human genome consists of protein-coding genes (exons), while the rest is made up of non-coding regions (introns). When a cell needs to use the information encoded in the DNA, it first converts the relevant gene into a molecule called RNA through a process called transcription. The RNA transcript then travels out of the nucleus (the compartment where the DNA is stored) into the cytoplasm, where it is translated into proteins by molecular machines called ribosomes. The ribosomes read the RNA sequence in groups of three bases, called codons, each of which corresponds to a specific amino acid, the building blocks of proteins. We know, for instance, that the codon GCC corresponds to the amino acid alanine, while the codon GGC corresponds to glycine. Cells can regulate gene expression at multiple levels, controlling when and how much of a particular protein is produced, allowing them to respond to changes in their environment and

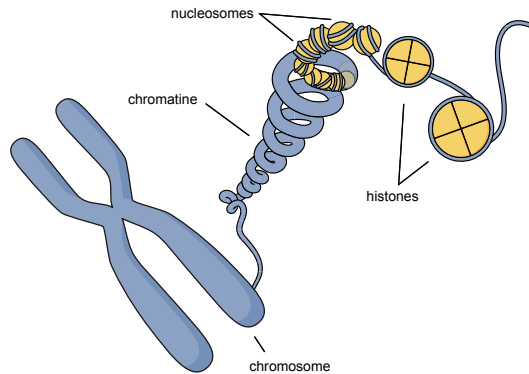


Figure 2.1.1: Illustration of the packing of the DNA molecule into chromosomes. The DNA double helix wraps around histone proteins to form nucleosomes, which further coil and fold to create the compact structure of a chromosome.

maintain an equilibrium state (homeostasis). By doing so, they differentiate into various types, each with specific functions and characteristics, such as muscle cells, nerve cells, and blood cells, among many others. For this reason, while the DNA sequence is the same in all cells of an organism, the RNA transcripts and therefore the proteins produced can vary greatly. Transcription factors, which are proteins that bind to specific DNA sequences, work together with other regulatory elements to control gene expression, and by that, the entire cell cycle, including division, growth, and death.

The lifespan of a cell is not infinite; whether due to programmed cell death (apoptosis) or external factor, the cell will eventually die. To ensure the survival of the organism, cells divide. There are two main types of cell division: mitosis and meiosis. Mitosis is the process by which a non-reproductive (somatic) cell divides into two identical daughter cells, each containing the same genetic material as the parent cell; the DNA is replicated and then equally distributed between the offspring cells. Meiosis, on the other hand, is a specialized form of cell division that occurs in the production of gametes (sperm and egg cells), where the DNA is replicated once but the cell divides twice, resulting in four non-identical daughter cells, each containing half the genetic material of the parent cell. In all sexually reproducing organisms, including humans, the genome is inherited from both parents, carrying twice the amount of genetic material found in gametes. In fact, a human cell contains 23 pairs of chromosomes, for a total of 46. These cells are called *diploid*, and the two copies of each chromosome are called homologous chromosomes. Note however that not all organisms are diploid; bacteria, for instance, are typically *haploid*, meaning they have only one copy of each chromosome, while some plants can be *polyploid*, having more than two copies of each chromosome. Although DNA is copied and transmitted with high fidelity, small changes in its sequence, both between

individuals and within the cells of a single organism, give rise to genetic variation, that is the basis for evolution and the diversity of life.

2.1.2 Genetic variation and somatic evolution

Genetic information is not perfectly identical across individuals; small variations in the DNA sequence contribute to diversity within a species. One of the most common forms of inherited variation is the *single nucleotide polymorphism* (SNP), which consists of a change in a single base at a specific position of the genome. For example, one individual may have an A where another has a G. Although many SNPs occur in non-coding regions and have no apparent effect, others can influence how genes are expressed or alter the structure and function of the proteins they encode. Typically, a nucleotide change is considered a SNP if it is present in a sufficiently large fraction of the population (e.g., at least 1%), distinguishing it from rare mutations that may occur in only a few individuals. In humans, scientists have identified more than 600 million SNPs and each person carries about 4-5 million of them. SNPs at a given genomic position (or *locus*) define different *alleles*, that is, alternative versions of the same DNA sequence found in a population. While many SNPs are biallelic (involving only two possible nucleotides), in principle, a single position could exhibit more than two variants, leading to multiple alleles at that locus. The set of alleles present in an individual is called their *genotype*, and the observable traits that result from the interaction of the genotype with the environment are called the *phenotype*. The relationship between genotype and phenotype is complex and object of intense research. As we already mentioned, the genotype of a sexually reproducing organism is determined by the combination of the two genomes inherited from the parents, which are therefore referred to as the maternal and paternal alleles. Thanks to a process called recombination, the alleles from the two parents are shuffled during the formation of gametes, and the newborn child is left with a unique combination of these alleles. This is why siblings can look different from each other, even though they share the same parents.

While SNPs represent variations inherited from our parents, new changes can also arise during an organism's lifetime. Every time a cell divides, it must copy roughly 3 billion base pairs of DNA. Even with the cell's sophisticated self-defense mechanisms designed to repair DNA damage, copying is not flawless and errors occasionally occur. These acquired changes are called *somatic mutations*, the most basic of which is the *single nucleotide variant* (SNV), a change in a single base of the DNA. In the context of cancer, most somatic mutations are biologically neutral and do not affect cellular fitness; these are called *passenger* mutations. However, if an SNV happens to strike a critical gene responsible for regulating cell division or DNA repair, it can act as a *driver* mutation. When the strict controls governing the cell cycle break down, the mutant cell can begin dividing uncontrollably. This unchecked proliferation leads to the formation of a mass of cells called a tumor. A tumor that remains localized may still be clinically dangerous because of its size or anatomical location. In oncology, malignancy is primarily defined by the ability of cells to invade surrounding tissues and, in many cases, spread to distant sites. When these cells acquire the ability to invade surrounding tissues and travel to distant parts of the body (in a process called metastasis) that the tumor is considered malignant, and the disease is classified as cancer.

Point mutations are not the only type of somatic mutation. During abnormal cell

divisions, especially in the chaotic environment of a tumor, alterations that affect large genomic segments can occur. Entire sections of DNA, ranging from a few thousand base pairs to whole chromosomes, can be duplicated or deleted. These large-scale changes are known as *copy-number aberrations* (CNAs), which are part of a broader category of mutations called *structural variants* (SVs). When a genomic segment is amplified, the newly created copies must reside somewhere within the nucleus. Sometimes, the extra copies are inserted directly adjacent to the original sequence, forming a tandem duplication on the same chromosome. In other cases, the copied material can be inserted at a different genomic location. Furthermore, these duplication or deletion events typically happen on just one of the two homologous chromosomes. This means that a CNA usually affects either the maternal or the paternal allele, leading to a phenomenon known as allelic imbalance, where the natural 1:1 ratio between inherited alleles is broken. To give an example, imagine a duplicating cell that accidentally splits the four copies of a gene unevenly, giving three copies to one daughter cell and only one copy to the other. The first cell still has both versions of the gene (the maternal and paternal alleles), but the second cell has lost one of them, resulting in what is called a *loss of heterozygosity* (LOH) (Figure 2.1.2). Normally, the two alleles at a locus can partially compensate for each other if one is damaged or mutated, but in the case of LOH, this buffering effect is lost, which can have significant consequences for the cell's function and survival. Be it as it may, the quantity of certain specific portions of the genetic material of a cell can change from its normal amount (i.e., two copies for diploid organisms).

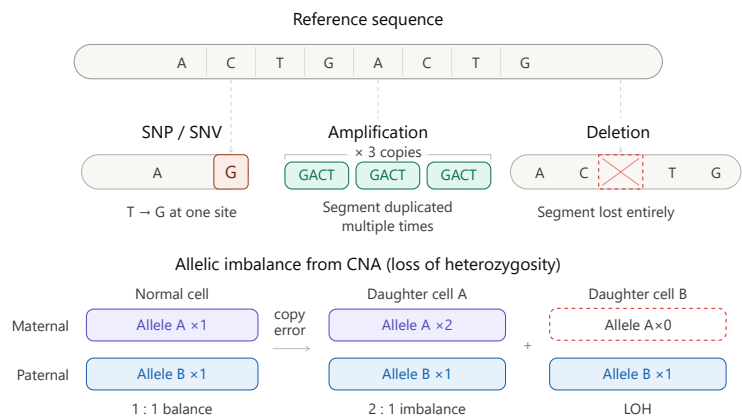


Figure 2.1.2: Types of genetic variation. (Top) A reference DNA sequence illustrating three mutation types: an SNP/SNV (single base substitution, highlighted in red); a CNA amplification (a segment duplicated to three tandem copies); and a segment deletion. (Bottom) Allelic imbalance arising from a copy error during cell division. A normal cell carries one maternal (purple) and one paternal (blue) allele in a 1:1 ratio. After an unequal division, one daughter cell receives two copies of allele A (2:1 imbalance), while the other loses allele B entirely — a loss of heterozygosity (LOH).

Many solid epithelial tumors, like breast, ovarian, and colorectal cancers, are characterized by *aneuploidy* (changes in copy numbers) [18]. Tumors with many CNAs, as in the case of *whole-genome duplication* (WGD) events, are often more unstable and aggressive, and patients with such tumors tend to have a worse prognosis [19, 20]. The *chromosomal instability* (CIN) of these tumors, term used to describe the high rate of copy-number changes, represents a major challenge for treatment of cancer, as it generates highly heterogeneous populations of cells within the same tumor, which then becomes hard to isolate and neutralize. Recent cutting-edge clinical trials are now testing the efficacy of personalized treatments that target cells with specific cancer driving mutations, effectively improving the prognosis of patients with aggressive tumors [21]. Sequencing the genome of a tumor is therefore no longer just descriptive, it actively informs targeted therapies and early intervention. This is the main reason why new technologies for sequencing the DNA of tumors at single-cell resolution are being developed and refined at a rapid pace.

2.1.3 Sequencing technologies

To understand the genomic variations driving a tumor, scientists must first extract and “read” the genetic material. At a high level, DNA sequencing works by taking long, continuous strands of DNA and breaking them into millions of smaller, manageable fragments. Specialized machines then chemically analyze these fragments base by base (A, C, G, T), converting the physical molecules into digital text strings known as *reads*. These short reads are finally mapped back to a standard reference genome like pieces of a massive jigsaw puzzle, revealing the exact location and nature of any mutations.

The field of genomics was born with first-generation techniques like Sanger sequencing [22], which were highly accurate but slow and expensive. A massive shift occurred in the mid-2000s with the advent of Next-Generation Sequencing (NGS) [23]. By parallelizing the sequencing process, NGS allowed millions of fragments to be read simultaneously, drastically reducing costs and paving the way for large scale milestones like the complete sequencing of the human genome.

As sequencing matured, researchers began to target not just DNA, but also RNA. The distinction between the two is crucial for understanding cell biology. DNA sequencing (DNA-seq) captures the cell’s static genetic blueprint, identifying the permanent mutations and structural aberrations inherited or acquired over time. Conversely, RNA sequencing (RNA-seq) provides a dynamic snapshot of the cell’s current state by measuring gene expression [24]—that is, which specific genes are actively being transcribed to build proteins at that exact moment.

Traditionally, tissues have been analyzed using *bulk sequencing*, where millions of cells are homogenized and sequenced together. This approach enabled the generation and the analysis of large datasets, leading to landmark projects like The Cancer Genome Atlas (TCGA) [25], a catalogue of genomic profiles for thousands of tumors across dozens of cancer types. However, bulk sequencing has a critical limitation: it averages the signal across all cells, masking the underlying heterogeneity. To make an analogy, it is like blending together a number of fruits, in different proportions and then trying to identify the quantity and quality of any specific fruit in the mix. Even the most experienced taster would struggle in this task, and of course, this example is only a

simplification of the complex genomic landscape of a tumor. Single-cell sequencing was later developed to solve this by physically isolating individual cells before reading their DNA [9]. While early single-cell methods required Whole-Genome Amplification (WGA), which introduces coverage non-uniformity, allelic imbalance artifacts, and amplification noise. These limitations are a major reason why copy-number signals, which are robust to low coverage, became central in single-cell tumor phylogenetics. Modern scalable technologies such as Direct Library Preparation (DLP/DLP+) [26, 27] and Acoustic Cell Tagmentation (ACT) [28] bypass preamplification entirely. This allows researchers to accurately read the genome of thousands of individual cells with high fidelity.

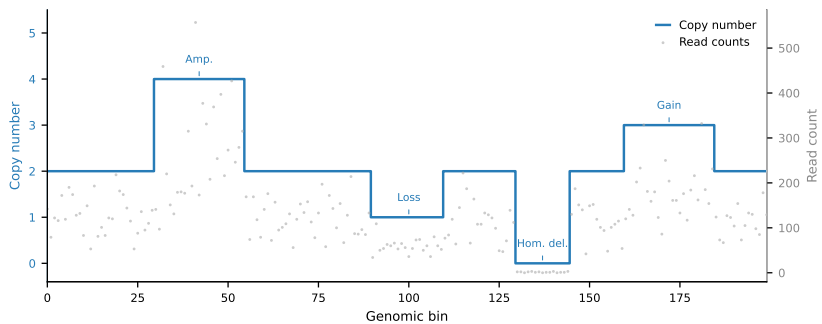


Figure 2.1.3: Read counts and underlying copy-number profile across genomic bins. Grey dots show the raw read count observed in each fixed-size bin for a simulated single cell, drawn from a Negative Binomial distribution whose mean is proportional to the local copy number. The blue step line indicates the true integer copy-number state (0–5). Four copy-number events are highlighted: a two-copy amplification (Amp.), a single-copy gain (Gain), a single-copy loss (Loss), and a homozygous deletion (Hom. del.). The spread of read counts around the expected mean reflects the overdispersion intrinsic to single-cell sequencing data, illustrating why computational methods must explicitly account for this noise when inferring copy-number states from raw read counts.

Analyzing these single-cell datasets requires dedicated computational approaches, both because of the scale of the data and the noise inherent in the sequencing process. For copy-number aberrations, the genome is typically divided into fixed-size, non-overlapping windows called *bins*. The fundamental data point is the *read count*—the number of sequencing reads that align to each specific bin. In a healthy diploid cell, the read count should be roughly uniform across the genome. If a bin shows a sudden spike or drop in reads, it provides direct statistical evidence of a copy-number amplification or deletion (see Figure 2.1.3). However, this raw data is heavily corrupted by noise [29]. Certain regions of the genome might fail to be sequenced (dropout), reads might map to the wrong locations, and biochemical biases (like GC-content bias) can artificially warp read counts. Bioinformaticians have developed sophisticated normalization techniques to correct for these artifacts, but the resulting data is still a noisy, incomplete proxy for the true underlying state of the cell’s genome. For this reason, computational methods for tumor phylogenetics must be designed to account for the inherent uncertainty in the data and the number of latent factors that, while not directly observed, are crucial for understanding the evolutionary history of the tumor.

2.2 Probabilistic modeling and inference

All models are wrong, but some are useful.

George E. P. Box

At the heart of any quantitative science lies the concept of a model—a mathematical abstraction of a real-world process that manifests itself in some form, which we can measure, or observe, to collect data. Whether predicting the weather or tracing the evolution of a tumor, models allow us to formalize our assumptions, make predictions, and understand the hidden mechanics of a system. In the context of single-cell DNA sequencing, the data we observe during a sequencing experiment is a noisy matrix of read counts. This is the *observed* data. The biological phenomena we actually care to understand, such as the true copy-number profile of a cell or the phylogenetic tree connecting different clones, are not directly observed¹. We refer to these unobserved quantities as *latent variables*. The central challenge of this thesis is the problem of inference: how can we use the observed data to learn about the latent variables, and therefore the underlying biology of the tumor? To answer this question, we need a mathematical framework that can handle uncertainty while being efficient enough for practical use given the scale and complexity of the current single-cell datasets.

This section introduces the key concepts and tools of probabilistic modeling and inference that we will rely on throughout the thesis. First, we review some basic notions of probability and statistics, mainly to set the notation and terminology. Then, we discuss how to represent complex dependencies between observed and latent variables using probabilistic graphical models. Finally, we explore the inference techniques related to this thesis, ranging from maximum likelihood estimation to fully Bayesian methods and variational inference.

2.2.1 Notions of probability and statistics

To handle uncertainty we rely on probability theory [30, 31]. A fundamental concept in probabilistic modeling is the random variable, which represents an unknown quantity whose exact value is subject to chance. Formally, a random variable X is a measurable function from a sample space Ω to a measurable space, e.g., the real numbers $\mathbb{R}$. The sample space Ω represents the set of all possible outcomes of an underlying experiment.

Rather than holding a single fixed value, X is characterized by a probability distribution that assigns a likelihood to each of its possible outcomes. If X is continuous, this is described by a probability density function (PDF) $p_X(x)$, such that the probability of X falling within an interval $[a, b]$ is given by $\int_a^b p_X(x)dx$. If X is discrete, it is described

¹Techniques such as fluorescence in situ hybridization (FISH) and controlled *in-vitro* or xenograft lineage tracing can, in specific experimental settings, provide approximate measurements of copy number and clonal relationships. In the modeling framework considered here, these quantities are considered unobserved.

by a probability mass function (PMF). Throughout this thesis, we generally use the notation $p_X(\cdot)$ interchangeably for both densities and masses, relying on the context to distinguish between them. We also contract the subscript when the variable over which the distribution is defined is clear from the context, writing simply $p(x)$.

When dealing with multiple variables, we define their joint distribution $p(x, y)$. If we observe the value of $Y = y$, our belief about X is updated according to the conditional probability $p(x | y) = \frac{p(x, y)}{p(y)}$. This relationship is famously formalized by Bayes' theorem:

$$p(x | y) = \frac{p(y | x)p(x)}{p(y)} = \frac{p(y | x)p(x)}{\int p(y | x')p(x')dx'} . \quad (2.2.1)$$

We denote observed data as y , unobserved latent variables as x , and the fixed parameters of our model as θ . The probability of observing the data y given a specific latent state x and parameters θ is written as $p(y | x, \theta)$. When viewed as a function of θ for a fixed observation y , this is known as the *likelihood function*.

2.2.2 Complex dependencies and latent variables

Latent variables do more than just represent unobserved biological entities; they act as hidden factors in the model that influence the observed data, allowing us to model complex dependencies and sequences. If we assume that each data point y_i is independent given the latent variable z_i , the joint distribution factorizes as $p(\mathbf{y}, \mathbf{z} | \theta) = \prod_{i=1}^N p(y_i | z_i, \theta)p(z_i | \theta)$. This factorization dramatically simplifies both the mathematical representation and the computational burden of the model.

As models grow in complexity, keeping track of conditional dependencies becomes cumbersome. Probabilistic graphical models offer a rigorous yet intuitive framework to represent these dependencies using graphs [32]. A Directed Acyclic Graph (DAG) $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consists of a set of nodes $\mathcal{V}$ representing variables in this model, and directed edges $\mathcal{E}$ representing conditional dependencies. Given a DAG, the joint distribution of all variables factorizes into the product of the conditional distributions of each node given its parents:

$$p(v_1, \dots, v_K) = \prod_{k=1}^K p(v_k | \text{pa}(v_k)) , \quad (2.2.2)$$

where $\text{pa}(v_k)$ denotes the set of parent nodes of v_k in the graph.

By convention, nodes representing observed data are shaded, while unshaded nodes represent latent variables or parameters. To concisely denote a set of independent and identically distributed (i.i.d.) variables, we use a rectangular bounding box known as a *plate*, annotated with the range of the index over which the variables are repeated.

Figure 2.2.4 visualizes a classical finite mixture model. For instance, consider a Poisson mixture model used to analyze the expression level $y_{ig} \in \mathbb{N}$ of G specific genes across N single cells. The latent variable $z_i \in \{1, \dots, K\}$ represents the unobserved cell type. The mixing proportions π , which sum to one ($\sum_{k=1}^K \pi_k = 1$), parametrize the Categorical distribution (also known as Multinoulli distribution), such that we can write $z_i \sim \text{Cat}(\pi)$, and $p(z_i = k | \pi) = \pi_k$. Conditioned on the cell type $z_i = k$, the gene expression

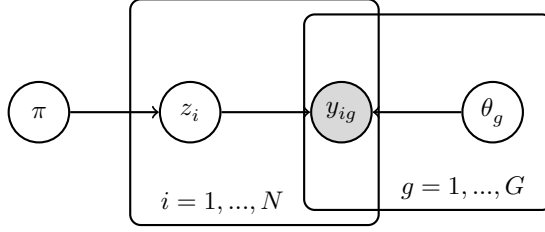


Figure 2.2.4: Directed Acyclic Graph (DAG) for a finite mixture model. The shaded node y_{ig} represents the observed data, while the unshaded node z_i is the latent cluster assignment. The parameters π and θ_g govern the categorical distribution and the emission distributions, respectively. The plates denote N independent observations and G genes.

follows a Poisson distribution $p(y_{ig} | z_i = k, \theta_g) = \text{Pois}(y_{ig}; \theta_{kg})$, where θ_{kg} is the mean expression for that cell type and gene.

Hidden Markov Models

A Hidden Markov Model (HMM) is a statistical model with unobservable (hidden) states that evolve sequentially over an index $m = 1, \dots, M$, such as discretized time or, in our case, positions along a DNA strand [33]. Let z_m denote the latent state at position m , and $\mathbf{z} = (z_1, \dots, z_M)$ the full sequence of latent variables.

The defining assumption of an HMM is the *Markov property*, which states that each state depends only on the previous one and is conditionally independent of all earlier states. Formally,

$$p(z_m | z_1, \dots, z_{m-1}) = p(z_m | z_{m-1}), \quad m = 2, \dots, M. \quad (2.2.3)$$

A stochastic process satisfying this property is called a *Markov process*. An HMM therefore assumes that the latent sequence $\mathbf{z}$ forms a Markov process, while the states themselves remain unobserved. Assuming that z_m takes values in a finite set of K possible states, the Markov property allows us to define the states transitions simply via a transition matrix $\mathbf{A}$, where each entry is $A_{ij} = p(z_m = j | z_{m-1} = i)$. The initial state is drawn from a categorical distribution π such that $p(z_1 = i) = \pi_i$. The observations $\mathbf{y} = (y_1, \dots, y_M)$ are generated conditionally independently given the current state, dictated by an emission distribution $p(y_m | z_m = i) = f(y_m; \phi_i)$, parameterized by ϕ .

The joint distribution of the complete sequence and the latent states then factorizes elegantly, as visualized in the DAG in Figure 2.2.5:

$$p(\mathbf{y}, \mathbf{z}) = p(z_1)p(y_1 | z_1) \prod_{m=2}^M p(z_m | z_{m-1})p(y_m | z_m). \quad (2.2.4)$$

Once the parameters are established, the primary goals in an HMM are evaluating the marginal likelihood of the data $p(\mathbf{y})$, computing the probability of a specific state at

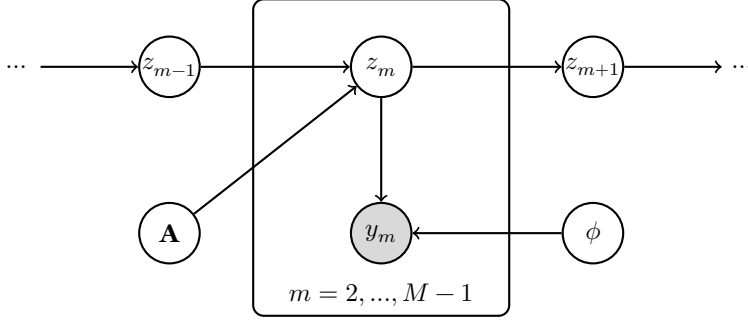


Figure 2.2.5: Directed Acyclic Graph for a Hidden Markov Model. The latent states z_m form a Markov chain, and each state emits an observation y_m . The initial state is governed by π (omitted), while transitions and emissions are controlled by A and ϕ respectively.

a given position given the observations up to that position, i.e., $p(z_m \mid y_{1:m})$ (filtering), estimating retrospective states $p(z_m \mid \mathbf{y})$ (smoothing), finding the most probable sequence of states $\arg \max p(\mathbf{z} \mid \mathbf{y})$ (decoding). Thanks to the Markov property, all these tasks can be solved efficiently using dynamic programming algorithms like the Forward-Backward algorithm and the Viterbi algorithm.

2.2.3 Inference

Inference is the engine that drives probabilistic modeling: it is the statistical process of fitting an abstract model to the observed data to draw conclusions about the latent variables and parameters. Depending on our goals and the complexity of the model, inference can mean finding a single best point estimate, mapping out the full landscape of uncertainty, or computing expected values of hidden states.

Maximum likelihood

One of the most fundamental approaches to inference is Maximum Likelihood Estimation (MLE) [34]. The logic is intuitive: we seek the parameters $\hat{\theta}$ that maximize the probability of the observed data under our model. Formally, this is expressed as an optimization problem over the log-likelihood:

$$\hat{\theta}_{\text{MLE}} = \arg \max_{\theta} \log p(\mathbf{y} \mid \theta). \quad (2.2.5)$$

Maximum likelihood approaches are typically straightforward, but they are prone to *overfitting* when applied to overly flexible models on limited data. As shown in Figure 2.2.6, an excessively complex model might achieve a very high likelihood by memorizing the noise in the observed data rather than capturing the underlying signal. To mitigate this, practitioners often employ model selection techniques like the Akaike Information Criterion (AIC) or the Bayesian Information Criterion (BIC). These introduce a regularization penalty based on the number of parameters, favoring simpler and more generalizable models in accordance with Occam’s razor [31].

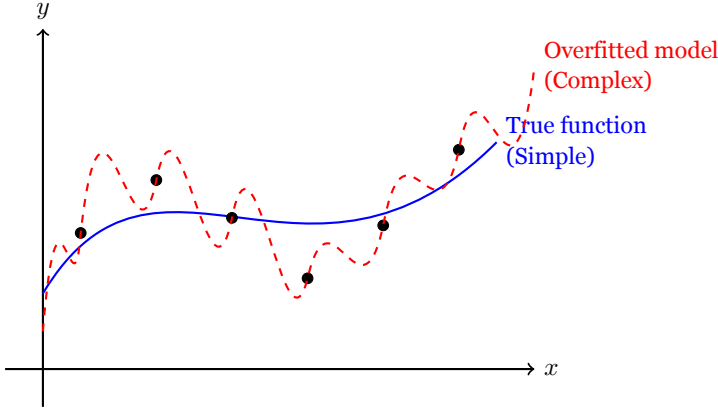


Figure 2.2.6: Illustration of overfitting. A flexible, complex model (red dashed line) perfectly hits all data points by minimizing the fitting error, but completely misses the simpler true underlying function (blue line).

When the model contains latent variables $\mathbf{z}$, the marginal likelihood $p(\mathbf{y} | \theta)$ requires integrating (or summing) over all possible states of $\mathbf{z}$, i.e., $p(\mathbf{y} | \theta) = \sum_{\mathbf{z}} p(\mathbf{y}, \mathbf{z} | \theta)$. This marginalization often makes direct optimization intractable. In these cases, we can sometimes use Expectation-Maximization (EM), an iterative algorithm. The EM algorithm alternates between two steps: the E-step computes the expected value of the complete-data log-likelihood with respect to the distribution of the latent variables given the current parameters $\theta^{(k)}$:

$$Q(\theta | \theta^k) = \mathbb{E}_{\mathbf{z} | \mathbf{y}, \theta^k} [\log p(\mathbf{y}, \mathbf{z} | \theta)] = \sum_{\mathbf{z}} p(\mathbf{z} | \mathbf{y}, \theta^k) \log p(\mathbf{y}, \mathbf{z} | \theta). \quad (2.2.6)$$

The M-step then updates the parameters by maximizing this expected log-likelihood: $\theta^{k+1} = \arg \max_{\theta} Q(\theta | \theta^k)$.

For HMMs specifically, the E-step requires computing the marginal probabilities $p(z_t | \mathbf{y}, \theta^k)$, known as the smoothing probabilities. These probabilities, as mentioned earlier, can be computed efficiently with the Forward-Backward algorithm.

Bayesian inference

While MLE provides a single point estimate, Bayesian inference offers a richer, more rigorous alternative [35]. Instead of treating θ as a fixed unknown, Bayesian methods treat parameters as random variables with a prior distribution $p(\theta)$. This prior encapsulates our assumptions before seeing any data. By applying Bayes' theorem, we compute the posterior distribution over both the latent variables and the parameters:

$$p(\mathbf{z}, \theta | \mathbf{y}) = \frac{p(\mathbf{y} | \mathbf{z}, \theta) p(\mathbf{z} | \theta) p(\theta)}{p(\mathbf{y})}. \quad (2.2.7)$$

The denominator, $p(\mathbf{y})$, is the *evidence* or marginal likelihood, obtained by marginalizing the joint model over all latent variables and parameters (sums for discrete variables and

integrals for continuous ones). The factorization in the numerator defines a *generative model*, i.e., a probabilistic specification of how latent variables and parameters generate observed data through the joint distribution $p(\mathbf{y}, \mathbf{z}, \theta) = p(\mathbf{y} | \mathbf{z}, \theta)p(\mathbf{z} | \theta)p(\theta)$. Bayesian inference then amounts to inverting this generative process to infer latent structure from observations. The posterior distribution quantifies exactly how confident we should be about our latent variables and parameters. If the data is noisy and ambiguous, the posterior remains broad, honestly reflecting our uncertainty.

Markov Chain Monte Carlo

For all but the simplest models, calculating the evidence $p(\mathbf{y})$ requires an intractable high-dimensional integral. To bypass this, since the advent of modern computing, statisticians have turned to Monte Carlo methods, which approximate the posterior distribution by drawing random samples from it. Markov Chain Monte Carlo (MCMC) is a framework for exact posterior inference; it constructs a Markov chain whose stationary distribution is the exact target posterior. By recording the steps of the chain, we build an empirical histogram that approximates the posterior. While mathematically guaranteed to converge given infinite time, exploring the distribution relies on random walks which can suffer from poor mixing, meaning the chain struggles to transition between different high-probability modes and can be prohibitively slow.

One of the most widely taught MCMC algorithms is Gibbs sampling [36]. By iteratively sampling each variable from its full conditional distribution $p(z_i | \mathbf{z}_{-i}, \mathbf{y}, \theta)$ with $\mathbf{z}_{-i} = (z_1, \dots, z_{i-1}, z_{i+1}, \dots, z_N)$, we can construct a Markov chain that is guaranteed to converge to the true posterior. If the full conditionals are tractable, Gibbs sampling is simple and efficient. However, in many complex models, these conditionals cannot be sampled from directly. In such cases, the Metropolis-Hastings [37, 38] (MH) algorithm provides a more general solution. Given a current state x , the MH algorithm proposes a new state x' from a transition kernel $T(x' | x)$. The proposal is accepted with probability:

$$\alpha = \min \left(1, \frac{p(x' | \mathbf{y})T(x | x')}{p(x | \mathbf{y})T(x' | x)} \right). \quad (2.2.8)$$

Notice that the intractable evidence $p(\mathbf{y})$ cancels out in the ratio $\frac{p(x' | \mathbf{y})}{p(x | \mathbf{y})}$, allowing us to evaluate the acceptance probability using only the joint probability.

In many applications, MCMC methods are still the gold standard for inference, providing asymptotically exact samples from the posterior in a reasonable amount of time. However, there are several considerations to be made regarding their practical use. First, since convergence is not guaranteed in finite time, it is crucial to run multiple chains and use diagnostic tools to assess convergence. In most real scenarios, checking convergence is non-trivial, requiring careful analysis of trace plots, autocorrelation, and other statistics. Even if the chains appear to have converged, they might still be stuck in local modes, leading to biased estimates. Second, the choice of the proposal distribution T in MH can significantly affect the efficiency of the algorithm. Poorly chosen proposals can lead to low acceptance rates and slow mixing. Adaptive MCMC and simulated annealing are only a few of the numerous strategies developed to mitigate these issues, but any adopted strategy typically requires careful tuning and validation

for each specific model and dataset. Tuning MCMC algorithms is particularly challenging due to the long run times and the high dimensionality of the parameter space. In practice, when there is a necessity of testing multiple complex models or performing inference on a large amount of multi-modal data, MCMC methods are often too computationally expensive [39]. More efficient algorithms like Hamiltonian Monte Carlo (HMC) [40] have been developed to improve the exploration of the posterior, but adapting these methods to discrete latent variables, such as cluster assignments or tree topologies, is non-trivial and an active area of research [41, 42]. This motivates the need for faster, albeit approximate, inference techniques, which we will discuss in the next section.

Approximate inference

When exact inference via MCMC is computationally challenging, Variational Inference (VI) provides an approximate alternative [43]. Variational inference fundamentally changes the paradigm: it transforms the sampling problem into an optimization problem. We propose a parameterized family of tractable distributions $q_\phi(\mathbf{z}, \theta)$ and search for the parameters ϕ that minimize the Kullback-Leibler (KL) divergence to the true posterior

$$\text{KL}(q_\phi(\mathbf{z}, \theta) \parallel p(\mathbf{z}, \theta \mid \mathbf{y})) = \int q_\phi(\mathbf{z}, \theta) \log \frac{q_\phi(\mathbf{z}, \theta)}{p(\mathbf{z}, \theta \mid \mathbf{y})} d\mathbf{z} d\theta. \quad (2.2.9)$$

Since the true posterior $p(\mathbf{z}, \theta \mid \mathbf{y})$ is intractable, we cannot compute this KL divergence directly. Instead, VI maximizes an alternative objective called the Evidence Lower Bound (ELBO):

$$\text{ELBO}(\phi) = \mathbb{E}_{q_\phi} [\log p(\mathbf{y}, \mathbf{z}, \theta)] - \mathbb{E}_{q_\phi} [\log q_\phi(\mathbf{z}, \theta)]. \quad (2.2.10)$$

It is straightforward to show that the log evidence decomposes as $\log p(\mathbf{y}) = \text{ELBO}(\phi) + \text{KL}(q_\phi \parallel p(\mathbf{z}, \theta \mid \mathbf{y}))$. Since the KL divergence is strictly non-negative, the ELBO lower-bounds the true log evidence. Maximizing the ELBO is equivalent to minimizing the KL divergence. Furthermore, the ELBO has been shown to be particularly useful as an approximation of the model evidence, enabling efficient model selection, such as choosing the number of clusters K in a mixture model [44].

To make this optimization tractable, the variational distribution is often factorized using the mean-field approximation, assuming that the latent variables and parameters are mutually independent, $q_\phi(\mathbf{z}, \theta) = \prod_j q_j(z_j) \prod_k q_k(\theta_k)$. While this breaks complex posterior correlations, it enables faster inference.

Optimization under the mean-field assumption is commonly performed using Coordinate Ascent Variational Inference (CAVI). CAVI iteratively updates each variational parameter by setting the its distribution proportional to the expected joint distribution over all other variables (in log space):

$$\log q_j(z_j) \propto \mathbb{E}_{-j} [\log p(\mathbf{y}, \mathbf{z}, \theta)]. \quad (2.2.11)$$

For example, returning to our Poisson mixture model, updating the variational distribution for the cluster assignment $q(z_i)$ involves taking the expectation of the complete-data

log-likelihood and log-prior with respect to the current variational distributions of the mixing proportions π and cluster means θ . This coordinate ascent algorithm is guaranteed to monotonically increase the ELBO, iteratively minimizing the distance to the true posterior. However, like many iterative optimization methods, CAVI is susceptible to local optima and heavily depends on initialization, often requiring multiple restarts to find the global optimum.

2.3 Phylogenetics

Phylogenetics is the formal study of evolutionary relationships among biological entities. By analyzing heritable traits, like molecular sequences, we can reconstruct the history of these entities in the form of an evolutionary tree, or *phylogeny*. In this tree, the leaves represent the observed present-day entities, while the internal nodes represent their unobserved common ancestors. There are many approaches to inferring phylogenies, each with its own assumptions and computational trade-offs. In recent years, phylogenetic methods have been increasingly applied to study cancer. As we already discussed, tumors evolve through a process of clonal expansion driven by somatic mutations, which can therefore be used as markers to reconstruct the tumor’s evolutionary history, similarly to how genetic differences are used to reconstruct the history of species. The purpose of this section is to provide the necessary background on the concepts and methods used in the development of this thesis, not to give an exhaustive review of the field of phylogenetic inference. Specifically, we start by introducing substitution models, which provide a mathematical framework to model the stochastic process of mutation accumulation along the branches of a phylogenetic tree. We then discuss the different classes of phylogenetic inference methods, with a focus on distance-based approaches like Neighbor-Joining. Finally, we review the different metrics used to compare and evaluate phylogenetic trees.

2.3.1 Models of evolution

To reconstruct an evolutionary history from observed data, we must first define a formal model of evolution. A model of evolution describes the stochastic process by which genetic states change over time along the branches of a phylogenetic tree. It specifies the possible states a biological character can take and assigns probabilities to the transitions between these states over a given evolutionary distance. By providing a mathematical framework to evaluate the likelihood of observing specific sequence differences given a hypothetical tree, evolutionary models allow us to (i) quantify how closely related different observations are, (ii) correct for unseen multiple mutations, and ultimately (iii) identify the tree topology and branch lengths that best explain the observed biological data.

Modeling sequence evolution

In phylogenetics, the evolution of molecular sequences over time is typically modeled as a continuous-time Markov chain (CTMC) [45]. Let $X(t)$ be a random variable representing the state of a sequence character at time $t \geq 0$. The character can be, for example, a DNA base from $\{A, C, G, T\}$, or, on a more related note, a copy-number state from $\{0, 1, 2, \dots, K\}$ where K is the maximum copy number considered. The fundamental

assumption of a CTMC is the Markov property. This property, as already mentioned in Section 2.2.2 for discrete sequences, can be formally expressed as follows. For any time t and any small time increment $\Delta t > 0$, the probability of transitioning to a new state j at time $t + \Delta t$ depends only on the current state i at time t , and is conditionally independent of the entire history of states prior to t :

$$\mathbb{P}(X(t + \Delta t) = j \mid X(t) = i, X(u) = x(u) \text{ for } 0 \leq u < t) = \mathbb{P}(X(t + \Delta t) = j \mid X(t) = i), \quad (2.3.12)$$

The transition probabilities over a finite time interval t are collected into a matrix $\mathbf{P}(t)$, where each entry is defined as the probability of transitioning from state i to state j after time t :

$$P_{ij}(t) = \mathbb{P}(X(t) = j \mid X(0) = i). \quad (2.3.13)$$

For an infinitesimal time step Δt , transition probabilities are governed by the instantaneous rate matrix $\mathbf{Q} = (q_{ij})$. The off-diagonal entries q_{ij} ($i \neq j$) give the instantaneous rate of transitioning from state i to state j , defined as

$$q_{ij} = \lim_{\Delta t \rightarrow 0} \frac{P_{ij}(\Delta t)}{\Delta t}, \quad i \neq j. \quad (2.3.14)$$

To ensure that each row of the transition probability matrix $\mathbf{P}(t)$ sums to 1 for all t , the diagonal elements are defined as the negative sum of the off-diagonal rates,

$$q_{ii} = - \sum_{j \neq i} q_{ij}. \quad (2.3.15)$$

With this definition, $\mathbf{Q}$ uniquely determines the transition probabilities through the Kolmogorov equations, giving

$$\mathbf{P}(t) = \exp(\mathbf{Q}t). \quad (2.3.16)$$

Substitution models

Changes in molecular sequences consist of discrete events where one character state is replaced by another, known as *substitutions*. Depending on the type of data and the biological context, different substitution models have been developed to capture the specific patterns of mutation. A substitution model specifies the state space $\mathcal{S}$ of possible character states and the transition rates between them. The simplest model for DNA sequences is the Jukes-Cantor (JC69) model [13]. It assumes that all bases are equally frequent (i.e., $1/4$ for all bases in $\mathcal{S} = \{A, C, G, T\}$) and that all substitution rates between different bases are equal to a constant α . The JC69 rate matrix is therefore:

$$\mathbf{Q}_{\text{JC}} = \begin{pmatrix} -3\alpha & \alpha & \alpha & \alpha \\ \alpha & -3\alpha & \alpha & \alpha \\ \alpha & \alpha & -3\alpha & \alpha \\ \alpha & \alpha & \alpha & -3\alpha \end{pmatrix}. \quad (2.3.17)$$

By solving the Kolmogorov differential equations $\mathbf{P}(t) = \exp(\mathbf{Q}_{\text{JC}}t)$, we can derive the exact probability $P_{ij}(t)$ of a site changing from state i to a different state j over

an evolutionary time t . For the JC69 model, this probability of observing a specific difference is:

$$P_{ij}(t) = \frac{1}{4} - \frac{1}{4}e^{-4\alpha t} \quad \text{for } i \neq j. \quad (2.3.18)$$

Since there are three possible different states to mutate into, the total probability p of observing any change at a given site is $p = 3 \times P_{ij}(t) = \frac{3}{4}(1 - e^{-4\alpha t})$. Given two sequences, we can estimate p as the observed proportion of sites that differ between them, denoted $\hat{p}$. However, this raw proportion only reflects whether two sites differ, and does not account for multiple hits at the same site. To correct for this, we use the Jukes–Cantor substitution model to relate $\hat{p}$ to the underlying evolutionary time t . Defining the evolutionary distance as the expected number of substitutions per site, $d = 3\alpha t$, and solving for d in terms of $\hat{p}$, we obtain the Jukes–Cantor correction formula. This correction follows from the ML estimate under the reversible JC69 model:

$$d = -\frac{3}{4} \ln \left(1 - \frac{4}{3}\hat{p} \right). \quad (2.3.19)$$

This formula corrects for “multiple hits”—the phenomenon where a single site mutates multiple times over a long evolutionary period, causing the raw proportion of differences p to underestimate the true evolutionary distance d .

To illustrate this, consider two short DNA sequences of length 10:

Seq 1: A C G T A C G T A C
Seq 2: A C A T A T G T A C

There are 2 differing sites out of 10, meaning the raw observed difference is $p = 2/10 = 0.2$. Applying the JC69 correction, the estimated true evolutionary distance is $d = -\frac{3}{4} \ln(1 - \frac{4}{3}(0.2)) \approx 0.233$. The corrected distance is larger than the raw difference, reflecting the likelihood that some sites experienced multiple unobserved mutations.

More complex classic models, such as the Kimura 2-parameter (K80) model, introduce different rates for transitions (e.g., $A \leftrightarrow G$) and transversions (e.g., $A \leftrightarrow C$). In the context of tumor phylogenetics, traditional sequence evolution models are often adapted or replaced by tumor-specific models. For instance, the *infinite-sites assumption* posits that a mutation occurs at most once per genomic site across the entire evolutionary history of the tumor, and is never lost. This assumption, typically adopted in the analysis of point mutations (SNVs), drastically simplifies the state space and the space of possible trees. For copy-number aberrations (CNAs), models must account for large-scale genomic events where entire chromosomal segments are duplicated or deleted, requiring entirely different state spaces and transition rate definitions.

Ultrametric vs. non-ultrametric trees

When modeling the evolutionary history of a set of sequences, the resulting phylogenetic tree can be either ultrametric or non-ultrametric. An ultrametric tree is one where the distance from the root to every leaf is identical. Biologically, this corresponds to the assumption of a strict molecular clock, meaning that the rate of mutation is

constant across all lineages and over time. In such a tree, the branch lengths are directly proportional to elapsed physical time, and all sampled taxa (the leaves) are assumed to have been observed at the same point in time. While this assumption simplifies inference and may be appropriate for certain species-level phylogenies over long timescales, it is often violated in cancer evolution. As discussed in the previous section, tumor subclones can exhibit highly variable mutation rates due to genomic instability and different selective pressures, leading to non-ultrametric trees where branch lengths represent the varying number of accumulated mutations rather than strict chronological time.

2.3.2 Tree inference

Reconstructing the evolutionary history of a set of sequences is a complex computational challenge. The objective of tree inference is to find the phylogenetic tree—both its topology and branch lengths—that best explains the observed data. The observed biological sequences, also known as *taxa*, are the leaves of the tree, while the internal nodes represent hypothetical common ancestors. Methods for tree inference are broadly divided into two main categories: distance-based methods and character-based methods.

Distance-based methods

Distance-based methods are among the fastest and computationally most efficient approaches to phylogenetic inference. The starting point is a pairwise distance matrix $\mathbf{D}$, where each entry D_{ij} represents the estimated evolutionary distance between sequence i and sequence j , typically corrected for multiple substitutions using a model like Jukes-Cantor (as seen in the previous section). If the data were clean and evolved strictly according to the model, the distances would be perfectly *additive*, meaning the distance between any two leaves in the true tree would exactly equal the sum of the branch lengths along the path connecting them.

The most prominent distance-based algorithm is Neighbor-Joining (NJ) [14]. Instead of searching through the vast space of all possible trees, NJ uses a greedy, agglomerative clustering approach to construct the tree in polynomial time. NJ is a recursive agglomerative algorithm: at each iteration, it identifies a sibling pair, replaces it with a new composite taxon, and recurses on the reduced distance matrix until the tree is resolved. It iteratively identifies and joins the closest pair of nodes. To avoid incorrectly joining sequences that are only close because they share a long branch (a common issue in distance-based methods known as the “long branch attraction” problem), NJ optimizes a transformed distance measure called the Q -criterion. For any pair of nodes i and j , the Q -value is defined as:

$$Q(i, j) = (n - 2)D_{ij} - \sum_{k=1}^n D_{ik} - \sum_{k=1}^n D_{jk}. \quad (2.3.20)$$

At each step, the algorithm selects the pair (i, j) that minimizes $Q(i, j)$, joins them to a newly created internal node u , and calculates the branch lengths from u to i and

j :

$$d(i, u) = \frac{1}{2}D_{ij} + \frac{1}{2(n-2)} \left(\sum_{k=1}^n D_{ik} - \sum_{k=1}^n D_{jk} \right). \quad (2.3.21)$$

The distance matrix is then updated to replace i and j with the new node u , and the process is repeated until the tree is fully resolved. While NJ is fast and guaranteed to recover the correct tree if the distances are perfectly additive, its practical accuracy can degrade significantly when distances are heavily distorted by sequencing noise or limited sequence length, as is common in single-cell genomics [46, 47].

Maximum parsimony

Maximum Parsimony is one of the earliest character-based methods developed for phylogenetics. The core philosophy of parsimony is Occam’s razor: the best evolutionary hypothesis is the one that requires the fewest assumptions. In the context of tree inference, this translates to finding the tree topology that explains the observed sequences with the minimum total number of evolutionary changes (mutations) [48].

To evaluate a specific tree under Maximum Parsimony, algorithms reconstruct the ancestral states at each internal node that minimize the total number of character changes along the branches. The algorithm assigns a discrete “parsimony score” to the tree, and the goal is to find the tree that minimizes this score. While conceptually elegant, Maximum Parsimony does not incorporate an explicit probabilistic model of evolution. As a result, it cannot easily correct for multiple substitutions at the same site, making it particularly susceptible to long branch attraction, where rapidly evolving lineages are erroneously grouped together. This limitation becomes especially severe when many mutations accumulate at the same sites or in the same lineages, because the central parsimony principle (fewest changes) is then no longer a good approximation of the true process. For these reasons, while historically significant, Maximum Parsimony is rarely used for complex, noisy data like single-cell sequencing.

Maximum likelihood and Bayesian inference

The most robust and statistically rigorous approaches to tree inference are Maximum Likelihood (ML) and Bayesian Inference. These methods directly utilize the substitution models described earlier to calculate the probability of the observed sequences given a specific tree topology and branch lengths.

In Maximum Likelihood, the goal is to find the single tree and set of parameters that maximize the probability of the observed data. Bayesian inference, on the other hand, seeks the posterior distribution of trees given the data and prior distributions over topologies, branch lengths, and model parameters. Bayesian phylogenetics is particularly powerful because it inherently quantifies the uncertainty in the reconstruction, providing not just a single best tree, but a credible set of trees that are supported by the data.

However, the space of possible phylogenetic trees grows superexponentially with the number of leaves. For n taxa, there are $(2n-5)!!$ possible unrooted and $(2n-3)!!$ rooted bifurcating tree topologies. This combinatorial explosion makes exhaustive search

impossible for anything more than a dozen sequences. Consequently, inference relies on heuristic search algorithms or Markov Chain Monte Carlo (MCMC) methods to explore the tree space. In MCMC frameworks like those implemented in MrBayes [49], the algorithm proposes small structural modifications to the current tree—such as Nearest Neighbor Interchange (NNI) or Subtree Pruning and Regrafting (SPR) moves—and accepts or rejects them based on the improvement in the posterior probability.

While MCMC is mathematically guaranteed to sample from the true posterior distribution eventually, exploring the rugged landscape of tree space is notoriously difficult. Chains often get trapped in local optima and suffer from slow convergence, requiring massive computational resources to analyze large datasets. Developing efficient algorithms for proposing or directly sampling trees is therefore a major area of ongoing research. The challenge of defining and sampling from probability distributions over trees is a complex problem that requires its own set of mathematical tools, which we will introduce formally in the next section.

2.3.3 Quantifying tree similarity

Comparing two phylogenetic trees is a highly non-trivial problem [50]. Unlike Euclidean spaces where the distance between two vectors is straightforward to compute, the space of phylogenetic trees is a complex, discrete combinatorial structure. A phylogenetic tree topology is uniquely defined by its nodes, edges, and the specific labeling of its leaves. As we have seen, enumerating all possible trees may be prohibitive even for a modest number of taxa.

Measuring the “distance” between any two trees in this vast discrete space requires defining a formal mathematical (dis)similarity measure that captures the meaningful structural differences between them. We can classify tree distance measures into two broad categories: set-based metrics and transfer-based metrics. In the former, the distance is calculated by decomposing the trees into smaller, comparable combinatorial units—such as splits, clades, or subtrees—and calculating the symmetric difference between these sets. In the latter, the distance is defined as the minimum number of edit operations required to transform one tree into the other, where an edit operation is a discrete modification of the tree structure, such as moving a subtree from one location to another. There is no single universally accepted tree distance; instead, different measures have been proposed to capture different aspects of tree similarity, each with its own strengths and weaknesses. In this section, we will review some of the most commonly used tree distance measures, focusing on those which are relevant for comparing the topologies of phylogenetic trees, rather than their branch lengths.

Robinson-Foulds distance

The Robinson-Foulds (RF) distance [51] is arguably the most common metric (dissimilarity measure which satisfies the triangle inequality) for comparing phylogenetic trees. It evaluates the similarity between two unrooted trees based purely on the set of bipartitions (splits) induced by their internal edges. Removing any internal edge divides the set of taxa (leaves) into two disjoint sets, creating a bipartition. Let B_1 be the set of bipartitions in tree T_1 , and B_2 the set of bipartitions in tree T_2 . The RF distance is the

symmetric difference between these sets, defined as the number of bipartitions present in one tree but missing in the other: $|B_1 \setminus B_2| + |B_2 \setminus B_1|$. For fully resolved (binary) trees with n leaves, the maximum possible RF distance is $2n - 6$. Because the RF distance strictly compares identical splits, it is highly sensitive to small topological changes: slightly repositioning a single taxon can dramatically increase the distance.

Quartet distance

Another widely used topological metric is the quartet distance [52]. A quartet is the unrooted topology connecting four taxa. For any four distinct leaves in a fully resolved tree, there are exactly three possible unrooted topologies. A phylogenetic tree is fully defined by the set of quartets it induces. The quartet distance between two unrooted trees T_1 and T_2 is defined as the number of subsets of four taxa for which the induced quartet topology differs between the two trees. Because the number of possible quartets for a tree with n leaves is $\binom{n}{4}$, the maximum quartet distance grows rapidly with tree size, giving a more fine-grained measure of topological difference than the RF distance. This typically makes the quartet distance more robust to small, localized changes in the tree. The quartet distance is generally considered more robust and less sensitive to small, localized topology changes than the RF distance, as displacing a single leaf only affects the quartets containing that specific taxon.

Transfer distance

Instead of counting shared sub-structures, a transfer distance counts the minimum number of discrete structural operations required to transform the topology of T_1 into T_2 . Depending on the specific edit operations allowed, different transfer distances can be defined. For instance, the commonly used Subtree Pruning and Regrafting (SPR) operation defines a distance where one can cut a subtree from one location (prune) and reattach it elsewhere in the tree (regraft).

Computing transfer distances is generally more computationally intensive than set-based metrics, as it requires searching through the space of possible edit operations to find the optimal sequence that transforms one tree into the other (typically an NP-hard problem) [53, 54]. However, efficient algorithms and heuristics have been developed for specific types of edit operations, making it feasible to compute transfer distances for moderately sized trees [55, 56, 57].

Transfer metrics often align more directly with plausible evolutionary edit events than purely set-theoretic metrics like the RF distance, therefore providing a more meaningful framework for comparing and evaluating phylogenetic trees, and ultimately exploring the tree space (like in the context of MCMC sampling) [58, 41].

2.4 Tree sampling

In Bayesian phylogenetics and other applications involving evolutionary models, it is often necessary to sample from a distribution of possible trees, rather than simply finding a single optimal tree. This allows us to quantify uncertainty over the tree topology and branch lengths. Drawing random trees according to specific probability

distributions is a classic problem that connects probability theory, combinatorics, and graph theory. In this section, we review the foundational concepts of spanning trees and the algorithms used to sample them from graphs.

2.4.1 Graphs, spanning trees, and arborescences

To formally discuss tree sampling, we start with graph-theory definitions [59]. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, W)$ be a weighted directed graph on a set $\mathcal{V}$ with n nodes, where each arc $(u, v) \in \mathcal{E}$ has a non-negative real-valued weight $w_{uv} > 0$. The weight represents the transition probability, cost, or distance between nodes. If the edges have no direction, the graph is undirected, and an edge is an unordered pair $\{u, v\}$.

A *tree topology*, or simply a *tree*, is an undirected, connected graph with no cycles. If an undirected graph $\mathcal{G}$ is connected, a *spanning tree* of $\mathcal{G}$ is a subgraph that contains all vertices in $\mathcal{V}$ and is a tree. Because a spanning tree on n vertices must contain exactly $n - 1$ edges, adding any edge from $\mathcal{G}$ outside that tree creates a cycle, and removing any tree edge disconnects it.

In many phylogenetic applications, particularly those modeling time or evolutionary descent from an ancestor, we are interested in directed trees. The directed analogue of a tree is an *arborescence* (also called a directed rooted tree). A spanning arborescence $\mathcal{T}$ of $\mathcal{G}$ rooted at a specific vertex $r \in \mathcal{V}$ is a directed tree in which all arcs point away from the root r , such that every vertex $v \in \mathcal{V} \setminus \{r\}$ has exactly one directed path from r to v .

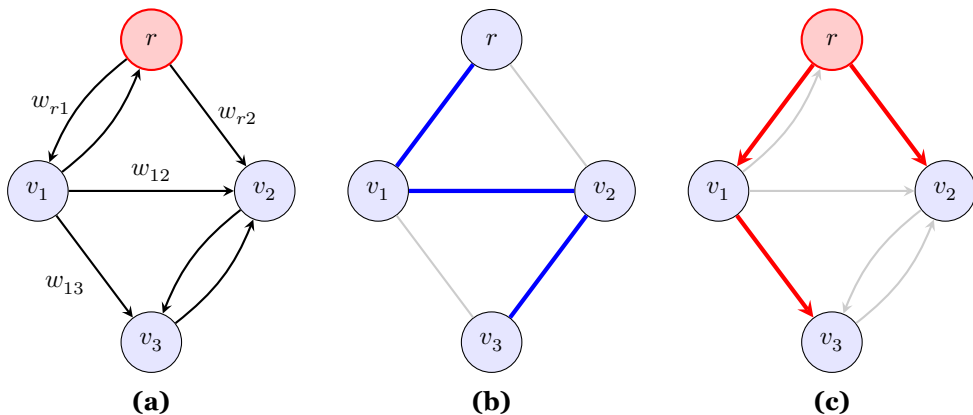


Figure 2.4.7: Visualizing trees and arborescences. (a) A weighted directed graph $\mathcal{G}$ with root node r . (b) A valid spanning tree (ignoring edge directions). (c) A valid spanning arborescence rooted at r , where all arcs point away from the root.

Figure 2.4.7 visually clarifies the distinction between a generic graph, an undirected spanning tree, and a directed spanning arborescence. In Bayesian phylogenetics and other generative models, a *random* spanning arborescence $\mathcal{T}$ is drawn from a probability

distribution over all valid arborescences. We formally define the problem of sampling an arborescence as drawing $\mathcal{T}$ such that its probability is proportional to its weight:

$$\mathbb{P}(\mathcal{T}) = \frac{1}{Z} \mathcal{W}(\mathcal{T}), \quad \mathcal{W}(\mathcal{T}) = \prod_{(u,v) \in \mathcal{T}} w_{uv}, \quad (2.4.22)$$

where $\mathcal{W}(\mathcal{T})$ is the product of the edge weights in $\mathcal{T}$, and Z is a normalization constant. The structural properties of arborescences make computing this constant and sampling the trees significantly more complex than in the undirected case.

2.4.2 The Matrix-Tree Theorem

The Matrix-Tree Theorem, originally discovered by Gustav Kirchhoff in 1847 [60], provides a remarkable algebraic method to count the exact number of spanning trees in a graph. For a connected, undirected graph $\mathcal{G}$ without self-loops, we define the degree matrix $\mathbf{D}$ as a diagonal matrix where D_{ii} is the degree of vertex i , i.e., the sum of the weights of edges incident to i . We define the adjacency matrix $\mathbf{A}$ such that $A_{ij} = 1$ if there is an edge between i and j , and 0 otherwise. The Laplacian matrix $\mathbf{L}$ is defined as:

$$\mathbf{L} = \mathbf{D} - \mathbf{A}. \quad (2.4.23)$$

Kirchhoff's theorem states that the number of spanning trees in $\mathcal{G}$ is equal to the value of any cofactor of $\mathbf{L}$. This is a second reduction step: after constructing $\mathbf{L}$, we remove the i -th row and i -th column of $\mathbf{L}$ to form the reduced Laplacian $\mathbf{L}^{(i)}$, the number of spanning trees is given by $\tau(\mathcal{G}) = \det(\mathbf{L}^{(i)})$.

This theorem was later extended to directed graphs by Tutte [61, 62] and generalized to compute the sum of the weights of all spanning arborescences, where the weight of an arborescence is defined as the product of the weights of its edges. For a digraph with edge weights w_{uv} , the weighted in-degree of vertex v is $\sum_u w_{uv}$. We define the directed Laplacian $\mathbf{L}$ such that its off-diagonal entries are $L_{uv} = -w_{uv}$, and its diagonal entries are the weighted in-degrees $L_{vv} = \sum_{u \neq v} w_{uv}$. Tutte's Matrix-Tree Theorem states that the sum of the weights of all spanning arborescences rooted at r , denoted $\tau_r(\mathcal{G})$, is given by the determinant of the reduced Laplacian obtained by removing the r -th row and r -th column, elegantly extending Kirchhoff's result to the directed case:

$$\tau_r(\mathcal{G}) = \det(\mathbf{L}^{(r)}). \quad (2.4.24)$$

This algebraic result is profoundly important for probabilistic inference, as it allows us to compute the exact normalization constant (the partition function) for probability distributions over directed trees in polynomial time [63] $O(n^3)$ via matrix determinant computations.

2.4.3 Algorithms for tree sampling

Once we can compute the total weight of all trees using the Matrix-Tree Theorem, we can define a probability distribution where the probability of drawing a specific arborescence T is proportional to its weight:

$$\mathbb{P}(T) = \frac{\prod_{(u,v) \in T} w_{uv}}{\det(\mathbf{L}^{(r)})}. \quad (2.4.25)$$

Sampling from this distribution efficiently is non-trivial. One prominent class of algorithms relies on exact determinant computations [63]. These methods build the tree edge-by-edge. At each step, they compute the marginal probability of an edge belonging to a spanning arborescence by evaluating determinants of appropriately modified Laplacian matrices. While exact and highly flexible for weighted directed graphs, determinant-based methods generally require $O(n^3)$ operations per edge, making them computationally intensive for very large graphs.

An alternative, elegant approach uses random walks. Aldous [64] and Broder [65] independently discovered that simulating a random walk on an undirected graph and keeping only the edges that discover previously unvisited vertices yields a spanning tree with probability proportional to its weight. Later, Wilson [66] introduced a much more efficient algorithm using Loop-Erased Random Walks (LERW) which can also be applied in the directed case. Wilson’s algorithm builds a tree by picking a root node and simulating a loop-erased random walk from an unvisited vertex. Whenever the walk intersects its own path, the resulting loop is erased. Once the walk hits the growing tree, the loop-erased path is permanently added to the tree. This process is repeated until all vertices are included. Wilson’s algorithm is typically considered to be faster than determinant-based methods for large graphs, but the speed depends on the structure of the graph and the distribution of edge weights. In fact, if the edge weights are highly skewed, the random walk may take a long time to discover new vertices, leading to long runtimes. A key factor is the graph *conductance*, which informally measures how easily the random walk can move between different parts of the graph. More precisely, the conductance of a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ can be defined as

$$\Phi(\mathcal{G}) = \min_{\emptyset \subsetneq C \subset \mathcal{V}} \frac{\sum_{u \in C, v \in \mathcal{V} \setminus C} w_{uv}}{\sum_{u, v \in C} w_{uv}}, \quad (2.4.26)$$

where the minimum is taken over all nonempty, proper subsets $C \subset \mathcal{V}$. Low conductance means that there exist subsets of vertices that are weakly connected to the rest of the graph, so that a random walk tends to remain trapped in these regions for a long time before escaping. As a result, the time required to visit all vertices (the cover time) can become very large, making random walk-based methods such as Wilson’s algorithm inefficient in these settings. Interestingly, the same low conductance that hampers random walk sampling also leads to numerical instability in determinant-based methods, as the Laplacian matrix becomes ill-conditioned, making it difficult to compute the determinant accurately [67, 68].

2.5 Research gaps

With the terminology and conceptual framework now established, the previous sections can be brought together to identify the methodological gaps that motivate the work presented in this thesis. First, while technologies like scDNA-seq provide cell-level measurements, the resulting data is noisy, hampered by dropouts, amplification biases, and the fundamental complexity of modeling structural variations like copy-number aberrations. Current probabilistic models must bridge the gap between raw, sparse read counts and latent cell-specific biological states. This requires models (e.g., Hidden Markov Models) that accommodate the noise profiles of scDNA-seq within a comprehensive end-to-end probabilistic framework, rather than relying on multi-step pipelines

with ad-hoc preprocessing and downstream tree inference (e.g., copy number calling followed by separate tree building), rather than jointly modeling the data generation process and the underlying evolutionary dynamics.

Second, classical phylogenetic methods, such as Maximum Parsimony and Maximum Likelihood, were primarily developed for the evolution of species over vast timescales. They often rely on simplifying assumptions like the independence of point mutations. These assumptions break down in the context of somatic tumor evolution, which is characterized by rapid, punctuated bursts of genomic instability — reminiscent of a Cambrian explosion — along with allelic imbalance, and highly heterogeneous mutation rates. Furthermore, distance-based methods like Neighbor-Joining, while computationally efficient, often fail to accurately resolve tree topologies when provided with heavily distorted or uncorrected distances derived from noisy CNA profiles. Moreover, tumor evolution typically originates from a common ancestor (a normal, healthy cell), yet standard Neighbor-Joining is fundamentally designed to reconstruct unrooted trees.

Third, there is a critical need for rigorous uncertainty quantification in tumor phylogenetics. A single point-estimate tree topology often masks the uncertainty inherent in sparse and noisy single-cell data. While Bayesian inference offers a theoretically sound framework to capture this uncertainty by computing posterior distributions over trees and latent states, standard Markov Chain Monte Carlo (MCMC) approaches do not scale well. The rugged, superexponential landscape of directed tree topologies leads to poor mixing and slow convergence. Variational inference presents a promising alternative, yet applying it to discrete tree structures requires specialized algorithms to compute expectations and sample discrete arborescences efficiently.

Finally, the foundational algorithmic challenge of sampling these trees remains a bottleneck. As detailed in the previous section, calculating the normalization constant via the Matrix-Tree Theorem requires evaluating matrix determinants, which becomes numerically unstable for poorly connected graphs with low conductance. Conversely, random walk-based sampling methods (like Wilson’s algorithm) fail to terminate in a reasonable time on such graphs. Developing stable, polynomial-time algorithms to sample spanning arborescences under arbitrary edge weights is therefore not just a theoretical pursuit, but a practical necessity for enabling scalable Bayesian inference in phylogenetics.

The research presented in this thesis directly addresses these intertwined gaps, introducing novel algorithms and comprehensive probabilistic frameworks tailored to the unique dynamics of tumor evolution. In the next two chapters, we further expand on these themes, framing the thesis contributions in the context of existing literature and outlining the specific innovations that the present work brings to the field.

3 Copy number evolution and distance-based phylogenetics

The reconstruction of a tumor’s phylogeny is at the very heart of cancer diagnosis, prognosis, and treatment. At the same time, due to the interplay of complex biological processes and technical limitations, it is also one of the most challenging problems in computational biology. Research in the field is continually identifying and addressing the many hallmarks of cancer, such as genomic instability and mutation, which drive the evolutionary process of tumor development and progression, and the unlocking of phenotypic plasticity that allows cancer cells to adapt and survive in changing environments [69, 70]. Several studies have linked the acquisition of copy-number aberrations (CNAs) and whole-genome duplications (WGDs) to intra-tumor heterogeneity (ITH) and phenotypic plasticity, suggesting that these genomic alterations provide a rich substrate for evolutionary diversification and adaptation in many types of cancer [71, 72, 73, 74, 75, 75]. Understanding how cells acquire and lose copies of genomic regions is therefore a fundamental aspect of tumor evolution and a critical step towards the development of effective therapies. In addition to their biological significance, CNAs are generally easier to detect than point mutations in single-cell DNA sequencing (scDNA-seq) data. This makes them an attractive marker for phylogenetic reconstruction. However, the sparse and noisy nature of scDNA-seq complicates accurate inference. Standard pipelines that rely on intermediate, often flawed, copy-number calls frequently propagate errors downstream, leading to inaccurate phylogenies. In this chapter, we review state-of-the-art methods for modeling copy-number evolution and inferring single-cell phylogenies. By highlighting the limitations of existing approaches, we motivate the methodological developments presented in this thesis.

In Section 3.1, we review the main models of copy-number evolution; in Section 3.2, we discuss the challenges of *calling* copy-number profiles from single-cell data and how to model the noise in the data; finally, in Section 3.3, we review the main methods for reconstructing phylogenies from pre-computed evolutionary distances, and we discuss the limitations. The last section, Section 3.4, is devoted to summarizing the

contributions of this thesis in the context of copy-number evolution and distance-based phylogenetic reconstruction, explicitly addressing the weaknesses of current methods here reviewed.

3.1 Evolutionary distances

Early studies of copy-number evolution in tumors were limited to fluorescence in situ hybridization (FISH) which provided only a coarse-grained view of single-cell genomic alterations, with a copy-number event spanning a specific gene or an entire chromosome [76]. With next-generation sequencing (NGS), researchers could finally profile copy-number changes across the entire genome at high resolution, allowing for more sophisticated models of copy-number evolution.

To measure the evolutionary divergence among cells based on their copy-number profiles, we need a way to quantify the distance between these profiles. In mathematical terms, a copy-number profile (CNP) is a vector of integers, where each entry corresponds to the number of copies of a specific genomic sections. Because we observe copy-number events only by counting the sequencing reads that land on the corresponding portion of the genome, these section units are often called *bins*, which in single cell data typically span between 100kb to 5Mb (resp. kilo and mega base pairs). Pioneering studies treated these CNPs as high-dimensional vectors and applied standard distance metrics to quantify their divergence. The Euclidean distance, the Manhattan distance and the Hamming distance are the most commonly used metrics in this context [9, 77, 76, 78, 79, 80, 81]. Such approaches are computationally efficient and straightforward to implement, making them attractive for large datasets. However, they fail to capture the complex biological processes underlying copy-number evolution, such as the fact that a single copy-number event can affect multiple contiguous bins. This oversimplification leads to inaccurate distance estimates and, consequently, flawed phylogenetic reconstructions.

This section explores how evolutionary distances between cells are computed. We start by discussing the Minimum Event Distance (MED) and the Copy Number Tree (CNT) problem, followed by the use of breakpoints as evolutionary markers, and conclude by highlighting the need for probabilistic distance estimation.

3.1.1 The Minimum Event Distance and the Copy Number Tree problem

The introduction of the MEDICC algorithm [82] was a significant milestone in this field, as it provided a principled way to quantify the amount of amplification and deletion events of contiguous bins. The Minimum Event Distance (MED) is defined as the minimum number of *segmental* gains or losses required to transform one CNP into another. In this setting, a single copy-number event is therefore identified as a segment with two coordinates (start and end positions), and the magnitude of the change (+1 for a gain, -1 for a loss). This definition of distance tries to capture the biological reality of how genomes evolve, serving as a way of quantifying the evolutionary divergence between populations of cells based on their copy-number profiles. Like in classical phylogenetics, the ultimate goal is to reconstruct the evolutionary history of the tumor, which is represented as a tree.

El-Kebir et al. [7] formalized this problem as the Copy Number Tree (CNT) problem, which asks for the tree that best explains the observed copy-number profiles of the cells, under the maximum parsimony assumption. A copy-number tree is a rooted binary tree where the leaves represent the observed cells, the edges are labeled with copy-number events, and the root node represents the healthy diploid ancestor. The CNT problem is thereby shown to be NP-hard.

Following the introduction of the MED model, several generalizations have been proposed to better reflect biological constraints and solve the CNT problem. For instance, Cordonnier and Lafond [83] generalized the model by allowing individual events to alter copy numbers by any amount with assigned positive costs, developing the method known as *cnp2cnp*. Another method, MEDALT [84], applies Edmonds' algorithm [85] to find the rooted directed minimum spanning tree (RDMST) (also known as minimum spanning arborescence), over the matrix of MED distances between cells. Specifically, the reconstructed *aneuploidy lineage tree* (ALT) is not a canonical phylogeny (binary tree with cells at the leaves), but rather a multifurcating tree where nodes are single cells, and edges are labeled with the MED between the two cells. This approach, therefore, does not explicitly model ancestral states and will incorrectly assign extant cells as internal nodes, potentially obscuring evolutionary relationships. Additionally, the MED distance is computed via greedy optimization of a parsimony score, which can fall short in the presence of high copy-number instability, where repeated or overlapping events may be underestimated, propagating errors into the reconstructed lineage tree. It is worth noting that the MED distance, as computed by MEDALT, adopts the *zero-absorption assumption*, which states that genomic segments with zero copies (where the DNA is completely lost) cannot subsequently gain copies. However, this is done simply by assigning an infinite cost to any event that would increase the copy number of a segment with zero copies, a strategy which naively assumes that the input copy-number profiles are perfectly accurate.

The Lazac method [86] relaxes the zero-absorption constraint to devise a closed-form solution for computing the MED distance between two CNPs. By allowing copy-number gains from zero, formalized in the zero-agnostic copy-number transformation (ZCNT), the problem of computing MED distance can be written as a linear program, which can be solved in polynomial time. The tree reconstruction is then performed by optimizing the parsimony score over the ZCNT distance matrix through a hill-climbing heuristic with nearest neighbor interchange (NNI) search. Despite the elegant solution, the ZCNT model is biologically unrealistic, and for this reason, it remains unused in practice.

Finally, the MEDICC2 method [87] significantly builds upon its predecessor by explicitly modeling loss of heterozygosity (LOH) and whole-genome duplication (WGD) events, which would otherwise be misinterpreted as consecutive segmental gains and losses. This method has been applied to large single-cell datasets, demonstrating improved accuracy and scalability compared to previous approaches. MEDICC2 employs a finite-state transducer [88] (a generalization of a finite-state automaton) to model the transformation of one CNP into another. Each state in the transducer corresponds to a specific composition of copy-number events, and the optimal path through the transducer is obtained via a single-source shortest path algorithm, yielding the MED distance between the two CNPs. Since the MED distance is not symmetric (due to the

zero-absorption assumption), the shortest path between two cells takes into account the unknown profile of the shared ancestor. Due to its improved accuracy and scalability, MEDICC2 has become a widely adopted method for reconstructing copy-number trees from single-cell data, and has been applied in several studies to elucidate the evolutionary history of tumors [89, 90, 91, 92].

3.1.2 Breakpoints as markers of copy-number evolution

One of the primary challenges in modeling copy-number evolution under the MED framework is its underlying computational complexity. Current MED-based methods struggle to scale to large datasets, such as those generated by modern scDNA-seq technologies containing thousands of cells. Even with the efficient algorithms implemented in MEDICC2, scalability remains a significant bottleneck, as the number of pairwise distance computations grows quadratically with the number of cells. Moreover, the resolution of the copy-number profiles (i.e., the number of bins) can also impact the computational burden, as more bins lead to more complex state spaces in the transducer model. To sidestep this scalability bottleneck, some approaches consider only breakpoints—the genomic locations joining adjacent segments with differing copy numbers. Figure 3.1.1 illustrates the concept of breakpoints in the context of copy-number evolution.

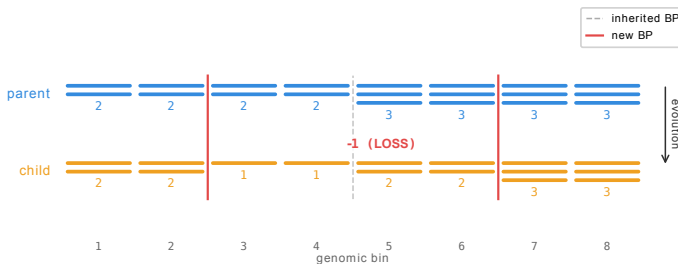


Figure 3.1.1: Figurative representation of copy-number evolution. Each horizontal line depicts one DNA copy within a genomic bin; the integer below each bin states the copy number explicitly. New breakpoints (BP, solid red) mark positions where the child’s segmentation differs from the parent’s, reflecting a copy-number change event. Inherited BPs (dashed gray) are boundaries shared by both profiles.

Breakpoint-based phylogenetics was first developed for bulk-sequencing data [93], leveraging the fact that breakpoints are observed in all cells that descend from the same ancestor where the breakpoint was acquired, i.e., when the copy-number event happens. This marker of evolutionary relationships is particularly useful in the context of bulk-sequencing data, where the observed magnitude of copy-number changes is often confounded by the presence of different subpopulations of cells, making it difficult to accurately estimate copy-number profiles. However, it has since been successfully applied in several single-cell phylogenetic methods, such as Sitka [94], which models breakpoint events as random variables in a probabilistic framework, and explores the space of trees using a Markov Chain Monte Carlo (MCMC) approach. The tree that maximizes the posterior distribution is then selected as the final phylogeny. Although the

method is particularly fast, as it does not require pairwise comparison, the lack of copy-number magnitude information can lead to inaccurate tree reconstructions, especially in the presence of noisy data, where breakpoints may be missed or falsely detected. Another method, DICE [95], revisits the standard distance metrics (e.g., Hamming distance) to define a new distance metric based on breakpoints. In particular, given a vector of copy numbers s , the breakpoint vector s' is defined as the vector of differences between adjacent bins, i.e., $s'_i = s_{i+1} - s_i$. Then, the *DICE-bar* distance between two copy-number profiles s and t is defined as the sum of the square roots of the absolute differences between their breakpoint vectors, i.e., $d(s, t) = \sum_i \sqrt{|s'_i - t'_i|}$. This metric avoids that large copy-number events are weighted more heavily than small events, which happens when applying standard distance metrics to copy-number profiles. The full distance matrix with pairwise DICE-bar distances is then fed into a fast distance-based tree reconstruction algorithm, FastME [96]. Despite the computational efficiency of the DICE-bar distance, it is not clear how well it captures the true evolutionary divergence between cells, especially in realistic noise conditions.

Contribution

Relying on discrete copy-number calls makes current phylogenetic methods highly sensitive to sequencing noise. To address this limitation, we develop a principled probabilistic model of copy-number evolution.

While models for copy-number evolution have certainly advanced—moving from simple geometric distances to sophisticated finite-state transducers like MEDICC2—relying on called copy-number profiles remains fundamentally problematic. Copy-number calling from sparse single-cell data is often inaccurate and lacks resolution, meaning that downstream distance estimation and tree reconstruction are highly sensitive to upstream calling errors. This suggests that a more robust approach would be to estimate distances directly from the raw read counts, without relying on intermediate copy-number calls. By modeling the read counts directly, we can better accommodate the statistical uncertainty inherent in single-cell data, leading to more accurate distance estimates and, consequently, more reliable phylogenetic reconstructions. In the following sections, we elaborate on the nature of single-cell data, how copy numbers can be inferred from it, and, more importantly, how noise can be modeled to enable joint inference of copy-number profiles and evolutionary distances.

3.2 Calling copy-number profiles from single-cell data

As briefly mentioned in Chapter 2, whole-genome sequencing (WGS) technologies do not provide a direct readout of copy-number profiles. Instead, they produce a set of short DNA fragments called *reads*, which are then aligned to a reference genome. Calling copy-number profiles from these reads is a non-trivial task, as the observed read counts are influenced by various sources of noise and technical biases.

In this section, we first review the main challenges of scDNA-seq data based on short reads; then we discuss the concept of segmentation, which is a crucial step in copy number calling, exploring the different approaches in the literature; finally, we focus on how noise can be modeled to enable robust inference of copy-number profiles and evolutionary distances.

3.2.1 The challenges of scDNA-seq data

Single-cell DNA sequencing technologies broadly fall into three categories: short-read sequencing with whole-genome amplification (WGA), short-read direct sequencing (without pre-amplification), and long-read sequencing. WGA-based methods, such as DOP-PCR and the now-discontinued 10x Genomics CNV solution, require very little input DNA and can achieve up to $0.5\times$ coverage. However, the amplification step introduces substantial artifacts and biases that can confound downstream copy-number calling [29]. In contrast, direct sequencing methods, such as DLP+ [27] and ACT [81], circumvent WGA entirely. While they typically yield much lower coverage (around $0.01\times$ to $0.1\times$), they sequence the genome with higher fidelity, producing significantly more uniform coverage distributions. DLP+ has been widely adopted in recent years for large-scale single-cell studies, while ACT provides arguably even lower overdispersion and has been applied effectively to characterize breast cancer tissues. Lastly, long-read technologies (e.g., Oxford Nanopore, PacBio), capable of spanning 10kb–100kb fragments, hold great promise for resolving complex structural variations. However, their high cost and low throughput have so far limited their widespread adoption for single-cell copy-number evolution studies.

Given these trade-offs, we focus primarily on short-read data, and specifically on direct sequencing, which is rapidly becoming the standard for high-fidelity single-cell copy-number analysis. The principal challenge with direct sequencing, and short-read in general, is low coverage, which leads to noisy read counts. Indeed, because of the short reads and the low coverage, the smallest portion of the DNA for which we get statistically meaningful read counts is the bin, which typically spans between 100 kb and 5 Mb. For example, given a bin size $G = 100$ kb and a read length $L = 150$ bp, a uniform coverage of $C = 0.02\times$ implies an expected number of reads $E[N]$ calculated as:

$$E[N] = \frac{C \times G}{L} = \frac{0.02 \times 100,000}{150} \approx 13.33 \quad (3.2.1)$$

This represents very little information for robust statistical inference. Moreover, reads are not uniformly distributed across the genome, a phenomenon characterized by overdispersion. In a standard Poisson model, the variance is constrained such that $\text{Var}[N] = E[N]$. However, in real-world sequencing, the variance is better captured by a model where $\text{Var}[N] = \mu + \alpha\mu^2$, with μ as the mean and α the dispersion parameter. Consequently, the number of reads in a bin can be much higher or lower than the expected value, even in the absence of copy-number changes.

Given a wide enough bin size, the readcount becomes more robust to noise, but it also leads to a loss of resolution, as smaller copy-number events may be missed, although they may be biologically relevant [97, 98].

Besides noise, other technical biases can affect the read counts. A common step in sequencing data analysis is to correct for GC content and mappability biases. In the first case, the number of reads in a bin is influenced by its GC content, with extreme GC content leading to lower read counts. Typically, GC bias is corrected by fitting a regression model to the read counts as a function of GC content and adjusting the counts accordingly. Mappability bias arises from the fact that some genomic regions are more

difficult to sequence and align reads, for instance highly repetitive regions, which can lead to ambiguous read mapping. Mappability bias is often corrected by weighting the read counts based on the mappability of the corresponding bins. Finally, outliers in the read counts can be filtered out using various statistical methods, such as z-score filtering or robust regression techniques. Although these correction steps are essential, the procedures are quite standard and do not directly address the fundamental issue of noise in the read counts. In the next section, we assume that the read counts are easily corrected for these biases, and we focus on how to model the noise in the data to enable robust inference of copy number profiles and evolutionary distances.

Segmentation

A key step in calling copy-number profiles is *segmentation*, which consists of partitioning the genome into contiguous segments of constant copy number. This step may be separated from the copy number calling step, where the copy number of each segment is estimated. There are three main approaches to segmentation, which we briefly review here.

Sliding window This is the simplest approach, where the genome is divided into bins of fixed or variable size, and a statistical test is applied to determine whether the read counts in adjacent bins are significantly different. In that case, a breakpoint is called, and a new segment is created. Various copy-number calling methods employ this approach, such as Ginkgo [99], SCNV [100] and SEACON [101], typically adopting the circular binary segmentation (CBS) algorithm [102].

Objective function This approach formulates segmentation as an optimization problem, where the goal is to find the breakpoints that minimize a certain objective function. For instance, segmentation can be cast as a penalized least-squares problem, where piecewise constant curves are fitted to the observed read counts by minimizing the residual error while penalizing the number of breakpoints; this is precisely how Copy-Number [103] performs segmentation on bulk-sequencing data.

From a statistical perspective, such formulations can be interpreted as maximum likelihood estimation under appropriate noise models, with the penalty term acting as a regularizer that controls model complexity.

Bayesian approaches, in contrast, model segmentation as a probabilistic process, where the breakpoints are treated as random variables. The optimization function is then defined as the posterior distribution over the breakpoints given the observed read counts. Methods such as SCICoNE [104] adopt this approach.

Hidden Markov Models Due to the sequential nature of the genome, Hidden Markov Models (HMMs) are a natural choice for segmentation. Read counts from consecutive bins are modeled as emissions from a hidden state representing the copy number of the bin. The transition probabilities between hidden states capture the likelihood of copy-number changes, while the emission probabilities model the distribution of read counts given a specific copy-number state (as described in the next section). Segments

of constant copy number are then identified as the most likely sequence of hidden states given the observed read counts, which can be efficiently computed using the well-known Viterbi algorithm [105].

HMMcopy [106] and Aneupfinder [107] are two methods that use HMMs for segmentation and copy-number calling. They both model the read counts as emissions from a hidden state representing the copy number of the bin (with states from 0 to 10), but they differ in their choice of emission distributions — Normal for HMMcopy and Negative-Binomial for Aneupfinder. HMMcopy was initially developed for the array Comparative Genomic Hybridization (aCGH) data, but with the advent of Next Generation Sequencing, it found widespread application in single-cell data. When compared to Aneupfinder, HMMcopy is more robust to noise, but it seems to fail in reliably choosing the right ploidy of cells. Also, HMMcopy has been reported to be sensitive to the tuning of its parameters, which requires careful manual calibration [108, 109]. One limitation of both methods is that they do not leverage the shared information across cells to improve the accuracy of copy-number calls. While this approach is attractive for its computational efficiency, it can lead to inaccurate calls, especially in higher noise settings (e.g., for higher resolution copy numbers, hence smaller bin sizes).

Finally, SCONE2 [110] (the successor of SCONE [111]) proposes a probabilistic framework that models the read counts as being generated from copy numbers that evolve following a continuous-time Markov process along the branches of the tree. By integrating over the uncertainty in the copy-number profiles, SCONE2 can estimate evolutionary distances between pairs of cells directly from the read counts, without relying on intermediate copy-number calls. This approach allows for more robust inference of evolutionary relationships between cells, especially in the presence of noisy data.

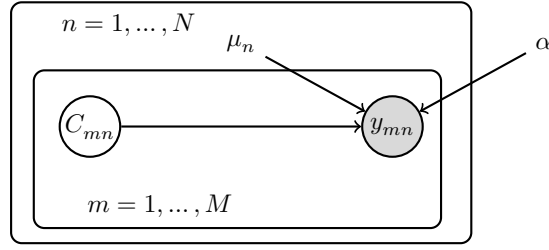
Contribution

Hidden Markov Models, combined with continuous-time Markov processes, provide a statistically principled way to estimate evolutionary distances directly from read counts. We develop such a model; a similar approach was concurrently and independently introduced by the authors of SCONE2.

3.2.2 Modeling noise in copy-number calling

The simplest observation model is to assume the read count of a bin is drawn from a Poisson distribution, where the mean is proportional to the copy number of the bin. However, as we mentioned, the variance of the read counts is often much higher than the mean, and a more appropriate model is the Negative-Binomial, as illustrated in Figure 3.2.2. This requires estimating the *overdispersion* parameter α . Other methods include the Multinomial model, paired with a Dirichlet prior to model the distribution of reads across bins, or a Normal distribution over normalized read counts.

Ploidy estimation This is required to distinguish between cells with different ploidy levels, for instance diploid vs tetraploid cells, which can be a common occurrence in tumors. If a cell has undergone a whole-genome duplication (WGD), its copy-number



$$y_{mn} \sim \text{NegBinom}(\mu_n C_{mn}, \alpha)$$

Figure 3.2.2: Example of a generative model (DAG) of the observed read count y_{mn} for cell n at bin b , where the counts are assumed to be drawn from a Negative-Binomial distribution with mean $\mu_n \times C_{mn}$ and overdispersion α . The mean is therefore modeled as a function of the copy number C_{mn} , which is a latent variable representing the true copy-number profile of the cell.

profile will be double that of a diploid cell. However, the total number of reads that are observed in a cell, also known as read depth, does not double accordingly, because it is influenced mainly by the sequencer’s capacity. In practice, this decoupling of physical DNA content from sequencing output creates a significant challenge: the data lacks an absolute reference scale. Because total read depth is constrained by the sequencer’s capacity, a WGD cell’s reads are redistributed across a genome of size $G_{WGD} = 2 \times G_{Diploid}$, effectively “diluting” the signal per locus. As illustrated in Figure 3.2.3, a doubled tetraploid profile can easily be misinterpreted as a diploid profile. The fundamental difficulty lies in distinguishing a stable diploid state ($n = 2$) from a doubled tetraploid state ($n = 4$) without external anchors often leading models to incorrectly fit the data to a lower ploidy state [108].

It is therefore necessary to estimate the ploidy level of each cell to correctly interpret the variation in read counts and accurately call copy-number profiles. Calibrating the ploidy estimation is crucial, and typically involves fitting different ploidy models to the data and selecting the one that best explains the observed read counts. Intuitively, the higher the variation in read counts across bins, the more likely it is that the cell has a higher ploidy level.

There are, however, some external anchors that can help to resolve ploidy ambiguity. With the DLP+ platform, it is possible, for instance, to obtain an estimate of the diameter of the nucleus through imaging [27]. Since the diameter is proportional to the genetic content of the cell, it can be used as an external reference to calibrate ploidy estimation. However, this is not always accurate, especially for cells in S-phase, which are actively replicating their DNA [112]. A more effective approach is to leverage the B-allele frequency (BAF), which is the proportion of reads supporting the alternative allele at heterozygous single-nucleotide polymorphisms (SNPs). In the next paragraph, we discuss how BAF can be used to infer ploidy levels in single-cell DNA sequencing data.

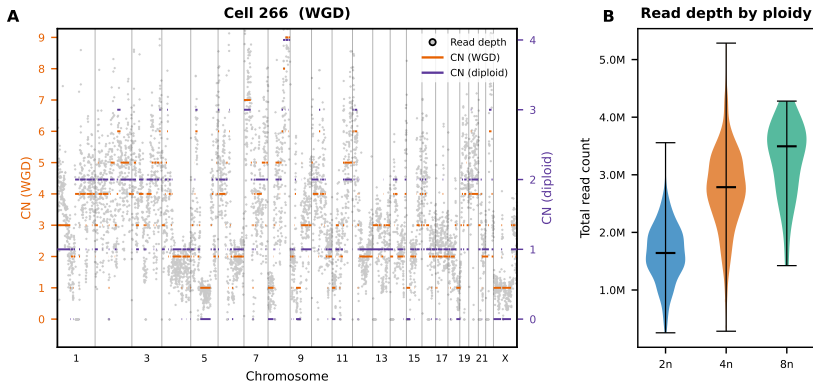


Figure 3.2.3: Whole-genome doubling inflates copy-number states without proportionally increasing read depth. **(A)** Genome-wide read depth (grey, normalised to copy-number scale) and inferred copy-number profile (orange) for one representative WGD cell from the SPECTRUM-OV-075 single-cell whole-genome sequencing dataset with DLP+ protocol [92]. The purple line draws the integer part of the inferred copy number divided by two, illustrating the ambiguity introduced by genome doubling. Vertical grey lines demarcate chromosome boundaries. **(B)** Violin plots of total per-cell read counts stratified by ploidy class (diploid, tetraploid, and octaploid). Although mean ploidy increases twofold at each step, the distributions of total read depth overlap substantially, demonstrating that sequencing depth is not linearly proportional to ploidy, and that read counts alone cannot be used as a proxy for WGD status.

B-allele frequency and phasing We have already established that scDNA-seq data cannot be used to accurately call point mutations due to the low coverage (whether they are somatic or not). However, it is still possible to leverage the information from heterozygous SNPs to infer the B-allele frequency (BAF) over a large enough bin so that the number of observed SNPs is sufficient to obtain a robust estimate. The BAF is defined as the proportion of reads supporting the alternative allele. In a certain bin, if the copy number is n and the number of copies of the alternative allele is k , then the expected BAF is k/n . In cases where it is not important to distinguish between the two alleles, the BAF can be mirrored around $1/2$ to obtain the mirrored BAF (mBAF), which can be defined as $mBAF = \max(BAF, 1 - BAF)$. From Figure 3.2.4, we can see that the SNPs imbalance captured by the read counts of the alternative and reference alleles can be used to distinguish between different ploidy levels. In fact, when only looking at the total read counts in Figure 3.2.3, chromosome 13 could be interpreted as a haploid (one copy instead of two), but as we plot the imbalance between the reads mapping to the alternative and reference alleles, it is clear that the total copy number must be at least 3 in order to generate a mBAF that is neither 0.5 nor 1.

With a procedure called *phasing*, it is possible to determine which alleles belong to the same parental haplotype. Tools such as SHAPEIT [113] and Eagle2 [114] are used in many standard bioinformatics pipelines to perform phasing on both bulk and single-cell data by mapping the observed genotypes to a reference panel of haplotypes. The output of these tools is a phased variant call format (VCF) file, which contains the phased genotypes for each SNP, indicating which alleles belong to the same haplotype.

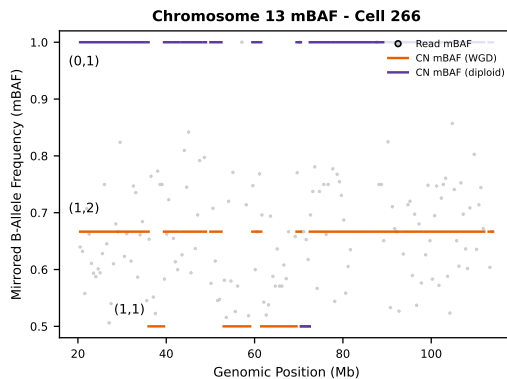


Figure 3.2.4: Allele-specific copy-number profile for chromosome 13 in a whole-genome doubled cell. Mirrored B-allele frequency (mBAF) is shown across the genomic position (in Mb) for a representative WGD cell from the SPECTRUM-OV-075 dataset. Grey dots represent the empirical mBAF calculated directly from raw allele-specific read counts. Solid lines indicate the expected mBAF corresponding to the inferred allele-specific copy-number states: the orange line shows the expected mBAF under the WGD model fit, while the purple line shows the expected ratio if the cell were strictly diploid.

These phased genotypes can then be used to assign the observed read counts to specific haplotypes, generating haplotype-specific read counts. In this case, the non-mirrored BAF becomes informative, as values different from $1/2$ correspond to copy-number changes affecting a specific haplotype.

Building on this principle, CHISEL [115] was the first method to leverage the BAF to perform haplotype-specific copy-number calling. The method uses both total read counts and B-allele frequencies from phased read counts over 5Mb bins. By clustering (normalized) read counts and BAFs across bins and cells using k-means, CHISEL identifies groups of bins that share the same allele-specific copy-number states. With a dynamic-programming algorithm, these copy-number states are assigned to the haplotypes consistently across cells. The main limitation of this method is that the accuracy in copy-number calls drops significantly when the bin size is smaller than 5Mb, which makes it unable to spot smaller copy-number events, especially in rare subpopulations of cells. SIGNALS was later developed [116] to overcome this limitation. By fitting a Hidden Markov Model (HMM) with Beta-Binomial modeling the haplotype-specific copy number, SIGNALS can call copy-number profiles at a higher resolution (500kb bins) than CHISEL. However, this method relies on total copy-number profiles to be called beforehand (e.g., with HMMcopy), therefore does not leverage phased read counts to choose the optimal ploidy and refine the total copy-number profiles. A more recent method, SEACON [101], improves slightly on CHISEL by using Gaussian Mixture Models (GMMs) instead of k-means for clustering the bins, and by identifying break-points across cells rather than for each cell independently, which allows for a more balanced trade-off between precision (avoiding false positives) and recall (avoiding false negatives) in the inferred copy-number profiles.

While these observation models are essential for identifying copy-number profiles

(CNPs) in isolated cells, understanding the developmental history of the tumor requires us to measure the evolutionary divergence between these cells.

3.3 Distance-based tree reconstruction and the missing root

Once we establish a distance measure between cells, that is capable of capturing the true evolutionary divergence in terms of copy-number changes, either from the called copy-number profiles or directly from read counts, the next step is to reconstruct a tumor phylogeny that agrees with the observed data.

3.3.1 Distance-based tumor phylogenies

This step is overlooked in many research articles, where distance measures between copy-number profiles are estimated and then fed into standard distance-based tree reconstruction algorithms, or simply ordered in a dendrogram using off-the-shelf hierarchical clustering methods [115, 87, 117, 95, 101]. However, this approach is sub-optimal, and in most cases it leads to wrong tree topologies [118, 45]. Hierarchical clustering methods like HDBSCAN [119] group cells based on their similarity, but they do not explicitly model the evolutionary process, ignoring cases of convergent evolution or different rates of evolution across lineages. Neighbor Joining (NJ) is a principled distance-based method that constructs an evolutionary tree and, as described in Section 2.3.2, does so being aware of the “long branch attraction” problem, thus providing more accurate tree reconstructions than hierarchical clustering.

The success of NJ in classical phylogenetics has led to its widespread adoption in the context of copy-number evolution, where it often serves as the default tree reconstruction algorithm. It is fast and typically robust to noisy distance matrices, making it a popular choice for large datasets. However, its accuracy critically depends on the quality of the input distances. As pointed out by Mallory et al. [108], distances based directly on the observed differences between two copy-number profiles—such as the Hamming or Manhattan distance—can grossly underestimate the true pairwise evolutionary distance. This underestimation occurs because simple metrics fail to account for multiple, overlapping mutations that may have occurred at the same genomic segment over time.

To properly reflect the sum of copy-number aberrations (CNAs) on the paths from the cells’ common ancestor, a distance correction is required, which in turn necessitates a stochastic model of evolution. The Minimum Event Distance (MED), utilized by methods like MEDICC2 [87], attempts to bridge this gap by explicitly inferring the copy-number profile of the least common ancestor to determine the minimum number of events separating two cells. While this is a vast improvement over naive geometric distances, it still operates under a parsimony framework and lacks a true stochastic model of evolution over continuous time. In contrast, probabilistic methods like SCONE2 [110] provide a more rigorous approach. By modeling copy number evolution as a continuous-time Markov process, SCONE2 can naturally compute evolution-corrected distances that integrate over the uncertainty of intermediate states and varying evolutionary rates.

Even then, the use of standard neighbor joining is sub-optimal for copy-number trees.

The fundamental issue is a crucial biological oversight: unlike standard species phylogenies, where the root is entirely unknown, tumor evolution always originates from a known root—the healthy diploid ancestor of the patient. Unrooted algorithms like NJ are explicitly designed to reconstruct topologies without assuming a root, making them oblivious to the directionality of copy-number changes from the normal state.

This information can be incorporated by using distance-based algorithms designed specifically for rooted phylogenies. The Deepest Least Common Ancestor Neighbor Joining (DLCA-NJ) algorithm [120] adapts the classical NJ framework by utilizing distances not only between pairs of taxa but also their distances to a defined root. By iteratively joining pairs of nodes while explicitly enforcing the distance constraints relative to the diploid root, DLCA-NJ ensures that the reconstructed tree remains biologically coherent with respect to the direction of evolution.

Contribution

A suitable algorithm that explicitly models the healthy diploid ancestor as the root already exists (DLCA-NJ), but it was never adopted in the field. Experiments show that it can significantly improve the accuracy of tree reconstruction.

3.3.2 Scalability of tree reconstruction algorithms

With the increased availability of large single-cell datasets, the computational scalability of tree reconstruction algorithms becomes a critical concern. For n cells, distance-based tree reconstruction typically runs with a time complexity of $O(n^3)$ (this is the case of NJ, FastME and DLCA-NJ). When the computation of the distance matrix is done with a simple, vectorized distance metric, such as the Hamming or Manhattan distance, the tree reconstruction step is the main bottleneck. However, when the distance matrix is computed with a more complex, model-based distance metric, such as the MED distance or the SCONE2 distance, the burden of computation shifts to the distance matrix. Computing the full distance matrix requires $O(n^2 f(m))$ time, where $f(m)$ is the time complexity of computing the distance between two cells considering m bins. In the case of the MED distance, $f(m)$ is the time required to find the shortest path through the finite-state transducer. With the lazy-composition strategy implemented in MEDICC2, the time complexity of computing the MED distance is shown to be linear in the number of bins (or *segments*, in MEDICC2 terminology) or, in the worst case, log-linear [121]. The simulation experiments in the original paper do not exceed the length of 200 bins, but in real datasets with bin size of 500kb, the number of bins is typically around 6000. When copy numbers are estimated at a higher resolution, the computation of the MED between each pair of cells can become prohibitively expensive, especially when the number of cells is in the order of thousands.

This is also the case for the SCONE2 distance, which requires integrating over the latent copy-number profiles of the common ancestor. Specifically, if $K = 10$ is the number of possible copy-number states for each bin (a tunable parameter), then the time complexity of computing the SCONE2 distance for each pair of cells is $O(mK^4)$. This is because the algorithm needs to consider all possible combinations of sequential pairs of copy-number states for the two cells when evaluating the forward likelihood of

the copy-number HMM.

As a result, the more sophisticated a distance measure is, the more computationally expensive it becomes to compute the full distance matrix, which in turn limits the scalability of tree reconstruction algorithms that rely on it. To address this issue, the Sparse Neighbor Joining (SNJ) [122] algorithm was developed. Briefly, instead of requiring the entire $O(n^2)$ matrix of distances upfront, SNJ reconstructs the tree by evaluating only a sparse subset of distances. By focusing computations on the nearest evolutionary neighbors and inferring missing distances dynamically as nodes are joined, the algorithm significantly reduces the time complexity down to $O(f(m)n \log n)$, allowing for increased scalability of the overall phylogenetic inference process.

Contribution

Computing full distance matrices using complex models does not scale to thousands of cells. Sparse matrix approaches trade off some accuracy for major runtime improvements; extending this sparse framework to the rooted (DLCA) setting is a central technical contribution of this thesis.

3.4 Summary of contributions

The first part of this thesis focuses on rectifying the major flaws in the standard distance-based phylogenetic pipeline: the propagation of errors from pre-estimated profiles, the algorithmic failure to utilize the known evolutionary root, and the excessive computational burden of complex distance models for large datasets. To address these issues, we developed two interrelated methods, Cellmates and SparseRNJ.

Cellmates: probabilistic distance estimation from read counts To solve the pipeline error problem, we bypass the fragile intermediate step of discrete copy number calling entirely. We developed Cellmates (Paper A), a novel Hidden Markov Model framework that estimates biologically meaningful evolutionary distances directly from noisy, raw read counts. By integrating over the latent copy number states of the cells and their shared ancestor, Cellmates effectively accounts for the complex dynamics of copy number evolution, acquiring statistical power directly from the raw data.

SparseRNJ: Scalable rooted tree reconstruction To properly incorporate the root information and the address the scalability bottlenecks of complex distance models, we developed Paper B, SparseRNJ [2]. This algorithm takes the statistically principled DLCA-NJ method—which strictly enforces a known root—and adapts it into a sparse matrix framework. Instead of evaluating all pairs of nodes, SparseRNJ computes and utilizes only a subset of the most critical distances. We prove that this sparse approach retains the theoretical guarantees of DLCA-NJ while dramatically reducing the time complexity, therefore scaling rooted copy number tree inference to thousands of cells even with computationally intensive distance measures.

Discussion The impact of SparseRNJ on the performance of Cellmates, for which it was specifically designed, is still under additional investigation. However, preliminary

results show that the combination of these two methods can achieve accurate tree reconstructions on large datasets, where the use of standard algorithms like NJ or DLCA-NJ would be computationally infeasible.

While these contributions on distance-based phylogeny reconstruction represent a major leap over naive pipelines, they inherently compress complex genomic sequence data into single scalar distances. This compression inevitably discards information and limits rigorous joint uncertainty quantification, which motivates the shift toward fully probabilistic joint inference frameworks explored in the next chapter.

4 Joint Bayesian inference in tumor phylogenetics

While the distance-based methods explored in the previous chapter are fast and highly scalable, they have a fundamental limitation: by compressing sequence data into a scalar distance matrix, they lose information and weaken the direct connection between raw observations and inferred topology. In noisy single-cell data, this makes it difficult to fit mechanistic evolutionary models in a statistically principled way. Generative Bayesian approaches address this issue by explicitly modeling both the data-generating process and sequencing noise, enabling principled inference and uncertainty quantification [123].

The ultimate flaw of traditional phylogenetic pipelines—calling discrete states, estimating distances, and then building a tree—is that early decisions cannot be reversed. If a breakpoint is misidentified or a copy number is called incorrectly due to noise, that error propagates downward, distorting the final tree without any mechanism for correction.

In this chapter, we explore the transition from pipeline-based methods to comprehensive joint Bayesian inference, where the observation model and the evolutionary history are unified within a single probabilistic framework. The main focus will be on the challenges in the context of copy-number aware tumor phylogenetics from single-cell DNA data, but the principles and techniques are broadly applicable across various types of genomic data and evolutionary models.

The chapter is outlined as follows. In Section 4.1, we review Bayesian models of clonal evolution, discussing their advantages and inherent challenges. Section 4.2 introduces Variational Inference (VI) for tree topologies, presenting a scalable alternative to traditional MCMC methods. In Section 4.3, we briefly review the problem of sampling spanning trees from a graph, and discuss the specific challenges posed by the skewed edge weights that arise in the context of tumor phylogenetics. Finally, Section 4.4

summarizes the specific thesis contributions related to variational inference for copy-number based tumor phylogenetics, including the development of a novel algorithm for stable sampling of spanning arborescences and the implementation of VICTree, a comprehensive joint inference method.

4.1 Bayesian models of clonal evolution

In the Bayesian framework, the central goal is posterior inference: we aim to infer the posterior distribution over the latent variables of interest, such as the tree topology, branch lengths, clonal structure, and copy-number profiles, given the observed data (e.g., read counts from single-cell sequencing). This involves defining a generative model that captures the biological processes underlying tumor evolution and the technical noise in the data.

Given the data D , we want to infer the posterior distribution over the tree T , and some latent variable Z which models, for instance, the assignment of cells to clones and their associated mutational profiles. This can be expressed mathematically as

$$p(T, Z|D) = \frac{p(D|T, Z)p(T, Z)}{p(D)}. \quad (4.1.1)$$

As already discussed in Chapter 2, the likelihood $p(D|T, Z)$ captures the data-generating process, while the prior $p(T, Z)$ encodes our assumptions about the evolutionary history and clonal structure. The denominator $p(D)$ is the marginal likelihood of the data, which is typically intractable to compute directly.

Crucially, the likelihood and the prior are deeply intertwined through the latent biological states. For example, the likelihood of observing specific read counts in a single cell depends entirely on its latent copy-number profile. In turn, this profile is not independent but is governed by the tree structure and the cell’s clonal assignment, which act as priors over the genomic states. Consequently, we cannot separate the inference of the evolutionary tree from the estimation of the copy-number profiles; they must be jointly inferred to ensure that the physical observation model correctly informs the phylogenetic history, and vice versa. Several studies have already highlighted the importance of performing joint phylogenetic inference, where the tree is not only the final output of a downstream analysis, but more generally a nuisance parameter which aids the inference of complex models of tumor evolution [124, 125].

In this section, we will revisit the key components of the generative model. After briefly reviewing the data likelihood and the copy-number models, which have already been covered in Chapter 3, we will delve into the modeling of the clonal structure characterizing the mutational profiles and the tree that represents the evolutionary history.

4.1.1 The Bayesian framework for tumor evolution

In a fully Bayesian setting, all unknown quantities are treated as random variables. This includes not only the tree topology and clonal structure, but also the parameters of the observation model, such as copy-number states and noise parameters. Priors therefore

play a central role, especially in tumor phylogenetics where the data is noisy and limited. They act as a regularizing force, preventing overfitting and guiding inference toward biologically plausible solutions.

A key challenge in this setting is multimodality. For instance, as discussed in Chapter 3, different ploidy levels can produce similar relative copy-number profiles, leading to multiple modes in the likelihood landscape. This makes posterior inference particularly difficult and highlights the importance of structured priors and joint modeling.

Model complexity is another critical aspect. The number of clones is typically unknown, and naive approaches may overfit by introducing spurious subpopulations. Bayesian methods naturally address this through the marginal likelihood (or its approximations), which penalizes overly complex explanations of the data.

Finally, recent work has emphasized the importance of joint inference across multiple latent variables and data modalities. Methods such as PhylEx [126] and PhyloWGS [127] illustrate how integrating bulk and single-cell data, or combining SNVs and CNAs, leads to more robust and informative models of tumor evolution.

4.1.2 Data likelihood and copy-number models

The data likelihood and copy-number models have been introduced in Chapter 3, and we briefly recall the key elements here.

In single-cell DNA sequencing, the observed data typically consists of read counts aggregated over genomic bins. These counts are modeled using probabilistic distributions whose parameters depend on the underlying copy-number state. While a Poisson model provides a simple starting point, it fails to capture the overdispersion observed in real data. A more appropriate choice is the Negative Binomial distribution, which introduces an additional dispersion parameter to account for variability beyond the mean.

Copy-number profiles themselves are latent variables that must be inferred. Due to the sequential structure of the genome, they are sometimes modeled using Hidden Markov Models (HMMs), where each hidden state represents a copy number, and emissions correspond to observed read counts. This allows one to capture both spatial dependencies along the genome and the stochastic nature of the sequencing process.

Importantly, these copy-number states are not independent across cells. They are coupled through the clonal structure and the evolutionary tree, which define how genomic alterations are shared and inherited. As a result, the likelihood $p(D \mid T, Z)$ depends on the tree and clonal assignments through these latent copy-number profiles, reinforcing the need for joint inference.

4.1.3 Clonal structure

Clones are groups of cells that share a common set of mutations. When modeling tumor evolution, we often assume that cells within the same clone are genetically identical, and that the evolutionary history can be represented as a tree where each node corresponds to a clone and edges represent ancestral relationships. Due to the intra-tumor heterogeneity (ITH) typically observed in many cancer types, this assumption is

likely violated, with different cells within the same clone exhibiting distinct mutational profiles. However, inferring the clonal structure is still a critical step in understanding tumor evolution, as it provides a simplified representation of the complex evolutionary dynamics and allows us to identify key subpopulations of interest.

In mathematical terms, we can represent the clonal structure as a partition of the set of cells into subsets (clones), each with its own set of mutations (SNVs and CNAs). The evolutionary history can then be represented as a tree where each node corresponds to a clone, and edges represent ancestral relationships between clones. If we isolate the clonal assignment problem, we can see it as a clustering problem, where we want to group cells based on their genetic similarity. Specifically, given the

In bulk sequencing, the clonal structure must be deconvolved from the observed mixture, which is a challenging problem in itself. PyClone [128] is a pioneering method that uses a Bayesian nonparametric model — the Dirichlet process (DP) — to infer the clonal structure from bulk sequencing data. It models the observed variant allele frequencies as a mixture of subclonal populations, allowing for flexible inference of the number of clones and their evolutionary relationships. Here copy numbers are assumed to be known, previously estimated from the data, and are used to define the priors over the data likelihood parameters. The study, however, suggests that PyClone is somewhat sensitive to the copy-number predictions used to inform the prior. This highlights the importance of integrating copy-number inference directly into the joint model, rather than relying on separate estimation steps that can introduce errors.

In single-cell sequencing, we have the advantage of observing individual cells, but we still need to infer which cells belong to which clones and how those clones are related evolutionarily. Cells are typically assigned to clones based on their mutational profiles the goal of inference is to find the probability that a given cell belongs to a particular clone.

BiTSC2 [129] is the first method that performs joint inference of the clonal structure and the tumor phylogeny based on SNVs and CNAs. It employs a Markov Chain Monte Carlo (MCMC) algorithm to jointly sample the tree topology, clonal assignments, and mutation matrices. By integrating the likelihoods of both mutation types into a unified scoring function, the model navigates the joint posterior space. The methods is proven to be more accurate than the integer linear programming (ILP) counterpart, SCARLET [130], which, at the time, was the state-of-the-art method for combining SNV and CNA mutations for tumor phylogenetics from scDNA-seq data. However, BITSC2 belongs to the set of methods that only consider CNAs relevant as support for somatic point mutations, therefore modelling them at the locus level and not throughout the whole genome.

Another method that performs joint Bayesian inference is CONET [131]. It relies on pre-estimated candidate breakpoints but distinguishes itself by jointly inferring the tumor phylogeny and performing copy-number calling within a unified probabilistic Bayesian framework. The method utilizes two inputs: per-breakpoint data, representing read count differences at candidate loci, and per-bin data, which is used to regularize the tree structure through a count discrepancy penalty. In CONET, the latent topology is a mutation tree: branches represent copy number events and cells are assigned to nodes based on their copy-number profiles. The number of different clones is not fixed,

but rather determined by the tree structure. Trees inferred by CONET are typically large, with many empty nodes. While allowing for a fine-grained representation of the evolutionary history, this can lead to overfitting and poor interpretability, especially when the data is noisy and sparse.

Contribution

We devise a novel probabilistic model for joint inference of the clonal structure of tumors based on single-cell DNA sequencing by modeling copy-number dependencies across the genome and across clones.

Overall, these methods show that the core problem is to identify a meaningful clonal structure from noisy data, while the tree provides a way to organize these clones through ancestral relationships. Even when the exact tree is uncertain or varies across models, imposing a hierarchical structure helps enforce consistency in how mutations accumulate and constrains the space of possible solutions. In this sense, the tree is not the primary object of interest, but a useful representation that makes the inferred clonal structure more interpretable and biologically coherent. This naturally leads to the question of how to model and explore the space of such trees, which we address in the next section by reviewing common probabilistic approaches used in the literature.

4.1.4 Generative models for tree space

Tree modeling in cancer aims to represent the evolutionary relationships among tumor populations while accommodating different levels of detail, from coarse groupings of cells into clones to finer representations at the level of individual mutations or cells. Rather than distinguishing rigid categories, we view trees as structures where nodes correspond to latent cellular populations (e.g., clones) and edges encode ancestral relationships, with observed data providing indirect evidence of this hierarchy. The key challenge is to define probabilistic models over such tree spaces that can adapt their complexity to the data, enabling inference of both the number of populations and their evolutionary organization without imposing strong structural constraints a priori.

The tree-structured stick-breaking [132] (TSSB) is a common prior used in Bayesian nonparametric models to define a distribution over tree structures. It is a generalization of the stick-breaking process used in Dirichlet processes, allowing for the construction of hierarchical models where the number of branches at each node can be potentially infinite. This prior is particularly useful in tumor phylogenetics, as it allows for flexible modeling of the clonal structure and evolutionary history without requiring a fixed number of clones or tree depth. PhyloWGS [127] is an example of a method that uses a tree-structured stick-breaking prior to model the clonal evolution of tumors from bulk sequencing data. It infers the clonal structure and evolutionary relationships by modeling the observed variant allele frequencies as a mixture of subclonal populations, allowing for flexible inference of the number of clones and their evolutionary relationships. PhylEx [126] builds on this framework to perform joint inference of the clonal structure and tumor phylogeny from single-cell and bulk sequencing data, demonstrating the power of this approach for integrating multiple data modalities and

capturing the complex evolutionary dynamics of tumors. A limitation of the TSSB prior is that it can lead to overly complex tree structures, especially when the data is noisy and sparse.

CONET [131] and SCiCoNE [104] explore the tree space by defining a set of local moves that allow them to traverse the space of possible tree topologies. These moves typically involve operations such as adding or removing nodes, changing the parent-child relationships between nodes, or swapping subtrees. By iteratively applying these moves and evaluating the likelihood of the resulting tree given the data, these methods can explore the tree space and identify high-probability regions. For downstream analyses which require a single tree, they typically select the maximum a posteriori (MAP) tree, which is the tree with the highest posterior probability given the data. However, this approach does not provide a full characterization of the uncertainty in the inferred tree, and can be sensitive to local optima in the tree space. This approach to tree inference is common among many methods in the literature, including known methods for single-cell phylogenetics based on somatic point mutations rather than copy numbers [133, 134].

While MCMC is widely adopted in Bayesian phylogenetics, it has severe limitations in the tumor setting. The state-space of possible phylogenetic trees is massive and grows super-exponentially with the number of cells. Consequently, MCMC suffers from poor mixing and combinatorial explosion, making it extremely inefficient and often computationally intractable for large-scale single-cell datasets.

In these settings explored so far, the posterior distribution over trees does not have a closed-form expression, therefore we can only explore it through sampling. Such models are therefore hard to adapt to a Variational Inference framework, which requires a tractable variational family and the ability to compute expectations over it. To adapt such models to a Variational Inference (VI) framework, one must construct a tractable variational family over the tree space and establish a method to compute or smoothly estimate expectations over this combinatorial structure. As we will discuss, representing the tree as a set of edge probabilities over a fully connected graph of clones provides a pathway to achieve this.

4.2 Scalable joint inference from sequencing data

As single-cell sequencing datasets scale to thousands of cells, traditional MCMC approaches become computationally difficult not only because of tree-space size, but because copy-number models induce strong dependencies across genomic bins and clonal lineages. This coupled latent structure makes joint exploration of topology and copy-number configurations particularly hard. There is a pressing need for scalable inference methods capable of handling the massive complexity of joint phylogenetic models without sacrificing the rigorous uncertainty quantification provided by the Bayesian framework. In this section, we introduce Variational Inference (VI) as a powerful alternative to MCMC, detailing how it transforms the integration over latent variables into an optimization problem.

4.2.1 VI framework

Variational Inference (VI) approaches Bayesian phylogenetics by approximating the exact posterior with a parameterized family of distributions. The objective is to find the member of this family that minimizes the Kullback-Leibler (KL) divergence to the true posterior.

The key idea behind VI is to define a family of tractable distributions (the variational family) that can approximate the true posterior distribution over the latent variables of interest. By optimizing the parameters of the variational distribution to maximize the Evidence Lower Bound (ELBO), we can obtain a good approximation to the true posterior without needing to sample from it directly. This approach is particularly appealing in the context of tumor phylogenetics, where the state-space of possible trees and clonal structures is enormous, and MCMC methods struggle to explore it effectively.

PyClone-VI [135] is one of the method that adopted VI to improve the scalability of an existing MCMC-based method, PyClone. By replacing the MCMC inference with VI, PyClone-VI achieves a significant speedup and increases the accuracy of clonal deconvolution, demonstrating the practical benefits of this approach. As its predecessor, PyClone-VI does not integrate tree inference in the model, but rather relies on a separate step to infer the tree structure based on the inferred clonal structure.

Inferring trees with VI Variational inference has historically been difficult to apply to phylogenetic models because the tree likelihood is typically non-conjugate with respect to priors over other the tree and other parameters, preventing closed-form coordinate updates and tractable expectations under the variational distributions.

Sequential Monte Carlo (SMC) methods have been proposed as a way to perform inference over tree space [136, 137, 138] by approximating the posterior distribution with a set of weighted particles that represent different tree topologies. However, these methods can be computationally expensive and may struggle to scale to large datasets, especially when the tree space is vast and complex.

A more recent approach to tree inference is VBPI [139, 140] based on the author’s previous work on subsplit Bayesian networks (SBNs) [141]. The key idea is to define a variational family over tree topologies using a Bayesian network that represents splits in the tree, which allows for efficient computation of the ELBO and optimization of the variational parameters via stochastic gradient ascent. This factorization effectively reduces the complexity of inference from super-exponential to polynomial time with respect to the number of nodes, making VI computationally feasible.

VaiPhy [142] proposes a variational inference framework for even faster phylogenetic inference. The method employs a sequential importance sampling (IS) scheme to compute expectations over the unrooted tree space. The sampling scheme, inspired by [143], distinguishes itself in the context of phylogenetics by a non-local proposal distribution. The edges of the proposed tree are added sequentially with Bernoulli trials, where the probability of adding an edge is proportional to the weight of the maximum spanning tree (MST) that can be formed by adding that edge to the current partial tree. VaiPhy demonstrates notable improvements in scalability, but, like the other methods

mentioned, it is designed for unrooted trees and does not directly address the unique challenges of tumor phylogenetics, such as the need to model rooted trees.

Contribution

We introduce a novel edge-factorized variational distribution over spanning rooted trees (arborescences), enabling direct and scalable variational inference of clonal tree topology based on copy numbers.

4.2.2 ELBO and model selection

Variational Inference inherently provides a framework for model selection through the ELBO [144]. By maximizing the ELBO, VI jointly optimizes the fit to the data (expected log-likelihood) while penalizing the complexity of the variational distribution (the KL divergence from the prior). This acts as a mathematically rigorous application of Occam’s razor. In the context of tumor phylogenetics, the ELBO provides a principled, easily computable metric to compare models of varying complexity, such as determining the optimal number of clones K in a highly heterogeneous tumor, a task that typically requires computationally expensive marginal likelihood estimations in MCMC frameworks.

Figure 4.2.1 illustrates this principle on a breast cancer dataset of approximately 10^4 single cells from a single tissue, sequenced with 10x Genomics scDNA-seq [145]. For each candidate clone number K , the model is fit from multiple random initializations, and the final ELBO is summarized by its mean and standard deviation across runs. This example shows the typical behavior expected when ELBO is informative for model selection: a stable, peaked trend over K with a clear maximum at $K = 9$.

Contribution

We demonstrate the utility of the ELBO for model selection in tumor phylogenetics, providing a rigorous framework for determining the optimal number of clones and tree complexity.

4.3 Sampling spanning arborescences

Sampling tree topologies is a fundamental component of both MCMC and VI approaches to phylogenetic inference. As already discussed, in the context of tumor phylogenetics based on copy-number changes, we can leverage the fact that all tumor subclones originate from a common ancestor, which allows us to represent the evolutionary history as a spanning arborescence with a fixed root corresponding to the normal diploid clone.

Proposal-based strategies such as importance sampling or combinatorial sequential Monte Carlo approximate expectations over tree space by drawing candidate topologies from an auxiliary distribution and reweighting them to match the target posterior. The effectiveness of these approaches depends on how well the proposal distribution

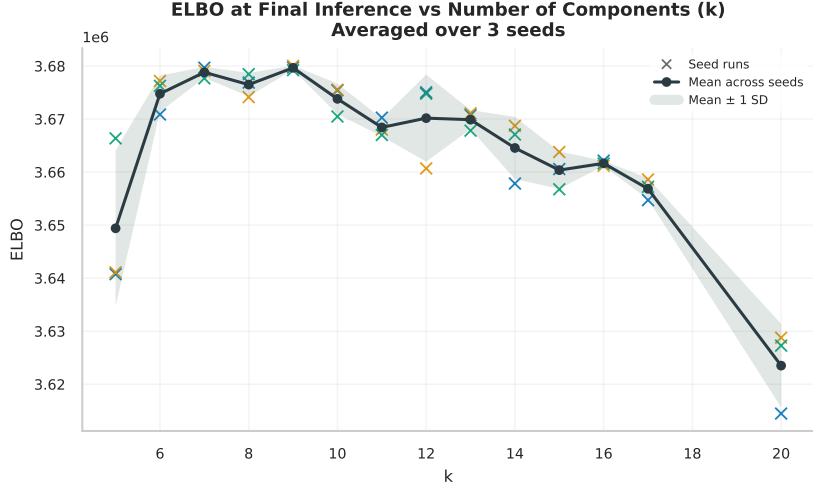


Figure 4.2.1: Illustration of ELBO-driven model selection on a breast cancer 10x Genomics scDNA-seq dataset [145]. Each marker denotes the final ELBO from one run at a fixed clone number K ; the solid line shows the mean across runs and the shaded region indicates one standard deviation. The averaged ELBO profile is convex and attains its maximum at $K = 9$, exemplifying how ELBO can identify a well-supported model complexity.

covers high-probability regions of tree space, which typically fails in the context of tumor phylogenetics due to the vastness of the tree space [146]. In contrast, when the number of nodes K is fixed, the edge-weight parameterization above defines a normalized distribution directly over spanning arborescences, enabling sampling from the variational posterior itself rather than from an external proposal.

This assumption makes inference over tree space computationally tractable, as the dimensionality of the combinatorial space and the number of variational parameters both scale rapidly with K , while still allowing uncertainty over all admissible topologies at the chosen resolution. If we consider the space of all possible tree topologies that can be formed by connecting the observed clones, we can represent this space as a graph where nodes correspond to clones and edges represent potential ancestral relationships.

To define a tractable variational posterior over this combinatorial space, we parameterize a distribution over rooted spanning trees using edge-specific weights. In particular, let w_{uv} denote a non-negative weight associated with a directed edge from clone u to clone v . These weights define a structured variational distribution over arborescences of the form

$$q(T) \propto \prod_{(u,v) \in T} w_{uv}, \quad (4.3.2)$$

where the product runs over edges in the sampled tree T . In the context of Bayesian inference, the weights constitute the free parameters of the target tree distribution, and they can be learned, for instance, via variational updates which maximize the ELBO

as described in the previous section. Expectations with respect to $q(T)$ can then be estimated by sampling spanning arborescences from this distribution.

Sampling random spanning trees from a graph with edge weights is a well-studied problem in graph theory, and brief overview of the relevant algorithms is provided in Chapter 2.

However, sampling spanning arborescences from biological graphs presents significant graph-theoretical and numerical challenges. Specifically, weights that are learned by optimizing complex models of tumor evolution from noisy single-cell data often exhibit weakly connected structures with highly skewed edge weights (see Figure 4.3.2) Some evolutionary transitions are highly probable, while others are biologically implausible, resulting in edge weights that span many orders of magnitude. Moreover, the induced graphs can exhibit low conductance (as defined in Section 2.4), meaning that probability mass is weakly connected across regions of the graph. When applying standard sampling algorithms based on the Matrix-Tree theorem, such as Wilson’s algorithm or classical determinantal methods, these skewed weights frequently cause severe numerical underflow, rendering the sampling process unstable or entirely broken.

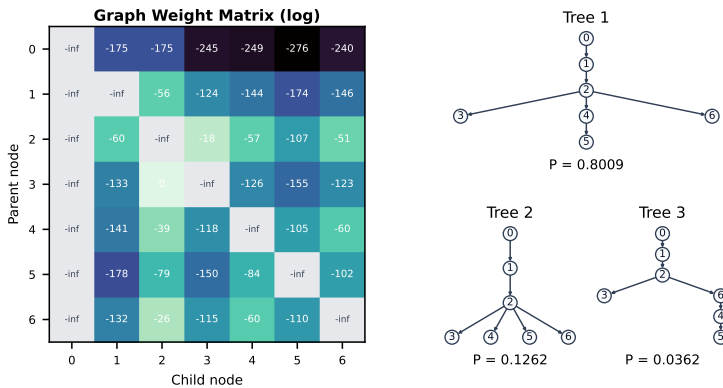


Figure 4.3.2: Example of a weight matrix representing the inferred variational distribution over trees with $K = 7$ clones after running VICTree on a DLP+ dataset. On the right side, the top-three most probable trees are shown, with their corresponding probabilities, which are computed by self-normalizing a sample of 10^4 trees. The matrix exhibits strongly skewed edge weights for which standard sampling algorithms would fail.

Contribution

We developed a novel, numerically stable algorithm for sampling spanning arborescences from graphs with highly skewed edge weights.

4.4 Thesis contributions

The second part of this thesis focuses on making joint Bayesian inference both scalable and mathematically robust, culminating in tools that fully integrate observation model,

latent biological states, and topological inference.

VICTree: Comprehensive joint Bayesian inference To perform joint inference of the clonal tree and the copy-number profiles, we developed VICTree [4] (Paper D), a novel variational inference framework that allows to approximate the posterior distribution over tree topologies and copy-number profiles simultaneously in an efficient manner. The main contribution of VICTree lies in a principled model of copy-number dependencies across the genome and across clones based on non-homogeneous HMMs, which are optimized jointly with the tree topology through efficient variational updates.

The variational framework allows VICTree to scale to datasets with thousands of cells, while still rigorously quantifying uncertainty over the tree topology and evolutionary parameters.

Stable generation of arborescences In its initial implementation, VICTree employs an importance sampling scheme to estimate expectations over tree space. To improve the quality of expectation estimates and the stability of the sampling procedure, we also developed Castaway [3] a novel algorithm for sampling spanning arborescences, which we presented in Paper C. The algorithm runs in polynomial-time and implements Wilson’s random walk, but shortcut via dynamic programming.

Discussion Although the impact of the new sampling algorithm has not been fully realized in a second iteration of VICTree, it has already been implemented as a standalone tool. We plan to present a comprehensive evaluation of the new sampling algorithm in a future publication, to show how it can improve the joint inference procedure.

5 Summary of the papers

The research presented in this thesis is built upon four main papers, each tackling specific computational and algorithmic challenges in tumor phylogenetics. This chapter provides an exhaustive summary of these contributions, detailing how they address the research gaps identified in Chapter 2 and how they fit together to form a cohesive narrative.

5.1 Paper A

Cellmates: a method for single-cell phylogeny reconstruction with proper evolutionary distances from copy numbers

Vittorio Zampinetti*, Harald Melin*, Marzie Abdolhamdi, Jens Lagergren

Manuscript, 2026 DOI: not yet available

Copy-number aberrations (CNAs) are a hallmark of somatic tumor evolution. Single-cell DNA sequencing technologies have enabled the profiling of CNAs at the resolution of individual cells, providing a powerful tool for reconstructing the evolutionary history of tumors. However, the noisy nature of single-cell read counts and the complex mutational processes underlying CNA accumulation pose significant challenges for accurate phylogeny reconstruction. Existing methods often rely on simplistic distance metrics like the Euclidean distance or Manhattan distance between raw read counts, which fail to account for the actual biological process of mutation accumulation and the presence of unobserved (latent) intermediate states. More advanced distance measures, such as the Minimum-Event Distance (MED), attempt to model the mutational process but are not corrected for evolutionary time and thus are not suitable for distance-based phylogeny reconstruction methods.

Paper A introduces *Cellmates*, a novel computational framework that combines a biologically informed model of CNA evolution with corrected distance estimation suitable

* Authors contributed equally.

for fast, distance-based phylogeny reconstruction. The core innovation is the use of a multi-layer Hidden Markov Model (HMM) that operates directly on phased single-cell read counts, without relying on multi-step preprocessing pipelines. By modeling the transition probabilities between latent allele-specific copy-number states over evolutionary time, Cellmates calculates statistically rigorous, additive evolutionary distances between pairs of cells. Furthermore, ad-hoc algorithmic optimizations are introduced to ensure that these distances can be computed efficiently even for large datasets, making them practical for real-world applications.

Through extensive simulations, the paper demonstrates that these distances can be used with classical and tailored distance-based algorithms to reconstruct phylogenies that are more accurate than those obtained with current state-of-the-art methods.

5.2 Paper B

Scalable and robust phylogenetic tree reconstruction from copy-number data with Sparse Rooted Neighbor Joining

Vittorio Zampinetti, Harald Melin, Alva Hallin, Jens Lagergren

Preprint, bioRxiv, 2026

DOI: 10.64898/2026.07.30.739152

Distance-based phylogenetic reconstruction from single-cell copy-number data is widely used in cancer genomics, yet most approaches focus primarily on designing pairwise distance measures while relying on classical Neighbor Joining (NJ) for tree inference. However, NJ reconstructs unrooted trees and ignores the known diploid ancestor that naturally defines the root in copy-number evolution. Paper B revisits the interaction between distance estimation and tree reconstruction and argues that root-aware algorithms are more appropriate in this setting. In particular, we identify Deepest Least Common Ancestor Neighbor Joining (DLCA–NJ) as a natural choice when root-to-LCA distances are available, such as those produced by probabilistic CNA models like SCONE2. By leveraging this additional information, DLCA–NJ improves reconstruction accuracy and root placement compared to standard NJ, especially when combined with model-corrected evolutionary distances.

To address scalability limitations of distance-based methods, with Paper B we introduce Sparse Rooted Neighbor Joining (SRNJ), a novel adaptation of Sparse–NJ to rooted tree reconstruction. SRNJ avoids computing the full pairwise distance matrix by constructing a small backbone tree and iteratively inserting leaves using centroid-based exploration and local DLCA decisions. This reduces the number of distance evaluations from $O(n^2)$ to $O(n \log n)$, yielding an overall complexity of $O(f(m)n \log n)$, where $f(m)$ denotes the cost of computing distances between sequences of length m . The paper provides theoretical guarantees showing that SRNJ is consistent and inherits the optimal edge ℓ_∞ -radius of DLCA–NJ, ensuring robustness to noise while trading only a minor loss in accuracy relative to the full algorithm.

Extensive experiments on synthetic data, realistic CNA simulations, and a breast cancer single-cell dataset demonstrate that DLCA–NJ consistently outperforms NJ across multiple tree comparison metrics, particularly in recovering correct root placement and nested clade structure. SRNJ achieves performance close to DLCA–NJ while

significantly reducing computational cost, making it suitable for larger datasets where full distance computation is prohibitive. The results also highlight the importance of using model-corrected distances, as improvements from rooted reconstruction are most pronounced when paired with probabilistic distance estimators such as *SCONCE2*. Overall, this work emphasizes the value of incorporating root information in CNA phylogenetics and introduces a scalable rooted reconstruction algorithm with strong theoretical and empirical performance.

5.3 Paper C

Sampling random spanning arborescences in graphs with low conductance

Vittorio Zampinetti, Harald Melin, Jens Lagergren

Statistics & Probability Letters, 2025

DOI: 10.1016/j.spl.2025.110481

Transitioning from point-estimate reconstructions (like those produced by SRNJ) to full Bayesian uncertainty quantification requires the ability to sample trees from probability distributions. A critical hurdle in Bayesian phylogenetics and related machine learning domains is sampling random spanning arborescences (rooted directed trees) proportional to their edge weights.

Classical approaches to this problem fall into two categories: determinant-based methods (via the Matrix-Tree Theorem) and random-walk methods (such as Wilson’s algorithm). Determinant-based methods scale poorly and suffer from numerical instability when dealing with ill-conditioned matrices, a common scenario in matrices derived from real-world data. Conversely, random walks can suffer from extremely long cover times on graphs with low conductance, which are frequently encountered in phylogenetic inference due to the presence of many low-probability edges.

Paper C resolves this fundamental algorithmic gap. It proposes *Castaway*, a novel, polynomial-time sampling algorithm inspired by Wilson’s random walk, but executed via dynamic programming. Instead of explicitly simulating walks, *Castaway* computes the exact transition and loop-erasure probabilities, effectively bypassing the cover-time limitations of standard random walks and the numerical instability of the Matrix-Tree Theorem. The paper demonstrates via simulation experiments that *Castaway* can efficiently sample from distributions over arborescences even in the presence of low-conductance bottlenecks, where standard off-the-shelf methods fail. This advancement enables rigorous Bayesian inference over tree topologies in any setting where edge weights can be defined, including tumor phylogenetics. In particular, this method is a critical component of the variational inference framework introduced in Paper D, allowing for efficient expectation computations over the space of clonal trees.

5.4 Paper D

VICTree - a Variational Inference method for Clonal Tree reconstruction

Harald Melin*, **Vittorio Zampinetti***, Andrew McPherson, Jens Lagergren

Research in Computational Molecular Biology (RECOMB), 2024

DOIs: 10.1101/2024.02.14.580312 (full version, bioRxiv; peer-reviewed for RECOMB), 10.1007/978-1-0716-3989-4_46 (short paper, proceedings)

While distance-based pipeline methods offer scalable tree inference, they often suffer from error propagation because distance estimation and tree reconstruction are performed in isolation. Joint Bayesian inference overcomes this by simultaneously modeling tree topology, branch lengths, and latent ancestral copy-number states, but traditional Markov Chain Monte Carlo (MCMC) methods scale poorly to the vast, complex space of tumor phylogenies, especially when accounting for the sequential dependencies of copy-number evolution. To address these limitations, Paper D introduces *VICTree* (Variational Inference for Clonal Trees). Building upon the biological modeling of copy-number evolution from Paper A and the tailored tree sampling algorithms developed in Paper C, *VICTree* provides a scalable variational inference framework for rigorous, joint Bayesian reconstruction of tumor evolution.

At its core, the model defines a highly structured generative framework where cells are assigned to a set of latent ancestral clones, which are related via a directed rooted tree (an arborescence). The observed single-cell read counts are modeled using emission distributions conditioned on these latent copy-number states. By leveraging the aforementioned tree sampling techniques to compute expectations over the space of all possible clonal trees, *VICTree* iteratively refines both the tree topology and the continuous parameters.

Notably, the Evidence Lower Bound (ELBO), which is the objective of the optimization process, can naturally be used for model selection, allowing for the principled comparison of trees with different sizes and topologies. This approach yields an accurate detection of clones and a robust reconstruction of the evolutionary history, performances that are validated through experiments on simulated data and real single-cell datasets. Ultimately, *VICTree* delivers a comprehensive, probabilistically sound view of tumor evolution that explicitly quantifies the uncertainty inherent in noisy single-cell data.

6 Conclusions

6.1 Summary of contributions

The thesis presents a series of methodological advances in the inference of tumor phylogenies from the analysis of single-cell copy-number data.

In this *kappa*, I attempted at building a coherent methodological narrative around a central question: how to infer tumor evolutionary history from noisy single-cell copy-number data while preserving biological realism, uncertainty quantification, and computational scalability. After introducing the problem and motivating the need for new methods in Chapter 1, Chapter 2 briefly reviews the most fundamental concepts from tumor phylogenetics and Bayesian inference that are relevant for the understanding of the subsequent chapters.

Chapter 3 frames and develops the distance-based line of work. It first places copy-number tree reconstruction in the context of the limitations of classical pipelines and then introduces methods that reduce error propagation and improve scalability in rooted settings. In this narrative, Paper A (Cellmates) provides probabilistic distance estimation directly from read counts, and Paper B (SparseRNJ) provides scalable rooted tree reconstruction algorithm compatible with computationally expensive but biologically grounded distances.

Chapter 4 then formalizes the transition to joint Bayesian inference. It positions tree topology as a latent variable coupled to clonal assignments and copy-number states, reviews why this coupling challenges standard MCMC-based approaches, and motivates the adoption of variational inference. Within this framework, Paper D (VICTree) contributes a joint variational model for clonal structure, copy-number evolution, and rooted topology, while Paper C (Castaway) contributes a stable algorithm for sampling spanning arborescence that addresses numerical instability and poor graph connectivity effects encountered with real data.

Taken together, the thesis contribution is twofold: it provides a conceptual bridge

between distance-based and fully Bayesian paradigms, and it contributes algorithmic components that make both paradigms more practical for real single-cell tumor phylogenetics.

6.2 Limitations and Future work

An immediate next step is a systematic evaluation of Castaway inside VICTree, to quantify gains in sampling stability and posterior quality on large real datasets. In parallel, Cellmates should be combined with SparseRNJ in large-scale experiments to measure end-to-end improvements in both runtime and reconstruction quality relative to classical distance-based pipelines.

A second direction is extending joint inference to allele-specific copy-number states. This requires richer observation and transition models that distinguish maternal and paternal copy-number evolution, while preserving computational tractability for large cohorts.

As a side note, evaluating Bayesian inference methods for tumor phylogenetics remains challenging. Unlike point-estimate tree reconstruction, there is no universally accepted benchmark for comparing posterior quality across different model families. Metrics such as ELBO are useful within a model class but are often not directly comparable across different probabilistic formulations. This makes fair cross-method evaluation difficult and highlights the need for community standards that combine simulation-based calibration, posterior predictive checks, and clinically relevant downstream criteria.

6.3 Concluding remarks

Copy-number evolution in cancer remains an active research area with substantial open questions. At the same time, sequencing technologies are rapidly evolving, and one design goal of the methods developed in this thesis is adaptability: the probabilistic formulations can be extended to new measurement models and new data modalities as they emerge.

Overall, this thesis aims to bridge theoretical advances in Bayesian inference, phylogenetics, and graph algorithms with practical cancer applications by developing methods that are both biologically grounded and computationally scalable on real-world data. Although motivated by tumor phylogenetics, the core ideas are broadly relevant to other inference problems in computational biology that involve latent structure, combinatorial state spaces, and noisy high-dimensional observations.

Bibliography

- [1] Vittorio Zampinetti, Harald Melin, Marzie Abdolhamdi, et al. “Cellmates: a method for single-cell phylogeny reconstruction with proper evolutionary distances from copy numbers”. Manuscript. 2026.
- [2] Vittorio Zampinetti, Harald Melin, Alva Hallin, et al. “Scalable and robust phylogenetic tree reconstruction from copy-number data with Sparse Rooted Neighbor Joining”. In: *bioRxiv* (2026). DOI: 10.64898/2026.07.30.739152. URL: <http://dx.doi.org/10.64898/2026.07.30.739152>.
- [3] Vittorio Zampinetti, Harald Melin, and Jens Lagergren. “Sampling random spanning arborescences in graphs with low conductance”. In: *Statistics & Probability Letters* 226 (Nov. 2025), p. 110481. ISSN: 0167-7152. DOI: 10.1016/j.spl.2025.110481. URL: <http://dx.doi.org/10.1016/j.spl.2025.110481>.
- [4] Harald Melin, Vittorio Zampinetti, Andrew McPherson, et al. “VICTree-A Variational Inference Method for Clonal Tree Reconstruction”. In: *International Conference on Research in Computational Molecular Biology*. Springer. 2024, pp. 429–433.
- [5] Hyuna Sung, Jacques Ferlay, Rebecca L. Siegel, et al. “Global Cancer Statistics 2020: GLOBOCAN Estimates of Incidence and Mortality Worldwide for 36 Cancers in 185 Countries”. In: *CA: A Cancer Journal for Clinicians* 71.3 (Feb. 2021), pp. 209–249. ISSN: 1542-4863. DOI: 10.3322/caac.21660. URL: <http://dx.doi.org/10.3322/caac.21660>.
- [6] Samuel Aparicio and Elaine Mardis. “Tumor heterogeneity: next-generation sequencing enhances the view from the pathologist’s microscope”. In: *Genome Biology* 15.9 (Oct. 2014). ISSN: 1474-760X. DOI: 10.1186/s13059-014-0463-6. URL: <http://dx.doi.org/10.1186/s13059-014-0463-6>.
- [7] Wafik S. El-Deiry, Barry Taylor, and Joel W. Neal. “Tumor Evolution, Heterogeneity, and Therapy for Our Patients With Advanced Cancer: How Far Have We Come?” In: *American Society of Clinical Oncology Educational Book* 37

- (May 2017), e8–e15. ISSN: 1548-8756. DOI: 10.1200/edbk_175524. URL: http://dx.doi.org/10.1200/EDBK_175524.
- [8] Nicholas McGranahan and Charles Swanton. “Clonal Heterogeneity and Tumor Evolution: Past, Present, and the Future”. In: *Cell* 168.4 (Feb. 2017), pp. 613–628. ISSN: 0092-8674. DOI: 10.1016/j.cell.2017.01.018. URL: <http://dx.doi.org/10.1016/j.cell.2017.01.018>.
 - [9] Nicholas Navin, Jude Kendall, Jennifer Troge, et al. “Tumour evolution inferred by single-cell sequencing”. en. In: *Nature* 472.7341 (Apr. 2011), pp. 90–94. ISSN: 0028-0836, 1476-4687. DOI: 10.1038/nature09807.
 - [10] P C Nowell. “The clonal evolution of tumor cell populations”. en. In: *Science* 194.4260 (Oct. 1976), pp. 23–28.
 - [11] Rameen Beroukhi, Craig H. Mermel, Dale Porter, et al. “The landscape of somatic copy-number alteration across human cancers”. In: *Nature* 463.7283 (Feb. 2010), pp. 899–905. ISSN: 1476-4687. DOI: 10.1038/nature08822. URL: <http://dx.doi.org/10.1038/nature08822>.
 - [12] Bernard Wood and Terry Harrison. “The evolutionary context of the first hominins”. In: *Nature* 470.7334 (Feb. 2011), pp. 347–352. ISSN: 1476-4687. DOI: 10.1038/nature09709. URL: <http://dx.doi.org/10.1038/nature09709>.
 - [13] Thomas H. Jukes and Charles R. Cantor. “Evolution of Protein Molecules”. In: *Mammalian Protein Metabolism*. Elsevier, 1969, pp. 21–132. ISBN: 9781483232119. DOI: 10.1016/b978-1-4832-3211-9.50009-7.
 - [14] Naruya Saitou and Masatoshi Nei. “The neighbor-joining method: a new method for reconstructing phylogenetic trees”. In: *Molecular biology and evolution* 4.4 (1987), pp. 406–425.
 - [15] Fred Griffith. “The Significance of Pneumococcal Types”. In: *Journal of Hygiene* 27.2 (Jan. 1928), pp. 113–159. ISSN: 0022-1724. DOI: 10.1017/s0022172400031879. URL: <http://dx.doi.org/10.1017/s0022172400031879>.
 - [16] A. D. Hershey and Martha Chase. “INDEPENDENT FUNCTIONS OF VIRAL PROTEIN AND NUCLEIC ACID IN GROWTH OF BACTERIOPHAGE”. In: *Journal of General Physiology* 36.1 (Sept. 1952), pp. 39–56. ISSN: 0022-1295. DOI: 10.1085/jgp.36.1.39. URL: <http://dx.doi.org/10.1085/jgp.36.1.39>.
 - [17] Eric S. Lander, Lauren M. Linton, Bruce Birren, et al. “Initial sequencing and analysis of the human genome”. In: *Nature* 409.6822 (Feb. 2001), pp. 860–921. ISSN: 1476-4687. DOI: 10.1038/35057062. URL: <http://dx.doi.org/10.1038/35057062>.
 - [18] David J. Gordon, Benjamin Resio, and David Pellman. “Causes and consequences of aneuploidy in cancer”. In: *Nature Reviews Genetics* 13.3 (Jan. 2012), pp. 189–203. ISSN: 1471-0064. DOI: 10.1038/nrg3123. URL: <http://dx.doi.org/10.1038/nrg3123>.
 - [19] Nicholas McGranahan, Rebecca A Burrell, David Endesfelder, et al. “Cancer chromosomal instability: therapeutic and diagnostic challenges: ‘Exploring aneuploidy: the significance of chromosomal imbalance’ review series”. In: *EMBO reports* 13.6 (May 2012), pp. 528–538. ISSN: 1469-3178. DOI: 10.1038/embor.2012.61. URL: <http://dx.doi.org/10.1038/embor.2012.61>.
 - [20] Sally M. Dewhurst, Nicholas McGranahan, Rebecca A. Burrell, et al. “Tolerance of Whole-Genome Doubling Propagates Chromosomal Instability and

- Accelerates Cancer Genome Evolution". In: *Cancer Discovery* 4.2 (Feb. 2014), pp. 175–185. ISSN: 2159-8290. DOI: 10.1158/2159-8290.cd-13-0285. URL: <http://dx.doi.org/10.1158/2159-8290.cd-13-0285>.
- [21] Alona Sosinsky, John Ambrose, William Cross, et al. "Insights for precision oncology from the integration of genomic and clinical data of 13,880 tumors from the 100,000 Genomes Cancer Programme". In: *Nature Medicine* 30.1 (Jan. 2024), pp. 279–289. ISSN: 1546-170X. DOI: 10.1038/s41591-023-02682-0. URL: <http://dx.doi.org/10.1038/s41591-023-02682-0>.
- [22] F. Sanger, S. Nicklen, and A. R. Coulson. "DNA sequencing with chain-terminating inhibitors". In: *Proceedings of the National Academy of Sciences* 74.12 (Dec. 1977), pp. 5463–5467. ISSN: 1091-6490. DOI: 10.1073/pnas.74.12.5463. URL: <http://dx.doi.org/10.1073/pnas.74.12.5463>.
- [23] Michael L. Metzker. "Sequencing technologies — the next generation". In: *Nature Reviews Genetics* 11.1 (Dec. 2009), pp. 31–46. ISSN: 1471-0064. DOI: 10.1038/nrg2626. URL: <http://dx.doi.org/10.1038/nrg2626>.
- [24] Zhong Wang, Mark Gerstein, and Michael Snyder. "RNA-Seq: a revolutionary tool for transcriptomics". In: *Nature reviews genetics* 10.1 (2009), pp. 57–63.
- [25] John N Weinstein, Eric A Collisson, Gordon B Mills, et al. "The Cancer Genome Atlas Pan-Cancer analysis project". In: *Nature Genetics* 45.10 (Sept. 2013), pp. 1113–1120. ISSN: 1546-1718. DOI: 10.1038/ng.2764. URL: <http://dx.doi.org/10.1038/ng.2764>.
- [26] Hans Zahn, Adi Steif, Emma Laks, et al. "Scalable whole-genome single-cell library preparation without preamplification". In: *Nature Methods* 14.2 (Jan. 2017), pp. 167–173. ISSN: 1548-7105. DOI: 10.1038/nmeth.4140. URL: <http://dx.doi.org/10.1038/nmeth.4140>.
- [27] Emma Laks, Andrew McPherson, Hans Zahn, et al. "Clonal Decomposition and DNA Replication States Defined by Scaled Single-Cell Genome Sequencing". In: *Cell* 179.5 (Nov. 2019), 1207–1221.e22. ISSN: 0092-8674. DOI: 10.1016/j.cell.2019.10.026. URL: <http://dx.doi.org/10.1016/j.cell.2019.10.026>.
- [28] Darlan C Minussi, Michael D Nicholson, Hui Ye, et al. "High-throughput single-cell DNA sequencing of acute myeloid leukemia". In: *Nature communications* 12.1 (2021), pp. 1–15.
- [29] Charles Gawad, Winston Koh, and Stephen R. Quake. "Single-cell genome sequencing: current state of the science". In: *Nature Reviews Genetics* 17.3 (Jan. 2016), pp. 175–188. ISSN: 1471-0064. DOI: 10.1038/nrg.2015.16. URL: <http://dx.doi.org/10.1038/nrg.2015.16>.
- [30] Christopher M Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.
- [31] Kevin P Murphy. *Machine Learning: A Probabilistic Perspective*. MIT press, 2012.
- [32] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. eng. Adaptive Computation and Machine Learning. Cambridge: MIT Press, 2009. ISBN: 9780262258357.
- [33] Zoubin Ghahramani. "An introduction to hidden Markov models and Bayesian networks". In: *International journal of pattern recognition and artificial intelligence* 15.01 (2001), pp. 9–42.

- [34] Ronald A Fisher. “On the mathematical foundations of theoretical statistics”. In: *Philosophical Transactions of the Royal Society of London. Series A , Containing Papers of a Mathematical or Physical Character* 222 (1922), pp. 309–368.
- [35] Andrew Gelman, John B Carlin, Hal S Stern, et al. *Bayesian Data Analysis*. CRC press, 2013.
- [36] Stuart Geman and Donald Geman. “Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-6.6 (Nov. 1984), pp. 721–741. ISSN: 0162-8828. DOI: 10.1109/tpami.1984.4767596. URL: <http://dx.doi.org/10.1109/TPAMI.1984.4767596>.
- [37] Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, et al. “Equation of state calculations by fast computing machines”. In: *The Journal of chemical physics* 21.6 (1953), pp. 1087–1092.
- [38] W Keith Hastings. “Monte Carlo sampling methods using Markov chains and their applications”. In: *Biometrika* 57.1 (1970), pp. 97–109.
- [39] Alp Kucukelbir, Rajesh Ranganath, Andrew Gelman, et al. “Automatic Variational Inference in Stan”. In: *Advances in Neural Information Processing Systems*. Ed. by C. Cortes, N. Lawrence, D. Lee, et al. Vol. 28. Curran Associates, Inc., 2015. URL: https://proceedings.neurips.cc/paper_files/paper/2015/file/352fe25daf686bdb4edca223c921acea-Paper.pdf.
- [40] Radford M. Neal. “Monte Carlo Implementation”. In: *Bayesian Learning for Neural Networks*. Springer New York, 1996, pp. 55–98. ISBN: 9781461207450. DOI: 10.1007/978-1-4612-0745-0_3. URL: http://dx.doi.org/10.1007/978-1-4612-0745-0_3.
- [41] Chris Whidden and Frederick A. Matsen. “Quantifying MCMC Exploration of Phylogenetic Tree Space”. In: *Systematic Biology* 64.3 (Jan. 2015), pp. 472–491. ISSN: 1063-5157. DOI: 10.1093/sysbio/syv006. URL: <http://dx.doi.org/10.1093/sysbio/syv006>.
- [42] Vu Dinh, Arman Bilge, Cheng Zhang, et al. “Probabilistic Path Hamiltonian Monte Carlo”. In: *Proceedings of the 34th International Conference on Machine Learning*. Ed. by Doina Precup and Yee Whye Teh. Vol. 70. Proceedings of Machine Learning Research. PMLR, June 2017, pp. 1009–1018. URL: <https://proceedings.mlr.press/v70/dinh17a.html>.
- [43] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. “Variational inference: A review for statisticians”. In: *Journal of the American statistical Association* 112.518 (2017), pp. 859–877.
- [44] Badr-Eddine Cherief-Abdellatif. “Consistency of ELBO maximization for model selection”. In: *Proceedings of The 1st Symposium on Advances in Approximate Bayesian Inference*. Ed. by Francisco Ruiz, Cheng Zhang, Dawen Liang, et al. Vol. 96. Proceedings of Machine Learning Research. PMLR, Feb. 2019, pp. 11–31. URL: <https://proceedings.mlr.press/v96/cherief-abdellatif19a.html>.
- [45] Joseph Felsenstein. *Inferring phylogenies*. Sinauer associates Sunderland, MA, 2004.
- [46] K. Atteson. “The Performance of Neighbor-Joining Methods of Phylogenetic Reconstruction”. In: *Algorithmica* 25.2–3 (June 1999), pp. 251–278. ISSN:

- 0178-4617. DOI: 10.1007/pl00008277. URL: <http://dx.doi.org/10.1007/PL00008277>.
- [47] D. Bryant. “Neighbor-Net: An Agglomerative Method for the Construction of Phylogenetic Networks”. In: *Molecular Biology and Evolution* 21.2 (Aug. 2003), pp. 255–265. ISSN: 1537-1719. DOI: 10.1093/molbev/msh018. URL: <http://dx.doi.org/10.1093/molbev/msh018>.
 - [48] Walter M Fitch. “Toward defining the course of evolution: minimum change for a specific tree topology”. In: *Systematic zoology* 20.4 (1971), pp. 406–416.
 - [49] Fredrik Ronquist and John P Huelsenbeck. “MrBayes 3: Bayesian phylogenetic inference under mixed models”. In: *Bioinformatics* 19.12 (2003), pp. 1572–1574.
 - [50] Pablo A. Goloboff and Martín E. Morales. “On the effect of measures for comparing trees on the representation of treespace”. In: *Cladistics* 41.4 (2025), pp. 398–415. DOI: <https://doi.org/10.1111/cla.12614>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1111/cla.12614>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1111/cla.12614>.
 - [51] David F Robinson and Leslie R Foulds. “Comparison of phylogenetic trees”. In: *Mathematical biosciences* 53.1-2 (1981), pp. 131–147.
 - [52] George F Estabrook, FR McMorris, and Christopher A Meacham. “Comparison of undirected phylogenetic trees based on subtrees of four evolutionary units”. In: *Systematic Zoology* 34.2 (1985), pp. 193–200.
 - [53] B. DasGupta, X. He, T. Jiang, et al. “On distances between phylogenetic trees”. In: *Proceedings of the Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*. SODA '97. New Orleans, Louisiana, USA: Society for Industrial and Applied Mathematics, 1997, pp. 427–436. ISBN: 0898713900.
 - [54] Benjamin L Allen and Mike Steel. “Subtree transfer operations and their induced metrics on evolutionary trees”. In: *Annals of combinatorics* 5.1 (2001), pp. 1–15.
 - [55] Yu Lin, V. Rajan, and B. M. E. Moret. “A Metric for Phylogenetic Trees Based on Matching”. In: *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 9.4 (July 2012), pp. 1014–1022. ISSN: 1545-5963. DOI: 10.1109/tcbb.2011.157. URL: <http://dx.doi.org/10.1109/TCBB.2011.157>.
 - [56] F. Lemoine, J.-B. Domelevo Entfellner, E. Wilkinson, et al. “Renewing Felsenstein’s phylogenetic bootstrap in the era of big data”. In: *Nature* 556.7702 (Apr. 2018), pp. 452–456. ISSN: 1476-4687. DOI: 10.1038/s41586-018-0043-0. URL: <http://dx.doi.org/10.1038/s41586-018-0043-0>.
 - [57] Miraine Dávila Felipe, Jean-Baka Domelevo Entfellner, Frédéric Lemoine, et al. “Distribution and asymptotic behavior of the phylogenetic transfer distance”. In: *Journal of Mathematical Biology* 79.2 (Apr. 2019), pp. 485–508. ISSN: 1432-1416. DOI: 10.1007/s00285-019-01365-0. URL: <http://dx.doi.org/10.1007/s00285-019-01365-0>.
 - [58] Alix Boc, Hervé Philippe, and Vladimir Makarenkov. “Inferring and Validating Horizontal Gene Transfer Events Using Bipartition Dissimilarity”. In: *Systematic Biology* 59.2 (Jan. 2010), pp. 195–211. ISSN: 1063-5157. DOI: 10.1093/sysbio/syp103. URL: <http://dx.doi.org/10.1093/sysbio/syp103>.
 - [59] Reinhard Diestel. *Graph theory*. Springer Publishing Company, 2017.

- [60] Gustav Kirchhoff. “Ueber die Auflösung der Gleichungen, auf welche man bei der Untersuchung der linearen Vertheilung galvanischer Ströme gefü hrt wird”. In: *Annalen der Physik* 148.12 (1847), pp. 497–508.
- [61] W. T. Tutte. “The dissection of equilateral triangles into equilateral triangles”. In: *Mathematical Proceedings of the Cambridge Philosophical Society* 44.4 (Oct. 1948), pp. 463–482. ISSN: 1469-8064. DOI: 10 . 1017 / s030500410002449x. URL: [http : / / dx . doi . org / 10 . 1017 / S030500410002449X](http://dx.doi.org/10.1017/S030500410002449X).
- [62] Patrick De Leenheer. “An elementary proof of a matrix tree theorem for directed graphs”. In: *SIAM Review* 62.3 (2020), pp. 716–726.
- [63] Charles J Colbourn, Wendy J Myrvold, and Eugene Neufeld. “Determinant-based algorithms for spanning tree probabilities”. In: *Algorithmica* 15 (1996), pp. 1–14.
- [64] David Aldous. “The random walk construction of uniform spanning trees and uniform routed trees”. In: *SIAM Journal on Discrete Mathematics* 1.1 (1990), pp. 4–11.
- [65] Andrei Z Broder. “Generating random spanning trees”. In: *Foundations of Computer Science, 1989., 30th Annual Symposium on. IEEE. 1989*, pp. 442–447.
- [66] David Bruce Wilson. “Generating random spanning trees more quickly than the cover time”. In: *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing. ACM. 1996*, pp. 296–303.
- [67] Jeff Cheeger. “A lower bound for the smallest eigenvalue of the Laplacian”. In: *Problems in Analysis*. Ed. by Robert C. Gunning. Princeton: Princeton University Press, 1970, pp. 195–200.
- [68] Lloyd N. Trefethen and David Bau III. *Numerical Linear Algebra*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 1997. DOI: 10 . 1137 / 1 . 9780898719574. eprint: [https : // epubs . siam . org / doi / pdf / 10 . 1137 / 1 . 9780898719574](https://epubs.siam.org/doi/pdf/10.1137/1.9780898719574). URL: [https : // epubs . siam . org / doi / abs / 10 . 1137 / 1 . 9780898719574](https://epubs.siam.org/doi/abs/10.1137/1.9780898719574).
- [69] Douglas Hanahan. “Hallmarks of Cancer: New Dimensions”. In: *Cancer Discovery* 12.1 (Jan. 2022), pp. 31–46. ISSN: 2159-8290. DOI: 10 . 1158/2159-8290 . cd-21-1059. URL: [http : // dx . doi . org / 10 . 1158 / 2159-8290 . CD-21-1059](http://dx.doi.org/10.1158/2159-8290.CD-21-1059).
- [70] Douglas Hanahan. “Hallmarks of cancer—Then and now, and beyond”. In: *Cell* (Jan. 2026). ISSN: 0092-8674. DOI: 10 . 1016 / j . cell . 2025 . 12 . 049. URL: [http : // dx . doi . org / 10 . 1016 / j . cell . 2025 . 12 . 049](http://dx.doi.org/10.1016/j.cell.2025.12.049).
- [71] Jacques S. Beckmann, Xavier Estivill, and Stylianos E. Antonarakis. “Copy number variants and genetic traits: closer to the resolution of phenotypic to genotypic variability”. In: *Nature Reviews Genetics* 8.8 (Aug. 2007), pp. 639–646. ISSN: 1471-0064. DOI: 10 . 1038/nrg2149. URL: [http : // dx . doi . org / 10 . 1038 / nrg2149](http://dx.doi.org/10.1038/nrg2149).
- [72] Giovanni Ciriello, Martin L Miller, Bülent Arman Aksoy, et al. “Emerging landscape of oncogenic signatures across human cancers”. In: *Nature Genetics* 45.10 (Sept. 2013), pp. 1127–1133. ISSN: 1546-1718. DOI: 10 . 1038/ng . 2762. URL: [http : // dx . doi . org / 10 . 1038 / ng . 2762](http://dx.doi.org/10.1038/ng.2762).
- [73] Jacob Househam, Timon Heide, George D. Cresswell, et al. “Phenotypic plasticity and genetic control in colorectal cancer evolution”. In: *Nature* 611.7937

- (Oct. 2022), pp. 744–753. ISSN: 1476-4687. DOI: 10.1038/s41586-022-05311-x. URL: <http://dx.doi.org/10.1038/s41586-022-05311-x>.
- [74] Priscilla K. Brastianos, Scott L. Carter, Sandro Santagata, et al. “Genomic Characterization of Brain Metastases Reveals Branched Evolution and Potential Therapeutic Targets”. In: *Cancer Discovery* 5.11 (Nov. 2015), pp. 1164–1177. ISSN: 2159-8290. DOI: 10.1158/2159-8290.cd-15-0369. URL: <http://dx.doi.org/10.1158/2159-8290.cd-15-0369>.
- [75] Andrew McPherson, Andrew Roth, Emma Laks, et al. “Divergent modes of clonal spread and intraperitoneal mixing in high-grade serous ovarian cancer”. In: *Nature Genetics* 48.7 (May 2016), pp. 758–767. ISSN: 1546-1718. DOI: 10.1038/ng.3573. URL: <http://dx.doi.org/10.1038/ng.3573>.
- [76] Salim Akhter Chowdhury, Stanley E. Shackney, Kerstin Heselmeyer-Haddad, et al. “Algorithms to Model Single Gene, Single Chromosome, and Whole Genome Copy Number Changes Jointly in Tumor Phylogenetics”. In: *PLoS Computational Biology* 10.7 (July 2014). Ed. by Sergei L. Kosakovsky. Pond, e1003740. ISSN: 1553-7358. DOI: 10.1371/journal.pcbi.1003740. URL: <http://dx.doi.org/10.1371/journal.pcbi.1003740>.
- [77] Yong Wang, Jill Waters, Marco L. Leung, et al. “Clonal evolution in breast cancer revealed by single nucleus genome sequencing”. In: *Nature* 512.7513 (July 2014), pp. 155–160. ISSN: 1476-4687. DOI: 10.1038/nature13600. URL: <http://dx.doi.org/10.1038/nature13600>.
- [78] Ruli Gao, Alexander Davis, Thomas O McDonald, et al. “Punctuated copy number evolution and clonal stasis in triple-negative breast cancer”. In: *Nature Genetics* 48.10 (Aug. 2016), pp. 1119–1130. ISSN: 1546-1718. DOI: 10.1038/ng.3641. URL: <http://dx.doi.org/10.1038/ng.3641>.
- [79] Luciano G Martelotto, Timour Baslan, Jude Kendall, et al. “Whole-genome single-cell copy number profiling from formalin-fixed paraffin-embedded samples”. In: *Nature Medicine* 23.3 (Feb. 2017), pp. 376–385. ISSN: 1546-170X. DOI: 10.1038/nm.4279. URL: <http://dx.doi.org/10.1038/nm.4279>.
- [80] Russell Schwartz and Alejandro A. Schäffer. “The evolution of tumour phylogenetics: principles and practice”. In: *Nature Reviews Genetics* 18.4 (Feb. 2017), pp. 213–229. ISSN: 1471-0064. DOI: 10.1038/nrg.2016.170. URL: <http://dx.doi.org/10.1038/nrg.2016.170>.
- [81] Darlan C. Minussi, Michael D. Nicholson, Hanghui Ye, et al. “Breast tumours maintain a reservoir of subclonal diversity during expansion”. In: *Nature* 592.7853 (Mar. 2021), pp. 302–308. ISSN: 1476-4687. DOI: 10.1038/s41586-021-03357-x. URL: <http://dx.doi.org/10.1038/s41586-021-03357-x>.
- [82] Roland F. Schwarz, Anne Trinh, Botond Sipos, et al. “Phylogenetic Quantification of Intra-tumour Heterogeneity”. In: *PLoS Computational Biology* 10.4 (Apr. 2014). Ed. by Niko Beerenwinkel, e1003535. ISSN: 1553-7358. DOI: 10.1371/journal.pcbi.1003535. URL: <http://dx.doi.org/10.1371/journal.pcbi.1003535>.
- [83] Garance Cordonnier and Manuel Lafond. “Comparing copy-number profiles under multi-copy amplifications and deletions”. In: *BMC Genomics* 21.S2 (Apr. 2020). ISSN: 1471-2164. DOI: 10.1186/s12864-020-6611-3. URL: <http://dx.doi.org/10.1186/s12864-020-6611-3>.
- [84] Fang Wang, Qihan Wang, Vakul Mohanty, et al. “MEDALT: single-cell copy number lineage tracing enabling gene discovery”. In: *Genome Biology* 22.1

- (Feb. 2021). ISSN: 1474-760X. DOI: 10.1186/s13059-021-02291-5. URL: <http://dx.doi.org/10.1186/s13059-021-02291-5>.
- [85] Jack Edmonds. “Optimum branchings”. In: *Journal of Research of the National Bureau of Standards Section B Mathematics and Mathematical Physics* 71B.4 (Oct. 1967), p. 233. ISSN: 0022-4340. DOI: 10.6028/jres.071b.032. URL: <http://dx.doi.org/10.6028/jres.071b.032>.
- [86] Henri Schmidt, Palash Sashittal, and Benjamin J. Raphael. “A zero-agnostic model for copy number evolution in cancer”. In: *PLOS Computational Biology* 19.11 (Nov. 2023). Ed. by Philipp M Altrock, e1011590. ISSN: 1553-7358. DOI: 10.1371/journal.pcbi.1011590. URL: <http://dx.doi.org/10.1371/journal.pcbi.1011590>.
- [87] Tom L. Kaufmann, Marina Petkovic, Thomas B. K. Watkins, et al. “MEDICC2: whole-genome doubling aware copy-number phylogenies for cancer evolution”. In: *Genome Biology* 23.1 (Nov. 2022). ISSN: 1474-760X. DOI: 10.1186/s13059-022-02794-9. URL: <http://dx.doi.org/10.1186/s13059-022-02794-9>.
- [88] Mehryar Mohri, Fernando Pereira, and Michael Riley. “Weighted finite-state transducers in speech recognition”. In: *Computer Speech & Language* 16.1 (Jan. 2002), pp. 69–88. ISSN: 0885-2308. DOI: 10.1006/csla.2001.0184. URL: <http://dx.doi.org/10.1006/csla.2001.0184>.
- [89] Timour Baslan, John P. Morris, Zhen Zhao, et al. “Ordered and deterministic cancer genome evolution after p53 loss”. en. In: *Nature* 608.7924 (Aug. 2022), pp. 795–802. ISSN: 0028-0836, 1476-4687. DOI: 10.1038/s41586-022-05082-5. URL: <https://www.nature.com/articles/s41586-022-05082-5> (visited on 11/17/2025).
- [90] Joana Reis Andrade, Annie Dinky Gallagher, Jovanna Maharaj, et al. “Disentangling the roles of aneuploidy, chromosomal instability and tumour heterogeneity in developing resistance to cancer therapies”. en. In: *Chromosome Research* 31.4 (Sept. 2023), p. 28. ISSN: 1573-6849. DOI: 10.1007/s10577-023-09737-5. URL: <https://link.springer.com/10.1007/s10577-023-09737-5> (visited on 11/17/2025).
- [91] Marc A. Vittoria, Ryan J. Quinton, and Neil J. Ganem. “Whole-genome doubling in tissues and tumors”. en. In: *Trends in Genetics* 39.12 (Dec. 2023), pp. 954–967. ISSN: 01689525. DOI: 10.1016/j.tig.2023.08.004. URL: <https://linkinghub.elsevier.com/retrieve/pii/S0168952523001877> (visited on 11/17/2025).
- [92] Andrew McPherson, Ignacio Vázquez-García, Matthew A. Myers, et al. “Ongoing genome doubling shapes evolvability and immunity in ovarian cancer”. en. In: *Nature* 644.8078 (Aug. 2025), pp. 1078–1087. ISSN: 0028-0836, 1476-4687. DOI: 10.1038/s41586-025-09240-3. URL: <https://www.nature.com/articles/s41586-025-09240-3> (visited on 11/17/2025).
- [93] Eric Letouzé, Yves Allory, Marc A Bollet, et al. “Analysis of the copy number profiles of several tumor samples from the same patient reveals the successive steps in tumorigenesis”. In: *Genome Biology* 11.7 (July 2010). ISSN: 1474-760X. DOI: 10.1186/gb-2010-11-7-r76. URL: <http://dx.doi.org/10.1186/gb-2010-11-7-r76>.
- [94] Sohrab Salehi, Fatemeh Dorri, Kevin Chern, et al. “Cancer phylogenetic tree inference at scale from 1000s of single cell genomes”. In: *Peer Community*

- Journal* 3 (July 2023). ISSN: 2804-3871. DOI: 10.24072/pcjournal.292. URL: <http://dx.doi.org/10.24072/pcjournal.292>.
- [95] Samson Weiner and Mukul S Bansal. “DICE: fast and accurate distance-based reconstruction of single-cell copy number phylogenies”. In: *Life Science Alliance* 8.3 (Dec. 2024), e202402923. ISSN: 2575-1077. DOI: 10.26508/lsa.202402923. URL: <http://dx.doi.org/10.26508/lsa.202402923>.
- [96] R. Desper. “Theoretical Foundation of the Balanced Minimum Evolution Method of Phylogenetic Inference and Its Relationship to Weighted Least-Squares Tree Fitting”. In: *Molecular Biology and Evolution* 21.3 (Dec. 2003), pp. 587–598. ISSN: 1537-1719. DOI: 10.1093/molbev/msh049. URL: <http://dx.doi.org/10.1093/molbev/msh049>.
- [97] Craig H Mermel, Steven E Schumacher, Barbara Hill, et al. “GISTIC2.0 facilitates sensitive and confident localization of the targets of focal somatic copy-number alteration in human cancers”. In: *Genome Biology* 12.4 (Apr. 2011). ISSN: 1474-760X. DOI: 10.1186/gb-2011-12-4-r41. URL: <http://dx.doi.org/10.1186/gb-2011-12-4-r41>.
- [98] Oscar Krijgsman, Beatriz Carvalho, Gerrit A. Meijer, et al. “Focal chromosomal copy number aberrations in cancer—Needles in a genome haystack”. In: *Biochimica et Biophysica Acta (BBA) - Molecular Cell Research* 1843.11 (Nov. 2014), pp. 2698–2704. ISSN: 0167-4889. DOI: 10.1016/j.bbamcr.2014.08.001. URL: <http://dx.doi.org/10.1016/j.bbamcr.2014.08.001>.
- [99] Tyler Garvin, Robert Aboukhalil, Jude Kendall, et al. “Interactive analysis and assessment of single-cell copy-number variations”. In: *Nature Methods* 12.11 (Sept. 2015), pp. 1058–1060. ISSN: 1548-7105. DOI: 10.1038/nmeth.3578. URL: <http://dx.doi.org/10.1038/nmeth.3578>.
- [100] Xuefeng Wang, Hao Chen, and Nancy R Zhang. “DNA copy number profiling using single-cell sequencing”. In: *Briefings in Bioinformatics* 19.5 (Feb. 2017), pp. 731–736. ISSN: 1477-4054. DOI: 10.1093/bib/bbx004. URL: <http://dx.doi.org/10.1093/bib/bbx004>.
- [101] Samson Weiner, Bingjun Li, and Sheida Nabavi. “Improved allele-specific single-cell copy number estimation in low-coverage DNA-sequencing”. en. In: *Bioinformatics* 40.8 (Aug. 2024). Ed. by Anthony Mathelier, btae506. ISSN: 1367-4803, 1367-4811. DOI: 10.1093/bioinformatics/btae506. URL: <https://academic.oup.com/bioinformatics/article/doi/10.1093/bioinformatics/btae506/7731720> (visited on 11/17/2025).
- [102] A. B. Olshen, E. S. Venkatraman, R. Lucito, et al. “Circular binary segmentation for the analysis of array-based DNA copy number data”. In: *Biostatistics* 5.4 (Oct. 2004), pp. 557–572. ISSN: 1468-4357. DOI: 10.1093/biostatistics/kxh008. URL: <http://dx.doi.org/10.1093/biostatistics/kxh008>.
- [103] Gro Nilsen, Knut Liestøl, Peter Van Loo, et al. “Copynumber: Efficient algorithms for single- and multi-track copy number segmentation”. In: *BMC Genomics* 13.1 (Nov. 2012). ISSN: 1471-2164. DOI: 10.1186/1471-2164-13-591. URL: <http://dx.doi.org/10.1186/1471-2164-13-591>.
- [104] Jack Kuipers, Mustafa Anil Tuncel, Pedro F Ferreira, et al. “Single-cell copy number calling and event history reconstruction”. In: *Bioinformatics* 41.3 (Feb. 2025). Ed. by Russell Schwartz. ISSN: 1367-4811. DOI: 10.1093/bioinformatics/btaf072. URL: <http://dx.doi.org/10.1093/bioinformatics/btaf072>.

BIBLIOGRAPHY

- [105] A. Viterbi. “Error bounds for convolutional codes and an asymptotically optimum decoding algorithm”. In: *IEEE Transactions on Information Theory* 13.2 (Apr. 1967), pp. 260–269. ISSN: 1557-9654. DOI: 10.1109/tit.1967.1054010. URL: <http://dx.doi.org/10.1109/TIT.1967.1054010>.
- [106] Sohrab P. Shah, Xiang Xuan, Ron J. DeLeeuw, et al. “Integrating copy number polymorphisms into array CGH analysis using a robust HMM”. In: *Bioinformatics* 22.14 (July 2006), e431–e439. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btl238. URL: <http://dx.doi.org/10.1093/bioinformatics/btl238>.
- [107] Bjorn Bakker, Aaron Taudt, Mirjam E. Belderbos, et al. “Single-cell sequencing reveals karyotype heterogeneity in murine and human malignancies”. In: *Genome Biology* 17.1 (May 2016). ISSN: 1474-760X. DOI: 10.1186/s13059-016-0971-7. URL: <http://dx.doi.org/10.1186/s13059-016-0971-7>.
- [108] Xian F. Mallory, Mohammadamin Edrisi, Nicholas Navin, et al. “Methods for copy number aberration detection from single-cell DNA-sequencing data”. In: *Genome Biology* 21.1 (Aug. 2020). ISSN: 1474-760X. DOI: 10.1186/s13059-020-02119-8. URL: <http://dx.doi.org/10.1186/s13059-020-02119-8>.
- [109] Xian F. Mallory, Mohammadamin Edrisi, Nicholas Navin, et al. “Assessing the performance of methods for copy number aberration detection from single-cell DNA sequencing data”. In: *PLOS Computational Biology* 16.7 (July 2020). Ed. by Ilya Ioshikhes, e1008012. ISSN: 1553-7358. DOI: 10.1371/journal.pcbi.1008012. URL: <http://dx.doi.org/10.1371/journal.pcbi.1008012>.
- [110] Sandra Hui and Rasmus Nielsen. “SCONCE2: jointly inferring single cell copy number profiles and tumor evolutionary distances”. In: *BMC Bioinformatics* 23.1 (Aug. 2022). ISSN: 1471-2105. DOI: 10.1186/s12859-022-04890-w. URL: <http://dx.doi.org/10.1186/s12859-022-04890-w>.
- [111] Sandra Hui and Rasmus Nielsen. “SCONCE: a method for profiling copy number alterations in cancer evolution using single-cell whole genome sequencing”. In: *Bioinformatics* 38.7 (Jan. 2022). Ed. by Can Alkan, pp. 1801–1808. ISSN: 1367-4811. DOI: 10.1093/bioinformatics/btac041. URL: <http://dx.doi.org/10.1093/bioinformatics/btac041>.
- [112] Paul Jorgensen, Nicholas P. Edgington, Brandt L. Schneider, et al. “The Size of the Nucleus Increases as Yeast Cells Grow”. In: *Molecular Biology of the Cell* 18.9 (Sept. 2007). Ed. by Orna Cohen-Fix, pp. 3523–3532. ISSN: 1939-4586. DOI: 10.1091/mbc.e06-10-0973. URL: <http://dx.doi.org/10.1091/mbc.e06-10-0973>.
- [113] Robin J. Hofmeister, Diogo M. Ribeiro, Simone Rubinacci, et al. “Accurate rare variant phasing of whole-genome and whole-exome sequencing data in the UK Biobank”. In: *Nature Genetics* 55.7 (June 2023), pp. 1243–1249. ISSN: 1546-1718. DOI: 10.1038/s41588-023-01415-w. URL: <http://dx.doi.org/10.1038/s41588-023-01415-w>.
- [114] Po-Ru Loh, Petr Danecek, Pier Francesco Palamara, et al. “Reference-based phasing using the Haplotype Reference Consortium panel”. In: *Nature Genetics* 48.11 (Oct. 2016), pp. 1443–1448. ISSN: 1546-1718. DOI: 10.1038/ng.3679. URL: <http://dx.doi.org/10.1038/ng.3679>.
- [115] Simone Zaccaria and Benjamin J Raphael. “Characterizing allele- and haplotype-specific copy numbers in single cells with CHISEL”. en. In: *Nat. Biotechnol.* 39.2 (Feb. 2021), pp. 207–214.

- [116] Tyler Funnell, Ciara H. O’Flanagan, Marc J. Williams, et al. “Single-cell genomic variation induced by mutational processes in cancer”. In: *Nature* 612.7938 (Oct. 2022), pp. 106–115. ISSN: 1476-4687. DOI: 10.1038/s41586-022-05249-0. URL: <http://dx.doi.org/10.1038/s41586-022-05249-0>.
- [117] Hongyu Shi, Marc J. Williams, Gryte Satas, et al. “Allele-specific transcriptional effects of subclonal copy number alterations enable genotype-phenotype mapping in cancer cells”. In: *Nature Communications* 15.1 (Mar. 2024). ISSN: 2041-1723. DOI: 10.1038/s41467-024-46710-0. URL: <http://dx.doi.org/10.1038/s41467-024-46710-0>.
- [118] Stefan Ivanovic and Mohammed El-Kebir. “CNRein: an evolution-aware deep reinforcement learning algorithm for single-cell DNA copy number calling”. In: *Genome Biology* 26.1 (Apr. 2025). ISSN: 1474-760X. DOI: 10.1186/s13059-025-03553-2. URL: <http://dx.doi.org/10.1186/s13059-025-03553-2>.
- [119] Ricardo J. G. B. Campello, Davoud Moulavi, and Joerg Sander. “Density-Based Clustering Based on Hierarchical Density Estimates”. In: *Advances in Knowledge Discovery and Data Mining*. Springer Berlin Heidelberg, 2013, pp. 160–172. ISBN: 9783642374562. DOI: 10.1007/978-3-642-37456-2_14. URL: http://dx.doi.org/10.1007/978-3-642-37456-2_14.
- [120] Ilan Gronau and Shlomo Moran. “Neighbor joining algorithms for inferring phylogenies via LCA distances”. en. In: *J. Comput. Biol.* 14 (1 Jan. 2007), pp. 1–15. DOI: 10.1089/cmb.2006.0115.
- [121] Cyril Allauzen, Michael Riley, Johan Schalkwyk, et al. “OpenFst: A General and Efficient Weighted Finite-State Transducer Library”. In: *Implementation and Application of Automata*. Springer Berlin Heidelberg, pp. 11–23. ISBN: 9783540763352. DOI: 10.1007/978-3-540-76336-9_3. URL: http://dx.doi.org/10.1007/978-3-540-76336-9_3.
- [122] Semih Kurt, Alexandre Bouchard-Côté, and Jens Lagergren. “Sparse Neighbor Joining: rapid phylogenetic inference using a sparse distance matrix”. en. In: *Bioinformatics* 40 (12 2024). DOI: 10.1093/bioinformatics/btae701.
- [123] Marc J. Williams, Andrea Sottoriva, and Trevor A. Graham. “Measuring Clonal Evolution in Cancer with Genomics”. In: *Annual Review of Genomics and Human Genetics* 20.1 (Aug. 2019), pp. 309–329. ISSN: 1545-293X. DOI: 10.1146/annurev-genom-083117-021712. URL: <http://dx.doi.org/10.1146/annurev-genom-083117-021712>.
- [124] John P. Huelsenbeck, Bruce Rannala, and John P. Masly. “Accommodating Phylogenetic Uncertainty in Evolutionary Studies”. In: *Science* 288.5475 (June 2000), pp. 2349–2350. ISSN: 1095-9203. DOI: 10.1126/science.288.5475.2349. URL: <http://dx.doi.org/10.1126/science.288.5475.2349>.
- [125] Gabriel W. Hassler, Andrew F. Magee, Zhenyu Zhang, et al. “Data Integration in Bayesian Phylogenetics”. In: *Annual Review of Statistics and Its Application* 10.1 (Mar. 2023), pp. 353–377. ISSN: 2326-831X. DOI: 10.1146/annurev-statistics-033021-112532. URL: <http://dx.doi.org/10.1146/annurev-statistics-033021-112532>.
- [126] Seong-Hwan Jun, Hosein Toosi, Jeff Mold, et al. “Reconstructing clonal tree for phylo-phenotypic characterization of cancer using single-cell transcriptomics”. In: *Nature Communications* 14.1 (Feb. 2023). ISSN: 2041-1723. DOI: 10.1038/s41467-023-36202-y. URL: <http://dx.doi.org/10.1038/s41467-023-36202-y>.

BIBLIOGRAPHY

- [127] Amit G Deshwar, Shankar Vembu, Christina K Yung, et al. “PhyloWGS: Reconstructing subclonal composition and evolution from whole-genome sequencing of tumors”. In: *Genome Biology* 16.1 (Feb. 2015). ISSN: 1465-6906. DOI: 10.1186/s13059-015-0602-8. URL: <http://dx.doi.org/10.1186/s13059-015-0602-8>.
- [128] Andrew Roth, Jaswinder Khattra, Damian Yap, et al. “PyClone: statistical inference of clonal population structure in cancer”. In: *Nature Methods* 11.4 (Mar. 2014), pp. 396–398. ISSN: 1548-7105. DOI: 10.1038/nmeth.2883. URL: <http://dx.doi.org/10.1038/nmeth.2883>.
- [129] Ziwei Chen, Fuzhou Gong, Lin Wan, et al. “BiTSC2: Bayesian inference of tumor clonal tree by joint analysis of single-cell SNV and CNA data”. In: *Briefings in Bioinformatics* 23.3 (Apr. 2022). ISSN: 1477-4054. DOI: 10.1093/bib/bbac092. URL: <http://dx.doi.org/10.1093/bib/bbac092>.
- [130] Gryte Satas, Simone Zaccaria, Geoffrey Mon, et al. “SCARLET: Single-Cell Tumor Phylogeny Inference with Copy-Number Constrained Mutation Losses”. In: *Cell Systems* 10.4 (Apr. 2020), 323–332.e8. ISSN: 2405-4712. DOI: 10.1016/j.cels.2020.04.001. URL: <http://dx.doi.org/10.1016/j.cels.2020.04.001>.
- [131] Magda Markowska, Tomasz Cakała, Błażej Miasojedow, et al. “CONET: copy number event tree model of evolutionary tumor history for single-cell data”. In: *Genome Biology* 23.1 (June 2022). ISSN: 1474-760X. DOI: 10.1186/s13059-022-02693-z. URL: <http://dx.doi.org/10.1186/s13059-022-02693-z>.
- [132] Zoubin Ghahramani, Michael Jordan, and Ryan P Adams. “Tree-Structured Stick Breaking for Hierarchical Data”. In: *Advances in Neural Information Processing Systems*. Ed. by J. Lafferty, C. Williams, J. Shawe-Taylor, et al. Vol. 23. Curran Associates, Inc., 2010. URL: https://proceedings.neurips.cc/paper_files/paper/2010/file/a5e00132373a7031000fd987a3c9f87b-Paper.pdf.
- [133] Katharina Jahn, Jack Kuipers, and Niko Beerenwinkel. “Tree inference for single-cell data”. In: *Genome Biology* 17.1 (May 2016). ISSN: 1474-760X. DOI: 10.1186/s13059-016-0936-x. URL: <http://dx.doi.org/10.1186/s13059-016-0936-x>.
- [134] Hamim Zafar, Anthony Tzen, Nicholas Navin, et al. “SiFit: inferring tumor trees from single-cell sequencing data under finite-sites models”. In: *Genome Biology* 18.1 (Sept. 2017). ISSN: 1474-760X. DOI: 10.1186/s13059-017-1311-2. URL: <http://dx.doi.org/10.1186/s13059-017-1311-2>.
- [135] Sierra Gillis and Andrew Roth. “PyClone-VI: scalable inference of clonal population structures using whole genome data”. In: *BMC Bioinformatics* 21.1 (Dec. 2020). ISSN: 1471-2105. DOI: 10.1186/s12859-020-03919-2. URL: <http://dx.doi.org/10.1186/s12859-020-03919-2>.
- [136] Alexandre Bouchard-Côté, Sriram Sankararaman, and Michael I. Jordan. “Phylogenetic Inference via Sequential Monte Carlo”. In: *Systematic Biology* 61.4 (Jan. 2012), pp. 579–593. ISSN: 1063-5157. DOI: 10.1093/sysbio/syr131. URL: <http://dx.doi.org/10.1093/sysbio/syr131>.
- [137] Liangliang Wang, Alexandre Bouchard-Côté, and Arnaud Doucet. “Bayesian Phylogenetic Inference Using a Combinatorial Sequential Monte Carlo Method”. In: *Journal of the American Statistical Association* 110.512 (Oct.

- 2015), pp. 1362–1374. ISSN: 1537-274X. DOI: 10.1080/01621459.2015.1054487. URL: <http://dx.doi.org/10.1080/01621459.2015.1054487>.
- [138] Antonio Khalil Moretti, Liyi Zhang, Christian A Naesseth, et al. “Variational combinatorial sequential Monte Carlo methods for Bayesian phylogenetic inference”. In: *Uncertainty in Artificial Intelligence*. PMLR. 2021, pp. 971–981.
- [139] Cheng Zhang. “Improved Variational Bayesian Phylogenetic Inference with Normalizing Flows”. In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, et al. Vol. 33. Curran Associates, Inc., 2020, pp. 18760–18771. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/d96409bf894217686ba124d7356686c9-Paper.pdf.
- [140] Cheng Zhang and Frederick A. Matsen IV. “A Variational Approach to Bayesian Phylogenetic Inference”. In: *Journal of Machine Learning Research* 25.145 (2024), pp. 1–56. URL: <http://jmlr.org/papers/v25/22-0348.html>.
- [141] Cheng Zhang and Frederick A Matsen IV. “Generalizing Tree Probability Estimation via Bayesian Networks”. In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio, H. Wallach, H. Larochelle, et al. Vol. 31. Curran Associates, Inc., 2018. URL: https://proceedings.neurips.cc/paper_files/paper/2018/file/b137fdd1f79d56c7edf3365fea7520f2-Paper.pdf.
- [142] Hazal Koptagel, Oskar Kviman, Harald Melin, et al. “VaiPhy: a Variational Inference Based Algorithm for Phylogeny”. In: *Advances in Neural Information Processing Systems*. Ed. by S. Koyejo, S. Mohamed, A. Agarwal, et al. Vol. 35. Curran Associates, Inc., 2022, pp. 14758–14770. URL: https://proceedings.neurips.cc/paper_files/paper/2022/file/5e956fef0946dc1e39760f94b78045fe-Paper-Conference.pdf.
- [143] Persi Diaconis. “Sequential importance sampling for estimating the number of perfect matchings in bipartite graphs: An ongoing conversation with Laci”. In: *Building Bridges II: Mathematics of László Lovász*. Springer, 2020, pp. 223–233.
- [144] Sanae Lotfi, Pavel Izmailov, Gregory Benton, et al. “Bayesian model selection, the marginal likelihood, and generalization”. In: *International Conference on Machine Learning*. PMLR. 2022, pp. 14223–14247.
- [145] 10x Genomics. *PBMCs from C57BL/6 mice (v1, 150x150), Single Cell Immune Profiling Dataset by Cell Ranger v3.1.0*. 10x Genomics dataset. Dataset citation format provided by 10x Genomics. July 2019.
- [146] Mathieu Fourment, Andrew F Magee, Chris Whidden, et al. “19 Dubious Ways to Compute the Marginal Likelihood of a Phylogenetic Tree Topology”. In: *Systematic Biology* 69.2 (Aug. 2019). Ed. by David Posada, pp. 209–220. ISSN: 1076-836X. DOI: 10.1093/sysbio/syz046. URL: <http://dx.doi.org/10.1093/sysbio/syz046>.

Appended papers

Paper A

Paper A

Cellmates: a method for single-cell phylogeny reconstruction with proper evolutionary distances from copy numbers

VITTORIO ZAMPINETTI, HARALD MELIN, MARZIE ABDOLHAMDI, AND JENS LAGERGREN

Manuscript

Cellmates: a method for single-cell phylogeny reconstruction with proper evolutionary distances from copy numbers

Vittorio Zampinetti^{1,2}, Harald Melin², Marzie Abdolhamdi², and Jens Lagergren²

¹ Politecnico di Torino, Italy

² KTH Royal Institute of Technology, Stockholm

Abstract. Copy number alterations (CNAs) provide key information on tumor evolution and are now accessible at single-cell resolution through low-coverage DNA sequencing. Reconstructing accurate phylogenies from scDNA data is challenged by the complexity of copy number evolution and the inherent noise in single-cell measurements. Several methods have been developed for this purpose, but unlike classical sequence-based phylogenetics, they do not apply probabilistic distance corrections that account for finite-length observations. Such corrections, as in the Jukes-Cantor model, are essential for theoretically consistent tree inference. We present Cellmates, a statistical framework for reconstructing copy number phylogenies from single-cell read counts using probabilistically corrected, additive distances. Such distances are estimated through an efficient expectation-maximization procedure, implemented with the use of a tailored Hidden Markov Model, and then combined with a neighbor-joining variant that explicitly accounts for a diploid root node. Results on simulated and real tumor datasets show that Cellmates reconstructs accurate phylogenies and outperforms other methods while remaining computationally efficient. As a consequence, Cellmates provides a statistically grounded framework for copy number evolution from single-cell data.

Keywords: Copy numbers · maximum likelihood · tumor phylogeny

1 Introduction

Cancer is a complex and evolving disease, characterized by the accumulation of somatic genetic alterations that drive tumor progression and result in intra-tumor heterogeneity (ITH) [15,19]. Among the various genetic changes observed in tumors, copy number alterations (CNAs) and chromosomal instabilities (CIN), such as amplifications and deletions, are hallmarks of cancer [18], associated with poor clinical outcome and drug resistance [15,24]. These structural alterations not only have critical clinical implications but also serve as key markers for reconstructing the evolutionary pathways of cancer [9].

The advent of advanced single-cell (sc) DNA sequencing technologies, such as Direct Library Preparation (DLP) [12,31], has enabled whole-genome sequencing (WGS) of thousands of individual cells, but with low coverage (0.01-0.1%). Low coverage results in noise in the data, and CN events induce dependencies between genomic sites, making CN inference and phylogenetic tree reconstruction a difficult task [9].

To address these challenges, several computational methods have been developed for the task of reconstructing tumor phylogenies from scDNA data [9, 11, 14, 17, 22, 23, 26, 30]. To overcome the intractability arising from the CN dependencies and to allow scaling to many cells, the methods implement different procedures: reduce the state-space of the CNs across the genome [14, 22], cluster cells and infer clonal trees [11, 17, 30] and compute pairwise distances between cells before reconstructing the tree [7, 9, 23, 29].

This paper is concerned with the latter category, which has been proven successful; MEDICC2 [9] is a state-of-the-art method for reconstructing phylogenetic trees from CN data, particularly in the context of whole-genome duplication (WGD). It has already been applied in several large-scale and high-impact studies [1,2,16,25]. However, the method relies on pre-estimated CN profiles (CNP), i.e. the CN state across the genome, without the context of the phylogenetic tree, thereby preventing the propagation of statistical uncertainty from the read data to the tree inference, and possibly removing information on focal (small) CNAs. Furthermore, the minimal event distance (MED) of MEDICC2 does not explicitly account for homoplasy, a common precaution applied in molecular phylogenetics, e.g. in the Jukes-Cantor (JC) [8] and Kimura [10] models, which statistically adjust for multiple mutations in the same genomic site over branches in the tree. SCONCE [7] introduced this probabilistic distance correction for CN evolution. Instead of modeling CN events directly, SCONCE models breakpoints, i.e. shifts in the CNPs induced by CN events, by a Markov process over branches, and uses such model to estimate the cell-specific rates of CN change. The successor, SCONCE2 [7], then further defines a triplet structure which relates pairs of cells by a latent median node, representing the lowest common ancestor (LCA) of the pair, together with a root node representing a diploid cell. The

evolution rates estimated by SCONE are thereby combined and used to compute pairwise distances between cells as the sum of the cell-to-median distances, which are then used for tree reconstruction.

For tree reconstruction, all of these distance-based methods rely on the Neighbor Joining algorithm [21], which does not take into account the root-to-median distance, and require probabilistic correction to get a biologically correct tree.

To address these limitations, we introduce Cellmates, a framework for reconstructing single-cell trees, by modeling copy number evolution over phylogenetic trees using a Hidden Markov Model (HMM) framework. Similar to the triplet structure introduced by SCONE2 [7], we construct a triplet-based HMM using the Jukes-Cantor model for breakpoints, which we call the Jukes-Cantor Breakpoint (JCB) model. This enables an Expectation-Maximization (EM) algorithm for maximum-likelihood (ML) estimation of the branch lengths of the triplet. Equipped with the root-to-median distances of the triplet, we then reconstruct the tree with an agglomeration procedure that merges the pair of units sharing the deepest last common ancestor, which we refer to as Root-preserving Neighbor Joining (RP-NJ). We developed RP-NJ independently, but it coincides with the Deepest Least Common Ancestor Neighbor Joining (DLCA-NJ) algorithm of Gronau and Moran [6]: RP-NJ is our name for the same algorithm, and we claim no novelty for it. What is new here is the setting in which it is used. DLCA-NJ requires distances from a fixed root to the last common ancestor of every pair of leaves, quantities that are not available from copy number profiles by any established means; our EM procedure supplies exactly these quantities, and to our knowledge this is the first time the algorithm has been applied to copy number evolution or, more generally, to tumor phylogenetics.

An overview of the framework is shown in Figure 1. We demonstrate the benefit of our RP-NJ method over standard NJ on simulated datasets in Section 3.1. In Section 3.2, to compare the performance of our method with MEDICC2 and DICE, we extend the benchmark datasets of DICE [29], generated with the CN simulation tool CNAsim [28]. Finally, we analyze two ovarian cancer datasets using our framework and briefly assess the biological relevance of the inferred trees.

In summary, our contributions are:

- **Root-preserving Neighbor Joining (RP-NJ)**: the adoption of DLCA-NJ [6], an existing rooted distance-based reconstruction algorithm never previously applied in tumor evolution, whose requirements are met exactly by the triplet distances our EM procedure estimates;
- **the Jukes-Cantor breakpoint (JCB) model**: a Jukes-Cantor model for copy number evolution based on breakpoints;
- **Cellmates**: a framework for tumor phylogeny reconstruction from scDNA-seq data combining the EM algorithm for inferring JCB maximum-

likelihood distances over a triplet with noisy-copy numbers or reads as input and the RP-NJ algorithm.

The code for reproducing the experiments is available at github.com/Lagergren-Lab/cellmates.

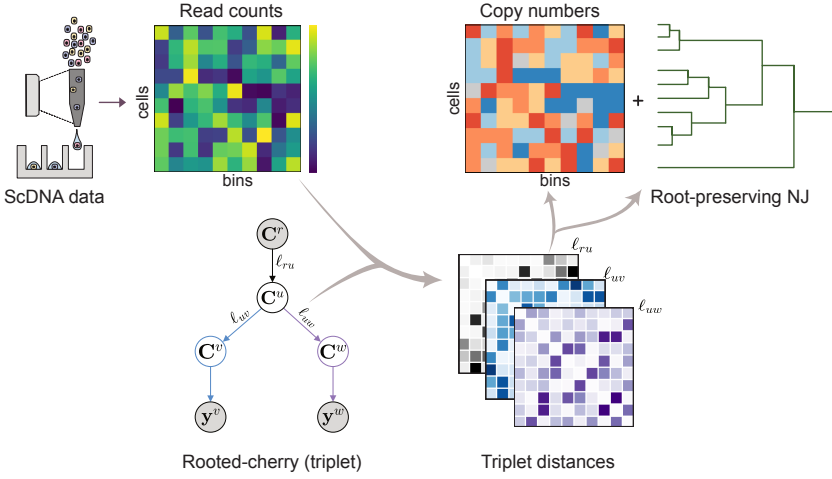


Fig. 1: Overview of the Cellmates framework.

2 Methods

2.1 The Model

We model copy number evolution along a rooted tree. If the copy number state at every vertex were known, and the branch lengths were fixed, the likelihood of the full tree is obtained by multiplying, over all edges and genomic bins, the parent-child transition probabilities. These transitions depend only on the copy number of the parent and the length of the connecting edge, making the likelihood fully factorized and easy to interpret.

For each parent-child pair, the model defines a conditional distribution over the child's copy number sequence. Across the genome this distribution satisfies the Markov property, and along the tree it propagates from the root to the leaves. Placing this rule on every edge gives a generative model; starting from a diploid root, one can generate the copy number sequences of all descendants exactly according to the same probabilities that appear in the likelihood. This generative view corresponds directly to the decomposition of the tree likelihood.

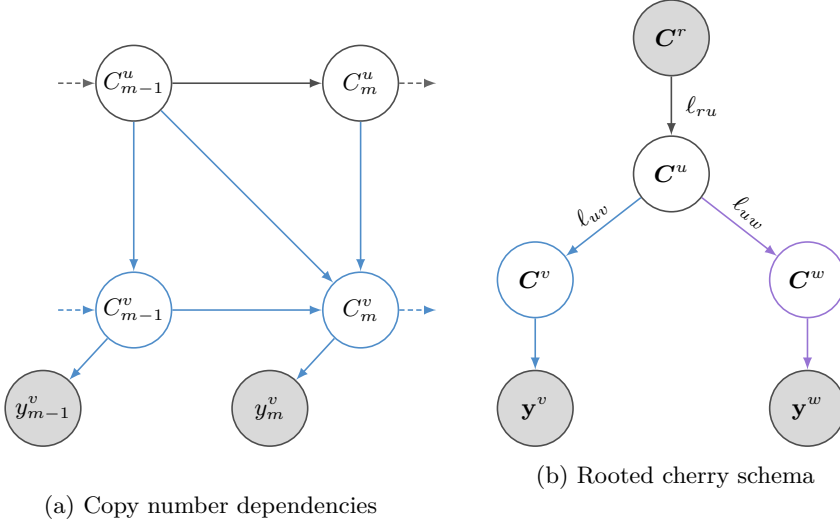


Fig. 2: Model and optimization overview. (a) CopyTree dependencies across bins. (b) Rooted cherry structure with branch lengths ℓ_{ru} , ℓ_{uv} , and ℓ_{uw} .

Because the evolutionary process is Markov on the tree, any local subtree inherits its distribution from the full model. In particular, a rooted cherry, consisting of the root, an internal node u (the least common ancestor of two leaves), and the three connecting branches, has a well-defined marginal distribution over the copy number sequences of (u, v, w) . This marginal depends only on the three branch lengths $(\ell_{ru}, \ell_{uv}, \ell_{uw})$ and is unaffected by the rest of the tree. Cellmates uses this property to estimate local evolutionary distances from pairs of cells. The observational model enters only at the leaves; given a latent copy number at each bin, the read counts are conditionally independent. This layer modifies the likelihood but does not alter the underlying Markov structure of the copy number evolution. These observations motivate Cellmates approach; we treat each rooted cherry as a self-contained probabilistic unit, fit its parameters using a triplet HMM, and combine these local estimates during tree reconstruction.

We now focus on a single rooted cherry $T^\#$, a four-node rooted tree. We consider a larger rooted binary tree $T \in \mathcal{T}$, where $\mathcal{T}$ is the space of all binary trees that have n leaves. The rooted cherry consists of a root node r , its descendant u , which is the least common ancestor (LCA) of the two leaves v and w descending from that and three directed edges $E(T^\#) = \{(r, u), (u, v), (u, w)\}$, representing the ancestral relationship between the nodes, as shown in Figure 2b).

To model copy number variation along the genome, we partition the genome into M fixed-sized bins, each aggregating sequencing reads mapped

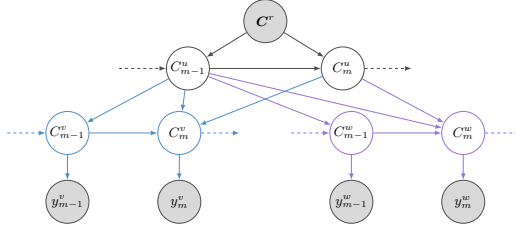


Fig. 3: Triple-HMM over two bins for the rooted cherry. Blue nodes and arrows correspond to node v , purple to node w , while the root r and observations $\mathbf{y}$ are shown in gray.

within its genomic interval. For each leaf v , the observed data form an M -dimensional vector $\mathbf{y}^v = (y_1^v, \dots, y_M^v)$, where y_m^v denotes the normalized read count in bin m . These observations arise from an underlying latent copy number profile $\mathbf{C}^v = (C_1^v, \dots, C_M^v)$, where each $C_m^v \in \{0, 1, \dots, K\}$ represents the discrete copy number state at bin m . Within each rooted cherry $T^\#$, Cellmates models the dependency between copy number profiles through a local evolutionary process along the three branches $(r \rightarrow u)$, $(u \rightarrow v)$, and $(u \rightarrow w)$. This structure allows us to describe the joint evolution of $\mathbf{C}^v$ and $\mathbf{C}^w$ while conditioning on their shared ancestor $\mathbf{C}^u$, and to estimate branch lengths that define a pairwise evolutionary distance between v and w .

We adopt the same CopyTree model as introduced in VICTree [17], where the latent copy number profiles evolve along the arcs of a given tree topology T according to the parent-child dependencies. Here we introduce ℓ as an arc distance instead of $\varepsilon \in [0, 1]$, an arc-specific transition probability, which is the original parametrization of the CopyTree model in VICTree. For each arc $u \rightarrow v \in E(T)$ with branch length ℓ_{uv} , the copy number of the child cell at bin m , C_m^v , depends on its own previous bin C_{m-1}^v and on the parent's copy number at bins m and $m-1$, C_m^u and C_{m-1}^u , forming a set of hierarchically interlaced Markov chains along the tree. Such dependencies are shown in Figure 2a. While we follow the exact CopyTree formulation of the VICTree, our goal in Cellmates is to integrate it with a phylogenetic substitution model, specifically the JCB model [8], to describe mutation processes along branches.

Because the classical Felsenstein pruning algorithm [5] accounts only for vertical dependencies along the tree (and not horizontal dependencies across the genomic bins), we define instead a Hidden Markov Model (HMM) over a rooted cherry, coupling the vertical (tree) and horizontal (genomic) dimensions. This yields a triple HMM, which jointly models the copy number evolution within each bin, across bins, and along the tree. To make the evolutionary distances additive along the tree, each branch length ℓ_{uv} is defined

in relation to a rate matrix Q_{uv} as follows

$$Q_{uv} = \begin{cases} -K\alpha_{uv}, & \text{if } C_m^v - C_{m-1}^v = C_m^u - C_{m-1}^u, \\ \alpha_{uv}, & \text{if } C_m^v - C_{m-1}^v \neq C_m^u - C_{m-1}^u, \end{cases} \quad (1)$$

where K is the maximum copy number state allowed and α_{uv} is the transition rate parameter. Under the continuous-time Markov framework, the transition matrix along a branch of length $\ell_{uv} = \alpha_{uv}t$ is $P_{uv}(t) = \exp(tQ_{uv})$.

2.2 JCB Model

We introduce the basic Jukes-Cantor-Breakpoint model, denoted JCB, mostly following the presentation of the JC model in [8]. In the traditional JC model used for nucleotide substitutions, the number of possible states is equal to 4 (corresponding to 4 nucleotides), and each state changes to any of the other three with equal instantaneous rate α . In our setting, latent variables represent copy number states rather than nucleotides, and therefore, if we let K be the maximum copy number allowed, the number of possible copy number states is $K + 1$ (including zero copies).

The model concerns how the breakpoints of a CN sequence $C_1^v, \dots, C_M^v$ of a cell v existing at time $t+h$ depend on those of a CN sequence $C_1^u, \dots, C_M^u$ of a parental cell u that existed at time t . A parental breakpoint C_{m-1}^u, C_m^u may given C_{m-1}^v either change or not and, importantly: change means $C_{m-1}^u - C_m^u \neq C_{m-1}^v - C_m^v$ and no change $C_{m-1}^u - C_m^u = C_{m-1}^v - C_m^v$. Notice that there, given C_{m-1}^v, C_m^u, C_m^v , K values of C_{m-1}^u that corresponds to change and a single value corresponding to no change. Solving the differential equation in A.1, group the JCB transition probabilities into two categories: the probability of no change (same breakpoint event in parent and child) $p_0(\ell)$, and the probability of change (any different breakpoint event) $p_1(\ell)$. Under the symmetric continuous-time model with rate α towards each of K alternative states, these are

$$p_0(\ell) = \frac{1}{K+1} + K \frac{e^{-(K+1)\ell}}{K+1}, \quad (2)$$

$$p_1(\ell) = \frac{1}{K+1} - \frac{e^{-(K+1)\ell}}{K+1}. \quad (3)$$

This sets the ground for the Jukes-Cantor correction for branch lengths. Given an observed proportion of copy number changes $\hat{p}$ between two copy number profiles, the JC-corrected branch length is

$$\ell = -\frac{1}{K+1} \log \left(1 - \frac{K+1}{K} \hat{p} \right). \quad (4)$$

2.3 Triple-HMM

This subsection is devoted to the triple-HMMs, i.e., HMMs for triplets, such as in Figure 2a, with a diploid root, edge lengths, JCB models associated with the edges, and cells as leaves, i.e., the leaves emit observations. The two leaves v and w emit observed sequences $\mathbf{y}^v = y_1^v, \dots, y_M^v$ and $\mathbf{y}^w = y_1^w, \dots, y_M^w$, respectively. Let the JCB transition over the edge $u \rightarrow v$, with length ℓ_{uv} , be denoted by $p_{\Delta^u \Delta^v}^v(\ell_{uv})$. As described above, for each $i \in \{v, w\}$, $m \in [M]$ and hidden CN state $\mathbf{C} = \{\mathbf{C}^v, \mathbf{C}^w\}$, there is a pre-computed emission probability $B_m^i = p(y_m^i | C_m^i)$. For a triplet configuration, $\mathbf{C}_{m-1}^\# = (C_{m-1}^u, C_{m-1}^v, C_{m-1}^w)$, an emission probability B_{m-1} and a transition probability A_{m-1} , to any $\mathbf{C}_m^\# = (C_m^u, C_m^v, C_m^w)$, are induced, as described below. Together, $\{B_m : m \in [M]\}$ and $\{A_m : m \in [M-1]\}$ define the triple-HMM.

Using the notation introduced above, the emission and transition probabilities are defined as follows:

$$B_m(y_m^v, y_m^w | \mathbf{C}_m^\#) = B_m^v(y_m^v | C_m^v) B_m^w(y_m^w | C_m^w) \quad (5)$$

$$A_{m-1}(\mathbf{C}_{m-1}^\#, \mathbf{C}_m^\#) = p_{0, \Delta^u}^u(\ell_{ru}) p_{\Delta^u, \Delta^v}^v(\ell_{uv}) p_{\Delta^u, \Delta^w}^w(\ell_{uw}) \quad (6)$$

where $\Delta^i = C_m^i - C_{m-1}^i$ and $\Delta^r = 0$ for the diploid root.

2.4 Expectation-Maximization for triple-HMM

Inference in the CopyTree model is challenging because the likelihood

$$p(\mathbf{y}^{vw} | \ell, \psi) = \sum_{\mathbf{C}} p(\mathbf{y}^{vw}, \mathbf{C} | \ell, \psi). \quad (7)$$

requires summing over all latent copy number configurations $\mathbf{C}$ across bins and tree nodes, which is computationally intractable to optimize directly with respect to branch length ℓ and emission parameters ψ . To estimate these parameters efficiently, we employ the expectation-maximization (EM) algorithm, which alternates between computing expectations over the tree latent states (E-step) and maximizing the expected complete-data log-likelihood (M-step). The E-step consists of fixing ℓ and maximizing the ELBO with respect to q , thereby setting

$$q^t = p(\mathbf{C}^\# | \mathbf{y}^{vw}, \ell^t, \psi^t). \quad (8)$$

We derive the expected data log-likelihood, also referred to as *Q-function*:

$$\begin{aligned}
Q((\ell, \psi), (\ell^t, \psi^t)) &= \\
&= \mathbb{E}_{p(\mathbf{C}^\# | \mathbf{y}^{vw}, \ell^t, \psi^t)} [\log p(\mathbf{C}^\# | \mathbf{y}^{vw} | \ell, \psi)] \quad (9) \\
&= \mathbb{E}_{p(\mathbf{C}^\# | \mathbf{y}^{vw}, \ell^t, \psi^t)} [\log p(\mathbf{y}^v | \mathbf{C}^v, \psi_v) \\
&\quad + \log p(\mathbf{y}^w | \mathbf{C}^w, \psi_w) + \log p(\mathbf{C}^\# | \ell)] \quad (10) \\
&= \mathbb{E}_{p(\mathbf{C}^\# | \mathbf{y}^{vw}, \ell^t, \psi^t)} \left[\sum_m [\log p_{0, \Delta_m^u(\ell_{ru})}] \right. \\
&\quad + [\log p_{\Delta_m^u, \Delta_m^v(\ell_{uv})}] + [\log p_{\Delta_m^u, \Delta_m^w(\ell_{uw})}] \\
&\quad \left. + \log p(y_m^v | C_m^v, \psi_v) + \log p(y_m^w | C_m^w, \psi_w) \right] \quad (11)
\end{aligned}$$

and the Q-term can be divided into two terms: $Q_A(\ell) + Q_B(\psi)$ as the likelihood terms do not depend on ℓ . So our problem breaks into two parts, one for emission and the other for transition probabilities.

Now we can use the EM algorithm for arc distances. We initialize by setting $\ell^0 = (\ell_{ru}^0, \ell_{uv}^0, \ell_{uw}^0)$ to random values. The E-step consists in fixing ℓ and maximizing the ELBO wrt q , thereby setting

$$q^t = p(\mathbf{C}^\# | \mathbf{y}^{vw}, \ell^t). \quad (12)$$

In the M-step we fix q and maximize with respect to ℓ . Our Q-function is $Q_A(\ell, \ell^t)$. Then decompose the expression and take the derivative of Q with respect to each of the length variables. For $\Delta' \neq \Delta$ we can split the expression writing

$$\mathbb{E}_{p(\mathbf{C}^\# | \mathbf{y}^{vw}, \ell^t)} [\log p_{\Delta_m^u, \Delta_m^v}(\ell_{uv})] = D(\ell_{uv}^t) \log p_{\Delta, \Delta'}(\ell_{uv}) + D'(\ell_{uv}^t) \log p_{\Delta, \Delta}(\ell_{uv}), \quad (13)$$

where $D(\ell_{uv}^t)$ is the expected number of breakpoints between u and v and $D'(\ell_{uv}^t)$ is the expected number of times we observe a co-mutation. The computation of $D(\ell_{uv}^t)$ and $D'(\ell_{uv}^t)$ is written explicitly in Appendix A.5 and uses the forward-backward algorithm. If we take the derivative of $Q_A(\ell, \ell^t)$ with respect to each arc-distance and solve by maximization, we get the EM estimates:

$$\hat{p}_{t+1} = \frac{D(\ell_{uv}^t)}{D(\ell_{uv}^t) + D'(\ell_{uv}^t)}, \quad (14)$$

with $\hat{p}$ being estimated proportion of changes.

Even though the M-step with respect to transition parameters ℓ above is independent of emission parameters, $D(\ell)$ is effected by the emission parameters when calculating the marginals in the E-step. For cell specific emission models, e.g., Gaussian cell-specific baseline described below, we can extend the EM-algorithm to infer quadruplet specific emission parameters, ψ_v and ψ_w , associated to the leaves v and w .

Equation (11) can be specified for any observational model, e.g., Normal or Poisson. For each cell v (and analogously w), we assume a cell-specific baseline μ_v and precision τ_v . Given the latent copy number $C_m^v \in \{0, 1, \dots, K\}$ at bin m , the observation follow Normal distribution

$$y_m^v | C_m^v, \mu_v, \tau_v \sim \mathcal{N}(C_m^v \mu_v, \tau_v^{-1}). \quad (15)$$

The corresponding log-density is:

$$\log p(y_m^v | C_m^v, \mu_v, \tau_v) = \frac{1}{2} \log \tau_v - \frac{1}{2} \log(2\pi) - \frac{\tau_v}{2} (y_m^v - C_m^v \mu_v)^2. \quad (16)$$

The same form holds for w with parameters μ_w, τ_w . Let $\psi = \{\mu_v, \tau_v, \mu_w, \tau_w\}$ denote the emission parameters and ψ^t their value at EM iteration t . The equation 9 with respect to ψ in the M-step is

$$Q_B(\psi, \psi^t) = \mathbb{E}_{p(\mathbf{C}^\# | \mathbf{y}^{vw}, \ell, \psi^t)} \left[\sum_m \log p(y_m^v | C_m^v, \psi_v) + \log p(y_m^w | C_m^w, \psi_w) \right]. \quad (17)$$

Using 16 and dropping the constants independent of ψ , the contribution from cell v becomes

$$Q_B^v(\mu_v, \tau_v) \equiv \frac{M}{2} \log \tau_v - \frac{\tau_v}{2} \sum_{m=1}^M \mathbb{E}[(y_m^v - C_m^v \mu_v)^2] + \text{const}. \quad (18)$$

Using posterior marginals from the E-step, the following variables are defined:

$$\gamma_1^v(m) = \mathbb{E}_{p(\mathbf{C}^\# | \mathbf{y}^{vw}, \ell, \psi^t)}[C_m^v], \quad (19)$$

$$\gamma_2^v(m) = \mathbb{E}_{p(\mathbf{C}^\# | \mathbf{y}^{vw}, \ell, \psi^t)}[(C_m^v)^2]. \quad (20)$$

Identical definitions apply to the variable w . For iteration $t + 1$, we find the optimal μ^{t+1} and τ^{t+1} by taking the derivative with respect to each parameter and setting it to zero and finally we find the M-step updates:

$$\mu_v = \frac{\sum_{m=1}^M y_m^v m_1^v(m)}{\sum_{m=1}^M m_2^v(m)}, \quad (21)$$

$$\tau_v = \frac{M}{\sum_{m=1}^M \mathbb{E}[(y_m^v - C_m^v \mu_v)^2]}. \quad (22)$$

2.5 Root-preserving Neighbor Joining

We reconstruct the tree using a recursive, root-aware agglomeration procedure (Algorithm 1), which is the DLCA–NJ algorithm of Gronau and Moran [6] operating on our estimated triplet distances; we restate it here in

our notation for self-containedness. For each pair of cells we use the triplet $(\ell_{ru}, \ell_{uv}, \ell_{uw})$, where ℓ_{ru} denotes the estimated distance from the root to the inferred last common ancestor (LCA), and ℓ_{uv}, ℓ_{uw} represent the distances from that LCA to each leaf, as defined in Section 2. In standard NJ, only the leaf-to-leaf component of the distance is used, therefore the triplet would have to be collapsed to a single metric, $\ell_{uv} + \ell_{uw}$, while the shared ancestral term ℓ_{ru} is discarded. This reduction is employed in SCONCE2 [7]. The algorithm described below combines the three distance components allowing for more accurate topologies, as we show in Section 3.

At each step of the reconstruction, the algorithm considers a working set of *units*, where a unit may be either a single leaf (cell) or an already assembled subtree. For every pair of units (v, w) we retain the full triplet structure $(\ell_{ru}, \ell_{uv}, \ell_{uw})$. The next merge event is determined by selecting the pair of units with the *largest shared ancestral depth*, i.e. the pair maximizing the related ℓ_{ru} length, which we hereby denote with d_{vw} for convenience. This is the deepest-LCA criterion of [6]. It parallels the role of the minimum-distance criterion in Neighbor Joining, but unlike NJ, the criterion preserves the ancestral depth component rather than collapsing the measurement to $\ell_{uv} + \ell_{uw}$.

Once two units (v, w) are merged, they form a new unit, denoted x . The distances between x and any remaining unit c must then be updated. This update is performed by averaging the corresponding distances from the two merged units. Specifically, for each remaining unit c , we update the median (shared-ancestry) distance as

$$d_{xc} = \frac{1}{2}(d_{vc} + d_{wc}). \quad (23)$$

From this, we compute the corresponding prospective internal branch length

$$L_x(c) = d_{vw} - d_{xc}, \quad (24)$$

which represents the inferred distance between x and the LCA that x would share with c if the pair (x, c) were to be selected for merging in a subsequent iteration. This value is stored but not yet committed to the tree, ensuring that branch lengths remain consistent once c participates in an actual merge event. The edge lengths attaching v and w to their new internal ancestor are then obtained directly from the preserved quantities or from the initial lengths ℓ_{uv}, ℓ_{uw} in case they are atomic units (i.e. leaves).

Because the method maintains all three components of the triplet distance during merging, the resulting reconstruction preserves deeper structure encoded in the evolutionary signal. By doing so, RP-NJ yields topologies whose branching order and branch lengths more accurately reflect the ancestral relationships inferred from the data. This is also reflected in its theoretical guarantees: DLCA-NJ has an optimal edge ℓ_∞ -radius of $\frac{1}{2}$, against $\frac{1}{4}$ for NJ [6].

RecTree 1 Root-preserving NJ

Input units S , shared ancestral depths D , internal branch lengths L **Output** T

```

1: if  $|S| = 2$  then
2:   define  $\{v, w\} = S$ 
3:    $T = \{(r, v, L_v(w)), (r, w, L_w(v))\}$ 
4: else
5:    $v, w = \operatorname{argmax}_{s, t \in S} d_{st}$ 
6:   merge  $x = (v, w)$ 
7:   update  $D', L'$  ▷ as in Eqs. (23)(24)
8:    $T = \text{RECTREE}(S \setminus \{v, w\} \cup \{x\}, D', L')$ 
9:    $T = T \cup \{(x, v, L_v(w)), (x, w, L_w(v))\}$  ▷ add merged units to the tree
10: end if
11: return  $T$ 

```

3 Experiments

We carry out synthetic and real data experiments to evaluate our framework. Initially, we compare RP-NJ to the original NJ [21] and the balanced minimum evolution algorithm [4], which are the two main algorithms used for phylogenetic tree reconstruction, even in the context of copy number trees. Then we assess the performances of the tree reconstruction algorithm applied to the triplet distances estimated by our EM algorithm, benchmarking Cellmates against two other copy number methods for phylogenetic tree reconstruction. Finally, we apply Cellmates to two datasets of ovarian cancer to demonstrate its robustness in real scenarios.

3.1 Improved tree reconstruction on triplet distances

Triplet distances between pairs of cells, as defined in Section 2 and in [7], effectively characterize two cells in the context of a copy number phylogeny. However, in order to employ traditional phylogeny reconstruction algorithms, the triplet distance must be reduced to a single distance metric, that is, $\ell_{uv} + \ell_{uw}$. We investigate the impact of our tailored root-preserving NJ algorithm by simulating binary trees of varying sizes and varying branch lengths and reconstructing the phylogeny using the triplet distances from the generated tree. We compute the Robinson-Foulds metric [20], that is, the number of clades that are present in one tree but not in the other, and we normalize it so that it is always between 0 and 1, simply by dividing it by the maximum number of possible mismatches, which equals $2(N - 3)$. We choose to run on five tree sizes, $N \in \{10, 20, 50, 100, 200\}$ and three levels of mutation rate. The mutation rate is determined by the branch lengths sampled over the given tree. We first simulate an ultra-metric tree

via birth-death process, then rescale the lengths so that the maximum leaf-root length matches the desired number of mutations (10, 20 and 50 CNAs). In Figure 4 we show the results of the experiment that compares our tree reconstruction algorithm with Neighbor Joining (from `scikit-bio` Python library) and Balanced Minimum Evolution (as implemented in the FastME software package [13]).

RP-NJ outperforms the other methods, especially for smaller cell counts. This clearly shows how NJ is not the correct approach to tree reconstruction when distances are computed with respect to a median node. As tree size increases, however, the three methods show similar performance.

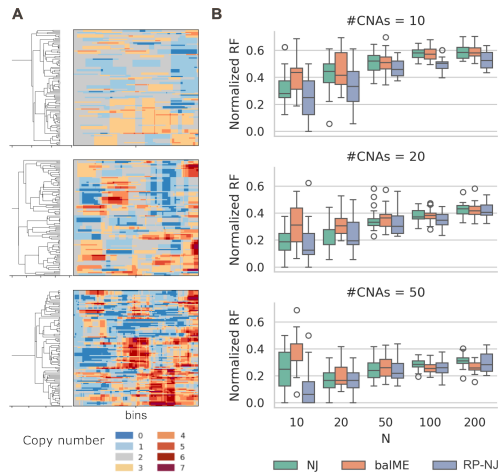


Fig. 4: Tree reconstruction algorithms benchmark. **A** Illustration of copy numbers and tree simulation with different copy number mutation rates for 100 cells and 1000 bins. **B** Boxplots showing aggregated normalized RF metric over 20 datasets for each size (N cells) and mutation rate.

3.2 Synthetic copy number data

In order to benchmark our method against other similar methods, we run Cellmates on 20 synthetic datasets generated with CNAsim [28], a tool for simulating single-cell copy number alteration (CNA) data from tumors. We compare our method with MEDICC2 [9] and DICE [29]. The latter, despite its simple distance metric, constitutes an excellent baseline thanks to the extremely fast execution time. Moreover, the benchmarks presented in DICE paper [29] provides further comparisons with currently available methods. Since both DICE and MEDICC2 operate on integer copy number profiles,

we generate noisy copy number datasets of 50 cells, 1000 bins, noise level of 0.1 (according to the error model in CNAsim) and two different copy number rates, 5 and 10 CNAs per edge. Furthermore, the cells are grouped in 4 subclones that are distinguished by larger (clonal) copy number events. For this experiment we adopt an observation model on integer copy numbers, parametrized by an error rate ε (in place of the normalized reads, as presented in Section 2). This model is further described in the Appendix. We complement the RF distance metric with an additional modified F1 score that measures how well the cells grouped into the same clone are represented in the tree. This score is the same adopted in [29]. Cellmates perform similarly to the other two methods, as can be seen in Figure 5. However, while being slightly less accurate in terms of RF distance, it shows better F1 score. This discordance in the metrics could be reflected in the sensitivity of the RF distance to small changes in the tree [3].

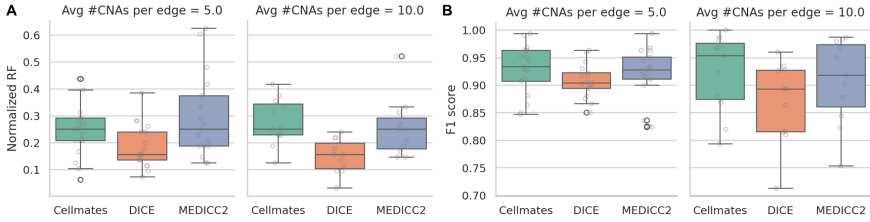


Fig. 5: Comparison of Cellmates with two other copy number reconstruction methods. **A** Normalized RF distance against the true tree and **B** F1 score against the true clones generated by CNAsim. Boxplots aggregate results from 10 datasets.

3.3 Ovarian cancer samples from DLP+ data

We further test our framework on two small datasets of ovarian cancer single-cell WGS data obtained with DLP+ protocol [12]. The reads have been corrected for GC-content and mappability bias according to the original study. Also, since the sample presents whole-genome duplicated cells, we use ploidy-corrected read counts to correctly identify the ploidy, provided in the data published with the original study [16]. We adopt our Normal observation model as described in Section 2 and run Cellmates with a maximum copy number state of $K = 8$. The counts are grouped into bins of fixed size (5Mb), resulting in a dataset with 598 bins for 32 tumor cells. In Figure 6, we show that the resulting tree from the analysis on the OV-125 dataset correctly identifies cells that have undergone WGD and further separates them into

two subclones (lower clades in Figure 6), one of which presents a larger copy number gain in the q arm of chromosome 2 and a lower copy number gain in chromosome 3. Copy numbers are called as a part of the framework using the Viterbi algorithm on the triple-HMM with the inferred branch lengths. The mean absolute deviation from the copy number called by HMMcopy is 0.162. The second dataset, OV-110, consists of 56 cells, all WGD. The subclonal structure is less clear but almost half of the cells have copy number 4 on the p arm of chromosome 3 and these are all under the same clade according to the reconstructed tree. Also, 4 cells stand out having one more copy than the others in chromosome 1. The mean absolute deviation from HMMcopy is 0.22 and the higher difference might be due to the fact that some higher copy numbers are capped at 8. We show the analysis on dataset OV-110 in the Appendix (Figure 7).

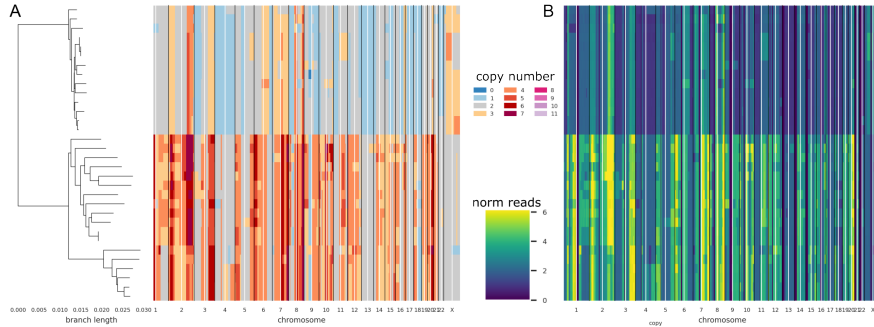


Fig. 6: **A** Phylogenetic tree and copy number profile inferred by Cellmates for the OV-125 dataset from the SPECTRUM cohort (32 tumor cells). **B** Read counts per bin and cell, corrected for GC content, mappability bias and scaled for ploidy.

4 Discussion

The Cellmates framework introduces a statistically principled approach to reconstructing tumor phylogenies from single-cell read count data, advancing beyond prior distance-based methods, such as MEDICC2, by incorporating probabilistic distance correction and a root-aware reconstruction procedure. Although developed independently from SCONE2, Cellmates builds on similar conceptual foundations for modeling copy number evolution—most notably in its use of a Jukes-Cantor breakpoint model, and further improves on the phylogenetic tree reconstruction by adopting Root-preserving Neighbor Joining (RP-NJ), i.e. DLCA–NJ [6], which explicitly maintains a diploid-root

constraint. This combination enables Cellmates to leverage the full structure of triplet distances-including leaf-leaf components and ancestral depth-rather than collapsing them into a single scalar metric as in standard NJ, resulting in more accurate topologies with branching patterns that better reflect true evolutionary history. Experimental evaluation on synthetic data shows that the RP-NJ algorithm outperforms the standard methods used for copy number trees. Analysis on ovarian cancer single-cell WGS data further validates Cellmates' robustness, accurately recovering whole-genome duplication events and identifying distinct subclonal expansions supported by the inferred copy number states. Its generative modeling foundation enables extensions such as statistical confidence assessment through bootstrapped tree sampling. Future works also might explore alternative RP-NJ agglomeration criteria and more sophisticated observation models (e.g., allele-specific counts). As such, our framework provides a practical and reliable solution for reconstructing copy number phylogenies directly from single-cell sequencing data and represents an important step forward in modeling the evolutionary dynamics of copy number change.

References

1. Andrade, J.R., Gallagher, A.D., Maharaj, J., McClelland, S.E.: Disentangling the roles of aneuploidy, chromosomal instability and tumour heterogeneity in developing resistance to cancer therapies. *Chromosome Research* **31**(4), 28 (Sep 2023). <https://doi.org/10.1007/s10577-023-09737-5>, <https://link.springer.com/10.1007/s10577-023-09737-5>
2. Baslan, T., Morris, J.P., Zhao, Z., Reyes, J., Ho, Y.J., Tsanov, K.M., Bermeo, J., Tian, S., Zhang, S., Askan, G., Yavas, A., Lecomte, N., Erakky, A., Varghese, A.M., Zhang, A., Kendall, J., Ghiban, E., Chorbadjiev, L., Wu, J., Dimitrova, N., Chadalavada, K., Nanjangud, G.J., Bandlamudi, C., Gong, Y., Donoghue, M.T.A., Socci, N.D., Krasnitz, A., Notta, F., Leach, S.D., Iacobuzio-Donahue, C.A., Lowe, S.W.: Ordered and deterministic cancer genome evolution after p53 loss. *Nature* **608**(7924), 795–802 (Aug 2022). <https://doi.org/10.1038/s41586-022-05082-5>, <https://www.nature.com/articles/s41586-022-05082-5>
3. Briand, S., Dessimoz, C., El-Mabrouk, N., Lafond, M., Lobinska, G.: A generalized Robinson-Foulds distance for labeled trees. *BMC Genomics* **21**(S10), 779 (Nov 2020). <https://doi.org/10.1186/s12864-020-07011-0>, <https://bmcbgenomics.biomedcentral.com/articles/10.1186/s12864-020-07011-0>
4. Desper, R., Gascuel, O.: Fast and Accurate Phylogeny Reconstruction Algorithms Based on the Minimum-Evolution Principle. *Journal of Computational Biology* **9**(5), 687–705 (Oct 2002). <https://doi.org/10.1089/106652702761034136>, <http://www.liebertpub.com/doi/10.1089/106652702761034136>
5. Felsenstein, J.: Evolutionary trees from dna sequences: a maximum likelihood approach. *Journal of molecular evolution* **17**(6), 368–376 (1981)

6. Gronau, I., Moran, S.: Neighbor joining algorithms for inferring phylogenies via LCA distances. *J. Comput. Biol.* **14**, 1–15 (2007). <https://doi.org/10.1089/cmb.2006.0115>
7. Hui, S., Nielsen, R.: Sconce: a method for profiling copy number alterations in cancer evolution using single-cell whole genome sequencing. *Bioinformatics* **38**(7), 1801–1808 (Jan 2022). <https://doi.org/10.1093/bioinformatics/btac041>, <http://dx.doi.org/10.1093/bioinformatics/btac041>
8. Jukes, T.H., Cantor, C.R.: *Evolution of Protein Molecules*, pp. 21–132. Elsevier (1969). <https://doi.org/10.1016/b978-1-4832-3211-9.50009-7>
9. Kaufmann, T.L., Petkovic, M., Watkins, T.B.K., Colliver, E.C., Laskina, S., Thapa, N., Minussi, D.C., Navin, N., Swanton, C., Van Loo, P., Haase, K., Tarabichi, M., Schwarz, R.F.: MEDICC2: whole-genome doubling aware copy-number phylogenies for cancer evolution. *Genome Biology* **23**(1), 241 (Nov 2022). <https://doi.org/10.1186/s13059-022-02794-9>, <https://genomebiology.biomedcentral.com/articles/10.1186/s13059-022-02794-9>
10. Kimura, M.: A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences. *Journal of molecular evolution* **16**(2), 111–120 (1980)
11. Kuipers, J., Tuncel, M.A., Ferreira, P., Jahn, K., Beerenwinkel, N.: Single-cell copy number calling and event history reconstruction (Apr 2020), preprint
12. Laks, E., McPherson, A., Zahn, H., Lai, D., Steif, A., Brimhall, J., Biele, J., Wang, B., Masud, T., Ting, J., Grewal, D., Nielsen, C., Leung, S., Bojilova, V., Smith, M., Golovko, O., Poon, S., Eirew, P., Kabeer, F., Ruiz de Algora, T., Lee, S.R., Taghiyar, M.J., Huebner, C., Ngo, J., Chan, T., Vatr-Watts, S., Walters, P., Abrar, N., Chan, S., Wiens, M., Martin, L., Scott, R.W., Underhill, T.M., Chavez, E., Steidl, C., Da Costa, D., Ma, Y., Coope, R.J.N., Corbett, R., Pleasance, S., Moore, R., Mungall, A.J., Mar, C., Cafferty, F., Gelmon, K., Chia, S., CRUK IMAXT Grand Challenge Team, Marra, M.A., Hansen, C., Shah, S.P., Aparicio, S.: Clonal decomposition and DNA replication states defined by scaled Single-Cell genome sequencing. *Cell* **179**(5), 1207–1221.e22 (Nov 2019)
13. Lefort, V., Desper, R., Gascuel, O.: FastME 2.0: A Comprehensive, Accurate, and Fast Distance- Based Phylogeny Inference Program: Table 1. *Molecular Biology and Evolution* **32**(10), 2798–2800 (Oct 2015). <https://doi.org/10.1093/molbev/msv150>, <https://academic.oup.com/mbe/article-lookup/doi/10.1093/molbev/msv150>
14. Markowska, M., Cakała, T., Miasojedow, B., Aybey, B., Juraeva, D., Mazur, J., Ross, E., Staub, E., Szczurek, E.: CONET: copy number event tree model of evolutionary tumor history for single-cell data. *Genome Biol.* **23**(1), 128 (Jun 2022)
15. McGranahan, N., Burrell, R.A., Endesfelder, D., Novelli, M.R., Swanton, C.: Cancer chromosomal instability: therapeutic and diagnostic challenges: 'Exploring aneuploidy: the significance of chromosomal imbalance' review series. *EMBO reports* **13**(6), 528–538 (May 2012). <https://doi.org/10.1038/embor.2012.61>, <http://dx.doi.org/10.1038/embor.2012.61>
16. McPherson, A., Vázquez-García, I., Myers, M.A., Al-Rawi, D.H., Zatzman, M., Weiner, A.C., Freeman, S., Mohibullah, N., Satas, G., Williams, M.J., Ceglia, N., Norkūnaitė, D., Zhang, A.W., Li, J., Lim, J.L.P., Wu, M., Choi,

- S., Havasov, E., Grewal, D., Shi, H., Kim, M., Schwarz, R.F., Kaufmann, T., Dinh, K.N., Uhlitz, F., Tran, J., Wu, Y., Patel, R., Ramakrishnan, S., Kim, D., Clarke, J., Green, H., Ali, E., DiBona, M., Varice, N., Kundra, R., Broach, V., Gardner, G.J., Roche, K.L., Sonoda, Y., Zivanovic, O., Kim, S.H., Grisham, R.N., Liu, Y.L., Viale, A., Rusk, N., Lakhman, Y., Ellenson, L.H., Tavaré, S., Aparicio, S., Chi, D.S., Aghajanian, C., Abu-Rustum, N.R., Friedman, C.F., Zamarin, D., Weigelt, B., Bakhoun, S.F., Shah, S.P.: Ongoing genome doubling shapes evolvability and immunity in ovarian cancer. *Nature* **644**(8078), 1078–1087 (Aug 2025). <https://doi.org/10.1038/s41586-025-09240-3>, <https://www.nature.com/articles/s41586-025-09240-3>
17. Melin, H., Zampinetti, V., McPherson, A., Lagergren, J.: Victree-a variational inference method for clonal tree reconstruction. In: International Conference on Research in Computational Molecular Biology. pp. 429–433. Springer (2024)
 18. Negrini, S., Gorgoulis, V.G., Halazonetis, T.D.: Genomic instability - an evolving hallmark of cancer. *Nature Reviews Molecular Cell Biology* **11**(3), 220–228 (Mar 2010). <https://doi.org/10.1038/nrm2858>, <http://dx.doi.org/10.1038/nrm2858>
 19. Nowell, P.C.: The clonal evolution of tumor cell populations. *Science* **194**(4260), 23–28 (Oct 1976)
 20. Robinson, D., Foulds, L.: Comparison of phylogenetic trees. *Mathematical Biosciences* **53**(1-2), 131–147 (Feb 1981). [https://doi.org/10.1016/0025-5564\(81\)90043-2](https://doi.org/10.1016/0025-5564(81)90043-2), <https://linkinghub.elsevier.com/retrieve/pii/0025556481900432>
 21. Saitou, N., Nei, M.: The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Molecular biology and evolution* **4**(4), 406–425 (1987)
 22. Salehi, S., Dorri, F., Chern, K., Kabeer, F., Rusk, N., Funnell, T., Williams, M.J., Lai, D., Andronescu, M., Campbell, K.R., McPherson, A., Aparicio, S., Roth, A., Shah, S.P., Bouchard-Côté, A.: Cancer phylogenetic tree inference at scale from 1000s of single cell genomes. *Peer Community Journal* **3**, e63 (2023). <https://doi.org/10.24072/pcjournal.292>, <https://peercommunityjournal.org/articles/10.24072/pcjournal.292/>
 23. Schwarz, R.F., Trinh, A., Sipos, B., Brenton, J.D., Goldman, N., Markowitz, F.: Phylogenetic quantification of intra-tumour heterogeneity. *PLoS computational biology* **10**(4), e1003535 (2014)
 24. Turajlic, S., Sottoriva, A., Graham, T., Swanton, C.: Resolving genetic heterogeneity in cancer. *Nature Reviews Genetics* **20**(7), 404–416 (Mar 2019). <https://doi.org/10.1038/s41576-019-0114-6>, <http://dx.doi.org/10.1038/s41576-019-0114-6>
 25. Vittoria, M.A., Quinton, R.J., Ganem, N.J.: Whole-genome doubling in tissues and tumors. *Trends in Genetics* **39**(12), 954–967 (Dec 2023). <https://doi.org/10.1016/j.tig.2023.08.004>, <https://linkinghub.elsevier.com/retrieve/pii/S0168952523001877>
 26. Wang, F., Wang, Q., Mohanty, V., Liang, S., Dou, J., Han, J., Minussi, D.C., Gao, R., Ding, L., Navin, N., Chen, K.: MEDALT: single-cell copy number lineage tracing enabling gene discovery. *Genome Biol.* **22**(1), 70 (Feb 2021)
 27. Wazir, I., Garry, T., Ashbourne, P., Barclay, P., Flynn, P., Frederick, K., Wakeford, M.: Mathematics 2012 Edition-Standard Level: Developed Specifically for the IB Diploma. Pearson (2012)

28. Weiner, S., Bansal, M.S.: CNAsim: improved simulation of single-cell copy number profiles and DNA-seq data from tumors. *Bioinformatics* **39**(7), btad434 (Jul 2023). <https://doi.org/10.1093/bioinformatics/btad434>, <https://academic.oup.com/bioinformatics/article/doi/10.1093/bioinformatics/btad434/7224244>
29. Weiner, S., Bansal, M.S.: DICE: fast and accurate distance-based reconstruction of single-cell copy number phylogenies. *Life Science Alliance* **8**(3), e202402923 (Mar 2025). <https://doi.org/10.26508/lsa.202402923>, <https://www.life-science-alliance.org/lookup/doi/10.26508/lsa.202402923>
30. Zaccaria, S., Raphael, B.J.: Characterizing allele- and haplotype-specific copy numbers in single cells with CHISEL. *Nat. Biotechnol.* **39**(2), 207–214 (Feb 2021)
31. Zahn, H., Steif, A., Laks, E., Eirew, P., VanInsberghe, M., Shah, S.P., Aparicio, S., Hansen, C.L.: Scalable whole-genome single-cell library preparation without preamplification. *Nat. Methods* **14**(2), 167–173 (Feb 2017)

A Model formulation

A.1 Derivation for the Differential equation

We have the following differential equation

$$\frac{dp_{\Delta,\Delta'}(l)}{dl} \tag{25}$$

$$= -K\alpha p_{\Delta,\Delta'}(l) + \alpha \sum_{\Delta^* \neq \Delta'} p_{\Delta,\Delta^*}(l) \tag{26}$$

$$= -K\alpha p_{\Delta,\Delta'}(l) + \alpha(1 - p_{\Delta,\Delta'}(l)) \tag{27}$$

$$= -(K+1)\alpha p_{\Delta,\Delta'}(l) + \alpha. \tag{28}$$

The differential equation given by (25) to (28) is separable, so for $p_{\Delta,\Delta}(t)$ it can be solved as follows [27]:

$$\begin{aligned}
\frac{dp_{\Delta,\Delta}(t)}{dt} &= -(K+1)\alpha p_{\Delta,\Delta}(t) + \alpha \\
&\Downarrow \\
\frac{dp_{\Delta,\Delta}(t)}{-(k+1)\alpha t + \alpha} &= dt \\
&\Downarrow \\
-\frac{dp_{\Delta,\Delta}}{\alpha((k+1)t - 1)} &= dt \\
&\Downarrow \\
-\int \frac{dp_{\Delta,\Delta}}{\alpha((k+1)t - 1)} &= \int dt \\
&\Downarrow \\
-\frac{1}{\alpha} \int \frac{dp_{\Delta,\Delta}}{(k+1)t - 1} &= x + c \\
&\Downarrow \\
-\frac{1}{\alpha} \frac{\ln((k+1)t)}{(k+1)} &= x + c \\
&\Downarrow \\
\ln(k+1)t - 1 &= -\alpha(k+1)t + c' \\
&\Downarrow \\
(k+1)t - 1 &= e^{-\alpha(k+1)t + c'} \\
&\Downarrow \\
(k+1)t &= 1 + e^{-\alpha(k+1)t + c'} \\
&\Downarrow \\
t &= \frac{1 + e^{-\alpha(k+1)t + c'}}{(k+1)}
\end{aligned}$$

Since

$$p_{\Delta,\Delta}(0) = 1$$

it follows that

$$\begin{aligned}
p_{\Delta,\Delta}(0) &= \frac{1 + e^{c'}}{(k+1)} \\
&\Downarrow \\
e^{c'} &= k \\
&\Downarrow \\
p_{\Delta,\Delta}(t) &= \frac{1}{k+1} + \frac{k}{k+1} e^{-\alpha(k+1)t}
\end{aligned}$$

Here we show how we derive $l_{uv}^{i+1} = \operatorname{argmax}_{l_{uv}} Q(\mathbf{l}, \mathbf{l}^i)$.

$$\frac{d}{dl} p_{\Delta,\Delta}(l) = -\alpha K e^{-l\alpha(K+1)} \quad (29)$$

$$\frac{d}{dl} p_{\Delta,\Delta'}(l) = \alpha e^{-l\alpha(K+1)} . \quad (30)$$

$$\begin{aligned}
\frac{d \log p_{\Delta,\Delta}(l)}{dl} &= \frac{\alpha(K+1)}{1 + K e^{-l\alpha(K+1)}} \left(-K e^{-l\alpha(K+1)} \right) \\
&= -\frac{\alpha(K+1) K e^{-l\alpha(K+1)}}{1 + K e^{-l\alpha(K+1)}} .
\end{aligned} \quad (31)$$

$$\frac{d \log p_{\Delta,\Delta'}(l)}{dl} = \frac{\alpha(K+1) e^{-l\alpha(K+1)}}{1 - e^{-l\alpha(K+1)}} . \quad (32)$$

We then solve the maximization equation, replacing $e^{-l(K+1)}$ with x and D, D' accordingly for simplicity

$$\begin{aligned}
& \frac{dQ(\mathbf{l}, \mathbf{l}^i)}{dl_{uv}} = 0 \\
& \Downarrow \\
& D(l_{uv}^i) \frac{\alpha(K+1)}{1 - e^{-l\alpha(K+1)}} e^{-l\alpha(K+1)} = D'(l_{uv}^i) \frac{\alpha(K+1)Ke^{-l\alpha(K+1)}}{1 + Ke^{-l\alpha(K+1)}} \\
& \Downarrow \\
& D \frac{x(K+1)}{1-x} = D' \frac{(K+1)Kx}{1+Kx} \\
& \Downarrow \\
& D(1+Kx) = (K-Kx)D' \\
& \Downarrow \\
& DKx + D'Kx = KD' - D \\
& \Downarrow \\
& x = \frac{KD' - D}{K(D' + D)} \\
& \Downarrow \\
& l_{uv} = -\frac{1}{\alpha(K+1)} \log \frac{KD' - D}{K(D' + D)} \\
& \Downarrow \\
& l_{uv} = -\frac{1}{\alpha(K+1)} \log \left(1 - \frac{K+1}{K} \frac{D'}{D' + D} \right)
\end{aligned} \tag{33}$$

A.2 Derivation for the Poisson emission model

Model Setup We denote by y_m^v (respectively y_m^w) the observed read-count in bin $m = 1, \dots, M$ of leaf cell v (resp. w). The latent copy-number state at bin m is denoted

$$C_m^v \in \{0, 1, \dots, K\}, \quad C_m^w \in \{0, 1, \dots, K\}.$$

We assume that, conditional on the latent state,

$$\begin{aligned}
y_m^v &| \{C_m^v = j\}, \lambda_v \sim \text{Poisson}(\mu = \lambda_v j), \\
y_m^w &| \{C_m^w = j\}, \lambda_w \sim \text{Poisson}(\mu = \lambda_w j).
\end{aligned}$$

Recalling the mass-function

$$p(y; \mu) = \frac{\mu^y e^{-\mu}}{y!}, \quad \log p(y; \mu) = y \log \mu - \mu - \log(y!),$$

we have, for leaf v ,

$$\log p(y_m^v \mid C_m^v = j, \lambda_v) = y_m^v \log(\lambda_v j) - \lambda_v j - \log(y_m^v!),$$

and similarly for leaf w .

Complete-Data Log-Likelihood Assuming independence of counts across bins and between leaves conditional on latent states, the complete-data log-likelihood is

$$\log p(\mathbf{y}^{vw} \mid \mathbf{C}, \psi) = \sum_{m=1}^M \left[\log p(y_m^v \mid C_m^v, \lambda_v) + \log p(y_m^w \mid C_m^w, \lambda_w) \right].$$

Substituting the above gives

$$\begin{aligned} \log p(\mathbf{y}^{vw} \mid \mathbf{C}, \psi) &= \sum_{m=1}^M \left[y_m^v \log(\lambda_v C_m^v) - \lambda_v C_m^v - \log(y_m^v!) \right] \\ &\quad + \sum_{m=1}^M \left[y_m^w \log(\lambda_w C_m^w) - \lambda_w C_m^w - \log(y_m^w!) \right]. \end{aligned}$$

E-Step Define responsibilities

$$\gamma_{m,j}^v = \mathbb{P}(C_m^v = j \mid \mathbf{y}^{vw}, \psi^t), \quad \gamma_{m,j}^w = \mathbb{P}(C_m^w = j \mid \mathbf{y}^{vw}, \psi^t).$$

Then up to additive constants independent of λ_v, λ_w ,

$$\begin{aligned} Q(\lambda_v, \lambda_w, \psi^t) &= \sum_{m=1}^M \sum_{j=0}^K \gamma_{m,j}^v \left[y_m^v \log(\lambda_v j) - \lambda_v j \right] \\ &\quad + \sum_{m=1}^M \sum_{j=0}^K \gamma_{m,j}^w \left[y_m^w \log(\lambda_w j) - \lambda_w j \right] + \text{const.} \end{aligned}$$

M-Step We maximize Q with respect to λ_v (and similarly for λ_w). For λ_v ,

$$Q(\lambda_v) = \sum_{m,j} \gamma_{m,j}^v \left[y_m^v \log \lambda_v + y_m^v \log j - \lambda_v j \right] + \text{const.}$$

Differentiating,

$$\frac{\partial Q}{\partial \lambda_v} = \sum_{m,j} \gamma_{m,j}^v \left(\frac{y_m^v}{\lambda_v} - j \right) = 0 \quad \implies \quad \lambda_v = \frac{\sum_{m=1}^M \sum_{j=0}^K \gamma_{m,j}^v y_m^v}{\sum_{m=1}^M \sum_{j=0}^K \gamma_{m,j}^v j}.$$

Similarly,

$$\lambda_w = \frac{\sum_{m=1}^M \sum_{j=0}^K \gamma_{m,j}^w y_m^w}{\sum_{m=1}^M \sum_{j=0}^K \gamma_{m,j}^w j}.$$

Summary Thus we obtain closed-form EM updates for the scaling parameters λ_v and λ_w , expressed in terms of posterior responsibilities $\gamma_{m,j}^v$ and $\gamma_{m,j}^w$.

A.3 Proof of Lemma

Lemma 1 (Cherry Marginalization). *Consider a rooted tree T with root r and a Markov chain on tree (MCT) on latent copy number state vectors $\{C^x : x \in T\}$ evolving along each edge $(u \rightarrow v) \in E(T)$ with transition probability*

$$p(C^v \mid C^u, \ell_{uv})$$

and root prior $\pi(C^r)$, in our model with fixed initial state $C^r = (2, \dots, 2)$. Let u be an internal node and v, w two descendant leaves whose least common ancestor (LCA) is precisely u . Then the marginal joint distribution

$$p(C^u, C^v, C^w \mid \ell)$$

depends only on the three branch-lengths ℓ_{ru} , ℓ_{uv} , ℓ_{uw} .

Proof. Full joint factorization. By construction the Markov-on-tree process yields

$$p(\{C^x : x \in V\} \mid \ell) = \pi(C^r) \prod_{(u \rightarrow v) \in E(T)} p(C^v \mid C^u, \ell_{uv}).$$

This is the standard phylogenetic factorization.

Marginalization over all other nodes. Let $U = \{u, v, w\}$. Then the marginal distribution of interest is

$$p(C^u, C^v, C^w \mid \ell) = \sum_{\{C^x : x \in V \setminus U\}} \pi(C^r) \prod_{(x \rightarrow y) \in E(T)} p(C^y \mid C^x, \ell_{xy}).$$

By summing over all latent states outside U , we formally marginalize the process.

Collapse of the $r \rightarrow u$ path. Note that all edges other than $(r \rightarrow u)$, $(u \rightarrow v)$ and $(u \rightarrow w)$ form sub-trees that either feed into u or descend from u . Because of the Markov property, summing over the latent states on those edges yields a marginal distribution on C^u of the form, writing P_{ru} for the path from r to u ,

$$\pi_u(C^u \mid \ell_{ru}) := \sum_{\{C^x : x \in P_{ru}\}} \pi(C^r) \prod_{(x \rightarrow y) \in P_{ru}} p(C^y \mid C^x, \ell_{xy}).$$

Since this path may contain several edges, one may define an effective branch-length

$$\ell_{ru} = \sum_{(x \rightarrow y) \in P_{ru}} \ell_{xy},$$

so that the marginal prior for C^u depends only on ℓ_{ru} .

Restriction to the three relevant edges. Once the states outside $\{u, v, w\}$ have been summed out, the only remaining transitions are those on the edges $(u \rightarrow v)$ and $(u \rightarrow w)$. Hence the marginal joint becomes

$$p(C^u, C^v, C^w \mid \ell) = \pi_u(C^u \mid \ell_{ru}) p(C^v \mid C^u, \ell_{uv}) p(C^w \mid C^u, \ell_{uw}).$$

Because each factor depends only on $\ell_{ru}, \ell_{uv}, \ell_{uw}$, the entire marginal depends solely on these three branch-lengths.

Conclusion. We have shown that despite the full tree containing many nodes and edges, the marginal distribution over the triple (u, v, w) is a function only of the collapsed root-to- u distance ℓ_{ru} and the two child edge lengths ℓ_{uv}, ℓ_{uw} . This completes the proof. $\square$

This result justifies treating each “rooted cherry” unit (even when the root-to-LCA path is collapsed) as a self-contained probabilistic block whose parameters (branch-lengths) can be estimated independently of the remainder of the tree.

A.4 JitterCopy Observation Model.

For each genomic bin m , let the latent true copy number be $C_m \in \{0, 1, \dots, K-1\}$. The observed copy number is an integer $C'_m \in \mathbb{Z}_{\geq 0}$.

Conditioned on the latent state, the observation is generated by sampling from a Gaussian distribution with mean equal to the true copy number and a standard deviation proportional to it:

$$X_m \sim \mathcal{N}(C_m, (C_m \cdot \varepsilon)^2), \quad C'_m = \text{round}(X_m), \quad (34)$$

where $\varepsilon > 0$ is a global noise parameter.

Since C'_m is discrete, the emission probability is given by the probability mass of this Gaussian within the rounding interval:

$$p(C'_m = y \mid C_m = c, \varepsilon) = \Phi\left(\frac{y + 0.5 - c}{c \cdot \varepsilon}\right) - \Phi\left(\frac{y - 0.5 - c}{c \cdot \varepsilon}\right),$$

where $\Phi(\cdot)$ denotes the standard normal cumulative distribution function.

A.5 Forward-backward algorithm

The computation of $D(\ell_{uv}^t)$ and $D'(\ell_{uv}^t)$ uses the standard forward-backward algorithm for Hidden Markov Models. Let $\mathbf{C}_m^\# = (C_m^u, C_m^v, C_m^w)$ denote the latent state at genomic bin $m \in \{1, \dots, M\}$. The forward probabilities $\alpha_m(\mathbf{C}_m^\#)$ and backward probabilities $\beta_m(\mathbf{C}_m^\#)$ are computed recursively as

usual. Using these, the posterior marginal probability of a transition between states at bin $m - 1$ and m is given by:

$$\begin{aligned}\xi_m(\mathbf{C}_{m-1}^\#, \mathbf{C}_m^\#) &= P(\mathbf{C}_{m-1}^\#, \mathbf{C}_m^\# \mid \mathbf{y}^{vw}, \boldsymbol{\ell}^t, \boldsymbol{\psi}^t) \\ &= \frac{\alpha_{m-1}(\mathbf{C}_{m-1}^\#) A_{m-1}(\mathbf{C}_{m-1}^\#, \mathbf{C}_m^\#) B_m(\mathbf{C}_m^\#) \beta_m(\mathbf{C}_m^\#)}{P(\mathbf{y}^{vw} \mid \boldsymbol{\ell}^t, \boldsymbol{\psi}^t)}\end{aligned}\quad (35)$$

where A_{m-1} is the transition probability, B_m is the emission probability, and the denominator is the likelihood of the observation sequence.

Recall that $\Delta_m^i = C_m^i - C_{m-1}^i$ for $i \in \{u, v, w\}$. The expected number of breakpoints (changes) between u and v across all bins is the sum of the posterior probabilities of transitions where the copy number change differs between u and v :

$$D(\ell_{uv}^t) = \sum_{m=2}^M \sum_{\mathbf{C}_{m-1}^\#, \mathbf{C}_m^\#} \xi_m(\mathbf{C}_{m-1}^\#, \mathbf{C}_m^\#) \mathbb{I}(\Delta_m^u \neq \Delta_m^v), \quad (36)$$

where $\mathbb{I}(\cdot)$ is the indicator function.

Similarly, the expected number of co-mutations (no changes) between u and v is the sum of the posterior probabilities of transitions where the copy number change is identical:

$$D'(\ell_{uv}^t) = \sum_{m=2}^M \sum_{\mathbf{C}_{m-1}^\#, \mathbf{C}_m^\#} \xi_m(\mathbf{C}_{m-1}^\#, \mathbf{C}_m^\#) \mathbb{I}(\Delta_m^u = \Delta_m^v). \quad (37)$$

These expected quantities are directly used in the M-step to update the branch length parameters (and analogously for the branch uw).

B Additional experiments

In this section, we present the full analysis and phylogenetic tree inferred for the OV-110 dataset, which consists of 56 whole-genome duplicated (WGD) cells from the SPECTRUM cohort. As described in Section 3.3, the subclonal structure in OV-110 is less pronounced compared to OV-125. However, Cellmates effectively groups nearly half of the cells that share a distinct copy number gain (copy number 4) on the p arm of chromosome 3 into a single clade. Additionally, the inferred phylogeny successfully isolates a small group of four cells characterized by an extra copy in chromosome 1. The mean absolute deviation from the baseline HMMcopy calls is 0.22, with the slight increase in deviation likely resulting from our limit capping maximum copy numbers at 8.

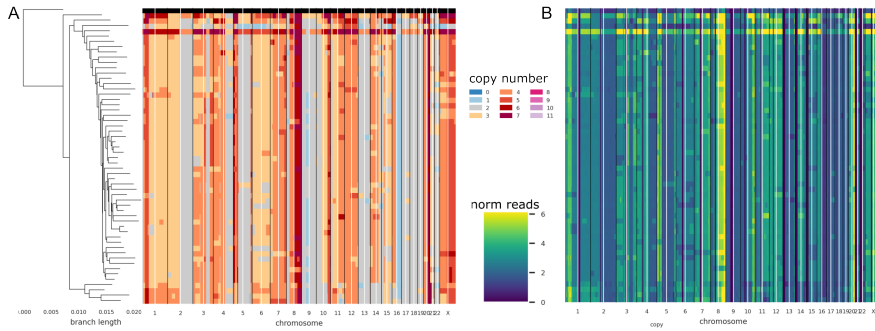


Fig. 7: **A** Phylogenetic tree and copy number profile inferred by Cellmates for the OV-110 dataset from the SPECTRUM cohort (56 cells and 598 bins of size 5Mb). **B** Read counts have been corrected as described in Section 3.3.

Paper B

Paper B

Scalable and robust phylogenetic tree reconstruction from copy number data with Sparse Rooted Neighbor Joining

VITTORIO ZAMPINETTI, HARALD MELIN, ALVA HALLIN,
AND JENS LAGERGREN

Preprint

bioRxiv. DOI: 10.64898/2026.07.30.739152

Scalable and robust phylogenetic tree reconstruction from copy-number data with Sparse Rooted Neighbor Joining

Vittorio Zampinetti^{1,2*}, Harald Melin², Alva Hallin²,
Jens Lagergren²

¹Department of Mathematical Sciences, Politecnico di Torino,
Turin, Italy.

²School of Electrical Engineering and Computer Science
(EECS), KTH Royal Institute of Technology, Stockholm,
Sweden.

*Corresponding author(s). E-mail(s): vz@kth.se;

Abstract

Background: Phylogenetic tree reconstruction from single cell data based on copy-number alterations (CNAs) is an important problem in cancer genomics. Methods have been developed to address this problem by computing pairwise distances between copy-number profiles and employing a tree reconstruction algorithm. Despite the tight interplay between distance estimation and tree reconstruction, these two steps are often treated as separate problems, with the choice of the reconstruction algorithm receiving little attention. Most methods rely on classical Neighbor Joining (NJ), an algorithm designed for unrooted phylogenies that does not account for the fixed diploid root inherent to copy-number evolution.

Results: We identify the Deepest Least Common Ancestor NJ (DLCA-NJ), not previously applied in this context, as the appropriate algorithm for phylogenies from copy-number data. By leveraging the known diploid root, it consistently outperforms standard NJ on simulated benchmarks across all evaluated metrics, with the most pronounced improvement in root placement accuracy. Building on these

findings, we introduce Sparse Rooted Neighbor Joining (SRNJ), a scalable adaptation of DLCA–NJ. SRNJ significantly reduces running time while trading off only a minor loss in accuracy. We provide theoretical and empirical evidence of robustness to mutation rate using both synthetic and real biological datasets.

Conclusions: Rooted NJ variants offer a principled way to exploit the known diploid root when reconstructing phylogenies from copy-number data, and SRNJ extends this advantage to datasets whose size places the full distance matrix out of reach. The gains are clearest where distances are reliable, as on simulated data, while on real data accuracy appears to be constrained by distance estimation rather than by the reconstruction algorithm, leaving room for improvement as callers advance.

Keywords: tumor phylogenetics, copy-number alterations, neighbor joining

1 Introduction

Copy-number alterations (CNAs) are a hallmark of cancer genomes and play a crucial role in tumor evolution and heterogeneity. Single-cell DNA sequencing technologies have enabled the profiling of CNAs at single-cell resolution, providing valuable insights into the evolutionary history of tumors [1–4]. CNAs can be used to identify driver mutations, understand tumor progression, and inform treatment strategies [5, 6]. To this end, several computational methods have been developed to reconstruct phylogenetic trees from CNA data, addressing challenges such as noise, missing data, and complex events like whole-genome duplications (WGDs) [7–13].

Just like in classical phylogenetics, the most scalable methods for tree reconstruction from CNA data are distance-based, called so because they rely only on pairwise distances between samples to infer the topology, where a sample can be either a copy-number profile or a sequence of raw DNA read-counts. In particular, these methods are based on two key components, namely the estimation of pairwise distances between samples, and the application of a tree reconstruction algorithm that takes these distances as input and infers the tree topology by optimizing some criterion. For the sake of clarity, the latter will be hereafter referred to as *distance-based* method, whereas the combination of the two will be referred to as *sequence-based* method, following the naming convention of [14]. While often treated as sequential, the two steps of a sequence-based method are fundamentally linked and may overlap. Early distance-based methods use simple distance estimates such as the Euclidean or Manhattan distance between copy-number (CN) profiles [3, 4], which must be considered a poor approximation of the underlying evolutionary process, given that copy-number alterations

affect contiguous genomic regions, introducing strong dependencies between adjacent bins. More advanced distances include the Minimal Event Distance (MED), implemented in MEDICC2 [10]. This widely used approach computes the minimum number of copy-number alteration events required to transform one copy-number profile into another and uses this quantity for phylogenetic reconstruction. As such, MED is based on a minimum-event criterion rather than an explicit probabilistic model of CNA evolution. In classical phylogenetics, model-based distances account for the possibility that multiple evolutionary events may affect the same genomic locus over time, providing a statistical framework for distance estimation under an assumed evolutionary process [15]. Such models can improve phylogenetic inference in settings where recurrent or overlapping mutational events are common [16, 17]. For classical substitution models these model-corrected distances are *additive*: the distance between any two taxa equals the sum of the edge lengths along the path joining them in the tree, the property that distance-based reconstruction relies on. An uncorrected raw distance may instead deviate from additivity and bias the resulting trees [16, 17]. Among copy-number methods, SCONE2 [13] is, to our knowledge, the only one that attempts an analogous correction, implementing a probabilistic model of copy-number evolution and using it to estimate a triplet of distance measures between two sequences: t_1 (distance from the root to the Least Common Ancestor, or LCA), and t_2, t_3 (distances from the LCA to the two leaves). Unlike the classical case, however, these model-based estimates are not guaranteed to be additive on the copy-number tree; we make this distinction precise in Appendix A.1. Despite this rich output, it is assumed that the tree reconstruction should proceed by building a distance matrix with $t_2 + t_3$ as the leaf-to-leaf distance and then applying a standard tree reconstruction algorithm, such as Neighbor Joining (NJ) [18]. This approach disregards t_1 , the root-to-LCA distance, which carries valuable information about the divergence of the two leaves from the known root and can therefore be used to improve tree reconstruction.

Standard NJ reconstructs unrooted trees from leaf-to-leaf distances and therefore it does not utilize the known ancestral root. This is a significant limitation in cancer genomics, as the healthy diploid state provides a natural root for the tumor phylogeny. While root information can be incorporated by treating the root as an outgroup, this may lead to errors when the outgroup is distant [19–21]. Specialized algorithms exist that incorporate root information directly: Asymmetric NJ (ANJ) [22] uses distances from leaves to the LCA, while Deepest Least Common Ancestor Neighbor Joining (DLCA–NJ) [23] utilizes root-to-LCA distances. We find that DLCA–NJ, although not previously used for cancer phylogenetics, is better suited to this context than standard NJ. It constructs the tree directly from the root using the root-to-LCA distance, which corresponds exactly to the t_1 distance estimated by

SCONCE2. This makes DLCA–NJ a natural choice for tree reconstruction from copy-number data.

As data availability increases in both sample size n and sequence length m (i.e., the number of genomic bins), the estimation of the full pairwise distance matrix becomes the primary computational bottleneck. Let $O(f(m))$ be the time complexity of computing the distance between a single pair of sequences. While simple distance estimators are linear in m , modern probabilistic models explicitly account for hidden evolutionary events to increase accuracy, at the cost of substantially higher per-pair complexity. For SCONCE2, for example, computing a single pairwise distance requires $O(mK^4)$ time, where K is the maximum copy-number state (typically set to 10), making the full $O(n^2mK^4)$ matrix computation the dominant bottleneck regardless of the relative sizes of m and n [13, 24].

In the specific context of copy-number evolution, current methods primarily mitigate these computational costs by segmenting genomic bins through breakpoint inference [9, 10, 12]. This reduces the effective dimension of m , although the benefit may be limited when noisy or highly fragmented copy-number profiles lead to the inference of many spurious breakpoints, which inflate the effective dimension and propagate calling error into the distance estimates [14]. In contrast, Sparse Neighbor Joining (SNJ) [24] avoids the estimation of all n^2 pairwise distances by identifying optimal merges while computing only a logarithmic fraction of the distance matrix, maintaining theoretical guarantees with a reduced complexity of $O(f(m)n \log n)$.

In this work, we demonstrate that replacing traditional NJ with DLCA–NJ significantly improves reconstruction accuracy for methods that use model-based distances such as those from SCONCE2, with the largest gains in root placement metrics. Furthermore, we adapt the SNJ approach to the rooted setting, deriving the *Sparse Rooted Neighbor Joining* (SRNJ). This algorithm extends the DLCA framework to large datasets by avoiding full matrix computation, achieving a worst-case time complexity of $O(f(m)n \log n)$. The remainder of this paper is organized as follows. In Section 2, we provide the theoretical grounds for rooted tree reconstruction and introduce the SRNJ method and its properties. In Section 3, we present benchmarking experiments on synthetic and real datasets to evaluate the accuracy and execution time of the proposed methods, followed by a discussion of the results and current limitations. Finally, we conclude with a summary of our findings and future research directions in Section 4.

2 Materials and Methods

2.1 Deepest-LCA Neighbor Joining

In classical phylogenetics, distance-based algorithms reconstruct a tree topology T from a non-negative distance matrix D defined over n observations referred to as *taxa* (singular: taxon). These taxa may represent any distinct biological units or data profiles, such as genomic sequences or read-count profiles. Recall from the introduction that a distance matrix is *additive* when each entry equals the sum of the edge lengths along the path connecting the two taxa in a tree. The most widely used tree reconstruction algorithm is Neighbor Joining [18], which iteratively merges pairs of taxa that minimize the Q criterion defined as

$$Q(i, j) = (n - 2)D(i, j) - \sum_k D(i, k) - \sum_k D(j, k), \quad (1)$$

where $D(i, j)$ is the distance between taxa i and j . If D is additive for a tree T , then NJ reconstructs T exactly. Since in practice most distance matrices are noisy estimates of the underlying additive matrix, research has characterized the robustness of NJ to errors in the distance matrix. In particular, it has been shown that NJ always reconstructs the true tree if the error in the distance matrix is less than half of the minimum edge length in the tree, or in other words, the l_∞ -radius of NJ is $\frac{1}{2}$. Another crucial result is that NJ correctly reconstructs all edges in the true tree T that are longer than 4 times the maximum error in the distance matrix; NJ is said to have *edge* l_∞ -radius of $\frac{1}{4}$ [25, 26].

While in classical phylogenetics the ancestral sequence at the root is typically unknown, cancer phylogenetics offers a distinct advantage. Because tumor cells evolve from a healthy diploid genome, the root of the tree is effectively known: it corresponds to the diploid state, i.e., a uniform copy number of 2 across all genomic loci. This property allows distance estimation methods to calculate the distance from the root to each observed taxon, providing an additional source of information beyond just the distances between taxa. Given a fixed root, the distance from the root to the LCA of two taxa provides an alternative measure to the leaf-to-leaf distance. Following the terminology in [23], we refer to this distance as the LCA distance. Let L be the matrix consisting of estimated root-to-LCA distances for each pair of taxa. Then one can show that if D is additive for a tree T with root r , then the matrix L defined as

$$L(i, j) = \frac{1}{2}(D(i, r) + D(j, r) - D(i, j)), \quad (2)$$

consists of the true distances from the root to the LCAs in T . However, LCA distances can also be estimated directly from triplets of sequences, using, e.g., maximum likelihood techniques, potentially leading to more accurate estimates [27, 28]. More generally, we can say that any non-negative matrix L over a set S is an LCA-matrix if $L(i, i) = \max_{j \in S} L(i, j)$ for all $i \in S$ and $L(i, j) \geq \min(L(i, k), L(j, k))$ for all $i, j, k \in S$ (three-point condition). The DLCA–NJ algorithm reconstructs a rooted tree from an LCA-matrix L by iteratively merging pairs of taxa with the maximum (deepest) LCA distance, hence its name. After a merge, the LCA-matrix is updated by means of a reduction formula which has to be defined in a way that preserves the additivity. For example, any reduction formula of the form

$$L(v, k) \leftarrow \alpha L(i, k) + (1 - \alpha) L(j, k), \quad (3)$$

where v is the new node obtained by merging i and j , and with $\alpha \in [0, 1]$, maintains the additivity of the LCA-matrix. Notably, DLCA–NJ has been shown to be more robust than NJ in terms of edge ℓ_∞ -radius, with an optimal radius of $\frac{1}{2}$ instead of the $\frac{1}{4}$ of NJ [23]. Although this property does not mean that DLCA–NJ is more accurate than NJ in general terms, it suggests that exploiting the known root can lead to improved reconstruction of short edges in phylogenies from copy-number data.

2.2 Sparse Rooted NJ

Standard DLCA–NJ, like NJ, requires the prior computation of the full $n \times n$ LCA distance matrix and runs in $O(n^3)$ time. However, the cost of computing the distance matrix typically dominates the overall running time of the sequence-based method. To mitigate this, we adapt the Sparse NJ framework [24] to the rooted case, deriving SRNJ.

SRNJ bypasses the full matrix computation by selecting taxa to compare based on the current state of the reconstructed tree. The algorithm starts by building a rooted backbone tree using DLCA–NJ on a subset of k_0 taxa, small enough such that the full LCA distance matrix can be computed in $O(f(m)n \log n)$ time (e.g., $k_0 = \sqrt{n \log n}$). Then, it iteratively finds the edges where to insert the remaining leaves by exploring the tree and calculating the distances as needed. This strategy ensures that only $O(n \log n)$ distances are computed, while preserving the consistency of DLCA–NJ.

2.2.1 Overview of SRNJ

Let T^{k_0} be the rooted binary backbone tree built on a subset of $k_0 \geq 2$ leaves for which the full LCA distance matrix is computed. At each iteration $k = k_0 + 1, \dots, n$, a new leaf l_k is added to the current tree T^{k-1} by finding the correct edge for its insertion. The algorithm first identifies a *centroid* node in T^{k-1} , chosen so that, if removed, the largest connected component has at

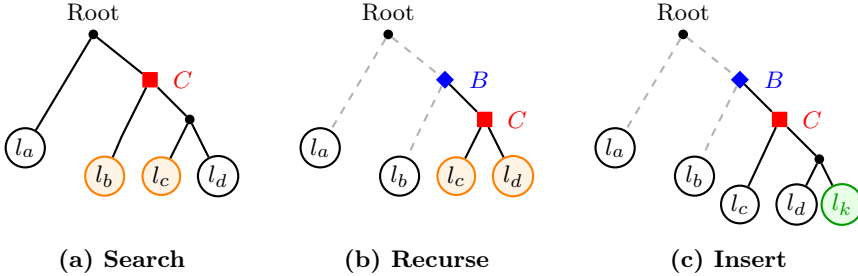


Fig. 1 SRNJ iteration steps. (a) A centroid C is chosen and orienting leaves l_b, l_c are selected. (b) After comparing them with l_k , the algorithm recurses into the l_c subtree; the old centroid becomes a barrier B and a new C is chosen, with l_c, l_d as orienting leaves. (c) The recursion continues towards the l_d subtree, at which point, with only one leaf left, l_k is inserted on the edge connecting C to l_d .

most half of the $k - 1$ leaves. The existence of such a node is ensured by the Centroid Decomposition Theorem [29]. Then, two *orienting* leaves, l_a, l_b , are selected at random, one from each of the two subtrees below the centroid, i.e., following the edges pointing away from the root. The LCA distance between the new leaf l_k and each of the two orienting leaves is computed, and the DLCA–NJ tree among these three leaves is built to determine which of the three components to explore next. Two cases can occur: either l_k forms a cherry with one of the orienting leaves, or l_a, l_b form a cherry and l_k is placed as sibling of their parent. In the first case, l_k appears to be closer to one of the orienting leaves than to the other, and therefore the algorithm explores the component containing the orienting leaf that forms the cherry, while in the second case, it explores the component that contains the root. In both cases, the current centroid becomes a *barrier* node. A barrier node isolates the edges remaining to be explored. While the leaves beyond this node remain available as potential orienting leaves, the barrier effectively represents them during the next centroid selection step, ensuring that the algorithm explores the tree in a balanced way, i.e., it only counts as one leaf for the purpose of selecting the next centroid. Since at each step the algorithm recurses into a component with fewer than half the current leaves, the search depth is at most $\log_2(k - 1)$ for a tree of $k - 1$ leaves, yielding the $O(\log n)$ distance computations per insertion. Whenever the recursion is called on a subtree that only contains one leaf, one barrier or the root node, the new leaf l_k is inserted on the edge connecting such node to the current centroid. While the SRNJ algorithm focuses exclusively on recovering the tree topology, edge lengths can be estimated in a subsequent post-processing step using standard distance-based optimization criteria such as ordinary or weighted least squares [15]. A schematic representation of the algorithm is shown in Figure 1.

2.2.2 Selection of orienting leaves

The selection of orienting leaves is a crucial step in the algorithm, as it determines the accuracy of the direction chosen at each recursion. When the distance $f(m)$ is expensive to compute (e.g., $O(mK^4)$ for SCONE2), we subsample $\log n$ candidates at random from each subtree and evaluate only those. Scoring $\log n$ candidates at each of the $O(\log n)$ recursion levels increases the total number of distance evaluations to $O(f(m)n \log^2 n)$, still substantially lower than the $O(f(m)n^2)$ full-matrix cost.

Alternatively, when the pairwise distance can be computed in $O(m)$ time with a small constant—for instance the Hamming distance on rounded copy-number profiles, or the NLL distance (see Appendix A.2)—all leaves in each subtree can be scored without materially increasing the overall cost, and no subsampling is needed.

For both regimes, the orienting leaf can be selected either as the candidate with the smallest leaf-to-leaf distance or, following the logic of DLCA–NJ, as the one with the largest estimated LCA depth. We evaluate these variants empirically in Section 3.4. Crucially, this heuristic changes only *which* candidates are scored, not the insertion rule applied afterwards: on an additive matrix any valid pair of orienting leaves yields the correct local topology, so subsampling preserves the consistency guarantee of Section 2.3.1 and affects only the constant and logarithmic factors in the running time, leaving the $O(f(m)n \log n)$ bound of the original sparse method unchanged. The cheap-proxy regime is likewise a scoring device only; the NLL distance it can use is defined in Appendix A.2.

2.3 Theoretical properties of SRNJ

2.3.1 Consistency

Here we use *consistency* in the sense of the reconstruction algorithm: a method is consistent if it recovers the true tree exactly whenever the input distance matrix is additive, regardless of the distance estimator used to produce it.

DLCA–NJ is consistent: if $D = D_T$ is additive for a tree T , the algorithm reconstructs T exactly. SRNJ invokes DLCA–NJ only on selected triplets of D . If D is additive, every submatrix is also additive, and DLCA–NJ returns the correct local topology at each insertion step. Hence each recursive subtree selection follows the true path toward the correct edge. By induction over the insertion order, every taxon is inserted at the correct edge, and SRNJ reconstructs T exactly. Therefore SRNJ is consistent.

2.3.2 Robustness

The greedy insertion strategy of SRNJ means it does not inherit the edge ℓ_∞ -radius guarantee of DLCA–NJ. At each insertion step only a local quartet is evaluated; errors on short edges can therefore compound across iterations in ways that a global algorithm avoids by averaging over the full distance matrix. As a consequence, short edges are more susceptible to misplacement by SRNJ than by DLCA–NJ. This limitation can be mitigated by considering multiple orienting leaves at each iteration, as described above, increasing the likelihood of selecting the correct subtree for exploration, and is empirically evaluated in Section 3.4.

2.3.3 Complexity Analysis

The time complexity of the algorithm is $O(f(m)n \log n)$, where $f(m)$ is the time complexity of computing the distance between two sequences of length m . In fact, the algorithm computes the full distance matrix for a subset of $\sqrt{n \log n}$ leaves to build the backbone tree, which takes $O(f(m)n \log n)$ time. Then, for each of the remaining $n - \sqrt{n \log n}$ leaves, it computes the distance to at most $2 \log n$ new leaves, which also takes $O(f(m)n \log n)$ time in total. While the iterative nature of the algorithm limits global parallelization compared to traditional Neighbor-Joining, this is more than compensated for by the reduction in total distance calculations from $O(n^2)$ to $O(n \log n)$. Parallelization can still be effectively applied to the backbone matrix construction and the independent distance evaluations required for each leaf insertion.

2.4 Evaluation metrics

To evaluate the accuracy of the reconstructed trees, we use four commonly used metrics in phylogenetics: the Robinson-Foulds (RF) distance [30], the quartet distance [31], the transfer distance [32] and the triplet distance [33], which is specifically designed for rooted trees. The RF distance measures the number of splits (bipartitions of the leaf set) that are present in one tree but not in the other. In particular, we use the normalized RF distance, which is the RF distance divided by the maximum possible RF distance for trees with the given number of leaves i.e.,

$$\text{RF}_{\text{norm}}(T_1, T_2) = \frac{\text{RF}(T_1, T_2)}{2n - 6}, \quad (4)$$

where n is the number of leaves in the trees. This allows us to compare the accuracy of the reconstructed trees across different datasets with varying number of leaves. The RF distance is widely used due to its simplicity and computational efficiency. However, it only measures whether splits are identical or not, and therefore does not capture how different two incompatible

splits are. As a consequence, small topological changes (e.g., moving a single taxon) can alter several splits and lead to a relatively large RF distance, while large rearrangements that affect the same set of splits are penalized equally. Moreover, because the number of splits grows linearly with the number of leaves, the RF distance often takes values close to its maximum for moderately large trees, reducing its ability to discriminate between different levels of topological error. The quartet distance, on the other hand, counts the number of quartets (sets of four leaves) whose induced topology differs between the two trees, reported as a fraction of all $\binom{n}{4}$ quartets. Because this fraction is taken over a number of quartets that grows much faster than the number of splits underlying the RF distance, it varies more smoothly and provides a more fine-grained comparison between trees with a large number of leaves [34]. The transfer distance [32] provides a more fine-grained comparison than the RF distance by measuring how similar individual edges are between two trees. For an edge b^* in the reference tree T_{ref} , the transfer index is defined as the minimum Hamming distance between the bipartition induced by b^* and any bipartition induced by an edge b in the estimated tree T_{est} . Intuitively, this quantity corresponds to the minimum number of leaves that must be moved from one side of the bipartition to the other to make an edge in T_{est} match b^* . To obtain a single score for the whole tree, we compute the transfer index for each internal edge of T_{ref} and report their arithmetic mean. Additionally, as a way of assessing the accuracy of root placement, we compute the transfer index only for the edge connecting the root to its children in T_{ref} and normalize it by the maximum number of leaves that can be transferred, which is the minimum number of leaves on either side of the bipartition minus one. We denote this metric the *root-split* distance. The triplet distance [33] (introduced as *triples* distance in the original paper) is the rooted analog of the quartet distance. It considers all $\binom{n}{3}$ subsets of three leaves and for each subset $\{i, j, k\}$, the metric determines if the rooted topological relationship is consistent between the reference and the estimated tree (e.g., $(i, (j, k)) \neq ((i, j), k)$). The triplet distance is defined as the number of such subsets with differing topologies, normalized by the total number of possible triplets. Because the hierarchical nesting of a triple is determined by the path to the root, this metric is highly sensitive to root misplacement, so—together with the root-split distance defined above—it serves as an indicator of root-placement accuracy. All distances except the transfer distance are always normalized to take values between 0 and 1, even if not specifically mentioned in the text. Tree distances were computed using standard implementations: Robinson-Foulds via *DendroPy* [35], quartet and triplet distances via *tqDist* [36], and transfer distance via *Booster* [37].

3 Results and Discussion

3.1 Benchmark on *in-house* simulated data

We benchmarked the algorithms on data generated with a simple *in-house* CN evolution model, which allows us to compute the LCA distances using the ground truth simulated CN sequences, including those of ancestral sequences, thereby avoiding CN calling errors. Trees are generated with a birth-death process (birth-rate=1) and the number of CNAs on each edge is sampled from a Poisson distribution with mean $\lambda = 5$. Each CNA event represents a gain or a loss (therefore changing the copy number by 1) of a segment whose start position is drawn uniformly over the bins and whose length is sampled from another Poisson with mean $\mu = 200$ bins. The number of bins is fixed to $m = 1000$ and the number of leaves varies in $N \in \{20, 50, 100, 200, 500\}$. An example of such simulated data is shown in Figure A1 in the Appendix. We run NJ, DLCA-NJ, SNJ and SRNJ on the same dataset and compare the reconstructed trees with the true tree using the metrics described above. The distances are computed from the ancestral CN profiles by means of a simple evolutionary model similar to the one introduced in [11] (see Appendix A.1 for details); for each pair of leaves, we compute three distances (root-to-LCA and the two LCA-to-leaf distances) from the known CN profiles. Using the known ancestral profiles makes this an idealized setting that isolates the reconstruction algorithm from distance-estimation error and thus provides an upper bound on achievable accuracy; the effect of estimating distances from data instead is examined in the subsequent experiments (Section 3.2). We note that the distance computation, while accounting for dependencies in the copy-number profiles, is a simplified evolutionary model (with the Markov property) and does not perfectly match the generative process used for simulation, which is biologically more realistic. The distances are therefore not guaranteed to be additive on the true tree, and as such, some residual reconstruction error is expected even when ancestral profiles are known exactly. We evaluate the Sparse Rooted NJ algorithm using three variants that differ in the size of the orienting-leaf candidate pool, selecting in all cases the candidate with the smallest leaf-to-leaf distance. SRNJ-1 uses a single random orienting leaf, and SRNJ-log samples $\log n$ random candidates. SRNJ-all scores its candidates—all inserted leaves—with a cheap surrogate distance rather than with the expensive estimator (e.g., SCONE2) used by SRNJ-log. Its full surrogate distance matrix is precomputed once, before SRNJ runs, at a cost of $\alpha m N^2$, which we assume to be far smaller than SRNJ’s running time. These variants allow us to assess the impact of the candidate-pool size on the accuracy of the reconstructed trees, and to evaluate the trade-off between accuracy and computational cost. A broader comparison of orienting leaf selection strategies, including proxy-distance approaches,

is presented in Section 3.4. For comparison, we also execute the original SNJ algorithm using the smallest leaf-to-leaf distance selection criterion on $\log n$ random candidates, as described in the original paper. Additionally, we include ANJ [22] and FastME [38] in the comparison for a more comprehensive benchmark. ANJ is included as another rooted NJ variant that uses leaf-to-LCA distances rather than root-to-LCA distances, allowing us to isolate the benefit of the specific LCA formulation used by DLCA–NJ. FastME combines NJ with heuristic tree search strategies to optimize the balanced minimum evolution criterion, without any guarantee of consistency, but with the same time complexity of NJ, making it a valid alternative to NJ. The results in Figure 2 show that DLCA–NJ outperforms NJ in the quartet, triplet and transfer distances across all dataset sizes, with the improvement being more pronounced for larger datasets. In terms of RF distance, DLCA–NJ is better than or comparable to NJ, with the gap closing for the largest datasets. FastME shows the best accuracy in terms of RF distance, but it is outperformed by DLCA–NJ in terms of quartet and triplet distance, which indicate a better placement of the root for the latter. This effect is even more evident in the root-split distance (i.e., accuracy of the root bipartition) shown in Figure 3, where only DLCA–NJ and the SRNJ variants reach perfect accuracy, while NJ, SNJ, ANJ and FastME misplace the root increasingly often as the number of leaves grows. SRNJ shows a significant improvement over its NJ counterpart (SNJ) with respect to all metrics, with a pronounced improvement in RF and transfer distance when more than a single orienting-leaf candidate is used (SRNJ-log and SRNJ-all). Compared to DLCA–NJ, with larger datasets, SRNJ shows particularly lower performance in terms of RF distance. This might be due to a combination of the greedy insertion strategy and the limited resolution of the RF distance, as the other metrics do not show such a pronounced drop in performance. Notably, the performance of SRNJ in terms of quartet and triplet distance is inferior only to that of DLCA–NJ, despite the greedy insertion. This suggests that SRNJ effectively recovers local nested relationships and sub-clade structures better than the other algorithms not based on LCA distances, even if some global topological errors are introduced by the greedy insertion strategy.

3.2 Realistic CNA simulation and distance estimation

For a more realistic benchmark where CN sequences are not known, we evaluated the performance of DLCA–NJ and SRNJ in combination with distances estimated via the state-of-the-art methods SCONCE2 and MEDICC2. We simulated datasets of different sizes, with $N \in \{50, 100\}$ leaves, and fixed sequence length of $m = 1000$ genomic bins, equally divided into 10 chromosomes. A dataset consists of a tree topology and a set of sequences at the leaves, where each sequence contains the read-counts for each genomic bin

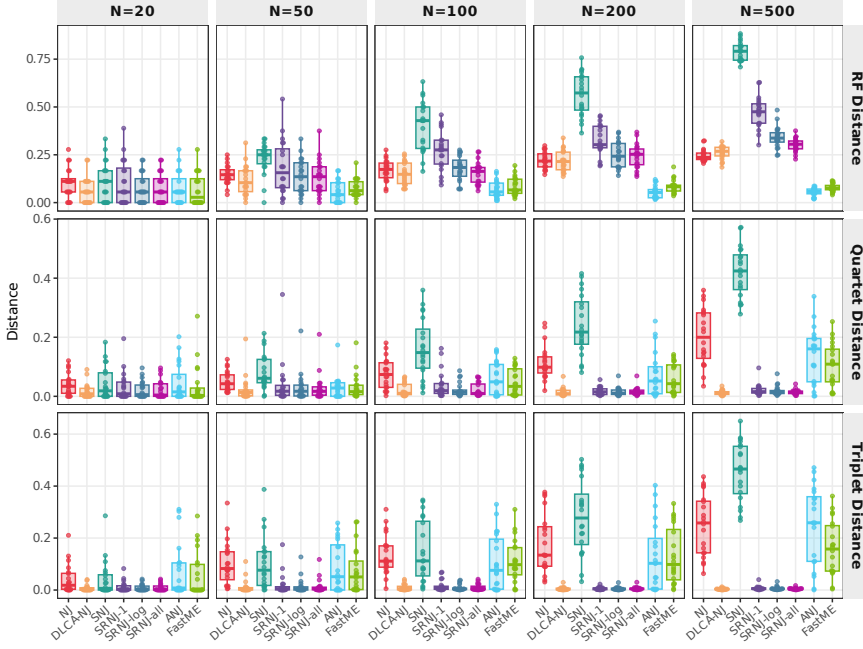


Fig. 2 Reconstruction accuracy on *in-house* simulated data. Boxplot aggregate scores over 20 different data seeds, for $N \in \{20, 50, 100, 200, 500\}$ leaves (columns) and the RF, quartet and triplet distances (rows; lower is better). SRNJ is executed in three variants that all select the orienting leaf with the smallest leaf-to-leaf distance but differ in the candidate-pool size: a single random leaf (SRNJ-1), log n random candidates (SRNJ-log), or all inserted leaves with NLL proxy (SRNJ-all).

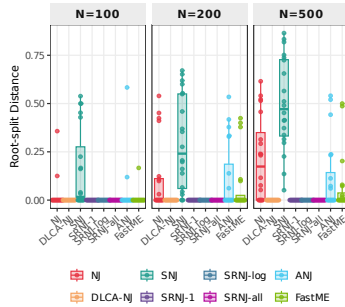


Fig. 3 Root placement accuracy on *in-house* simulated data. Boxplots showing the *root-split* distance in the experiment with *in-house* simulated data. Distances are computed from ground-truth ancestral copy-number profiles using the JC-breakpoint correction (Appendix A.1).

and is generated according to the CN profile of the corresponding leaf. The simulated datasets were generated with CNAsim [39], a simulator of copy-number evolution that allows us to generate trees with varying numbers of leaves and mutation rates. We set the average number of CNAs per edge to $\lambda \in \{2, 3\}$, which is a realistic scenario for cancer evolution, especially in the case of punctuated evolution, where a single cell can carry from many tens to hundreds of CNAs [3, 40].

We repeated the simulation for 20 replicates for each dataset size and ran SCONE2 (with maximum copy-number state $K = 10$) on each dataset to estimate the t_1 , t_2 and t_3 distances, which were then used as input for NJ and DLCA–NJ. Like in the original SCONE2 paper, we summed the t_2 and t_3 edge lengths to form a standard leaf-to-leaf distance matrix for NJ, while we adopted the t_1 edge lengths as the root-to-LCA distances for DLCA–NJ. We include MEDICC2 in the comparison to assess the impact of using a model-based distance measure instead of the MED distance, which is not derived from an explicit probabilistic model of evolution. Since MEDICC2 requires copy-number profiles as input, we used the total copy numbers inferred by SCONE2 for a fair comparison, and reconstructed the tree with NJ with outgroup rooting, as in the original MEDICC2 paper. We report the distribution of the quartet distance, RF distance, transfer distance and root-split distance between the true tree and the reconstructed trees for the different combinations of distance estimation methods and tree reconstruction algorithms in Figure 4(b) (see Figure A2 in the Appendix for a complete overview). Also, we report the per-replicate difference $\Delta = \text{SCONE2} - \text{MEDICC2}$ for each reconstruction method in Figure 4(a); negative values indicate better performance with SCONE2 distances.

We find that SCONE2 distance estimates lead to more accurate trees than MEDICC2 distances: paired with DLCA–NJ, SCONE2 distances significantly reduce the reconstruction error over MEDICC2 distances. On these inferred distances, however, the margin of DLCA–NJ over the plain NJ ordering is small and not statistically significant; the algorithmic benefit of root-aware ordering is isolated on exact distances, in the in-house benchmark and in the ground-truth control (Appendix A.5). In absolute terms, DLCA–NJ with SCONE2 distances achieves the best performance, indicating the need for both a model-based distance estimation and a tree reconstruction algorithm that can leverage the root information. We also ran the three versions of Sparse Rooted NJ described in the previous experiment. We find that SRNJ achieves its best performance with the larger candidate pools (SRNJ-log and SRNJ-all, which are close to each other) and, although slightly less accurate than the dense DLCA–NJ, it inherits the same benefit of the model-based SCONE2 distances over MED distances and improves on its own sparse baseline SNJ when considering the root-split distance;

the single-candidate SRNJ-1 is consistently the weakest variant. On these inferred distances the SRNJ-over-SNJ margin is modest — the full advantage is recovered once copy-number-calling error is removed, as shown by the ground-truth control in Appendix A.5. Due to the long execution time of SCONCE2 (see Section 3.3), we were not able to run the experiment with larger datasets, but we expect the benefits of DLCA–NJ and SRNJ to be even more pronounced for larger trees, as shown in the previous experiment with *in-house* simulated data.

3.3 SRNJ running time

SRNJ is a sequence-based method that takes sequences as input and relies on the distance estimation procedure as a subroutine to compute distances for individual pairs only when needed. However, neither MEDICC2 nor SCONCE2 software implementations support pairwise distance queries. For the previous experiments we bypassed this issue by computing the full distance matrix and then only using the required distances for SRNJ, but this approach does not allow to evaluate the actual running time of SRNJ. Therefore, to compare the scalability of the different algorithms, we emulate the total running time by tracking the number of required distance evaluations when executing SNJ and SRNJ, and we multiply it by the empirical time required by SCONCE2 to compute a single distance. For this purpose, we timed the execution of SCONCE2 in the previous experiment for each dataset with $N = 100$ and estimated the mean time required per pairwise distance computation as the total time divided by the number of pairs (i.e., $n(n - 1)/2$), obtaining 220.5 ± 66.5 seconds (mean $\pm$ SD over 20 replicates). This approach allows for a comparison of the running times shown in Figure 5, directly highlighting how the reduced number of distance computations in SRNJ translates to significant gains in total core-hours.

3.4 Effect of orienting leaf selection strategy

We evaluate how the choice of orienting leaf selection affects the accuracy of SRNJ across tree sizes $N \in \{20, 50, 100, 200, 500\}$ using CNAsim-generated data (20 seeds per condition, $\lambda = 5$). Beyond the two built-in SRNJ criteria (min d and max LCA), we consider three additional families of selection strategies. The *ground-truth* (GT) oracle selects orienting leaves using the exact LCA distances computed from known ancestral sequences, providing an upper bound on the accuracy achievable by improved selection. The *Hamming* and *NLL* proxies are cheap surrogates derived from observed copy-number profiles and a healthy-cell baseline, requiring no additional tree inference steps.

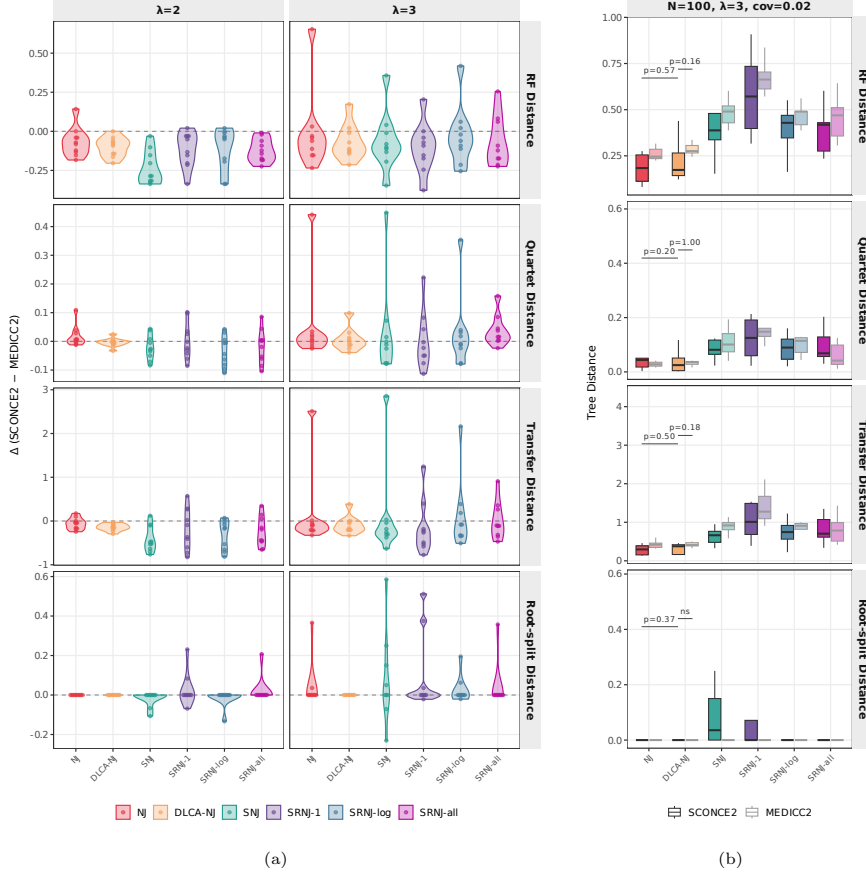


Fig. 4 Accuracy with SCONCE2 and MEDICC2 distances on CNAsim data. Aggregated results of the experiment with synthetic data generated with CNAsim (20 replicates) and distances estimated with SCONCE2 and MEDICC2. Panel (a) shows the per-replicate difference $\Delta = \text{SCONCE2} - \text{MEDICC2}$ for each tree reconstruction method (NJ, DLCA-NJ, SNJ, SRNJ-1, SRNJ-log, SRNJ-all) on simulated datasets of $N = 100$ cells, across mutation rates $\lambda \in \{2, 3\}$. Rows show (top to bottom) RF distance, quartet distance, transfer distance and root-split distance between the true tree and the reconstructed tree. Negative values indicate better performance by SCONCE2. Panel (b) shows the tree accuracy for $\lambda = 3$, with SCONCE2 (opaque) and MEDICC2 (transparent) compared side by side. Brackets indicate paired Wilcoxon tests p-values. The three SRNJ variants all select the orienting leaf with the smallest leaf-to-leaf distance and differ in the candidate pool: SRNJ-1 (single random leaf), SRNJ-log (log n random candidates) and SRNJ-all (all inserted leaves).

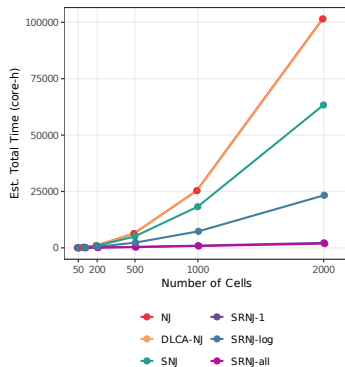


Fig. 5 Estimated running time in core-hours. Estimated running time in core-hours for the different algorithms on simulated datasets of increasing size (N up to 2000 cells), obtained by multiplying the number of expensive (SCONCE2) pairwise distance evaluations by the measured per-pair cost. NJ and DLCA–NJ require the full $O(N^2)$ distance matrix; SNJ and the SRNJ variants query far fewer pairs. SRNJ-all scores all candidate orienting leaves with a precomputed cheap distance matrix (computed once, $O(N^2)$ but with a small constant and not requiring SCONCE2), so its number of expensive distance evaluations matches that of SRNJ-1 rather than growing with the candidate-pool size.

Figure 6 shows that the choice of orienting leaf selection strategy has a pronounced effect on RF distance, especially for larger trees, while quartet and triplet distances are less sensitive to this choice. Among the SRNJ default strategies, the min- d criterion consistently outperforms max-LCA across all tree sizes and metrics. The GT oracle demonstrates that improving the selection of orienting leaves beyond the default SRNJ rules can substantially reduce the RF distance, approaching the accuracy of full DLCA–NJ at larger N . Notably, the Hamming and NLL proxy strategies achieve accuracy comparable to the GT oracle for RF distance, and both consistently outperform the default SRNJ sampling strategies (SRNJ-1, SRNJ-log, SRNJ-all), suggesting that scoring all candidate leaves with a numerically efficient distance is more effective than sampling a logarithmic subset with the exact distance. For quartet and triplet distances, all SRNJ strategies perform similarly, and the differences between selection criteria are small relative to the gap between SRNJ and DLCA–NJ. These results indicate that better orienting-leaf selection primarily reduces global topological errors (captured by RF distance), while local branching relationships are well recovered regardless of the strategy.

We note that on these CNAsim datasets the pairwise SNJ baseline performs comparably to the SRNJ variants, in contrast to the clear separation observed on the in-house data. As we show in Appendix A.5, this reflects the limited reliability of the SCONCE2-inferred distances rather than a genuine parity of the methods: on the very same CNAsim datasets, evaluated with

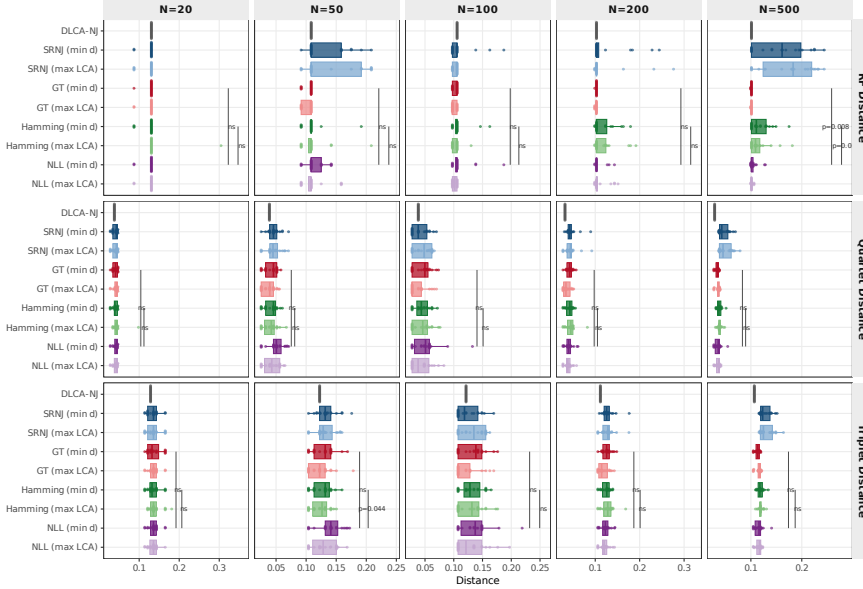


Fig. 6 Orienting-leaf selection strategies on CNAsim data. Horizontal box plots (with individual seed points) of three tree-distance metrics (RF, quartet, and triplet distance) for nine orienting-leaf selection strategies applied to SRNJ, faceted by number of cells ($N \in \{20, 50, 100, 200, 500\}$, 20 seeds each). Strategies are grouped by distance source: baseline DLCA–NJ (grey), SRNJ defaults (blue), ground-truth oracle (red), Hamming proxy (green), and NLL proxy (purple). Within each group, two selection criteria are compared: *min-d* selects the orienting leaf with the smallest pairwise cell-to-cell distance, while *max-LCA* follows the DLCA–NJ logic and selects the leaf whose LCA with the new taxon is estimated to be deepest (i.e., most distant from the root), using the available distance as a proxy. Lower tree distance indicates better reconstruction.

ground-truth distances, the root-aware SRNJ again reconstructs the trees far more accurately than SNJ. The in-house benchmark, where distances are exact, is therefore the setting that isolates the algorithmic contribution of root information.

3.5 Breast cancer dataset

To evaluate tree reconstruction accuracy on real biological data, we apply DLCA–NJ, SRNJ and NJ to a breast cancer single-cell DNA dataset [41]. We compare the results with the clonal analysis reported in [42]. The dataset includes 10,000 cells from four tumor regions and one normal region. Due to the prohibitively high computational cost of running SCONE2 on the full set, we focus on a subset of 500 cells from tumor region E, with bins of size 5Mb ($m = 620$). These cells were sampled so as to match the proportions of the eight clones (J–I to J–VIII) identified in the original CHISEL analysis.

We use the normal region as diploid pseudo-bulk required for SCONE2 distance estimation, and we set the maximum ploidy to $K = 8$. As a control, we compute a MEDICC2 distance matrix using the allele-specific copy-number profiles from CHISEL.

We evaluate how well the tree clades represent the clones inferred by CHISEL using the *clade-F1* score. For each clone, this metric identifies the clade with the highest F1 score and then computes the average of these scores across all clones. These clones are not a known ground truth but the output of the original CHISEL analysis, so the clade-F1 score should be read as agreement with a reference partition rather than with a true clonal structure. We run 50 bootstrap samples of 200 cells each from our 500-cell subset to quantify the sampling variability of this agreement across methods; this controls for variability due to cell subsampling but not for any systematic error in the reference clustering itself, which remains a limitation of the real-data evaluation.

As shown in Figure 7, the allele-specific MEDICC2 distances yield substantially higher clade-F1 scores than SCONE2 for every reconstruction method, a pattern consistent with reconstruction accuracy on this dataset being constrained by the inferred distances—which combine copy-number calling and branch-length estimation—rather than by the tree-building algorithm; separating these two contributions would require further experiments beyond the scope of this work. Within each distance source the reconstruction methods are close. With SCONE2 distances, DLCA–NJ shows a small improvement over NJ that does not reach statistical significance (paired Wilcoxon $p > 0.1$), and with MEDICC2 the two are essentially indistinguishable, reflecting the previous simulation experiment where the benefit of DLCA–NJ was more pronounced with model-based distances. The SRNJ algorithm falls slightly below the dense NJ and DLCA–NJ; the small F1 gap (of the order of 0.05) is significant but consistent with trading a small amount of accuracy for significant gains in scalability (Figure 7). We note that the clade-F1 score is insensitive to rooting and fine topology, the aspects where SRNJ’s root-aware reconstruction is most advantageous on ground truth, so this real-data metric does not reward exactly those strengths.

3.6 Discussion

Taken together, these results highlight a gap between classical phylogenetics and the current copy-number evolution literature regarding the use of model-based distances. While corrected distances are standard in traditional phylogenetics, they remain uncommon in CNA studies. Our experiments suggest that model-based distance estimates, such as those provided by SCONE2, can improve reconstruction accuracy, particularly under higher mutation rates and when combined with root-aware algorithms. Intuitively, a probabilistic model can account for hidden or overlapping events that a

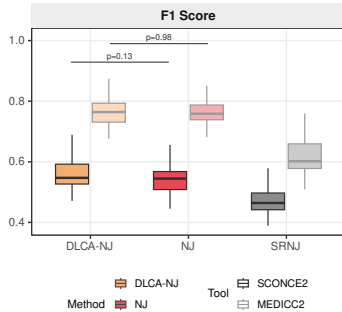


Fig. 7 Clade-F1 score on the 10x breast cancer dataset. Boxplots showing the distribution of the clade-F1 score across 50 different bootstrap samples of 200 cells from region E of the 10x Genomics breast cancer dataset. The clade-F1 score is computed by comparing the clades in the reconstructed tree with the clones inferred in the original CHISEL analysis. Brackets indicate paired Wilcoxon test p-values.

raw event count misses—the same rationale that motivates classical distance corrections—so these estimates move in the direction of a proper correction, even though in the copy-number setting their additivity is not established. The behaviour of SRNJ with accurate distances, as shown in the in-house benchmark, further suggests that rooted, iterative approaches preserve local branching orders and nested relationships more effectively than unrooted global methods.

A consideration that runs through these experiments is our reliance on SCONE2 for copy-number calling. We are effectively constrained to it because it is, to our knowledge, the only method that natively outputs the decomposed root-to-LCA (t_1) and leaf-to-LCA (t_2, t_3) distances that DLCA-NJ and SRNJ require; most callers provide only the copy-number profiles, from which these ancestral distances would still have to be estimated. The algorithms themselves are agnostic to the caller and would work with any future method that supplies, or permits the estimation of, these distances. We have therefore established the advantage of root-aware reconstruction on ground-truth and controlled simulations, where the distances are known or reliably estimated, and we make no claim of such an advantage in the real-data regime, where the quality of copy-number inference dominates. Whether more accurate callers would expose the same root-aware advantage seen in simulation is an important open question that our experiments cannot settle.

4 Conclusion

In this work, we have demonstrated that incorporating root information through LCA distances improves phylogenetic tree reconstruction for CNA

data. While the DLCA–NJ framework itself is not a contribution of this work, we established its specific suitability for CNA analysis, where the root is fixed and known. Building on it, we derived a novel algorithm, Sparse Rooted Neighbor Joining, which reduces the time complexity to $O(f(m)n \log n)$, where $f(m)$ is the cost of a single distance estimation. We proved the consistency of SRNJ and evaluated its accuracy and robustness on both synthetic and real single-cell copy-number data.

The main limitation of SRNJ is practical rather than theoretical: existing distance estimation methods for copy-number evolution do not expose pairwise queries—they compute a full distance matrix at once rather than returning the distance of a single pair on demand, which is the access pattern SRNJ needs to realise its complexity advantage. Making these estimators more modular is therefore a practical prerequisite. A second limitation is that, on real data, reconstruction accuracy appears to be constrained by distance estimation (copy-number calling together with branch-length estimation) rather than by the reconstruction algorithm.

These limitations point to the natural directions for future work: adapting model-based distance estimation methods to answer individual pairwise queries, and improving the selection of orienting leaves through more advanced heuristics. More broadly, we hope this work encourages the development of probabilistic, model-based distance estimators such as SCONCE2. Their cost per pair has been a deterrent to their adoption, but that cost is multiplied by a quadratic number of pairs only for dense algorithms; under SRNJ the number of required estimates grows as $O(n \log n)$, so an estimator that is expensive but accurate becomes affordable at scales where computing a full distance matrix would not be. We believe this shifts the trade-off in favour of investing in better distance models, and provides guidance for future approaches to copy-number evolution and tumor phylogenetics.

Declarations

Ethics approval and consent to participate. Not applicable.

Consent for publication. Not applicable.

Availability of data and materials. The dataset supporting the conclusions of this article is publicly available from 10x Genomics [41], at https://cf.10xgenomics.com/samples/cell-dna/1.1.0/breast_tissue_aggr_10k/breast_tissue_aggr_10k_web_summary.html. All code required to reproduce the experimental results reported here is available in the SRNJ repository, <https://github.com/Lagergren-Lab/srnj>.

- Project name: SRNJ (Sparse Rooted Neighbor Joining)
- Project home page: <https://github.com/Lagergren-Lab/srnj>
- Archived version: <https://doi.org/10.5281/zenodo.21398124>

- Operating system(s): Platform independent
- Programming language: Python
- Other requirements: see the repository README for dependency versions
- License: GNU General Public License v3.0 (29 June 2007)
- Any restrictions to use by non-academics: None

Competing interests. The authors declare that they have no competing interests.

Funding. No funding was received for this work.

Authors’ contributions. VZ developed the method, implemented the software, designed and ran the experiments, analysed the results and wrote the manuscript. HM took part in project discussions and contributed ideas to the method. AH designed and ran the experiments comparing the orienting-leaf selection strategies. JL conceptualised and supervised the study. All authors read and approved the final manuscript.

Acknowledgements. This work used computational resources provided by the National Academic Infrastructure for Supercomputing in Sweden (NAISS), partially funded by the Swedish Research Council through grant agreement no. 2022-06725.

Appendix A Supplementary Material

A.1 Simulation of *in-house* data

The *in-house* datasets used throughout the benchmark are generated by a single segment-based model of copy-number evolution. A cell lineage tree is drawn from a birth–death process (birth-rate 1), and each edge receives a number of CNA events sampled from a Poisson distribution with mean $\lambda = 5$, independently of the branch length. Each event is a gain or loss (changing the copy number by ± 1) over a contiguous segment whose length is sampled from a Poisson with mean $\mu = 200$ bins, starting from a diploid root over $m = 1000$ bins. Because every edge receives a comparable number of events, internal (shared) edges accumulate substantial ancestral signal, which is exactly what the root-aware methods exploit. Figures 2, 3 and A3 all use this same generative model.

We favour this in-house model over CNAsim for two reasons. First, it exposes the exact ancestral copy-number profiles at every internal node, so the root-to-LCA distances can be computed from the ground truth without any copy-number-calling error, isolating the behaviour of the tree-reconstruction algorithms. Second, it lets us control directly how copy-number evolution is distributed between internal (shared) and terminal (private) edges — the knob varied in the event-sharing experiment

(Appendix A.4) — which CNAsim does not expose. CNAsim is instead used for the realistic distance-estimation experiments, where copy numbers are called from simulated reads.

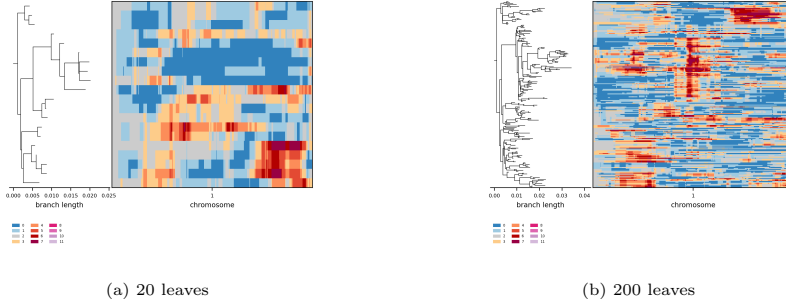


Fig. A1 Example *in-house* simulated datasets. Example of two simulated datasets with the *in-house* model, featuring 20 (a) and 200 (b) leaves respectively.

Evolutionary model for distance correction

We first fix terminology, following classical phylogenetics. A *raw* distance is computed directly from the observed profiles. A *model-corrected* distance replaces that raw dissimilarity by an estimate obtained under a probabilistic model: given observations x and y and a family of distributions $\{P_d\}$ indexed by a distance parameter d , it is the maximum-likelihood estimate $\hat{d}(x, y) = \arg \max_d P_d(x, y)$. An *additive* distance is one that equals, for every pair, the sum of the edge lengths along the connecting path in some tree (the tree-metric property). Classical corrections, such as the Jukes–Cantor correction, are model-corrected *and* known to be additive. The distances used in this work—both the SCONCE2 estimates and the model below—are model-corrected in this statistical sense; we do not claim that they are additive on the true copy-number tree, and treat additivity as an open question.

To compute the corrected distances for the experiment with *in-house* simulated data, we use a simple evolutionary model of copy-number evolution similar to the one used in [11]. Specifically, we compute the mutation rate $\hat{p}$ over each edge in the tree as the number of breakpoints divided by the total number of bins. A breakpoint between two nodes u, v in the tree is detected over a pair of adjacent bins $(i, i + 1)$ if

$$C_{i+1}^u - C_i^u \neq C_{i+1}^v - C_i^v, \quad (\text{A1})$$

where C_i^u is the known copy number of node u at bin i . Then, we correct the distance taking into account the maximum allowed copy number K as

follows:

$$D(u, v) = -\frac{1}{K+1} \log \left(1 - \frac{K+1}{K} \hat{p} \right). \quad (\text{A2})$$

This correction is analogous to the Jukes-Cantor correction and easily derived by modeling the copy-number evolution as a continuous-time Markov process with $K+1$ states (from 0 to K) and equal transition rates between any pair of states.

A.2 Orienting-leaf selection

NLL distance

The NLL distance is a pairwise distance derived from how much each cell’s copy-number profile deviates from a healthy-cell baseline. Let $\mathbf{x}_1, \dots, \mathbf{x}_n$ be the normalised copy-number profiles of the tumor cells and $\mathbf{x}_{n+1}, \dots, \mathbf{x}_{n+m_0}$ those of m_0 normal cells. A per-bin Gaussian baseline is estimated as

$$\mu_b = \frac{1}{m_0} \sum_{k>n} x_k^b, \quad \sigma_b = \max(\text{std}_{k>n}(x_k^b), \varepsilon), \quad (\text{A3})$$

with $\varepsilon = 10^{-3}$. Each cell i is scored at each bin b by the negative log-likelihood (NLL) under $\mathcal{N}(\mu_b, \sigma_b^2)$:

$$\ell_i^b = \frac{1}{2} \left(\frac{x_i^b - \mu_b}{\sigma_b} \right)^2 + \frac{1}{2} \log(2\pi\sigma_b^2). \quad (\text{A4})$$

The pairwise NLL distance is then the mean absolute difference of these profiles:

$$D_{\text{NLL}}(i, j) = \frac{1}{B} \sum_{b=1}^B |\ell_i^b - \ell_j^b|. \quad (\text{A5})$$

For the max-LCA variant, a root-distance proxy $r_i = \frac{1}{B} \sum_b \ell_i^b$ is used together with the additive-tree LCA decomposition $C_{ij} = (r_i + r_j - D_{\text{NLL}}(i, j))/2$ to estimate LCA depths. All operations are $O(m)$ per pair with a small constant, making this distance suitable for scoring all candidate orienting leaves without a subsampling step.

A.3 Additional results on CNAsim data

Figure A2 reports the complete results of the CNAsim experiment of Section 3, of which Figure 4 shows a representative subset: it spans all five tree distances, the three dataset sizes ($N \in \{20, 50, 100\}$) and both mutation rates ($\lambda \in \{2, 3\}$), for every combination of distance estimator and reconstruction algorithm. The trends discussed in the main text hold across these conditions. Trees reconstructed from SCONE2 distances are generally more

accurate than those obtained from MED distances; the dense algorithms (NJ and DLCA–NJ) retain a margin over their sparse counterparts; and among the SRNJ variants the single-candidate SRNJ-1 is consistently the weakest, while SRNJ-log and SRNJ-all remain close to one another. Reconstruction error grows with the number of cells for all methods. Notably, the root-split distance is close to zero for nearly every method and condition, so on these inferred distances it does not separate the rooted from the unrooted algorithms; the rooting advantage is instead apparent where the distances are exact (Figures 3 and A3).

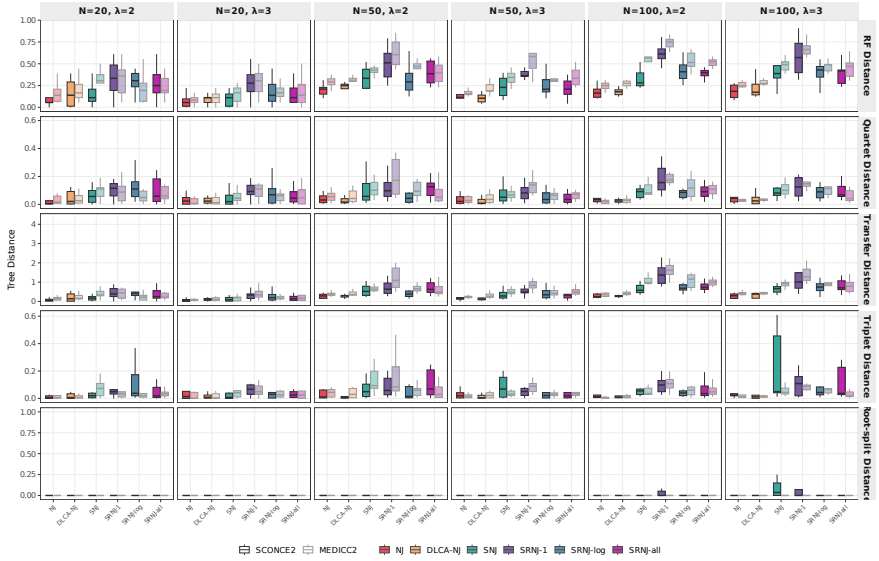


Fig. A2 Complete tree-distance results on CNAsim data. Boxplots of tree distances between the true tree and reconstructed trees on **CNAsim-simulated data** across all evaluated conditions. Rows show (top to bottom) RF, quartet, transfer, triplet and root-split distance. Columns correspond to combinations of dataset size ($N \in \{20, 50, 100\}$ cells) and mutation rate ($\lambda \in \{2, 3\}$). Each panel compares six tree reconstruction methods (NJ, DLCA–NJ, SNJ, SRNJ-1, SRNJ-log, SRNJ-all) applied to copy-number distances computed by SCONCE2 (opaque) and MEDICC2 (transparent). The SRNJ variants differ in the orienting-leaf candidate pool (single random leaf, log n random candidates, all inserted leaves). The sequencing coverage is $0.02\times$.

A.4 Effect of event sharing on reconstruction accuracy

The root-to-LCA formulation is expected to help most when copy-number evolution is dominated by early, shared events, so that internal (shared) edges carry strong ancestral signal, and to help least when evolution is mostly private to individual lineages. To probe this dependence directly, we vary

the amount of copy-number evolution placed on internal (shared) edges of the *in-house* simulated trees using the same segment model as the main benchmark (Appendix A.1): terminal edges keep the base rate $\lambda = 5$, while internal edges receive a Poisson number of events with a controlled mean λ_{int} that we sweep from 0 (star-like trees, evolution almost entirely private) upward ($N = 100$ cells, 20 replicates). Figure A3 reports the reconstruction accuracy as a function of the realised mean number of shared events per internal edge. When shared ancestral signal is scarce, all methods struggle to recover the internal structure; as the number of shared events grows, the LCA-based methods (DLCA–NJ and SRNJ) increasingly outperform NJ and SNJ, confirming that the root-to-LCA formulation is most beneficial in the shared-evolution regime typical of punctuated tumor clonal expansions. The separation is clearest in the triplet and root-split distances (the rooting-sensitive metrics), where DLCA–NJ and SRNJ approach zero error while SNJ remains high; SRNJ also stays well below SNJ in RF distance across the whole range.

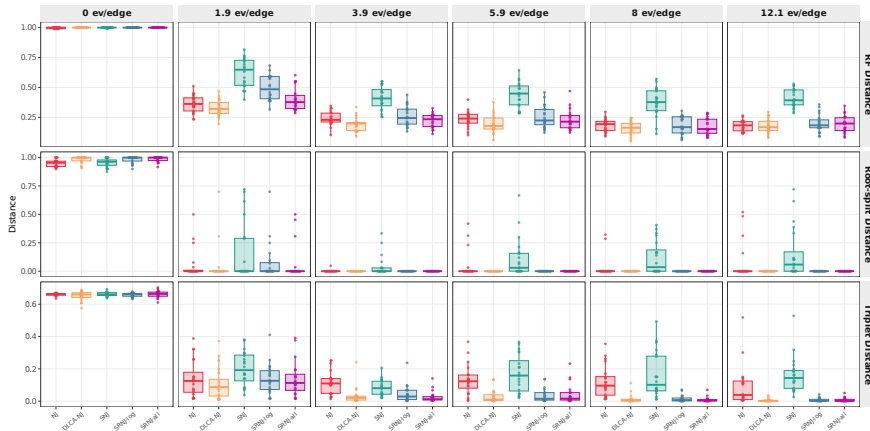


Fig. A3 Effect of shared ancestral events on reconstruction accuracy. Reconstruction accuracy on *in-house* simulated data (same segment model as Fig. 2) as a function of the realised mean number of shared copy-number events per internal edge (left: star-like trees with mostly private, terminal events; right: strongly clonal trees with early, shared bursts), for $N = 100$ cells over 20 replicates. Rows show RF, root-split and triplet distance; lower is better. As shared ancestral signal increases, the LCA-based methods (DLCA–NJ, SRNJ) increasingly outperform NJ and SNJ, most clearly in the rooting-sensitive triplet and root-split distances.

A.5 Simulation of CNAsim data

The realistic distance-estimation experiments (Figs 4 and A2) use datasets generated with CNAsim [39], which simulates read counts from a tumor

copy-number evolution model and therefore allows the copy numbers to be inferred (by SCONCE2 or MEDICC2/HMMcopy) rather than assumed known. Figure A4 shows two example datasets. Each dataset contains, in addition to the tumor cells, a set of normal diploid cells; the copy-number profiles combine focal and chromosome-arm-level events, together with a strong founder amplification shared by all tumor cells.

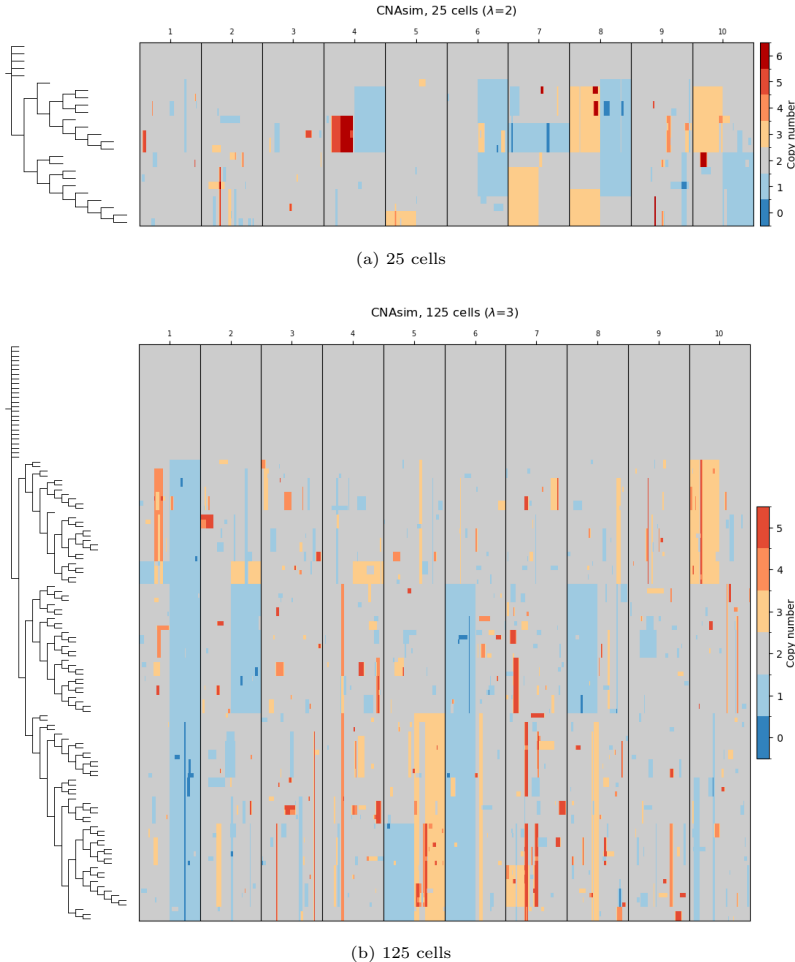


Fig. A4 Example CNAsim-simulated tumor datasets. Example CNAsim-simulated tumor datasets with 25 (a) and 125 (b) cells, of which 20 and 100 respectively are tumor cells and the remainder are normal diploid cells (the ground-truth cell tree on the left, inferred copy numbers as a heatmap over the ten chromosomes).

Ground-truth control.

On the in-house data the distances are computed from the known copy-number profiles and are therefore exact, whereas on the CNAsim data they are estimated by SCONE2 from noisy reads. To determine whether the smaller advantage of the root-aware methods on CNAsim data (Fig 4) originates from the CNAsim generative model or from copy-number-inference error, we re-evaluated the same CNAsim datasets using ground-truth distances computed from the true (simulated) ancestral profiles, exactly as for the in-house benchmark. With these exact distances DLCA–NJ and SRNJ reconstruct the CNAsim trees accurately ($RF \approx 0.07$ at $N = 100$), far better than the pairwise NJ and SNJ, mirroring the in-house result; once SCONE2-inferred distances are used instead their accuracy degrades sharply ($RF \approx 0.4\text{--}0.6$) and all methods become comparable. The muted CNAsim+SCONE2 result is therefore driven by copy-number-inference error rather than by the CNAsim trees, separating the contribution of the tree reconstruction algorithm from that of the copy-number caller.

References

- [1] Zahn, H., Steif, A., Laks, E., Eirew, P., VanInsberghe, M., Shah, S.P., Aparicio, S., Hansen, C.L.: Scalable whole-genome single-cell library preparation without preamplification. *Nature Methods* **14**(2), 167–173 (2017) <https://doi.org/10.1038/nmeth.4140>
- [2] Laks, E., McPherson, A., Zahn, H., Lai, D., Steif, A., Brimhall, J., Biele, J., Wang, B., Masud, T., Ting, J., *et al.*: Clonal decomposition and dna replication states defined by scaled single-cell genome sequencing. *Cell* **179**(5), 1207–1221 (2019) <https://doi.org/10.1016/j.cell.2019.10.026>
- [3] Minussi, D.C., Nicholson, M.D., Ye, H., Davis, A., Wang, K., Baker, T., Tarabichi, M., Sei, E., Du, H., Rabbani, M., *et al.*: Breast tumours maintain a reservoir of subclonal diversity during expansion. *Nature* **592**(7853), 302–308 (2021) <https://doi.org/10.1038/s41586-021-03357-x>
- [4] Zhao, T., Chiang, Z.D., Morriss, J.W., LaFave, L.M., Murray, E.M., Del Priore, I., Meli, K., Lareau, C.A., Nadaf, N.M., Li, J., Earl, A.S., Macosko, E.Z., Jacks, T., Buenrostro, J.D., Chen, F.: Spatial genomics enables multi-modal study of clonal heterogeneity in tissues. *Nature* **601**(7891), 85–91 (2022) <https://doi.org/10.1038/s41586-021-04217-4>
- [5] Greaves, M., Maley, C.C.: Clonal evolution in cancer. *Nature* **481**(7381), 306–313 (2012) <https://doi.org/10.1038/nature10762>

- [6] Papaemmanuil, E., Gerstung, M., Bullinger, L., Gaidzik, V.I., Paschka, P., Roberts, N.D., Potter, N.E., Heuser, M., Thol, F., Bolli, N., Gundem, G., Loo, P.V., Martincorena, I., Ganly, P., Mudie, L., McLaren, S., O'Meara, S., Raine, K., Jones, D.R., Teague, J.W., Butler, A.P., Greaves, M.F., Ganser, A., Döhner, K., Schlenk, R.F., Döhner, H., Campbell, P.J.: Genomic classification and prognosis in acute myeloid leukemia. *New England Journal of Medicine* **374**(23), 2209–2221 (2016) <https://doi.org/10.1056/NEJMoa1516192>
- [7] Satas, G., Zaccaria, S., Mon, G., Raphael, B.J.: SCARLET: single-cell tumor phylogeny inference with copy-number constrained mutation losses. *Cell systems* **10**(4), 323–332 (2020) <https://doi.org/10.1016/j.cels.2020.04.001>
- [8] Wang, F., Wang, Q., Mohanty, V., Liang, S., Dou, J., Han, J., Minussi, D.C., Gao, R., Ding, L., Navin, N., *et al.*: MEDALT: single-cell copy number lineage tracing enabling gene discovery. *Genome biology* **22**(1), 70 (2021) <https://doi.org/10.1186/s13059-021-02291-5>
- [9] Markowska, M., Cakała, T., Miasojedow, B., Aybey, B., Juraeva, D., Mazur, J., Ross, E., Staub, E., Szczurek, E.: CONET: copy number event tree model of evolutionary tumor history for single-cell data. *Genome Biology* **23**(1), 128 (2022) <https://doi.org/10.1186/s13059-022-02693-z>
- [10] Kaufmann, T.L., Petkovic, M., Watkins, T.B.K., Colliver, E.C., Laskina, S., Thapa, N., Minussi, D.C., Navin, N., Swanton, C., Van Loo, P., Haase, K., Tarabichi, M., Schwarz, R.F.: MEDICC2: whole-genome doubling aware copy-number phylogenies for cancer evolution. *Genome Biology* **23**(1), 241 (2022) <https://doi.org/10.1186/s13059-022-02794-9>
- [11] Melin, H., Zampinetti, V., McPherson, A., Lagergren, J.: VICTree-A variational inference method for clonal tree reconstruction. In: *International Conference on Research in Computational Molecular Biology*, pp. 429–433 (2024). https://doi.org/10.1007/978-1-0716-3989-4_46 . Springer
- [12] Kuipers, J., Tuncel, M.A., Ferreira, P.F., Jahn, K., Beerenwinkel, N.: Single-cell copy number calling and event history reconstruction. *Bioinformatics* **41**(3), 072 (2025) <https://doi.org/10.1093/bioinformatics/btaf072>
- [13] Hui, S., Nielsen, R.: SCONCE2: jointly inferring single cell copy number profiles and tumor evolutionary distances. *BMC Bioinformatics* **23**, 348 (2022) <https://doi.org/10.1186/s12859-022-04890-w>

- [14] Mallory, X.F., Edrisi, M., Navin, N., Nakhleh, L.: Methods for copy number aberration detection from single-cell dna-sequencing data. *Genome Biology* **21**(1) (2020) <https://doi.org/10.1186/s13059-020-02119-8>
- [15] Felsenstein, J.: *Inferring Phylogenies*. Sinauer Associates, Sunderland, MA (2003)
- [16] Jin, L., Nei, M.: Limitations of the evolutionary parsimony method of phylogenetic analysis. *Molecular Biology and Evolution* **7**(1), 82–102 (1990) <https://doi.org/10.1093/oxfordjournals.molbev.a040588>
- [17] Huelsenbeck, J.P.: Performance of phylogenetic methods in simulation. *Systematic Biology* **44**(1), 17–48 (1995) <https://doi.org/10.1093/sysbio/44.1.17>
- [18] Saitou, N., Nei, M.: The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Mol. Biol. Evol.* **4**, 406–425 (1987) <https://doi.org/10.1093/oxfordjournals.molbev.a040454>
- [19] Wheeler, W.C.: Nucleic acid sequence phylogeny and random outgroups. *Cladistics* **6**(4), 363–367 (1990) <https://doi.org/10.1111/j.1096-0031.1990.tb00550.x>
- [20] Pearson, T., Hornstra, H.M., Sahl, J.W., Schaack, S., Schupp, J.M., Beckstrom-Sternberg, S.M., O’Neill, M.W., Priestley, R.A., Champion, M.D., Beckstrom-Sternberg, J.S., *et al.*: When outgroups fail; phylogenomics of rooting the emerging pathogen, *Coxiella burnetii*. *Systematic biology* **62**(5), 752–762 (2013) <https://doi.org/10.1093/sysbio/syt038>
- [21] Outlaw, D.C., Ricklefs, R.E.: Rerooting the evolutionary tree of malaria parasites. *Proceedings of the National Academy of Sciences* **108**(32), 13183–13187 (2011) <https://doi.org/10.1073/pnas.1109153108>
- [22] Zhai, Y., Bouchard-Côté, A.: Inferring rooted population trees using asymmetric neighbor joining. *Ann. Appl. Stat.* **10**, 2047–2074 (2016) <https://doi.org/10.1214/16-aos964>
- [23] Gronau, I., Moran, S.: Neighbor joining algorithms for inferring phylogenies via LCA distances. *J. Comput. Biol.* **14**, 1–15 (2007) <https://doi.org/10.1089/cmb.2006.0115>
- [24] Kurt, S., Bouchard-Côté, A., Lagergren, J.: Sparse neighbor joining: rapid phylogenetic inference using a sparse distance matrix. *Bioinformatics* **40** (2024) <https://doi.org/10.1093/bioinformatics/btae701>

- [25] Atteson, K.: The Performance of Neighbor-Joining Methods of Phylogenetic Reconstruction. *Algorithmica* **25**(2-3), 251–278 (1999) <https://doi.org/10.1007/PL00008277>
- [26] Mihaescu, R., Levy, D., Pachter, L.: Why Neighbor-Joining Works. *Algorithmica* **54**(1), 1–24 (2009) <https://doi.org/10.1007/s00453-007-9116-4>
- [27] Ranwez, V., Gascuel, O.: Improvement of distance-based phylogenetic methods by a local maximum likelihood approach using triplets. *Mol Biol Evol* **19**(11), 1952–1963 (2002) <https://doi.org/10.1093/oxfordjournals.molbev.a004019>
- [28] Levy, D., Yoshida, R., Pachter, L.: Beyond pairwise distances: neighbor-joining with phylogenetic diversity estimates. *Mol Biol Evol* **23**(3), 491–498 (2005) <https://doi.org/10.1093/molbev/msj059>
- [29] Jordan, C.: Sur les assemblages de lignes. *Journal für die reine und angewandte Mathematik* **70**, 185–190 (1869) <https://doi.org/10.1515/crll.1869.70.185>
- [30] Robinson, D.F., Foulds, L.R.: Comparison of phylogenetic trees. *Mathematical Biosciences* **53**(1-2), 131–147 (1981) [https://doi.org/10.1016/0025-5564\(81\)90043-2](https://doi.org/10.1016/0025-5564(81)90043-2)
- [31] Estabrook, G.F., McMorris, F.R., Meacham, C.A.: Comparison of undirected phylogenetic trees based on subtrees of four evolutionary units. *Systematic Biology* **34**(2), 193–200 (1985) <https://doi.org/10.2307/sysbio/34.2.193>
- [32] Dávila Felipe, M., Domelevo Entfellner, J.-B., Lemoine, F., Truszkowski, J., Gascuel, O.: Distribution and asymptotic behavior of the phylogenetic transfer distance. *Journal of Mathematical Biology* **79**(2), 485–508 (2019) <https://doi.org/10.1007/s00285-019-01365-0>
- [33] Critchlow, D.E., Pearl, D.K., Qian, C.: The triples distance for rooted bifurcating phylogenetic trees. *Systematic Biology* **45**(3), 323–334 (1996) <https://doi.org/10.1093/sysbio/45.3.323>
- [34] Steel, M.A., Penny, D.: Distributions of tree comparison metrics—some new results. *Systematic Biology* **42**(2), 126–141 (1993) <https://doi.org/10.1093/sysbio/42.2.126>
- [35] Moreno, M.A., Holder, M.T., Sukumaran, J.: DendroPy 5: a mature Python library for phylogenetic computing. *The Open Journal* (2024).

<https://doi.org/10.21105/joss.06943>

- [36] Sand, A., Holt, M.K., Johansen, J., Brodal, G.S., Mailund, T., Pedersen, C.N.S.: tqDist: a library for computing the quartet and triplet distances between binary or general trees. *Bioinformatics* **30**(14), 2079–2080 (2014) <https://doi.org/10.1093/bioinformatics/btu157>
- [37] Lemoine, F., Domelevo Entfellner, J.-B., Wilkinson, E., Correia, D., Dávila Felipe, M., De Oliveira, T., Gascuel, O.: Renewing Felsenstein’s phylogenetic bootstrap in the era of big data. *Nature* **556**(7702), 452–456 (2018) <https://doi.org/10.1038/s41586-018-0043-0>
- [38] Lefort, V., Desper, R., Gascuel, O.: FastME 2.0: A Comprehensive, Accurate, and Fast Distance- Based Phylogeny Inference Program: Table 1. *Molecular Biology and Evolution* **32**(10), 2798–2800 (2015) <https://doi.org/10.1093/molbev/msv150>
- [39] Weiner, S., Bansal, M.S.: CNAsim: improved simulation of single-cell copy number profiles and DNA-seq data from tumors. *Bioinformatics* **39**(7), 434 (2023) <https://doi.org/10.1093/bioinformatics/btad434>
- [40] Gao, R., Davis, A., McDonald, T.O., Sei, E., Shi, X., Wang, Y., Tsai, P.-C., Casasent, A., Waters, J., Zhang, H., Meric-Bernstam, F., Michor, F., Navin, N.E.: Punctuated copy number evolution and clonal stasis in triple-negative breast cancer. *Nature Genetics* **48**(10), 1119–1130 (2016) <https://doi.org/10.1038/ng.3641>
- [41] 10x Genomics: Aggregate of Breast Tissue Nuclei Sections 10K Cells, Single Cell DNA Dataset by Cell Ranger DNA v1.1.0. https://cf.10xgenomics.com/samples/cell-dna/1.1.0/breast_tissue_aggr_10k/breast_tissue_aggr_10k_web_summary.html. Sample ID: breast_tissue_aggr_10k; Reference: GRCh37-1.0.0. Accessed 16 July 2026 (2019)
- [42] Zaccaria, S., Raphael, B.J.: Characterizing allele- and haplotype-specific copy numbers in single cells with CHISEL. *Nature Biotechnology* **39**(2), 207–214 (2021) <https://doi.org/10.1038/s41587-020-0661-6>

Paper C

Paper C

Sampling random spanning arborescences in graphs with low conductance

VITTORIO ZAMPINETTI, HARALD MELIN, AND JENS
LAGERGREN

Published in
Statistics & Probability Letters

Sampling random spanning arborescences in graphs with low conductance

Vittorio Zampinetti^{a,b,*}, Harald Melin, Jens Lagergren^a

^a*Royal Institute of Technology (KTH), Stockholm, Sweden*

^b*Politecnico di Torino, Italy*

Abstract

Sampling random spanning arborescences in directed graphs is critical for applications in network analysis, optimization, and machine learning. While many state-of-the-art methods perform well on graphs with high conductance, they often fail or generalize poorly on low-conductance graphs. Inspired by Wilson's algorithm, we propose a novel sampling approach that overcomes this limitation by using dynamic programming to compute random walk probabilities. This avoids both inefficient walk simulations and numerically unstable Laplacian determinant calculations. Our method demonstrates superior efficiency and sampling quality in simulations, and is the only one to handle low-conductance graphs effectively.

1. Introduction

Generating random rooted spanning trees in directed weighted graphs, here referred to as *random spanning arborescences*, is a fundamental problem in graph theory (Lyons and Peres, 2017), and the applications in computer science and machine learning range from network analysis (Avena et al., 2018) to natural language processing (Koo et al., 2007) and phylogenetics (Melin et al., 2024; Koptagel et al., 2022).

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, W)$ be a weighted directed graph on a set $\mathcal{V}$ with n nodes where each arc $(i, j) \in \mathcal{E}$ has a non-negative real-valued weight w_{ij} . A spanning arborescence $\mathcal{T}$ of $\mathcal{G}$ is a directed tree rooted at

*Corresponding author

Email addresses: vz@kth.se (Vittorio Zampinetti), haraldme@kth.se, jensl@kth.se (Harald Melin, Jens Lagergren)

some node $r \in \mathcal{V}$ in which all arcs point towards the root. A *random* spanning arborescence $\mathcal{T}$ is drawn from a distribution over all such trees where the probability of sampling $\mathcal{T}$ is proportional to its weight. Specifically, the probability of sampling a spanning arborescence $\mathcal{T}$ is given by

$$\mathbb{P}(\mathcal{T}) = \frac{1}{Z} \mathcal{W}(\mathcal{T}), \quad \mathcal{W}(\mathcal{T}) = \prod_{(i,j) \in \mathcal{T}} w_{ij},$$

where $\mathcal{W}(\mathcal{T})$ is the weight of $\mathcal{T}$, and Z is a normalization constant.

There are two main approaches to this problem. One is to use the classical Matrix-Tree Theorem (Kirchhoff, 1847; Tutte, 1948) which states that the normalization constant Z , i.e., the total weight of spanning trees in a directed graph, is equal to the determinant of the Laplacian of the weight matrix. Kulkarni (1990) and Colbourn et al. (1996) gave algorithms for using the MTT to sample spanning trees in directed graphs which run in $O(n^4)$ and $O(n^3)$ time respectively. Computing the determinant of a matrix is, however, known to be numerically unstable when the matrix is ill-conditioned (i.e., has low condition number) (Lloyd, 2013). The other approach is to sample spanning trees by simulating random walks on the graph and the algorithm by Wilson (1996) is the most well-known and widely used. Random walk based methods run in cover time, i.e., the expected time for a random walk to visit all the nodes in the graph, and thus benefit from high graph conductance. The *conductance* of a directed graph $\mathcal{G}$ can be defined as follows

$$\Phi(\mathcal{G}) = \min_{C \subset \mathcal{V}} \frac{\sum_{u \in C, v \in \mathcal{V} \setminus C} w_{uv}}{\sum_{u, v \in C} w_{uv}}.$$

Here, the minimum is taken over all nonempty, proper subsets $C \subset \mathcal{V}$ to avoid trivial partitions of the graph. In graphs with low conductance, the cover time cannot be bounded by a polynomial function of n and random walks become inefficient.

Recent applications of machine learning have prompted a demand for sampling arborescences algorithms to be used as estimators of the posterior distribution in Bayesian inference methods such as variational inference (Melin et al., 2024) or of gradients in deep learning (Paulus et al., 2021). However, graphs that arise from data typically have no guarantees of high conductance. Notably, if the graph has low conductance, by Cheeger’s inequality (Cheeger, 1970), the

Laplacian matrix is ill-conditioned and the determinant of the Laplacian matrix cannot be computed accurately. Determinant based methods will therefore fail, leaving no viable alternative. Several algorithms exist for the undirected case (Aldous, 1990; Madry et al., 2014; Durfee et al., 2017), but they do not extend to the directed setting. The problem of sampling spanning arborescences from low-conductance graphs has not been given sufficient attention in the literature, and the existing methods do not seem to be able to handle this case.

In this paper, we propose a novel method inspired by Wilson’s random walk algorithm. Instead of constructing an arborescence by simulating a random walk, we devise a dynamic programming algorithm to efficiently compute and update the probabilities of all random walk outcomes. Our approach addresses the limitations of existing methods, resulting in a sampling technique which allows to efficiently draw samples of random spanning arborescences even in the presence of low conductance.

In Section 2, we introduce relevant graph theory concepts and describe Wilson’s algorithm for sampling random spanning arborescences. In Section 3 we present our dynamic programming approach, which we call *Castaway*, and elaborate on how to use it for sampling random arborescences in low-conductance graphs. Then, in Section 4, we show that our method compares well with the state-of-the-art in terms of time, accuracy and feasibility, by reporting the results of experiments on simulated graphs. Finally, we conclude the paper with a discussion on the potential applications of our method and future work. The code for reproducing the experiments is available at github.com/Lagergren-Lab/treesampling.

2. Related methods

The algorithm by Wilson (1996) provides a method for sampling random spanning arborescences in directed graphs by simulating random walks.

Definition 1. (*Random walk on graph*) Given a weighted directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, W)$, a random walk on $\mathcal{G}$ is a Markov chain over the nodes where the probability of transitioning to node v from node u is given by the normalized weight $\tilde{w}_{uv} = w_{uv} / \sum_{v'} w_{uv'}$.

The realization of the random walk, simply called a *walk*, is a sequence of nodes $v_1, \dots, v_k$ such that v_1 is the starting node and v_k is the last node visited. The probability of a random walk from v_1 to v_k is given by the product of the transition probabilities along the walk. If the walk visits a node more than once, the path is said to contain a *loop*.

Definition 2. (*Loop-erased random walk*) A random walk where all loops are erased. The loop-erased random walk generates a sequence of distinct nodes $v_1, \dots, v_k$ (a path) where each arc (v_i, v_{i+1}) is the last arc leaving v_i in the random walk.

Wilson's algorithm. Given a fixed root $r \in \mathcal{V}$, the algorithm initialize the tree to contain only the root node. Then a loop-erased random walks starts from $v_1 \neq r$, picked from any of the nodes not in the tree. The walk proceeds until it transitions to the root. The path thereby generated from v_1 to r is added to the tree. Another loop-erased random walk is started from the next node v'_1 not in the tree and terminates when a node in the tree is visited. The path so obtained is appended to the tree. The process continues until all nodes have been added to the tree. The algorithm is guaranteed to sample a random spanning tree with probability proportional to the product of the weights of its arcs. However, the following argument shows that the algorithm may take a long time to visit all nodes in the graph. For example, in a rooted directed graph with a single root node r and only two other nodes v_1 and v_2 that are strongly connected to each other and weakly connected to r , the random walk starting from v_1 will cycle between v_1 and v_2 for a time inversely proportional to the conductance of the graph before visiting the root and therefore terminating the walk.

3. Methods

We now describe our algorithm Castaway for sampling random spanning arborescences. We give an overview of the algorithm and then describe the details of the computation of the probabilities. Subsequently, we discuss instability issues for graphs with low conductance and how to resolve them.

3.1. Algorithm overview

Let $(v_1, \dots, v_k)$ be a walk on $\mathcal{G} = (\mathcal{V}, \mathcal{E}, W)$ right before reaching a node in the current tree $\mathcal{S} = (\mathcal{V}_{\mathcal{S}}, \mathcal{E}_{\mathcal{S}})$ in an iteration of Wilson's algorithm. We are interested in the first and last nodes v_1, v_k . The probability of walking on the set of nodes $\mathcal{U} = \mathcal{V} \setminus \mathcal{V}_{\mathcal{S}}$ from v_1 to v_k is given by the sum of the probabilities of all walks with the same endpoints. This quantity, that we denote by $P_{\mathcal{U}}(v_1, v_k)$, can be computed efficiently with dynamic programming. Let $\mathcal{O}$ be the *out set*, that is the set of nodes which, once reached, terminate the random walk, and initialize it to $\mathcal{O} = \mathcal{V}_{\mathcal{S}}$. The probability that v_k is the *exit node*, that is the last node visited in the walk before reaching $\mathcal{O}$, is given by

$$q_{v_k}(v_1) = P_{\mathcal{U}}(v_1, v_k) \sum_{v_o \in \mathcal{O}} \tilde{w}_{v_k v_o}. \quad (1)$$

Then, for any node $v_o \in \mathcal{O}$, the probability of ending the walk in v_o and therefore adding the arc (v_k, v_o) is $\tilde{w}_{v_k v_o}$. After adding the arc (v_k, v_o) to the tree, v_k is removed from the set $\mathcal{U}$. This step fixes the last arc in the loop-erased random walk started from v_1 and, in order to further draw arcs backwards in the path, we set $\mathcal{O} = \{v_k\}$ as the new out set. The table $P_{\mathcal{U}}$ is updated by subtracting the probability of all walks that go through v_k . The algorithm continues by drawing the node $v_{k-1} \in \mathcal{U}$ proportionally to $P_{\mathcal{U}}(v_1, v_{k-1})\tilde{w}_{v_{k-1} v_k}$ and adding the arc (v_{k-1}, v_k) to the tree. This is repeated until the start node v_1 joins the tree. Then as in Wilson's algorithm, the next start node v'_1 is picked from the set of nodes not in the tree $\mathcal{U}$ and the process is repeated. The algorithm ends when all nodes are added to the tree. We provide the pseudocode in Appendix B in the supplementary material online.

Probability table. The table is built by iteratively adding nodes to the set $\mathcal{U} = \{v_1, \dots, v_m\}$ from those not in the tree. Let $\mathcal{U}^{(k)}$ be the set of k nodes in the graph that have been added to $\mathcal{U}$. Initially, $\mathcal{U}^{(1)} = \{v_1\}$ and $P_{\mathcal{U}^{(1)}}(v_1, v_1) = 1$. We define $P_{\mathcal{U}^{(k)}}(v_i, v_j)$ recursively:

$$P_{\mathcal{U}^{(k)}}(v_i, v_j) = P_{\mathcal{U}^{(k-1)}}(v_i, v_j) + R_{\mathcal{U}^{(k-1)}}^{\text{out}}(v_i, v_k) \times \rho_{\mathcal{U}^{(k)}}(v_k) \times R_{\mathcal{U}^{(k-1)}}^{\text{in}}(v_k, v_j), \quad (2)$$

where

$$\begin{aligned}
R_{\mathcal{U}^{(k-1)}}^{\text{out}}(v_i, v_k) &= \sum_{v \in \mathcal{U}^{(k-1)}} P_{\mathcal{U}^{(k-1)}}(v_i, v) \tilde{w}_{vv_k}, \\
R_{\mathcal{U}^{(k-1)}}^{\text{in}}(v_k, v_j) &= \sum_{v \in \mathcal{U}^{(k-1)}} \tilde{w}_{v_kv} P_{\mathcal{U}^{(k-1)}}(v, v_j) \text{ and} \\
\rho_{\mathcal{U}^{(k)}}(v_k) &= \left(1 - \sum_{u \in \mathcal{U}^{(k-1)}} \tilde{w}_{v_k u} R_{\mathcal{U}^{(k-1)}}^{\text{out}}(u, v_k) \right)^{-1} \quad (3)
\end{aligned}$$

are the probabilities of walking out of $\mathcal{U}^{(k-1)}$, walking in $\mathcal{U}^{(k-1)}$, and, from v_k , returning to itself through $\mathcal{U}^{(k)}$ respectively. Full derivation of the formula is given in Appendix A in the supplementary material online.

3.2. Resolving instability

The algorithm may fail to build the probability table due to numerical instability. In particular, when a node v_k is added to the set of nodes, the return probability $\rho_{\mathcal{U}^{(k)}}(v_k)$ may numerically evaluate to 1, causing the the geometric series in Equation (3) to diverge. This phenomenon arises because a random walk restricted to $\mathcal{U}$ visits v_k with such high frequency that limited machine precision leads to numerical overflow. To characterize this behaviour, we identify a set of nodes — hereafter referred to as *component* — in which the ratio between the total weight of outgoing arcs and the total weight of internal arcs is vanishingly small. This structure corresponds to a region of low conductance. Furthermore, a high return probability at v_k often implies that such node receives high-weighted arcs even from outside the component. The presence of one or more such nodes in the graph increase the likelihood of near-linear dependencies in the Laplacian matrix, which in turn complicates the computation of the determinant. We fix to this issue by handling nodes that lead to such instability in a special way. We call these nodes *crasher nodes*. By excluding the crasher nodes from the set $\mathcal{U}$, the probability table can be computed without numerical issues. The probability of attaching a node v_k to the tree is approximated with

$$q_{v_k}(v_1) = \sum_{u \in \mathcal{U}} P_{\mathcal{U}}(v_1, u) \sum_{c \in \mathcal{X}} \tilde{w}_{uc} R_{\mathcal{U}}^{\text{in}}(c, v_k) \sum_{v_o \in \mathcal{O}} \tilde{w}_{v_kv_o}, \quad (4)$$

where $\mathcal{X}$ is the set of crasher nodes in the graph. The formula relies on the reasonable assumption that any random walk that leaves a component more than once gives a negligible contribution to the total probability of the walks. Edge cases such as v_k being a crasher node or v_1 being a crasher node are further discussed in Appendix C in the supplementary material online. The original algorithm is extended to handle this case by identifying crasher nodes as follows: if the return probability of a node becomes infinite, we add that node to the set of crasher nodes and adopt the new formula for sampling exit nodes. This fix brings the complexity of the algorithm up to $O(n^4K)$ which is still polynomial in the number of nodes n and K being the number of crasher nodes.

4. Experiments

To evaluate our method we run experiments on simulated graphs. We assess the performance of the algorithm compared to Wilson’s algorithm and algorithm A8 in Kulkarni (1990) (hereafter referred to as *Kulkarni*), in terms of running time and distribution of the sampled trees. Although Kulkarni’s algorithm is not the fastest determinant-based method—Colbourn’s runs in $O(n^3)$ —it is more numerically stable, and the runtime difference is irrelevant for the purpose of our investigation. In order to highlight the weaknesses of the existing methods, we generate graphs that have low conductance. We address the unstable case of the *crasher nodes* as described in Section 3.2 and in order to do so, we simulate graphs with a certain structure. These graphs have K components, each of which contains $\lfloor n/(K+1) \rfloor$ nodes that have arcs with unit weights between each other and ε -weighted arcs to nodes outside their component, with $\varepsilon > 0$ being a parameter of the simulation. Each component contains a crasher node. The remaining $n - \lfloor n/(K+1) \rfloor$ nodes are further divided such that one node is set as the root, with no incoming arcs (zero weight) and ε -weighted outgoing arcs, and the rest of the nodes have ε -weighted arcs to all other nodes except the crasher nodes, which receive unit weight arcs. A schematic representation of the graph structure is shown in Figure 1.

We generate graphs with $n = 7$ nodes divided into $K = 2$ crasher components with two nodes each, one root node and another two-node non-crasher component. This size is small enough to allow for

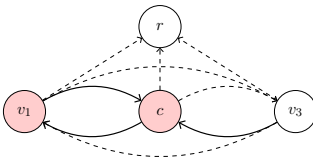


Figure 1: Example of a graph with four nodes and one crasher component (red). Dashed arcs represent ε -weighted arcs, while solid arcs have unit weights. The crasher c is visited many times with high probability by a random walk from any node inside and outside the component, e.g., v_1 and v_3 respectively.

a complete enumeration of the spanning arborescences, which we use to compute the ground truth and benchmark the algorithms. The ε -weight, or *weak weight*, takes values such that $\log(\varepsilon) \in \{-2, -3, -5, -7, -10, -20, -40, -100\}$, and for each value, we generate 10 perturbed matrices of weights. We run the three algorithms on $N = 1000$ samples and compare three metrics: running time, mean squared error (MSE) of the marginal distribution of arcs and fraction of probability mass covered by the sampled trees. To improve the accuracy of all algorithms, the sampled trees are re-weighted, meaning that their frequency is replaced by the tree weight divided by the total weight of the sampled trees. The Castaway algorithm is executed in its $O(n^3)$ version until it fails to compute the probability table due to numerical overflow. Past that point, we run the algorithm with the instability fix described in Section 3.2. Figure 2 summarizes the results of the experiments. In particular, running times are plotted in Figure 2a and show that Castaway is the only method that allows to sample spanning arborescences with weaker weights in a reasonable time, while still being able to sample the correct distribution, as shown in Figure 2b. Castaway is thus proven to outperform the other two methods since, as the weak weight decreases, Wilson’s algorithm rapidly becomes impractical, and for $\log(\varepsilon) \leq -40$ Kulkarni’s algorithm fails to sample arborescences due to errors in the determinant computation.

We repeat the experiment on larger graphs ($n = 100$) with unchanged number of components and reduced sample size ($N = 100$) to keep the running time reasonable. The results are shown in Figure 2c. The increased size of the graph does not allow for a complete enumeration of the spanning arborescences, which is why the MSE is not computed. The fraction of total probability mass can only be

computed when ε is large enough for the determinant computation to be stable. Nevertheless, the running time follows the same trend as in the smaller graphs experiment and the quality of the sampled trees are similar across the three algorithms.

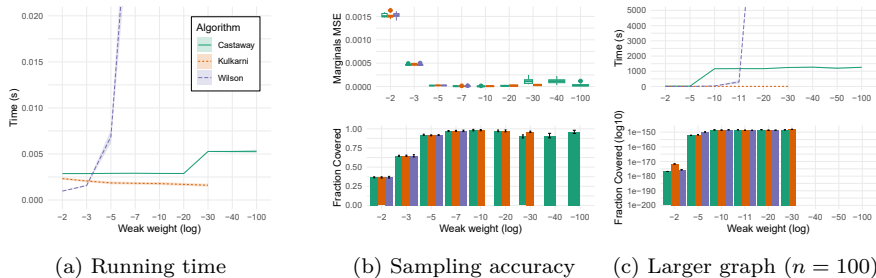


Figure 2: (a) Average running time over 10 differently weighted graphs plotted as a function of $\log(\varepsilon)$. Shaded regions show the standard deviation of the estimates. The lineplot for Kulkarni is incomplete due to failure in computing the determinant in low-conductance graphs. (b) Boxplots (upper plot) show the MSE for marginal distribution of arcs against ground truth and bars (lower plot) show the fraction of probability mass covered by the sampled trees. Error bars indicate one standard deviation from the estimates. (c) Running time and fraction of probability mass covered by the sampled trees for graphs with $n = 100$ nodes.

5. Conclusion

We present a novel random spanning arborescence sampling algorithm for directed graphs. The method runs in polynomial time with n as the number of nodes in the graph and succeeds in sampling spanning trees even in the presence of weak connections, where existing methods fail. We demonstrate the effectiveness of our method through simulation experiments on graphs with weak weights and show that it is the only method able to sample spanning arborescences in a reasonable time. Future work may include the extension of the algorithm to sampling arborescences without a fixed root.

References

Lyons R. and Peres Y., *Probability on Trees and Networks*, Cambridge University Press, 2017.

- Avena L., Castell F., Gaudillière A., and Mélot C., *Random Forests and Networks Analysis*, Journal of Statistical Physics, vol. 173, pp. 985–1027, 2018.
- Koo T., Globerson A., Carreras X., and Collins M., *Structured Prediction Models via the Matrix-Tree Theorem*, Proc. EMNLP-CoNLL, pp. 141–150, 2007.
- Melin H., Zampinetti V., McPherson A., and Lagergren J., *VIC-Tree: a Variational Inference Method for Clonal Tree Reconstruction*, Proc. RECOMB 2024, Cambridge, MA, USA, pp. 429–433, 2024.
- Koptagel H., Kviman O., Melin H., Safiniani N., and Lagergren J., *VaiPhy: a Variational Inference Based Algorithm for Phylogeny*, Adv. Neural Inf. Process. Syst., vol. 35, pp. 14758–14770, 2022.
- Kirchhoff G., *Ueber die Auflösung der Gleichungen, auf welche man bei der Untersuchung der linearen Vertheilung galvanischer Ströme geführt wird*, Ann. Phys., vol. 148, no. 12, pp. 497–508, 1847.
- Tutte W. T., *The Dissection of Equilateral Triangles into Equilateral Triangles*, Math. Proc. Cambridge Philos. Soc., vol. 44, no. 4, pp. 463–482, Oct. 1948.
- Kulkarni V. G., *Generating Random Combinatorial Objects*, Journal of Algorithms, vol. 11, no. 2, pp. 185–207, 1990.
- Colbourn C. J., Myrvold W. J., and Neufeld E., *Two Algorithms for Unranking Arborescences*, Journal of Algorithms, vol. 20, no. 2, pp. 268–281, 1996.
- Trefethen L. N. and Bau D. III, *Numerical Linear Algebra*, SIAM, Philadelphia, PA, 1997.
- Wilson D. B., *Generating Random Spanning Trees More Quickly than the Cover Time*, Proc. 28th ACM Symp. Theory Comput., pp. 296–303, 1996.
- Paulus M. B., Choi D., Tarlow D., Krause A., and Maddison C. J., *Gradient Estimation with Stochastic Softmax Tricks*, arXiv preprint arXiv:2006.08063, 2021.

- Cheeger J., *A Lower Bound for the Smallest Eigenvalue of the Laplacian*, in: Gunning R. C. (Ed.), *Problems in Analysis: A Symposium in Honor of Salomon Bochner*, Princeton Univ. Press, Princeton, NJ, pp. 195–199, 1970.
- Aldous D. J., *The Random Walk Construction of Uniform Spanning Trees and Uniform Labelled Trees*, SIAM Journal on Discrete Mathematics, vol. 3, no. 4, pp. 450–465, 1990.
- Madry A., Straszak D., and Tarnawski J., *Fast Generation of Random Spanning Trees and the Effective Resistance Metric*, Proc. 26th ACM-SIAM Symp. Discrete Algorithms, 2014.
- Durfee D., Kyng R., Peebles J., Rao A., and Sachdeva S., *Sampling Random Spanning Trees Faster than Matrix Multiplication*, Proc. 49th ACM SIGACT Symp. Theory Comput., pp. 730–742, 2017.

Appendix A. Dynamic programming for probability table

Given any subset of nodes $\mathcal{U} \subseteq \mathcal{V} \setminus \{r\}$, we compute the probability $P_{\mathcal{U}}(v, u)$ that, for any pair of nodes $u, v \in \mathcal{U}$, stores the probability of reaching u from v in a random walk on the nodes in $\mathcal{U}$. The table is initially computed for the singleton set $\mathcal{U}^{(1)} = \{v_1\}$ and the only value in the table is $P_{\mathcal{U}^{(1)}}(v_1, v_1) = 1$. Then we iteratively expand the set of nodes by adding one node at a time and updating the probability table. Specifically, let $P_{\mathcal{U}^{(k-1)}}(b, \cdot)$ be the already computed probability table for the set of nodes $\mathcal{U}^{(k-1)} = \{v_1, \dots, v_{k-1}\}$ and let $\mathcal{U}^{(k)} = \mathcal{U}^{(k-1)} \cup \{v_k\}$. The new probability of a walk between $v_i, v_j \in \mathcal{U}^{(k)}$ is the sum of probabilities of all possible walks that end in v_j starting from v_i , which can be decomposed into two cases: either the walk never visits the node v_k or not. This leads to the following dynamic programming formula:

$$P_{\mathcal{U}^{(k)}}(v_i, v_j) = P_{\mathcal{U}^{(k-1)}}(v_i, v_j) + R_{\mathcal{U}^{(k-1)}}^{\text{out}}(v_i, v_k) \times \rho_{\mathcal{U}^{(k)}}(v_k) \times R_{\mathcal{U}^{(k-1)}}^{\text{in}}(v_k, v_j),$$

with

$$R_{\mathcal{U}^{(k-1)}}^{\text{out}}(v_i, v_k) = \sum_{v' \in \mathcal{U}^{(k-1)}} P_{\mathcal{U}^{(k-1)}}(v_i, v') \tilde{w}_{v'v_k},$$

$$R_{\mathcal{U}^{(k-1)}}^{\text{in}}(v_k, v_j) = \sum_{v' \in \mathcal{U}^{(k-1)}} \tilde{w}_{v_kv'} P_{\mathcal{U}^{(k-1)}}(v', v_j)$$

being the probabilities of walking out of and in $\mathcal{U}$. We define $\rho_{\mathcal{U}^{(k)}}(v_k)$ to be the return probability of node v_k in the set $\mathcal{U}^{(k)}$, namely the sum of probabilities of all walks on $\mathcal{U}^{(k)}$ that start from v_k and return to v_k . Given the table at iteration $k-1$, the return probability of v_k is simply given by the geometric series

$$\rho_{\mathcal{U}^{(k)}}(v_k) = \sum_{i=0}^{\infty} \rho_{\mathcal{U}^{(k)},1}(v_k)^i = \frac{1}{1 - \rho_{\mathcal{U}^{(k)},1}(v_k)},$$

where $\rho_{\mathcal{U}^{(k)},1}(v_k)$ is the probability of returning to v_k exactly once and is equal to

$$\rho_{\mathcal{U}^{(k)},1}(v_k) = \sum_{u \in \mathcal{U}^{(k-1)}} \tilde{w}_{v_k u} R_{\mathcal{U}^{(k-1)}}^{\text{out}}(u, v_k) = \sum_{u \in \mathcal{U}^{(k-1)}} R_{\mathcal{U}^{(k-1)}}^{\text{in}}(v_k, u) \tilde{w}_{uv_k}.$$

Using this, we can also derive the probability of reaching v_j from v_k in $\mathcal{U}^{(k)}$

$$P_{\mathcal{U}^{(k)}}(v_j, v_k) = R_{\mathcal{U}^{(k-1)}}^{\text{out}}(v_j, v_k) \rho_{\mathcal{U}^{(k)}}(v_k). \quad (\text{A.1})$$

The symmetric case $P_{\mathcal{U}^{(k)}}(v_k, v_i)$ is analogous:

$$P_{\mathcal{U}^{(k)}}(v_k, v_i) = \rho_{\mathcal{U}^{(k)}}(v_k) R_{\mathcal{U}^{(k-1)}}^{\text{in}}(v_k, v_i).$$

Finally we set $P_{\mathcal{U}^{(k)}}(v_k, v_k) = \rho_{\mathcal{U}^{(k)}}(v_k)$. These expressions allow us to build the probability table $P_{\mathcal{U}}(f, \cdot)$ from in $O(n^3)$ so that it can be used for sampling arcs as described in the previous section. After an arc is added, the table is updated since a node is removed from the current $\mathcal{U}$. It is possible to do so without having to start from scratch. In fact, each entry in the table can be updated in constant time with the following formula derived from Equations (2) and (A.1):

$$\begin{aligned} P_{\mathcal{U}^{(k-1)}}(v_j, v_i) &= P_{\mathcal{U}^{(k)}}(v_j, v_i) - R_{\mathcal{U}^{(k-1)}}^{\text{out}}(v_i, v_k) \rho_{\mathcal{U}^{(k)}}(v_k) R_{\mathcal{U}^{(k-1)}}^{\text{in}}(v_k, v_j) \\ &= P_{\mathcal{U}^{(k)}}(v_j, v_i) - \frac{P_{\mathcal{U}^{(k)}}(v_i, v_k) P_{\mathcal{U}^{(k)}}(v_k, v_j)}{\rho_{\mathcal{U}^{(k)}}(v_k)}, \end{aligned}$$

Appendix B. Pseudocode

Random Spanning Arborescence Sampler (Castaway) 1

Input: $\mathcal{G} = (\mathcal{V}, \mathcal{E}, W), r \in V$

Output: random spanning arborescence $\mathcal{T}$

```

1:  $\mathcal{S} \leftarrow \emptyset, \mathcal{O} \leftarrow \{r\}$ 
2:  $\mathcal{U} \leftarrow \mathcal{V} \setminus r$ 
3: while  $\mathcal{U} \neq \emptyset$  do
4:    $P_{\mathcal{U}} \leftarrow \text{buildTable}(\mathcal{U})$ 
5:    $v_1 \sim \text{Uniform}(\mathcal{U})$ 
6:    $v_k \sim \text{Discrete}(q(v_1))$  ▷ Eq (1)
7:    $o \sim \text{Discrete}((\tilde{w}_{v_k o})_{o \in \mathcal{O}})$ 
8:    $\mathcal{S} \leftarrow \mathcal{S} \cup \{(v_k, o)\}$ 
9:   while  $v_k \neq v_1$  do
10:     $\mathcal{O} \leftarrow \{v_k\}$ 
11:     $\mathcal{U} \leftarrow \mathcal{U} \setminus v_k$ 
12:     $P_{\mathcal{U}} \leftarrow \text{buildTable}(\mathcal{U})$ 
13:     $v_k \sim \text{Discrete}(q(v_1))$ 
14:    if  $|\mathcal{O}| > 1$  then
15:       $o \sim \text{Discrete}((\tilde{w}_{v_k o})_{o \in \mathcal{O}})$ 
16:    else
17:       $o \leftarrow \text{pop}(\mathcal{O})$ 
18:       $\mathcal{O} \leftarrow \{v_k\}$ 
19:       $\mathcal{S} \leftarrow \mathcal{S} \cup \{(v_k, o)\}$ 
20:    end if
21:  end while
22: end while
23:  $\mathcal{T} \leftarrow \mathcal{S}$  ▷ Spanning arborescence
24: return  $\mathcal{T}$ 

```

Appendix C. Sampling with crasher nodes

In the following we complete the description of Castaway sampling when the Laplacian matrix is ill-conditioned to the point that the computation of the required probability table becomes numerically

unstable. As discussed in Section 3.2 we identify sets of nodes (components) that are not well connected to the rest of the graph and for each component we characterize the node which first causes the dynamic programming algorithm to fail, hence we call it a *crasher node*.

The probability table is built excluding crasher nodes and when candidate nodes are sampled to be added to the tree, we have to consider several cases. More specifically, as argued in Section 3.1, we consider the walk on the graph $R = (v_1, \dots, v_k)$ such that v_1 is picked in any order from the nodes not in the tree, and v_k is the last node before the random walk exits to the tree. In the context of crasher nodes, we have to distinguish between the following cases: the starting node v_1 is a crasher node, the last node v_k is a crasher node, the start and end nodes coincide with a crasher node or a crasher node is visited somewhere in the middle of the random walk. The latter case is the one discussed in Section 3.2 and describes a random walks that starts in a node $v_1 \in \mathcal{U}$, visits a crasher node c from a second node in $\mathcal{U}$, then from the crasher node it walks back to the set $\mathcal{U}$ and eventually reaches v_k which is the last node before exiting to the tree. The formula for the probability of this event, for any $v_k \in \mathcal{U}$ is given by Equation (4).

If v_1 is a crasher node, then the probability of exiting to the tree from $v_k \in \mathcal{U}$ is given by only the second term of Equation (4):

$$R_{\mathcal{U}}^{\text{in}}(v_1, v_k) \sum_{v_o \in \mathcal{O}} \tilde{w}_{v_k v_o},$$

while if v_k is a crasher node, then the probability is given by the first term

$$R_{\mathcal{U}}^{\text{out}}(v_1, v_k) \sum_{v_o \in \mathcal{O}} \tilde{w}_{v_k v_o}.$$

Finally, in the degenerate case where $v_1 = v_k$ is a crasher node, the probability of it being the last node before exiting to the tree equals

$$\sum_{v_o \in \mathcal{O}} \tilde{w}_{v_k v_o}.$$

Appendix D. Numerical Results

The following table contains the results of the experiments presented in Section 4.

$\log(\varepsilon)$	$\log(\Phi(\mathcal{G}))$	Edge Marginal MSE			Edge Co-occurrence MSE			Fraction Covered		
		Wilson	Kulkarni	Castaway	Wilson	Kulkarni	Castaway	Wilson	Kulkarni	Castaway
-2	-1.4	1.5e-3	1.5e-3	1.5e-3	6.0e-4	6.0e-4	6.0e-4	0.37	0.36	0.37
-3	-2.2	4.6e-4	4.7e-4	4.7e-4	2.1e-4	2.1e-4	2.1e-4	0.65	0.65	0.65
-5	-3.9	2.3e-5	2.4e-5	2.1e-5	1.2e-5	1.3e-5	1.1e-5	0.92	0.91	0.92
-7	-5.8	6.9e-6	6.9e-6	7.8e-6	3.8e-6	3.8e-6	4.4e-6	0.98	0.97	0.97
-10	-8.5	×	8.0e-6	6.0e-6	×	4.1e-6	3.3e-6	×	0.98	0.98
-20	-18	×	2.0e-5	1.7e-5	×	9.1e-6	8.0e-6	×	0.97	0.97
-30	-27	×	3.1e-5	1.2e-4	×	1.4e-5	5.9e-5	×	0.96	0.90
-40	-36	×	2.6e-3	1.2e-4	×	1.1e-3	5.8e-5	×	0.78	0.91
-100	-91	×	×	3.5e-5	×	×	2.1e-5	×	×	0.96

× indicates NaN/unavailable values.

Paper D

Paper D

VICTree: a variational inference method for clonal tree reconstruction

HARALD MELIN, VITTORIO ZAMPINETTI, ANDREW MCPHERSON, AND JENS LAGERGREN

Published in
RECOMB 2024 Conference Proceedings

VICTree - a Variational Inference method for Clonal Tree reconstruction

Harald Melin^{*1}, Vittorio Zampinetti^{*1,2}, Andrew McPherson³, and Jens Lagergren¹

¹ School of EECS, KTH Royal Institute of Technology, Stockholm, Sweden
`{haraldme,jensl}@kth.se`

² Department of Mathematical Sciences, Politecnico di Torino, Italy
`vittorio.zampinetti@polito.it`

³ Computational Oncology, Memorial Sloan Kettering Cancer Center, New York, NY, USA
`mcphera1@mskcc.org`

Abstract. Clonal tree inference brings crucial insights to the analysis of tumor heterogeneity and cancer evolution. Recent progress in single cell sequencing has prompted a demand for more advanced probabilistic models of copy number evolution, coupled with inference methods which can account for the noisy nature of the data along with dependencies between adjacent sites in copy number profiles. We present VICTree, a variational inference based algorithm for joint Bayesian inference of clonal trees, together with a novel Tree-structured Mixture Hidden Markov Model (TSMHMM) which combines HMMs related through a tree with a mixture model. For the tree inference, we introduce a new algorithm, LARS, for sampling directed labeled multifurcating trees. To evaluate our proposed method, we conduct experiments on simulated data and on samples of multiple myeloma and breast cancer. We demonstrate VICTrees capacity for reliable clustering, clonal tree reconstruction, copy number evolution and the utility of the ELBO for model selection. Lastly, VICTree’s results are compared in terms of quality and speed of inference to other state-of-the-art methods. The code for VICTree is available on GitHub: github.com/Lagergren-Lab/victree.

Keywords: Tumor phylogenetics · copy number evolution · scDNA-seq · clonal tree · Bayesian inference · Variational inference.

⁴ *Equal contribution

1 Introduction

The well-established clonal theory of cancer [27] describes cancer development as a process of phenotypically distinct cell sub-populations, referred to as clones, undergoing Darwinian evolution driven by somatic genetic mutations. The resulting tumor heterogeneity is a substantial barrier for successful treatment [10]. To analyze the genetic content of a tumor, the recent single cell whole genome sequencing (scWGS) technology Direct Library Preparation (DLP) [35], has shown proficiency in producing data of thousands of single cells suitable for copy number (CN) analysis [30, 35]. Yet, the data exhibits imperfections and the experimental process is inherently noisy. These limitations motivates the necessity of adopting phylogenetic methods able to incorporate uncertainty in the inferences.

CN deletion and duplication events may effect multiple genomic sites. This has sparked recent surge in research of models able to accommodate for dependencies within the CN profile (CNP), e.g., in form of clustering methods [29], and phylogenetic methods which also account for site-dependent CN evolution [16, 25, 30, 32, 34]. Despite their demonstrated strengths, the phylogenetic methods either rely on heuristic approaches [16, 32, 34] which lack the rigor of probabilistic modeling, or on simplifying assumptions and pre-processing steps that break the full Bayesian aspiration [25, 30].

Markov chain Monte Carlo (MCMC) has played a pivotal role in Bayesian phylogenetics, both in classical setting [7, 12] and in the CN tumor setting [25, 30]. However, the introduction of dependencies within the CNPs causes the size of the hidden state-space to be subject to combinatorial explosion, which has proven difficult for MCMC in classical phylogenetics [13], where the marginalization over the hidden states to evaluate the likelihood becomes intractable. In the CN clonal tree setting, a cluster assignment and CN state must be given in order to evaluate the likelihood. Thus, MCMC algorithms dedicated to fully explore this state-space will be limited. Therefore, previous tumor phylogenetic CN based methods circumvent this problem by requiring the unobservable CNPs as input [30], these are obtained e.g. by running HMMCopy [22], and by implicitly segmenting the CNPs through a transformation [30] or user-input to select potential breakpoints [25]. Unfortunately, the decisions taken in such preprocessing cannot be reversed later and may lead to artifacts. Variational inference [2, 14] is alternative to MCMC and has delivered substantial speed-up of inference in a variety of settings [1], including in the context of classical phylogenetics [13, 18, 26, 36, 37], but is yet to enter the tumor phylogenetic arena.

We present VICTree, a method that utilizes VI for joint Bayesian inference of site-dependent CN evolution, clonal tree reconstruction, cell-to-clone clustering and cell specific read baselines, using only GC-correction and normalization for data pre-processing. VICTree draws inspiration from both the tumor and classical phylogenetic literature.

To model site-dependent CN evolution, we extend the dependency graph between hidden states of the phylo-HMM model [31], which describes the evolution of site-dependent chromatin's in a phylogenetic tree, and infuse it with a transition model tailored to tumor phylogenetics. To this we add a mixture model and an emission model, extending the Mixture Hidden Markov Model (MHMM) [11, 28] of CopyMix [29]. The combination yields a Tree structured MHMM (TSMHMM), a novel type of HMM, suitable for tumor phylogeny from the clonal perspective.

In order to achieve efficient and accurate inference, we employ a coordinate ascent VI (CAVI) approach and derive update equations to the associated variational distributions. To estimate the non-standard expected value w.r.t tree topologies, we employ Importance sampling [17] (IS) as in [18]. Furthermore, we introduce a novel algorithm for labeled arborescences sampling, LARS, to use as proposal distribution for our IS procedure. Lastly, we identify that and describe why VI based clonal analysis is haunted by the problem of component degeneracy, i.e., high similarity between clones. We resolve the problem by extending our VI algorithm with a novel domain specific split-and-merge-operation, thereby, taking VI based Bayesian clonal analysis from the theoretically interesting to the practically applicable. As a bonus, the use of VI equips us with a lower bound on the marginal likelihood, a metric which encodes Occam's Razor, fit for model selection [24]. By assembling these methods and the TSMHMM, we reach our aspiration of a fully automated Bayesian inference procedure of clonal phylogenies.

Contributions

- **VICTree** - The first framework for fully automated Bayesian inference over genomic site-dependent CN evolution, clonal tree topologies, cell-to-clone assignment, clonal prevalence and cell specific emission baseline and precision.
- **TSMHMM** - A novel type of HMM, the Tree-Structured Mixture-HMM, suitable for the tumor clonal tree context.
- **LARS** - a novel algorithm for sampling labeled multifurcating rooted trees (also known as *arborescences*) induced by a weighted graph.
- **Split-and-merge algorithm** - a novel Split-and-merge algorithm tailored to the context of clonal trees, taking VI based clonal analysis from the theoretically interesting to the practically applicable.

2 Background

Given observations, $\mathbf{y}$, described by a set of latent variables and parameters, $\mathbf{x}$, of a generative model $p(\mathbf{x}, \mathbf{y})$, Bayesian inference is concerned with finding

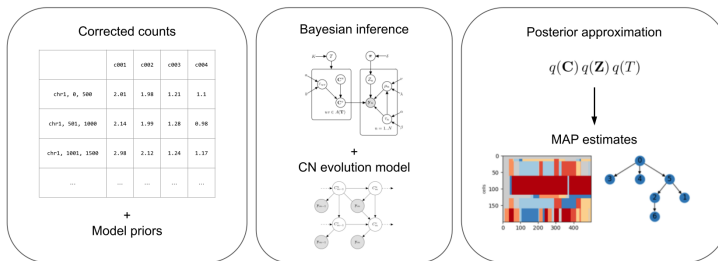


Fig. 1: Overview of VICTree. Bin counts are the only required input data and have to be corrected for GC content and ploidy (neutral CN equals 2). The method performs Bayesian inference through CAVI updates by optimizing the parameters of the model here introduced. The output produced is the set of variational distribution, from which it is possible to retrieve MAP estimates for downstream analysis (CN calling, clustering of cells into clones and CN-aware evolution tree).

the posterior distribution:

$$p(\mathbf{x}|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{x})p(\mathbf{x})}{p(\mathbf{y})} \quad (1)$$

Unfortunately, the posterior is intractable due to the denominator in 1, called the marginal likelihood or evidence, and therefore an approximate Bayesian inference procedure must be applied.

Variational Inference (VI) [2, 14] addresses this by finding a surrogate distribution, called a variational distribution, $q_\phi(\mathbf{x})$, parameterized by ϕ , which best approximates the posterior $p(\mathbf{x}|\mathbf{y})$, in terms of minimal KL-Divergence, D_{KL} , between q and p . Furthermore, instead of minimizing D_{KL} explicitly, maximizing the Evidence Lower Bound, hereinafter referred to as *ELBO*, is preferred. The ELBO is defined as:

$$\text{ELBO}_q(\phi) := \mathbb{E}_{q_\phi(\mathbf{x})} \left[\log \frac{p(\mathbf{x}, \mathbf{y})}{q_\phi(\mathbf{x})} \right] = \log p(\mathbf{y}) - D_{KL}(q_\phi(\mathbf{x}) \| p(\mathbf{x}|\mathbf{y})) \quad (2)$$

Considering that $p(\mathbf{y})$ is a constant, it is straightforward to see that maximizing the ELBO w.r.t. ϕ is equivalent to minimizing D_{KL} , which is the ultimate objective of the inference process, and, since $D_{KL} \geq 0$, the ELBO lower bounds $p(\mathbf{y})$. What remains is to choose a family of distributions for $q_\phi(\mathbf{x})$. A popular choice is the mean-field family, which assumes that $q_\phi(\mathbf{x})$ factorizes as $q_\phi(\mathbf{x}) = \prod_j q_{\phi_j}(x_j)$ where x_j is a subset of $\mathbf{x}$. This assumption leads to the coordinate ascent VI (CAVI) method [2, 14] where each $q_{\phi_j}(x_j)$ is optimally updated iteratively by:

$$\log q_{\phi_j}(x_j)^* \stackrel{\pm}{=} \mathbb{E}_{-x_j} [\log p(\mathbf{x}, \mathbf{y})] , \quad (3)$$

where $\mathbb{E}_{-x_j}[\cdot]$ denotes the expectation over all latent variables but x_j and $\stackrel{\pm}{=}$ symbolizes equality up to additive constants. When simplifying 3 for a specific model, we might be able to recognize the distribution of $q_{\phi_j}(x_j)$ along with the optimal ϕ_j . In the particular case where the full conditional, $p(x_j|y, x_{-j})$, is in the exponential family, for conjugate priors w.r.t. $p(y, x_{-j}|x_j)$, the corresponding $q^*(x_j)$ will be in the same family as the prior $p(x_j)$ [9].

Before we divulge in the derivation of $q_{\phi}(\mathbf{x})$ of VICTree, we first describe the proposed model $p(\mathbf{x}, \mathbf{y})$ in the next section.

3 Proposed model

This section describes our proposed model and the related variables under analysis. The model can be split into two modules:

- *CopyTree*: a tree-structured Markov model that generates CNPs with horizontal and hierarchical dependencies, and
- the mixture emission model, an emission model which associates the observations to the hidden states of CopyTree by a mixture over the tree nodes (hereon referred to as *clones*).

In Section 4 we explain how variational inference is used for estimating the posterior distribution of the model.

3.1 The CopyTree model

We introduce a tree-structured Markov Model (MM) for latent CNPs linked by parent-child dependencies within a given tree topology, T , containing arcs $A(T)$. Given an arc, $uv \in A(T)$ the CN at site m for clone v , C_m^v , depends not only on C_{m-1}^v , but also on the parent clone’s CNs at site m and $m-1$, C_m^u and C_{m-1}^u . Such dependencies are depicted in Figure 2a. We can see this as a set of hierarchically interlaced Markov models where the transition probability from one CN state to the subsequent one is a function of four CN states and an arc-specific transition probability, $\varepsilon_{uv} \in [0, 1]$.

We call this function the *CN coherence function* (4) and it models the CN evolution both along the clone CNP and over the ancestry according to:

$$h_{\varepsilon_{uv}}(C_m^v | C_m^u, C_{m-1}^v, C_{m-1}^u) = \begin{cases} 1 - \varepsilon_{uv}, & \text{if } C_m^v - C_{m-1}^v = C_m^u - C_{m-1}^u \\ \varepsilon_{uv}/(A-1), & \text{if } C_m^v - C_{m-1}^v \neq C_m^u - C_{m-1}^u \end{cases}, \quad (4)$$

where A is the maximum CN. If ε_{uv} is small, the CN coherence function formalizes the intuition that if we observe a CN change in the CNP of u , then we are likely to observe the same CN change in u ’s child v . Note, the state $C_{m-1}^u = 2, C_m^u = 3$ and $C_{m-1}^v = 4, C_m^v = 5$ is treated as coherent despite the

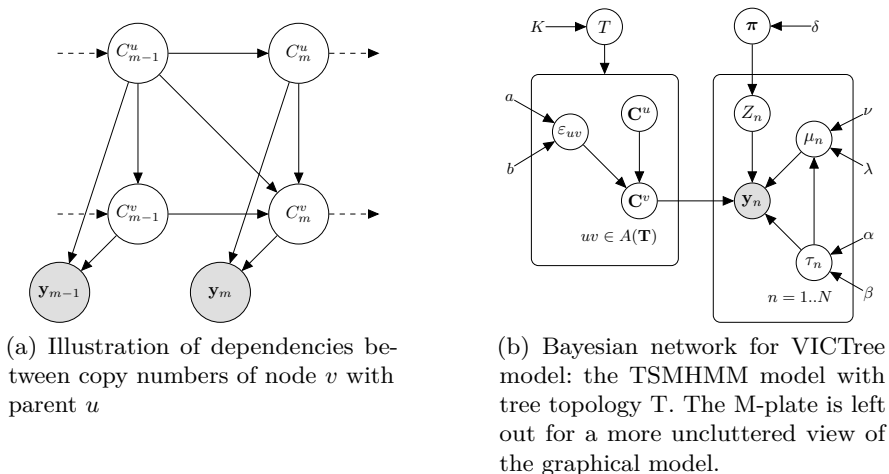


Fig. 2: Variables dependency graphs.

evident difference in CN between these two clones, e.g. between C_m^u and C_m^v . This may at first seem like a model-misspecification, however, this difference will be reflected in an incoherence in a preceding bin, i.e. a bin $m' < m - 1$, of the CNPs.

We construct a first simple model by treating all CN changes that do not satisfy the CN coherence condition as equal, similar to the Jukes-Cantor model [15] in classical phylogeny. Furthermore, ϵ_{uv} can be interpreted as a distance between the two clones, indeed we will refer to it as *arc distance*. For what concerns the edge cases, the root node is set as a healthy clone with constant CN i.e. $C_m^r = 2$ for all m , and the initial state probability for any internal node depends only on the initial state of the parent node.

This fully describes the generative process behind the hidden states of our TSMHMM. To associate the hidden states with observed data, we turn the emission model in the next section.

3.2 The mixture emission model

We relate our observations, $\mathbf{Y}$, to the latent CopyTree state by a Mixture Hidden Markov Model (MHMM) [11, 28], inspired by the MHMM with Gaussian emissions used in [29], with cell-to-clone assignments $\mathbf{Z}$, concentration parameter $\boldsymbol{\pi}$, cell baseline $\boldsymbol{\mu}$ and emission precision $\boldsymbol{\tau}$. More specifically, for cell n assigned to clone k through $\mathbf{Z}$, we let the corrected reads, $Y_{mn} | Z_n = k, \mu_n, \tau_n, C_m^k \sim \mathcal{N}(\mu_n \cdot C_m^k, \tau_n^{-1})$, which we will refer to as *corrected reads*, since they are assumed to be corrected for GC content and normalized so that 2 is the neutral CN. GC correction is a pre-processing step that

corrects for the read bias of low GC content regions and transforms the data from natural to real numbers, thus motivating a continuous distribution for the observational model.

3.3 Tree structured MHMM

The combined statistical model is a Tree-structured MHMM for a fixed tree T . Together with a uniform prior distribution over the finite set of all possible tree topologies $\mathcal{T}_K$ for a fixed number of nodes K , this completes our generative model, $p(\mathbf{Y}, \mathbf{C}, \boldsymbol{\varepsilon}, \mathbf{Z}, \boldsymbol{\pi}, T, \boldsymbol{\mu}, \boldsymbol{\tau})$. The Bayesian network for the full generative model can be seen in figure 2b and the corresponding distributions are:

$$\begin{aligned} p(\mathbf{Y}_{mn} | C_m^v, Z_n = k, \mu_n, \tau_n) &= \mathcal{N}(\mu_n \cdot C_m^k, \tau_n^{-1}), \\ p(\mathbf{C} | T, \boldsymbol{\varepsilon}) &= \text{CopyTree}(\mathbf{C}, T, \boldsymbol{\varepsilon}), \quad p(\mathbf{Z} | \boldsymbol{\pi}) = \text{Categorical}(\boldsymbol{\pi}), \\ p(\mu_n, \tau_n) &= \text{NormalGamma}(\nu, \lambda, \alpha, \beta), \\ p(\boldsymbol{\pi}) &= \text{Dirichlet}(\boldsymbol{\delta}), \quad p(\varepsilon_{uv}) = \text{Beta}(a, b), \quad p(T) = U(\mathcal{T}_K), \end{aligned} \quad (5)$$

Where $\text{CopyTree}(\mathbf{C}, T, \boldsymbol{\varepsilon})$ represents the joint distribution of the interleaved Markov model described in Section 3.1, which depends on the topology by means of its arcs, and on the arcs' distances $\boldsymbol{\varepsilon}$. We now turn our attention to the framework for performing joint Bayesian inference on the model.

4 VICTree

In the Bayesian setting, the main goal is to infer the posterior distribution, that is the probability of the model parameters given the observation, as mentioned in Section 2. We assume the following factorization of the variational distribution:

$$q(\mathbf{C}, \mathbf{Z}, T, \boldsymbol{\Psi}) = q(T)q(\boldsymbol{\pi}) \prod_{uv \in A(\mathcal{G})} q(\varepsilon_{uv}) \prod_{n=1}^N q(\mu_n, \tau_n) q(Z_n) \prod_{u=1}^K q(\mathbf{C}^u), \quad (6)$$

where $\mathcal{G}$ is the fully connected directed graph with K nodes and $A(\mathcal{G})$ is the set of directed arcs, which does not contain self connections. With abuse of notation, each q over different variables has unique learnable parameters, which are left implicit for uncluttered notation.

Unsurprisingly, as our variational updates for $\mathbf{Z}, \boldsymbol{\pi}, \boldsymbol{\varepsilon}, \boldsymbol{\mu}, \boldsymbol{\tau}$ follow the conditional conjugacy mentioned in Section 2, our CAVI updates become:

$$\begin{aligned} q^*(Z_n) &= \text{Categorical}(\tilde{\gamma}_n), \quad q^*(\boldsymbol{\pi}) = \text{Dirichlet}(\tilde{\boldsymbol{\delta}}), \\ q^*(\varepsilon_{uv}) &= \text{Beta}(\tilde{a}_{uv}, \tilde{b}_{uv}), \quad q^*(\mu_n, \tau_n) = \text{NormalGamma}(\tilde{\nu}_n, \tilde{\lambda}_n, \tilde{\alpha}_n, \tilde{\beta}_n) \end{aligned} \quad (7)$$

where the expressions of the variational parameters, $\tilde{\gamma}_n, \tilde{\delta}, \tilde{a}_{uv}, \tilde{b}_{uv}, \tilde{\nu}_n, \tilde{\lambda}_n, \tilde{\alpha}_n, \tilde{\beta}_n$ can be found in the Supplementary Material, together with their derivations.

The updates of $q(T)$ and $q(\mathbf{C})$, however, are less orthodox. For $q(\mathbf{C})$, the CAVI update can be simplified to the form:

$$\log q^*(\mathbf{C}^v) \stackrel{\pm}{=} \mathbb{E}_T \left[\mathbb{E}_{C^{-v}Z\Psi} [\boldsymbol{\eta}_T(v)] + \sum_{w \in \text{ch}(v, T)} \mathbb{E}_{C^{w\varepsilon}} [\boldsymbol{\alpha}(w)] \right] \mathbf{t}(\mathbf{C}^v) \quad (8)$$

where, $\boldsymbol{\eta}_T(v)$ is the vector of natural parameters associated to the sufficient statistic $\mathbf{t}(\mathbf{C}^v)$ of node v , $\boldsymbol{\alpha}_T(w)$ is the vector of natural parameters viewed from child w of v , which has the same form of $\boldsymbol{\eta}_T(v)$ thanks to conjugacy. $\mathbf{t}(\mathbf{C}^v)$ is the sufficient statistic of the CNP at node v . These terms can be seen as variational messages coming from both parent and children of node v as in [33] (see Supplementary Material, Section S2). The outermost expectation value w.r.t. T is non-trivial and must be approximated, e.g. using a Monte Carlo sampling. Appealingly, the optimal $q(\mathbf{C})$ for the TSMHMM corresponds to a distinct non-homogeneous Markov chain for each clone, where the parameter update balances two terms; one enforcing the CN evolution w.r.t. the CopyTree prior and the other forcing the distribution to adapt the observed data.

A detailed derivation and description of 8 can be found in Section S3.1.

The CAVI update of $q(T)$ simplifies as:

$$\log q^*(T) \stackrel{\pm}{=} \mathbb{E}_{C^r} [\log p(\mathbf{C}^r)] + \sum_{uv \in A(T)} \mathbb{E}_{C^v C^u \varepsilon} [\log p(\mathbf{C}^v | \mathbf{C}^u)] + \log p(T) \quad (9)$$

which is not on the form of some known distribution. To handle this we adopt a similar procedure as in [18]; we construct a fully connected graph, $\mathcal{G}$ and let the associated weight matrix, $W(\mathcal{G})$, be determined by the second term of eq. 9 (Note: the first and third term in 9 are constant over trees), allowing us to evaluate sampled trees on $q(T)$. Whenever an update requires an expectation w.r.t. T , e.g. in eq. 8, we employ IS [17], using our novel sampler LARS as proposal distribution, $g(T)$. This concludes our VI framework, the pseudocode for the algorithm can be found in Algorithm 1.

4.1 LARS

Here, we describe our IS proposal distribution $g(T)$, a distribution over labeled arborescences of $W(\mathcal{G})$, induced by our LARS algorithm. Ideally, we would sample a tree with respect to its normalized weight using the matrix tree theorem. However, this is costly, instead we prescribe the Diaconis heuristic for the Markov chain tree theorem [3, 6], by replacing the sum of tree weights with the maximum spanning tree (MST) weight. Consider the directed graph G with arcs $A(G)$. For any subset of arcs $S \subseteq A(G)$ and arc $e \in A(G)$, let

$A(S, e)$ be the set of arborescences of G that contains both the arcs of S and e and, moreover, let $A'(S, e)$ be the set of arborescences of G that contain the arcs of S but not e . Let $e_1, \dots, e_L$ be a random permutation of the arcs of G . Then the next state of the Markov chain tree is obtained by selecting the arc e_i with probability

$$\theta_i = \frac{\max_{T \in A(S_i, e_i)} \tilde{q}(T)}{\max_{T \in A(S_i, e_i)} \tilde{q}(T) + \max_{T \in A'(S_i, e_i)} \tilde{q}(T)}.$$

The maximization problem is solved by Edmonds' algorithm [8]. The proposal distribution from which we sample is then constructed as a product of Bernoulli trials, i.e., $g(T) = \prod_{i=1}^{K-1} \theta_i$, if e_i is accepted. Further details of the algorithm is described in Section S6 of the Supplementary Material.

4.2 The split-and-merge algorithm

Inference of mixture models, e.g., in the context of VI, MCMC or the EM-algorithm, is susceptible to mixture component degeneracy, resulting in convergence to poor local optima. This section examines the origin of this issue in CN based clustering and an algorithm to counteract the phenomena. The algorithm is designed to address domain specific issue of shared hidden states between the clones; segments of the CNPs of each clone may be shared between all clones, *clonal profile*, between two or more clones, *semi-clonal profile*, or fully clone specific, *subclonal profile*.

We will illustrate how this effects our VI optimization with an extreme case: one of the K clusters, e.g. the k^{th} cluster, adapts $q(\mathbf{C}^k)$ to the signal of the clonal profile in the data before the remaining clusters. Consequently, each cell n will be assigned to the k^{th} cluster with a probability $q(Z_n)$ which depends on the ratio between the clonal profile and subclonal profile signal in the observations. If the ratio is high, then optimization converges to a degenerate local optima where $q(Z_n = k) = 1$ for all n and $q(\mathbf{C}^k)$ learns the clonal profile and averages the subclonal part over all cells. As $q(Z_n = j) = 0$ for all $j \neq k$ and $q(\mathbf{C}^j)$ is updated w.r.t. $q(Z_n = j)$, the result is $K - 1$ degenerate components.

To escape these local optima, we adapt a split-and-merge algorithm for VI [4] to the clonal clustering context. We merge clusters i and j based on the distance between $q(\mathbf{C}^i)$ and $q(\mathbf{C}^j)$. We split a cluster i into a degenerated cluster j if there exists a group of cells within i , that when updating $q(\mathbf{C}^j)$ on that group, maximize the ELBO w.r.t. to all other possible splits. More details are given in section S7.

4.3 ELBO calculation

To monitor the status of the inference procedure and assess the fitness of the current posterior approximation we use the ELBO presented in Section 2. In

the VICTree setting, the ELBO is derived as follows:

$$\begin{aligned} \text{ELBO}_q &= \mathbb{E}_q [\log p(\mathbf{Y}, \mathbf{C}, \mathbf{Z}, T, \boldsymbol{\Psi})] - \mathbb{E}_q [\log q(\mathbf{C}, \mathbf{Z}, T, \boldsymbol{\Psi})] \\ &= \mathbb{E}_q [\log p(\mathbf{Y}|\mathbf{C}, \mathbf{Z}, \boldsymbol{\Psi}) + \log p(\mathbf{C}|\mathbf{Z}, T, \boldsymbol{\Psi}) + \log p(\mathbf{Z}|\boldsymbol{\Psi}) \\ &\quad + \log p(\boldsymbol{\Psi}) + \log p(T)] + H(q). \end{aligned} \quad (10)$$

Each one of the expectation terms can be computed separately and such derivations are shown in the Supplementary Material. It should be noted that any expectation over the tree topology space is approximated as explained in Section 4.1, therefore it is useful to view each such expectation as $\mathbb{E}_{T,X} [f(T, X)] = \mathbb{E}_T [\mathbb{E}_X [f(T, X)]]$ where f is some function of T and a generic r.v. X .

5 Experimental results

In order to evaluate our method we conduct experiments on both synthetic data, to get advantage of a ground-truth, and real data, so to assess the practical efficacy of inference. We compare our results on real data against HMMCopy [22] (in copy-number calling) and CONET [25] (in copy-number calling and clonal tree inference).

5.1 Simulated data

Data have been simulated from the generative model presented in Section 3. We show the results of two distinct experiments: one to assess the overall performance of VICTree and provide a benchmark on its downstream analysis capability, and one to assess robustness on the amount of copy-number mutations. In the first experiment, for three sizes of datasets, *small*, *medium*, *large* i.e. respectively $(K, M, N) \in \{(6, 1000, 300), (9, 3000, 600), (12, 5000, 1000)\}$, we compute *adjusted-rand index* (ARI) on cell assignment, copy-number calling *mean absolute deviation* (MAD), and *arc accuracy* on the reconstructed tree (i.e. the proportion of arcs in the MAP tree that also belong to the ground truth tree).

In the second experiment we simulate data of *medium* size with increasing number of mutations along the genome and compute the same measures. Mutations are generated at a higher frequency by multiplying the first parameter of the Beta distribution of ε_{uv} by a factor ≥ 1 , which we denote as *mutation rate*. The results for this experiment are shown in the Supplementary Material S2.

For all these experiments, VICTree has been configured by setting the hyper parameters to fixed values, partly dependent on the size of the datasets, and mainly derived from explorative data analysis on real datasets such as the ones here under study. More specifically, the update *step-size* was set

to 0.3 and LARS sample size $L = 5$. We fix the model priors (as defined in Equation 5) to $\delta = (3, \dots, 3)$, $\nu = 1$, $\lambda = 20 \times M$, $\alpha = 500$, $\beta = 50$, $a = 5$, $b = M$. The value of K was set to the same (normally unknown) value from which the data has been generated. We motivate this choice by showing that by running VICTree with different values of K on synthetic data, the ELBO clearly suggests which value is best for each dataset (see Figure 4).

Our method performs well in all the above mentioned objectives, namely cell clustering, clonal CN calling, and tree reconstruction, achieving perfect score in most of the datasets as shown in Figure 3.

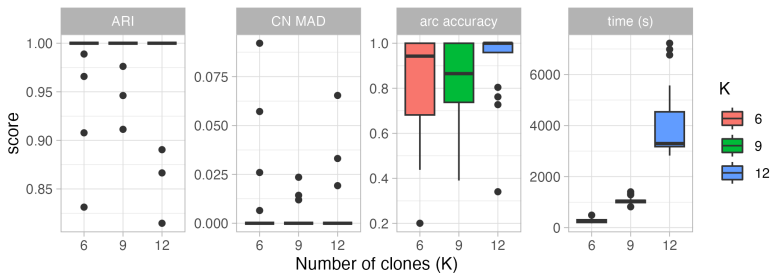


Fig. 3: Scores against ground truth over 20 synthetic datasets for three values of K (number of clones). Clustering performance is quantified by the Adjusted-Rand index while CN calling performance by mean absolute deviation. Arc accuracy is the proportion of arcs in the ground truth tree that are included in the MAP estimate tree. Dots represent outliers. Time refers to the execution of VICTree on one core of a Intel Xeon Gold 6130.

5.2 Multiple Myeloma datasets

We apply VICTree to several datasets of multiple myeloma extracted using the 10x Genomics technology [23]. This section provides a description of the results over the sample MM29, while the results of other two samples (MM03 and MM04) are left in the Supplementary Material together with the comparison to HMMcopy CN calling (see Section S1.4).

In Figure 5 we show the clonal CNP of each cell that is the Viterbi path of $q(\mathbf{C}^v)$, ordered by the estimated MAP cell-to-clone assignment given by the inferred $q(\mathbf{Z})$, together with the inferred MAP tree. Inference has been performed over a number of different K values. We only present the results with $K = 14$ as it is shown to be a fair trade-off between model complexity and ELBO (see Supplementary Figure S6), although there's evidence in support of a larger K .

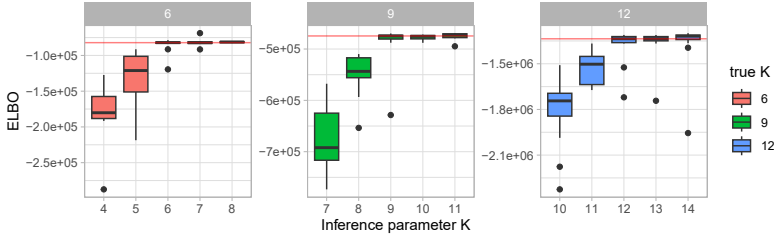


Fig. 4: ELBO plot showing model selection on synthetic data. Each facet shows runs on equally sized datasets (i.e. small, medium, large). On the horizontal axis the number of nodes K chosen for inference. Boxplots show the result over 20 datasets.

5.3 SA501X3F xenograft breast cancer dataset

We run our method on the DLP dataset SA501X3F [35] which features 260 single cells for 18175 bins of width 150kb, setting the maximum number of clones K to 12. VICTree is able to infer the clonal copy number profile, assign cells to the clones and reconstruct a clonal tree (MAP estimate) within an hour⁵, just by providing it the corrected reads matrix without any pre-processing step. Results are shown in Figure 5 and a comparison with CONET [25] CN calling and reconstructed clonal tree is left in the Supplementary Material S3.

6 Discussion

We have presented a framework for fully automated Bayesian inference in clonal tumor phylogenetics, requiring a minimal amount data pre-processing steps and user inputs. To achieve this, we have presented several new modular models and subroutines: CopyTree and the TSMHMM with associated CAVI updates, LARS a rooted tree sampler, and a novel domain specific split-and-merge algorithm, which may be applied independently in the field of tumor phylogenetics. The composite algorithm, VICTree, is demonstrated to infer clonal trees, CNPs and cell-to-clone assignments in high accordance with the ground truth in 5.1 and retrieves similar MAP output to that of HMMCopy in 5.2 with the addition of a distribution over trees explaining the process of CN evolution. In relation to CONET, VICTree infers trees that, while capturing the same evolution process, summarize the single-cells information in a less cluttered fashion (S3). Furthermore, VICTree’s is robust to high

⁵ the time refers to the execution of VICTree on a CPU Intel Core i7-855U

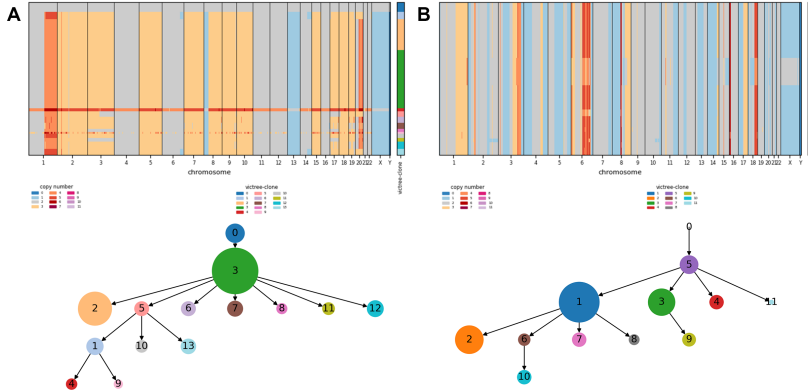


Fig. 5: Inferred clonal copy number profile and MAP estimate of the clonal tree of two real datasets. **A** Sample MM29 (multiple myeloma) is obtained by running VICTree with $K = 14$ while **B** sample SA501X3F (breast cancer) with $K = 12$.

mutation rates in the data S2, a scenario where the user-input required by CONET is cumbersome.

The presented probabilistic model features a large number of dependencies between variables, so to deeply explain the nature of the data from its source, not relying on other intermediate steps or broad assumptions which might not hold in all cases. This substantially increases the complexity of inference (e.g. for a site-specific HMM) and our method alleviates this with the efficiency of variational inference.

Bayesian inference have been already adopted in the field of tumor phylogenetics [19, 25], and while other methods only partially engage in the inference process, often relying on additional tools and approximation, our approach offers a comprehensive Bayesian framework supported by rigorous statistical techniques.

In terms of quality of inference, VICTree demonstrates strong clustering and CN calling capabilities, while simultaneously inferring trees in the vicinity of the true underlying tree. Furthermore, the experiment in Section 5.1 demonstrates VICTree’s robustness to high rates of CN events events, where other methods currently struggle. Also, despite the large set of learned parameters, VICTree reaches convergence in short overall execution time as shown in Figure 3, making it more efficient than traditional MCMC methods.

Regarding the experimental results, we show that LARS is a robust algorithm for sampling labeled arborescences from a weighted graph. It works effectively alongside the inferred weights in the variational distribution over the tree space. As stated in the introduction, the ELBO’s lower bound on

the marginal likelihood offers a unique tool for model selection in tumor phylogenetics. Indeed, the ELBO is higher for K 's in the vicinity of the true K according to 4. In real data experiments however, such as S4, the ELBO does not plateau with increasing K . In the context of clonal tree reconstruction, we can therefore choose to limit the granularity of clones based on the ELBO trend line, by identifying a trade off between a lower model complexity and a higher fitness to the data.

6.1 Future Work

The modular framework of VICTree facilitates a myriad of independent improvement opportunities. The proposal distribution $g(T)$ could be swapped with a scheme that is more fit for larger K . Furthermore, ensembles or mixtures of variational distributions can be applied to better accommodate for multimodality [20, 21]. The CN evolution model specified in Equation (4) can be improved to explicitly model specific CN events such as whole-genome duplication events. Furthermore, the observational model is unidentifiable, in that it is observationally equivalent under different combinations of μ_n and C_m^v . In light of this, although we still achieve accurate inference by properly setting the model priors, an observational model which includes ploidy (i.e. the average CN of a clone) could be developed so to improve the TSMHMM. Here, the unpacked update equations in Section S3 expedites such framework development.

References

1. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. *J. Mach. Learn. Res.* (2003)
2. Blei, D.M., Kucukelbir, A., McAuliffe, J.D.: Variational inference: A review for statisticians. *Journal of the American Statistical Association* **112**(518), 859–877 (2017). <https://doi.org/10.1080/01621459.2017.1285773>
3. Broder, A.: Generating random spanning trees. In: 30th Annual Symposium on Foundations of Computer Science. pp. 442–447 (1989). <https://doi.org/10.1109/SFCS.1989.63516>
4. Bryant, M., Sudderth, E.: Truly nonparametric online variational inference for hierarchical dirichlet processes. *Advances in Neural Information Processing Systems* **25** (2012)
5. Cayley, A.: The Collected Mathematical Papers of Arthur Cayley. The University Press (1897)
6. Diaconis, P.: Sequential Importance Sampling for Estimating the Number of Perfect Matchings in Bipartite Graphs: An Ongoing Conversation with Laci, pp. 223–233. Springer Berlin Heidelberg, Berlin, Heidelberg (2019). https://doi.org/10.1007/978-3-662-59204-5_6
7. Drummond, A.J., Rambaut, A.: BEAST: Bayesian evolutionary analysis by sampling trees. *BMC Evol. Biol.* **7**, 214 (Nov 2007)
8. Edmonds, J.: Optimum branchings. *J. Res. Natl. Bur. Stand.* **71B**(4), 233 (Oct 1967)
9. Ghahramani, Z., Beal, M.: Propagation algorithms for variational bayesian learning. In: Leen, T., Dietterich, T., Tresp, V. (eds.) *Advances in Neural Information Processing Systems*. vol. 13. MIT Press (2000)
10. Greaves, M., Maley, C.C.: Clonal evolution in cancer. *Nature* **481**(7381), 306–313 (Jan 2012)
11. Helske, S., Helske, J.: Mixture hidden markov models for sequence data: The seqHMM package in R. *J. Stat. Softw.* **88**, 1–32 (Jan 2019)
12. Huelsenbeck, J.P., Ronquist, F.: MRBAYES: Bayesian inference of phylogenetic trees. *Bioinformatics* **17**(8), 754–755 (Aug 2001)
13. Jojic, V., Jojic, N., Meek, C., Geiger, D., Siepel, A., Haussler, D., Heckerman, D.: Efficient approximations for learning phylogenetic HMM models from data. *Bioinformatics* **20 Suppl 1**, i161–8 (Aug 2004)
14. Jordan, M.I., Ghahramani, Z., Jaakkola, T.S., Saul, L.K.: An introduction to variational methods for graphical models. *Mach. Learn.* **37**(2), 183–233 (Nov 1999)
15. Jukes, T.H., Cantor, C.R., et al.: Evolution of protein molecules. *Mammalian protein metabolism* **3**, 21–132 (1969)
16. Kaufmann, T.L., Petkovic, M., Watkins, T.B.K., Colliver, E.C., Laskina, S., Thapa, N., Minussi, D.C., Navin, N., Swanton, C., Van Loo, P., Haase, K., Tarabichi, M., Schwarz, R.F.: MEDICC2: whole-genome doubling aware copy-number phylogenies for cancer evolution. *Genome Biol.* **23**(1), 241 (Nov 2022)
17. Kloek, T., van Dijk, H.K.: Bayesian estimates of equation system parameters: An application of integration by monte carlo. *Econometrica* **46**(1), 1–19 (1978)
18. Koptagel, H., Kviman, O., Melin, H., Safinianaini, N., Lagergren, J.: Vaiphy: a variational inference based algorithm for phylogeny. In: Koyejo, S., Mohamed,

- S., Agarwal, A., Belgrave, D., Cho, K., Oh, A. (eds.) *Advances in Neural Information Processing Systems*. vol. 35, pp. 14758–14770. Curran Associates, Inc. (2022)
19. Kuipers, J., Tuncel, M.A., Ferreira, P., Jahn, K., Beerenwinkel, N.: Single-cell copy number calling and event history reconstruction (Apr 2020), preprint
 20. Kviman, O., Melin, H., Koptagel, H., Elvira, V., Lagergren, J.: Multiple importance sampling elbo and deep ensembles of variational approximations. In: *International Conference on Artificial Intelligence and Statistics*. pp. 10687–10702. PMLR (2022)
 21. Kviman, O., Molén, R., Hotti, A., Kurt, S., Elvira, V., Lagergren, J.: Learning with miselbo: The mixture cookbook. *arXiv preprint arXiv:2209.15514* (2022)
 22. Lai, D., Ha, G., Shah, S.: HMMcopy: Copy number prediction with correction for GC and mappability bias for HTS data (2023). <https://doi.org/10.18129/B9.bioc.HMMcopy>
 23. Lee, H., Ahn, S., Maity, R., Leblay, N., Ziccheddu, B., Truger, M., Chojnacka, M., Cirrincione, A., Durante, M., Tilmont, R., et al.: Mechanisms of antigen escape from bcma-or gprc5d-targeted immunotherapies in multiple myeloma. *Nature Medicine* pp. 1–12 (2023)
 24. Lotfi, S., Izmailov, P., Benton, G., Goldblum, M., Wilson, A.G.: Bayesian model selection, the marginal likelihood, and generalization. In: *International Conference on Machine Learning*. pp. 14223–14247. PMLR (2022)
 25. Markowska, M., Cakała, T., Miasojedow, B., Aybey, B., Juraeva, D., Mazur, J., Ross, E., Staub, E., Szczurek, E.: CONET: copy number event tree model of evolutionary tumor history for single-cell data. *Genome Biol.* **23**(1), 128 (Jun 2022)
 26. Moretti, A.K., Zhang, L., Naesseth, C.A., Venner, H.K., Blei, D.M., Peer, I.: Variational combinatorial sequential monte carlo methods for bayesian phylogenetic inference. In: *Conference on Uncertainty in Artificial Intelligence* (2021)
 27. Nowell, P.C.: The clonal evolution of tumor cell populations. *Science* **194**(4260), 23–28 (Oct 1976)
 28. van de Pol, F., Langeheine, R.: Mixed markov latent class models. *Sociol. Methodol.* **20**, 213–247 (1990)
 29. Safinianaini, N., de Souza, C.P.E., Roth, A., Koptagel, H., Toosi, H., Lagergren, J.: Copymix: Mixture model based single-cell clustering and copy number profiling using variational inference. *bioRxiv* (2023). <https://doi.org/10.1101/2020.01.29.926022>, preprint
 30. Salehi, S., Dorri, F., Chern, K., Kabeer, F., Rusk, N., Funnell, T., Williams, M.J., Lai, D., Andronescu, M., Campbell, K.R., McPherson, A., Aparicio, S., Roth, A., Shah, S.P., Bouchard-Côté, A.: Cancer phylogenetic tree inference at scale from 1000s of single cell genomes. *Peer Community Journal* **3**, e63 (2023). <https://doi.org/10.24072/pcjournal.292>, <https://peercommunityjournal.org/articles/10.24072/pcjournal.292/>
 31. Siepel, A., Haussler, D.: Combining phylogenetic and hidden markov models in biosequence analysis. In: *Proceedings of the seventh annual international conference on Research in computational molecular biology*. pp. 277–286. RE-COMB '03, Association for Computing Machinery, New York, NY, USA (Apr 2003)

32. Wang, F., Wang, Q., Mohanty, V., Liang, S., Dou, J., Han, J., Minussi, D.C., Gao, R., Ding, L., Navin, N., Chen, K.: MEDALT: single-cell copy number lineage tracing enabling gene discovery. *Genome Biol.* **22**(1), 70 (Feb 2021)
33. Winn, J., Bishop, C.M., Jaakkola, T.: Variational message passing. *J. Mach. Learn. Res.* (2005)
34. Zaccaria, S., Raphael, B.J.: Characterizing allele- and haplotype-specific copy numbers in single cells with CHISEL. *Nat. Biotechnol.* **39**(2), 207–214 (Feb 2021)
35. Zahn, H., Steif, A., Laks, E., Eirew, P., VanInsberghe, M., Shah, S.P., Aparicio, S., Hansen, C.L.: Scalable whole-genome single-cell library preparation without preamplification. *Nat. Methods* **14**(2), 167–173 (Feb 2017)
36. Zhang, C.: Improved variational bayesian phylogenetic inference with normalizing flows. In: *Proceedings of the 34th International Conference on Neural Information Processing Systems*. pp. 18760–18771. No. Article 1575 in NIPS’20, Curran Associates Inc., Red Hook, NY, USA (Dec 2020)
37. Zhang, C., Matsen, IV, F.A.: Variational Bayesian Phylogenetic Inference (2019)

Supplementary Material

S1 Additional experiments

S1.1 Synthetic data simulation

S1.2 Robustness on high mutation rate

S1.3 Breast cancer sample

S1.4 Multiple Myeloma samples

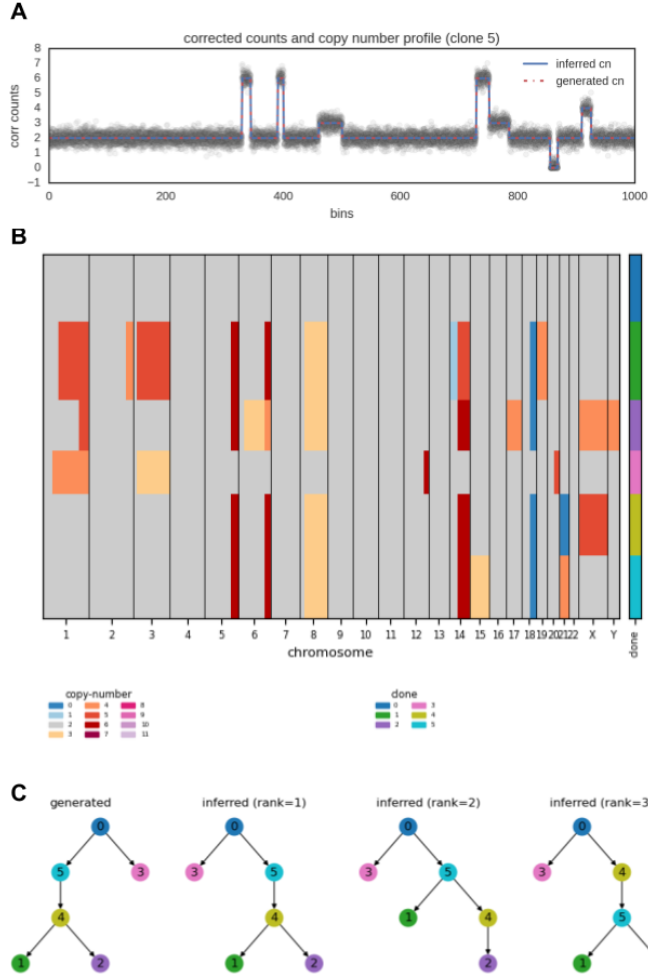


Fig. S1: Example of synthetic dataset with $K = 6$ clones. **A** Corrected counts of cells assigned to clone 5 with ground-truth and inferred copy number profile (CN-MAD = 0.0). **B** Clonal copy number profile with cells on the vertical axis, arranged by inferred clonal assignment (ARI = 1.0). **C** Generated (ground-truth) tree and set of inferred trees, ranked by importance in the final sample from $q(T)$ using LARS.

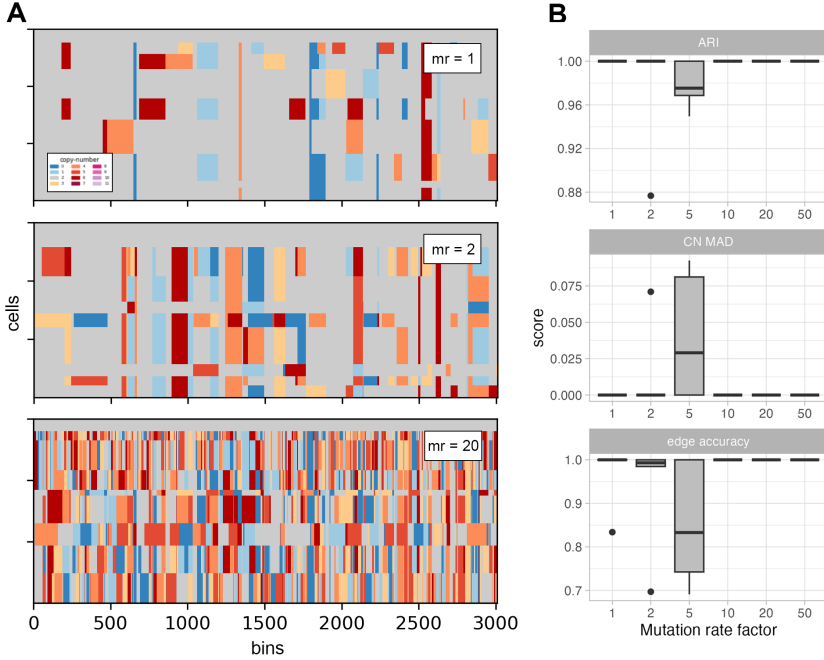


Fig. S2: Illustration on the robustness over increasing rate of copy number mutation. **A** Simulated copy number evolution of 9 clones with three different values for the mutation rate parameter: 1 (default), 2 and 20. **B** ARI (clustering), CN MAD (copy number calling) and edge accuracy scores against ground truth on 10 synthetic datasets for 6 different values of mutation rate parameter.

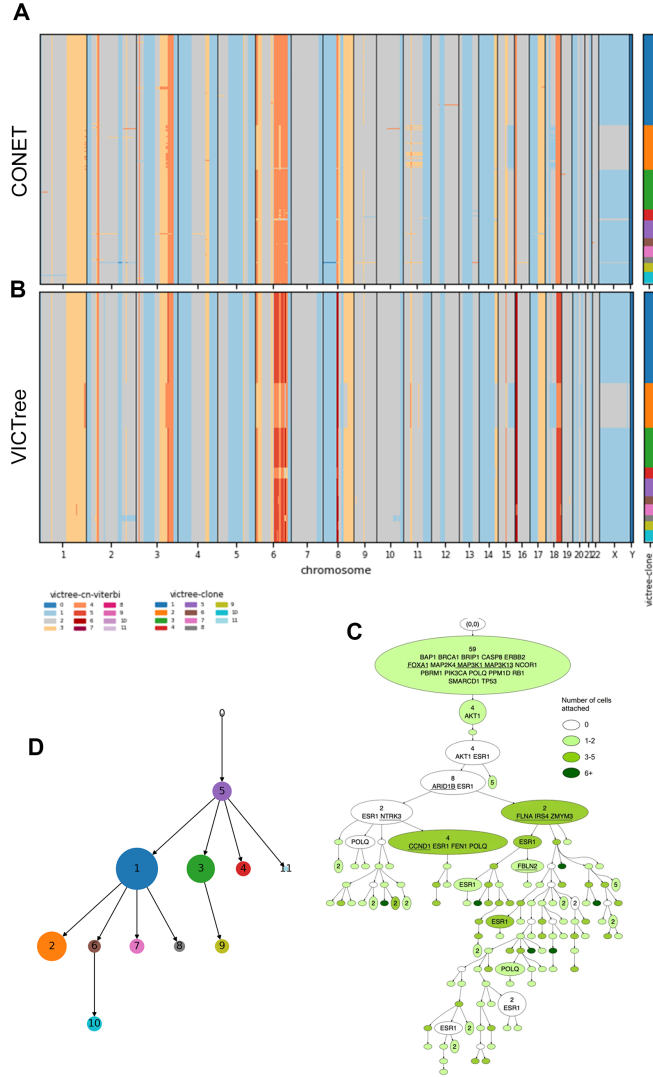


Fig. S3: Sample of xenograft breast cancer SA501X3F. Features 260 single cells for 18175 bins of width 150kb. **A** CONET copy-number calling (sorted and annotated by VICTree clustering). **B** VICTree clonal copy-number calling and clustering of cells annotated on the right. **C** MAP tree from running VICTree with $K = 14$, where each nodes has size proportionally to the clone size according to VICTree result, and color matching the legend on the plots above. **D** CONET tree as shown in [25].

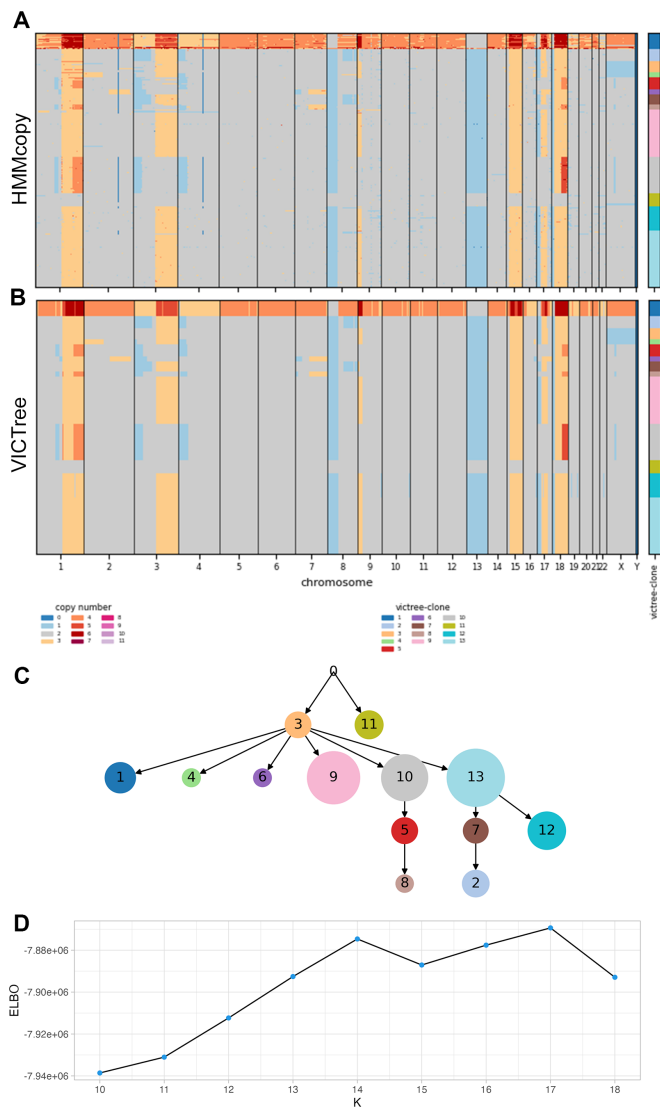


Fig.S4: Sample of multiple myeloma MM03. Features 2208 single cells for 5298 bins of width 500kb. **A** HMMcopy copy-number calling. **B** VICTree clonal copy-number calling and clustering of cells annotated on the right. **C** MAP tree from running VICTree with $K = 14$, where each nodes has size proportionally to the clone size according to VICTree result, and color matching the legend on the plots above. **D** Plot of the ELBO for runs with several values of K .

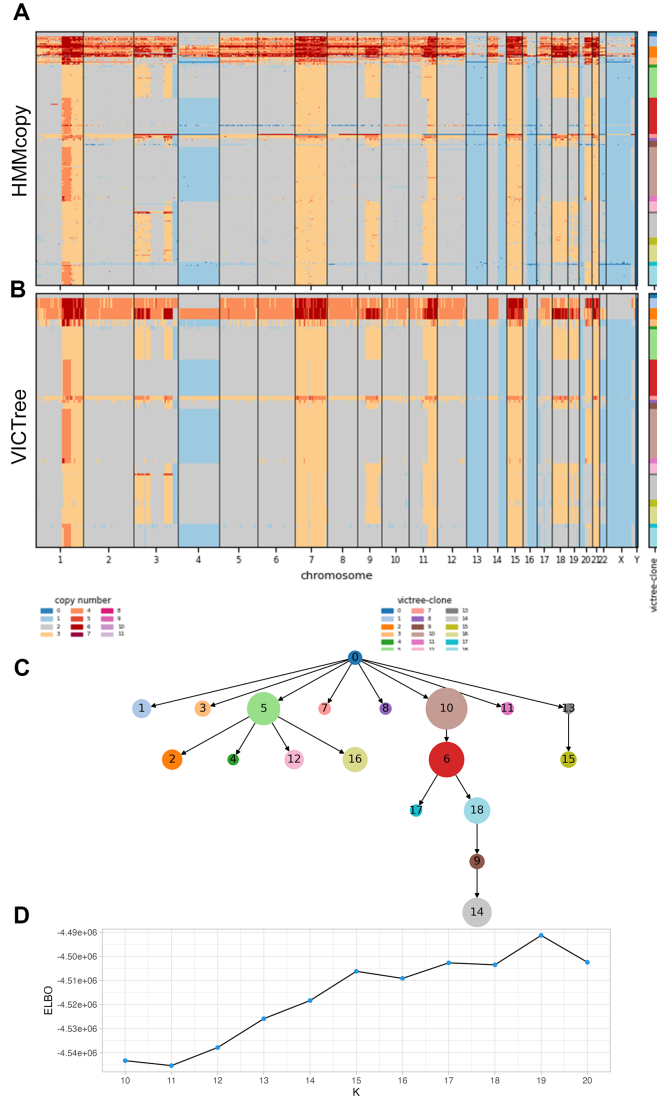


Fig.S5: Sample of multiple myeloma MM04. Features 1003 single cells for 5286 bins of width 500kb. **A** HMMcopy copy-number calling. **B** VICTree clonal copy-number calling and clustering of cells annotated on the right. **C** MAP tree from running VICTree with $K = 19$, where each nodes has size proportionally to the clone size according to VICTree result, and color matching the legend on the plots above. **D** Plot of the ELBO for runs with several values of K .

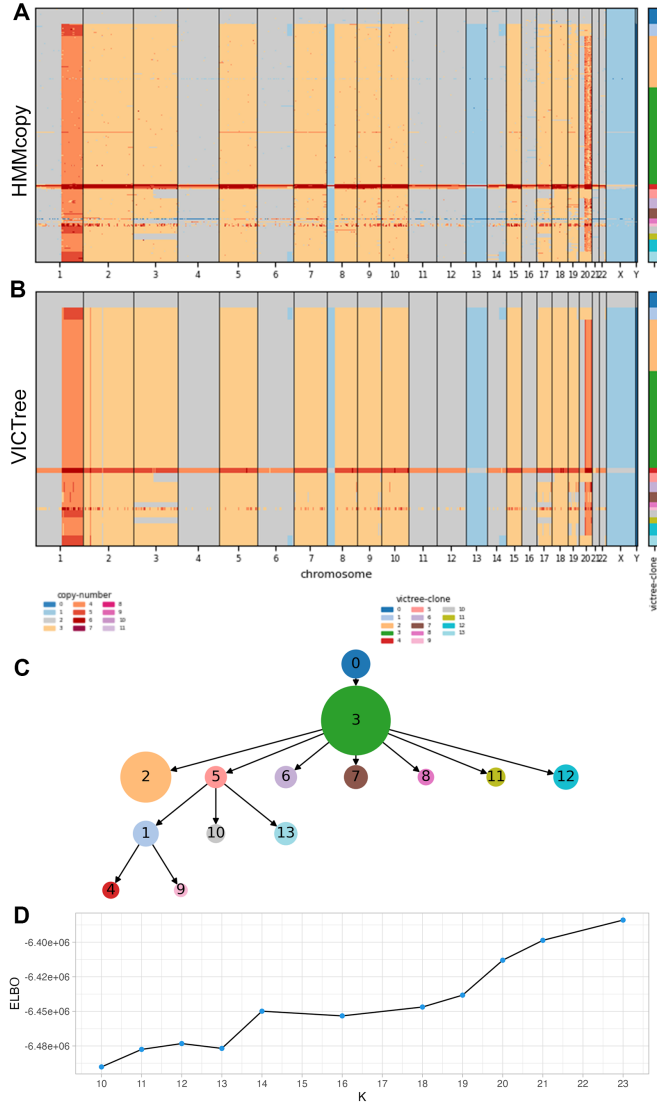


Fig.S6: Sample of multiple myeloma MM29. Features 1616 single cells for 5285 bins of width 500kb. **A** HMMcopy copy-number calling. **B** VICTree clonal copy-number calling and clustering of cells annotated on the right. **C** MAP tree from running VICTree with $K = 14$, where each nodes has size proportionally to the clone size according to VICTree result, and color matching the legend on the plots above. **D** Plot of the ELBO for runs with several values of K .

S2 CopyTree in exponential form

In the following sections we prove that CopyTree model belongs to the exponential family of distributions. We first introduce the notation by showing how standard non-homogeneous HMMs can be written in exponential form and then we extend the proof for the tree-structured HMM which we call CopyTree. The derivations in this sections will also be useful for Section S3 and Section S4 since the notation is first introduced here.

S2.1 HMM in exponential form

Let M be the length of a discrete Markov chain and Σ be the finite discrete set of values for the Markov chain states (alphabet). Let $\mathbf{X}$ be the $|\Sigma| \times M$ matrix representing a sequence $\mathbf{C}$ where each column is the one-hot-encoding of a state, i.e.

$$X_{im} = \mathbb{1}(C_m = i)$$

Also, we introduce two mappings which are useful for parameters vectorization:

$$\begin{aligned} \iota(m, j) &= (m - 1)|\Sigma| + j \\ \kappa(m, j, j') &= (m - 1)|\Sigma|^2 + (j - 1)|\Sigma| + j', \end{aligned} \tag{S1}$$

with j, j' taking values in Σ . Note that the vectors are linear and quadratic functions of the alphabet length.

Let us denote with x_m the m -th column of X , which is the one-hot-encoding for the state at site m , and with Y_m the vector of observations for site m . Then we can write the joint probability as

$$\begin{aligned} P(\mathbf{X}, \mathbf{Y}) &= P(Y_1|x_1)P(x_1) \prod_{m=2}^M P(Y_m|x_m)P(x_m|x_{m-1}) \\ &= P(Y_1|x_1) \prod_{m=2}^M P(Y_m|x_m) \prod_{i \in \Sigma} \pi_{\iota(1,i)}^{X_{i1}} \prod_{j \in \Sigma} \Phi_{\kappa(m,i,j)}^{X_{i,m-1}X_{jm}} \end{aligned} \tag{S2}$$

where $\boldsymbol{\pi}$ and $\boldsymbol{\Phi}$ are the initial and transition probability vectors respectively, with indices provided by the mapping just described ⁶. For simplicity, we can define the pairwise state column vector $\tilde{x}$ which is indexed by κ and has values given by

$$\tilde{x}_{\kappa(m,i,j)} := X_{i,m-1}X_{j,m} = \mathbb{1}(C_{m-1} = i, C_m = j).$$

⁶ $\boldsymbol{\pi} = (\pi_1, \dots, \pi_{|\Sigma|})$ is actually a 1-D vector, which means there is no need to use ι mapping

We can then re-write part of (S2) as exponential functions of the parameters.

$$\begin{aligned}
P(x_1) &= \prod_{i \in \Sigma} \pi_i^{X_{i1}} \\
&= \exp\left(\sum_{i \in \Sigma} X_{i1} \log \pi_i\right) \\
&= \exp(\log \boldsymbol{\pi}^\top x_1)
\end{aligned} \tag{S3}$$

$$\begin{aligned}
\prod_{m=2}^M P(x_m | x_{m-1}) &= \prod_{m=2}^M \prod_{i \in \Sigma} \prod_{j \in \Sigma} \Phi_{\kappa(m,i,j)}^{X_{i,m-1} X_{j,m}} \\
&= \exp\left(\sum_{m=2}^M \sum_{i \in \Sigma} \sum_{j \in \Sigma} X_{i,m-1} X_{j,m} \log \Phi_{\kappa(m,i,j)}\right) \\
&= \exp(\log \boldsymbol{\Phi}^\top \tilde{x})
\end{aligned} \tag{S4}$$

where $\log \boldsymbol{\pi} := (\log \pi_1, \dots, \log \pi_{|\Sigma|})$. Thus, all together in matrix notation:

$$P(\mathbf{X}) = \exp\left(\begin{bmatrix} \log \boldsymbol{\pi} \\ \log \boldsymbol{\Phi} \end{bmatrix}^\top \begin{bmatrix} x_1 \\ \tilde{x} \end{bmatrix}\right). \tag{S5}$$

Definition 1 (Exponential family form).

Let X be a random vector whose distribution belongs to the exponential family and $\boldsymbol{\theta}$ denote its parameters vector. Then the density function of X is

$$f_X(\mathbf{x} | \boldsymbol{\theta}) = \exp[\eta_X(\boldsymbol{\theta}) \cdot \mathbf{t}_X(\mathbf{x}) - A_X(\boldsymbol{\theta}) + B_X(\mathbf{x})], \tag{S6}$$

where $\mathbf{t}_X(\mathbf{x})$ is the sufficient statistic vector, $\eta_X(\boldsymbol{\theta})$ is the natural parameter vector, $A_X(\boldsymbol{\theta})$ is the log partition and $B_X(\mathbf{x})$ is the log base measure.

If we assume that the emission probability can be written in exponential form (e.g. product of Gaussians with mean defined by $\mathbf{X}$) and that it is conjugate to $P(\mathbf{X})$, we can write $P(\mathbf{Y} | \mathbf{X})$ in terms of $\mathbf{t}_X(\mathbf{X})$

$$\begin{aligned}
P(\mathbf{Y} | \mathbf{X}) &= \prod_{m=1}^M P(Y_m | x_m) \\
&= \exp\left(\sum_{m=1}^M [\eta_{Y_m}(x_m)^\top \mathbf{t}_{Y_m}(Y_m) - A_{Y_m}(x_m) + B_{Y_m}(Y_m)]\right) \\
&= \exp(\eta_Y(\mathbf{X})^\top \mathbf{t}_Y(\mathbf{Y}) - A_Y(\mathbf{X}) + B_Y(\mathbf{Y})) \\
&= \exp(\eta_{YX}(\mathbf{Y})^\top \mathbf{t}_X(\mathbf{X}) - A_{YX}(\mathbf{Y}) + B_{YX}(\mathbf{X})).
\end{aligned} \tag{S7}$$

This reparametrization is also shown in [33] when presenting the Conjugate-Exponential models. It is worth noting that $P(\mathbf{Y}|\mathbf{X})$ can be thought as a contribution to the likelihood of X . Note that $\mathbf{t}_X$ is a concatenation of the flattened version of $\mathbf{X}$ and the $\tilde{x}$ vector previously introduced. The natural parameters vector $\eta_{YX}(\mathbf{Y})$ depends on the distribution posit on the observations. More specifically, $\eta_{YX}(\mathbf{Y})_{i(m,i)} = \log p(Y_m|C_m = i)$, function of Y with fixed C_m as it is multiplied by $X_{i,m}$, therefore “activated” or “deactivated”.

Putting altogether, after reparametrizing as in (S7) and rearranging the terms at the exponent, we obtain the complete formulation

$$\begin{aligned} P(\mathbf{X}, \mathbf{Y}) &= P(\mathbf{X})P(\mathbf{Y}|\mathbf{X}) \\ &= \exp \left(\begin{bmatrix} \log \boldsymbol{\pi} \\ \log \boldsymbol{\Phi} \end{bmatrix}^\top \begin{bmatrix} x_1 \\ \tilde{x} \end{bmatrix} + \eta_X(\mathbf{Y})^\top \mathbf{t}_{YX}(\mathbf{X}) - A_{YX}(\mathbf{Y}) - B_{YX}(\mathbf{X}) \right) \\ &= \exp \left(\begin{bmatrix} \log \boldsymbol{\pi} \\ \eta_{YX}(\mathbf{Y}) \\ \log \boldsymbol{\Phi} \end{bmatrix}^\top \begin{bmatrix} x_1 \\ \mathbf{t}_X(\mathbf{X}) \\ \tilde{x} \end{bmatrix} - A_{YX}(\mathbf{Y}) - B_{YX}(\mathbf{X}) \right). \quad (\text{S8}) \end{aligned}$$

Since $\eta_{YX}(\mathbf{Y})$ needs to match the length of $\mathbf{t}_X(\mathbf{X})$ i.e. $M \times |\Sigma| + M \times |\Sigma|^2$, we set $\eta_{YX}(\mathbf{Y})_{\kappa(m,j,j')} = \mathbf{0}$. We further notice that x_1 is just the first column of $\mathbf{X}$, hence the first $|\Sigma|$ elements of the flattened vector $\mathbf{t}_X(\mathbf{X})$. This means that we can group x_1 by incorporating $\log \boldsymbol{\pi}$ with $\eta_{YX}(\mathbf{Y})$. We split the sufficient statistic vector into $t_1(\mathbf{X}), t_2(\mathbf{X})$ and the natural parameter vector into $\eta_1(\mathbf{Y}, \boldsymbol{\theta}, \boldsymbol{\pi}), \eta_2(\boldsymbol{\Phi})$

$$P(\mathbf{X}, \mathbf{Y}) = \exp \left(\begin{bmatrix} \eta_1(\mathbf{Y}, \boldsymbol{\theta}, \boldsymbol{\pi}) \\ \eta_2(\boldsymbol{\Phi}) \end{bmatrix}^\top \begin{bmatrix} t_1(\mathbf{X}) \\ t_2(\mathbf{X}) \end{bmatrix} - A_{YX}(\mathbf{Y}) - B_{YX}(\mathbf{X}) \right) \quad (\text{S9})$$

where $\boldsymbol{\theta}$ is the set of extra parameters required in the emission distribution defined on the observations $\mathbf{Y}$.

S2.2 CopyTree in exponential form

We will now refer to the variables using their biological meaning e.g. denoting by “copy-number profile” the latent states of the Markov chain, but the derivations are context independent and can be applied to any such model. The purpose of this section is to show how the conditional probability in each node of the Bayesian graph Figure S7 can be expressed as a function of the central node v ’s sufficient statistic. These manipulations, besides allowing for Variational Message Passing [33], better explain the variational updates in section S3.

Root node Let's denote with $\mathbf{C}^r$ the copy number sequence of the root node and with $\mathbf{Y}^r$ the observations to it related. Since this does not depend on any other copy number sequence, we say that $(\mathbf{C}^r, \mathbf{Y}^r) \sim \text{HMM}(\phi, \psi)$ with $\phi = \{\pi, \Phi\}$ being the Markov chain parameters (initial state probability vector and transition matrix), and ψ being the set of parameters for the chosen emission distribution, whatever it may be.

Following the derivations from the previous section, the joint density function of $\mathbf{C}^r, \mathbf{Y}^r$ can be written in exponential form as

$$f_{\text{HMM}}(\mathbf{C}^r, \mathbf{Y}^r | \phi, \psi) = \exp(\eta_{\mathbf{C}^r}(\phi, \psi) \cdot \mathbf{t}_{\mathbf{C}^r}(\mathbf{C}^r) - A_{\mathbf{C}^r}(\phi, \psi) + B_{\mathbf{C}^r}(\mathbf{C}^r)). \quad (\text{S10})$$

For a more readable notation we simply specify the node to which the parameter is related, which implicitly sets the right variables upon which a certain function depends. E.g. $\eta(r) := \eta_{\mathbf{C}^r}(\phi, \psi)$. Equation S10 can be therefore rewritten as

$$f_{\text{HMM}}(\mathbf{C}^r, \mathbf{Y}^r | \phi, \psi) = \exp(\eta(r) \cdot \mathbf{t}(r) - A(r) + B(r)). \quad (\text{S11})$$

Both the sufficient statistic and the natural parameter vectors are easily represented as the concatenation of two vectors

$$\mathbf{t}(r) = \begin{bmatrix} t_1(r) \\ t_2(r) \end{bmatrix}, \quad \eta(r) = \begin{bmatrix} \eta_1(r) \\ \eta_2(r) \end{bmatrix}. \quad (\text{S12})$$

In subsection S2.1 we show that the vectors' elements are as follows

$$\begin{aligned} t_1(r)_{\iota(m,j)} &= \mathbb{1}(C_m^r = j) \\ t_2(r)_{\kappa(m,j,j')} &= \mathbb{1}(C_{m-1}^r = j, C_m^r = j') \\ \eta_1(r)_{\iota(1,j)} &= \log \pi_j + \log p_\psi(\mathbf{Y}_1 | C_1^r = j) \\ \eta_1(r)_{\iota(m,j)} &= \log p_\psi(\mathbf{Y}_m | C_m^r = j) \\ \eta_2(r)_{\kappa(m,j,j')} &= \log \Phi_{\kappa(m,j,j')}. \end{aligned} \quad (\text{S13})$$

Tree HMM joint We write the root node r HMM probability as the standard HMM joint since it does not have a parent node.

$$p_{\phi\psi}(\mathbf{C}^r, \mathbf{Y}^r) = p(C_1^r) p(\mathbf{y}_1^r | C_1^r) \prod_{m=2}^M p(C_m^r | C_{m-1}^r) p(\mathbf{y}_m^r | C_m^r), \quad (\text{S14})$$

Then, for any internal node u with parent v we get the probability of the HMM of a child v of u , given the parent as

$$\begin{aligned}
p_{\varepsilon\psi}(\mathbf{C}^v, \mathbf{Y}^v | \mathbf{C}^u) &= \\
&= p(C_1^v | \mathbf{C}^u) p(\mathbf{y}_1^v | C_1^v, \mathbf{C}^u) \prod_{m=2}^M p(C_m^v | C_{m-1}^v, \mathbf{C}^u) p(\mathbf{y}_m^v | C_m^v, \mathbf{C}^u) \\
&= p(C_1^v | C_1^u) p(\mathbf{y}_1^v | C_1^v) \prod_{m=2}^M p(C_m^v | C_{m-1}^v, C_m^u, C_{m-1}^u) p(\mathbf{y}_m^v | C_m^v) \quad (\text{S15})
\end{aligned}$$

where in step (S15) we assume that the transition probability of the child only depends on the two copy numbers of the parent at bin $m - 1, m$.

The complete joint probability $p_{\phi\psi\varepsilon}(\mathbf{C}, \mathbf{Y})$ should be derived from root to leaves given the two quantities above in a chain product.

Internal copy-number node For an internal node of the tree, we introduce the CN coherence function, defined over each single copy number of the sequence. For any node v and its parent u , the CN coherence function h over the copy number values at a specific bin m is defined as in (4).

For the first element of the sequence we simply redefine h to be

$$\begin{aligned}
h(j|i, \varepsilon_0) &= P(C_1^v = j | C_1^u = i) \\
&= \begin{cases} 1 - \varepsilon_0 & \text{if } i = j \\ \varepsilon_0 / (|\Sigma| - 1) & \text{otherwise,} \end{cases} \quad (\text{S16})
\end{aligned}$$

In the context of a variational message passing approach [33], it can be useful to draw a simple graph encoding the dependencies (see Figure S7).

The copy number sequences follow a Markov model. Let u, v the parent-child pair of node, as usual, and let $\mathbf{X}^u, \mathbf{X}^v$ be the copy number sequences of the respective nodes in the one-hot encoding described above.

$$\begin{aligned}
&\log f_{X^u}(\mathbf{X}^v | \mathbf{X}^u, \varepsilon) \\
&= \log P(x_1^v | x_1^u) + \sum_{m=2}^M \log P(x_m^v | x_{m-1}^v, x_m^u, x_{m-1}^u) \\
&= \sum_{i,j} X_{i,1}^u X_{j,1}^v \log h(j|i, \varepsilon) \\
&+ \sum_{m=2}^M \sum_{i,i'} \sum_{j,j'} X_{i,m-1}^u X_{j,m-1}^v X_{i',m}^u X_{j',m}^v \log h(j'|j, i', i, \varepsilon) \quad (\text{S17})
\end{aligned}$$

In subsection S2.2 we have decomposed the sufficient statistic vector for HMMs in

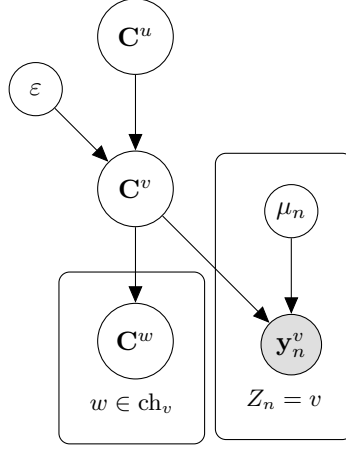


Fig. S7: Simple copy-tree Bayes graph representing the neighborhood of a node v of the tree. For a generic node v , $\mathbf{C}^v$ is the latent copy number sequence, $\mathbf{y}_n^v$ is the sequence of observations for each cell n in node v , ε^v the co-mutation parameter and μ_n is the baseline (per-copy expression).

$$\mathbf{t}_X(\mathbf{X}) = \begin{bmatrix} t_X^1(\mathbf{X}) \\ t_X^2(\mathbf{X}) \end{bmatrix}$$

with t_X^1 and t_X^2 being two column vectors indexed by ι and κ mappings respectively. More specifically,

$$\begin{aligned} t_X^1(\mathbf{X})_{\iota(m,i)} &= X_{i,m} \\ t_X^2(\mathbf{X})_{\kappa(m,i,i')} &= X_{i',m} X_{i,m-1}. \end{aligned}$$

With that in mind, we regroup (S17) to obtain the natural parameter vector $\eta_{X^v}(\mathbf{X}^u) = (\eta_{X^v}^1(\mathbf{X}^u), \eta_{X^v}^2(\mathbf{X}^u))^\top$, which is then equal to

$$\begin{aligned} \eta_{X^v}^1(\mathbf{X}^u, \varepsilon)_{\iota(1,j)} &= \sum_i X_{i,1}^u \log h(j|i, \varepsilon_0) \\ \eta_{X^v}^2(\mathbf{X}^u, \varepsilon)_{\kappa(m,j,j')} &= \sum_{i,i'} X_{i',m}^u X_{i,m-1}^u \log h(j'|j, i', i, \varepsilon) \quad 1 < m \leq M, \end{aligned} \tag{S18}$$

and the rest of the vector elements, namely $\eta_{X^v}^1(\mathbf{X}^u, \varepsilon)_{\iota(m,i)}$ for $1 < m \leq M$ and $\eta_{X^v}^2(\mathbf{X}^u, \varepsilon)_{\kappa(1,i,i')}$, are set to zero.

We can now write the conditional probability in (S17) as

$$\log f_{X^v}(\mathbf{X}^v | \mathbf{X}^u, \varepsilon) = \eta_{X^v}(\mathbf{X}^u, \varepsilon) \cdot \mathbf{t}_{X^v}(\mathbf{X}^v). \quad (\text{S19})$$

It is then useful to rewrite this as a function of the sufficient statistic of the parent node's copy number, so that the rest can be used as a message going from child to parent (see [33]). Precisely, we want to find the vector $\eta_{X^v X^u}$ function of $\mathbf{X}^v$ such that

$$\log f_{X^v}(\mathbf{X}^v | \mathbf{X}^u, \varepsilon) = \eta_{X^v X^u}(\mathbf{X}^v, \varepsilon) \cdot \mathbf{t}_{X^u}(\mathbf{X}^u)$$

By symmetry of (S17), and since $\mathbf{t}_{X^u}$ has the same form of $\mathbf{t}_{X^v}$ we obtain $\eta_{X^v X^u}(\mathbf{X}^v, \varepsilon) = (\eta_{X^v X^u}^1(\mathbf{X}^v, \varepsilon), \eta_{X^v X^u}^2(\mathbf{X}^v, \varepsilon))^\top$ such that

$$\begin{aligned} \eta_{X^v X^u}^1(\mathbf{X}^v, \varepsilon)_{\iota(1,i)} &= \sum_j X_{j,1}^v \log h(j|i, \varepsilon) \\ \eta_{X^v X^u}^2(\mathbf{X}^v, \varepsilon)_{\kappa(m,i,i')} &= \sum_{j,j'} X_{j',m}^v X_{j,m-1}^v \log h(j'|j, i', i, \varepsilon) \quad 1 < m \leq M, \end{aligned} \quad (\text{S20})$$

and the rest of the elements equal to zero. Note that even though the presented CN coherence function is symmetric, (S18) and (S20) might differ from the indices of h when it is not symmetric on the pairs (i, i') and (j, j') , therefore in general we have $\eta_{X^v} \neq \eta_{X^v X^u}$.

Observations Let $\mathbf{Y}^u$ be the set of observation at node v . For each bin m we have a collection of reads, one for each cell assigned to the node and for each gene belonging to the bin. We denote this collection of reads with $\mathbf{Y}_m^v$.

To simplify the model, we assume that each of these reads range from 1 to N , independently of the cell/bin they belong to, and we assume that they are drawn from a Normal distribution, conditioned on the bin's copy number, with mean $\mu_n C_m^v$.

The conditional log-distribution is then

$$\begin{aligned} \log f_{Y^v}(\mathbf{Y}^v | \mathbf{X}^v, \boldsymbol{\mu}, \boldsymbol{\tau}) &= \sum_{m=1}^M \log P(\mathbf{Y}_m^v | x_m^v, \boldsymbol{\mu}, \boldsymbol{\tau}) \\ &= \sum_{m=1}^M \sum_{n=1}^{N_v} \sum_{i \in \Sigma} X_{im} \log f_{\mathcal{N}}(y_{mn}^v; i \cdot \mu_n, \tau_n) \end{aligned} \quad (\text{S21})$$

which can be expressed in terms of $\mathbf{t}_{X^v}(\mathbf{X}^v)$

$$\log f_{Y^v}(\mathbf{Y}^v | \mathbf{X}^v, \boldsymbol{\mu}, \boldsymbol{\tau}) = \eta_{Y^v X^v}(\mathbf{Y}^v, \boldsymbol{\mu}, \boldsymbol{\tau}) \cdot \mathbf{t}_{X^v}(\mathbf{X}^v).$$

where $\mathbf{t}_{X^v}$ is the same as in (S2.2) and

$$\eta_{Y^v X^v}(\mathbf{Y}^v, \boldsymbol{\mu}, \boldsymbol{\tau}) = \begin{bmatrix} \eta_{Y^v X^v}^1 \\ \mathbf{0} \end{bmatrix}$$

with $\mathbf{0}$ null vector of length matching $t_{X^v}^2$ (i.e. $M \times |\Sigma|^2$) and $\eta_{Y^v X^v}^1$ with same length as $t_{X^v}^1$, and elements such that

$$\eta_{Y^v X^v}^1(\mathbf{Y}^v)_{i(m,i)} = \sum_{n=1}^N \log f_{\mathcal{N}}(y_{mn}^v; i \cdot \mu_n, \tau_n)$$

S3 Variational Updates

In this section we go through the derivations of the CAVI update for each variational distribution. All updates make use of the CAVI update equation (3).

S3.1 Copy numbers

Copy numbers pre-requisite We begin by observing that our model for a given T , when described as a directed graphical model on the variables

$$\{C_m^v : 1 \leq m \leq M, v \in V(T)\}$$

has a cyclic underlying undirected graph (see 2a).

However, it can be formulated as a directed graphical model on the variables

$$\{\mathbf{C}^v, v \in V(T)\}$$

where $p(\mathbf{C}^v | \mathbf{C}^{\rho_T(v)})$ is a conditional Markov model such that the underlying directed graphical model is a tree. In order to draw this conclusion we consider the conditional joint probability of observations and copy numbers

$$p(\mathbf{Y}, \mathbf{C} | \mathbf{Z}, T, \boldsymbol{\Psi}) = p(\mathbf{Y} | \mathbf{C}, \mathbf{Z}, \boldsymbol{\Psi}) p(\mathbf{C} | T, \boldsymbol{\Psi})$$

and show how to express the above conditional joint as a product of conditional Markov models:

$$\prod_{uv \in A(T)} M(\mathbf{C}^v | \mathbf{C}^u),$$

where the conditional Markov models depend on $\mathbf{Z}, \boldsymbol{\Psi}, \mathbf{Y}$. Notice that

$$\log p(\mathbf{Y} | \mathbf{C}, \mathbf{Z}, \boldsymbol{\Psi}) = \sum_{v \in V(T)} \sum_n \mathbb{1}(Z_n = v) \sum_m \log p(y_{mn} | C_m^v, \mu_n, \tau_n),$$

and that

$$\log p(\mathbf{C} | T, \boldsymbol{\Psi}) = \sum_{uv \in A(T)} \log p(\mathbf{C}^v | \mathbf{C}^u, \varepsilon_{uv}).$$

This implies

$$\begin{aligned} \log p(\mathbf{Y}, \mathbf{C} | \mathbf{Z}, T, \boldsymbol{\Psi}) &= \sum_n \mathbb{1}(Z_n = r) \sum_m \log p(y_{mn} | C_m^r, \mu_n, \tau_n) \\ &+ \sum_{uv \in A(T)} \left[\log p(\mathbf{C}^v | \mathbf{C}^u, \varepsilon_{uv}) + \sum_n [\mathbb{1}(Z_n = v) \sum_m \log p(y_{mn} | C_m^v, \mu_n, \tau_n)] \right] \end{aligned}$$

Hence, it is sufficient to find a conditional Markov model $M(\mathbf{C}^v | \mathbf{C}^u)$ for which the product of sufficient statistics and natural parameter is the outermost summand of the expression above, that is:

$$\begin{aligned}
 \log M(\mathbf{C}^v | \mathbf{C}^u) &= \\
 &= \log p(\mathbf{C}^v | \mathbf{C}^u, \varepsilon_{uv}) + \sum_n \mathbb{1}(Z_n = v) \sum_m \log p(y_{mn} | C_m^v, \mu_n, \tau_n) \\
 &= \log h_{\varepsilon_0}(C_1^v | C_1^u) + \sum_{m=2}^M \log h_{\varepsilon_{uv}}(C_m^v | C_{m-1}^v, C_m^u, C_{m-1}^u) \\
 &\quad + \sum_n [\mathbb{1}(Z_n = v) \sum_m \log p(y_{mn} | C_m^v, \mu_n, \tau_n)]. \tag{S22}
 \end{aligned}$$

The following shows how to express the Markov model for $M(\mathbf{C}^v | \mathbf{C}^u)$ in its exponential family form, that is, using the natural parameter $\boldsymbol{\eta}_T(v)$ and the sufficient statistics $\mathbf{t}_T(\mathbf{C}^v)$. Keep in mind that the sufficient statistics is a function of the copy number sequence, while the natural parameter depends on other parameters e.g. $\boldsymbol{\Psi}$. Let the parent node of v given the tree T simply be $\rho = \rho_T(v)$, and the quantities $\boldsymbol{\eta}_T(v), \mathbf{t}_T(\mathbf{C}^v)$ be just $\boldsymbol{\eta}(v), \mathbf{t}(\mathbf{C}^v)$, then the decomposition of the latters can be expressed in the following way. Referring to the mappings in (S1), similarly to the notation introduced in section S2, the sufficient statistics vector $\mathbf{t}(\mathbf{C}^v)$ is equal to

$$\begin{aligned}
 \mathbf{t}(\mathbf{C}^v) &= (\mathbf{t}_1(\mathbf{C}^v), \mathbf{t}_2(\mathbf{C}^v))^T \\
 t_1(\mathbf{C}^v)_{\iota(m,i)} &= \mathbb{1}(C_m^v = i) \\
 t_2(\mathbf{C}^v)_{\kappa(m,i,i')} &= \mathbb{1}(C_{m-1}^v = i, C_m^v = i').
 \end{aligned}$$

Natural parameters, $\boldsymbol{\eta}(v)$, and associated parameters, $\boldsymbol{\eta}'(v)$:

$$\begin{aligned}
 \boldsymbol{\eta}(v) &= (\boldsymbol{\eta}_1(v), \boldsymbol{\eta}_2(v)) \\
 \boldsymbol{\eta}_1(v) &= \boldsymbol{\eta}'_1(v) + \boldsymbol{\eta}''_1(v) \\
 \eta'_1(v)_{\iota(1,i)} &= \sum_j \mathbb{1}(C_1^\rho = j) \log h_{\varepsilon_0}(i|j) \\
 \eta'_1(v)_{\iota(m,i)} &= 0, \text{ for all } m \neq 1 \\
 \eta''_1(v)_{\iota(m,i)} &= \sum_n \mathbb{1}(Z_n = v) \log p(y_{mn} | C_m^v = i, \boldsymbol{\Psi}) \\
 \boldsymbol{\eta}'(v) &= (\boldsymbol{\eta}'_1(v), \boldsymbol{\eta}_2(v)) \\
 \eta_2(v)_{\kappa(m,i,i')} &= \sum_{j,j'} \mathbb{1}(C_{m-1}^\rho = j, C_m^\rho = j') \log h_{\varepsilon_{\rho v}}(i'|i, j', j),
 \end{aligned}$$

where both $\boldsymbol{\eta}(v)$ and $\boldsymbol{\eta}'(v)$ are vectors of dimension $|\Sigma| \times M + |\Sigma|^2 \times M$, and match the dimension of $\mathbf{t}(\mathbf{C}^v)$.

Notice that, as desired, the inner product of such natural parameters with sufficient statistic vectors, matches the Markov model in (S22):

$$\begin{aligned}
 \boldsymbol{\eta}(v) \mathbf{t}(\mathbf{C}^v) & \tag{S23} \\
 &= \sum_i \mathbb{1}(C_1^v = i) \sum_j \mathbb{1}(C_1^\rho = j) \log h_{\varepsilon_0}(i|j) \\
 &+ \sum_m \sum_i \mathbb{1}(C_m^v = i) \sum_n \mathbb{1}(Z_n = v) \log p(y_{mn} | C_m^v = i, \Psi) \\
 &+ \sum_m \sum_{i,i'} \mathbb{1}(C_{m-1}^v = i, C_m^v = i') \sum_{j,j'} \mathbb{1}(C_{m-1}^\rho = j, C_m^\rho = j') \log h_{\varepsilon_{\rho v}}(i'|i, j', j).
 \end{aligned}$$

Moreover,

$$\begin{aligned}
 \boldsymbol{\eta}'(v) \mathbf{t}(\mathbf{C}^v) &= \tag{S24} \\
 &= \sum_i \mathbb{1}(C_1^v = i) \sum_j \mathbb{1}(C_1^\rho = j) \log h_{\varepsilon_0}(i|j) \\
 &+ \sum_{m=2}^M \sum_{i,i'} \mathbb{1}(C_{m-1}^v = i, C_m^v = i') \sum_{j,j'} \mathbb{1}(C_{m-1}^\rho = j, C_m^\rho = j') \log h_{\varepsilon_{\rho v}}(i'|i, j', j) \\
 &= \log h_{\varepsilon_{\rho v}}(C_1^v | C_1^\rho) + \sum_{m=2}^M \log h_{\varepsilon_{\rho v}}(C_m^v | C_{m-1}^v, C_m^\rho, C_{m-1}^\rho).
 \end{aligned}$$

This should clarify the purpose of the natural and associated parameters: while the latter embodies the attributes of the Markov model over the copy number sequence, the former also links the sequence to the observations.

Copy numbers update We begin by noticing that $\mathbb{E}_{\mathbf{CZ}_\varepsilon}[\log p(\mathbf{C}^r)]$ is uniquely determined by the prior that has probability 1 for the reference genome which typically has copy number 2 everywhere. For the the remaining $K - 1$ vertices we provide the details for the standard CAVI updates. Let us first consider a fixed tree T and any node $v \neq r$.

$$\begin{aligned}
 \log q^*(\mathbf{C}^v) &= \mathbb{E}_{-C^v} [\log p(\mathbf{Y}, \mathbf{C}, \boldsymbol{\Psi}, \mathbf{Z}|T)] \\
 &\stackrel{\pm}{=} \mathbb{E}_{C^{-v} Z \Psi} [\log p(\mathbf{Y}|\mathbf{C}, \mathbf{Z}, \boldsymbol{\mu}, \boldsymbol{\tau}) p(\mathbf{C}|\boldsymbol{\Psi}, \mathbf{Z}, T)] \\
 &\stackrel{\pm}{=} \mathbb{E}_{C^{-v} Z \Psi} [\log p(\mathbf{Y}|\mathbf{C}, \mathbf{Z}, \boldsymbol{\mu}, \boldsymbol{\tau})] + \mathbb{E}_{C^{-v} \Psi} \left[\log p(\mathbf{C}^v) + \sum_{uv \in A(T)} \log p(\mathbf{C}^v|\mathbf{C}^u) \right] \\
 &\stackrel{\pm}{=} \mathbb{E}_{Z \Psi} \left[\sum_{n: Z_n = v} \log p(\mathbf{y}_n|\mathbf{C}^v, \mu_n, \tau_n) \right] + \sum_{uv \in A(T)} \mathbb{E}_{C^{-v}, Z, \varepsilon} [\log p(\mathbf{C}^v|\mathbf{C}^u)] \\
 &\stackrel{\pm}{=} \mathbb{E}_{Z \Psi} \left[\sum_{n: Z_n = v} \log p(\mathbf{y}_n|\mathbf{C}^v, \mu_n, \tau_n) \right] + \mathbb{E}_{C^{\rho_T(v)}, \varepsilon} [\log p(\mathbf{C}^v|\mathbf{C}^{\rho_T(v)})] \\
 &\quad + \sum_{w \in \text{ch}(v, T)} \mathbb{E}_{C^w \varepsilon} [\log p(\mathbf{C}^w|\mathbf{C}^v)] .
 \end{aligned}$$

Where $\rho_T(v)$ is the parent and $\text{ch}(v, T)$ is the set of children of vertex v in tree T . Moreover, as shown above,

$$\begin{aligned}
 &\mathbb{E}_{Z \Psi} \left[\sum_{n: Z_n = v} \log p(\mathbf{y}_n|\mathbf{C}^v, \mu_n, \tau_n) \right] + \mathbb{E}_{C^{\rho_T(v)}, \varepsilon} [\log p(\mathbf{C}^v|\mathbf{C}^{\rho_T(v)})] = \\
 &= \mathbb{E}_{C^{\rho_T(v)} Z \Psi} [M(\mathbf{C}^v|\mathbf{C}^{\rho_T(v)})] \\
 &= \mathbb{E}_{C^{\rho_T(v)} Z \Psi} [\boldsymbol{\eta}_T(v)] \mathbf{t}(\mathbf{C}^v)
 \end{aligned}$$

and, which requires some manipulations,

$$\begin{aligned}
 &\mathbb{E}_{C^w \varepsilon} [\log p(\mathbf{C}^w|\mathbf{C}^v)] \\
 &= \mathbb{E}_{C^w \varepsilon} [\boldsymbol{\eta}_T(\mathbf{C}^w) \mathbf{t}(\mathbf{C}^w)] \\
 &\stackrel{\pm}{=} \mathbb{E}_{C^w \varepsilon} [\boldsymbol{\eta}'_T(\mathbf{C}^w) \mathbf{t}(\mathbf{C}^w)] \\
 &= \mathbb{E}_{C^w \varepsilon} \left[\sum_i \mathbf{1}(C_1^w = i) \sum_j \mathbf{1}(C_1^v = j) \log h_{\varepsilon_{vw}}(i|j) \right. \\
 &\quad \left. + \sum_{m=2}^M \sum_{i, i'} \mathbf{1}(C_{m-1}^w = i, C_m^w = i') \sum_{j, j'} \mathbf{1}(C_{m-1}^v = j, C_m^v = j') \log h_{\varepsilon_{vw}}(i'|i, j', j) \right] \\
 &= \mathbb{E}_{C^w \varepsilon} [\boldsymbol{\alpha}(w)] \mathbf{t}(\mathbf{C}^v) ,
 \end{aligned} \tag{S25}$$

where

$$\alpha(w)_{\iota(m, i)} = \sum_j \mathbf{1}(C_m^w = j) \log h_{\varepsilon_{vw}}(j|i)$$

and

$$\alpha(w)_{\kappa(m,i,i')} = \sum_{j,j'} \mathbb{1}(C_{m-1}^w = j, C_m^w = j') \log h_{\varepsilon_{vw}}(j'|j, i', i).$$

Note that the above two expressions have j and i inverted in $h(\cdot)$ because we express the indexing with i , but the copy numbers indexed by j (which belong to node w), are named after i in (S25).

Consequently,

$$\begin{aligned} \log q^*(\mathbf{C}^v) &= \left(\mathbb{E}_{C^{-v}Z\Psi} [\boldsymbol{\eta}_T(v)] + \sum_{w \in \gamma_T(v)} \mathbb{E}_{C^w\varepsilon} [\boldsymbol{\alpha}(w)] \right) \mathbf{t}(\mathbf{C}^v) \\ &= \mathbb{E}_{C^{-v}Z\Psi} [\boldsymbol{\beta}_T(v)] \mathbf{t}(\mathbf{C}^v) \end{aligned} \quad (\text{S26})$$

where $\boldsymbol{\beta}_T(v)$ can be expressed as the concatenation of two vectors, i.e. $\boldsymbol{\beta}_T(v) = (\beta_{T,1}, \beta_{T,2})$, where $\beta_{T,1}(\mathbf{C}^v)$ and $\beta_{T,2}(\mathbf{C}^v)$ are defined as follows

$$\begin{aligned} \beta_{T,1_{\iota(m,i)}} &= \mathbb{1}(m=1) \left[\sum_j \mathbb{1}(C_1^p = j) \log h_{\varepsilon_{pv}}(i|j) \right. \\ &\quad \left. + \sum_{w \in \gamma_T(v)} \sum_j \mathbb{1}(C_1^w = j) \log h_{\varepsilon_{vw}}(j|i) \right] \\ &\quad + \sum_n \mathbb{1}(Z_n = v) \log p(y_{mn} | C_m^v = i), \end{aligned}$$

and

$$\begin{aligned} \beta_{T,2_{\kappa(m,i,i')}} &= \sum_{j,j'} \mathbb{1}(C_{m-1}^p = j, C_m^p = j') \log h_{\varepsilon_{pv}}(i'|i, j', j) \\ &\quad + \sum_{w \in \gamma_T(v)} \sum_{j,j'} \mathbb{1}(C_{m-1}^w = j, C_m^w = j') \log h_{\varepsilon_{vw}}(j'|j, i', i). \end{aligned}$$

Finally, when considering the VI-distribution over trees $q(T)$, rather than a fixed tree, we obtain the final update

$$\begin{aligned} \log q^*(\mathbf{C}^v) &= \mathbb{E}_{C^{-v}ZT\Psi} \left[\log p(\mathbf{Y} | \mathbf{C}^v, \mathbf{Z}) + \log p(\mathbf{C}^v | \mathbf{C}^{\rho_T(v)}) + \sum_{w \in \gamma_T(v)} \log p(\mathbf{C}^w | \mathbf{C}^v) \right] \\ &= \mathbb{E}_T \left[\mathbb{E}_{C^{-v}Z\Psi} [\boldsymbol{\eta}_T(v)] + \sum_{w \in \gamma_T(v)} \mathbb{E}_{C^w\varepsilon} [\boldsymbol{\alpha}(w)] \right] \mathbf{t}(\mathbf{C}^v). \end{aligned} \quad (\text{S27})$$

Therefore, we update the natural parameter as

$$\begin{aligned}\boldsymbol{\eta}_T^*(v) &= \mathbb{E}_T \left[\mathbb{E}_{C^\rho Z \Psi} [\boldsymbol{\eta}_T(v)] + \sum_{w \in \gamma_T(v)} \mathbb{E}_{C^w \varepsilon} [\boldsymbol{\alpha}(w)] \right] \\ &= \mathbb{E}_T [\mathbb{E}_{C \Psi Z} [\boldsymbol{\beta}_T(v)]] ,\end{aligned}$$

which can be decomposed, element by element, in

$$\begin{aligned}\mathbb{E}_{C^\rho Z \Psi} [\eta_1(v)_{\iota(1,i)}] &= \sum_j q(C_1^\rho = j) \log h_{\varepsilon_0}(i|j) \\ &\quad + \sum_n q(Z_n = v) \mathbb{E}_{\mu_n \tau_n} [\log p(y_{mn} | C_1^v = i)] \\ \mathbb{E} [\eta_1(v)_{\iota(m,i)}] &= \sum_n q(Z_n = v) \mathbb{E}_{\mu_n \tau_n} [\log p(y_{mn} | C_m^v = i)] \\ \mathbb{E}_{C^\rho \Psi} [\eta_2(v)_{\kappa(m,i,i')}] &= \sum_{j,j'} q(C_{m-1}^\rho = j, C_m^\rho = j') \mathbb{E}_\varepsilon [\log h_{\varepsilon_{\rho v}}(i'|i, j', j)] \quad (\text{S28}) \\ \mathbb{E}_{C^w} [\alpha_1(w)_{\iota(m,i)}] &= \sum_j q(C_m^w = j) \log h_{\varepsilon_0}(j|i) \\ \mathbb{E}_{C^w \varepsilon} [\alpha_2(w)_{\kappa(m,i,i')}] &= \sum_{j,j'} q(C_{m-1}^w = j, C_m^w = j') \mathbb{E}_\varepsilon [\log h_{\varepsilon_{vw}}(j'|j, i', i)] ,\end{aligned}$$

where

$$\begin{aligned}&\mathbb{E}_{\mu_n \tau_n} [\log p(y_{mn} | C_m^v = i)] \\ &= \mathbb{E}_{\mu_n \tau_n} \left[\log \left(\sqrt{\frac{\tau_n}{2\pi}} \exp \left(-\frac{\tau_n}{2} (y_{mn} - i\mu_n)^2 \right) \right) \right] \\ &\stackrel{+}{=} \mathbb{E}_{\mu_n \tau_n} \left[\frac{1}{2} \log \tau_n - \frac{\tau_n}{2} (y_{mn} - i\mu_n)^2 \right] \\ &= \frac{1}{2} \mathbb{E}_{\tau_n} [\log \tau_n] - \frac{y_{mn}^2}{2} \mathbb{E}_{\tau_n} [\tau_n] + i y_{mn} \mathbb{E}_{\mu_n \tau_n} [\mu_n \tau_n] - \frac{i^2}{2} \mathbb{E}_{\mu_n \tau_n} [\mu_n^2 \tau_n] .\end{aligned} \quad (\text{S29})$$

Given that the variational distribution over μ_n, τ_n is Normal-Gamma with parameters

$$\mu_n, \tau_n \stackrel{q}{\sim} \text{NormalGamma}(\tilde{\nu}_n, \tilde{\lambda}_n, \tilde{\alpha}_n, \tilde{\beta}_n) , \quad (\text{S30})$$

the expectations in the previous formula are equal to:

$$\begin{aligned}
 \mathbb{E}_{\tau_n} [\log \tau_n] &= \psi(\tilde{\alpha}_n) - \log \tilde{\beta}_n \\
 \mathbb{E}_{\tau_n} [\tau_n] &= \frac{\tilde{\alpha}_n}{\tilde{\beta}_n} \\
 \mathbb{E}_{\mu_n \tau_n} [\mu_n \tau_n] &= \tilde{\nu}_n \frac{\tilde{\alpha}_n}{\tilde{\beta}_n} \\
 \mathbb{E}_{\mu_n \tau_n} [\mu_n^2 \tau_n] &= \frac{1}{\tilde{\lambda}_n} + \tilde{\nu}_n^2 \frac{\tilde{\alpha}_n}{\tilde{\beta}_n}
 \end{aligned} \tag{S31}$$

S3.2 Tree topology

We begin by using the mean-field VI standard update:

$$\begin{aligned}
 \log q^*(T) &= \mathbb{E}_{CZ\Psi} [\log p(\mathbf{Y}, \mathbf{C}, \Psi, \mathbf{Z}, T)] \\
 &\stackrel{\pm}{=} \mathbb{E}_{CZ\Psi} [\log p(\mathbf{Y}, \mathbf{C} | \Psi, \mathbf{Z}, T)] + \log p(T) \\
 &\stackrel{\pm}{=} \mathbb{E}_{CZ\Psi} [\log p(\mathbf{Y} | \mathbf{C}, \mathbf{Z}) + \log p(\mathbf{C} | \Psi, \mathbf{Z}, T)] + \log p(T) \\
 &\stackrel{\pm}{=} \mathbb{E}_{C\Psi} \left[\log p(\mathbf{Y}^r, \mathbf{C}^r | \mathbf{Z}) + \sum_{uv \in A(T)} \log M(\mathbf{C}^v | \mathbf{C}^u) \right] + \log p(T) \\
 &= \mathbb{E}_{C^r Z \Psi} [\log p(\mathbf{Y}^r, \mathbf{C}^r | \mathbf{Z})] + \sum_{uv \in A(T)} \mathbb{E}_{C^v C^u Z \varepsilon} [\eta_T(v) \mathbf{t}(\mathbf{C}^v)] + \log p(T) \\
 &= \mathbb{E}_{C^r} [\log p(\mathbf{C}^r)] + \sum_{uv \in A(T)} \mathbb{E}_{C^v C^u \varepsilon} [\log p(\mathbf{C}^v | \mathbf{C}^u)] + \log p(T)
 \end{aligned} \tag{S32}$$

The expression in (S32) contains three terms that we further examine below.

First, $\mathbb{E}_{C^r Z} [\log p(\mathbf{C}^r | \mathbf{Z})]$, as stated above, is uniquely determined by the prior that has probability 1 for the reference healthy genome, which typically has copy number 2 everywhere. Furthermore, $\log p(T)$ is constant over the support of T as we assign a uniform prior.

What varies over T is the summand of the remaining term, $\mathbb{E}_{C^u C^v \varepsilon} [\log p(\mathbf{C}^v | \mathbf{C}^u)]$, for which we use the notation $w(uv)$ as it represents the unnormalized weight

of each arc in the variational distribution. It can be further simplified as:

$$w(uv) = \mathbb{E}_{C^u C^v \varepsilon} [\log p(\mathbf{C}^v | \mathbf{C}^u)] \quad (\text{S33})$$

$$= \mathbb{E}_{C^u C^v \varepsilon} \left[\log \prod_m \prod_{i, i', j, j'} h_{\varepsilon_{uv}}(i' | i, j', j) \mathbb{1}(C_{m-1}^v = i, C_m^v = i', C_{m-1}^u = j, C_m^u = j') \right] \quad (\text{S34})$$

$$= \sum_m \sum_{i, i'} \mathbb{E}_{C^v} [\mathbb{1}(C_{m-1}^v = i, C_m^v = i')] \sum_{j, j'} \mathbb{E}_{C^u} [\mathbb{1}(C_{m-1}^u = j, C_m^u = j')] \times \mathbb{E}_{\varepsilon} [\log h_{\varepsilon_{uv}}(i' | i, j', j)] \quad (\text{S35})$$

$$= \sum_m \sum_{i, i'} q(C_{m-1}^v = i, C_m^v = i') \sum_{j, j'} q(C_{m-1}^u = j, C_m^u = j') \mathbb{E}_{\varepsilon} [\log h_{\varepsilon_{uv}}(i' | i, j', j)] , \quad (\text{S36})$$

where $q(C_{m-1}^u, C_m^u)$ can be computed iteratively for all m with a single forward pass, which is all performed in $O(KMA)$.

$$q(C_m^u, C_{m-1}^u) = q(C_m^u | C_{m-1}^u = i) q(C_{m-1}^u = i) \quad (\text{S37})$$

$$q(C_m^u) = \sum_{i \in \Sigma} q(C_{m-1}^u, C_{m-2}^u = i) , \quad (\text{S38})$$

and the expectation of the CN coherence function is piecewise defined as

$$\mathbb{E}_{\varepsilon} [\log h_{\varepsilon_{uv}}(i' | i, j', j)] = \begin{cases} \psi(\tilde{b}_{uv}) - \psi(\tilde{a}_{uv} + \tilde{b}_{uv}) & \text{if } i - i' = j - j' \\ \psi(\tilde{a}_{uv}) - \psi(\tilde{a}_{uv} + \tilde{b}_{uv}) - \log A & \text{otherwise,} \end{cases} \quad (\text{S39})$$

following the expression in (4) and the fact that $q(\varepsilon_{uv}) = \text{Beta}(\tilde{a}_{uv}, \tilde{b}_{uv})$.

Note that the weight $w(uv)$ is the same for all trees that contains the edge uv . We use the weight matrix over the fully connected graph, $W(\mathcal{G})$, containing all $w(uv) : uv \in \mathcal{G}$, for sampling trees using 2 and evaluating our sampled trees on the variational distribution $q(T)$.

S3.3 Arc distances

The update for $q(\varepsilon_{uv})$ is derived as follows:

$$\log q^*(\varepsilon_{uv}) \tag{S40}$$

$$\begin{aligned} &= \mathbb{E}_{-\varepsilon_{uv}} [\log p(\mathbf{Y}, \mathbf{C}, \mathbf{Z}, T, \boldsymbol{\Psi})] \\ &\stackrel{\pm}{=} \mathbb{E}_{CT} [\log p(\mathbf{C}|T, \varepsilon)] + \log p(\varepsilon_{uv}) \\ &\stackrel{\pm}{=} \mathbb{E}_T [\mathbb{E}_{C^u C^v} [\log H(C^v|C^u)]] + \log p(\varepsilon_{uv}) \\ &= \mathbb{E}_T \left[\sum_{m=2}^M \sum_{i, i', j, j'} \bar{C}_{mii'}^v \bar{C}_{mjj'}^u \log h_{\varepsilon_{uv}}(i'|i, j', j) \right] \\ &\quad + (a-1) \log(\varepsilon_{uv}) + (b-1) \log(1 - \varepsilon_{uv}), \end{aligned} \tag{S41}$$

where $\bar{C}_{mii'}^v := q(C_{m-1}^v = i, C_m^v = i')$. Furthermore, the summation over the four copy numbers can be split into two separate cases, i.e. the two cases of the CN coherence function, which we denote by $A := \{i, i', j, j' | i - i' = j - j'\}$ and its complementary set $\neg A$. This being said, we rewrite (S41) as

$$\begin{aligned} &\left(\mathbb{E}_T \left[\sum_{m=2}^M \sum_{ii'jj'}^{\neg A} \bar{C}_{mii'}^v \bar{C}_{mjj'}^u \right] + a - 1 \right) \log(\varepsilon_{uv}) \\ &+ \left(\mathbb{E}_T \left[\sum_{m=2}^M \sum_{ii'jj'}^A \bar{C}_{mii'}^v \bar{C}_{mjj'}^u \right] + b - 1 \right) \log(1 - \varepsilon_{uv}) \end{aligned}$$

which leaves us with the updated parameters of the variational Beta distribution over ε_{uv} :

$$\begin{aligned} \tilde{a}_{uv} &= \mathbb{E}_T \left[\sum_{m=2}^M \sum_{ii'jj'}^{\neg A} \bar{C}_{mii'}^v \bar{C}_{mjj'}^u \right] + a \\ \tilde{b}_{uv} &= \mathbb{E}_T \left[\sum_{m=2}^M \sum_{ii'jj'}^A \bar{C}_{mii'}^v \bar{C}_{mjj'}^u \right] + b \end{aligned} \tag{S42}$$

S3.4 Assignment concentration

Derivation of $q(\pi)$:

$$\begin{aligned}
& \log q^*(\pi) \\
&= \mathbb{E}_{-\pi} [\log p(\mathbf{Y}, \mathbf{C}, \mathbf{Z}, \mathbf{T}, \Psi)] \\
&\stackrel{\pm}{=} \mathbb{E}_{-\pi} [\log p(\mathbf{Z}|\pi)] + \mathbb{E}_{-\pi} [\log p(\pi)] \\
&= \sum_{n=1}^N \sum_{k=1}^K q(Z_n = k) \log \pi_k + \log p(\pi) \\
&= \sum_{k=1}^K \log \pi_k \left[\sum_{n=1}^N q(Z_n = k) \right] + \sum_{k=1}^K \log \pi_k (\delta_k - 1) \\
&= \sum_{k=1}^K \log \pi_k \left[\delta_k + \sum_{n=1}^N q(Z_n = k) - 1 \right].
\end{aligned}$$

Therefore, $q^*(\pi)$ is a Dirichlet distribution with parameters $\tilde{\delta} = (\tilde{\delta}_1, \dots, \tilde{\delta}_K)$, where

$$\tilde{\delta}_k = \delta_k + \sum_{n=1}^N q(Z_n = k). \quad (\text{S43})$$

S3.5 Cell assignments

Derivation of $q(Z_n)$:

$$\begin{aligned}
& \log q^*(Z_n) \\
&= \mathbb{E}_{-Z_n} [\log p(\mathbf{Y}, \mathbf{C}, \mathbf{Z}, T, \Psi)] \\
&\stackrel{\pm}{=} \mathbb{E}_{-Z_n} [\log p(\mathbf{Y}|\mathbf{C}, \mathbf{Z}, T, \mu, \tau)] + \mathbb{E}_{-Z_n} [\log p(\mathbf{Z}|\pi)]
\end{aligned}$$

Note that $\log p(\mathbf{Y}|\mathbf{Z}, \mathbf{C}, \Psi)$ and $\log p(\mathbf{Z}|\pi)$ can be written as the sum of two terms, one that depends on Z_n and one that does not, i.e.,

$$\begin{aligned}
& \log p(\mathbf{Y}|\mathbf{Z}, \mathbf{C}, \Psi) \\
&= \sum_{m=1}^M \sum_{k=1}^K \sum_{j \in \Sigma} \mathbb{1}(Z_n = k) \mathbb{1}(C_m^k = j) D_{nmj} \\
&\quad + \sum_{l \neq n}^M \sum_{m=1}^M \sum_{k=1}^K \sum_{j \in \Sigma} \mathbb{1}(Z_l = k) \mathbb{1}(C_m^k = j) D_{lmj},
\end{aligned}$$

and

$$\log p(\mathbf{Z}|\pi) = \sum_{k=1}^K \mathbb{1}(Z_n = k) \log \pi_k + \sum_{l \neq n}^M \sum_{k=1}^K \mathbb{1}(Z_l = k) \log \pi_k,$$

where

$$\begin{aligned} D_{nmj} &= \log p(y_{mn} | C_m = j) \\ &= \log \left(\sqrt{\frac{\tau_n}{2\pi}} \exp \left(-\frac{\tau_n}{2} (y_{mn} - j\mu_n)^2 \right) \right). \end{aligned} \quad (\text{S44})$$

Consequently,

$$\begin{aligned} &\log q^*(Z_n) \\ &\stackrel{\pm}{=} \sum_{k=1}^K \mathbb{1}(Z_n = k) \left[\mathbb{E}_\pi [\log \pi_k] + \sum_{m=1}^M \sum_{j=1}^J \mathbb{E}_{C^k} [\mathbb{1}(C_m^k = j)] \mathbb{E}_{\mu_n \tau_n} [D_{nmj}] \right]. \end{aligned}$$

We simplify the expression further by:

$$\hat{\gamma}_{nk} = \mathbb{E}_\pi [\log \pi_k] + \sum_{m=1}^M \sum_{j \in \Sigma} q(C_m^k = j) \mathbb{E}_{\mu_n \tau_n} [D_{nmj}], \quad (\text{S45})$$

with $\mathbb{E}_\pi [\log \pi_k] = \psi(\tilde{\delta}_k) - \psi(\sum_r \tilde{\delta}_r)$, $\mathbb{E}_{\mu_n \tau_n} [D_{nmj}]$ computed as in (S29). Finally, we set:

$$\tilde{\gamma}_n = \frac{\exp(\hat{\gamma}_{nk})}{\sum_{l=1}^K \exp(\hat{\gamma}_{nl})} \quad (\text{S46})$$

and conclude that $q^*(Z_n) \sim \text{Categorical}(\tilde{\gamma}_n)$ with parameters $\tilde{\gamma}_n = (\tilde{\gamma}_{n1}, \dots, \tilde{\gamma}_{nK})$

S3.6 Baseline mean and precision

We use the Normal-Gamma joint prior on μ_n and τ_n , that factorizes over n , and introduce a joint variational distribution on the corresponding parameters. The mean-field standard update becomes:

$$\begin{aligned} &\log q^*(\mu_n, \tau_n) \\ &= \mathbb{E}_{-(\mu_n, \tau_n)} [\log p(\mathbf{Y}, \mathbf{C}, \mathbf{Z}, \boldsymbol{\Psi})] \\ &\stackrel{\pm}{=} \mathbb{E}_{-(\mu_n, \tau_n)} [\log p(\mathbf{Y} | \mathbf{C}, \mathbf{Z}, \boldsymbol{\Psi})] + \log p(\mu_n, \tau_n) \end{aligned}$$

Simplifying each term individually yields:

$$\begin{aligned} &\mathbb{E}_{-(\mu_n, \tau_n)} [\log p(\mathbf{Y} | \mathbf{C}, \mathbf{Z}, \boldsymbol{\Psi})] \\ &\stackrel{\pm}{=} \mathbb{E}_{-(\mu_n, \tau_n)} \left[\log \prod_{m,u,c} p(y_{mn} | C_m^u, \mu_n, \tau_n)^{\mathbb{1}(Z_n=u, C_m^u=c)} \right] \\ &\stackrel{\pm}{=} \sum_{m,u,c} q(C_m^u = c) q(Z_n = u) \left(\frac{1}{2} \log \tau_n - \frac{1}{2} \tau_n (y_{mn} - c \cdot \mu_n)^2 \right) \end{aligned} \quad (\text{S47})$$

and

$$\log p(\mu_n, \tau_n) \stackrel{\pm}{=} (\alpha - \frac{1}{2}) \log \tau_n - \beta \tau_n - \frac{1}{2} \tau_n \lambda (\mu_n - \mu)^2.$$

As our model prior is Gamma-Normal distributed, we suspect our variational distribution to be in the same family, and attempt to match the parameters associated with the independent variables of the Gamma-Normal. First we regroup the variables as follows:

$$\begin{aligned} \log q^*(\mu_n, \tau_n) &\stackrel{\pm}{=} -(\frac{\tau_n}{2}) [\mu_n^2 \cdot (\lambda + \sum_{m,u,c} q(C_m^u = c) q(Z_n = u) \cdot c^2) \\ &\quad - \mu_n (\lambda \mu + \sum_{m,u,c} q(C_m^u = c) q(Z_n = u) \cdot c \cdot y_{mn})] \\ &\quad + (\alpha + \frac{M}{2} - \frac{1}{2}) \log \tau_n - \tau_n (\beta + \frac{1}{2} (\mu^2 \lambda + \sum_m y_{mn}^2)). \end{aligned}$$

In order to get the expression on Normal-Gamma form we need to complete the square for the cross term. We do this by adding and subtracting the square of the factor of $\mu_n \tau_n$ and then multiplying it with τ_n ; the negative term is then passed to complete the square and the positive is added to the τ_n term:

$$\begin{aligned} \log q^*(\mu_n, \tau_n) &\stackrel{\pm}{=} -(\frac{\tau_n}{2}) (\lambda + \Sigma_{C,Z}(c^2)) [\mu_n - \frac{(\lambda \mu + \Sigma_{C,Z}(c, y_{mn}))}{\lambda + \Sigma_{C,Z}(c^2)}]^2 \\ &\quad + (\alpha + \frac{M}{2} - \frac{1}{2}) \log \tau_n - \tau_n (\beta + \frac{1}{2} (\mu^2 \lambda + \sum_m y_{mn}^2) - \frac{1}{2} \cdot \frac{(\mu \lambda + \Sigma_{C,Z}(c, y_{n,m}))^2}{\lambda + \Sigma_{C,Z}(c^2)}) \end{aligned}$$

where

$$\begin{aligned} \Sigma_{C,Z}(c^2) &= \sum_{m,u,c} q(Z_n = u) q(C_m^u = c) \cdot c^2 \\ \Sigma_{C,Z}(c, y_{mn}) &= \sum_{m,u,c} q(Z_n = u) q(C_m^u = c) \cdot c \cdot y_{mn}. \end{aligned}$$

By matching the independent variables with the parameters we see that $q^*(\mu_n, \tau_n)$ is indeed Normal-Gamma($\tilde{\nu}_n, \tilde{\lambda}_n, \tilde{\alpha}_n, \tilde{\beta}_n$) distributed with parameters:

$$\begin{aligned}
 \tilde{\nu}_n &= \frac{\mu\lambda + \Sigma_{C,Z}(c, y_{mn})}{\lambda + \Sigma_{C,Z}(c^2)} \\
 \tilde{\lambda}_n &= \lambda + \Sigma_{C,Z}(c^2) \\
 \tilde{\alpha}_n &= \alpha + \frac{M}{2} \\
 \tilde{\beta}_n &= \beta + \frac{1}{2}(\mu^2\lambda + \sum_m y_{mn}^2) - \frac{1}{2}\tilde{\nu}_n^2\tilde{\lambda}_n.
 \end{aligned} \tag{S48}$$

S4 ELBO computation

In this section we derive the entropy and negative cross-entropy terms of 10 in 4.3.

Since the expectation over the tree topology T in 10 requires a sampling technique to be estimated, i.e., it cannot be derived as for the other variables in closed form, we separate the expectation into two main parts:

$$\begin{aligned}
 \mathbb{E}_T[\mathbb{E}_{-T}[\log p(\mathbf{Y}|\mathbf{C}, \mathbf{Z}, \boldsymbol{\Psi}) + \log p(\mathbf{C}|\mathbf{Z}, T, \boldsymbol{\Psi}) + \log p(\mathbf{Z}|\boldsymbol{\Psi}) \\
 + \log p(\boldsymbol{\Psi})] + \log p(T)] + \mathbb{H}_q(\mathbf{C}, \mathbf{Z}, T, \boldsymbol{\Psi}).
 \end{aligned}$$

In the subsections below we present each component of the ELBO, by deriving the negative cross entropy terms of in the inner expectation and the variational distribution entropy, the latter which we write following the notation:

$$\mathbb{H}_q(x) = -\mathbb{E}_{q(x)}[\log q(x)].$$

S4.1 Observations

$$\begin{aligned}
 &\mathbb{E}_{CZ\Psi}[\log p(\mathbf{Y}|\mathbf{C}, \mathbf{Z}, \boldsymbol{\Psi})] \tag{S49} \\
 &= \mathbb{E}_{CZ\Psi} \left[\sum_{v,n,m,j} \mathbf{1}(Z_n = v) \mathbf{1}(C_m^v = j) \log p(y_{mn}|C_m^v = j) \right] \\
 &= \frac{1}{2} \sum_{v,n,m,j} q(Z_n = v) q(C_m^v = j) \mathbb{E}_{\mu_n \tau_n} \left[\log \tau_n - \tau_n (y_{mn} - j\mu_n)^2 - \log 2\pi \right] \\
 &= \frac{1}{2} \sum_{v,n,m,j} q(Z_n = v) q(C_m^v = j) \left[\mathbb{E}_{\tau_n} [\log \tau_n] - y_{mn}^2 \mathbb{E}_{\tau_n} [\tau_n] + 2jy_{mn} \mathbb{E}_{\mu_n \tau_n} [\mu_n \tau_n] \right. \\
 &\quad \left. - j^2 \mathbb{E}_{\mu_n \tau_n} [\mu_n^2 \tau_n] - \log 2\pi \right].
 \end{aligned}$$

Note that the values of the expectations above are standard expectations for Normal and Gamma random variables and can be found in subsection S3.1.

S4.2 Copy numbers

$$\begin{aligned}
 & \mathbb{E}_{CZ\Psi} [\log p(\mathbf{C}|\mathbf{Z}, T, \Psi)] \\
 &= \mathbb{E}_{C^r} [\log p(\mathbf{C}^r)] + \sum_{uv \in A(T)} \mathbb{E}_{C^u C^v \varepsilon} [\log p(\mathbf{C}^v | \mathbf{C}^u, \varepsilon)] \\
 &= \mathbb{E}_{C^r} [\log p(\mathbf{C}^r)] \\
 &+ \sum_{uv \in A(T)} \left(\sum_{i,j} q(C_1^v = j) q(C_1^u = i) h_{\varepsilon_0} \right. \\
 &\quad \left. + \sum_{m=2}^M \sum_{ii'jj'} \bar{C}_{mii'}^u \bar{C}_{mjj'}^v \mathbb{E}_{\varepsilon} [\log h_{\varepsilon}(j'|j, i', i)] \right),
 \end{aligned} \tag{S50}$$

where $\mathbb{E}_{\varepsilon} [\log h_{\varepsilon}(j'|j, i', i)]$ is computed as shown in (S39) and the variational distribution quantities (denoted as in Equation (S41)) for the copy numbers are computed with the forward-algorithm.

Furthermore, we can calculate the entropy by use of the marginals $q(C_m)$ in the following way:

$$\mathbb{H}_q(C) = - \sum_{umi} q(C_m^u = i) \log q(C_m^u = i). \tag{S51}$$

S4.3 Cell assignments

$$\begin{aligned}
 & \mathbb{E}_{Z\Psi} [\log p(\mathbf{Z}|\Psi)] \\
 &= \sum_n \mathbb{E}_{Z_n \pi} [\log p(Z_n | \pi)] \\
 &= \sum_n \sum_k q(Z_n = k) \mathbb{E}_{\pi} [\log \pi_k],
 \end{aligned} \tag{S52}$$

with $\mathbb{E}_{\pi} [\log \pi_k] = \psi(\tilde{\delta}_k) - \psi(\tilde{\delta}_0)$, $\tilde{\delta}_0 = \sum_i \tilde{\delta}_i$ such that $\tilde{\delta}$ is the parameter of the variational Dirichlet distribution over π .

The entropy is calculated as:

$$\mathbb{E}_{Z\Psi} [\log q(\mathbf{Z})] = \sum_n \sum_k q(Z_n = k) \log \tilde{\gamma}_{nk}.$$

S4.4 Concentration parameter

$$\begin{aligned}
 & \mathbb{E}_{\pi} [\log p(\boldsymbol{\pi})] \tag{S53} \\
 &= \int_{\boldsymbol{\pi}} q(\boldsymbol{\pi}') \log p(\boldsymbol{\pi}') d\boldsymbol{\pi}' \\
 &= \int_{\boldsymbol{\pi}} q(\boldsymbol{\pi}') \left(\sum_k (\delta_k - 1) \log \pi'_k - \log B(\boldsymbol{\delta}) \right) d\boldsymbol{\pi}' \\
 &= \sum_k (\delta_k - 1) \int_{\boldsymbol{\pi}} q(\boldsymbol{\pi}') \log \pi'_k d\boldsymbol{\pi}' - \log B(\boldsymbol{\delta}) \int_{\boldsymbol{\pi}} q(\boldsymbol{\pi}') d\boldsymbol{\pi}' \\
 &= \sum_k (\delta_k - 1) \mathbb{E}_{\pi \sim q} [\log \pi_k] - \log B(\boldsymbol{\delta}) \\
 &= \sum_k (\delta_k - 1) \psi(\tilde{\delta}_k) - (\delta_0 - K) \psi(\tilde{\delta}_0) - \log B(\boldsymbol{\delta})
 \end{aligned}$$

where

$$B(\boldsymbol{\delta}) = \frac{\prod_k \Gamma(\delta_k)}{\Gamma(\sum_k \delta_k)}, \quad \tilde{\delta}_0 = \sum_{k=1}^K \tilde{\delta}_k$$

and $\tilde{\boldsymbol{\delta}}$ refers to the variational concentration parameter, while $\boldsymbol{\delta}$ is the generative model parameter. The entropy of $q(\boldsymbol{\pi})$ is found by replacing the model parameter $\boldsymbol{\delta}$ with $\tilde{\boldsymbol{\delta}}$.

S4.5 Gaussian mean and precision

$$\begin{aligned}
 & \mathbb{E}_{\mu\tau} [\log p(\boldsymbol{\mu}, \boldsymbol{\tau})] \tag{S54} \\
 &= \sum_n \int_{\mu_n} \int_{\tau_n} q(\mu'_n \tau'_n) \log p(\mu'_n, \tau'_n) d\tau'_n d\mu'_n \\
 &= \sum_n \int_{\mu_n} \int_{\tau_n} q(\mu'_n | \tau'_n) q(\tau'_n) (\log p(\mu'_n | \tau'_n) + \log p(\tau_n)) d\tau'_n d\mu'_n \\
 &= \sum_n \left[\int_{\tau_n} q(\tau'_n) \left(\int_{\mu_n} q(\mu'_n | \tau'_n) \log p(\mu'_n | \tau'_n) d\mu'_n \right) d\tau'_n \right. \\
 &\quad \left. + \int_{\tau_n} q(\tau'_n) \log p(\tau_n) \left(\int_{\mu_n} q(\mu'_n | \tau'_n) d\mu'_n \right) d\tau'_n \right].
 \end{aligned}$$

The above expression is made of two terms (excluding the outermost summation): the first double integral which is the expectation over $q(\tau_n)$ of the negative cross-entropy $-H_{\mu_n | \tau_n}(q, p)$, and the second integral which is just the negative cross-entropy over τ_n of q and p , i.e. $-H_{\tau_n}(q, p)$.

Therefore we get

$$\begin{aligned} \mathbb{E}_{\mu\tau} [\log p(\boldsymbol{\mu}, \boldsymbol{\tau})] \\ = \sum_n \mathbb{E}_{q(\tau_n)} [\mathbb{E}_{q(\mu_n|\tau_n)} [\log p(\mu_n|\tau_n)]] + \mathbb{E}_{q(\tau_n)} [\log p(\tau_n)] . \end{aligned}$$

We compute the partial terms below (omitting q as all the expectations here are taken with respect to the variational distribution)

$$\begin{aligned} \mathbb{E}_{\mu_n|\tau_n} [\log p(\mu_n|\tau_n)] &= \frac{1}{2} \log \frac{\lambda\tau_n}{2\pi} - \frac{\lambda\tau_n}{2} \mathbb{E}_{\mu_n|\tau_n} [(\mu_n - \mu)^2] \\ &= \frac{1}{2} \log \frac{\lambda\tau_n}{2\pi} - \frac{\lambda\tau_n}{2} \text{Var}_{\mu_n|\tau_n} [\mu_n] \\ &= \frac{1}{2} \left(\log \lambda - \log 2\pi - \frac{\lambda}{\tilde{\lambda}_n} \right) + \frac{1}{2} \log \tau_n \end{aligned} \quad (\text{S55})$$

$$\begin{aligned} \mathbb{E}_{\tau_n} [\mathbb{E}_{\mu_n|\tau_n} [\log p(\mu_n|\tau_n)]] &= \frac{1}{2} \left(\log \lambda - \log 2\pi - \frac{\lambda}{\tilde{\lambda}_n} \right) + \frac{1}{2} \mathbb{E}_{\tau_n} [\log \tau_n] \\ &= \frac{1}{2} \left(\log \lambda - \log 2\pi - \frac{\lambda}{\tilde{\lambda}_n} + \psi(\tilde{\alpha}_n) - \log \tilde{\beta}_n \right) \end{aligned} \quad (\text{S56})$$

$$\begin{aligned} \mathbb{E}_{\tau_n} [\log p(\tau_n)] &= \log \frac{\beta^\alpha}{\Gamma(\alpha)} + (\alpha - 1) \mathbb{E}_{\tau_n} [\log \tau_n] - \beta \mathbb{E}_{\tau_n} [\tau_n] \\ &= \log \frac{\beta^\alpha}{\Gamma(\alpha)} + (\alpha - 1)(\psi(\tilde{\alpha}_n) - \log \tilde{\beta}) - \beta \frac{\tilde{\alpha}_n}{\tilde{\beta}_n} \end{aligned} \quad (\text{S57})$$

and then put altogether:

$$\begin{aligned} \mathbb{E}_{\mu\tau} [\log p(\boldsymbol{\mu}, \boldsymbol{\tau})] \\ = \sum_n \left[\frac{1}{2} \left(\log \lambda - \log 2\pi - \frac{\lambda}{\tilde{\lambda}_n} + \psi(\tilde{\alpha}_n) - \log \tilde{\beta}_n \right) \right. \\ \left. + \log \frac{\beta^\alpha}{\Gamma(\alpha)} + (\alpha - 1)(\psi(\tilde{\alpha}_n) - \log \tilde{\beta}) - \beta \frac{\tilde{\alpha}_n}{\tilde{\beta}_n} \right] \\ = -\frac{N}{2} \log 2\pi + \sum_n \frac{1}{2} \left(\log \lambda - \frac{\lambda}{\tilde{\lambda}_n} \right) + (\alpha - \frac{1}{2})(\psi(\tilde{\alpha}_n) - \log \tilde{\beta}_n) \\ + \alpha \log \beta - \log \Gamma(\alpha) - \beta \frac{\tilde{\alpha}_n}{\tilde{\beta}_n} . \end{aligned} \quad (\text{S58})$$

The variational entropy is computed as follows

$$\begin{aligned}
 \mathbb{H}_q(\boldsymbol{\mu}, \boldsymbol{\tau}) &= \sum_n \mathbb{E}_{\tau_n} [\mathbb{H}_q(\mu_n | \tau_n)] + \mathbb{H}_q(\tau_n) \\
 &= \sum_n \frac{1}{2} \left(\log 2\pi + 1 - \log \tilde{\lambda}_n - \psi(\tilde{\alpha}_n) + \log \tilde{\beta}_n \right) \\
 &\quad + \tilde{\alpha}_n - \log \tilde{\beta}_n + \log \Gamma(\tilde{\alpha}_n) + (1 - \tilde{\alpha}_n) \psi(\tilde{\alpha}_n) \\
 &= \frac{N}{2} \log 2\pi + \sum_n \frac{1}{2} \left(1 - \log \tilde{\beta}_n - \log \tilde{\lambda}_n \right) \\
 &\quad + \left(\frac{1}{2} - \tilde{\alpha}_n \right) \psi(\tilde{\alpha}_n) + \tilde{\alpha}_n + \log \Gamma(\tilde{\alpha}_n).
 \end{aligned} \tag{S59}$$

S4.6 Epsilon

$$\begin{aligned}
 &\mathbb{E}_{\varepsilon, T} [\log p(\boldsymbol{\varepsilon})] \\
 &= \mathbb{E}_{T \sim q(T)} \left[\sum_{uv \in T} \mathbb{E}_{\varepsilon_{uv}} [\log p(\varepsilon_{uv})] \right] \\
 &= \mathbb{E}_{T \sim q(T)} \left[\sum_{uv \in T} \mathbb{E}_{\varepsilon_{uv}} [\log B(a, b) - (a-1) \log \varepsilon_{uv} + (b-1) \log(1 - \varepsilon_{uv})] \right] \\
 &= \mathbb{E}_{T \sim q(T)} \left[\sum_{uv \in T} \left(\log B(a, b) - (a-1) [\psi(\tilde{a}_{uv}) - \psi(\tilde{a}_{uv} + \tilde{b}_{uv})] \right. \right. \\
 &\quad \left. \left. + (b-1) [\psi(\tilde{b}_{uv}) - \psi(\tilde{b}_{uv} + \tilde{a}_{uv})] \right) \right]
 \end{aligned} \tag{S60}$$

$$\begin{aligned}
 &\mathbb{E}_{\varepsilon, T} [\log q(\boldsymbol{\varepsilon})] \\
 &= \mathbb{E}_{T \sim q(T)} \left[\sum_{uv \in T} H(q(\varepsilon_{uv})) \right] \\
 &= \mathbb{E}_{T \sim q(T)} \left[\sum_{uv \in T} \left(\log B(\tilde{a}_{uv}, \tilde{b}_{uv}) - (\tilde{a}_{uv}-1) \psi(\tilde{a}_{uv}) - (\tilde{b}_{uv}-1) \psi(\tilde{b}_{uv}) \right. \right. \\
 &\quad \left. \left. + (\tilde{a}_{uv} + \tilde{b}_{uv} + 2) \psi(\tilde{a}_{uv} + \tilde{b}_{uv}) \right) \right]
 \end{aligned} \tag{S61}$$

S4.7 Tree topology

$$\mathbb{E}_T [\log q(T)] = - \sum_{T \in \mathcal{T}} q(T) \log q(T) \tag{S62}$$

since $q(T)$ is unnormalized, we use importance sampling to estimate the normalizing constant and evaluate on $q(T) = \frac{\sum_{l=1}^L w_l \tilde{q}(T)}{\sum_{l=1}^L w_l}$

$$\begin{aligned} \mathbb{E}_T [\log p(T)] &= \sum_{T \in \mathcal{T}} q(T) [-\log |\mathcal{T}|] \\ &= -\log |\mathcal{T}| \sum_{T \in \mathcal{T}} q(T) \\ &= -\log |\mathcal{T}|, \end{aligned} \tag{S63}$$

where in case of K nodes and considering labeled arborescences, $|\mathcal{T}| = K^{K-2}$ [5].

S5 VICTree

S5.1 VICTree pseudocode

The algorithm complexity is dominated by the update of $q(\mathbf{C})$ parameters, which requires $O(KMNA^2)$, at every iteration and for every unique tree in the sample.

Algorithm 1 VICTree pseudocode

```

1: Input: number of importance samples  $L$ , number of iterations  $I$ , initialized  $q$ .
2: for  $i = 1, \dots, I$  do
3:   update  $W(\mathcal{G})$  using (9)
4:   sample  $T_i = T_1, \dots, T_L$  trees using alg. 2
5:   for  $T_i = T_1, \dots, T_L$  do
6:     Update  $q_i(\mathbf{C}), q(\boldsymbol{\varepsilon})$  using  $T_i$  and (8,7)
7:   end for
8:    $q(\mathbf{C}) \leftarrow \sum_i w_{T_i} \cdot q_i(\mathbf{C})$ 
9:   Update  $q(\mathbf{Z}), q(\boldsymbol{\pi}), q(\boldsymbol{\mu}, \boldsymbol{\tau})$  using parameters in (7)
10: end for
11: return
    
```

S5.2 VICTree Importance sampling details

During inference process, we need to evaluate expectations over $q(T)$ of some function of the tree, but enumerating over all possible arborescences can be costly when considering directed labeled trees with several nodes. More specifically, by Cayley's formula [5] and Prüfer sequences it is proven that the number of labeled trees with n nodes is n^{n-2} . In our case the trees are rooted and directed, thus the topology space is of size n^{n-1} , which quickly becomes intractable.

Assuming we can draw samples from $q(T)$ distribution, we can approximate an expectation as follows.

$$\mathbb{E}_{q(T)} [f(T)] \approx \frac{1}{L} \sum_{l=1}^L f(T_l). \quad (\text{S64})$$

However, sampling from $q(T)$ is not straightforward, therefore we sample from an alternative distribution $g(T)$ and apply importance sampling [14] in order to evaluate the desired expectation. Let $\tilde{q}(T)$ denote the weight of an arborescence T , which is defined to be the product over the edge weights and let $w(T) = \tilde{q}(T)/\tilde{g}(T)$ be the importance weights. Then

$$\mathbb{E}_{q(T)} [f(T)] = \mathbb{E}_{g(T)} \left[\frac{\tilde{q}(T)}{\tilde{g}(T)} f(T) \right] \approx \frac{\sum_{l=1}^L w(T_l) f(T_l)}{\sum_{l=1}^L w(T_l)}. \quad (\text{S65})$$

Self-normalization (S65) is required since both $q(T)$ and $g(T)$ are evaluated up to a multiplication factor.

S6 LARS algorithm

The following algorithm runs with computational time $O(|E|K^3)$ since Edmond's algorithm requires $O(K^2)$. It is then clear that, for bigger trees, the complexity of a tree sample quickly overcomes the complexity of the other costly operations in each VICTree step, especially the copy-number variational distribution update $q(\mathbf{C})$ as mentioned in Section S5.1.

Algorithm 2 Labeled Arborescence Sampler (LARS)

- 1: **Input:** $\mathcal{G}$ fully connected graph, W corresponding adjacency matrix.
 - 2: $E \leftarrow$ ordering of arcs in $\mathcal{G}$, $S \leftarrow \emptyset$, $g \leftarrow 1$
 - 3: **while** $i < K - 1$ **do**
 - 4: **for** $e \in E$ **do**
 - 5: $T \leftarrow \text{Edmonds}(A(S, e))$
 - 6: $T' \leftarrow \text{Edmonds}(A'(S, e))$
 - 7: $\theta \leftarrow \frac{\tilde{q}(T)}{\tilde{q}(T) + \tilde{q}(T')}$
 - 8: $u \sim U(0, 1)$
 - 9: **if** $\theta \geq u$ **then**
 - 10: $S \leftarrow S \cup e$
 - 11: $g \leftarrow g \times \theta$
 - 12: **end if**
 - 13: **end for**
 - 14: **end while**
 - 15: **return** S, g
-

S7 Split-merge

We run our split-merge algorithm every 10th iteration during optimization. First, we merge by checking the L1 distance between Viterbi paths of all clusters CNPs $q(C^i)$, $q(C^j)$. If the distance is smaller than a threshold, we merge j into i by re-initializing the parameters of $q(C^j)$ randomly, setting $q(Z_n = j) = 0$ and doubling $q(Z_n = i)$. Then, if there exists a degenerate cluster, l , i.e. a cluster with total probability $\sum q(Z_n = l)$ smaller than a threshold, we run the split algorithm: For all candidate clusters k with $\sum q(Z_n = k) > 0.01$ we group the cells with $\arg \max_j q(Z_n = j) = k$ into two groups, k_1 and k_2 by KMeans. Then, we construct temporary $q(C^{k_1})$ and $q(C^{k_2})$ by applying the CAVI update using only the cells in k_1 and k_2 respectively, and then evaluate the ELBO using these new $q(C)$ s. We select to split the candidate cluster k that results in the maximum ELBO into l .