

UCSF

UC San Francisco Electronic Theses and Dissertations

Title

Following tumor progression step-by-step with CRISPR/Cas9-based single-cell lineage tracing technologies and improved computational methods

Permalink

<https://escholarship.org/uc/item/71s0r7t8>

Author

Jones, Matthew Gregory

Publication Date

2022

Peer reviewed|Thesis/dissertation

Following tumor progression step-by-step with CRISPR/Cas9-based single-cell lineage tracing technologies and improved computational methods

by
Matthew Jones

DISSERTATION

Submitted in partial satisfaction of the requirements for degree of
DOCTOR OF PHILOSOPHY

in

Biological and Medical Informatics

in the

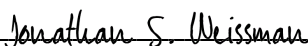
GRADUATE DIVISION

of the

UNIVERSITY OF CALIFORNIA, SAN FRANCISCO

Approved:

DocuSigned by:



Jonathan S. Weissman

E2C110A57BC64CD...

Chair

DocuSigned by:



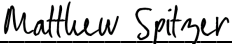
Nir Yosef

DocuSigned by:



Hani Goodarzi

DocuSigned by:



Matthew Spitzer

4793AE6232BE479...

Committee Members

To my wife, Aubri.

To my parents, Greg and Suzanne.

To my brother, Jack.

Acknowledgments

At some point over the past ten years, Berkeley got its claws in me: I arrived to Berkeley in the Fall of 2013 for my undergraduate degree, and despite my best efforts, I haven't left. Undoubtedly, this is because of the outstanding mentors, colleagues, and friends I have made on my journey to where I am now, completing my PhD. I give my deepest thanks to these people.

First and foremost, to my two advisors Nir Yosef and Jonathan Weissman. Through their endless generosity of time and insight, these past five years have been filled with endless excitement and learning. They equipped me with outstanding research environments and provided endless opportunities to pursue the science that I found most exciting. Through them, I have had the opportunity to travel across the world - to visit Nir in Israel in February of 2020 and to visit Jonathan in Cambridge in the final years of my PhD. To say that these experiences were enriching would be an understatement. They have taught me how to pursue science with an open mind and a critical eye, how to be a fair collaborator, how to consider the bigger picture for my work, and so much more. Any success I have in my future, I will be able to tie back to the lessons they have offered me through the PhD.

To my scientific colleagues in the Weissman and Yosef labs. In general, my thanks go to everyone I overlapped with in these labs. Specifically, I am greatly indebted to David DeTomaso who was my first mentor in the Yosef lab and to Jeff Quinn, Dian Yang, and Michelle Chan who formed the nucleus of the lineage tracing team in the Weissman lab when I joined. My special thanks go to Jeff who showed outstanding patience and mentorship both inside and outside the lab; to Dian for his patience bringing me into the field of cancer biology; and to Michelle for sharing with me her

excitement for lineage tracing and her insights at my important career crossroads. Thank you also to Romain Lopez and Adam Gayoso for being great colleagues and friends, and for teaching me a bit about deep generative modeling with patience and zeal. I also give thanks to all those whom I shared office space with in Berkeley: Tal Ashuach, Chenling Xu, Alex Khodaverdian, Sebastian Prillo, Richard Zhang, and Zoë Steier.

To the those I had the privilege of mentoring throughout my PhD: Richard Zhang, Robert Wang, Suhas Rao, Yanay Rosen, Ivan Kristanto, Kevin An, Khalil Ouardini, and Sohil Miglani. I am extremely thankful for the opportunity to mentor these students, publish with them, and watch them grow as scientists and thinkers during their tenure with us. I mean it when I say it: this is the best part of the job.

To my outstanding thesis committee, Matthew Spitzer and Hani Goodarzi. Thank you for your patience, flexibility, dedication, and rich insights throughout my PhD. I am grateful for your mentorship - both formally and informally - that you've given me during my PhD.

To my first research advisor, Russ Corbett-Detig. I am very thankful for his encouragement of my own independent research interests and patience in teaching me the basics of computational biology. I am also thankful that I can still reach out to him with questions and he will take the time to respond thoughtfully to my inquiries.

To my friends who've kept by me and who've I made along the way. They have kept me grounded in real life when science was near to sweeping me up completely. Truly, there are too many to name.

To Christ Church, my spiritual home these past few years. Thank you to the community for providing me a home away from home, lifelong friends, and teachings that I believe have made me

a better scientist and mentor.

To my family: my parents, Gregory and Suzanne Jones, who encouraged me to pursue this wild dream of mine; to my younger brother, Jack, who has always been a role model to me in so many ways. They never asked questions like "So, when are you getting a real job" or looked down on my career path: instead, I could always count on them for support and belief in my work.

And to my wife, Aubri, who has always supported me, has given me nothing but love throughout these years, and has been a constant reminder of what is truly important. I am not sure how I would have persevered through the darkest times without her.

Contributions

This dissertation represents scientific work I contributed to in inception and execution, some of which has been published previously. This section details the reference to each study and the contributions of each author.

In Chapter 2:

Matthew G. Jones*, Alex Khodaverdian*, Jeffrey J. Quinn*, Michelle M. Chan, Jeffrey A. Hussmann, Robert Wang, Chenling Xu, Jonathan S. Weissman†, and Nir Yosef†. Inference of single-cell phylogenies from lineage tracing data using Cassiopeia. *Genome Biol* 21, 92 (2020).

<https://doi.org/10.1186/s13059-020-02000-8>

* These authors contributed equally.

† Corresponding authors

Author contributions: M.G.J., A.K., J.J.Q, N.Y, and J.S.W. contributed to the design of the algorithm, interpretation of benchmarking results, and writing of the manuscript. A.K., C.X., and N.Y. conceived of the multi-state greedy algorithm and Steiner tree adaptation for the phylogeny inference problem. A.K. and M.G.J. implemented the algorithms and all code relevant to the project. M.G.J. and A.K. conducted all stress tests on synthetic datasets. R.W. and A.K. conducted experiments and theoretical work regarding the greedy heuristics robustness in lineage tracing experiments. J.J.Q. generated the in vitro reference dataset. M.G.J., J.J.Q, M.C, and J.A.H. designed the processing

pipeline for empirical lineage tracing data. M.G.J. and J.J.Q processed the reference dataset and M.G.J. reconstructed trees. The authors read and approved the final manuscript.

In Chapter 4:

Jeffrey J. Quinn*, Matthew G. Jones*, Ross A. Okimoto, Shigeki Nanjo, Michelle M. Chan, Nir Yoseft, Trever G. Bivona†, Jonathan S. Weissman†. Single-cell lineages reveal the rates, routes, and drivers of metastasis in cancer xenografts. *Science* 371 (2021).

<https://doi.org/10.1126/science.abc1944>

* These authors contributed equally.

† Corresponding authors

Author contributions: All authors contributed to the design of experiments and analysis. J.J.Q. engineered cell lines, processed tissues, and prepared sequencing libraries. R.A.O. performed mouse surgeries and imaging. M.G.J. and J.J.Q. processed lineage tracing sequencing data. S.N. and J.J.Q. performed invasion assays. M.G.J. performed phylogenetic reconstruction and analyzed the trees and scRNA-seq data. M.G.J. and N.Y. conceived and implemented the FitchCount algorithm. All authors aided in the interpretation of the analyses. J.J.Q., M.G.J., and J.S.W. wrote the manuscript, and all authors read and approved the final manuscript.

In Chapter 5:

Dian Yang*, Matthew G. Jones*, Santiago Naranjo, William M. Rideout III, Kyung Hoi (Joseph) Min, Raymond Ho, Wei Wu, Joseph M. Replogle, Jennifer L. Page, Jeffrey J. Quinn, Felix Horns, Xiaojie Qiu, Michael Z. Chen, William A. Freed-Pastor, Christopher S. McGinnis, David M. Patterson, Zev J. Gartner, Eric D. Chow, Trever G. Bivona, Michelle M. Chan, Nir Yosef†, Tyler Jacks†, Jonathan S. Weissman†. Lineage Tracing Reveals the Phylodynamics, Plasticity and Paths of Tumor Evolution. *Cell* (2022).

<https://doi.org/10.1016/j.cell.2022.04.015>

* These authors contributed equally.

† Corresponding authors

Author contributions: D.Y., M.G.J., T.J. N.Y. and J.S.W. conceived of, designed and led the analysis of the KP-tracer project. D.Y. constructed lineage tracing targeting vectors and engineered the mouse ES cells with the help from J.L.P. and W.F-P.. W.M.R III generated the KP-Tracer chimeric mice and S.N. transduced the mice. D.Y. and S.N. harvested tumors. D.Y. generated the single-cell RNAseq data with help from C.S.M., D.M.P., Z.J.G. and E.D.C.. W.W. and T.G.B analyzed the TCGA data. M.G.J. and N.Y. conceived of computational approaches and M.G.J. implemented these approaches. M.G.J., K.H.M. and D.Y. analyzed the data with help from F.H., X.Q., J.J.Q., R.H., M.Z.C., and M.M.C.. D.Y., M.G.J., N.Y., T.J., J.S.W. interpreted results. D.Y., M.G.J., T.J., N.Y. and J.S.W. wrote the manuscript, with input from all authors. J.S.W., T.J. and N.Y. supervised the project.

Abstract

Following tumor progression step-by-step with CRISPR/Cas9-based single-cell lineage tracing technologies and improved computational methods

Matthew Jones

Cellular lineages underlie several important biological phenomena, from embryogenesis to tumor development. Traditional approaches for studying for these lineages have been limited in their throughput or resolution, and have thus have been largely incapable of profiling lineage dynamics in complex organisms. Recently, advances in microfluidic devices, sequencing technologies, and molecular biology have facilitated a genomics revolution enabling researchers to profile molecular species at single-cell resolution. Simultaneously, progress in precise genome editing with CRISPR/Cas9 technologies have been coupled with the revolution in single-cell genomics to provide single-cell-resolution lineage tracing technologies.

In this thesis, I first describe computational methodology for inferring models of cell lineages, or phylogenies, from the CRISPR/Cas9-based lineage tracing technology. Using both simulated and real data, I demonstrate that our methodology is both scalable and accurate in comparison to other algorithms. I additionally detail the functionality of our end-to-end software suite, Cassiopeia, and speculate on lineage tracing data analysis best practices.

Next, I describe a series of applications of a CRISPR/Cas9-based lineage tracing technology and our computational tools to *in vivo* cancer models. In one application, I describe the first report of using such technologies to investigate the transcriptional drivers of metastatic dynamics in

a xenograft model of non-small-cell lung cancer. Next, I describe work in a genetically engineered mouse model of non-small-cell lung cancer in which we characterize the phylodynamics and evolutionary trajectories that govern a primary tumor as it evolves from a single, transformed cell to a complex, metastatic tumor.

Finally, I conclude by contextualizing how the work presented in this thesis fits into the larger picture of lineage tracing technologies and *in vivo* tumor studies and by speculating on how this informs future work.

Contents

1	Introduction	1
1.1	Measuring individual cells	1
1.2	Lineage tracing	4
1.2.1	Methodology in the genomics era	5
1.2.2	Phylogenetic inference	6
1.2.3	CRISPR/Cas9-based lineage tracing	7
1.3	Applying lineage tracing to tumor evolution	9
1.4	Scope of dissertation	11
1.5	Main contributions	12
1.5.1	Lineage tracing methodology	12
1.5.2	<i>In vivo</i> lineage tracing	13
1.5.3	Software development	14

I	Computational Approaches for Single-Cell Lineage Tracing	16
2	Inference of single-cell phylogenies from lineage tracing data using Cassiopeia	17
2.1	Abstract	18
2.2	Introduction	18
2.3	Results	21
2.3.1	Cassiopeia: A Scalable Framework for Single-Cell Lineage Tracing Phylogeny Inference	21
2.3.2	A Simulation Engine Enables a Comprehensive Benchmark of Lineage Reconstruction Algorithms	25
2.3.3	An <i>In Vitro</i> Reference Experiment Allows Evaluation of Approaches on Empirical Data	30
2.3.4	Generalizing Cassiopeia to Alternative & Future Technologies	36
2.4	Discussion	39
2.5	Methods	42
2.6	Supplementary Figures	89
3	Extending reproducible and accurate lineage analysis with Cassiopeia2.0	115
3.1	Introduction	116
3.2	Results	118
3.2.1	An overview the Cassiopeia2.0 Library	118
3.2.2	A simulation engine to test lineage tracing technology designs	121

3.2.3	Implementing and benchmarking algorithms in a unified framework	124
3.3	Discussion	126
3.4	Methods	128
II	<i>In vivo</i> Lineage Tracing in Cancer Models	137
4	Single-cell lineages reveal the rates, routes, and drivers of metastasis in cancer xenografts	138
4.1	Abstract	139
4.2	Introduction	139
4.3	Results	142
4.3.1	Tracing metastasis in a mouse xenograft model	142
4.3.2	Distinguishing clonal cancer populations	144
4.3.3	Single-cell–resolved cancer phylogenies	146
4.3.4	Inferring and quantifying past metastatic events from phylogenies	148
4.3.5	Transcriptional drivers of differences in metastatic phenotype	152
4.3.6	Heterogeneity and heritability of metastatic behavior in preimplantation cells	156
4.3.7	Evolution of metastatic phenotype	158
4.3.8	Tissue routes and topologies of metastasis	159
4.4	Discussion	162
4.5	Methods	165

4.6	Appendix	194
4.7	Supplementary Figures	210
5	Lineage Tracing Reveals the Phylodynamics, Plasticity and Paths of Tumor Evolution	234
5.1	Abstract	235
5.2	Introduction	235
5.3	Results	239
5.3.1	KP-Tracer mouse enables continuous and high-resolution lineage tracing of tumor initiation and progression	239
5.3.2	Rare subclones expand during tumor progression, marked by increased DNA copy number variation, cell cycle score, and fitness score	242
5.3.3	Integration of phylodynamics and transcriptome uncovers fitness-associated gene programs for KP tumors	245
5.3.4	Intratumoral transcriptional heterogeneity is driven by transient increases in plasticity of cell states	248
5.3.5	Mapping the phylogenetic relationships between cell states reveals common paths of tumor evolution	251
5.3.6	Loss of tumor suppressors alters tumor transcriptome, plasticity and evolutionary trajectory	256
5.3.7	Metastases originate from spatially localized, expanding subclones of primary tumors	258

5.4 Discussion	261
5.5 Methods	265
5.6 Supplementary Figures	302
III Conclusions	309
6 Conclusion	310
References	319
IV Publication Agreement	356

List of Figures

2.1	A generalized approach to lineage tracing & lineage reconstruction.	22
2.2	Cassiopeia algorithms outperform other phylogenetic reconstruction methods on simulated lineages.	26
2.3	An <i>in vitro</i> Reference Experiment.	32
2.4	Cassiopeia can reconstruct high-resolution phylogenetic trees from empirical lineage tracing data.	34
2.5	Cassiopeia builds highly accurate trees from large empirical datasets.	36
2.6	Generalizing Cassiopeia & future design principles of CRISPR-enabled lineage tracers.	38
2.7	Time complexity of lineage reconstruction approaches.	89
2.8	Evaluation of the stability of the maximum neighborhood size parameter.	90
2.9	Observed Frequency of Mutations is Measure of True Mutation Count.	91
2.10	Precision of Cassiopeia-Greedy First Split.	92
2.11	Benchmarking of parallel evolution on the greedy heuristic.	93
2.12	Determination of mutation rates used in simulation.	94

2.13 Triplets Correct Statistic.	95
2.14 Unthresholded Triplets Correct.	96
2.15 Parsimony of reconstructed trees of 400 cell simulated datasets	97
2.16 Benchmarking of lineage tracing algorithms on 1000 cell synthetic datasets.	98
2.17 Benchmarking of greedy and hybrid algorithms on large experiments.	99
2.18 Bootstrapping analysis of Cassiopeia and Neighbor-Joining with the Transfer Bootstrap Expectation statistic.	100
2.19 Reconstruction accuracy under over-dispersed state distributions.	101
2.20 Observed Proportion of Parallel Evolution in Simulations.	102
2.21 Determination of the indel prior transformation function.	103
2.22 Incorporation of priors into Cassiopeia.	104
2.23 Quality control metrics for the target site sequencing library processing pipeline.	105
2.24 Processing Pipeline for the <i>in vitro</i> dataset.	106
2.25 Identification of doublets using intBCs.	107
2.26 Estimation of Prior Probabilities for Tree Reconstruction.	108
2.27 Evaluation of algorithms on <i>in vitro</i> lineage tracing clones, First Split.	109
2.28 Evaluation of algorithms on <i>in vitro</i> lineage tracing clones, Second Split.	110
2.29 Exhaustion of Target Sites across Clones.	111
2.30 Vignette of Inferential Mistakes for Clone 3.	112
2.31 Parsimony scores from reconstructions of the GESTALT datasets.	113
2.32 "Phased Recorder" leverages variability across target sites	114

3.1	An overview of the Cassiopeia V2 Library.	119
3.2	Experimenting with lineage tracing design regimes.	122
3.3	Benchmarking algorithms on synthetic data.	125
4.1	Lineage tracing in a lung cancer xenograft model in mice.	142
4.2	High-resolution phylogenetic trees capture the histories of clonal cancer populations . .	147
4.3	Phylogenetic reconstructions are detailed and accurate.	149
4.4	Quantifying the diverse metastatic phenotypes of clonal populations directly from cell lineages.	150
4.5	Divergent metastatic phenotypes are driven by differences in gene expression.	154
4.6	Metastatic phenotype is predetermined, heritable, and reproducible.	157
4.7	Metastases were seeded via complex tissue routes and multidirectional topologies. . . .	161
4.8	Detailed schematic of lineage tracing methodology.	210
4.9	Cell line engineering strategy and estimation of clonal diversity.	211
4.10	Sequencing library construction and metrics.	212
4.11	Tracing the cell lineages of metastatic progression in three additional mice.	213
4.12	Identifying clonal populations by shared integration barcodes (intBCs).	214
4.13	Characteristics of the lineage tracer and quality-control of clonal populations.	215
4.14	Lineage tracing characteristics of clonal populations in additional mice.	216
4.15	Clonal populations exhibit distinct tissue distributions.	217

4.16	Assessing the accuracy of different measurements of metastatic rate and inference of tissue transitions using simulated lineages.	218
4.17	Relationship between phylogenetic distance and allelic distance for each clonal population.	219
4.18	The TreeMetRate is stable across tree reconstruction algorithms (Cassiopeia versus Neighbor-Joining).	220
4.19	Clonal populations exhibit broad metastatic phenotypes, measured by Tissue Dispersal Score, AlleleMetRate, and TreeMetRate.	221
4.20	The scMetRate measures metastatic potential decoupled from proliferative capacity. . .	222
4.21	The scMetRate measures metastatic potential decoupled from proliferative capacity. . .	223
4.22	Differential expression between non-metastatic and metastatic clonal populations in the primary tissue.	224
4.23	Metastasis-related gene signatures are correlated with metastatic potential.	225
4.24	Many of the same genes are associated with metastatic phenotype across all mice. . . .	226
4.25	Functional validation of five gene candidates in a different cell line (H1299s) and validation of CRISPRi and CRISPRa activity.	227
4.26	Functional validation of five gene candidates in a different cell line (H1299s) and validation of CRISPRi and CRISPRa activity.	228
4.27	Two pairs of clonal populations from mice M10k and M100k are related, enabling an experiment to determine the robustness and reproducibility of metastatic phenotype across independent mouse experiments.	229

4.28	Distinct transcriptional modules underlie distinct clade-specific metastatic behaviors in Clone #7.	230
4.29	Tissue transition probability matrices for each clonal population.	231
4.30	Describing the principal features of metastatic seeding routes.	232
4.31	Seeding topologies observed in each clonal population.	233
5.1	KP-Tracer mouse enables continuous and high-resolution lineage tracing of tumor initiation and progression	240
5.2	Rare subclones expand during tumor progression, marked by increased DNA copy number variation, cell cycle score, and fitness score	244
5.3	Integration of phylodynamics and transcriptome uncovers fitness-associated gene programs for KP tumors	246
5.4	Intratumoral transcriptional heterogeneity is driven by transient increases in plasticity of cell states	249
5.5	Mapping the phylogenetic relationships between cell states reveals common paths of tumor evolution	253
5.6	Loss of tumor suppressors alters tumor transcriptome, plasticity and evolutionary trajectory	255
5.7	Metastases originate from spatially localized, expanding subclones of primary tumors	259
5.8	KP-Tracer mouse genetic components, validation, and quality-control.	302
5.9	Characterization of KP-Tracer tumor subclonal expansions	303
5.10	Characterization of KP-Tracer transcriptomic fitness landscape	304

5.11 Validation of KP-Tracer EffectivePlasticity score and comparison to FitnessSignature . .	305
5.12 Validation of KP-Tracer Evolutionary Coupling and Fate clustering	306
5.13 Genetic perturbations shift the transcriptional fitness and plasticity landscape of KP- Tracer tumors	307
5.14 Lineage tracing illuminates the metastatic routes and origins in KP-Tracer tumors	308

Chapter 1

Introduction

This section will serve as a general introduction to the technologies and analytical methods underpinning my thesis. While in later chapters I will provide more pertinent background into methodology and biological systems, here I provide context necessary for a general audience to understand the later material.

1.1 Measuring individual cells

Each individual human contains approximately 37 trillion cells, but not every cell in a human looks or behaves the same. While each cell in an individual generally carries within it the same set of instructions in the form of a DNA sequence, cells can carry out a wide variety of physiological functions. For example, the muscles of the heart are responsible for pumping blood throughout an individual and the cells of the skin are responsible for maintaining skin-barrier integrity and repelling environmental

contaminants. This remarkable diversity of cellular function is due to the regulation of a cell's DNA sequence. Specifically, the sequence of nucleotides that constitute DNA is broken up into segments of functional units known as genes, each of which carries a specific function within the cell. A gene is used when it is transcribed, or "expressed", into RNA and translated into a sequence of amino acids that are folded into structures called proteins that are responsible for carrying out the function. Thus, by controlling which genes are transcribed and translated, a multiplicity of functions can be encoded by the same DNA sequence.

Given this, disease can often be attributed to the dysregulation of a specific function in a specific cell type. For example, sickle-cell anemia is a deficiency in the gene that encodes hemoglobin - a protein responsible for trafficking oxygen throughout an individual - and impacts red blood cell function. And so, because defects in single cells can lead to systems level effects like disease, it is of central interest to biologists to understand the intricacies of how single cells function.

Towards this goal, the past two decades have witnessed an explosion in technologies that allow one to profile the contents of single cells. First and foremost, technologies have merged to efficiently read DNA sequences, the cost of which has plummeted as these technologies have improved. Second, researchers have improved the efficiency of fundamental molecular reactions and have developed impressive assays that allow these molecular reactions to be performed in oil droplets with microfluidic devices. The union of these two developments - scalable DNA sequencing technologies and precise microfluidic devices - have precipitated the development of single-cell genomic assays. As the name suggests, these assays enable the profiling of genomic material in individual cells and thus allow one to approach a central goal of biomedicine to understand how individual cell

dysregulation leads to system-level disease.

Single-cell assays have proven to be remarkably flexible in terms of what can be measured. While the most pervasive single-cell assay measures the RNA content of single cells (single-cell RNA-sequencing [scRNA-seq] [168, 146, 298, 296]), platforms for measuring several other entities, like the surface-protein repertoire [244] or accessibility of DNA [28], have collectively broadened our window into the cell. Together, these technologies have propelled international, consortium-led efforts to create “atlases” of organisms that offer the promise of understanding how fluctuations in normal cell composition might lead to disease [49, 50].

It is undisputed that these technologies have had an immense impact on our understanding of the diversity of cellular function and composition of tissues. However, there is a complication in the interpretation of this data - namely that biology is dynamic. To appreciate this, consider the example above about the skin defending the individual against foreign pathogens. In this system, which cells are responsible for responding to a given pathogen? How does the system of cells comprising the skin mount an effective response? Naturally, this response requires coordination between different cells, and often requires cells to change their “state”, by modulating which genes are expressed and thus creating new cellular functions for a given threat. In this, do specific cells give rise to these other cells actually fighting the infection? Unfortunately, popular single-cell assays like scRNA-seq do not inform us how the system changes over time. To study such questions, we must leverage lineage tracing technologies which we now turn our attention to.

1.2 Lineage tracing

Cellular lineages - or the collection of cell divisions and physiological states that a cell population goes through - are ubiquitous across biology. Take for example that of embryogenesis: from a single, fertilized zygote, a complex multi-cellular body capable of planned movement and thought emerges over the course of a few months. Through this process, cells constantly change their function based on both intrinsic cues and the signals from their neighbors; remarkably, despite this complex process, development is quite reproducible. From a fundamental level, biologists are interested the ordering of these steps through embryogenesis, how they are coordinated, and how disruption in these steps might lead to disease. And of course, lineages are important on smaller scales, too - such as the hematopoietic lineage that populates your blood cells and immune system - and thus general approaches have been developed to answer such questions across a range of timescales.

Lineage tracing (or, alternatively, fate mapping) is a suite of techniques for systematically profiling such lineages [46, 284]. The simplest of these approaches consist of a researcher watching each individual cell division under a microscope and meticulously taking notes. Yet, as one might imagine, this approach does not necessarily scale. Thus, over the years more sophisticated approaches emerged. For example, some early methods introduce a mark, such as a fluorescent dye, into a single-cell that is passed on a cell divides and propagates from generation to generation. Then, after an experiment, a researcher can study the contributions to a final population of cells by stratifying cells based on their color - for example, all the red cells may have descended from the same progenitor.

1.2.1 Methodology in the genomics era

Akin to the revolution in single-cell genomics, there has been an explosion of new methodology for lineage tracing in complex organisms, enabling researchers to simultaneously map out cellular phenotypes (e.g., which genes are being used) and their lineage relationships at large scales [269]. While one might imagine that the most effective way to track lineages is via direct imaging (and there are sophisticated ways to do this [173]), many systems are not amenable to this either because they are not transparent or there are just too many cells to track. Thus, more commonly, there are two major classes of lineage tracing approaches used in practice: **prospective** and **retrospective**. In prospective approaches, researchers rely on heritable marks introduced into a clonal progenitor to map fates (for example, the dyes described above). In retrospective approaches, researchers rely on natural variability in observed cells - for example, the naturally occurring mutations that accrue as cells divide - to infer relationships.

Traditionally, prospective lineage tracing approaches report on *clonal* dynamics, such as the size and diversity of a particular population that descended from a specific cell. By nature, however, they are unable to report on *subclonal* dynamics. Such subclonal dynamics are critical for understanding how diversity originates within a population. On the other hand, *retrospective* approaches offer practitioners the opportunity to reconstruct "phylogenies" - full tree-like structures that summarize every relationship in the population. A good analogy for a phylogeny is that of family tree: with such a family tree, one can map out the interesting relationships over the years that brought the children living today to where they are. Similarly, in lineage tracing applications, such phylogenetic

approaches report on the entire lineage history of a population and enable one to derive insights into how important subpopulations originated in the population.

1.2.2 Phylogenetic inference

As mentioned above, a central component of retrospective lineage tracing approaches, such as the CRISPR/Cas9-based tracing used in this dissertation (see below and Chapter 2) is the inference of phylogenies. Here, I provide a more formal description of the problem. For the less mathematically-inclined reader, I suggest you skip to the next section.

Often, the phylogenetic inference problem is framed as deducing a tree structure, \mathcal{T} , over a set of vertices \mathcal{V} and edges \mathcal{E} , from a set of observations associated with the leaves of the tree \mathcal{L} . In genomic applications, these observations are often nucleotide sequences that can harbor differences, or mutations, across samples. In our CRISPR/Cas9-based applications discussed below, these observations are sets of indels observed in each cell.

Formally, we refer to the variables that define a sample as “characters”, and the observed values for these variables as “character states”. In the case of a nucleotide sequence, the positions sequenced are characters and the observed nucleotides are character states. In the case of CRISPR/Cas9-based lineage tracers, each editable site is a character and the observed indel in a cell is a character state.

Analytically, the problem of inferring \mathcal{T} is computationally intractable and the task of finding the most parsimonious or likely tree is NP-Hard [42, 76]. Intuitively, this is largely due to the fact

that identical mutations can occur at the same site in different sublineages in a process known as *homoplasy*. As such, algorithms typically must exhaustively search through a large set of possible topologies to find one that optimizes some criteria.

There are several approaches for completing these tasks that can be generally classified into one of two groups: *character-based* and *distance-based* approaches. Character-based approaches define an objective over each character that can be computed with respect to a tree structure and optimized. The most common character-based approaches are Maximum Parsimony and Maximum Likelihood, that seek to minimize the number of mutations or maximize the likelihood of the observed data given a tree structure, respectively. On the other hand, distance-based approaches define a *distance metric*, often denoted δ , which computes a distance between two samples given their character states. These distances can then be used in common algorithms like Neighbor-Joining [216] or UPGMA [234] to infer phylogenies. While these distance-based algorithms are far more efficient than character-based approaches, choosing the correct distance metric is computationally intractable.

1.2.3 CRISPR/Cas9-based lineage tracing

Recent advances in CRISPR/Cas9 genome engineering has enabled the usage of Cas9 to introduce random mutations at defined loci and thus create genomic diversity necessary for retrospective lineage tracing. Several such technologies have emerged to both trace lineages with Cas9-based mutations and read out these mutations with single-cell sequencing assays [178, 206, 4, 236, 37,

134]. These technologies generally rely on the same principles [15]: Cas9 is used to target a specific, often synthetic and expressed, genomic loci ("target site") to introduce a stable insertion or deletion ("indel") that is passed down through generations. At the end of an experiment, single-cell RNA-sequencing is often used to read out the indels that a cell carries. Then, one of several computational approaches can be used to reconstruct a phylogeny connecting each observed cell to its lineage history.

As mentioned above, the problem of tree inference is computationally intensive largely due to the problem of homoplasy. Though CRISPR/Cas9 offers a convenient approach for synthetically introducing heritable marks for lineage tracing, the technology suffers from a few notable challenges. First, these technologies typically suffer from missing data. Specifically, entire target sites can be lost due to Cas9 "resection" (in which Cas9 cuts out a large portion of the DNA sequence) or transcriptional silencing, creating missing data that is heritable and passed on through generations. Alternatively, single-cell assays can often lack sensitivity for rare transcripts and thus target sites can stochastically drop out during sequencing. Second, these lineage tracing experiments consist of far more samples than traditional phylogenetic applications: it is often routine to consider reconstructing lineage histories of clones with thousands of cells. As such, the hardness of the inference problem makes the task at hand more intractable because of the size of the input. Lastly, the states observed at each target site are non-traditional alphabets: instead of nucleotides or amino acids, investigators use indels to describe cellular relationships. This has two ramifications: first, this precludes users from using off-the-shelf software and probabilistic models based on nucleotide or amino acid substitution rates; and second, certain indels are much more common than others from Cas9 cuts, thus

increasing the likelihood of homoplasy in most applications.

Despite these difficulties, there have been extensive efforts to develop new computational approaches [127, 92] that address these challenges. Though there is much to be improved upon, these technologies have been successfully applied in a variety of contexts: for example, to study zebrafish neural development [206] and regeneration [4], mouse embryogenesis [37], and cancer metastasis [203, 228], among others.

1.3 Applying lineage tracing to tumor evolution

With the tools and framework described above, we are equipped with the ability to study dynamic processes. One key process that has previously been studied using phylogenetic tools and is particularly approachable with these tools is that of tumor development. In essence, this is because tumor development is a multistep process governed by evolutionary principles [188]. Through this evolutionary process, a transformed cell becomes a deadly tumor by first uncontrolled proliferation, then the acquisition of additional mutational events, restructuring of the tumor “microenvironment”, recruitment of tumorigenic cell types, and eventually the ability to extravasate from the primary tumor site and disseminate throughout the body in a process called “metastasis” [281]. Current research focuses in on several aspects of this process, such as: how clonal are tumors in origin? How reproducible is tumor progression? And, what are the evolutionary mechanisms underlying drug resistance in advanced tumors? Indeed, mechanistic knowledge of how tumors evolve over time within a patient would be instrumental in their clinical management [6].

Especially since the advent of high-throughput DNA sequencing, phylogenetic methodology has been foundational in understanding tumor evolution. Traditionally, investigators have used multi-region sampling from primary tumors (and occasionally associated metastatic tumors harvested from distal sites) and natural genetic variation in the form of point mutations or copy-number variations (CNVs) to infer the evolutionary history of a tumor [223]. Such approaches have elucidated key processes in tumor developments, such as the importance of subclonal evolution [185] and the routes of metastatic spread [87, 263].

However, as discussed in later chapters, there are a few critical factors limiting the feasibility of relying on natural variation for retrospective lineage tracing in tumors. First, we have very little control over how often mutations are introduced into the system; in the extreme case, we might not observe any natural variability allowing us to discern relationships between samples. Second, natural variability often confers a functional effect on cells. Specifically, sites that carry a large deleterious effect when mutated might be purified out of the population - in theory, this obfuscates inference. Lastly, often it is required that investigators perform whole genome sequencing (WGS) on samples to observe sufficient variability for phylogenetic inference. This is still a relatively expensive procedure that limits the throughput of the experiments. Overall as we show in later chapters, the synthetic lineage tracers discussed above (e.g., the CRISPR/Cas9-based tracers) help address several of these issues by controlling the mutation rates and locations.

1.4 Scope of dissertation

In this dissertation, I will describe my work in two parts: first, developing computational methodology for CRISPR/Cas9-based lineage tracers and, second, our applications of these methodologies to tumor evolution. I have been fortunate to work on several auxiliary projects throughout my PhD, but in the interest in of space I focus on the major thrusts of enabling and applying the technology.

In chapter 2, I present our first report of Cassiopeia - an end-to-end pipeline for single-cell lineage tracing analysis. This chapter was originally printed in *Genome Biology* in 2020 [127] and described our algorithmic methodology as well as a series of benchmarking experiments designed to compare the effectiveness of our algorithms against traditional approaches on the task of recovering phylogenetic relationships from CRISPR/Cas9-based lineage tracing technologies.

Chapter 3 provides an overview of how we have extended Cassiopeia to provide more flexible simulation and algorithmic development. Through the analysis of real-world datasets and the Allen Institute DREAM Challenge for lineage reconstruction [92], our team learned a great deal about how to both reconstruct and interpret lineages. We have incorporated several of these lessons into Cassiopeia and continue to build on this community-led effort to inform on new technologies and algorithms. This work represents a manuscript in preparation and is not yet published.

Chapter 4 presents our first application of the CRISPR/Cas9-based lineage tracer to study tumor evolution. This chapter borrows from our study, published in *Science* in 2021 [203], in which we engineered a metastatic human cancer cell line to carry our CRISPR/Cas9-based lineage tracing system and traced the spread of metastasis in a xenograft mouse model. Among the major findings

were the great variability in metastatic capabilities of each single cell and the identification of genes capable of tuning a cell's ability to metastasize. In this study, we also made methodological advances to infer the migration rates between various tissues with our *FitchCount* algorithm, which is presented in the appendix.

In many ways, Chapter 5 represents the climax of our current technological abilities as applied to tumor evolution. In this report, published in *Cell* in 2022, we describe a genetically engineered mouse model of lung adenocarcinoma that enabled us to study the evolution of a tumor from a single, transformed cell throughout its "life" to a metastatic tumor. We profile the phylodynamic patterns underlying tumor growth, the transcriptional determinants of proliferation, the major cell-state changes governing tumor development, and principles underlying metastatic dissemination. We additionally demonstrate how this model is amenable to further genetic perturbation, allowing investigators to study at high-resolution the entire life of a tumor across several genetic backgrounds in its native environment for the first time.

Lastly, I offer concluding thoughts and my perspective on future work in the conclusion.

1.5 Main contributions

1.5.1 Lineage tracing methodology

Development of an end-to-end pipeline for single-cell lineage tracing analysis. The first major outcome of my PhD work was the development of an end-to-end analysis framework for single-cell

lineage tracing analysis. This tool, called Cassiopeia, is the driving force for all the work presented in this dissertation. Within Cassiopeia, we have implemented several algorithms for the purposes of reconstructing cellular lineages. In the latter half of PhD, Cassiopeia served as the software library in which to extend new computational ideas and make them available to the community.

Learning evolutionary parameters from phylogenetic trees. Through my analysis of real-world datasets, I developed approaches for extracting biological signal from the paired single-cell RNA-seq and lineage tracing data. A notable example is the *FitchCount* algorithm, proposed in [203], which infers the relative transition probabilities of a cell metastasizing from one tissue to another. In the appendix to Chapter 4, we prove its correctness and speculate that it can be applied to a host of other tasks, like inferring cell state transition probabilities.

1.5.2 *In vivo* lineage tracing

A first report of *in vivo* lineage tracing of metastatic dissemination in a mouse model of lung cancer. In our first application of the lineage tracing technology to study tumor evolution, we engineered a metastatic human cell line and traced their metastatic spread throughout a mouse over the course of approximately 2.5 months. Using the analytical tools developed in the Cassiopeia package [127], we derived estimates for a clone's relative propensity for metastasizing between particular tissues, identified genes whose expression modulated a cell's ability to metastasize, as well as learned the stereotypical routes through tissues that clones took as they metastasized.

The tracing of an entire tumor's development with single-cell lineage tracing. We engineered a commonly-used genetically engineered mouse model of lung adenocarcinoma to carry our lineage tracing technology, thus enabling us to simultaneously induce oncogenic transformation and lineage tracing in the mouse lung epithelium. Using this system, we traced several tumors over the course of ~ 5 months and studied the gene expression signatures of subclonal evolution, the cell state transitions governing tumor development, and the requisite steps leading to metastasis.

1.5.3 Software development

An end-to-end software suite for lineage tracing analysis. Our solution to lineage reconstruction of CRISPR/Cas9-based lineage tracing data came with a software suite for end-to-end data analysis called Cassiopeia [127]. Our multi-threaded preprocessing pipeline provides users with tools for working with sequenced amplicon libraries and performs error correction, edit identification, and doublet detection, amongst other procedures. Moreover, our solvers and tools libraries provides users with flexible modules for reconstructing and analyzing lineages from any data modality, not just CRISPR/Cas9-based data. Cassiopeia is publicly available at <https://github.com/YosefLab/Cassiopeia>.

An open-source library for lineage reconstruction algorithms. Within Cassiopeia, we have implemented several previously published algorithms that can be used for solving lineages within

our data framework. Amongst these are the classic algorithms Neighbor-Joining [216] and UPGMA [234], as well as more recent algorithms like Spectral Neighbor Joining [122]. We designed our object oriented programming paradigm to be particularly amenable to extending the library with additional algorithms and hope it will be useful for the rest of the community.

Visualization frameworks for multi-modal lineage data. We built on our previous web-based, interactive data exploration framework, VISION [58], to support the use of phylogenetic trees for exploration. This tool, dubbed *PhyloVision* [128], is an open-source software tool enabling users to explore gene sets associated with evolutionary patterns and perform interactive exploration using the lineage relationships.

Part I

Computational Approaches for Single-Cell Lineage Tracing

Chapter 2

Inference of single-cell phylogenies from lineage tracing data using Cassiopeia

2.1 Abstract

The pairing of CRISPR/Cas9-based gene editing with massively parallel single-cell readouts now enables large-scale lineage tracing. However, the rapid growth in complexity of data from these assays has outpaced our ability to accurately infer phylogenetic relationships. First, we introduce Cassiopeia—a suite of scalable maximum parsimony approaches for tree reconstruction. Second, we provide a simulation framework for evaluating algorithms and exploring lineage tracer design principles. Finally, we generate the most complex experimental lineage tracing dataset to date, 34,557 human cells continuously traced over 15 generations, and use it for benchmarking phylogenetic inference approaches. We show that Cassiopeia outperforms traditional methods by several metrics and under a wide variety of parameter regimes, and provide insight into the principles for the design of improved Cas9-enabled recorders. Together, these should broadly enable large-scale mammalian lineage tracing efforts. Cassiopeia and its benchmarking resources are publicly available at www.github.com/YosefLab/Cassiopeia.

2.2 Introduction

The ability to track fates of individual cells during the course of biological processes such as development is of fundamental biological importance, as exemplified by the ground-breaking work creating cell fate maps in *C. elegans* through meticulous visual observation [248, 55]. More recently, CRISPR/Cas9 genome engineering has been coupled with high-throughput single-cell sequencing to enable lineage tracing technologies that can track the relationships between a large number of

cells over many generations (Figure 2.1a, [177, 140]). Generally, these approaches begin with cells engineered with one or more recording “target sites” where Cas9-induced heritable insertions or deletions (“indels”) accumulate and are subsequently read out by sequencing. A phylogenetic reconstruction algorithm is then used to infer cellular relationships from the pattern of indels. These technologies have enabled the unprecedented exploration of zebrafish [178, 206, 236, 270] and mouse development [134, 37].

However, the scale and complexity of the data produced by these methods are rapidly becoming a bottleneck for the accurate inference of phylogenies. Specifically, traditional algorithms for reconstructing phylogenies (such as Neighbor-Joining [216] or Camin-Sokal [29]) have not been fully assessed with respect to lineage tracing data and may not be well suited for analyzing large-scale lineage tracing experiments for several reasons. First, traditional algorithms were developed for the cases of few samples (in this case cells) and thus scalability is a major limitation (Additional file 1: Fig S1). Second, these algorithms are not well suited to handle the amount of missing data that is typical of lineage tracing experiments, which can be “heritable” (resulting from either large Cas9-induced resections that remove target sites or transcriptional silencing) or “stochastic” (caused by incomplete capture of target sites). Third, these approaches do not explicitly take into consideration the design principles of lineage-tracers, such as the irreversibility of mutations or the unedited state of the founder cell. Together, these reasons necessitate the development of an adaptable approach for reconstructing single-cell phylogenies and an appropriate benchmarking resource that can aid in the development of such algorithms.

Ideally, an algorithm for phylogeny inference from lineage tracing data would be robust to exper-

imental parameters (e.g. rate of mutagenesis, the number of Cas9 target sites), scalable to at least tens of thousands of cells, and resilient to missing data. In this study, we introduce Cassiopeia: a novel suite of three algorithms specifically aimed at reconstructing large phylogenies from lineage tracing experiments with special consideration for the Cas9-mutagenesis process and missing data. Cassiopeia's framework consists of three modules: (1) a greedy algorithm (Cassiopeia-Greedy), which attempts to construct trees efficiently based on mutations that likely occurred earliest in the experiment; (2) a near-optimal algorithm that attempts to find the most parsimonious solution using a Steiner-Tree approach (Cassiopeia-ILP); and (3) a hybrid algorithm (Cassiopeia-Hybrid) that blends the scalability of the greedy algorithm and the exactness of the Steiner-Tree approach to support massive single-cell lineage tracing phylogeny reconstruction. To demonstrate the utility of these algorithms, we compare Cassiopeia to existing methods using two resources: first, we benchmark the algorithms using a custom simulation framework for generating synthetic lineage tracing datasets across varying experimental parameters. Second, enabled by a customizable target-site processing pipeline (**Figure 2.1b**), we assess these algorithms using a new reference *in vitro* lineage tracing dataset consisting of 34,557 cells over 11 clonal populations. Finally, we use Cassiopeia to explore experimental design principles that could improve the next generation of Cas9-enabled lineage tracing systems.

2.3 Results

2.3.1 Cassiopeia: A Scalable Framework for Single-Cell Lineage Tracing

Phylogeny Inference

Typically, phylogenetic trees are constructed by attempting to optimize a predefined objective over characters (i.e. target sites) and their states (i.e. indels) [291]. Distance-based methods (such as Neighbor-Joining [216, 83, 181] or phylogenetic least-squares [33, 73]) aim to infer a weighted tree that best approximates the dissimilarity between nodes (i.e., the number of characters differentiating two cells should be similar to their distance in the tree). Alternatively, character-based methods aim to infer a tree of maximum parsimony [72, 67]. Conventionally, in this approach the returned object is a rooted tree (consisting of observed "leaves" and unobserved "ancestral" internal nodes) in which all nodes are associated with a set of character states such that the overall number of changes in character states (between ancestor and child nodes) is minimized. Finally, a third class of methods closely related to character-based ones takes a probabilistic approach over the characters using maximum likelihood [69, 202] or posterior probability [117] as an objective.

We chose to focus our attention on maximum parsimony-based methods due to the early success of applying these methods to lineage tracing data [206, 178] as well as the wealth of theory and applications of these approaches in domains outside of lineage tracing [148]. Our framework, Cassiopeia, consists of three algorithms for solving phylogenies. In smaller datasets, we propose the use of a Steiner-Tree approach (Cassiopeia-ILP) [300] for finding the maximum parsimony tree

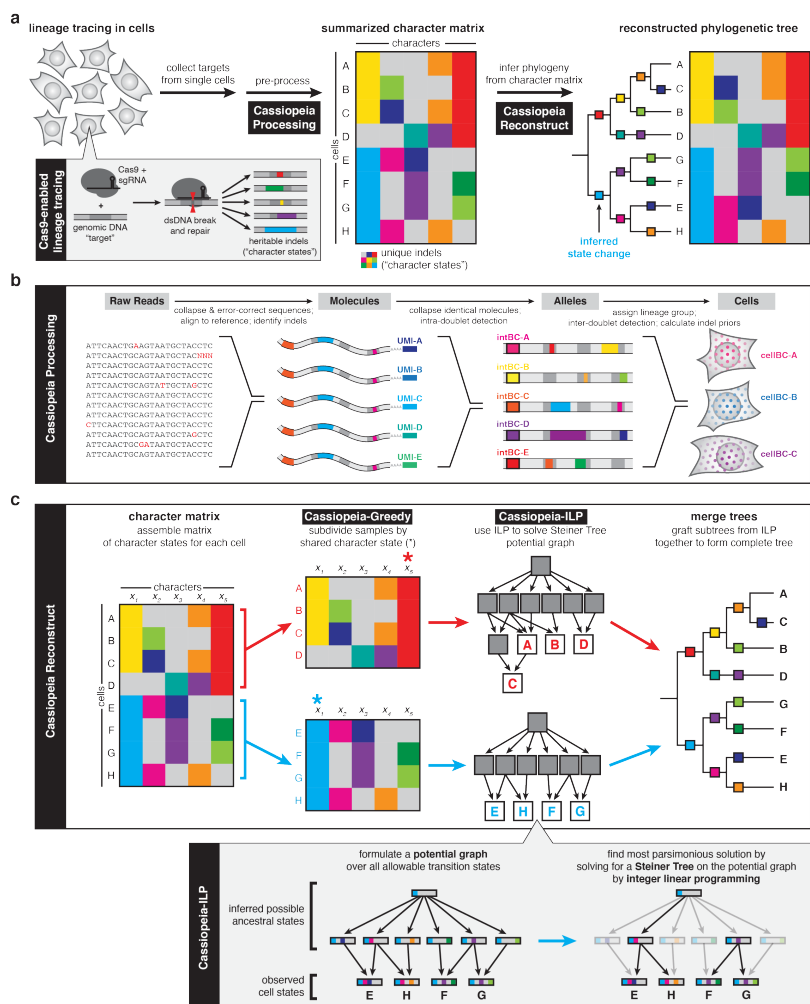


Figure 2.1: A generalized approach to lineage tracing & lineage reconstruction. The workflow of a lineage tracing experiment. First, cells are engineered with lineage tracing machinery, namely Cas9 that cuts a genomic target site; the target site accrues heritable, Cas9-induced indels (“character states”). Next, the indels are read off from single cells (e.g. by scRNA-seq) and summarized in a “character matrix”, where rows represent cells, columns represent individual target sites (or “characters”) and values represent the observed indel (or “character state”). Finally, the character matrix is used to infer phylogenies by one of various methods. (b) The Cassiopeia processing pipeline. The Cassiopeia software includes modules for the processing of target-site sequencing data: first, identical reads are collapsed together and similar reads are error-corrected; second, these reads are locally aligned to a reference sequence and indels are called from this alignment; third, unique molecules are aggregated per cell and intra-doublets are called from this information; finally, the cell population is segmented into clones (or lineage groups) and inter-doublets are called. This clones are then passed to Cassiopeia’s reconstruction module for phylogenetic inference. (c) The Cassiopeia reconstruction framework. Cassiopeia takes as input a “character matrix,” summarizing the mutations seen at heritable target sites across cells. Cassiopeia-Hybrid merges two novel algorithms: the “greedy” (Casiopeia-Greedy) and “Steiner-Tree / Integer Linear Programming” (Casiopeia-ILP) approaches. First, the greedy phase identifies mutations that likely occurred early in the lineage and splits cells recursively into groups based on the presence or absence of these mutations. Next, when these groups reach a predefined threshold, we infer Steiner-Trees, finding the tree of minimum weight connecting all observed cell states across all possible evolutionary histories in a “potential graph”, using Integer Linear Programming (ILP). Finally, these trees (corresponding to the maximum parsimony solutions for each group) are returned and merged into a complete phylogeny.

over observed cells. Steiner Trees have been extensively used as a way of abstracting network connectivity problems in various settings, such as routing in circuit design [95], and have previously been proposed as a general approach for finding maximum parsimony phylogenies [165, 276]. To adapt Steiner-Trees to single-cell lineage tracing, we devised a method for inferring a large underlying "Potential Graph" where vertices represent unique cells (both observed and plausible ancestors) and edges represent possible evolutionary paths between cells. Importantly, we tailor this inference specifically to single-cell lineage tracing assays: we model the irreversibility of Cas9 mutations and impute missing data using an exhaustive approach, considering all possible indels in the respective target sites (see methods). After formulating the Potential Graph, we use Integer Linear Programming (ILP) as a technique for finding near-optimal solutions to the Steiner Tree problem. Because of the NP-Hard complexity of Steiner Trees and the difficult approximation of the Potential Graph (whose effect on solution stability is assessed in **Figure 2.8**), the main limitation of this approach is that it cannot in practice scale to very large numbers of cells.

To enable Cassiopeia to scale to tens of thousands of cells, we apply a heuristic-based greedy algorithm (Cassiopeia-Greedy) to group cells using mutations that likely occurred early in the lineage experiment. Our heuristic is inspired by the idea of "perfect phylogeny" [250, 145] - a phylogenetic regime in which every mutation (here, Cas9-derived indels) are unique and occurred at most once. For the case of binary characters (i.e., mutated yes/ no without accounting for the specific indel), there exists an efficient algorithm [98] for deciding whether a perfect phylogeny exists and if so, to also reconstruct this phylogeny. However, two facets of the lineage tracing problem complicate the deduction of whether or not a perfect phylogeny exists: first, the "multi-state" nature of characters (i.e.

each character is not binary, but rather can take on several different states; which makes the problem NP-Hard) [23, 240]; and second, the existence of missing data [99]. To address these issues, we first take a theoretical approach and prove that since the founder cell (root of the phylogeny) is unedited (i.e. includes only uncut target sites) and that the mutational process is irreversible (i.e. edited sites cannot be recut by Cas9), we are able to reduce the multi-state instance to a binary one so that it can be resolved using a perfect phylogeny-based greedy algorithm. Though Cassiopeia-Greedy does not require a perfect phylogeny, we also prove that if one does exist in the dataset, our proposed algorithm is guaranteed to find it (Theorem 1; see Methods). Secondly, Cassiopeia-Greedy takes a data-driven approach to handle cells with missing data (see Methods). Unlike Cassiopeia-ILP, Cassiopeia-Greedy is not by design robust to parallel evolution (i.e. “homoplasy”, where a given state independently arises more than once in a phylogeny in different parts of the tree). However, we demonstrate theoretically that in expectation, mutations observed in more cells are more likely to have occurred fewer times in the experiment for sufficiently small, but realistic, ranges of mutation rates (see Methods; **Figure 2.9**), thus supporting the heuristic. Moreover, using simulations, we quantify the precision of this greedy heuristic for varying numbers of states and mutation rates, finding in general these splits are precise (especially in these regimes of realistic parameterizations; see Methods and **Figure 2.10**). Below, we further discuss simulation-based analyses that illustrate Cassiopeia-Greedy’s effectiveness with varying amounts of parallel evolution (**Figure 2.11**).

While Cassiopeia-ILP and Cassiopeia-Greedy are suitable strategies depending on the dataset, we can combine these two methods into a hybrid approach (Cassiopeia-Hybrid) that covers a far broader scale of dataset sizes (**Figure 2.1**). In this use case, Cassiopeia-Hybrid balances the

simplicity and scalability of the multi-state greedy algorithm with the exactness and generality of the Steiner-Tree approach. The method begins by splitting the cells into several major clades using Cassiopeia-Greedy and then separately reconstructing phylogenies for each clade with Cassiopeia-ILP. This parallel approach on reasonably sized sub-problems (~ 300 cells in each clade) ensures practical run-times on large numbers of cells (**Figure 2.7**). After solving all sub-problems with the Steiner Tree approach, we merge all clades together to form a complete phylogeny (**Figure 2.1c**).

2.3.2 A Simulation Engine Enables a Comprehensive Benchmark of Lineage Reconstruction Algorithms

To provide a comprehensive benchmark for phylogeny reconstruction, we developed a framework for simulating lineage tracing experiments across a range of experimental parameters. In particular, the simulated lineages can vary in the number of characters (e.g. Cas9 target sites), the number of states (e.g. possible Cas9-induced indels), the probability distribution over these states, the mutation rate per character, the number of cell generations, and the amount of missing data. We started by estimating plausible “default” values for each simulation parameter using experimental data (discussed below and indicated in **Figure 2.2**). In each simulation run, we varied one of the parameters while keeping the rest fixed to their default value. The probability of mutating to each state was found by interpolating the empirical distribution of indel outcomes (**Figure 2.12**, see Methods). Each parameter combination was tested using a maximum of 50 replicates or until convergence, each time sampling a set of 400 cells from the total 2^D cells (where D is the depth of the simulated tree).

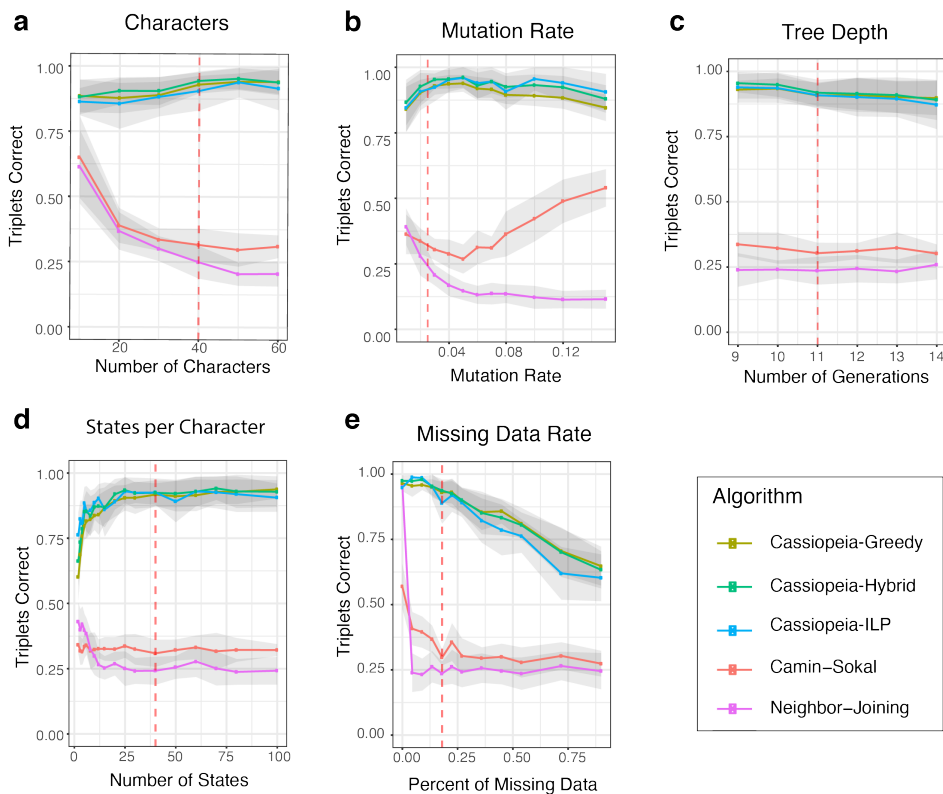


Figure 2.2: **Cassiopeia algorithms outperform other phylogenetic reconstruction methods on simulated lineages.** Accuracy is compared between five algorithms (Cassiopeia-Greedy, -ILP, and -Hybrid algorithms as well as Neighbor-Joining and Camin-Sokal) on 400 cells. Phylogeny reconstruction accuracy is assessed with the Triplets correct statistic across several experimental regimes: (a) the number of characters; (b) mutation rate (i.e. Cas9 cutting rate); (c) depth of the tree (or length of the experiment); (d), the number of states per character (i.e. number of possible indel outcomes); and (e) the dropout rate. Dashed lines represent the default value for each stress test. Between 10 and 50 replicate trees were reconstructed, depending on the stability of triplets correct statistic and overall runtime. Standard error over replicates is represented by shaded area.

We compare the performance of our Cassiopeia algorithms (Cassiopeia-ILP, -Greedy, and -Hybrid) as well as an alternative maximum-parsimony algorithm, Camin-Sokal (previously used in lineage tracing applications [178, 206]), and the distance-based algorithm Neighbor-Joining. We assess performance using a combinatoric metric, “Triplets Correct” (Figure 2.13, see Methods), which compares the proportion of cell triplets that are ordered correctly in the tree. Importantly, this statistic is a weighted-average of the triplets, stratified by the depth of the triplet (measured by the

distance from the root to the Latest Common Ancestor (LCA); see Methods). As opposed to other tree comparison metrics, such as Robinson-Foulds [211], we reason that combinatoric metrics [52] more explicitly address the needs of fundamental downstream analyses, namely determining evolutionary relationships between cells (though the triplets correct statistic largely agrees with distance-based metrics; **Figure 2.13b**).

Overall, our simulations demonstrate the strong performance and efficiency of Cassiopeia. Specifically, we see that the Cassiopeia suite of algorithms consistently finds more accurate trees as compared to both Camin-Sokal and Neighbor-Joining (**Figure 2.2a-e**, **Figure 2.14a-e**). Furthermore, not only are trees produced with Cassiopeia more accurate than existing methods, but also more parsimonious across all parameter ranges - serving as an indication that the trees reach a more optimal objective solution (**Figure 2.15**). Importantly, we observe that Cassiopeia-Hybrid and -Greedy are more effective than Neighbor-Joining in moderately large sample regimes (**Figure 2.16**). Notably, Cassiopeia-Greedy and -Hybrid both scale to especially large regimes (of up to 50,000 cells, a scale that includes the approximate upper limit of most current single-cell sequencing experiments) without substantial compromise in accuracy (**Figure 2.17**). In contrast, Camin-Sokal and Cassiopeia-ILP could not scale to such input sizes (**Figure 2.7**). Finally, we observe that under a bootstrapping analysis, Cassiopeia's modules are robust to lineage-tracing data (**Figure 2.18a,b**) as compared to Neighbor-Joining for reference (**Figure 2.18c**, though Neighbor-Joining's stability may be improved with more sophisticated distance functions and feature selection).

These simulations additionally grant insight into critical design parameters for lineage recording technology. Firstly, we observe that the "information capacity" (i.e. number of characters and

possible indels, or states) of a recorder confers an increase in accuracy for Cassiopeia's modules but not necessarily Camin-Sokal and Neighbor-Joining (though they do perform moderately well in low information capacity simulations; **Figures 2.2a,d**). This is likely because the greater size of the search space negatively affects the performance of these two algorithms (in other contexts referred to as the "curse of dimensionality" [266]). In addition to the information capacity, we find that indel distributions that tend towards a uniform distribution (and thus higher entropy) allow for more accurate reconstructions especially when the number of states is small or the number of samples is large (**Figure 2.19**). Unsurprisingly, the proportion of missing data causes a precipitous decrease in performance (**Figure 2.2e**). Furthermore, in longer experiments where the observed cell population is sampled from a larger pool of cells, we find that the problem tends to become more difficult (**Figure 2.2c**).

Furthermore, these results grant further insight into how Cassiopeia-Greedy is affected in regimes where parallel evolution is likely: such as in low information capacity regimes (e.g. where the number of possible indels is less than 10, **Figure 2.2d**), or with high mutation rates (**Figure 2.2b**). In both of these regimes, the proportion of parallel evolution mutations of all mutations increases (**Figure Eq. 1**). While Cassiopeia-ILP outperforms Cassiopeia-Greedy in these simulations, highlighting its utility to solve small, yet complex, datasets, we further explored Cassiopeia-Greedy's effectiveness in these regimes. To strengthen our previous theoretical results suggesting that indels observed in more cells are more likely to occur fewer times and earlier in the phylogeny (**Figure 2.9**), we explored how parallel evolution affects Cassiopeia-Greedy empirically with simulation. Specifically, we simulated trees with varying numbers of parallel evolution events at various depths and find overall

that while performance decreases with the number of these events, the closer these events occur to the leaves, the smaller the effect (**Figure 2.11**). Furthermore, we find that under the “default” simulation parameters (as determined by the experimental data; **Figure 2.12** and **2.3**), Cassiopeia-Greedy consistently makes accurate choices of the first indel event by which cells are divided into clades (**Figure 2.10b**). Of course in regimes where possible, Cassiopeia-ILP outperforms Cassiopeia-Greedy when there are few states (i.e. fewer than 10; **Figure 2.2d**) or high mutation rates (i.e. greater than 10%; **Figure 2.2b**).

Practically, the issue of parallel evolution can be addressed to some extent by incorporating state priors (i.e. probabilities of Cas9-induced indel formation). Ideally, Cassiopeia-Greedy would use these priors to select mutations that are low-probability, but observed at high frequency. Theoretically, this would be advantageous as low-probability indels are expected to occur fewer times in the tree (Eq. 1); thus if they appear at high frequency at the leaves, it is especially likely that these occurred earlier in the phylogeny. Furthermore, our precision-analysis indicates that Cassiopeia-Greedy’s decisions are especially precise if it chooses an indel with a low prior (**Figure 2.10**). To incorporate these priors in practice, we selected a link function (i.e. one translating observed frequency and prior probability to priority) that maximized performance for Cassiopeia-Greedy (**Figure 2.21**; see Methods). After finding an effective approach for integrating prior probabilities, we performed the same benchmarks, and found that in cases of likely parallel evolution the priors confer an increase in accuracy (e.g. with high mutation rates; **Figure 2.22**), especially in larger regimes (**Figure 2.17**).

Here, we have introduced a flexible simulator that is capable of fitting real data, and thus can be used for future benchmarking of algorithms. Using this simulator and a wide range of parameters, we

have demonstrated that Cassiopeia performs substantially better than traditional methods. Furthermore, these simulations grant insight into how Cassiopeia's performance is modulated by various experimental parameters, suggesting design principles that can be optimized to bolster reconstruction accuracy. Specifically, these simulations suggest that these technologies would benefit most from increases in information capacity, via more target sites or more diverse indel outcomes, and mutation rates tuned appropriately as to ensure low rates of parallel evolution. We anticipate that this resource will continue to be of use in exploring design principles of recorders and the effectiveness of novel algorithms.

2.3.3 An *In Vitro* Reference Experiment Allows Evaluation of Approaches on Empirical Data

Existing experimental lineage tracing datasets lack a defined ground truth to test against, thus making it difficult to assess phylogenetic accuracy in practice. To address this, we performed an *in vitro* experiment tracking the clonal expansion of human cells (A549 lung adenocarcinoma cell line) engineered with a previously described lineage tracing technology [37]. Here, we tracked the growth of 11 clones (each with non-overlapping target site sets for deconvolving clonal populations) over the course of 21 days (approx. 15 generations on average), randomly splitting the pool of cells into two plates every 7 days (**Figure 2.3a**; see Methods). At the end of the experiment, we sampled approximately 10,000 cells from each of the four final plates. This randomized plate splitting strategy establishes a course-grained ground truth of how cells are related to each other. Here, cells within

the same plate can be arbitrarily distant in their lineage, however there is only a lower bound on lineage dissimilarity between cells in different plates (since they are by definition at least separated by the number of mutations that have occurred since the last split). Thus, overall, on average we expect cells within the same plate to be closer to each other in the phylogeny than cells from different plates. However, due to the considerations discussed above, we also expect to see some cells more closely related across plates than within (**Figure 2.3a**, right), and indels relating these cells across plates are likely to have occurred before the split.

Our lineage recorder is based on a constitutively expressed target sequence consisting of three evenly spaced cut sites (each cut site corresponding to a character) and a unique integration barcode ("intBC") which we use to distinguish between target sites and thus more accurately relate character states across cells (**Figure 2.1b**). The target sites are randomly integrated into the genomes of founder cells at high copy number (on average 10 targets per cell or a total of 30 independently evolving characters; **Figure 2.3b**, **Figure 2.24c**). We built upon the processing pipeline in our previous work [37] to obtain confident indel information from scRNA-seq reads (**Figure 2.1b**, **Figure 2.24**, & **Figure 2.23**, see Methods for pre-processing procedures and guidelines, especially section "Guidelines for Final Quality Control"). In addition, we have added modules for the detection of cell doublets using the sets of intBCs in each clone and the indels detected within cells, and have determined an effective detection strategy using simulations (see Methods, **Figure 2.25**). Importantly, though not directly applicable here, this doublet detection can be supplemented by other approaches when transcriptional data [175, 283] or multiplexing barcodes [243] are available. Additionally, we rely on a data-driven approach for estimating the likelihoods of each indel (see Methods; **Figure 2.26**)

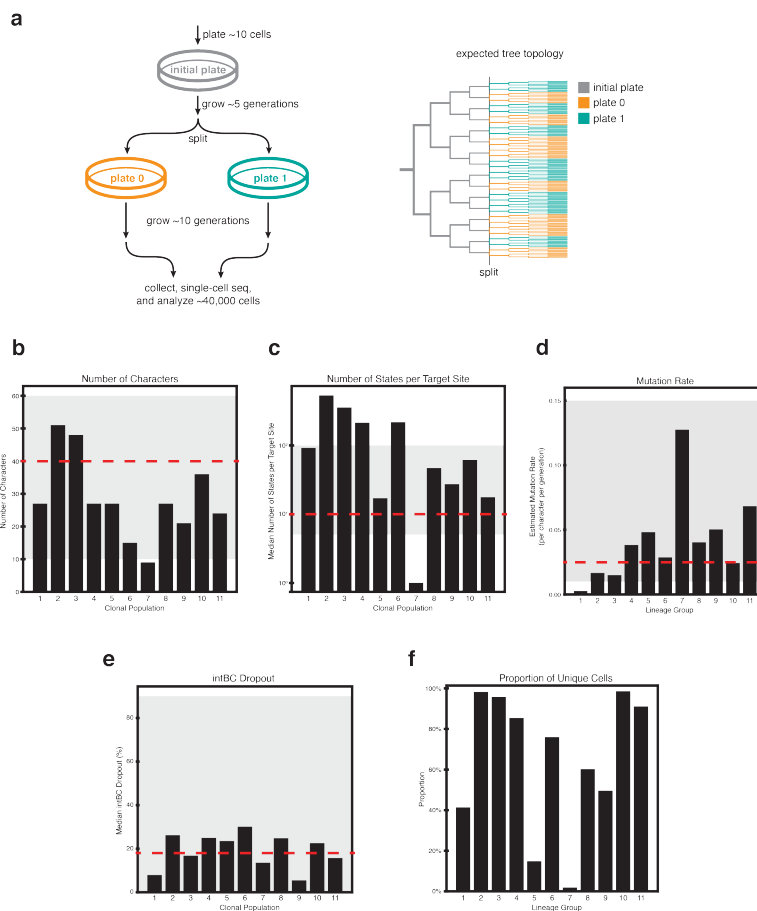


Figure 2.3: **An *in vitro* Reference Experiment.** (a) A reference lineage tracing dataset was generated using the technology proposed in Chan et al. [37] to human cells cultured *in vitro* for ~ 15 generations. A total of 34, 557 cells were analyzed after filtering and error correction. Only the initial split (into two plates) is shown. Analysis of the subsequent split (into four plates) is provided in Figure 2.28. (b-f) Summary of relevant lineage tracing parameters for each clonal population in the experiment: (b) the number of characters per clone; (c) number of states per target site; (d) the estimated mutation rate per target site; (e) median dropout per target site; and (f) the proportion of uniquely marked cells. Gray shading denotes parameter regimes tested in simulations and red-dashed lines denote the default values for each synthetic benchmarks.

because other approaches for indel-likelihood prediction [142, 40, 5] may be biased by cell-type or cell-state.

After quality control, error-correction, and filtering we proceeded with analyzing a total of 34, 557 cells across 11 clones. This diverse set of clonal populations represent various levels of indel diversity (i.e. number of possible states, Figure 2.3c), size of intBC sets (i.e. number of characters,

Figure 2.3b and **Figure 2.24c**), character mutation rates (**Figure 2.3d**, see Methods), and proportion of missing data (**Figure 2.3e**, see Methods). Most importantly, this dataset represents a significant improvement in lineage tracing experiments: it is the longest and most complex dataset to date in which the large majority of cells, over the entire cell population, have unique mutation states (71% after all quality-control and filtering; percentages of unique cells per clone is presented in **Figure 2.3f**), indicating a rich character state complexity for tree building.

We next reconstructed trees for each clone (excluding two which were removed through quality-control filters; see Methods) with our suite of algorithms, as well as Neighbor-Joining and Camin-Sokal (when computationally feasible). For both Cassiopeia-Greedy and Cassiopeia-Hybrid methods, we also compared tree reconstruction accuracy with or without prior probabilities. The tree for Clone 3, consisting of 7,289 cells, along with its character matrix and first split annotations (i.e. whether cells were initially split into plate 0 or plate 1, denoted as the plate ID), is presented in **Figure 2.4**. Interestingly, we find that certain indels indeed span the different plates, thus suggesting that Cassiopeia-Greedy chooses as early splits indels which likely occurred prior to the first separation of plates (though this could also be due to parallel events that occurred independently at each plate). Moreover, the character matrix and the nested dissection of the tree illustrate the abundant lineage information encoded in this clone (96% of the 7,289 observed cells have unique mutation states) which allows Cassiopeia to infer a relatively deep tree (**Figure 2.4d**). Despite this complexity, Cassiopeia infers a tree that largely agrees with the observed mutations: cells close to one another in the tree tend to have similar mutations (**Figure 2.4e**). By keeping track of which plate each cell came from we are able to evaluate how well the distances in a computationally-reconstructed tree

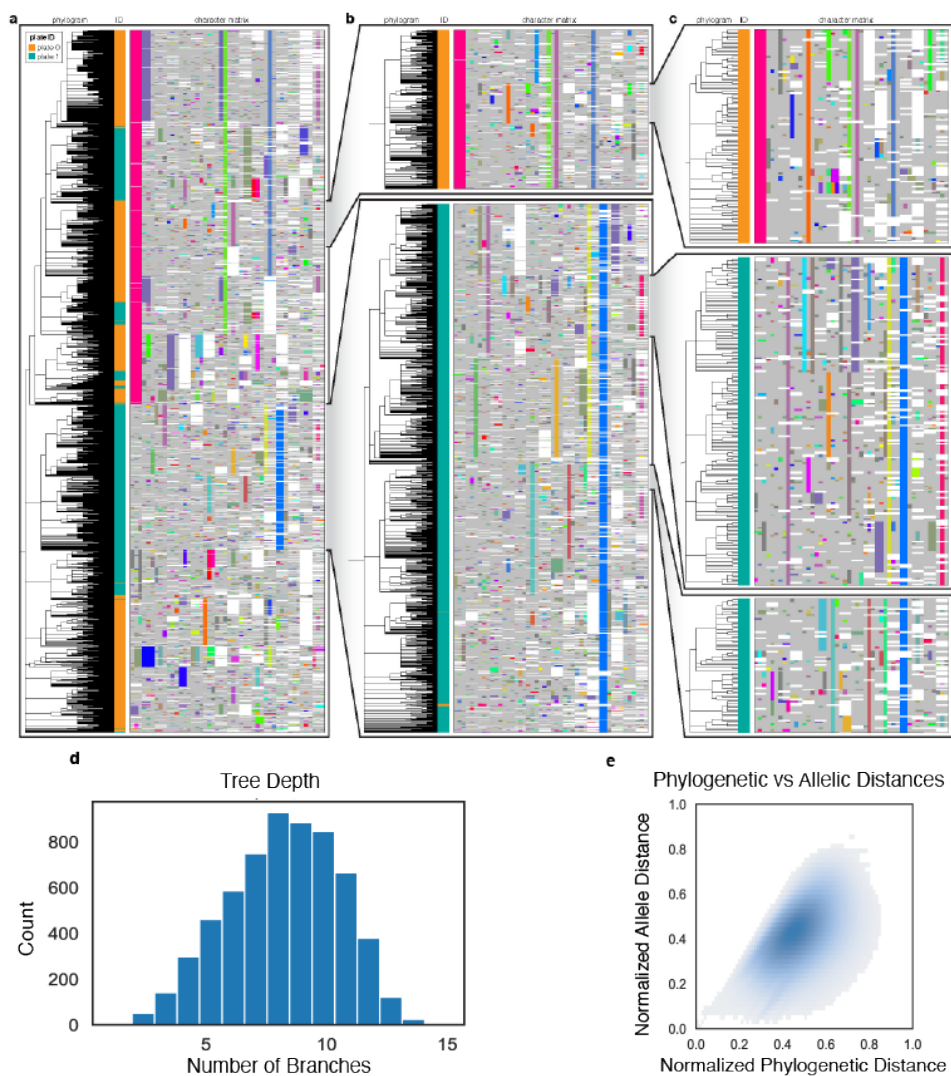


Figure 2.4: **Cassiopeia can reconstruct high-resolution phylogenetic trees from empirical lineage tracing data.** The full phylogenetic tree for Clone 3 (a), consisting of 7,289 cells, was reconstructed using Cassiopeia-Hybrid (with priors), and is displayed. The phylogram represents cell-cell relationships, and each cell is colored by sample ID at the first split (plate 0 or 1). The character matrix is displayed with each unique character state (or "indel") represented by distinct colors. (Light gray represents uncut sites; white represents missing values.) Of these 7,289 cells, 96% were uniquely tagged by their character states. (b-c) Nested, expanded views of the phylogram and character matrices. As expected, Cassiopeia correctly relates cells with similar character states, and closely related cells are found within the same culture plate. (d) A histogram of the tree-depth of each leaf from the root (mean = 8.22, max = 15). (e) Concordance between normalized allelic distance and normalized phylogenetic distance (see Methods; Pearson's correlation = 0.53).

reflect the distances in the experimental tree. Thus, we test the reconstruction ability of an algorithm using two metrics for measuring the association between plate ID and substructure: "Meta Purity" and "Mean Majority Vote" (see Methods). Both are predicated on the assumption that, just as in the real experiment, as one descends the reconstructed tree, one would expect to find cells more closely related to one another. In this sense, we utilize these two metrics for testing homogeneous cell labels below a certain internal node in a tree, which we refer to as a "clade".

We use these statistics to evaluate reconstruction accuracy for Clone 3 with respect to the first split labels (i.e. plate 0 or 1, **Figure 2.5**). In doing so, we find that Cassiopeia-Greedy and -Hybrid consistently outperform Neighbor-Joining. We find overall consistent results for the remainder of clones reconstructed (**Figure 2.27**, and additionally when considering the subsequent split into four plates - **Figure 2.28**), although Cassiopeia's modules have the greatest advantage in larger reconstructions. Specifically, Camin-Sokal and Neighbor-Joining perform similarly to Cassiopeia's modules on clones with few cells (e.g. Clone 11) or with low cell diversity (e.g. Clone 5, where target sites are "exhausted", possibly due to too-fast cutting, (**Figure 2.3f**, **Figure 2.29**)). Both cases indicate that in smaller and less complex clones traditional algorithms may be sufficient for reconstruction. Additionally, many of the issues described previously - parallel evolution, missing data, and information content - contribute to inferential errors in this empirical dataset (for example, **Figure 2.30**).

Overall, we anticipate that this *in vitro* dataset will serve as a valuable empirical benchmark for future algorithm development. Specifically, we have demonstrated how this dataset can be used to evaluate the accuracy of inferred phylogenies and illustrate that Cassiopeia consistently outperforms Neighbor-Joining for the purposes of reconstructing trees from single-cell lineage tracing technolo-

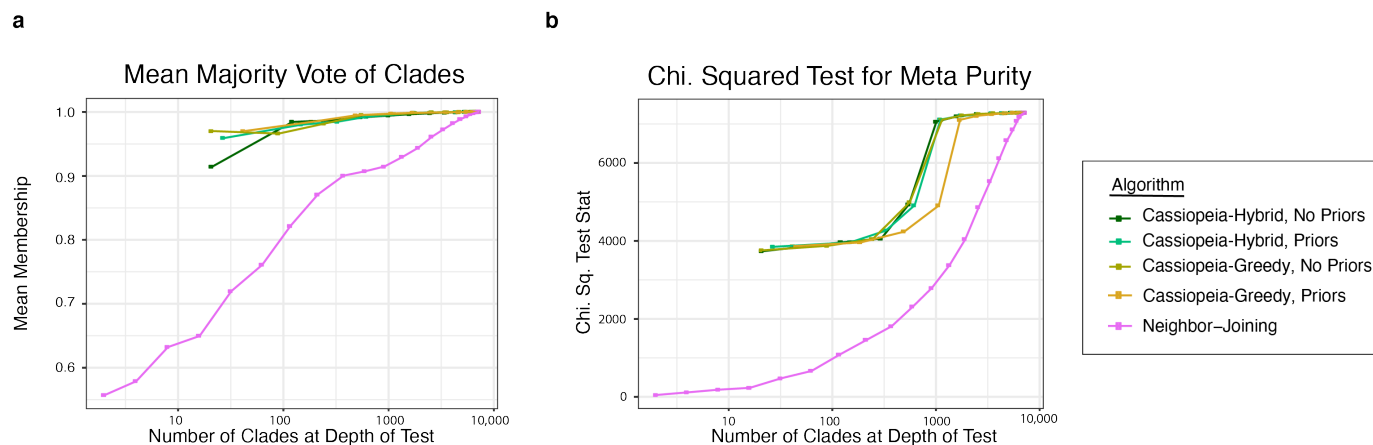


Figure 2.5: **Cassiopeia builds highly accurate trees from large empirical datasets.** The consistency between tree reconstructions are evaluated with respect to the first split. The Mean Majority Vote (a) and the Meta Purity test (b) were used for Cassiopeia-Hybrid and -Greedy (both with or without priors) and Neighbor-Joining. The statistics are plotted as a function of the number of clades at the depth of the test (i.e. the number of clades created by a horizontal cut at a given depth). All Cassiopeia approaches consistently outperform Neighbor-Joining by both metrics.

gies. Moreover, we demonstrate Cassiopeia’s scalability for reconstructing trees that are beyond the abilities of other maximum parsimony-based methods like Camin-Sokal as they currently have been implemented.

2.3.4 Generalizing Cassiopeia to Alternative & Future Technologies

While previous single-cell lineage tracing applications have proposed methods for phylogenetic reconstruction, they have been custom-tailored to the experimental system, requiring one to filter out common indels [236] or provide indel likelihoods [37]. We thus investigated how well Cassiopeia generalizes to other technologies with reconstructions of data generated with the GESTALT technology applied to zebrafish development [178, 206] (Figure 2.6a, Figure 2.31). Comparing Cassiopeia’s algorithms to Neighbor-Joining and Camin-Sokal (as applied in these previous studies [178, 206]),

we find that Cassiopeia-ILP consistently finds the most parsimonious solution. Furthermore, the Mean Majority Vote statistic also indicates that there is strong tissue-type enrichment as a function of tree depth, agreeing with Camin-Sokal's reconstruction which was used in the original study [206] (**Figure 2.6b**). Together, these results clearly demonstrates Cassiopeia's effectiveness for existing alternative lineage tracing technologies.

After establishing Cassiopeia's generalizability, we turned to investigating plausible next-generation lineage tracers. Recently, base-editing systems (**Figure 2.6c**) have been proposed to precisely edit $A > G$ [84], $C > T$ [149, 86] or possibly $C > N$ (N being any base as in [107]). The promise of base-editing lineage recorders is three-fold: first, a base editor would increase the number of editable sites (as compared to the ones that rely on Cas9-induced double-strand breaks [37, 178, 236]) although at the expense of number of states (at best 4, corresponding to A, C, T, and G). Second, a base-editing system would theoretically result in less dropout, since target site resection via Cas9-induced double-strand breaks is far less likely [149]. Third, it is hypothesized that base-editors would be less cytotoxic as it does not depend on inducing double strand breaks on DNA (although this relies on effective strategies for limiting off-target base-editing of DNA and RNA [290]). To evaluate the application of base editors for lineage tracing, we tested the performance of Cassiopeia in high-character, low-state regimes as would be the case in base editing (**Figure 2.6c**, see Methods). Using simulations with parameters deduced by a recent base editor application [107], we demonstrate that there appears to be an advantage of having more characters than states (**Figure 2.6c**). Of note, we did not observe any substantial deviation in these simulations from our initial scalability benchmarks in **Figure 2.7**. This suggests that base-editors may be a promising future direction for

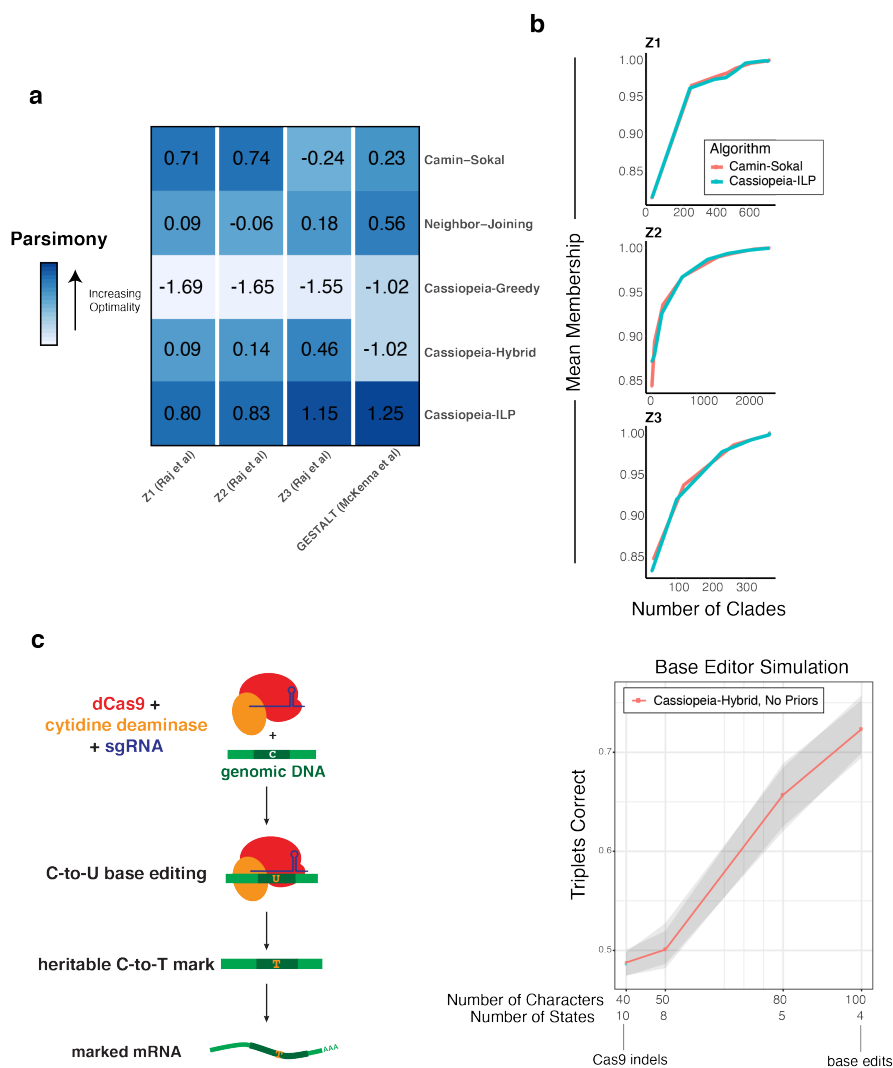


Figure 2.6: **Generalizing Cassiopeia & future design principles of CRISPR-enabled lineage tracers.** (a) Cassiopeia generalizes to alternative lineage tracing methods, as illustrated with the analysis of data from GESTALT technology [178, 206]). In a comparison of parsimony across Camin-Sokal, Neighbor-Joining, and Cassiopeia's methods, the Steiner-Tree approach consistently finds more parsimonious (i.e more optimal) solutions. Z-scores for each dataset are annotated over each tile. (b) Biological integrity of trees for each Zebrafish from Raj et al. [206], inferred with Cassiopeia-ILP, was assessed using the mean membership statistic (Methods) with respect to tissue type annotations from the original study. (c) Exploring information capacity of recorders with base-editors. A theoretical base-editor was simulated for 400 cells and reconstructions with Cassiopeia-Hybrid, with and without priors. We compared the accuracy of the reconstructions to the simulated tree using the triplets correct statistic. We describe the performance of Cassiopeia-Hybrid as the number of characters was increased (and consequently number of states was decreased.)

lineage tracing from a theoretical perspective.

Another potentially promising design consideration concerns the range of character mutation

rates and their variability across different target sites – a parameter that can be precisely engineered [131]. In this design, one would expect the variability to help distinguish between early and late branching points and consequently achieve better resolution of the underlying phylogeny [258, 134, 133]. We simulated “Phased Recorders” (**Figure 2.32**) with varying levels of target-site cutting variability and observe that this design allows for better inference when the distributions of mutation probabilities are more dispersed (**Figure 2.32b**). This becomes particularly useful when one can integrate accurate indel priors into Cassiopeia.

Overall, these results serve to illustrate how Cassiopeia and the simulation framework can be used to explore experimental designs. While there inevitably will be challenges in new implementations, these analyses demonstrate theoretically how design parameters can be optimized for downstream tree inference. In this way, the combination of our algorithms and simulations enables others to explore not only new algorithmic approaches to phylogenetic reconstruction but also new experimental approaches for recording lineage information.

2.4 Discussion

In this study, we have presented three resources supporting future single-cell lineage tracing technology development and applications. Firstly, we described Cassiopeia, a scalable and accurate maximum parsimony framework for inferring high-resolution phylogenies in single-cell lineage tracing experiments. Next, we introduced a simulation approach for benchmarking reconstruction methods and investigating novel experimental designs. Finally, we generated the largest and most diverse

empirical lineage tracing experiment to date, which we present as a reference for the systematic evaluation of phylogeny inference on real lineage tracing data. With the combination of these three resources, we have demonstrated the improved scalability and accuracy of Cassiopeia over traditional approaches for single-cell lineage tracing data and have explored design principles for more accurate tracing. To ensure broad use, we have made a complete software package, including the algorithms, simulation framework, and a processing pipeline for raw data, all publicly available at www.github.com/YosefLab/Cassiopeia.

The results highlighted in this manuscript demonstrate the variability in reconstruction accuracy for each of Cassiopeia's modules depending on the parameters. As introduced here, we suggest using Cassiopeia-ILP for small regimes (fewer than 200 cells) especially where there is low information capacity, Cassiopeia-Greedy for extremely large regimes (10,000 cells and larger), and Cassiopeia-Hybrid for intermediate regimes. Ideally, Cassiopeia-Hybrid could be run in all situations and transition appropriately between Cassiopeia-Greedy and -ILP depending on the complexity of the data. While here we use the number of cells as the criterion for transitioning, we anticipate there is a more consistent statistic (e.g. the entropy of a group of cells) for controlling the Cassiopeia-Hybrid transition that will make Cassiopeia more intuitive and effective with handling real data.

Though we illustrate that Cassiopeia provides the computational foundation necessary for future large-scale lineage tracing experiments, there are several opportunities for future improvement. First, the inclusion of prior probabilities increases Cassiopeia's performance only when parallel evolution is likely (e.g. with a high per-character mutation rate or in low character-state regimes). While maximum parsimony methods are attractive due to their non-parametric nature, future studies may

build on our work here by developing more powerful approaches for integrating prior mutation rates into maximum likelihood [69, 202] or Bayesian inference [118] frameworks, perhaps relying on recent literature that seeks to predict indel formation probabilities [142, 40, 5]. Future work in this space may also focus on using maximum parsimony solutions to further refine solutions in an effort to resolve branch length as with GAPML [70] or with paired transcriptomic observations [293]. Second, there exists a promising opportunity in developing new approaches for better handling of missing data. Determining a model which explicitly distinguishes between stochastic and heritable missing data may increase tree accuracy. Alternatively, adapting supertree methods (such as the Triple MaxCut algorithm [224]) for lineage tracing data may be an interesting direction as they have been effective for dealing with missing data (but only when this missing data is randomly distributed [286]). Aside from computational approaches for dealing with missing data, it is still unclear how much missing data is due to silencing, Cas9-resections, or stochastic dropout and experiments to elucidate the contributions of each will be helpful to the future design of lineage tracers. Third, while we provide theoretical and empirical evidence for our greedy heuristic, we note that there are opportunities for developing other heuristics - for example, by considering mutations in many characters rather than a single mutation as we do or using a distance-based heuristic.

The ultimate goal of using single-cell lineage tracers to create precise and quantitative cell fate maps will require sampling tens of thousands of cells (or more), possibly tracing over several months, and effectively inferring the resulting phylogenies. While recent studies [218] have highlighted the challenges in creating accurate CRISPR-recorders, our results suggest that with adequate technological components and computational approaches complex biological phenomena can be dissected

with single-cell lineage tracing methods. Specifically, we show that Cassiopeia and the benchmarking resources presented here meet many of these challenges. Not only does Cassiopeia provide a scalable and accurate inference approach, but also our benchmarking resources enable the systematic exploration of more accurate algorithms as well as more robust single-cell lineage tracing technologies. Taken together, this work forms the foundation for future efforts in building detailed cell fate maps in a variety of biological applications.

2.5 Methods

In vitro lineage tracing experiment

Plasmid design and cloning

The Cas9-mCherry lentivector, pHR-UCOE-SFFV-Cas9-mCherry(to be added to Addgene), was designed for stable, constitutive expression of enzymatically active Cas9, driven by the viral SFFV promoter, insulated with a minimal universal chromatin opening element (minUCOE), and tagged with C-terminal, self-cleaving P2A-mCherry. PCTXX is derived from pMH0001 (Addgene Cat#85969, active Cas9) with the BFP tag exchanged with mCherry. The P2A-mCherry tag was PCR amplified from pHR-SFFV-KRAB-dCas9-P2A-mCherry (Addgene Cat #60954; forward: GAGCAACG-GCAGCAGCGGATCCGGAGCTACTAACTTCAG; reverse: ATATCAAGCTTGCATGCCTGCAGGTC GACTTACTACTTGT ACAGCTCGTCCATGC) and inserted using Gibson Assembly (NEB) into SbfI/BamHI-digested pMH0001 (active Cas9). Resulting plasmid was used for lentiviral production as

described below.

The Target Site lentivector, PCT48 (available on Addgene), was derived from the reverse lentivector PCT5 (available on Addgene) containing GFP driven by the EF1a promoter. The sequence of the 10X amplicon with most common polyA location is the following:

```
AATCCAGCTAGCTGTGCAGCNNNNNNNNNNNNNNNNNATTCAACTGCAGTAATGCTACCT
CGTACTCACGCTTTCCAAGTGCTTGGCGTGCATCTCGGTCCTTTGTACGCCGAAAA
ATGGCCTGACAACTAAGCTACGGCAGCTGCCATGTTGGGTCATAACGATATCTCTG
GTTTCATCCGTGACCGAACATGTCATGGAGTAGCAGGAGCTATTAATTCGCGGAGGAC
AATGCGGTTTCGTAGTCACTGTCTTCCGCAATCGTCCATCGCTCCTGCAGGTGGCCTA
GAGGGCCCGTTTAAACCCGCTGATCAGCCTCGACTGTGCCTTCTAGTTGCCAGCCAT
CTGTTGTTTGCCCCTCCCCCGTGCCTTCCTTGACCCTGGAAGGTGCCACTCCCCTG
TCCTTTCTAATAAAAAAAAAAAAAAAAAAAAAAAAAA
```

where N denotes our 14bp random integration barcode. PCT5 was digested with SfiI and EcoRI within the 3'UTR of GFP. The Target Site sequence was ordered as a DNA fragment (gBlock, IDT DNA) containing three Cas9 cut-sites and a high diversity, 14-basepair randomer (integration barcode, or intBC). The fragment was PCR amplified with primers containing Gibson assembly arms compatible with SfiI/EcoRI-digested PCT5 (forward: GATGAGCTCTACAAATAATTAATTAA-GAATTCGTCACGAATCCAGCTAGCTGT; reverse: GGTTTAAACGGGCCCTCTAGGCCACCTGCA

GGAGCGATGG). The amplified Target Site fragment was inserted into the digested PCT5 backbone using Gibson Assembly. The assembled lentivector library was transformed into MegaX competent bacterial cells (Thermo Fisher) and grown in 1L of LB with carbenicillin at 100 $\mu\text{g}/\text{mL}$. Lentivector plasmid was recovered and purified by GigaPrep (Qiagen), and used for high-diversity lentiviral production as described below.

The triple-sgRNA-BFP-PuroR lentivector, PCT61 (available on Addgene), is derived from pBA392 (available on Addgene) as previously described [3, 130] containing three sgRNA cassettes driven by distinct U6 promoters and constitutive BFP and puromycin-resistance markers for selection. Importantly, the three PCT61 sgRNAs are complementary to the three cut-sites in the PCT48 Target Site. To slow the cutting kinetics of the sgRNAs to best match the timescale involved in the *in vitro* lineage tracing experiments [37], the sgRNAs contain precise single-basepair mismatches that decrease their avidity for the cognate cut-sites [91]. The triple-sgRNA lentivector was cloned using four-way Gibson assembly as described in [130]. Resulting plasmid was used for lentiviral production as described below.

Cell culture, DNA transfections, viral preparation, and cell line engineering

A549 cells (human lung adenocarcinoma line, ATCC CCL-185) and HEK293T were maintained in Dulbecco's modified eagle medium (DMEM, Gibco) supplemented with 10% FBS (VWR Life Science Seradigm), 2 mM glutamine, 100 units/mL penicillin, and 100 $\mu\text{g}/\text{mL}$ streptomycin. Lentivirus was produced by transfecting HEK293T cells with standard packaging vectors and TransIT-LTI trans-

fection reagent (Mirus) as described in ([3]). Target Site (PCT48) lentiviral preparations were concentrated 10-fold using Lenti-X Concentrator (Takara Bio). Viral preparations were frozen prior to infection. Triple-sgRNA lentiviral preparations were titered and diluted to a concentration to yield approximately 50% infection rate.

To construct the lineage tracing-competent cell line, A549 cells were transduced by serial lentiviral infection with the three lineage tracing components: (1) Cas9, (2) Target Site, and (3) triple-sgRNAs. First, A549 cells were transduced by Cas9 (mCherry) lentivirus and mCherry+ cells were selected to purity by fluorescence-activated cell sorting on the BD FACS Aria II. Second, A549-Cas9 cells were transduced by concentrated Target Site (GFP) lentivirus and GFP+ cells were selected by FACS; after sorting, Target Site infection and sorting were repeated two more times for a total of three serial lentiviral transfections, sorting for cells with progressively higher GFP signal after each infection. This strategy of serial transfection with concentrated lentivirus yielded cells with high copy numbers of the Target Site, which were confirmed by quantitative PCR. Third, A549 cells with Cas9 and Target Site were transduced by titered triple-sgRNA (BFP-PuroR) lentivirus and selected as described below.

***In vitro* lineage tracing experiment, single-cell RNA-seq library preparation, and sequencing**

One day following triple-sgRNA infection, cells were trypsinized to a single-cell suspension and counted using an Accuri cytometer (BD Biosciences). Approximately 25 cells were plated in a single well of a 96-well plate. Seven days post-infection, cells were trypsinized and split evenly into two

wells of a 96-well plate. Cells stably transduced by triple-sgRNA lentivirus were selected by adding puromycin at 1.5 $\mu\text{g}/\text{mL}$ on days 9 and 11 post-infection; puromycin-killed cells were removed by washing the plate with fresh medium. After 14 days, cells were trypsinized and split evenly for a second time into four wells of a 6-well plate. Finally, after 21 days in total, cells from the four wells were trypsinized to a single-cell suspension and collected.

Cells were washed with PBS with 0.04% w/v bovine serum albumin (BSA, New England Biolabs), filtered through 40 μm FlowMi filter tips filter tips (Bel-Art), and counted according to the 10x Genomics protocol. Approximately 14,000 cells per sample were loaded (expected yield: approximately 10,000 cells per sample) into the 10x Genomics Chromium Single Cell 3' Library and Gel Bead Kit v2, and cDNA was reverse-transcribed, amplified, and purified according to the manufacturer's protocol. Resulting cDNA libraries were quantified by BioAnalyzer, yielding the expected size distribution described in the manufacturer's protocol.

To prepare the Target Site amplicon sequencing library, resulting amplified cDNA libraries were further amplified with custom, Target Site-specific primers containing P5/P7 Illumina adapters and sample indices (forward: CAAGCAGAAGACGGCATAACGAGATXXXXXXXXX GTCTCGTGGGCTCG-GAGATGTGTATAAGAGACAGAATCCAGCTAGCTGTGCAGC;
reverse: CAAGCAGAAG ACGGCATAACGAGATXXXXXXXXXGTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGGCATGGACGAGCTGTACAAGT; "X" denotes sample indices). PCR amplification was performed using Kapa HiFi HotStart ReadyMix, as in [3], according to the following program:

melting at 95°C for 3 minutes, then 14 cycles at 98°C for 15 seconds and 70°C for 20 seconds. Approximately 12 fmol of template cDNA were used per reaction; amplification was performed in quadruplicate to avoid PCR-induced library biases, such as jack-potting. PCR products were re-pooled and purified by SPRI bead selection at 0.9x ratio and quantified by BioAnalyzer.

Target Site amplicon libraries were sequenced on the Illumina NovaSeq S2 platform. Due to the low sequence complexity for the Target Site library, a phiX genomic DNA library was spiked in at approximately 50% for increased sequence diversity. The 10x cell barcode and unique molecular identifier (UMI) sequences were read first (R1: 26 cycles) and the Target Site sequence was read second (R2: 300 cycles); sample identities were read as indices (I1 and I2: 8 cycles, each). Over 550M sequencing clusters passed filter and were processed as described below. All raw and processed data are available through GEO Series accession GSE146712.

Processing Pipeline

Read Processing

Each target site was sequenced using the Illumina Nova-seq platform, producing 300bp long-read sequences. The Fastq's obtained were quantitated using 10x's cellranger suite, which simultaneously corrects cell barcodes by comparing against a whitelist of 10x's approved cell barcodes.

For each cell, a consensus sequence for each unique molecule identifier (UMI) was produced by

collapsing similar sequences, defined by those sequences differing by at most 1 Levenshtein distance. A directed graph is constructed, where sequences with identical UMI's are connected to one another if the sequences themselves differ by at most one Levenshtein distance. Then, UMI's in this network are collapsed onto UMI's that have greater than or equal number of reads. This produces a collection of sequences indexed by the cell barcode and UMI information (i.e. there is a unique sequence associated with each UMI).

Before aligning all sequences to the reference, preliminary quality control is performed. Specifically, in cases where UMI's in a given cell still have not been assigned a consensus sequence, the sequence with the greatest number of reads is chosen. UMIs with fewer than 2 reads are filtered out, and cells with fewer than 10 UMIs are filtered out as well. Finally, a filtered file in Fastq format is returned.

Allele Calling

Alignment is performed with Emboss's Water local alignment algorithm. Optimal parameters were found by performing a grid search of gap open and gap extend parameters on a set of 1,000 simulated sequences, comparing a global and local alignment strategy. We found a gap open penalty of 20.0 and a gap extension penalty of 1.0 produced optimal alignments. The "indels" (insertions and deletions resulting from the Cas9 induced double-strand break) at each cut site in the sequences are obtained by parsing the cigar string from the alignments. To resolve possible redundancies in indels resulting from Cas9 cutting, the 5' and 3' flanking 5-nucleotide context is reported for each

indel.

UMI Error Correction

To correct errors in the UMI sequence either introduced during sequencing, PCR preparation, or data processing, we leverage the allele information. UMIs are corrected within groups of identical cell barcode-integration barcode pairs (i.e. we assume that only UMIs encoding for the same intBC in a given cell can be corrected). We reason that ideally, for a given integration barcodes, a cell will only report one sequence, or allele. Within these "equivalence classes," UMIs that differ by at most 1 Levenshtein distance (although this number can be user-defined) are corrected towards the UMI with a greater number of reads.

Cell-based Filtering

With the UMI corrected and indels calculated, the new "molecule table" is subjected to further quality control. Specifically, UMIs are filtered based on the number of reads (dynamically set to be the 99th percentile of the reads divided by 10), integration barcodes (denoting a particular integration site) can be error corrected based on a minimum hamming distance and identical indels (referred to as alleles), and in the case where multiple alleles are associated with a given integration barcode a single allele is chosen based on the number of UMIs associated with it.

Calling Independent Clones

Collections of cells part of the same clonal population, are identified by the set of integration barcodes each cell contains. Because all cells in the same clone are clonal, we reasoned that cells in the same clone should all share the same set of integration barcodes that the progenitor cell contained. Because of both technical artifacts (e.g. sequencing errors, PCR amplification errors) and biological artifacts (e.g. bursty expression, silenced regions) however, rather than looking for sets of non-overlapping sets, we perform an iterative clustering procedure. We begin by selecting the intBC that is shared amongst the most cells and assign any cell that contains this barcode to a cluster and remove these cells from the pool of unassigned cells. We perform this iteratively until at most k percent (in our case defined as .5% of cells are unassigned, which we assign to a "junk" clone.

Using the set of integration barcodes for each clone, we are able to identify doublets that consist of cells from different clones. Finally, after identifying doublets, to further filter out low quality integration barcodes, for each clone integration barcodes that are not shared by at least 10% of cells in a given clone are filtered out, producing the final allele table.

Guidelines for Final Quality Control

The thresholds discussed above are heuristic choices determined based on our hands-on experience with this type of target-site library processing. However, these thresholds will undoubtedly change depending on the sequencer used, the sequencing depth of the library, and the biological use case. For these reasons, we suggest that it is more effective to ensure that the final quality

control numbers indicate that the library was processed sufficiently.

We present distributions for the metrics we find to be the most useful in **Figure 2.23**: the UMIs per cellBC as a measure for how well sampled a cell is in (a), the reads per UMI as a measure for how confident one is of the UMI sequence in (b), UMIs per intBC as a measure for how confident one is of the called allele and intBC in (c), and a comparison of the number of UMIs versus the number of reads in (d), as a way of quickly assessing if there are any outlier UMIs.

Because this library was sequenced quite deeply, we do not expect typical applications to afford this degree of certainty. Instead, we suggest that cells should have at least 10 target-site UMIs, the reads/UMI distribution should have a mean at around 100-200 reads, and each intBC should have at least 5-10 UMIs associated with it. Cassiopeia's processing pipeline creates figures for each of these statistics after filtering and close attention should be paid to these figures during the processing of the target-site sequencing data.

Filtering of clones for Reconstruction

We filtered out clones upon two criteria: firstly, we removed clone 1 as we deduced that it had two defective guides; secondly, we removed lineages that reported fewer than 10% unique cells (thus removing clone 7). The remainder of clones were reconstructed.

Estimation of Per Character Mutation Rates

To estimate mutation rates per clone, we assume that every target site was mutated at the same rate and independently of one another across 15 generations. Assuming some mutation rate, p ,

per character, we know that the probability of not observing a mutation in d generations is $(1 - p)^d$ in a given character and that the probability of observing at least 1 mutation in that character is $1 - (1 - p)^d$. Then, giving this probability $1 - (1 - p)^d = m$ can be used as a probability of observing a mutated character in a cell and model the number of times a character appears mutated in a cell as a binomial distribution where the expectation is simply nm where n is the number of characters. Said simply, given this model, one would expect to see nm characters mutated in a cell). In this case, the empirical expectation is the mean number of times a given character appeared mutated in a cell (averaged across all cells), which we denote as K and propose that

$$K = nm = n * (1 - (1 - p)^d)$$

and thus p , the mutation rate, is

$$p = 1 - (1 - K/n)^d$$

Bulk Cutting Experiment to Determine Prior Probabilities of Indel Formation

Two and four days following triple-sgRNA (PCT61) infection, infected cells were selected by adding puromycin at 1.5 $\mu\text{g}/\text{mL}$; puromycin-killed cells were removed by washing the plate with fresh medium. Cells were split every other day, and 500k cells were collected on days 7, 14, and 28. Frozen cell pellets were lysed and the genomic DNA was extracted and purified by ethanol precipitation. The PCT48 Target Site locus was PCR amplified from genomic DNA samples (forward: TCGTCGGCAGCGTCA-

GATGTGTATAAGAGACAGAATCCAGCTAGCTGTGCAGC; reverse: GTCTCGTGGGCTCGGAGAT-GTGTATAAGAGACAGTCCGAGGCTGATCAGCG) and further amplified to incorporate Illumina adapters and sample indices (forward: AATGATACGGCGACCACCGAGATCTACACXXXXXXXXTCGT CGGCAGCGTCAG; reverse: CAAGCAGAAGACGGCATACGAGATXXXXXXXXGTCTCGTGGG CTCGGAG; "X" denotes sample indices). The subsequent amplicon libraries were sequenced on an Illumina MiSeq (paired end, 300 cycles each). Sequencing data was analyzed as described below.

Determining Prior Probabilities of Indel Formation

To determine the prior probabilities of edits, we leverage the fact that we have access to a large set of target sites (or intBCs) with a similar sequence (apart from the random barcode at the 5' end); namely, a total of 117 intBC across the 11 clones. To compute the prior probability for a given indel, we compute the empirical frequency of observing this mutation out of all unique edits observed. Specifically, we compute the prior probability of a given indel s , q_s as the following:

$$q_s = \frac{f(s)}{|I|}$$

where $f(s)$ is the number of intBC's that had s in at least one cell and $|I|$ is the number of intBCs that are present in the dataset.

As further support for this method, we used the bulk experiment consisting of many separately engineered A549 cells, as described in the previous section. The advantage of the bulk experiment is

that we have access to substantially more intBCs ($> 10k$), thus providing a more robust estimation of q_s . We therefore employed the same approach to estimate indel formation rates from the bulk data and find that the resulting rates correlate well with the indel rates estimated from the single cell lineage tracing experiment (**Figure 2.26**).

Doublet Detection

Methods to Detect Doublets

We hypothesized that doublets could come in two forms and that we could use various components of the intBC data structure to identify them. Namely, doublets could be of cells from the identical clone, here dubbed "intra-doublets", or doublets could be of cells from separate clones, here dubbed "inter-doublets."

In the case of "intra-doublets", we can utilize the fact that these cells will have a large overlap in their set of intBCs but will report "conflicting" alleles for each of these intBCs. Thus, to identify these doublets, we calculate the percentage of UMIs that are conflicting in each cell. Explicitly, for each cell we iterate over all intBCs and sum up the number of UMIs that correspond to an allele that conflicts with the more abundant allele for a given intBC; we then use the percentage of these UMIs to identify doublets. We perform this after all UMI and intBC correction in hopes of calling legitimate conflicts.

To deal with “inter-doublets”, we developed a classifier that leverages the fact that cells from different clones should have non-overlapping intBC sets. While this is the ideal scenario, often times intBCs are shared between clones for one of two reasons (1) the clustering assignments are noisy or (2) the transfections of intBCs resulted in two cells receiving the same intBC, even though cells are supposed to be progenitors of separate clones. Our strategy is thus: for each cell $c_i \in C$ calculate a “membership statistic”, $m_{i,k}$ for each clone $l_k \in L$. The membership statistic is defined as so:

$$m_{i,k} = \frac{\sum_{j \in I_k} \delta(i, j) p(j, k)}{\sum_{j \in I_k} (p(j, k))}$$

where I_k is the set of intBCs for the clone l_k and $p(j, k)$ is the prevalence rate of the intBC j in l_k . We use $\delta(i, j)$ as an indicator function for whether or not we observed the intBC j in the cell c_i . Intuitively, this membership statistic is a weighted similarity for how well the cell fits into each clone, where we are weighting by how much we are able to trust the intBC that is observed in the cell. To put all on the same scale, we normalize by total membership per cell, resulting in our final statistic, $m'_{i,k} = \frac{m_{i,k}}{\sum_{k'=0}^k m_{i,k'}}$ We then filter out doublets whose m' for their classified clone falls below a certain threshold.

Simulation of Doublets

We simulated two datasets to test our methods for identifying doublets and to find the optimal criterion on which to filter out doublets. To test this strategy, we took a single clone from our final Allele Table (the table relating all cells and their UMIs to clones) and formed 200 doublets by combining

the UMIs from two cells. We generated 20 of these datasets, and noted which cells were artificially introduced doublets.

Contrary to the strategy for simulating doublets from the same clone, we created artificial “inter” doublets from the final Allele Table by combining doublets from two different clones. Similarly, we generated 20 synthetic datasets each with 200 of these artificial doublets.

Identification of Decision Rule

To identify the optimal decision rule for calling both types of doublets, we tested decision rules ranging from 0 to 1.0 at 0.05 intervals and calculated the precision and recall at each of these rules. Taking these results altogether, we provide an optimal decision rule where the F-measure (or the weighted harmonic mean of the precision and recall) of these tests is maximal.

Algorithmic Approaches For Phylogenetic Reconstruction

One way to approach the phylogenetic inference problem is to view each target site as a “character” that can take on many different possible “states” (each state corresponding to an indel pattern induced by a CRISPR/Cas9 edit at the target site). Formally, these observations can be summarized in a “character matrix”, $M \in R^{n,m}$, which relates the n cells by a set of characters $\chi = \{\chi_1, \dots, \chi_m\}$ where each character χ_i can take on some k_i possible states. Here, each sample, or cell, can be described as a concatenation of all of their states over characters in a “character string”. From this

character matrix, the goal is to infer a tree (or phylogeny), where leaf nodes represent the observed cells, internal nodes represent ancestral cells, and edges represent a mutation event.

We first propose an adaption of a slow, but accurate, Steiner-Tree algorithm via Integer Lineage Programming (ILP) to the lineage tracing phylogeny problem. Then, we propose a fast, heuristic-based greedy algorithm which simultaneously draws motivation from classical perfect phylogeny algorithms, and the fact that mutations can only occur unidirectionally from the unmutated, or s_0 state. Lastly, we combine these two methods and present a hybrid method, which presents better results than our greedy approach, yet remains feasible to run over tens of thousands of cells.

Adaptation to Steiner Tree Problem

Steiner Trees are a general problem for solving for the minimum weight tree connecting a set of target nodes. For example, if given a graph $G = (V, E)$ over some V vertices and E edges, finding the Steiner-Tree over all $v \in V$ would amount to solving for the minimum spanning tree (MST) of G . While there exist polynomial time algorithms for the minimum-spanning tree, the general Steiner Tree problem, where the set of targets $T \subseteq V$ is designated, is NP-hard.

Previously, Steiner-Trees have been suggested to solve for the maximum parsimony solution to the phylogeny problem. Here, the graph would consist of all possible cells (both observed and unobserved) and each edge would consist of a possible evolutionary event connecting two states (e.g. a mutation). Generally, given a set of length- l binary "character-strings" (recall that these are the concatenation of all character states for a given sample), we can solve for the maximum

parsimony solution by finding the optimal Steiner Tree over the 2^l hypercube (i.e. graph). As a result, by converting our multi-state characters to binary characters via one hot encoding, theoretically, we should be able to compute the most parsimonious tree which best explains the observed data. However, in practice this method turns out to be infeasible, as we deal with hypercubes of size $O(2^{mn})$, where m is the number of characters, and n is the number of states. In the following, we will propose a method for estimating the underlying search space, providing us with a feasible solvable instance and a formulation of an Integer-Linear Programming (ILP) problem to solve for the optimal Steiner-Tree.

Approximation of Potential Graph

We first begin by constructing a directed acyclic graph (DAG) G , where nodes represent cells. We then take the source nodes, or nodes with in-degree 0, of G , and for each pair of source nodes, consider the latest common ancestor (LCA) they could have had. This LCA has an unmutated state for character χ_i if they disagree across two source nodes, and the same state as the two source nodes if they agree in value. If the edit distance between these two cells is below a certain threshold d , we add the LCA to G , along with directed edges to the two source nodes, weighted by the edit distance between the parent and the source. We repeat this process until only one node remains as a source: the root.

One may think that this step explodes with $O(n^2)$ complexity at each stage, where n is the number of source nodes in each prior stage, as we consider all pairs of source nodes. However, we note that the number of mutations per latest common ancestor is always less than both children, and

therefore, we eventually converge to the root. Therefore, when dealing with several hundred cells, the potential graph is feasible to calculate.

Furthermore, to add scalability to the approximation of the Potential Graph, we allow the user to provide a “maximum neighborhood size” which will be used to dynamically solve for the optimal LCA distance threshold d to use. One may think of this as the maximum memory or time allowed for optimizing a particular problem. Since the size of the Potential Graph can grow quite large in regards to the number of nodes, we iteratively create potential graphs for various threshold d and at each step ensure that the number of nodes in the network does not exceed the maximum neighborhood size provided. If at any point the number of nodes does exceed this maximum size, we return the potential graph inferred for an LCA threshold of $d - 1$.

Formulation of Integer Linear Programming Problem

Given our initial cells, S , the underlying potential graph drawn from such cells, G , and the final source node, or root, r from G , we are interested in solving for $\mathcal{T} = SteinerTree(r, S, G)$. We apply an integer linear programming (ILP) formulation of Steiner Tree, formulated in terms of network flows, with each demand being met by a flow from source to target. Below we present the Integer Linear Programming formulation for Steiner Tree. We use Gurobi [97], a standard ILP solver package

$$\begin{aligned}
& \text{minimize} && \sum_{(u,v) \in E} d_{uv}^b \cdot w(u, v) \\
& \text{subject to} && \sum_{(u,v) \in E} d_{uv} - \sum_{(v,w) \in E} d_{vw} = 0 && \forall v \notin S \cup \{r\} \\
& && \sum_{(r,w) \in E} d_{rw} = -|S| \\
& && \sum_{(u,s) \in E} d_{us} = 1 && \forall s \in S \\
& && d_{uv}^b \geq \frac{d_{uv}}{|S|} && \forall (u, v) \in E \\
& && d_{uv} \in \{0, \dots, |S|\} && \forall (u, v) \in E \\
& && d_{uv}^b \in \{0, \dots, 1\} && \forall (u, v) \in E
\end{aligned}$$

Each variable d_{uv} denotes the flow through edge (u, v) , if it exists; each variable d_{uv}^b denotes whether (u, v) is ultimately in the chosen solution sub-graph. The first constraint enforces flow conservation, and hence that the demands are satisfied, at all nodes and all conditions. The second constraint requires $|S|$ units of flow come out from the *root*. The third constraint requires that each target absorb exactly one unit of flow. The fourth constraint ensures that if an edge is used at any condition, it is chosen as part of the solution.

Below we explicitly define the algorithm in pseudocode.

```

1: function ILP-SOLVER(cells = S)
2:   Potential Graph G ← BUILD-POTENTIAL-GRAPH(S)
3:   if G == None then
4:     return GREEDY-SOLVER(S)
5:   r ← root of G
6:    $\mathcal{T}$  ← STEINER-TREE(r, G, S)
7:   return  $\mathcal{T}$ 

```

▷ Steiner Tree ILP Solver

```

8: function BUILD-POTENTIAL-GRAPH(cells = S, max lca length = k, max neighborhood size = N)
9:    $\mathcal{T}_0$  = None
10:  for all d ∈ [1, k] do
11:     $\mathcal{T}$  ← DiGraph()
12:    for all s ∈ S do
13:       $\mathcal{T}$  ←  $\mathcal{T} \cup \{s\}$ 
14:    sources ← all source nodes in  $\mathcal{T}$ 
15:    while len(sources) > 1 do
16:      for all v1, v2 ∈ sources do
17:        lca ← latest common ancestor of v1, v2
18:        if dist(lca, v1) + dist(lca, v2) ≤ d then
19:           $\mathcal{T}$  ←  $\mathcal{T} \cup \{(lca, v_1), (lca, v_2)\}$ 
20:        sources ← all source nodes in  $\mathcal{T}$ 
21:        if len(sources) ≥ N then
22:          return  $\mathcal{T}_{d-1}$ 
23:       $\mathcal{T}_d$  ←  $\mathcal{T}$ 
24:  return  $\mathcal{T}$ 

```

Stability Analysis of the Maximum Neighborhood Size Parameter

To evaluate the stability of the user-defined maximum neighborhood size parameter, we assessed the accuracy of the reconstructions for parameters varying from 800 to 14,000. We used trees simulated under default conditions (400 samples, 40 characters, 40 states per character, 11 generations, 2.5% mutation rate per character, and a mean dropout rate of 17%). The accuracy of trees were compared to the tree generated with a parameter of 14,000 using the triplets correct statistic. We

used 10 replicates to provide a sense for how stable a given accuracy is.

In addition to providing measures of accuracy, we also provide the optimal LCA threshold d found for a given maximum neighborhood size during the inference of these potential graphs. Using these analysis, we found that a maximum neighborhood size of 10,000 nodes seemed to be an ideal tradeoff between scalability and accuracy (as it is in the regime where accuracy saturates) for our default simulations. This corresponded to a mean LCA threshold, d , of approximately 5.

Heuristic-Based Greedy Method

On Perfect Phylogeny & Single Cell Lineage Tracing

In the simplest case of phylogenetics, each character is binary (i.e. $k_i = 2, \forall i \in m$) and can mutate at most once. This case is known as "perfect phylogeny" and there exist algorithms (e.g. a greedy algorithm by Dan Gusfield [98]) for identifying if a perfect phylogeny exists over such cells, and if so find one efficiently in time $O(mn)$, where m is the number of characters and n are the number of cells. However, several limitations exist with methods such as Gusfield's algorithm. One potential problem in using existing greedy perfect phylogeny algorithms for lineage tracing is that they require the characters to be binary. Indeed, if the characters are allowed to take any arbitrary number of states, the perfect phylogeny problem becomes NP-hard. However, while the number of states (CRISPR/Cas9-induced indels at a certain target site) in lineage tracing data can be large, these data benefit from an additional restriction that makes it more amenable for analysis with a greedy algorithm. Below, we show that because the founder cell (root of the phylogeny) is unedited

(i.e. includes only uncut target sites) and that the mutational process is irreversible, we are able to theoretically reduce the multi-state instance (as observed in lineage tracing) to a binary one so that it can be resolved using a greedy algorithm.

A second remaining problem in using these perfect phylogeny approaches is that we cannot necessarily expect every mutation to occur exactly once. In theory, it may happen that the same indel pattern is induced in exactly the same target site on two separate occasions throughout a lineage tracing experiment, especially if a large number of cell cycles takes place. A final complicating factor is that these existing greedy algorithms often assume that all character-states are known, whereas lineage tracing data is generated by single-cell sequencing, which often suffers from limited sensitivity and an abundance of “dropout” (stochastic missing data) events.

The Greedy Algorithm

We suggest a simple heuristic for a greedy method to solve the maximum parsimony phylogeny problem, motivated by the classical solution to the perfect phylogeny problem and irreversibility of mutation. Namely, we consider the following method for building the phylogeny: Given a set of cells, build a tree top-down by splitting the cells into two subsets over the most frequent mutation. Repeat this process recursively on both subsets until only one sample remains.

Formally, we choose to split the dataset into two subsets, $O_{i,j}$ and $\bar{O}_{i,j}$, such that $O_{i,j}$ contains cells carrying mutation s_j in χ_i , and $\bar{O}_{i,j}$ contains cells without s_j in χ_i . We choose i, j based on the following criteria:

$$i, j = \arg \max_{i, j} n_{i, j}$$

where $n_{i, j}$ is the number of cells that carry mutation s_j in character χ_i . We continue this process recursively until only one sample exists in each subset. We note that this method operates over cells with non-binary states, solving the first of problems addressed earlier.

A major caveat exists with methods such as the greedy method proposed by Gusfield, as well as the one proposed by us thus far: namely, they assume all character states are known (i.e. no dropout). However, in our practice, we often encounter dropout as a consequence of Cas9 cutting or stochastic, technical dropout due to the droplet-based scRNA-seq platform. To address this problem in our greedy approach, during the split stage, these cells are not initially assigned to either of the two subsets, $O_{i, j}$ or $\bar{O}_{i, j}$. Instead, for each individual sample which contains a dropped out value for chosen split character χ_i , we calculate the average percentage of mutated states shared with all other cells in $O_{i, j}$ and $\bar{O}_{i, j}$ respectively, and assign the sample to the subset with greater average value.

Appending the dropout resolution stage with the initial split stage, we present our greedy algorithm below in its entirety.

```

1: function GREEDY-SOLVER( $cells = S$ , prior probabilities =  $p$ )
2:   if  $len(S) = 1$  then
3:     return  $S$ 
4:    $root \leftarrow$  latest common ancestor across all  $S$ 
5:    $i, s_j \leftarrow$  maximally occurring character mutation pair in  $S$  weighted by priors  $p$ 
6:    $O_{i,j} \leftarrow$  all cells in  $S$  with mutation  $s_j$  in  $\chi_i$ 
7:    $\bar{O}_{i,j} \leftarrow$  all cells in  $S$  without mutation  $s_j$  in  $\chi_i$  and without dropout for  $\chi_i$ 
8:    $D_i \leftarrow$  all cells in  $S$  with dropout for  $\chi_i$  ▷ Note  $O_{i,j} \cup \bar{O}_{i,j} \cup D_i = S$ 
9:   for all  $s \in D_i$  do
10:    if  $s$  shares more mutated states on average with cells in  $O_{i,j}$  over  $\bar{O}_{i,j}$  then
11:       $O_{i,j} \leftarrow O_{i,j} \cup \{s\}$ 
12:    else
13:       $\bar{O}_{i,j} \leftarrow \bar{O}_{i,j} \cup \{s\}$ 
14:     $\mathcal{T}_L, \mathcal{T}_R \leftarrow$  GREEDY-SOLVER( $O_{i,j}, p$ ), GREEDY-SOLVER( $\bar{O}_{i,j}, p$ )
15:     $r_L, r_R \leftarrow$  root of  $\mathcal{T}_L, \mathcal{T}_R$  respectively
16:     $\mathcal{T} \leftarrow \mathcal{T}_L \cup \mathcal{T}_R \cup \{root\}$ 
17:     $\mathcal{T} \leftarrow \mathcal{T} \cup \{(root, r_L), (root, r_R)\}$ 
18:  return  $\mathcal{T}$ 

```

Overall, this method is very efficient, and scales well into tens of thousands of cells. Below, we show via proof below that this algorithm can find perfect phylogeny if one exists.

Cassiopeia-Greedy Algorithm Can Solve Multi-State Perfect Phylogeny

Here we show that while not required, Cassiopeia can solve the multi-state perfect phylogeny problem optimally. Importantly, however, Cassiopeia's effectiveness makes no assumption about perfect phylogeny existing in the dataset but rather leverages this concept to provide a heuristic for scaling into larger datasets.

To show how Cassiopeia's greedy method can solve perfect phylogeny optimally, we begin by introducing a few clarifying definitions prior to the main theorem. We define M as the original n cells

by n character k -state matrix (i.e. entries $\in \{s_0, \dots, s_{k-1}\}$). We say M has a zero root perfect phylogeny if there exists a tree \mathcal{T} over its elements and character extensions such that the state of the root is all zeros and every character state are mutated into at most once. In addition, we assume that all non-leaf nodes of \mathcal{T} have at least two children (i.e. if they only have one child, collapse two nodes into one node). Finally, we offer a definition for *character compatibility*:

Definition 1. (*Character Compatibility*). For a pair of binary characters, (χ_1, χ_2) , where the sets (O_1, O_2) contain the sets of cells mutated for χ_1 and χ_2 , respectively, we say that they are compatible if one of the following is true:

- $O_1 \subseteq O_2$
- $O_2 \subseteq O_1$
- $O_1 \cap O_2 = \emptyset$

This definition extends to multi-state characters as well, assuming they can be binarized.

Before proving the main theorem, we first prove the following lemma:

Lemma 1. *If M has a perfect phylogeny, then the most frequent character, mutation pair appears on an edge from the root to a direct child node.*

Proof. WLOG let $\chi_i : s_0 \rightarrow s_j$ denote the maximally occurring character, mutation pair within M .

Suppose by contradiction that this mutation does not appear on an edge directly from root to a child,

but rather on some edge (u, v) that is part of a sub-tree whose root r^* , is a direct child of the root. As r^* has at least two children, this implies that the mutation captured from the root to r^* must be shared by strictly more cells than $\chi_i : s_0 \rightarrow s_j$, thereby reaching a contradiction on $\chi_i : s_0 \rightarrow s_j$ being the maximally occurring mutation. \square

Theorem 1. *The greedy algorithm accurately constructs a perfect phylogeny over M if one exists.*

Proof. We approach via proof by induction. As a base case, a single is trivially a perfect phylogeny over itself.

Now suppose by induction that for up to $n - 1$ cells, if there exists a perfect phylogeny \mathcal{T} over such cells, then the greedy algorithm correctly returns the perfect phylogeny. Consider the case of n cells. By the above lemma, we know we can separate these n cells into two subsets based on the most frequent character, mutation pair $\chi_i : s_0 \rightarrow s_j$, $O_{i,j}$ and $\bar{O}_{i,j}$, where $O_{i,j}$ contains cells with mutation s_j over χ_i , and $\bar{O}_{i,j} = M - O_{i,j}$. By induction, the greedy algorithm correctly returns two perfect phylogenies over $O_{i,j}$ and $\bar{O}_{i,j}$, which we can merge at the root, giving us a perfect phylogeny over n cells. \square

Accounting for Prior Probability of Mutations

In most situations, the probability of mutation to each distinct state may not be uniform (i.e. character χ_1 mutating from the unmutated state s_0 to state s_4 may be twice as likely as mutating to state s_6). Therefore, we incorporate this information into choosing which character and mutation to split over

based on the following criteria:

$$i, j = \arg \min_{i,j} p_i(s_0, s_j)^{f(n_{i,j})}$$

where $p_i(s_0, s_j)$ is the probability that character χ_i mutates from the unmutated state s_0 to s_j and $f(n_{i,j})$ is some transformation of the number of cells that report mutation j in character i that is supposed to reflect the future penalty (number of independent mutations of character i to state j) we will have to include in the tree if we do not pick i, j as our next split. After a comparison of 5 different transformations (**Figure 2.21**), we find that $f(n_{i,j}) = n_{i,j}$ gives the best performance, leaving us with the following criteria for splittings:

$$i, j = \arg \min_{i,j} p_i(s_0, s_j)^{n_{i,j}}$$

A Hybrid Method for Solving Single Cell Lineage Tracing Phylogenies

Due to the runtime constraints of the Steiner Tree Method, it is infeasible for such method to scale to tens of thousand of cells. Therefore, we build a simple hybrid method which takes advantage of the heuristic proposed in the greedy algorithm and the theoretical optimality of the Steiner Tree method.

Recall that in the greedy method, we continued to choose splits recursively until only one sample was left per subset. In this method, rather than follow the same process, we choose a cutoff for each subset (e.g. 200 cells). Once a subset has reached a size lower than said cutoff, we feed each individual subset into the Potential Graph Builder and Steiner Tree solver, which compute an optimal phylogeny for the subset of cells. After an optimal subtree is found, we merge it back into the greedy

tree. Therefore, we build a graph whose initial mutations are chosen from the greedy method, and whose latter mutations are chosen more precisely via the Steiner Tree approach.

Below we present a pseudo-code algorithm for the hybrid method. We note the slight difference in greedy from before. Namely, greedy additionally accepts a cutoff parameter, and in addition to returning a network built up to that cutoff, returns all subsets that are still needed to be solved.

```

1: function CASSIOPEIA-HYBRID(cells =  $S$ , greedy cutoff =  $g$ )
2:    $\mathcal{T}, \mathcal{S} \leftarrow$  GREEDY-SOLVER( $S, g$ )
3:   for all  $S' \in \mathcal{S}$  do
4:      $\mathcal{T} \leftarrow \mathcal{T} \cup$  ILP-SOLVER( $S'$ )
5:   return  $\mathcal{T}$ 

```

This approach scales well when each instance of Steiner Tree is ran on an individual thread, and thus often takes only a few hours to run on several thousand cells.

Theoretical Analysis of Parallel Evolution

Estimating First and Second Moments of Double Mutations

Expected Number of Double Mutations. Under the framework of our simulation, we assume that each at each generation, every cell divides, and then each character of each cell undergoes random mutation independently. Let p be the probability that a particular character mutates, and q be the probability the character took on a particular mutated state given that it mutated. Let T be the true phylogenetic tree over the samples. According to our model, T must be a full binary tree, and the samples are leaves of T . Let X be the total number of times a particular mutation occurred in the

T . Let $X_{u,v}$ be an indicator variable for edge (u, v) such that:

$$X_{u,v} = \begin{cases} 1 & \text{if a mutation occurs on edge } (u, v) \\ 0 & \text{otherwise} \end{cases}$$

Let h be the height of the T , which is equalled to the number of generations. If v is at depth d in T , then the probability that a mutation occurs at (u, v) is $pq(1-p)^{d-1}$. Since there are 2^d nodes at depth d , we have:

$$\begin{aligned} E(X) &= \sum_{(u,v) \in T} E(X_{u,v}) \\ &= \sum_{d=1}^h 2^d pq(1-p)^{d-1} \\ &= \frac{2pq((2-2p)^h - 1)}{1-2p} \end{aligned} \tag{Eq. 1}$$

Let $n = 2^h$ is the number of cells in our sample. If $p > 0.5$, $E(X) \leq 2pq/(2p-1)$, if $p = 0.5$, $E(X) = 2pqh = O(\log n)$, and if $p < 0.5$, $E(X) = O(n^{\frac{1}{\log_2 2-2p}})$. Moreover, for fixed h , $E(X)$ has a single peak for $p \in [0, 1]$, meaning that it increases with p for sufficiently small values of p , and always increases with q . Intuitively, this is because $E(X)$ is small if 1) p is small enough that the character never mutates much throughout the experiment or 2) p is large enough that most mutations occur near the top of the tree, resulting in the extinction of unmutated cells early in the experiment. While $E(X)$ peaks for values of p in between, it is always directly proportional to q because X is simply equalled to q time the number of times the character mutated.

Variance of Double Mutations. We can compute the variance as:

$$\begin{aligned} \text{Var}(X) &= E(X^2) - E(X)^2 \\ &= 2 \sum_{(u,v) \neq (u',v')} E(X_{u,v} X_{u',v'}) + E(X) - E(X)^2 \end{aligned}$$

To compute $E(X_{u,v} X_{u',v'})$, we note that for a given pair of edges (u, v) and (u', v') , such that $LCA(u, u')$ is at depth d , u is at depth $d + l$, and u' is at depth $l + k$, the probability that a mutation occurred on both edges is $p^2 q^2 (1 - p)^{d+l+k}$. Thus, we have:

$$\begin{aligned} \sum_{(u,v) \neq (u',v')} E(X_{u,v} X_{u',v'}) &= \sum_{d=0}^{h-1} 2^d \sum_{k=0}^{h-d-1} \sum_{l=0}^{h-d-1} 2^{l+k} p^2 q^2 (1 - p)^{d+l+k} \\ &= p^2 q^2 \sum_{d=0}^{h-1} (2 - 2p)^d \left(\sum_{k=0}^{h-d-1} (2 - 2p)^k \right)^2 \\ &= \frac{p^2 q^2}{(2p - 1)^2} \sum_{d=0}^{h-1} (2p - 2)^d ((2p - 2)^{h-d} - 1)^2 \\ &\leq \frac{p^2 q^2}{(2p - 1)^2} \sum_{d=0}^{h-1} (2p - 2)^{2h-d} \\ &= (2p - 2)^{h+1} \frac{p^2 q^2}{(2p - 1)^2} \sum_{d=0}^{h-1} (2p - 2)^d \\ &\leq \frac{p^2 q^2 (2p - 2)^{2h+1}}{(2p - 1)^3} \end{aligned}$$

Thus, we can bound the variance as follows:

$$\text{Var}(X) \leq \frac{2p^2 q^2 (2p - 2)^{2h+1}}{(2p - 1)^3} + \frac{2pq(1 - (2 - 2p)^h)}{2p - 1} - \frac{4p^2 q^2 (1 - (2 - 2p)^h)^2}{(2p - 1)^2} \quad (\text{Eq. 2})$$

This means that in the case that $p > 0.5$:

$$\text{Var}(X) \leq \frac{2p^2 q^2}{(2p - 1)^3} + \frac{2pq}{2p - 1} - \frac{4p^2 q^2}{(2p - 1)^2}$$

In the case that $p = 0.5$:

$$Var(X) = O(h^3) = O(\log^3(n))$$

In the case that $p < 0.5$:

$$Var(X) = O(n^{\frac{2}{\log_2 2 - 2p}})$$

Least Squares Linear Estimate & Negative Correlation Between Frequency and Number of Double Mutations

To justify the greedy, we must show that if a mutation occurs frequently, then it is likely to have occurred less times throughout the experiment. Let Y be the frequency of a particular mutation in the samples. We estimate X given Y using the least squares linear estimate (LLSE) as follows:

$$L(X|Y) = E(X) + \frac{CoV(X, Y)}{Var(Y)}(Y - E(Y)) \quad (\text{Eq. 3})$$

Since $CoV(X, Y) = E(XY) - E(X)E(Y)$, we need only to compute $E(XY)$, which we do by expressing X and Y in terms of the same indicators:

$$Y = \frac{1}{2^h} \sum_{(u,v) \in T} 2^{\text{depth}(v)} X_{u,v}$$

As a sanity check, it can easily be verified that $E(Y) = q(1 - (1 - p)^h)$ by computing $E(Y)$ using

these indicators:

$$\begin{aligned}
 E(Y) &= 2^{-h} \sum_{d=1}^h 2^d (1-p)^{d-1} pq * 2^{h-d} \\
 &= pq \sum_{d=1}^h (1-p)^{d-1} \\
 &= q(1 - (1-p)^h)
 \end{aligned}$$

Thus, we can compute $E(XY)$ similar to how we computed $E(X^2)$ for Variance.

$$\begin{aligned}
 E(XY) &= 2^{-h} E\left(\sum_{(u,v) \in T} X_{u,v}\right) \left(\sum_{(u,v) \in T} 2^{\text{depth}(v)} X_{u,v}\right) \\
 &= 2^{-h} \left(2 \sum_{(u,v) \neq (u',v')} 2^{\text{depth}(v)} E(X_{u,v} X_{u',v'}) + \sum_{(u,v) \in T} 2^{\text{depth}(v)} E(X_{u,v}^2) \right) \\
 &= 2 * 2^{-h} \sum_{d=0}^{h-1} 2^d \sum_{k=0}^{h-1} \sum_{l=0}^{h-1} 2^{l+k} p^2 q^2 (1-p)^{d+l+k} * 2^{h-d-l-1} + E(Y) \\
 &= p^2 q^2 \sum_{d=0}^{h-1} \sum_{k=1}^{h-d-1} \sum_{l=0}^{h-d-1} (1-p)^d (2-2p)^k (1-p)^l + E(Y) \tag{Eq. 4} \\
 &= \frac{pq^2}{1-2p} \sum_{d=0}^{h-1} (2-2p)^{h-d} - 1) (1 - (1-p)^{h-d}) (1-p)^d + E(Y) \\
 &= \frac{pq^2}{1-2p} \left(2(2-2p)^h (1-2^{-h}) - \frac{(2-2p)(1-p)^h ((2-2p)^h - 1)}{1-2p} \right. \\
 &\quad \left. - \frac{1 - (1-p)^h}{p} + h(1-p)^h \right) + E(Y)
 \end{aligned}$$

Assuming that is $p < 1 - 1/\sqrt{2} \approx 0.29$ (based on our estimation of Cas9-cutting rates, this seems

to be a biologically relevant probability), we have:

$$\begin{aligned}\lim_{h \rightarrow \infty} \text{CoV}(X, Y) &= \left(2 - \frac{2 - 2p}{1 - 2p}\right) \frac{pq^2(2(1 - p)^2)^h}{1 - 2p} \\ &= -\infty\end{aligned}$$

since $2 < (2 - 2p)/(1 - 2p)$ when $p < 0.5$.

$\text{Var}(Y)$ can be computed using the same indicators:

$$\begin{aligned}\text{Var}(Y) &= 2 \sum_{i,j} E(Y_i Y_j) + \sum_i E(Y_i^2) - E(Y)^2 \\ \sum_{i,j} E(Y_i Y_j) &= 2^{-2h} \sum_{d=0}^{h-1} 2^d (1-p)^d \left(\sum_{k=0}^{h-d-1} 2^k (1-p)^k pq * 2^{h-d-k-1} \right)^2 \\ &= \frac{q^2}{4} \sum_{d=0}^{h-1} \left(\frac{1-p}{2} \right)^d \left(\frac{1 - (1-p)^{h-d}}{p} \right)^2 \\ &= \frac{q^2}{4} \sum_{d=0}^{h-1} \left(\frac{1-p}{2} \right)^d - \frac{2(1-p)^h}{2^d} + \frac{(1-p)^{2h}}{(2-2p)^d} \\ &= \frac{q^2}{4} \left(\frac{2(1 - (\frac{1-p}{2})^h)}{1+p} - 4(1-p)^h(1-2^{-h}) + \frac{(2-2p)(1-p)^{2h}(1 - (\frac{1}{2-2p})^h)}{1-2p} \right) \\ \sum_i E(Y_i^2) &= 2^{-2h} \sum_{d=1}^h 2^d (1-p)^{d-1} pq * 2^{2(h-d)} \\ &= \frac{pq}{2} \sum_{d=0}^{h-1} \left(\frac{1-p}{2} \right)^d \\ &= \frac{pq(1 - (\frac{1-p}{2})^h)}{1+p}\end{aligned} \tag{Eq. 5}$$

Note that if $p < 0.5$, every term in $Var(Y)$ converges to a constant as $h \rightarrow \infty$. Thus, if $(1 - p)^2 > 0.5$, then as the depth increases, X and Y become exponentially more negatively correlated. This means that for biologically relevant values of p , the frequency of a mutation in the samples is negatively correlated with number of times the mutation occurred, thus justifying the rationale of splitting the sample on more frequently occurring mutations.

Simulation For Tracking the Evolution of a Particular Mutation

To more efficiently simulate the number of occurrences of a particular mutation, we define $\{N_1, N_2, \dots, N_h\}$ as a Markov chain, where N_t is the number of unmutated cells at generation t , and $N_1 = 1$. Let $A_t \sim Bin(2N_t, p)$ be the number of cells that mutates at generation t , and $B_t \sim Bin(A_t, a)$ be the number of mutated cell that took on the particular state in question. The Markov chain evolves as $N_{t+1} = 2N_t - A_t$. Note that we assume, in this model, that mutation can only occur after cell division. Thus we have $X = \sum_{t=1}^h B_t$ and $Y = \sum_{t=1}^h 2^{t-h} B_t$.

Assessing the Precision of Greedy Splits.

To assess the precision of greedy splits, we first simulated 100 true phylogenies of 400 cells (without dropout) for all pairs of parameters in $num_states = \{2, 10, 40\}$ and $p_{cut} = \{0.025, 0.1, 0.4\}$. For each network, we assessed the precision of the greedy split as follows:

1. We used the criteria $i, j = \arg \max_{i,j} n_{i,j}$ to select the character χ_i and state j to split on (as Cassiopeia-Greedy would do). This group of cells that have a mutation j in character χ_i

is called G .

2. For define the a set of n subsets corresponding to cells that inherited the (character, state) pair (i, j) independently using the true phylogenies, and call this set $S = (s_1, s_2, \dots, s_n)$ (this corresponds to there being n parallel evolution events for the (character, state) pair (i, j)).
3. We presume that the largest group of cells in S is the "true positive" set (let this be defined as $s' = \arg \max_s |s_i|$). We then define the precision P as the proportion of true positives in the set G – i.e. $P = \frac{|s'|}{|G|}$.

Statistics for IVLT Analysis

Meta Purity Statistic

To calculate the agreement between clades (i.e. the leaves below a certain internal node of the tree) and some meta-value, such as the experimental plate from which a sample came from, we can employ a Chi-Squared test. Specifically, we can compute the following statistic: considering some M clades at an arbitrary depth d , we find the count of meta values associated with each leaf in each clade, resulting in a vector of values m_i comprised of these meta-counts for each clade i . We can form a contingency table summarizing these results, T , where each internal value is exactly $m_{i,j}$ - the counts of the meta item j in clade i . A Chi-Squared test statistic can be computed from this table.

To compare across different trees solved with different methods, we report the Chi-Squared Test Statistic as a function of the number of clades, or degrees of freedom of the test.

Mean Majority Vote Statistic

The Mean Majority Vote statistic seeks to quantify how coherent each clade is with respect to its majority vote sample at a given depth. For a given clade with leaves L_i where $|L_i| = n$, where every leaf $l_{i,j}$ corresponds to cell j in clade i has some meta label m_j , the majority vote of the clade is $v = \operatorname{argmax}_{m' \in M} \sum_{j \in n} \delta(j, m')$. Here M is the full set of possible meta values and $\delta(m_j, m')$ is an indicator function evaluating to 1 iff $m_j = m'$. The membership of this clade is then simply $\frac{\sum_{j \in n} \delta(m_j, v)}{n}$. Then, the mean membership is the mean of these membership statistics for all clades at a certain depth (i.e. if the tree were cut at a depth of d , the clades considered here are all the internal nodes at depth d from the root). By definition, this value ranges from $\frac{1}{|M|}$ to 1.0.

As above, to compare across different trees solved with various methods, we report this mean membership statistic as a function of the number of clades.

Triplets Correct Statistic

To compare the similarity of simulated trees to reconstructed trees, we take an approach which compares the sub-trees formed between triplets of the terminal states across the two trees. To do this, we sample $\sim 10,000$ triplets from our simulated tree and compare the relative orderings of each triplet to the reconstructed tree. We say a triplet is "correct" if the orderings of the three terminal states are conserved across both trees. This approach is different from other tree comparison statistics, such as Robinson-Foulds [211], which measures the number of edges that are similar between two trees.

To mitigate the effect of disproportionately sampling triplets relatively close to the root of the tree, we calculate the percentage of triplets correct across each depth within the tree independently (depth measured by the distance from the root to the Latest Common Ancestor (LCA) of the triplet). We then take the **average** of the percentage triplets correct across all depths. To further reduce the bias towards the few triplets that are sampled at levels of the tree with very few cells (i.e. few possible triplets), we modify this statistic to only take into account depths where there are at least 20 cells. We report these statistics without this depth threshold in **Figure 2.14**.

Allelic and Phylogenetic Distances

For the analysis in **Figure 2.4**, we define two metrics to capture cell-to-cell similarity: a normalized allelic distance and normalized phylogenetic distance. The normalized allelic distance is calculated as follows: for all target sites $\chi_m \in \{\chi_1, \dots, \chi_M\}$ in a pair of cells c_i and c_j :

1. if state in χ_m is the same in c_i and c_j , continue
2. else if state in χ_m is 0 or missing in either c_i or c_j increment the allelic distance by 1
3. else increment the allelic distance by 2

Finally, the allelic distance for a pair of cells is normalized by $2 * M$, where M is the number of target sites.

The phylogenetic distance is defined as simply the number of mutations separating the two cells c_i and c_j as implied by the tree (i.e. the number of mutations along the branches for the shortest

path separating c_i and c_j). The normalized phylogenetic distance is this distance, divided by the diameter (defined as the maximum phylogenetic distance between all pairs of cells) of the tree.

Bootstrapping Analysis

Bootstrapping was done using a custom function for sampling M target sites (i.e. characters) from an $N \times M$ character matrix with replacement and reconstructing trees from these bootstrapped samples. After performing tree inference, we collapsed "singles" using the `collapse.singles` function in the R package "ape". For the purposes of our robustness analysis, we sampled $B = 100$ trees from $N = 10$ simulated trees and used the Transfer Bootstrap Expectation (TBE) [159] statistic for assessing branch support for each clade as implemented in Booster (available at <https://github.com/evolbioinfo/booster/>).

Application of Camin-Sokal

We applied Camin-Sokal using the "mix" program in PHYLIP [68] as done for reconstructions for McKenna et al [178] and Raj et al [206]. To use "mix" we first factorized the characters into binary ones (thus ending up with $\sum_i s_i$ binary characters total, where s_i is the number of states that character i presented). Then, we one-hot encoded the states into this binary representation where every position in the binary string represented a unique state at that character. We thus encoded every cell as having a 1 in the position of each binary factorization corresponding to the state observed at that character. If the cell was missing a value for character i , the binary factorization of the character

was a series of '?' values (which represent missing values in PHYLIP "mix") of length s_i . Before performing tree inference, we weighted every character based on the frequency of non-zero (and non-missing values) observed in the character matrix. After PHYLIP "mix" found a series of candidate trees, we applied PHYLIP "consense" to calculate a consensus tree to then use downstream.

Application of Neighbor-Joining

We used Biopython's Neighbor-Joining procedure to perform all neighbor joining in this manuscript. We began similarly to the Camin-Sokal workflow, first factorizing all of the characters into a binary representation. Then, we applied the Neighbor-Joining procedure using the "identity" option as our similarity map.

Application of Cassiopeia

Reconstruction of simulated data

We used Cassiopeia-ILP with a maximum neighborhood size of 10,000 and time to converge of 12,600s. Cassiopeia-Hybrid used a greedy cutoff of 200, a maximum neighborhood size of 6000 and 5000s to converge. Cassiopeia-Greedy required no additional hyperparameters. Simulations with priors applied the exact prior probabilities used to generate the simulated trees.

Reconstruction of *in vitro* clones

. For both Cassiopeia-Hybrid with and without priors, we used a cutoff of 200 cells and each instance of Cassiopeia-ILP was allowed 12,600s to converge on a maximum neighborhood size of 10,000. Cassiopeia-ILP was applied with a maximum neighborhood size of 10,000 and a time to converge of 12,600s.

Simulation of Target Site Sequences for Alignment Benchmarking

To determine an optimal alignment strategy and parameters for our target site sequence processing pipeline, we simulated sequences and performed a grid search using Emboss's Water algorithm (a local alignment strategy). We simulated 5,000 sequences. For each sequence, we began with the reference sequence and subjected it to multiple rounds of mutagenesis determined by a Poisson distribution with $\lambda = 3$, and a maximum of 5 cuts. During each "cutting" event, we determined the outcomes as follows:

1. Determine the number of Cas9 proteins localizing to the target site in this iteration, where

$$n_{cas9} \sim \min(3, Pois(\lambda = 0.4)).$$

2. Determine the site(s) to be cut by choosing available sites randomly, where the probability of

being chosen is $p = \frac{1}{n_{uncut}}$ and n_{uncut} is the number of sites uncut on that sequence.

3. If $n_{cas9} = 1$, we determined the type of the indel by drawing from a Bernoulli distribution with a probability of success of 0.75 (in our case, a "success" meant a deletion and a "failure"

meant an insertion). We then determined by drawing from a Negative Binomial Distribution as so: $s \sim \min(30, \max(1, NB(0.5, 0.1)))$. In the case of an insertion, we added random nucleotides of size s to the cut site, else we removed s nucleotides.

4. In the case of $n_{cas9} \geq 2$, we performed a resection event where all nucleotides between the two cut sites selected were removed.
5. After a cut event, we appended the result of the Cas9 interaction to a corresponding CIGAR string

Our Water simulations were exactly 300bp, possibly extending past the Poly-A signal, as would be the case reading off a Nova-seq sequencer.

Upon simulating our ground truth dataset, we performed our grid search by constructing alignments with Water with a combination of gap open and gap extension penalties. We varied the gap open penalties between 5 and 50 and gap extension penalties between 0.02 and 2.02.

To score resulting alignments, we compared the resulting CIGAR string to our ground truth CIGAR string for each simulated sequence. To do so, we first split each cigar string into "chunks", corresponding to the individual deletions or insertions called. For example, for some CIGAR string $40M2I3D10M$, the chunks would be $40M$, $2I$, $3D$, and $10M$. Then, beginning with a max score of 1, we first deducted the difference between the number of chunks in the ground truth and the alignment. Then, in the case where the number of chunks were equal between ground truth and alignment, we deducted the percent nucleotides that differed between CIGARs. For example, if the ground truth was $100M$ and the alignment gave $95M$, the penalty would be 0.05.

To find the optimal set of parameters, we selected a parameter pair that not only scored very well, but also located in the parameter space where small perturbations in gap open and gap extension had little effect.

Simulation of Lineages for Algorithm Benchmarking

We simulated lineages using the following parameters:

1. The number of characters to consider, C
2. The number of states per character, S
3. The dropout per characters, $d_c \forall c \in C$
4. The depth of the tree (i.e. the number of binary cell division), D
5. The probability that a site can be mutated, p . This is a general probability of cutting
6. The rate at which to subsample the data at the end of the experiment, M

To simulate the tree, we begin by first generating the probability of each character mutating to a state, here represented as $p_c(0, s), \forall s \in S$. In order to do this, we fit a spline function to the inferred prior probabilities from the lineage tracing experiment. (refer to the section entitled "Determining Prior Probabilities of Indel Formation" for information on how we infer prior probabilities). We then draw S values from this interpolated distribution. We then normalize these mutation rates to sum to p , therefore allowing in general a p probability of mutating a character and $1 - p$ probability of

remaining uncut. In the case of the “State Distribution” simulations (**Figure 2.19**), we say that p_c is distributed as:

$$p_c = \theta * Unif(0, 1) + (1 - \theta) * F'(x)$$

where $F'(x)$ is the interpolated empirical distribution and θ is the mixture component.

Then, we simulate D cell divisions, where each cell division consists of allowing a mutation to take place at each character with probability p . In the case a mutation takes place, we choose a state to mutate to according to their respective probabilities. Importantly, once a character has been mutated in a cell, that character cannot mutate again.

At the end of the experiment, we sample M percent of the cells resulting in $2^D * M$ cells in the final lineage.

We find that this method for simulating lineages (in particular the method for generating a set of priors on how likely a given state is to form) is able to closely recapitulate observed lineages (**Figure 2.12**).

Metrics for Comparing Simulations to Empirical Data

We used three metrics of complexity to compare simulated clones to real clones:

- *Minimum Compatibility Distance*: For every pair of character, we define the Minimum Compatibility Distance as the minimum number of cells to be removed to obtain compatibility (Def. 1).

- *Number of Observable States per Cell*: The number of non-zero or non-missing values for each cell, across all characters (i.e. the amount of data that can be used for a reconstruction, per cell).
- *Number of Observable States per Character*: The number of non-zero or non-missing values across for each character, across all cells.

Parallel Evolution Simulations for Greedy Benchmarking

As shown above, our greedy approach should accurately reconstruct a lineage if a perfect phylogeny exists. In order to better quantify how much our greedy algorithm's performance is affected by parallel mutations, we decided to simulate "near perfect phylogenies", whereby we first began by simulating a perfect phylogeny, and afterwards introduced double mutated characters.

Specifically, we begin by simulating perfect phylogenies with $40 - k$ characters. We then fix a depth, d , and sample a node from said depth. We choose two grandchildren randomly from this node (one from each child) and introduce the same mutation on each of the edges from each child to grandchild, thereby violating the perfect phylogeny. We repeat this process k times. This thus creates an analysis, as presented in **Figure 2.11**, whereby accuracy can be evaluated as a function of both depth of parallel evolution, d , and the number of events that occurred, k .

Simulation of “Base Editor” Technologies

We used the simulation framework described above to simulate base-editor technologies. To explore the trade off between the number of states and the number of characters, we simulated trees with 40, 50, 80, and 100 characters while maintaining the product of characters and states equal at 400 (thus we had trees of 10, 8, 5, and 4 states per character, respectively). The dropout per character was set to 10%, the mutation rate per character was set to 1.04% (a previously observed mutation rate [107]), and 400 cells were sampled from a tree of depth 10. For each character/state regime, we generated 10 trees for assessing the consistency of results. We use a negative binomial model ($\sim NB(5, 0.5)$) as the editing outcome distribution (i.e. state distribution).

Simulation of “Phased Recorder” Technologies

To simulate the phased recorder, we used 5 different experiments varying mutation rates across 50 characters and 10 states per character. In each experiment, we chose a mutation rate for each character from one of 10 regimes, each differing in their relationship to the base mutation rate p_0 . To systematically implement this, mutation rate for χ_i is described as such:

$$m_i = p_0 * (1 + e_j * \lfloor \frac{i}{5} \rfloor)$$

where $p_0 = 0.025$ and e_j is a experiment scalar in $\mathbf{e} = \{0, 0.05, 0.1, 0.25, 0.5\}$. This means that for characters 1 – 5, $m_i = p_0$, for characters 6 – 10, $m_i = e_j p_0$, for characters 11 – 15, $m_i = 2e_j p_0$, etc. To summarize each experiment, we provide the ratio between the maximum and minimum

mutation rates, which is by definition $1 + 10r_j$ (because we had 50 characters). We compare two models of indel formation rates - the first being a negative binomial model ($\sim NB(5, 0.5)$), and the second being the spline distribution fit from empirical data.

We simulated 10 trees per regime and reconstructed trees with Cassiopeia with and without priors.

Reconstructions of GESTALT Datasets

We downloaded data corresponding to the original GESTALT study [178] and the more recent scGESTALT study from <https://datadryad.org/resource/doi:10.5061/dryad.478t9> and GSE105010, respectively. We created character matrices for input into Cassiopeia by creating pivot tables relating each cell the observed indel observed at each one of the 10 tandem sites on the GESTALT recorder. We then reconstructed trees from these character matrices using one of five algorithms: Camin-Sokal (used in the original studies), Neighbor-Joining, Cassiopeia's greedy method, Cassiopeia's Steiner Tree method, and Cassiopeia's hybrid method.

For each reconstruction, we record the parsimony of the tree, corresponding to the number of mutations that are inferred along the reconstructed tree. We display these findings in **Figure 2.6a**, where we have Z-normalized the parsimonies across the methods for each dataset to enable easier visualization of relative performances.

Visualization of Trees

To visualize trees we use the iTOL interface [\[45\]](#). Colors in the heatmap denote a unique mutation, gray denotes an uncut site, and white denotes dropout.

2.6 Supplementary Figures

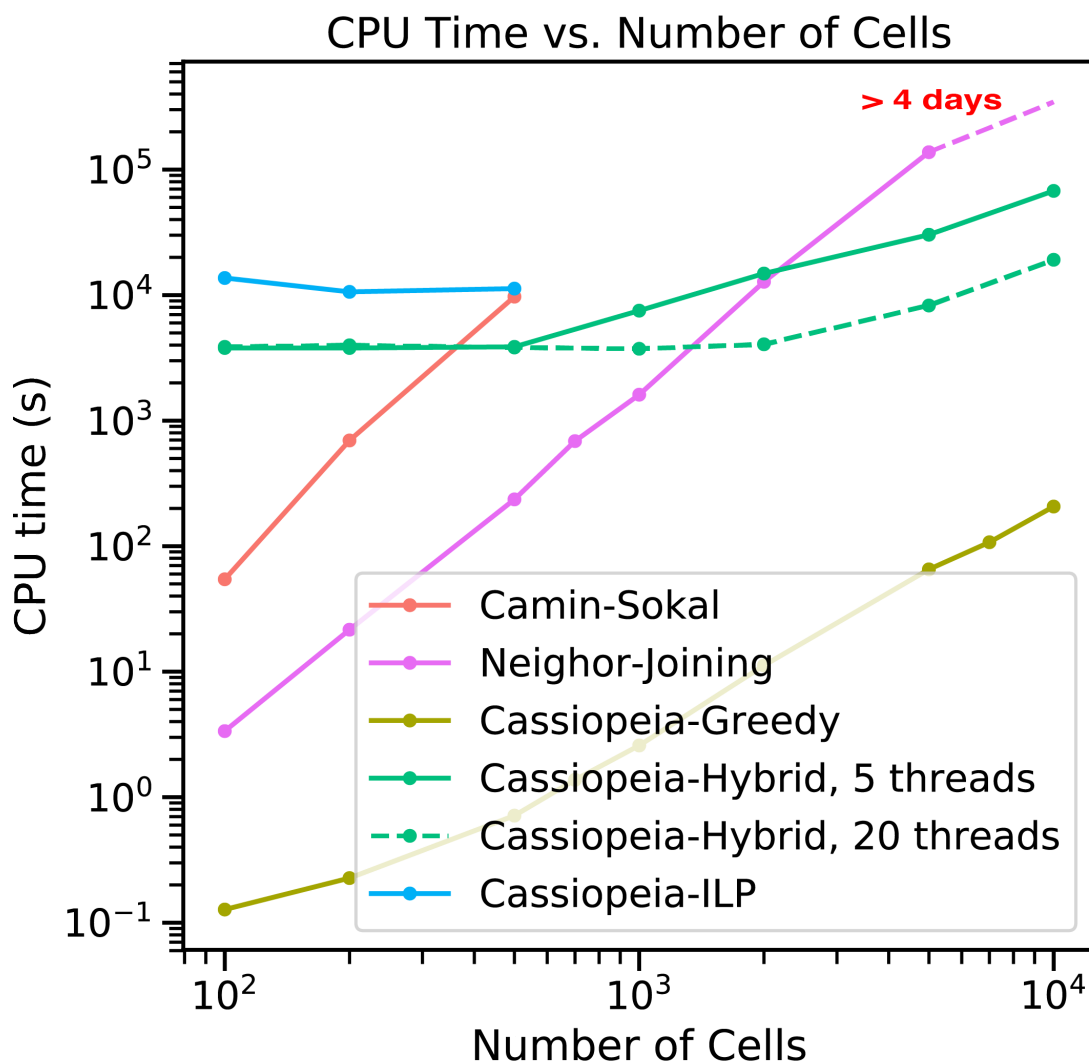


Figure 2.7: **Time complexity of lineage reconstruction approaches.** Time complexity, as measured in seconds, of each algorithm tested in this manuscript is compared using simulated datasets ranging from 100 cells to 10,000 cells. Default settings for the simulations were used (0.025 mutation rate, 40 characters, 10 states, and 0.18 median dropout rate). Cassiopeia was tested using default parameters of a maximum neighborhood size of 3000, time to converge of one hour, and a greedy cutoff of 200 cells. Cassiopeia was tested using 5 threads and 20 threads, illustrating the advantage of parallelizing the reconstruction algorithm. ILP, which was only run until 500 cells due to the infeasibility of running on larger datasets, was allowed 10000s to converge on a maximum neighborhood size of 20,000 (the default settings). Neighbor-Joining could not reconstruct a tree for 10,000 cells within 4 days when the reconstruction was terminated.

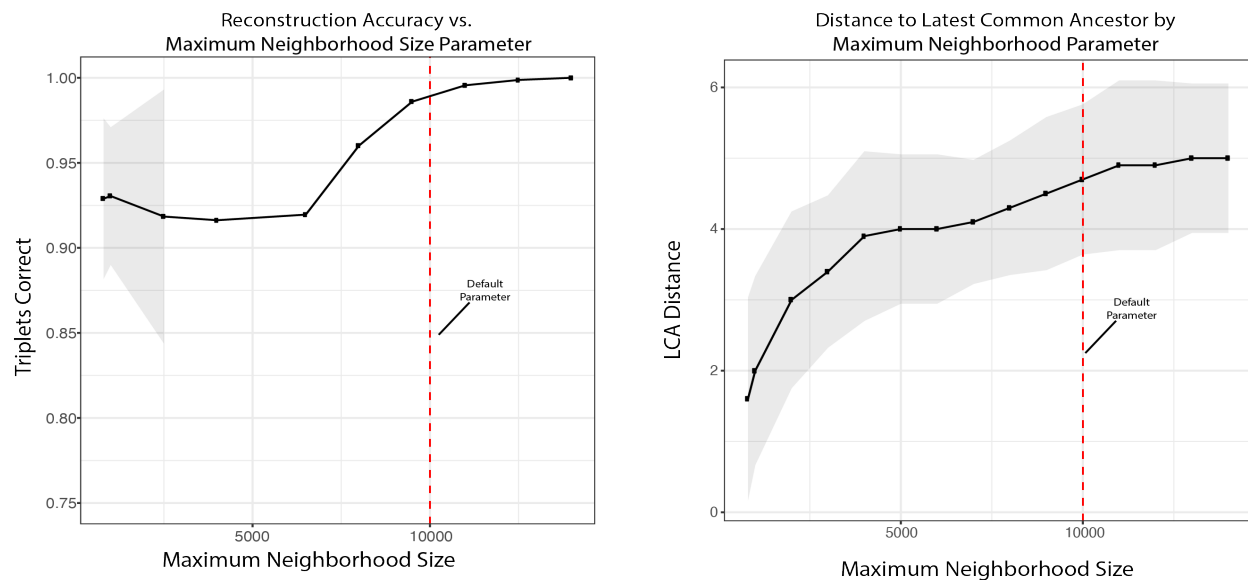


Figure 2.8: **Evaluation of the stability of the maximum neighborhood size parameter.** The maximum neighborhood size is a central parameter provided by the user when inferring the potential graph necessary as input to the Steiner-Tree solver (see methods). Here, we benchmark the stability of solutions with respect to several maximum neighborhood sizes using 10 trees with default parameters (40 characters, 40 states, 2.5% per-character mutation rate, depth of 11, and an average dropout rate of 17% per character). We quantify both the reconstruction accuracy with respect to the reconstructions found with the largest maximum neighborhood size (14,000 nodes) which displays a saturation at around 9,000 nodes. To provide intuition for the accuracy of the potential graph (represented as the maximum distance to the 'latest common ancestor' (LCA) which is dynamically solved for, given a maximum neighborhood size) we display the LCA allowed for each maximum neighborhood size parameter. In both figures, we display lines connecting the mean values; shaded regions are the standard deviation of the measurements across the 10 replicates.

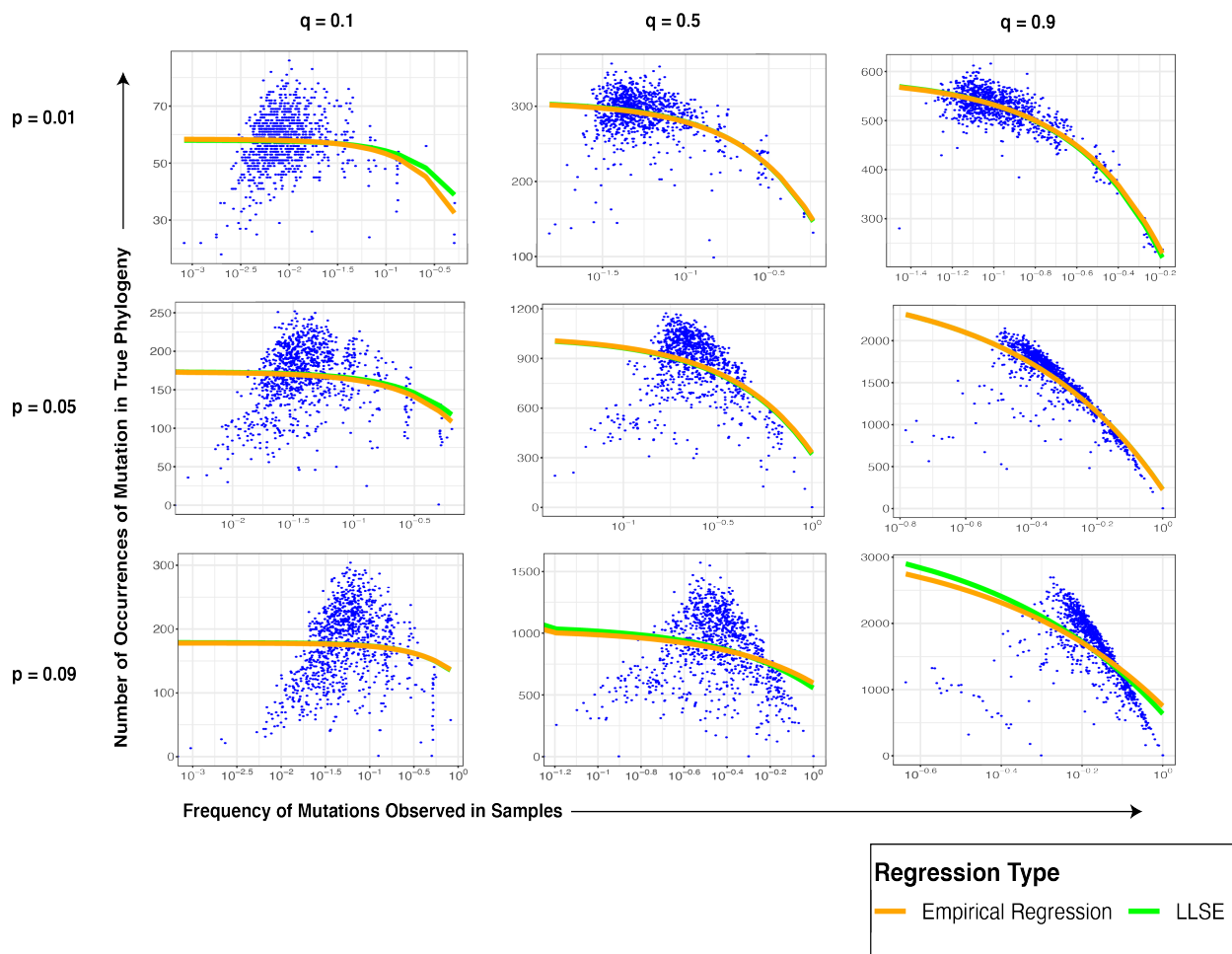


Figure 2.9: **Observed Frequency of Mutations is Measure of True Mutation Count.** The true number of occurrences of a mutation is estimated well by the observed frequency at leaves. We use a Linear Least Squares Estimate to quantify the relationship between the expected number of times a mutation occurred given the observed frequency at the leaves (Eq. 3). Using various rates for character and indel mutation rates (p and q , respectively) we show that this relationship is negative (i.e. greater observed frequencies tend to correspond to mutations that occurred few times near the top of the phylogeny) for a range of biologically-relevant values.

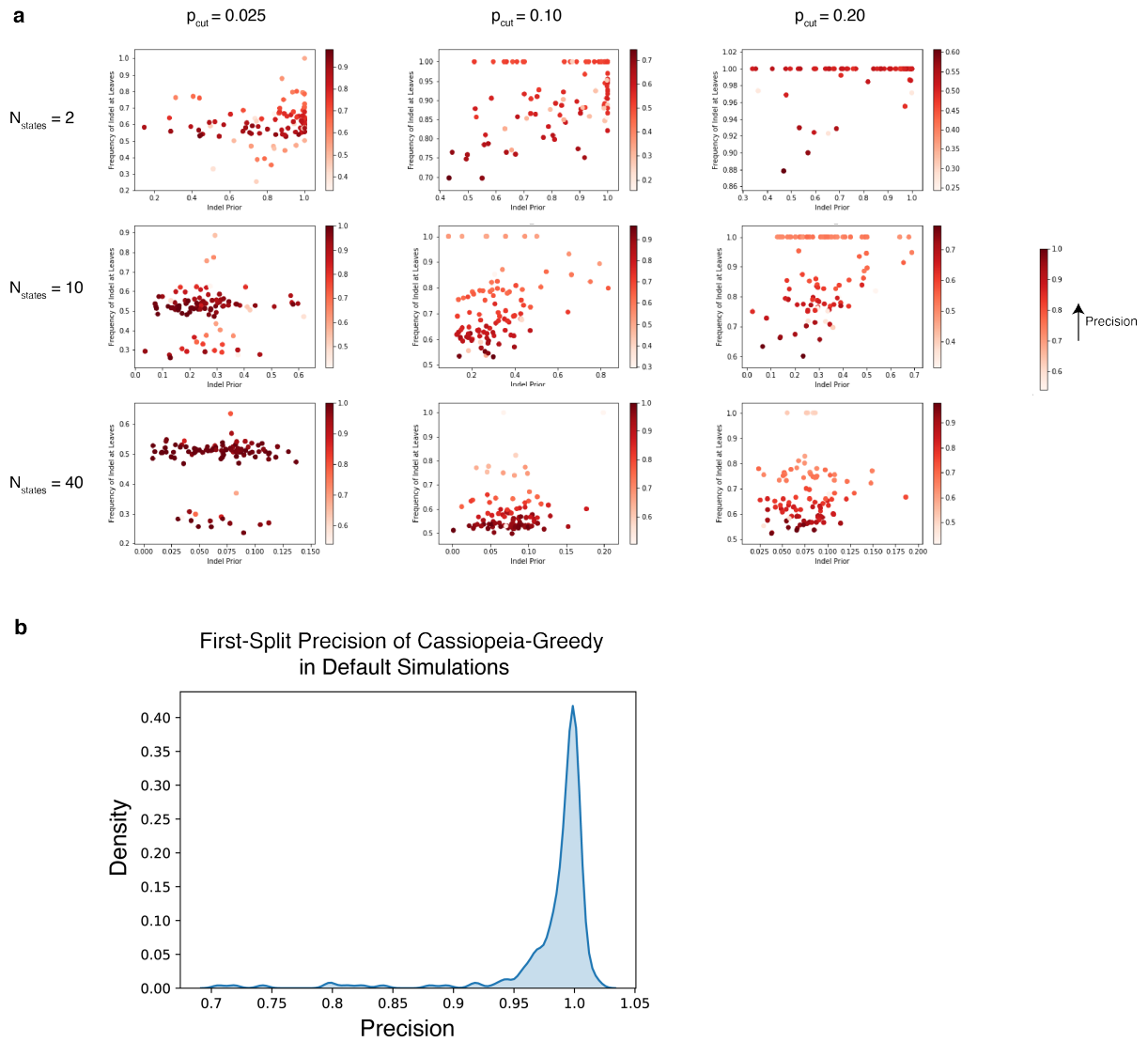


Figure 2.10: **Precision of Cassiopeia-Greedy First Split.** (a) The precision of greedy splits of 400 cells was measured with varying mutation rates and states per character, without dropout. For each pair of parameters (number of states and mutation rate), we measure precision as a function of the conditional probability of the selected (character, state) pair and the frequency of that mutation observed in the 400 cells. (The conditional probability for state j , $q(j)$ is defined as $Pr(\mathcal{X} \rightarrow j | \mathcal{X} \text{ mutates})$). Precision was defined as the proportion of true positives in the greedy split (see Methods). Each point indicates a replicate (100 per plot) and the heat represents the precision. (b) The density histogram (smoothed using a kernel density estimation procedure) of all first-split precision statistics from Cassiopeia-Greedy on default simulations (i.e. 40 characters, 40 states, 2.5% mutation rate, 11 generations, 400 cells, and 18% dropout rate). We measured a median precision of 0.99 across all default simulations.

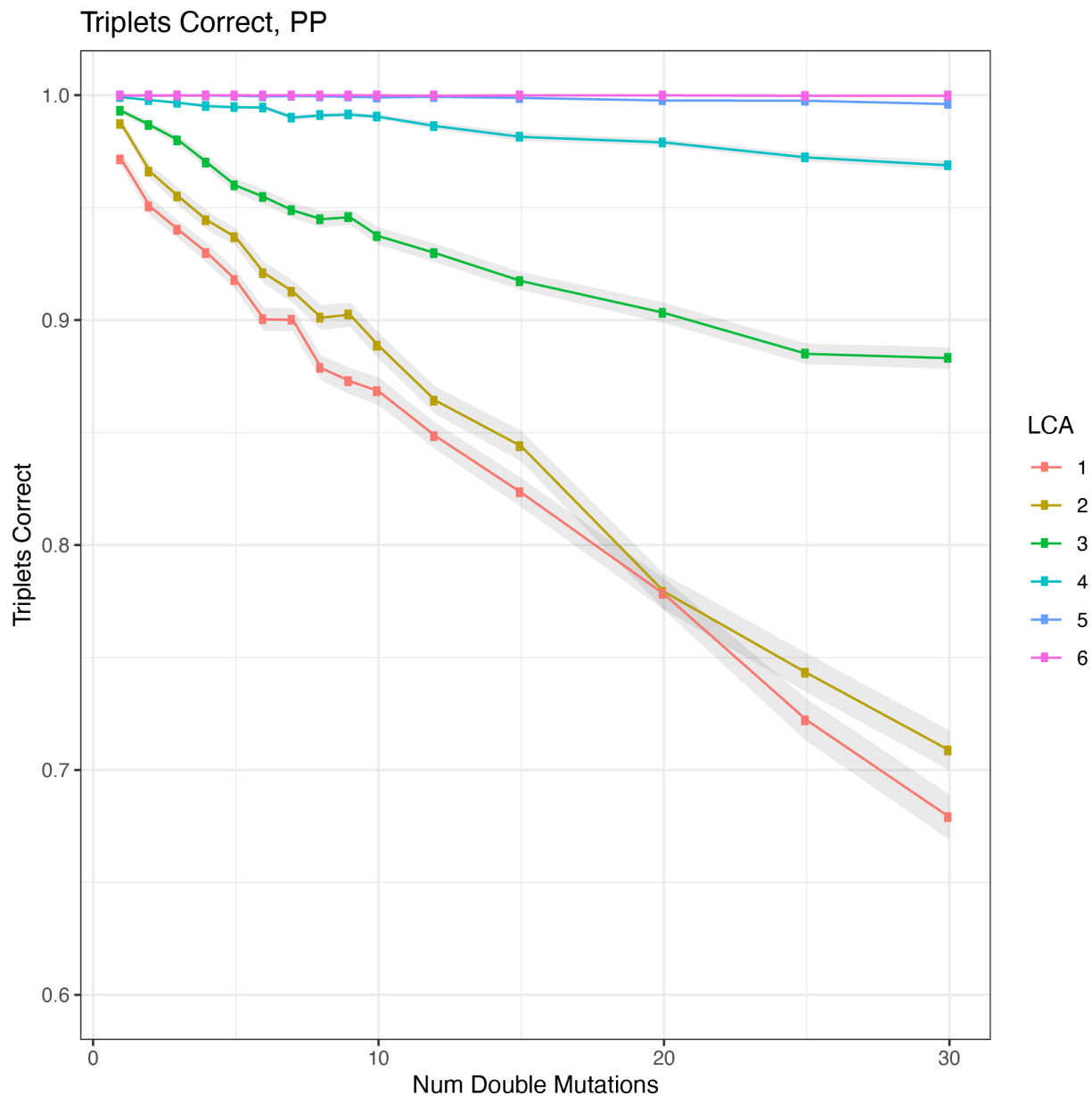


Figure 2.11: **Benchmarking of parallel evolution on the greedy heuristic.** The greedy heuristic, inspired by algorithms to solve the case of perfect phylogeny (see methods), is impacted by two factors: (1) the number of parallel evolution events (i.e. the same mutation occurs more than once in the experiment) and (2) the depth from the root these mutations occur at. Here, each line represents a series of experiments increasing the number of 'double mutations' (i.e. the simplest case of parallel evolution where a mutation occurs exactly twice) where the 'latest common ancestor' (LCA) is a set depth from the root.

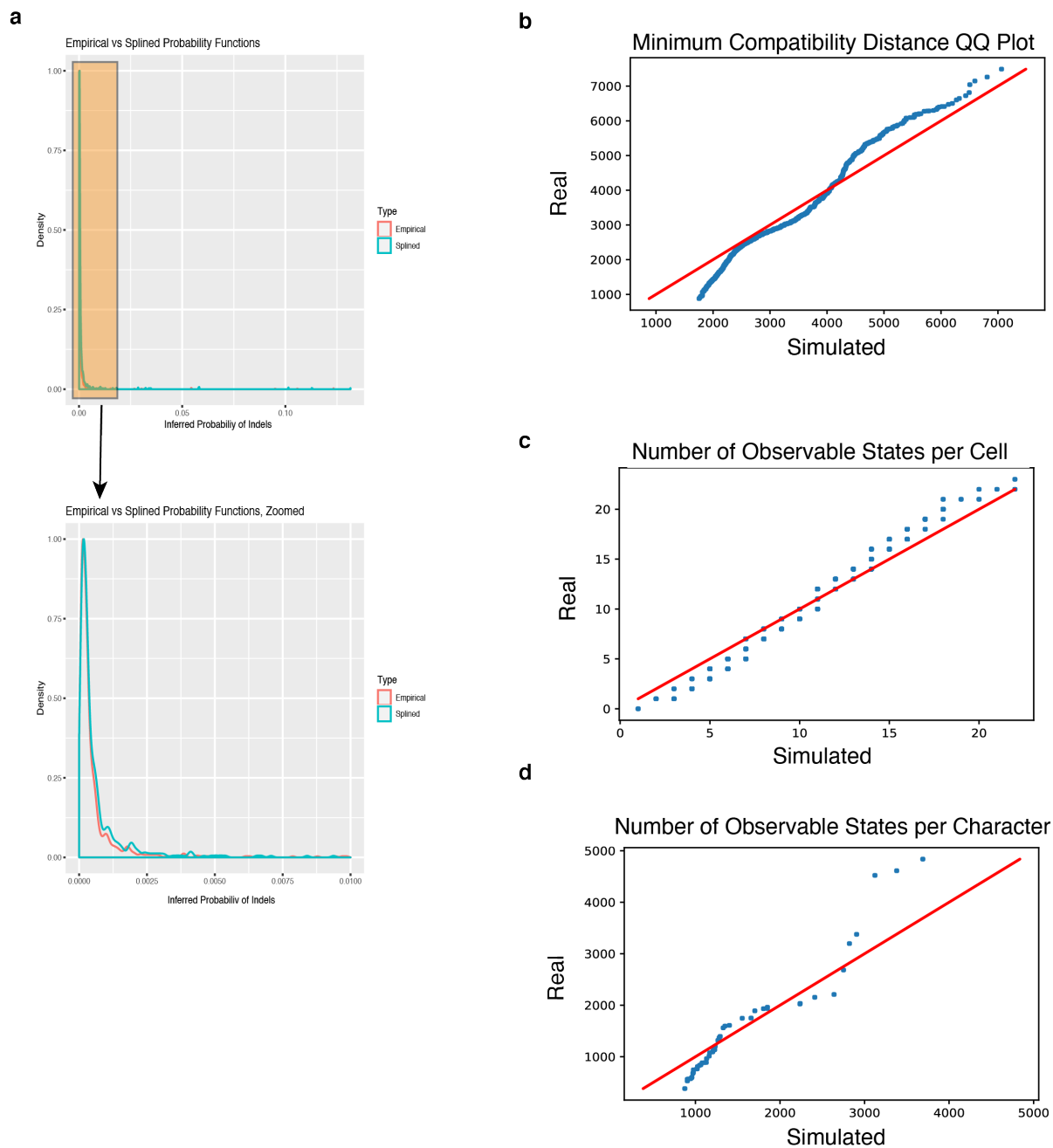


Figure 2.12: Determination of mutation rates used in simulation. We use an interpolation of the empirical indel distribution as input for the conditional probability of a state arising given a mutation. (a) A comparison of the empirical and 'splined' indel distributions; a zoomed in version is provided for comparison at low probabilities. (b-c) A comparison of three metrics between an observed clone (clone 3) and a simulated clone using inferred parameters. We used the number of character, states, per-character mutation rate, and dropout probabilities inferred from the empirical data; the indel formation rates were calculated using a polynomial spline function. (b) measures the 'minimum compatibility distance' for all pair-wise character combinations (see methods). (c) compares the number of observable states per cell. (d) compares the number of observable states per character.

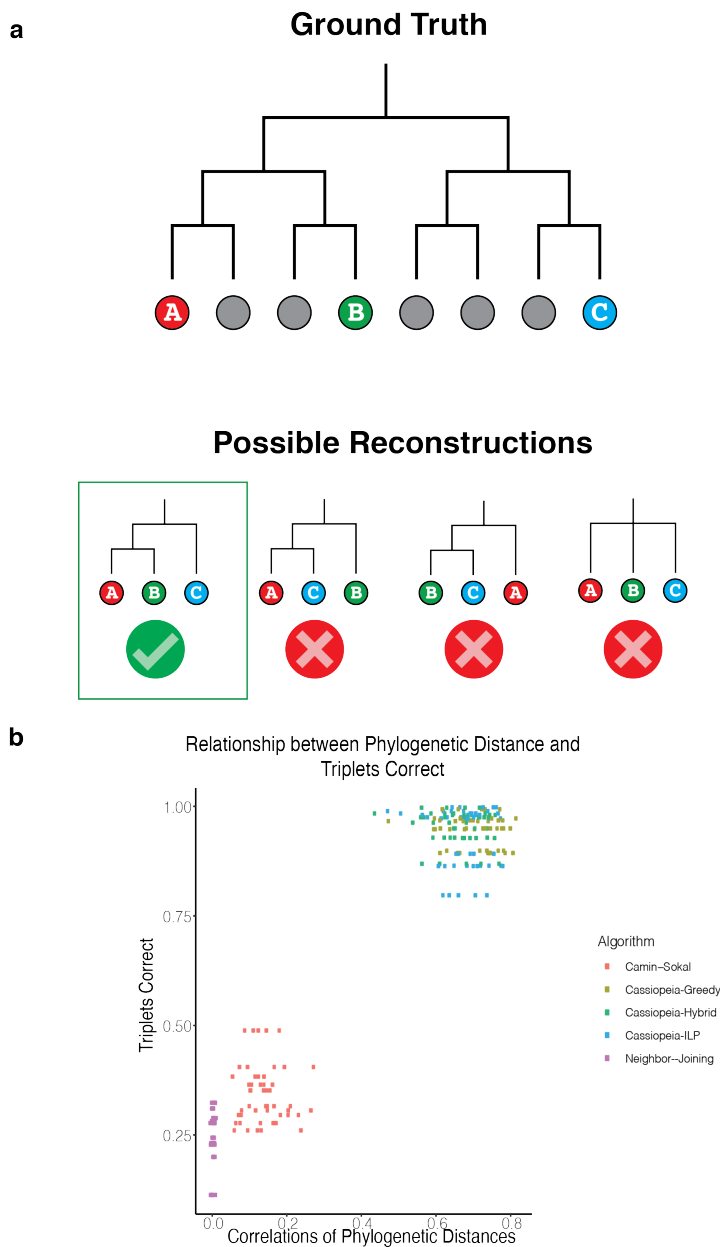


Figure 2.13: **Triplets Correct Statistic.** (a) Schematic for the Triplets Correct statistic, the combinatorial metric used to compare between trees. In this metric, we compare the relative orderings of three leaves between two trees (e.g. the “Ground Truth” and a reconstruction). There are four possible ways that a triplet could be ordered here, based on the relationship between each leaf and the Latest Common Ancestor (LCA) of the triplet. The statistic tallies the number of correct triplets and reports this value weighted by the depth of the LCA from the root. Importantly, this statistic is designed to avoid concerns of inappropriately weighting early splits as these might dominate the statistic. Specifically, the triplets are stratified in accordance to the depth of the LCA and the triplets correct is reported as an average across all LCA depths. This way, LCAs near the root will not dominate the score. (b) A comparison between the triplets correct statistic and the phylogenetic distance correlation (defined as the correlation of node-node distances between a simulated and reconstructed tree; see Methods) where we observe a Pearson correlation of 0.96.

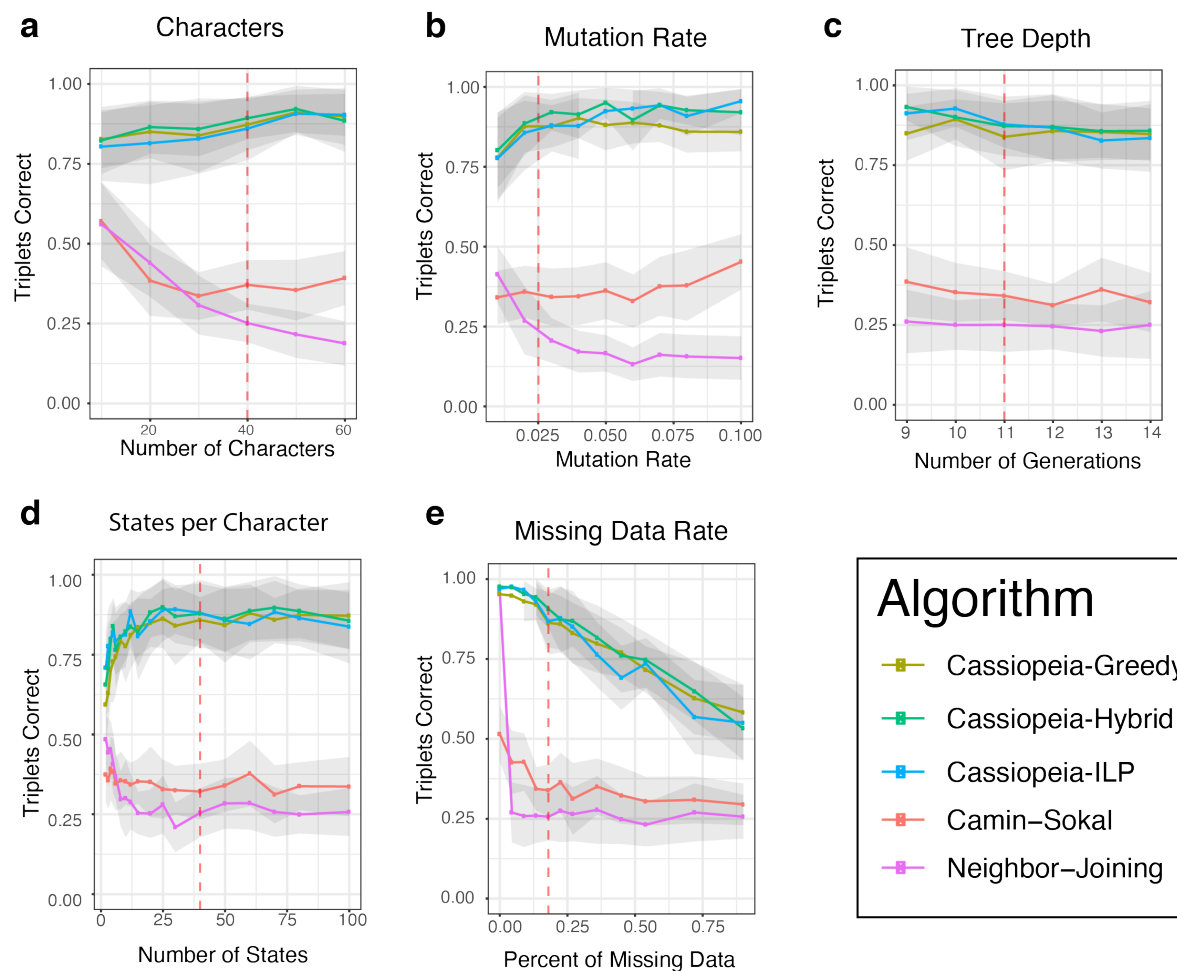


Figure 2.14: **Unthresholded Triplets Correct.** The Triplets Correct statistic reported for synthetic benchmarks presented in **Figure 2.2** without removing triplets whose LCA-depth was sampled deeply enough (by default, a given triplet at depth D is only considered if a sufficient number of triplets at depth D is observed). Here, the effective threshold is 0.

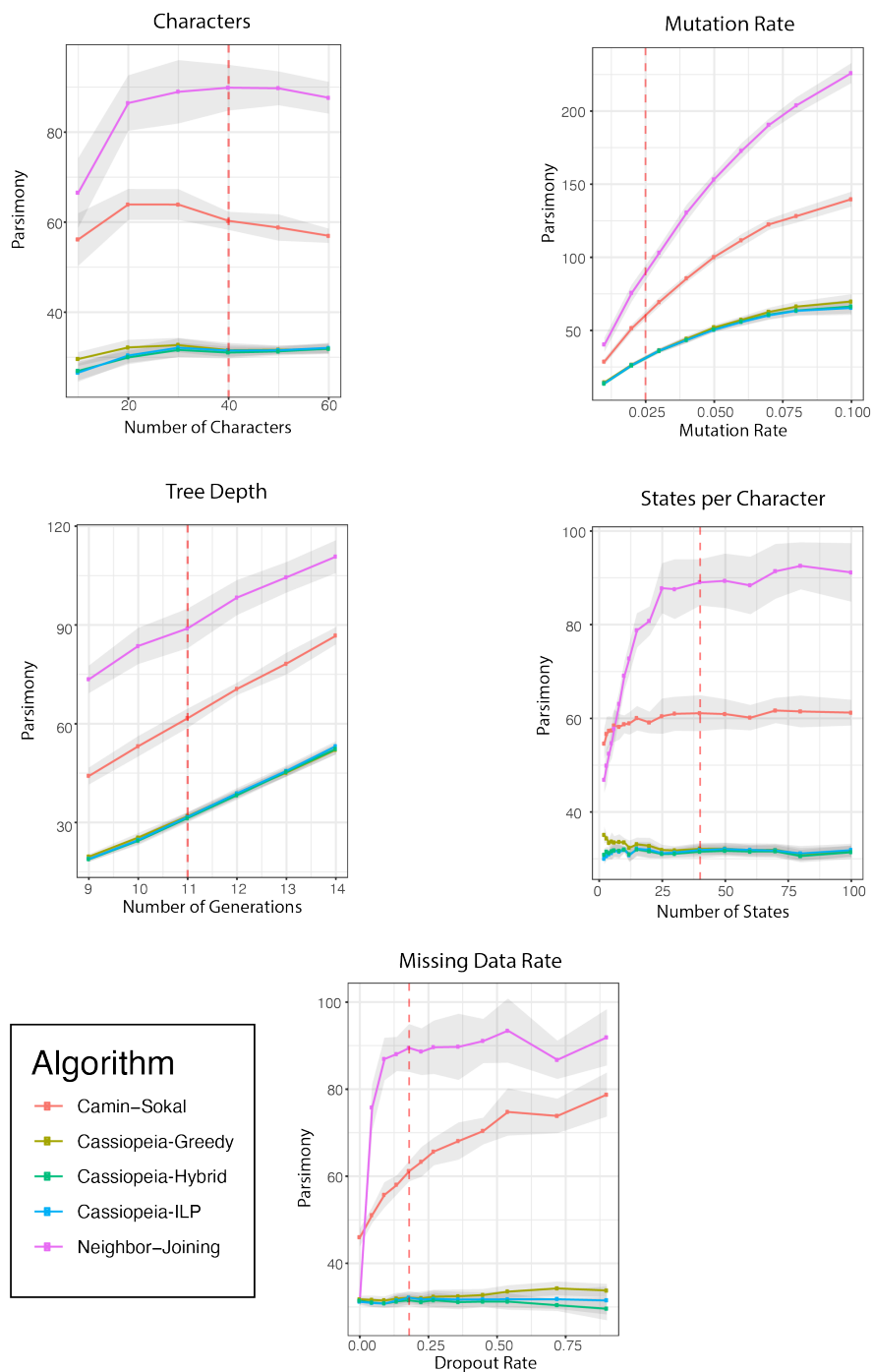


Figure 2.15: **Parsimony of reconstructed trees of 400 cell simulated datasets.** Parsimony scores (or number of evolutionary events) for each reconstructed network presented in **Figure 2.2** were calculated and compared across phylogeny reconstruction methods. Results are presented for the number of characters, the mutation rate, tree depth, number of states and dropout rate for all five algorithms used in this study. Standard error is represented by shaded area.

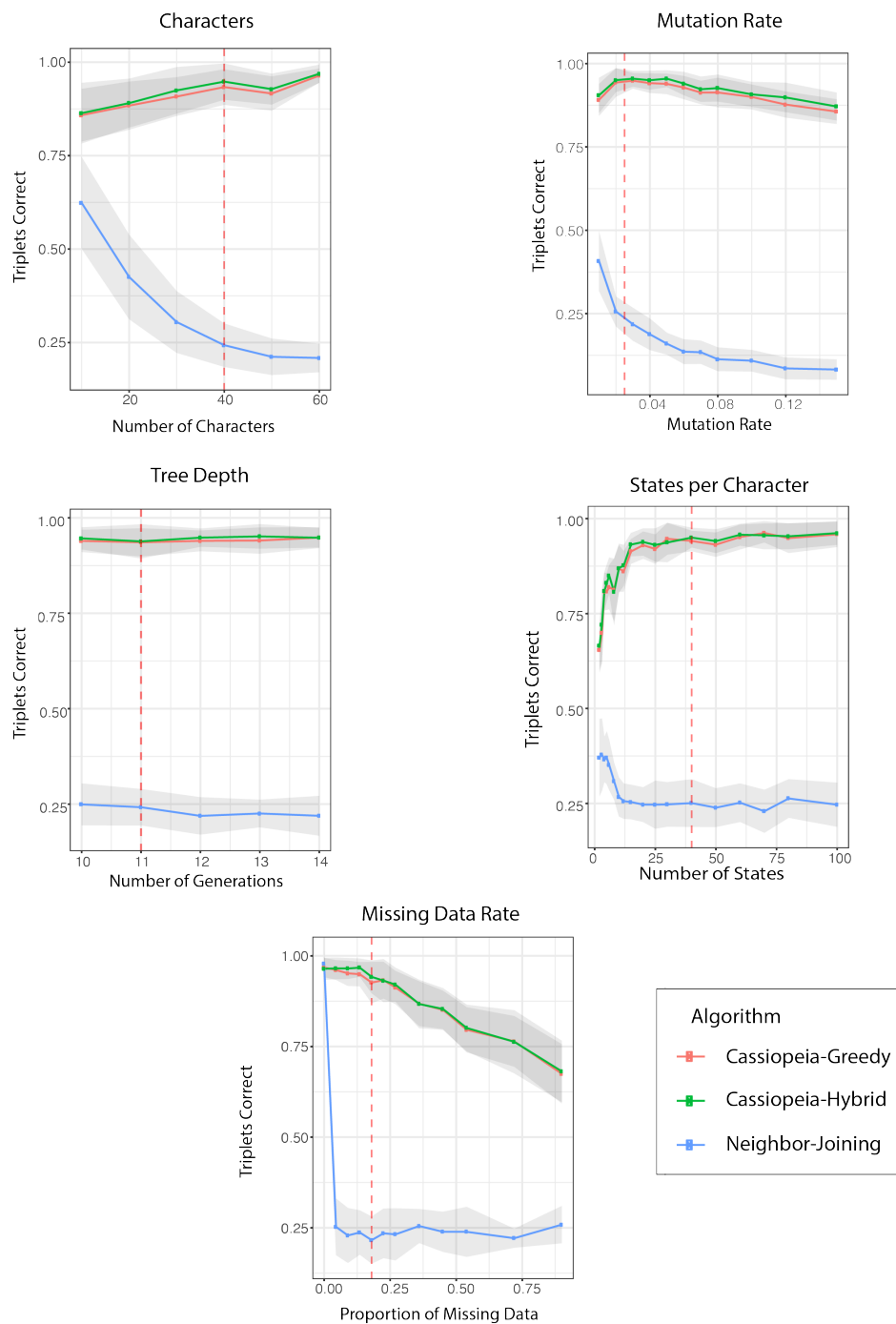


Figure 2.16: **Benchmarking of lineage tracing algorithms on 1000 cell synthetic datasets.** Phylogeny reconstruction algorithms were benchmarked on simulated trees consisting of 1,000 cells. The number of characters, character-wise mutation rate, length of experiment or tree depth, number of states, and dropout rate were tested. Due to scalability issues, only greedy, hybrid, and neighbor-joining were tested. Standard error is represented by shaded area.

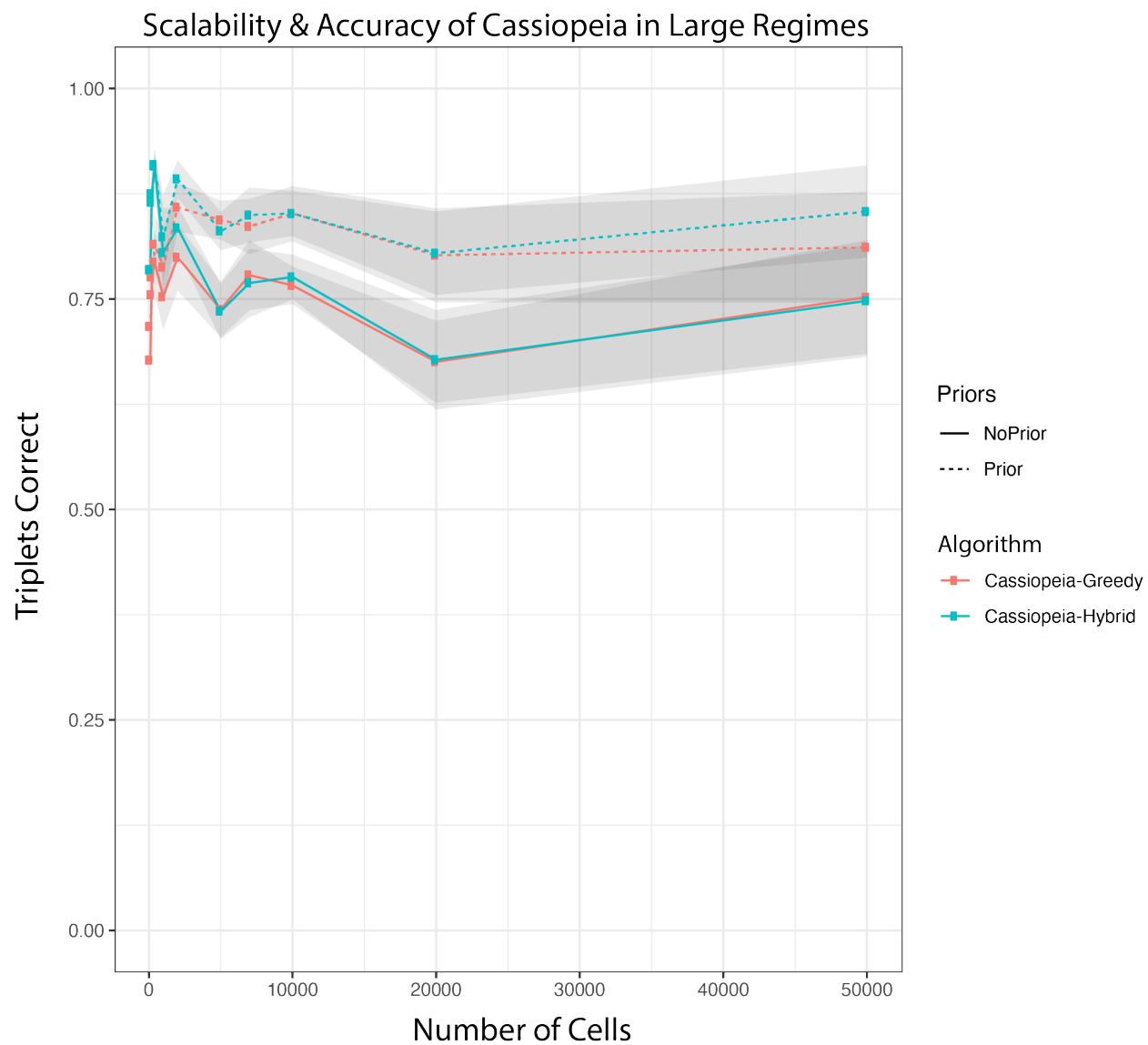
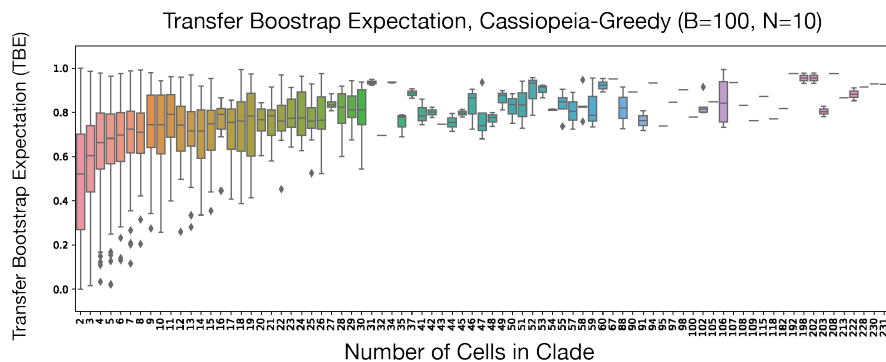
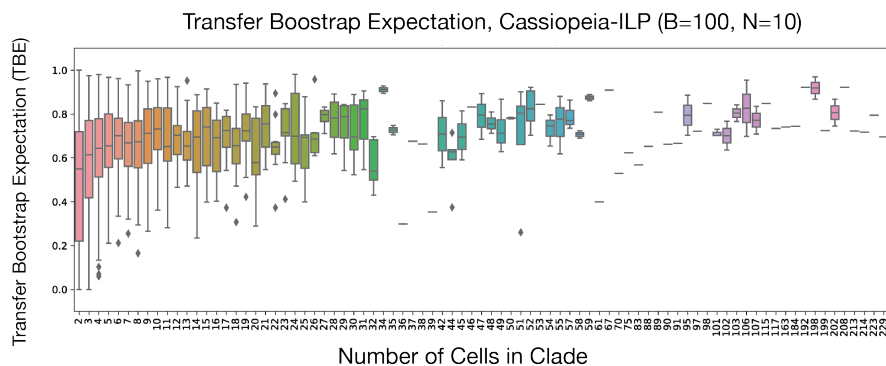


Figure 2.17: **Benchmarking of greedy and hybrid algorithms on large experiments.** Triplets correct is used to measure the accuracy of reconstructions using both hybrid and greedy algorithms on large trees (up to 50,000 cells). Of note, hybrid and greedy have comparable results on larger trees, which remain accurate even in these massive regimes. In addition, the knowledge of prior probabilities of particular states confers a large increase in accuracy.

a



b



c

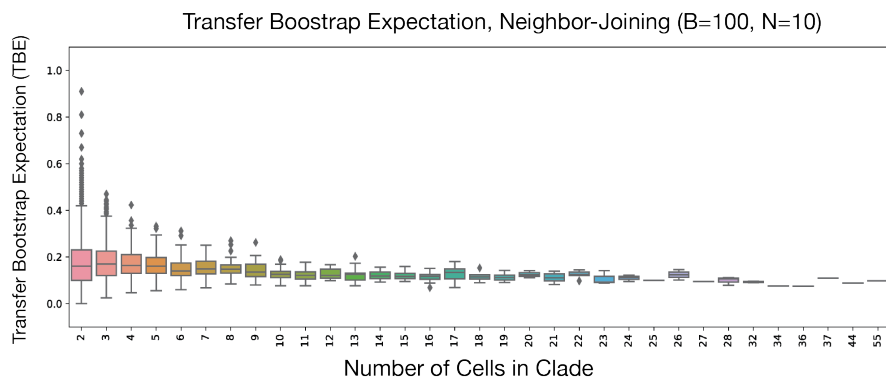


Figure 2.18: **Bootstrapping analysis of Cassiopeia and Neighbor-Joining with the Transfer Bootstrap Expectation statistic.** Bootstrap analysis of robustness for Cassiopeia-Greedy (a), -ILP (b), and Neighbor-Joining (c). 100 bootstrap samples ($B = 100$) were taken for 10 simulated trees ($N = 10$) by sampling characters with replacement and each matrix was used for reconstruction by each of the tree algorithms. The Booster software [61] was used to assess robustness of each clade in the original reconstruction, as measured with the Transfer Bootstrap Expectation (TBE) statistic. The distribution of TBE's is shown for each algorithm as a function of the size of the clade (i.e. a clade with two leaves underneath it will be of size 2).

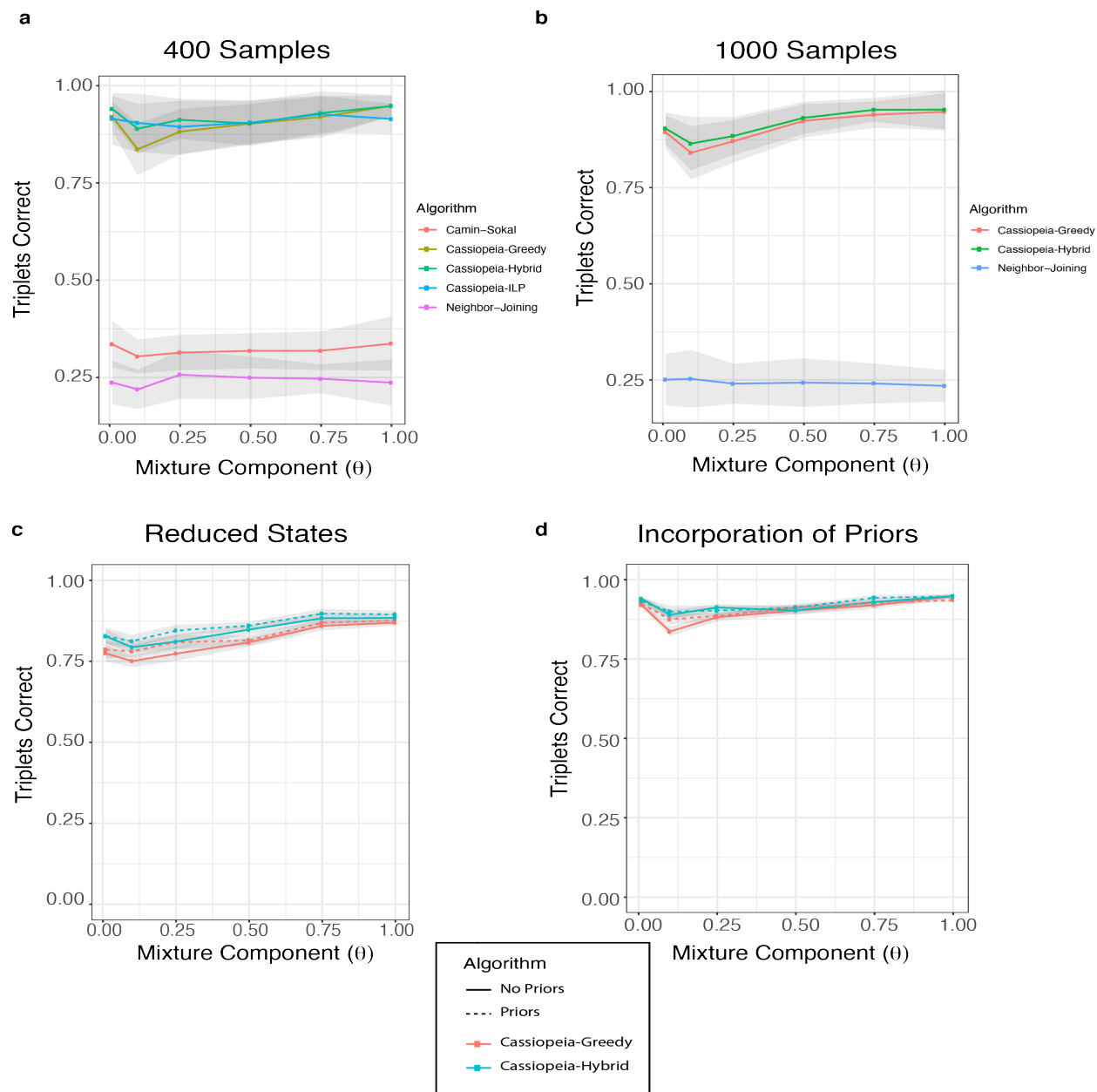


Figure 2.19: **Reconstruction accuracy under over-dispersed state distributions.** The effect of the indel distribution (i.e. the relative propensity for a given indel outcome) was explored in various regimes using a mixture model. Here, the mixture model consisted of mixing the inferred indel distribution with a uniform distribution between 0 and 1.0 with some probability θ (i.e. when $\theta = 1.0$, the indel distribution was uniform). In all simulations, we used default parameters for the simulated trees unless stated otherwise (40 characters, 40 states, depth of 11, median dropout rate of 17%, and a character mutation rate of 2.5%). (a) displays the results of all five algorithms over 400 samples. (b) displays results for simulations over 1000 samples for hybrid, greedy, and neighbor-joining methods. (c) Simulations for 400 samples using 10 states rather than 40 states per character. Dashed lines represent reconstructions performed with priors. (d) Simulations over 400 samples and 40 states, comparing results with and without priors. Dashed lines represent reconstructions performed with priors.

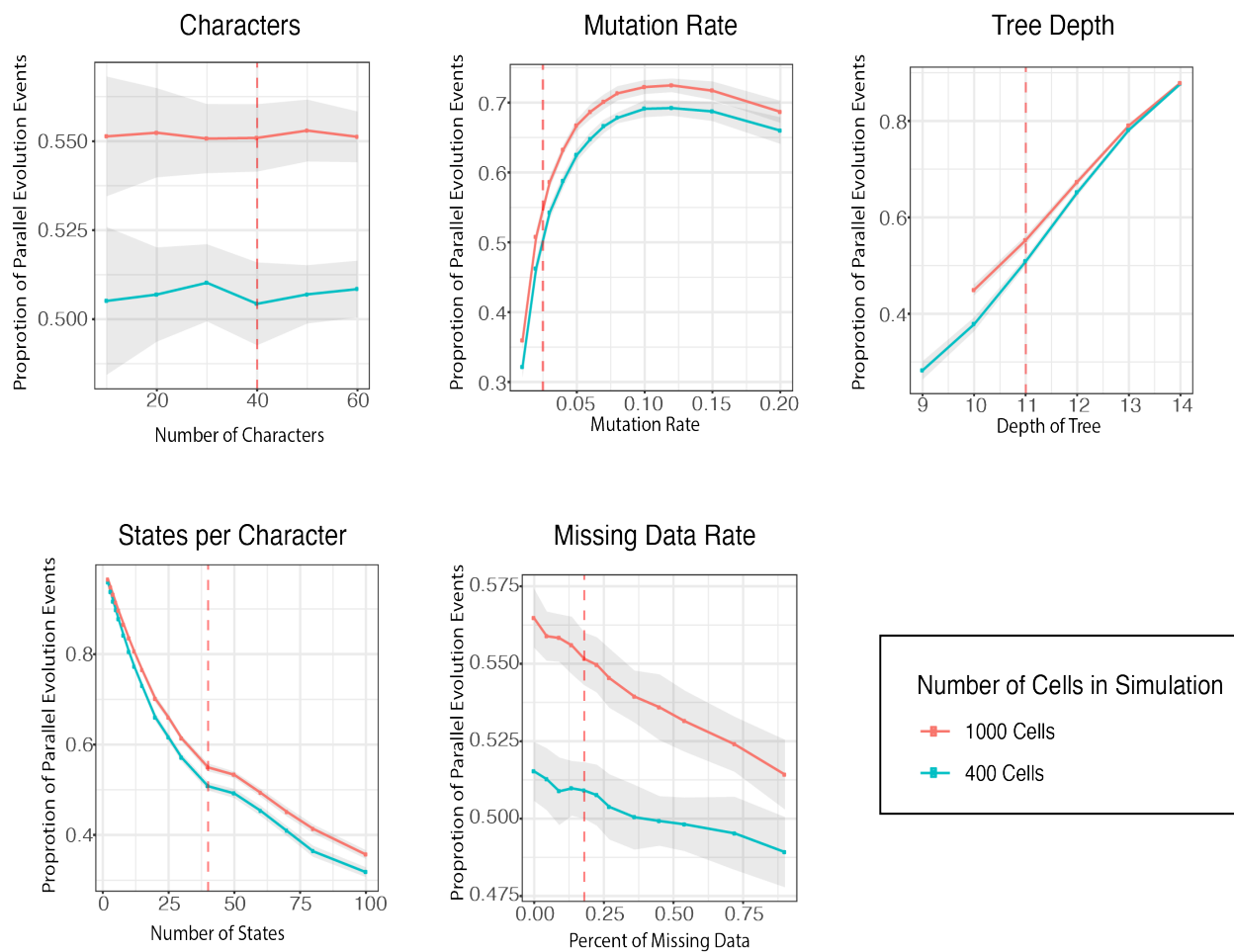


Figure 2.20: **Observed Proportion of Parallel Evolution in Simulations.** Inferred proportion of parallel evolution, as defined by the proportion of mutations that are observed more than once in a given tree, for the simulations presented in **Figure 2.2** and **Figure 2.16**.

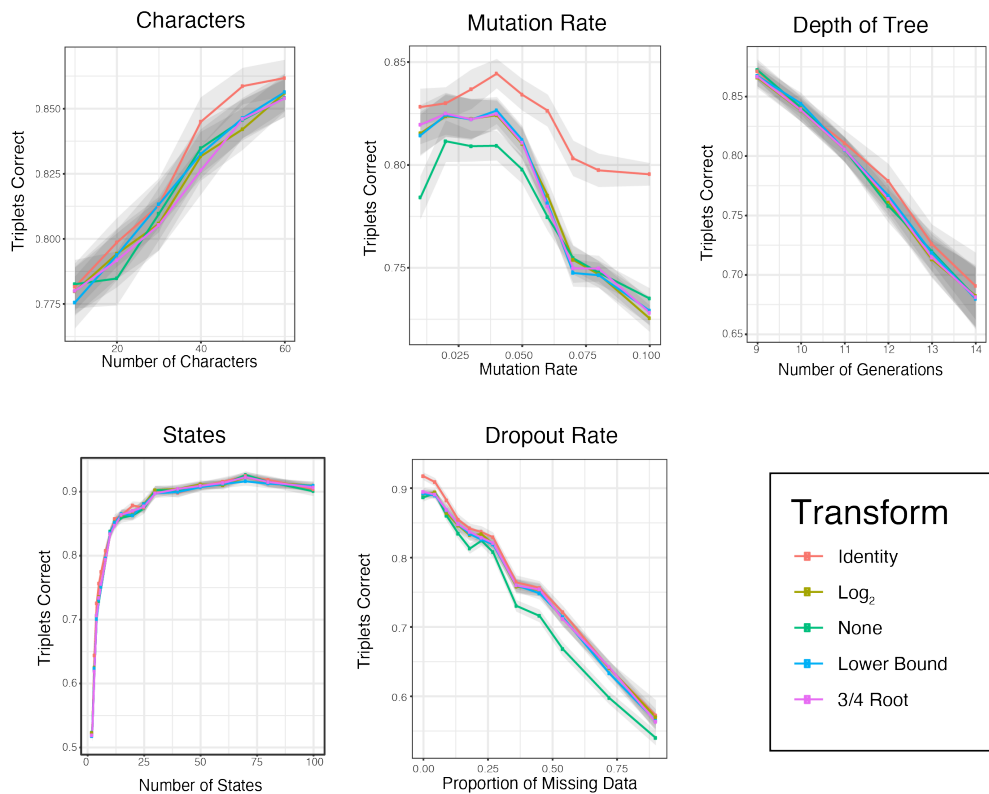


Figure 2.21: **Determination of the indel prior transformation function.** The effect of incorporating the prior probabilities of mutation events into the greedy algorithm is explored using synthetic datasets. The exact mutation probabilities used for simulations are used during reconstruction (i.e. the mutations drawn during simulation). Five possible transformations $f(n_{i,j})$, representing an approximation of the future penalty of not choosing this mutation (see methods) were tested for incorporation with the priors. The transformations were: (i) Identity ($f(n_{i,j}) = n_{i,j}$), (ii) Log₂ ($f(n_{i,j}) = \log_2(n_{i,j})$), (iii) None ($f(n_{i,j}) = 1$), (iv) Lower Bound ($f(n_{i,j}) = \min(n_{i,j}, \frac{N}{20.0})$), and (v) $\frac{3}{4}$ root ($f(n_{i,j}) = (n_{i,j})^{\frac{3}{4}}$). $n_{i,j}$ denotes the number of cells which report the mutation j in character i and N is the total number of samples. To test these transformations, we evaluated the resulting tree accuracy via Triplets Correct. Standard error is represented by shaded area.

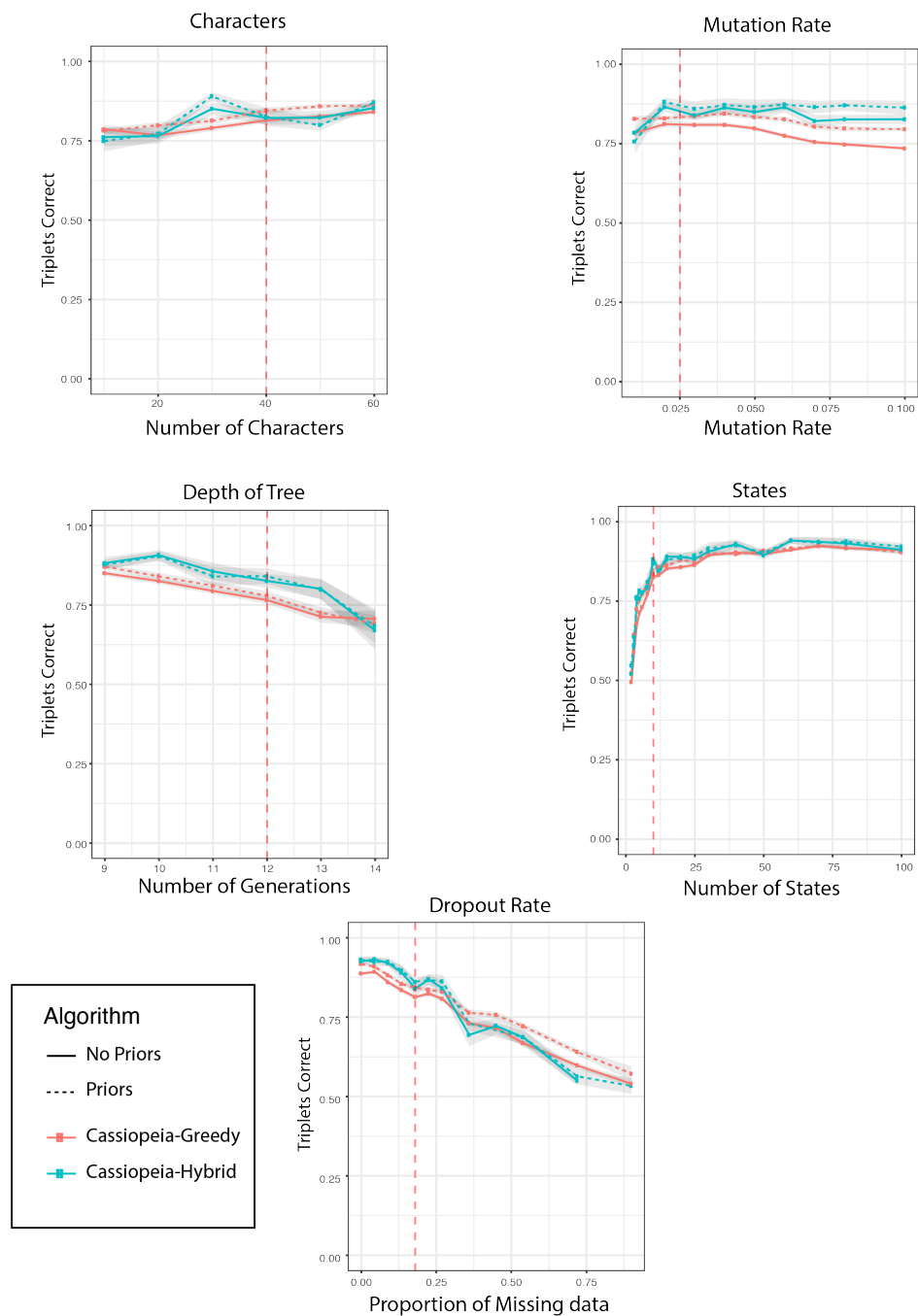


Figure 2.22: **Incorporation of priors into Cassiopeia.** A comparison of tree accuracy when using priors for both the greedy-only method and Cassiopeia. We compared performance as we varied the number of characters per cell, the mutation rate per character, the length of the experiment, the number of states per character, and the amount of missing data. Standard error is represented by shaded area.

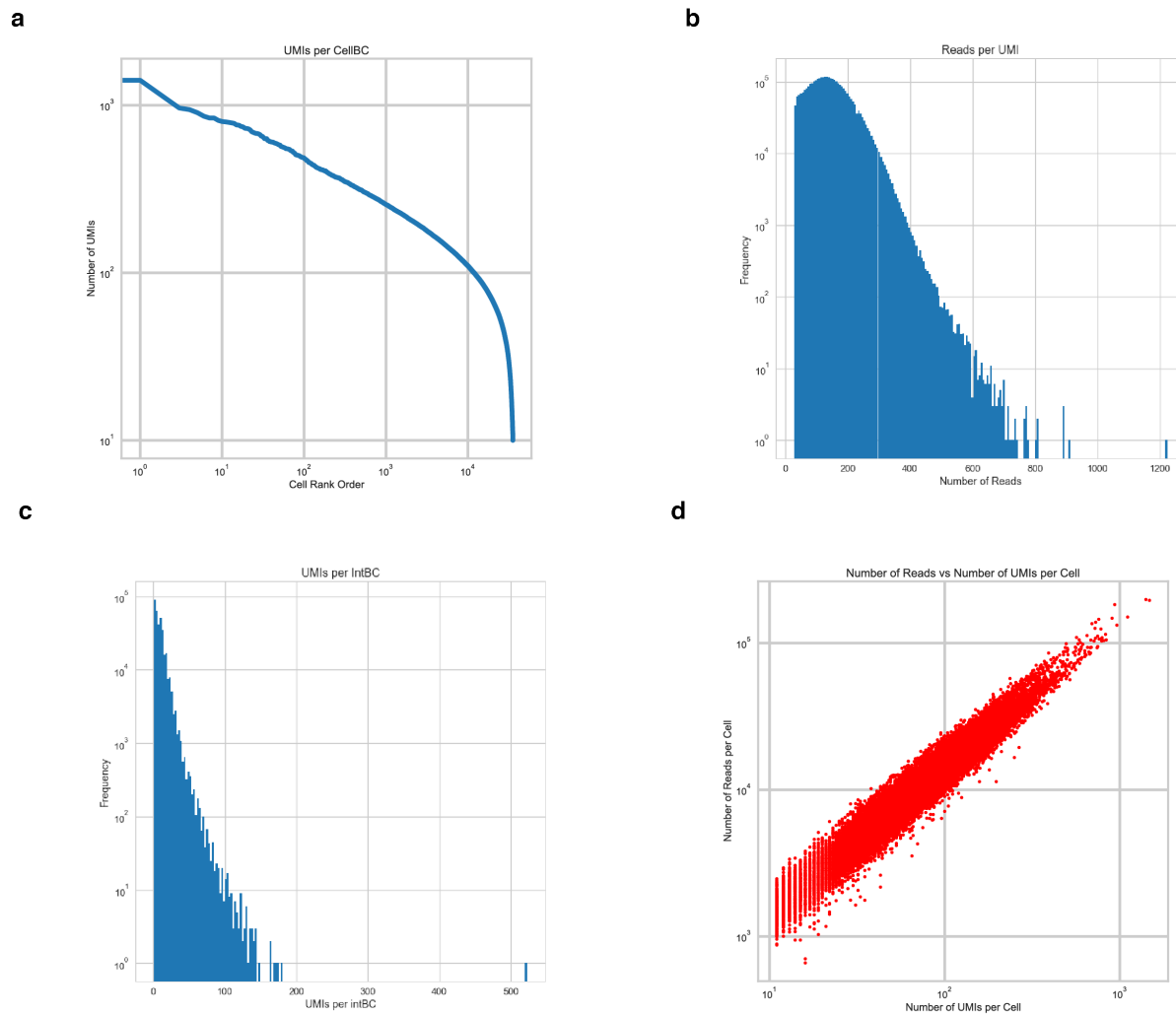


Figure 2.23: **Quality control metrics for the target site sequencing library processing pipeline.** (a-d) present quality control metrics after the processing pipeline. (a) Cells are ranked by the number of UMIs they contain, showing a median of 76; (b) The number of reads per UMI after UMI error correction and collaping, showing a median of 137; (c) The number of UMIs per integration barcode (intBC), showing a median of 7; (d) is the concordance between reads per cellBC and UMIs per cellBC, showing a pearson correlation of 0.96

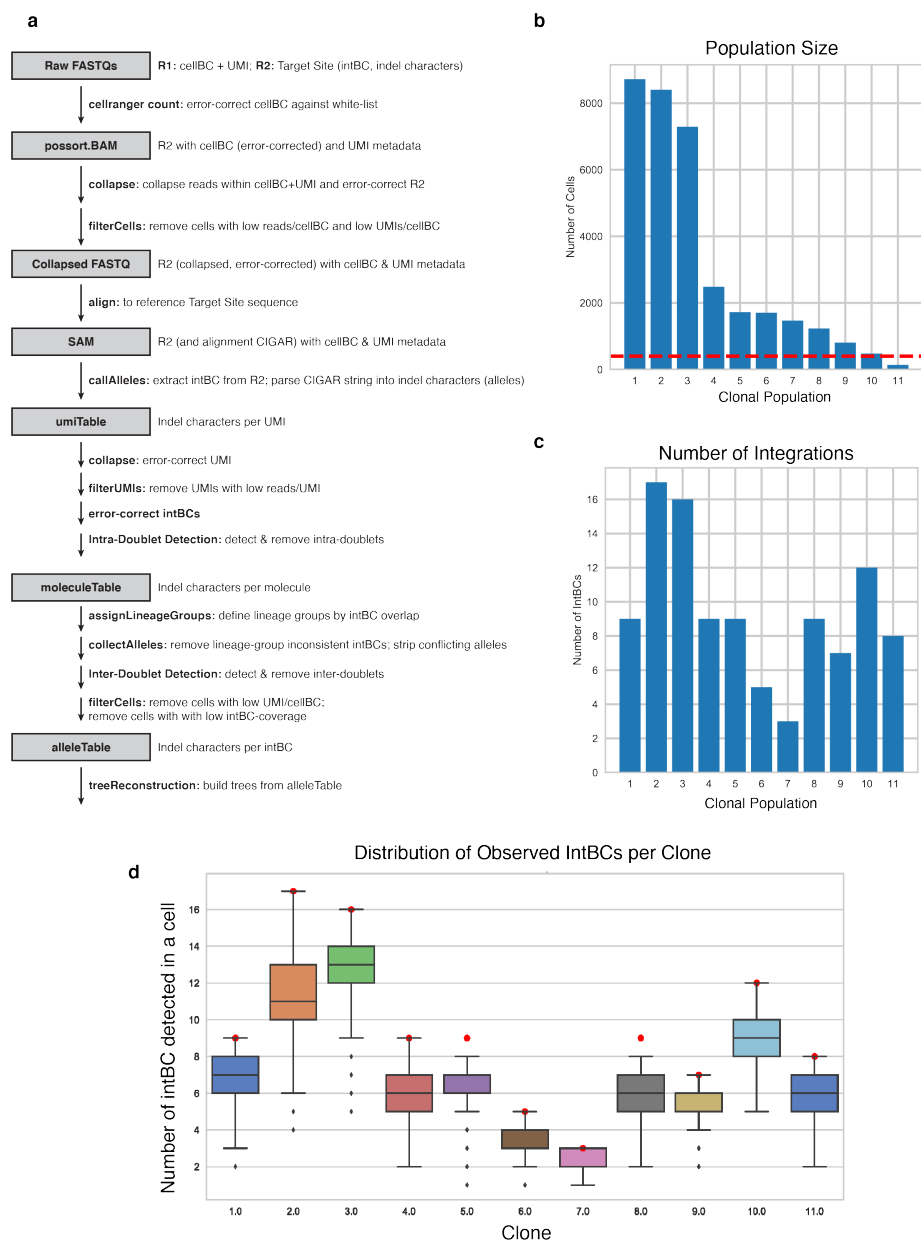


Figure 2.24: **Processing Pipeline for the *in vitro* dataset.** (a) shows a more in-depth flowchart of the Cassiopeia processing pipeline taking as input the raw FASTQs from a sequencing run and converting the observed reads into final trees. Cellranger “count” is used to map reads to dummy transcriptome (junk sequence that nothing will align to), filter cells, and read off the 10x cell barcodes and UMIs. The resulting BAM file is then passed through a series of cell filtering, UMI error correction, and allele mapping before becoming the final allele table that can be converted to character matrices for clone reconstruction. See methods for more detailed information for each step. (b-d) present additional summary statistics for the final allele table. (b) displays the number of cells per clone; (c) shows the median number of intBCs observed in each clone; (d) shows the distribution of the number of intBCs observed in each cell (red points are references to indicate the number of intBCs used to reconstruct the particular clone).

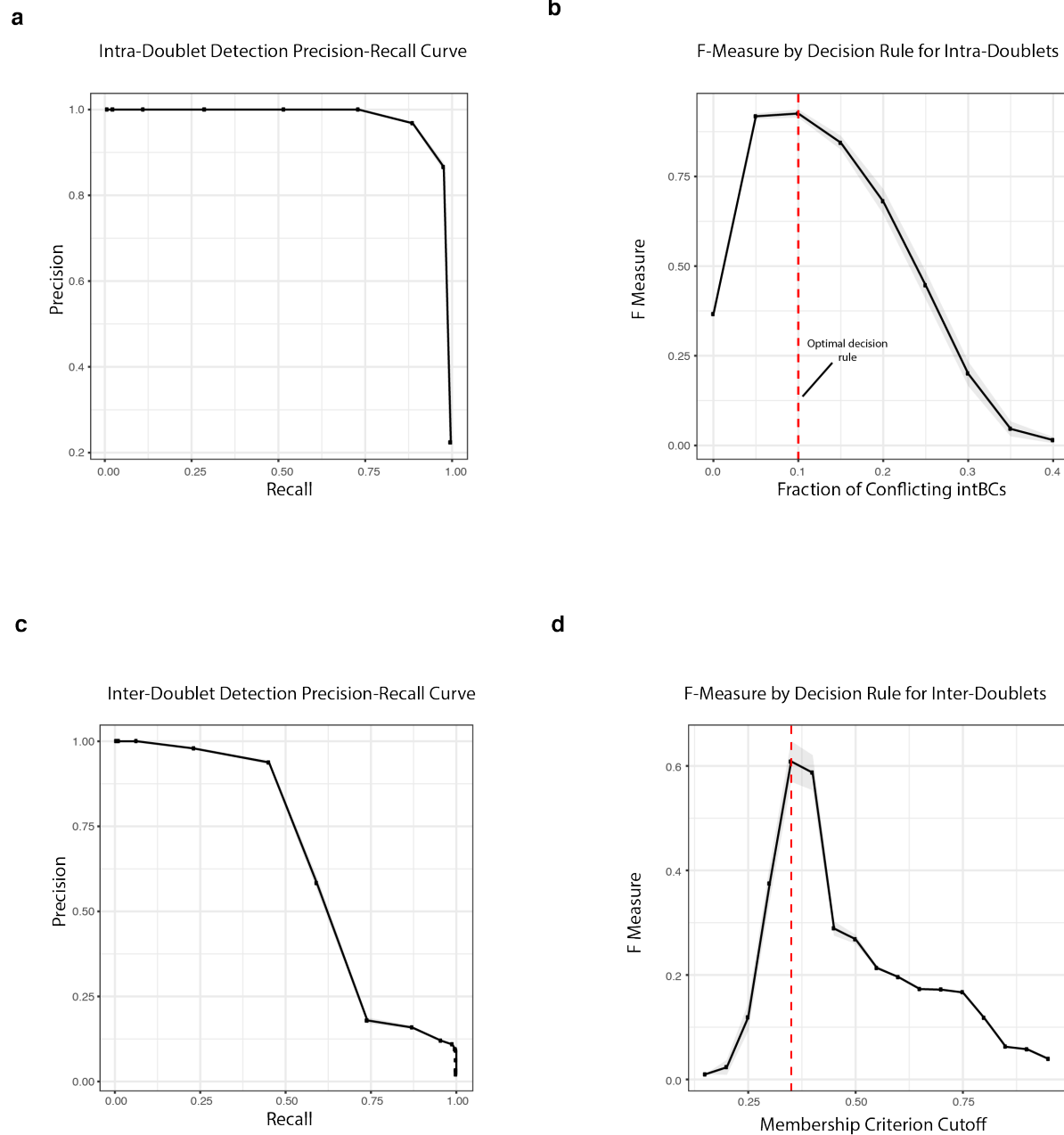


Figure 2.25: **Identification of doublets using intBCs.** IntBCs are used to identify doublets. (a-b) report the ability to identify doublets arising from the same clone, referred to as “intra”-doublets; (c-d) report the ability to identify doublets arising from different clones, referred to as “inter”-doublets. Doublets were simulated using the final allele table and 200 “intra”- and “inter”-doublets were created in each of 20 replicates. Precision-recall curves for intra- and inter-doublet detection methods are presented in (a) and (b), respectively. (c) and (d) present the F-measure (defined as the weighted harmonic mean between precision and recall) of detection methods for intra- and inter-doublets, respectively. Red-dashed lines denote the optimal decision rule for doublet detection. Standard error is represented by shaded area.

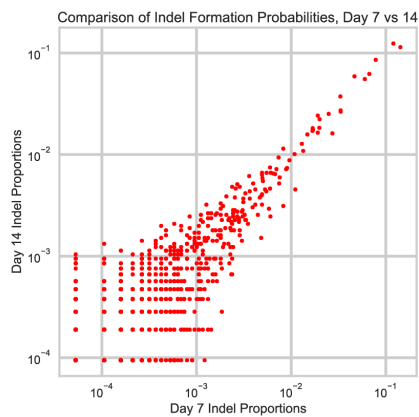
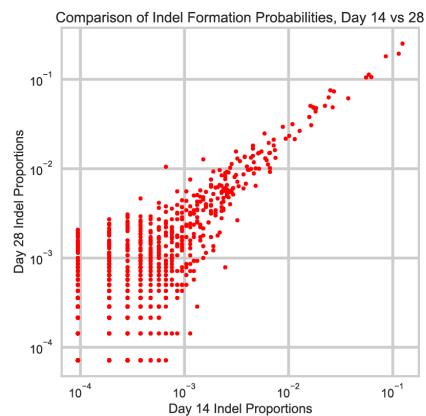
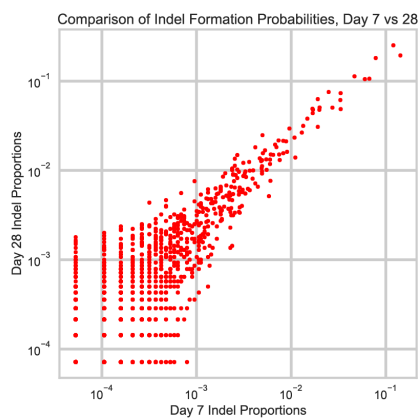
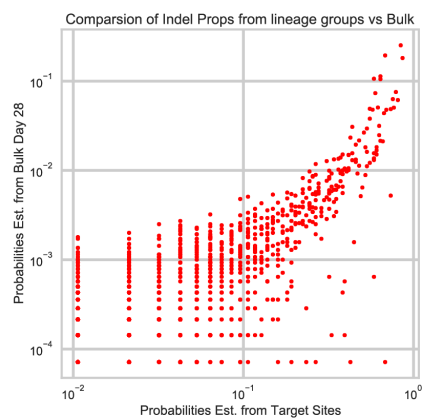
a**b****c****d**

Figure 2.26: Estimation of Prior Probabilities for Tree Reconstruction. Prior probabilities to be used during tree reconstruction can be determined from both a bulk assay and independent clonal populations. Prior probabilities of mutations were determined by calculating the proportion of unique intBCs that report a particular indel (see methods). The bulk assay consisted of several independent clones with non-overlapping intBCs grown over the course of 28 days. (a-c) report the correlation of indel formation probabilities between various time points in the bulk experiment. A strong correlation is observed between all time points: 7 and 14 (a), 14 and 28 (b) and 7 and 28 (c). Indel formation probabilities can also be calculated using the intBCs from each clone as independent measurements. Using this method, (d) reports the correlation between this lineage-group specific probability calculation and the last time point of the bulk assay.

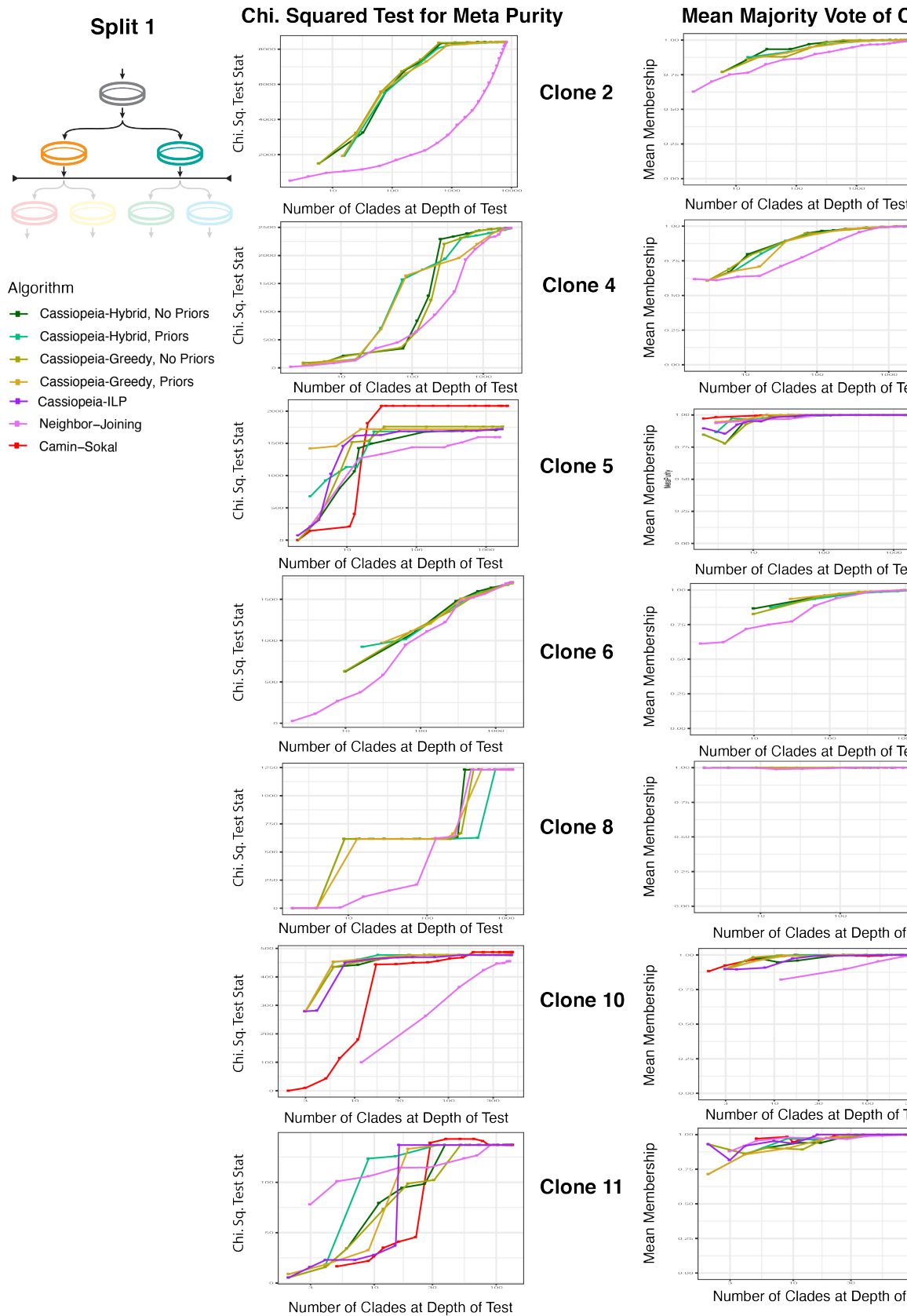


Figure 2.27: **Evaluation of algorithms on *in vitro* lineage tracing clones, First Split.** Trees were reconstructed for the remaining clones in the *in vitro* dataset that consisted of more than 500 unique cell states. LG2, LG4, LG6, and LG8 passed this threshold and were reconstructed with Cassiopeia (with and without priors), greedy-only (with and without priors) and Neighbor-Joining. The

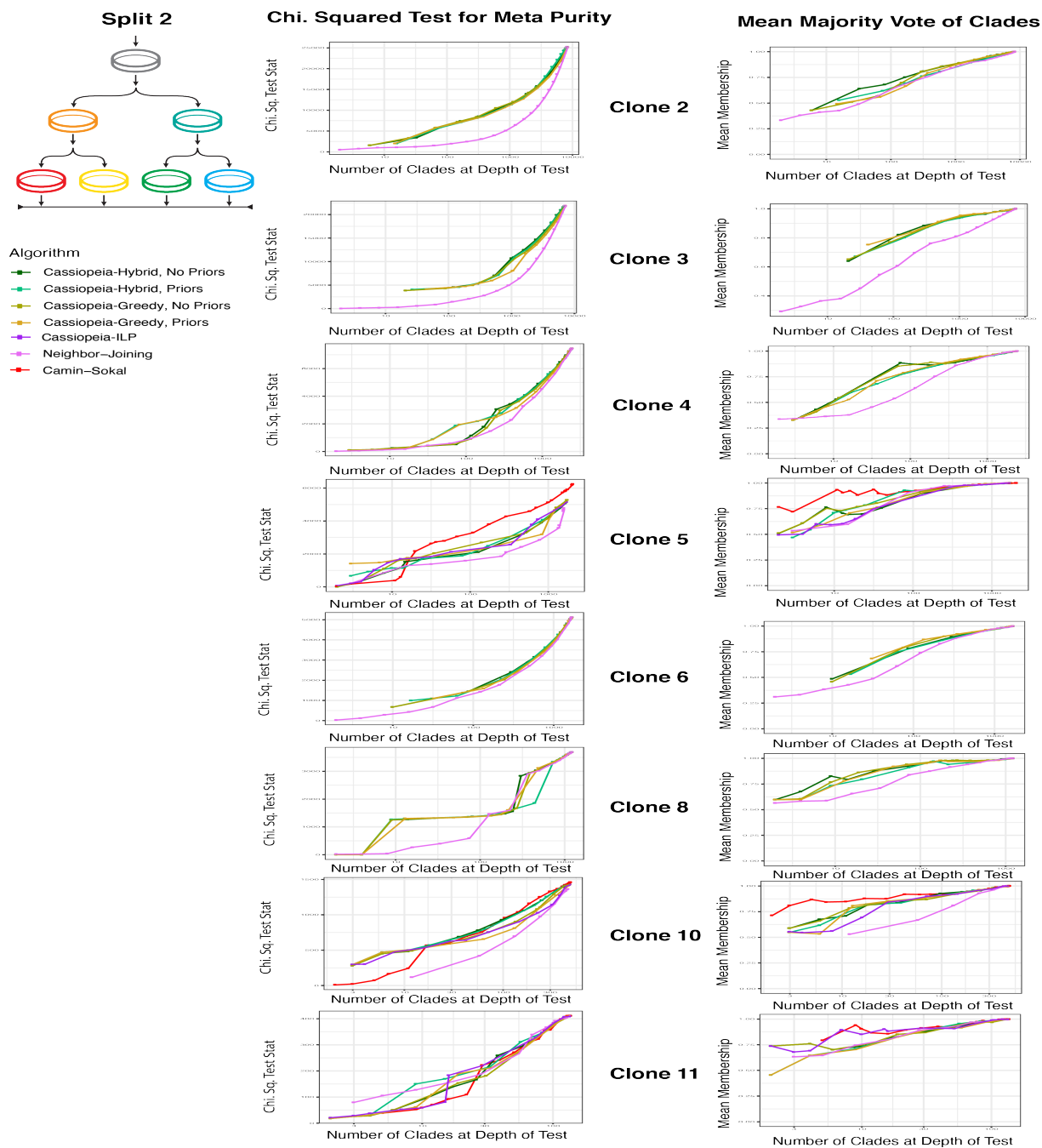


Figure 2.28: **Evaluation of algorithms on *in vitro* lineage tracing clones, Second Split.** Trees were reconstructed for the remaining clones in the *in vitro* dataset that consisted of more than 500 unique cell states. LG2, LG4, LG6, and LG8 passed this threshold and were reconstructed with Cassiopeia (with and without priors), greedy-only (with and without priors) and Neighbor-Joining. The statistics provided were taken with respect to the second split ID (see methods). For both Cassiopeia with and without priors, we used a cutoff of 200 cells and each instance of the ILP was allowed 5000s to converge on a maximum neighborhood size of 6000.

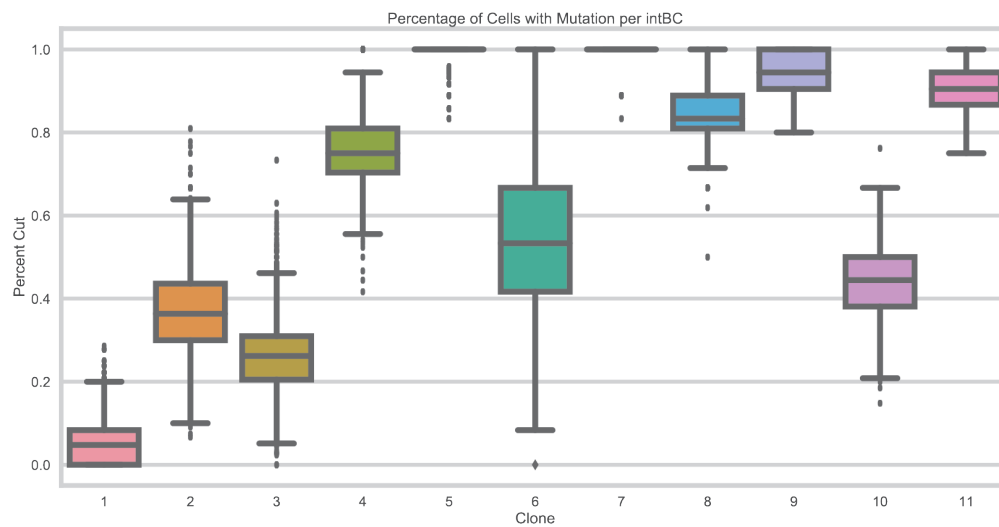
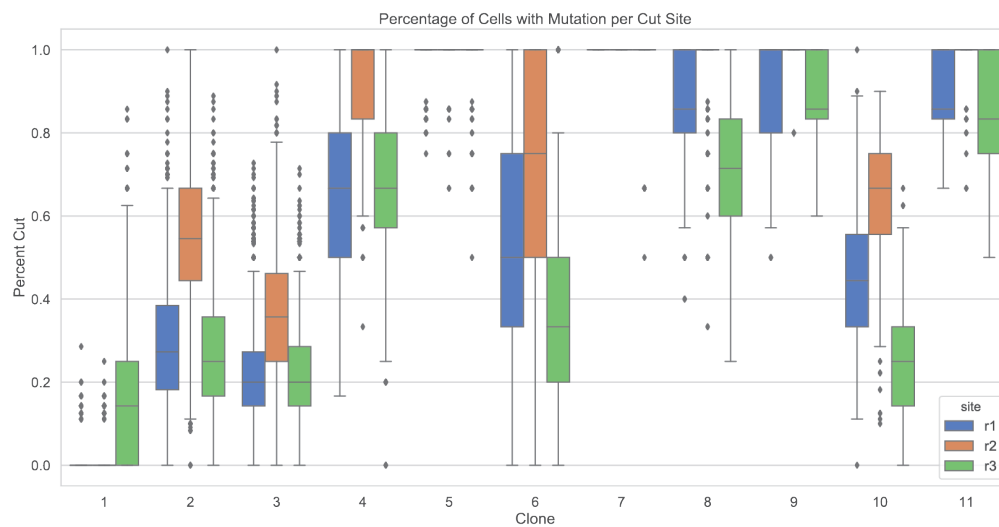
a**b**

Figure 2.29: **Exhaustion of Target Sites across Clones.** Target site exhaustion for each clone, as measured by the proportion of sites observed as edited after the experiment. (a) presents the percentage of mutated cells across all cut sites per clone. (b) details the distribution of mutated cells per cut site in each clone.

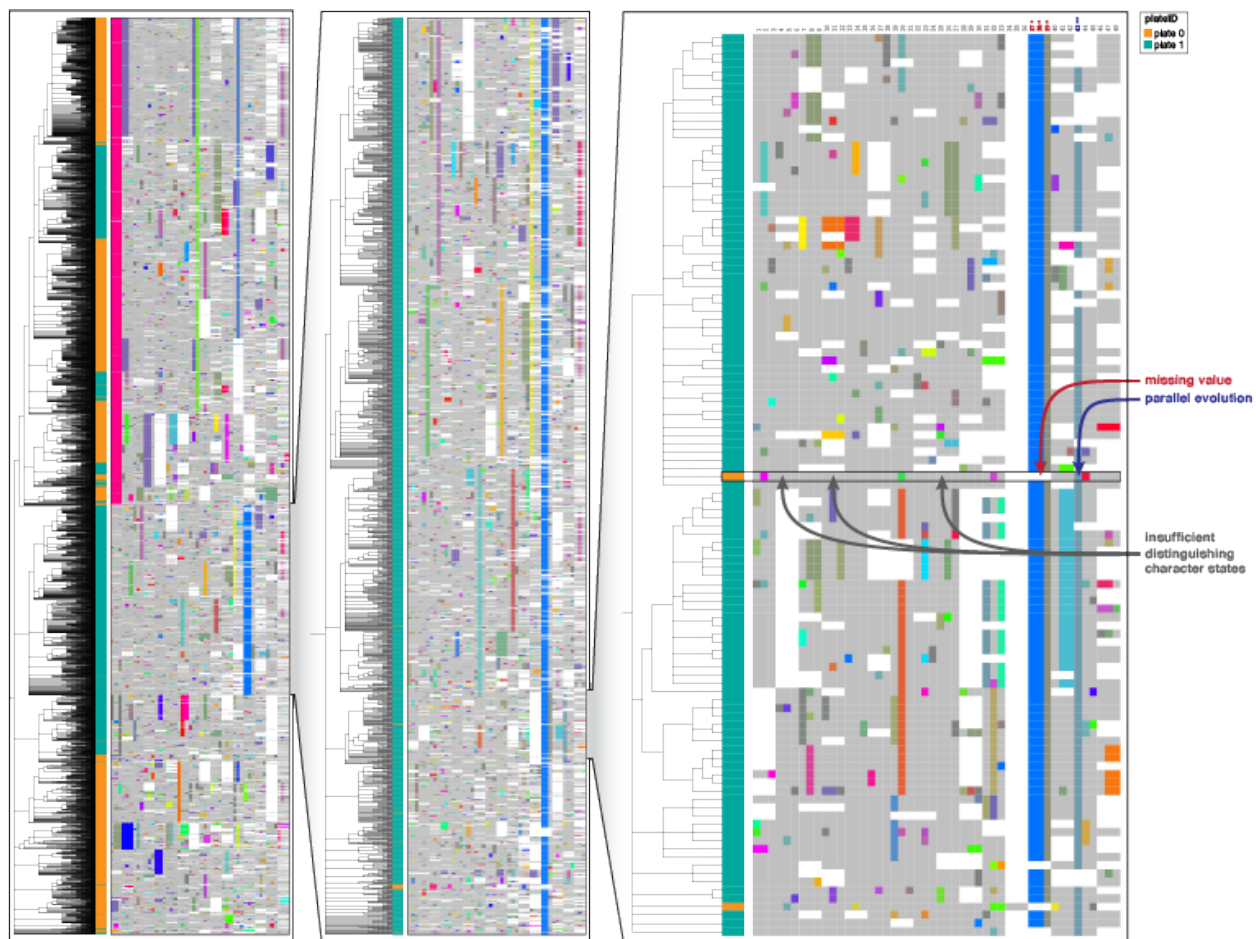


Figure 2.30: **Vignette of Inferential Mistakes for Clone 3.** An example from the reconstruction of Clone 3 with Cassiopeia-Hybrid where a cell has been misplaced in the tree due to several factors. In this case, it is clear that the cell was placed where it is due to an instance of parallel evolution of the state in character 43 (as annotated in the figure). Because the cell contained this state, it was grouped with cells of a different plate also containing this mutation. Furthermore, the cell contains few distinguishing mutations thus making it difficult to infer the true value of the missing values located in characters 37-39.

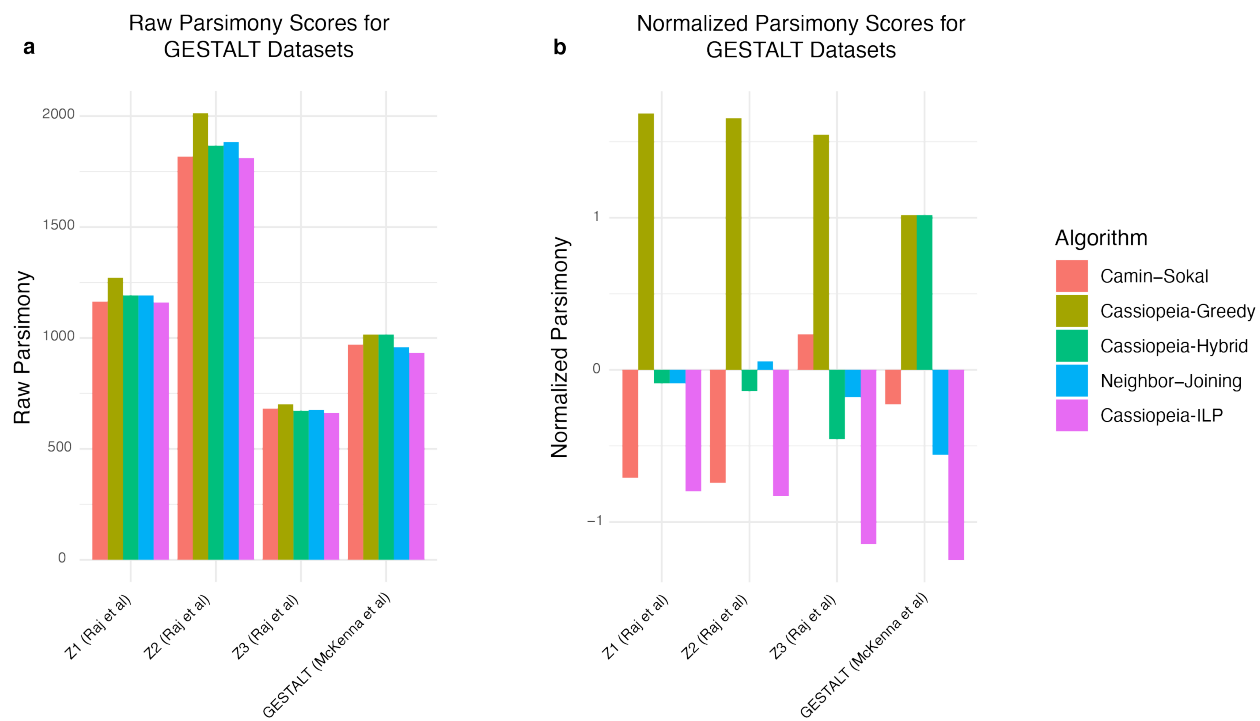
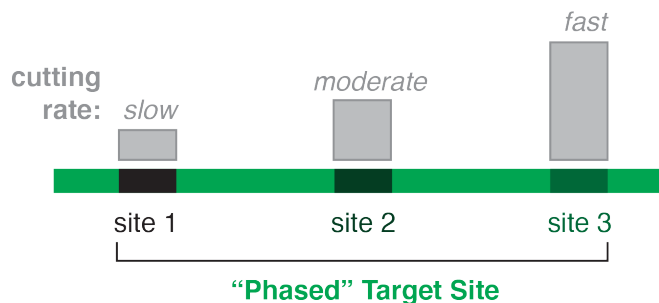
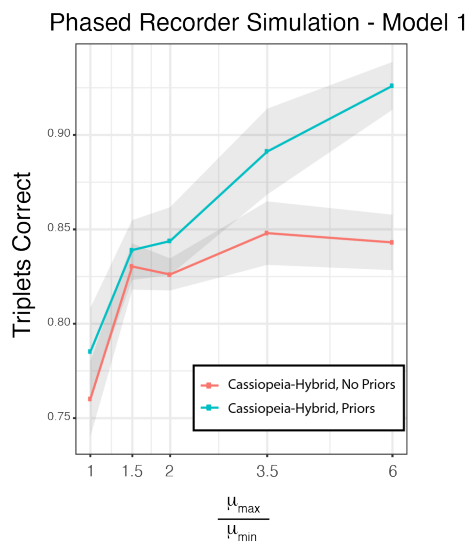


Figure 2.31: **Parsimony scores from reconstructions of the GESTALT datasets.** (a) Raw and (b) normalized parsimony scores for the parsimony scores from the GESTALT datasets. Camin-Sokal, Neighbor-Joining, Cassiopeia-Greedy, -Hybrid, and -ILP were run on datasets from Raj et al [6] and McKenna et al [3]. Raw parsimony scores are calculated as the number mutations present in a phylogeny (summing over the mutations along every edge of the tree). The normalized scores correspond to z-scores for each dataset.

a



b



c

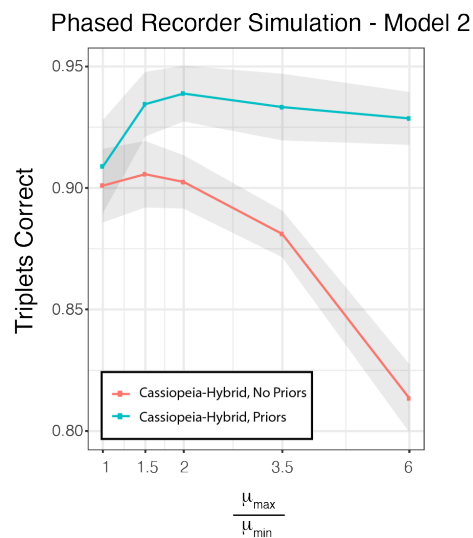


Figure 2.32: **“Phased Recorder” leverages variability across target sites.** (a) Design concept of the “Phased Recorder.” (a) We simulated a “phased” editor, where each character is mutated at variable rates. (b-c) We varied the amount each character could vary across 5 different experiments and simulated using two different indel formation rate models. Each cell had 50 characters with 10 states per character and a mean dropout of 10%. The amount of mutation variability is described with the ratio between the maximum and minimum mutation rates ($\frac{\mu_{max}}{\mu_{min}}$). Standard error is represented by shaded area. (b) Model 1 consists of drawing indels from a negative binomial distribution $NB(5, 0.5)$ where there are few “rare” indels. (c) Model 2 consists of drawing indels from the splined distribution of the empirical dataset’s indel formation rates, as used in other synthetic benchmarks.

Chapter 3

**Extending reproducible and accurate
lineage analysis with Cassiopeia2.0**

3.1 Introduction

Cell lineages underlie several important biological phenomena across scales and domains - such as embryogenesis, differentiation, and cancer progression - and deviations in these cell lineages can lead to several disease states. Thus it is critical to understand the dynamics of these processes over space and time. Lineage tracing technologies offer a suite of approaches for profiling these lineages by prospectively labeling and tracking the progeny of a progenitor population or inferring relationships between tissues or cells retrospectively [284, 269].

Recent technologies have greatly improved the resolution and control of lineage tracing by exploiting developments in single-cell assays and CRISPR/Cas9 engineering (hereafter referred to as "single-cell lineage tracing technologies"; reviewed in [15, 177, 269]). Generally, these technologies engineer cells with a synthetic array of Cas9 targets (often in the 3' untranslated region [UTR] of a fluorescent protein) that are stochastically cut by Cas9, resulting in a stable and heritable insertion or deletion ("indel"). Because these targets are often located on a fluorescent protein, they are transcribed and can be read out using single-cell sequencing assays. Since these Cas9-induced mutations are heritable and new mutations accrue over time, the relationships between observed samples can be deduced from the set of mutations using one of several algorithms and summarized with a "phylogenetic tree". These approaches have been successfully applied in model organisms to study development in zebrafish [178, 206] and mouse [37] as well as tumor progression [289] and metastasis [203, 228].

A key step in these technologies is the analysis of mutation profiles to infer phylogenetic trees.

Generally, the task of phylogenetic inference is computationally intractable due to the phenomenon of “homoplasy” in which an identical mutation appears in independent lineages, thus introducing ambiguity into the relationships between samples. For example, finding the most parsimonious or likely tree is NP-Hard [42, 76]. With regards to single-cell lineage tracing technologies, we and others have provided overviews of the limitations in the current technologies [218, 127]. Briefly, there are three main limitations. Firstly, often the scale of the problem exceeds traditional applications of phylogenetic algorithms and thus necessitates efficient, heuristic-based solutions. Secondly, often between 20% and 50% of the mutations in each cell are not observed, either due to the sensitivity of the single-cell assay or due to transcriptional silencing of the target site. Thirdly, the likelihood of Cas9-induced indels is “light-tailed” whereby a few outcomes are very likely while most are rare, therefore increasing the rate of homoplasy in the system.

To address these challenges, we and others have introduced algorithms that are specifically tailored to the task of phylogenetic inference in the context of single-cell lineage tracing [127, 70, 293, 66, 92]. Building on these efforts, here we discuss our continuing efforts on building an open-source computational library for lineage tracing data processing, algorithmic development, benchmarking, and analysis in a single, integrated development. These tools are publicly available on Github at <https://www.github.com/YosefLab/Cassiopeia>, which includes extensive documentation and tutorials. In this study, we demonstrate how these tools in Cassiopeia can be used to prioritize lineage tracing engineering regimes and benchmark new algorithms.

3.2 Results

3.2.1 An overview the Cassiopeia2.0 Library

Cassiopeia is an open-source, end-to-end pipeline for single-cell phylogenetic analysis written in the Python programming language. Though developed initially for CRISPR/Cas9 -based lineage tracing systems [127], the software suite described below can be utilized in a host of contexts, agnostic to the underlying data type. In its current state, Cassiopeia is split up into five core modules: *Cassiopeia-preprocess*, *Cassiopeia-solve*, *Cassiopeia-simulate*, *Cassiopeia-benchmark*, and *Cassiopeia-tools* (**Figure 3.1**). These modules interface together to enable end-to-end analysis of lineage tracing data as well as further experimentation.

Cassiopeia-preprocess provides a parallelizable pipeline for creating “character matrices” from sequencing data. To be more specific, the first step in any phylogenetic problem is to create an alignment of heritable characteristics, or “characters”, where the states of these characters inform on relationships between samples. In the genomics age, these characters are often positions in biological sequences that can take on different nucleotide or amino acid values. In the case at hand, these characters are Cas9 target sites that accrue indels. Regardless of the underlying data modality, this information is often summarized in “character matrix” summarizing the observed states at each character across cells.

Preprocessing of sequencing data from single-cell assays is a critical step before any phylogenetic analysis can be performed. In the context of the CRISPR/Cas9-based lineage technologies described above, these preprocessing pipelines traditionally have involved transcript quantification

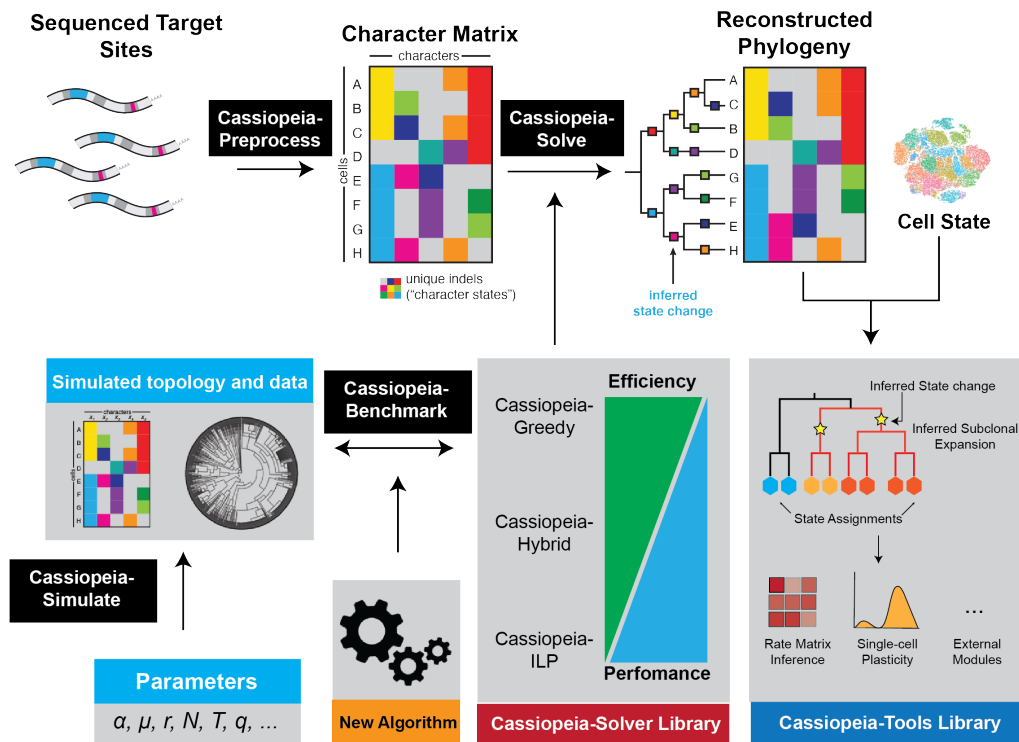


Figure 3.1: **An overview of the Cassiopeia V2 Library.** The Cassiopeia2.0 library is split up into five core modules for preprocessing sequencing data into character matrices, inferring phylogenies from character matrices, simulating lineage data, benchmarking new algorithms on simulated data, and analyzing phylogenetic trees with respect to paired phenotypic data.

and quality control filtering, consensus sequence identification, sequence alignment and indel identification, and error correction [127]. In some pipelines, technical artifacts like sequencing doublets are inferred and removed in addition to these steps. The product of this pipeline is the character matrix used for phylogenetic inference.

While in principle previously used preprocessing pipelines are similar in the types of procedures, each lineage-tracing technology requires special treatment due to unique target site constructs. As such, preprocessing has traditionally been performed using a set of scripts for the technology at hand [178, 37, 236]. From both a developer's and user's perspectives, this is not ideal in that it is

more difficult to troubleshoot errors and more burdensome to adapt to even minor changes in technologies. The **Cassiopeia-preprocess** module overcomes these issues by providing an integrated preprocessing pipeline that is both modular in its steps, allowing users to customize the pipeline, as well as flexible in its input, allowing users with various technologies to use the software.

After processing sequencing data, the **Cassiopeia-solve** module provides the tools for phylogenetic reconstruction. Subsequently, the *Cassiopeia-tools* library provides users with methodology to perform several downstream tasks on the inferred topology: for example, users can identify the extent to which there are subclones expanding faster than others. Moreover, in the context of the technologies discussed here that simultaneously measure the mRNA content of a cell, users can assess the heritability of transcriptional states on the tree.

Aside from analysis tools, the *Cassiopeia-simulate* and *Cassiopeia-benchmark* modules enable users to experiment with lineage tracing technology regimes on capturing cellular relationships or on algorithmic performance. For example, a user might be interested in if increasing the Cas9 mutation rate allows one to record more cell divisions and how it affects the performance of the Cassiopeia-Greedy algorithm? Collectively, the simulation engine offers users an affordable approach for experimenting with lineage tracing technologies and a common framework for testing algorithmic performance. In the following sections, we focus on how these two modules - for simulating and benchmarking - can be used to deduce optimal design principles for lineage tracers as well as efficiently assess the effectiveness of a new algorithm.

3.2.2 A simulation engine to test lineage tracing technology designs

An open question is how parameters of the lineage tracing technology might impact downstream tree recovery. For example, previous work has suggested that there are critical inference issues as a consequence of technology parameters [218]; more recent work has suggested how to improve inference, for example by increasing the number of Cas9 target sites [273]. With the simulation module, we have provided a principled and extendable approach for experimenting with how parameters affect tracing fidelity across tree topologies.

Cassiopeia-simulate splits up the simulation process into tree topology simulation and data simulation. By default, to generate the tree topology we utilize a generalized birth-death process with fitness for simulation of tree topologies [138, 51] (see Methods). This procedure provides a tree topology over a set of leaves with times specified on each branch length. Then, lineage tracing data can be overlaid on top of the tree topology using a continuous-time markov chain (CTMC) model of Cas9 cutting (see Methods). Users are provided control over on critical components of the technology like the number of Cas9 targets, the indel-outcome distribution, and the mutation rate.

To illustrate how the simulation engine can be used to explore new lineage tracing technologies, we simulated paired tree topologies and data while modulating the number of Cas9 target sites, the missing data rates, the Cas9 mutation rate, and the number of states per site (**Figure 3.2**). We first evaluated datasets in terms of the theoretical tree reconstruction accuracy (reflecting an upper bound on reconstruction performance; **Figure 3.2A**). Because relationships are only discernible if an informative mutation occurs along an edge, we can assess this theoretical performance by “collaps-

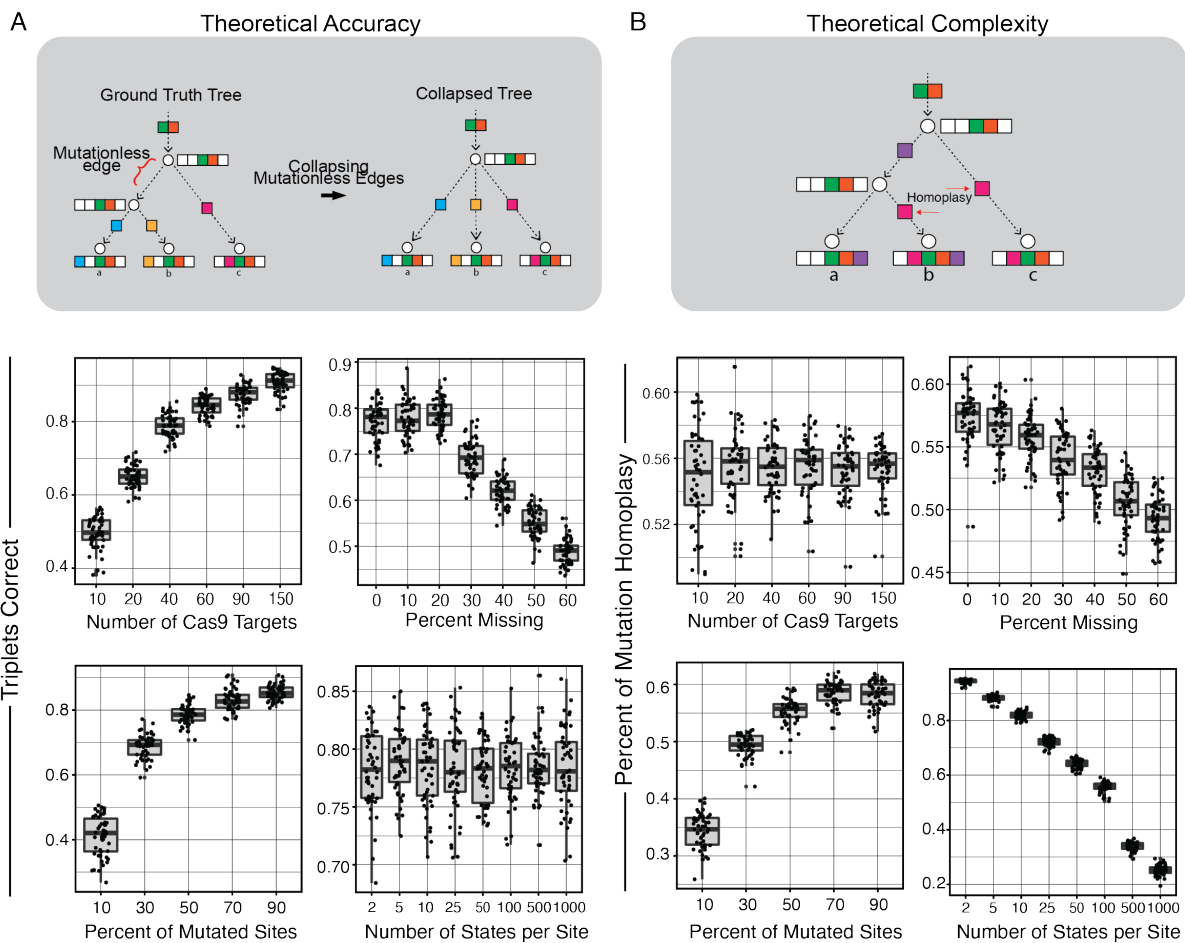


Figure 3.2: **Experimenting with lineage tracing design regimes.** Lineage tracing technologies can be defined by the set of parameters used for tracing lineages and assessed in terms of their theoretical reconstruction accuracy (A) as well as the complexity of the inference, as measured by homoplasmy (B). Each panel represents the modulation of a specific parameter, while holding the remaining parameters fixed. If a parameter is not being tested, it remains at the default: 40 Cas9 target sites, 20% missing data, 50% mutated sites, and 100 states per site.

ing" non-mutation-bearing edges and computing the resulting Triplets Correct (see Methods). This analysis revealed characteristic relationships between tracing parameters and recording capacity - for example, as expected from previous work [273], increasing the number of characters improved the theoretical accuracy.

Using the same dataset, we also evaluated the theoretical complexity of the inference problem by reporting the homoplasy burden in each tree (**Figure 3.2B**). Homoplasy is the fundamental source of ambiguity in phylogenetic inference: specifically, combinatorial algorithms are required to deduce if two samples share the same mutation because they descended from a common ancestor or if two separate lineages accrued the same mutation independently. Thus, using this measure, we can assess how difficult the inference will be for an algorithm. In doing so, we find parameters that can be tuned to reduce complexity: for instance, increasing the number of possible states reduces homoplasy substantially.

Importantly, we observe that the theoretical accuracy and complexity are not directly related to one another. While some parameters, like the mutation rate, have correlated relationships (namely, improving theoretical accuracy at the expense of complexity), others behave in less intuitively. For example, increasing the number of Cas9 targets does not improve complexity but it has a noticeable effect on theoretical performance. Together, these findings prompted us to explore how these characteristic relationships influenced algorithmic performance.

3.2.3 Implementing and benchmarking algorithms in a unified framework

Over the past century, several algorithms have been introduced to deduce phylogenetic relationships from character matrices. With the breadth of algorithms, there are several trade-offs with each approach. Generally, while there are precise approaches for solving for an optimal tree, these are computationally intractable on large inputs; however, more efficient algorithms necessarily use imperfect heuristics that can reduce performance. It is thus in the interest of a user to have several algorithms at their disposal so that they might choose the most relevant algorithm for inference.

To address this, Cassiopeia offers several common algorithms via a shared interface for tree reconstruction with the *TreeSolver* class. Users can choose from Cassiopeia algorithms - like Cassiopeia-Greedy or Cassiopeia-ILP [127] - or more traditional algorithms like Neighbor-Joining [216] or UPGMA [234]. Because of the shared interface, it is simple to extend the codebase by adding additional algorithms - for example, we have adapted algorithms from the literature like the Spectral Neighbor Joining [122] approach. Another advantage of this class-based design is the modularization of these approaches: for example, we can generalize the Cassiopeia-Greedy algorithm to use different criteria by which to cluster the cells.

With this modular framework, we tested the hypothesized that incorporating more sophisticated heuristics for clustering cells with Cassiopeia-Greedy would improve performance. Specifically, we modified the initial Cassiopeia-Greedy algorithm to incorporate a hill-climbing procedure to optimize a Maxcut criterion and thereby improve the initial Greedy split [233] (hereafter called "Maxcut Greedy"; see Methods). Using the tools in the Cassiopeia-benchmark module, we compared Maxcut

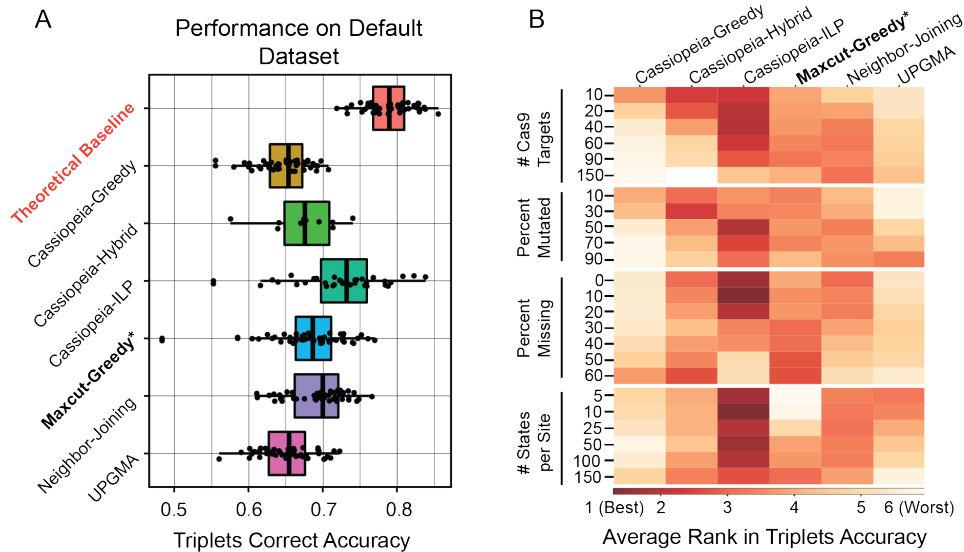


Figure 3.3: **Benchmarking algorithms on synthetic data.** Algorithms implemented in the Cassiopeia2.0 are compared in their ability to infer triplet relationships. (A) Triplet accuracy is assessed on a default benchmarking dataset for a set of algorithms against the theoretical baseline (red). Default parameters are (namely, 40 Cas9 Targets, 20% missing data rate, 100 states per site, and a mutation rate to achieve approximately 50% of mutated sites). (B) A summary of benchmarking performance across all parameter regimes, where average rank of algorithms in Triplets Accuracy is reported for each parameter combination.

Greedy's performance to a panel of existing algorithms across lineage tracing regimes (**Figure 3.3**).

To begin, we designated a set of reasonable default parameters informed by our previous applied work and simulated data on top of topologies with 400 leaves. As expected, we found that none of the algorithms in the panel improved on the theoretical benchmark and Cassiopeia-ILP appeared to provide the most accurate topologies (**Figure 3.3A**). However, we did observe that the Maxcut Greedy algorithm improved on the original Cassiopeia-Greedy algorithm and was competitive with both Cassiopeia-Hybrid and Neighbor-Joining.

We next assessed the performance of Maxcut Greedy compared to the panel of other algorithms across the lineage tracing design regimes. To succinctly report the comparative performance in each regime, we computed the average rank of each algorithm across independent replicates in each

parameter setting (**Figure 3.3B**). While Cassiopeia-ILP consistently outperforms all algorithms across regimes, and the Maxcut Greedy algorithm tends to outperform Cassiopeia-Greedy in all regimes, we observed that there were regime-specific performances of interest. For example, Cassiopeia-Greedy performs comparatively well with few characters or high dropout; Neighbor-Joining performs well with little dropout or few states. These findings suggest that by studying these patterns, one could determine the best algorithm for each individual reconstruction task.

3.3 Discussion

In this study, we have introduced an extended Cassiopeia2.0 codebase for single-cell phylogenetic analysis. Cassiopeia2.0 contains extends the modules originally described in our previous work [127] to include key modules for more sophisticated simulation, benchmarking, and post-reconstruction analysis. Collectively, these improved modules provided insights into lineage tracing technology engineering and analysis.

Using our simulation engine to test the theoretical accuracy and complexity of different lineage tracing regimes, we were able to learn which parameters would likely provide an empirical boost in performance. Among other observations, our findings that a major contributor to performance is the number of Cas9 targets complements our previous work deriving bounds on reconstruction accuracy for our algorithms [273]. Going forward, we believe that the flexibility of this simulation framework will continue to serve experimentalists in two ways: first, by highlighting which parameters can be modified to enable downstream inference. Secondly, this simulation engine will be useful in

assessing the advantages of entirely new technologies - for example those leveraging base editors [149], recombinases [43], or other methodology.

In addition to our analysis of the theoretical difficulty of each problem with our simulation engine, our benchmarking of algorithms highlighted the advantages of our object-oriented paradigm and key parameter-specific instances of comparative algorithmic performance. From a developer perspective, we believe that the object-oriented-programming paradigm used here supports modularity of algorithm design which is showcased in the implementation of the Maxcut Greedy algorithm. In principle, any new heuristic for Cassiopeia-Greedy can be used and included in these benchmarks; more broadly, the shared interface allows one to implement a host of algorithms and easily include it in benchmarks. From a user perspective, our benchmarking analysis suggested that no algorithm performs the best across regimes and rather careful attention should be applied to find the best algorithm for the case at hand. This suite of benchmarking tools can be used to formulate such rules as to when certain algorithms are best applied.

These findings motivate several avenues for future investigation and development. First, the finding that specific algorithms do better in specific regimes suggests that guidelines for users would be helpful. Specifically, creating a "recommender" model for suggesting an algorithm for a given reconstruction task based on features of the data would be a key advance. Second, best practices on downstream tasks still are uncertain. One area that should be explored is how to best combine inferences derived from multiple reconstruction algorithms, or across several trees in the same dataset. In the former case, "consensus" algorithms can be useful for assessing the uncertainty in any downstream analysis of a population. In the latter, "batch" learning may prove to provide more

stable estimates of important parameters governing dynamics as long as these dynamics are reproducible across populations. Third, there are many other tools proposed in the phylogenetic literature not currently supported in Cassiopeia2.0 - such as branch length estimation [144] or ancestral state prediction [191]. Future work can focus on implementing these approaches while simultaneously cultivating a community of developers for this open-source software as new methods emerge.

Taken together, we believe that Cassiopeia2.0 will be a useful resource for future algorithm development and analysis alike. As the recent DREAM challenge for lineage reconstruction indicates [92], this field is rapidly growing and we anticipate new algorithms emerging for several tasks related to single-cell phylogenetics. We hope that the study presented here and the tools implemented in Cassiopeia will serve these efforts well.

3.4 Methods

Simulating ground truth tree topologies

To simulate tree topologies, we utilize a generalized birth-death process (BDP) with the option for adding selection, as implemented in the Cassiopeia module *BirthDeathFitnessSimulator*. In the current study, we only consider simulations without selection. In this, the simulation framework is parameterized by distributions governing the waiting time to cell division f_λ and death rate f_μ , and a desired time-length T or number of extant cells N acting as stopping criteria.

The procedure begins with two nodes - a root node r connected to an extant sample, or leaf,

x represented by a tree \mathcal{T} with a single edge between r and x . Waiting times for leaf x until cell division, t_b , and death, t_d , are drawn from the specified distributions. Most commonly, these are exponential functions parameterized by rates λ and μ .

If $t_b < t_d$, a cell division is simulated by adding two children x_1 and x_2 to \mathcal{T} and setting the branch length for the edge $l_{r,x} = t_b$. If $t_d < t_b$, a cell death event occurs and leaf x is removed from the tree. If a cell division occurs, this process is repeated for the children x_1 and x_2 .

The simulation terminates when either the number of leaves meets the specified parameter N or the total time of the tree ($T_m = \max_{l \in \text{leaves}} \sum_{e:(u,v) \in \text{Edges}(\text{path}(r,x))} l_{u,v}$) exceeds the specified parameter T . In either case, an "ultrametric" tree is created by extending the branches leading from the direct parent to each leaf to T_m . Ultimately, this produces a neutrally-evolving tree (i.e., without selection) over a set of leaves L with branch lengths.

For the purposes of our simulations, we used the following parameters:

- $f_\lambda = 0.02 + \exp(11.46)$. This function represents a waiting-time to birth with a lower-bound of 0.02, reflecting the fact that cells cannot instantaneously divide.
- $f_\mu = \exp(15.80)$
- $N = 2,000$

Then, each tree was downsampled to 400 leaves afterwards to introduce stochasticity into the dataset. 50 trees were simulated using this approach.

Simulating lineage tracing data

Lineage tracing data simulation was performed with the *Cas9LineageTracingDataSimulator* class. This class implements a Continuous Time Markov Chain (CTMC) of Cas9-induced mutations over a tree topology \mathcal{T} with branch lengths. By design, this simulation framework supports specification of several parameters important for Cas9-based lineage tracing, namely: the number of Cas9 targets (n), the state distribution ($\mathbf{q} : \{q_1, \dots, q_s\}$ where q_i indicates the probability of indel i arising), the time-dependent mutation rate (λ , acting as the rate parameter to an exponential distribution), and the missing data rates (s indicating the stochastic dropout rate and h indicating the time-dependent probability of heritable loss).

In the simplest instantiation, each character is treated independently and has the opportunity to mutate over a given edge, (u, v) . So to simulate, we perform a depth-first traversal from the root of the tree and consider each edge from the parent u to child v . For each character that is unmutated at u , we introduce a mutation in v with probability $(1 - \exp(-\lambda * l_{u,v}))$. If a mutation is introduced, we draw a state from the state distribution \mathbf{q} . Additionally, if $h > 0$, then we introduce missing data with probability $(1 - \exp(-h * l_{u,v}))$. Finally, for each leaf, we introduce stochastic missing data over each character with probability s .

In more sophisticated instantiations, this simulator can emulate the effect of linkage between Cas9 targets on the same cassette. Specifically, if two targets on the same cassette are mutated in the same cell, we simulate a resection event by introducing missing data at these two sites and all the sites in between. The size of the cassette is a parameter of the simulation, and characters are

only treated independently if the size of each cassette is 1.

In the case of the experiments presented in this study, we determined the time-dependent mutation rate based off a desired proportion of mutated sites in the final character matrix. To do so, we specified this mutation proportion m and used a moment-matching to infer the mutation rate λ that was expected to generate a mutation proportion m :

$$\lambda = -\log(1 - m)/\text{mean}(T)$$

where $\text{mean}(T)$ is the mean time to any leaf in the tree.

To modulate the missing data rate for the experiments described above, we maintained the $s = 0.1$ and fluctuated the heritable rate h to contribute the remaining amount of missing data.

Assessing theoretical accuracy in ground truth trees

To assess the theoretical accuracy in ground truth trees given some lineage tracing data, we first collapse edges that connect two nodes, u and v , that contain the same character states. Importantly, for this edge collapsing, we do not consider characters with missing data in the two nodes. We refer to this collapsed tree as $\tilde{\mathcal{T}}$. Then, to infer the theoretical accuracy, we can compute the Triplets accuracy or Robinson-Foulds distance for $\tilde{\mathcal{T}}$ and \mathcal{T} . In this study, we reported the Triplets Accuracy.

Quantifying homoplasy in ground truth trees

We compute the homoplasy rates in each tree by counting the proportion of mutations that occur more than one time across the tree. To do so, we perform a depth first traversal of tree, maintaining a mapping of (mutation, character) tuples to the number of edges in the tree that report that mutation in that character. Then, the homoplasy rate is the sum of the number of times a each mutation occurred more than once divided by the total number of mutations in the tree. More formally, let M be the mapping of (s_i, x_j) tuples to the frequency of observations on edges (where s_i is state i and x_j is character j). We compute the number of excess mutations as $e' = \sum_{(s_i, x_j) \in M} M[s_i, x_j] - 1$.

Then, the homoplasy rate $H = \frac{e'}{\sum_{(s_i, x_j) \in M} M[s_i, x_j]}$.

Application of Cassiopeia algorithms

The Cassiopeia algorithms were employed with default parameters and without priors. Specifically, we used the following parameters for each algorithm:

Cassiopeia-Greedy: We used the *VanillaGreedySolver* class as implemented in Cassiopeia. Cells with missing data were handled using the default *assign_missing_average* procedure, which assigns the cell to the character split with the greatest similarity overall.

Cassiopeia-ILP: Cassiopeia-ILP was employed using default parameters: specifically, a maximum neighborhood size of 10,000; a maximum time to converge of 12,600s (3.5hrs); and a maximum potential graph LCA distance of 20.

Cassiopeia-Hybrid: We used an instantiation of Cassiopeia-Hybrid with the *HybridSolver* class

as originally described [127]. Specifically, the *top_solver* was the *VanillaGreedySolver* and the *bottom_solver* was the *ILPSolver*. We used the lca-based threshold for transitioning between the top- and bottom-solvers, with a threshold of 20.

Application of distance-based algorithms

The Neighbor-Joining [216] and UPGMA [234] algorithms used in this study were parameterized with a modified hamming distance function to compute the distances between cells. As previously described [203, 289], we compute the weighted hamming distance $\delta(u, v)$ between two cells by summing the per-character weighted-dissimilarity, $d_i(u, v)$:

$$d_i(u, v) = \begin{cases} 0 & \text{If } u_i == v_i \text{ or } (u_i == -1 \text{ or } v_i == -1) \\ 1 & \text{If } u_i \neq v_i \text{ and } (u_i == 0 \text{ or } v_i == 0) \\ 2 & \text{otherwise} \end{cases}$$

where u_i is the character state at character i in node u and -1 indicates missing data.

Using this dissimilarity function, we solved for trees using the *NeighborJoiningSolver* and *UPGMASolver* classes as implemented in *Cassiopeia*. For the *NeighborJoiningSolver*, we handled rooting the tree by specifying a new sample in the matrix corresponding to the uncut progenitor (all states equal to 0) and rooted at this sample.

Implementation of Maxcut Greedy

The Maxcut Greedy algorithm was implemented as a subclass of the *GreedySolver* class by extending the procedure for finding “splits”, or clusters, of cells. The *VanillaGreedySolver* procedure, as described previously [127], uses the frequency of states to deduce which character-state best separates the data. To build on this, we extracted the concept of optimizing the Maxcut criterion from a previous study [233] with a hill-climbing procedure from an initial proposed split.

Before describing the maxcut criterion, we define briefly a triplet: a triplet $(u, v|x)$ is a set of three nodes where u and v are considered the “in-group” and x the “out-group”. For a given character, we can derive the triplet relationship – for example, if u and v share the same state, but x contains a different state, this would produce the triplet $(u, v|x)$.

To compute the Maxcut criterion, we leverage the idea of a “super tree” over all induced triplets. Such a supertree is a tree structure that satisfies *all* triplets induced by each of the observed characters. Unfortunately, determining even if such a tree exists is an NP-Hard task [241]. Fortunately, previous work has proposed a heuristic-based solution with more tractable complexity [233]. This procedure begins by first inferring a “connectivity graph” over each pair in N samples where edges between samples u and v are weighted by the number of triplets that separate the two samples u and v minus the number of characters that group them together. More specifically, let $f_{(i,j)}$ be the number of samples that report state i in character x_j . We can define a per-character score with respect to two nodes u and v as follows:

$$m_j(u, v) = \begin{cases} -3(N - f_{(u_j, j)} - f_{(-1, x_j)}) & \text{If } u_i == v_i \\ f_{(v_j, j)} - 1 & \text{If } u_i \neq v_i \text{ and } u_j == 0 \\ f_{(u_j, j)} - 1 & \text{If } u_i \neq v_i \text{ and } v_j == 0 \\ f_{(u_j, j)} + f_{(v_j, j)} - 2 & \text{otherwise} \end{cases}$$

The final weight for an edge between samples u and v then is $\sum_j m_j(u, v)$. The original authors proposed using heuristic-based optimization approach to find a heaviest cut across edges in the resulting connectivity graph. This is performed by first offering a random cut, and then iteratively testing the weight of a new proposed cut by moving each sample across the initial cut. This procedure of proposing new cuts by moving each sample across the cut is performed several times until convergence.

The Maxcut Greedy algorithm's split-finding procedure uses this heuristic-based optimization procedure, but instead of initializing with a random cut, it begins by proposing a split with the original *VanillaGreedySolver* approach. The final algorithm is implemented as the *MaxCutGreedySolver* class in Cassiopeia.

Assessing accuracy of reconstructed phylogenies

To assess the Triplets Accuracy of the reconstructed tree, we first imputed the character states for every ancestral node in a tree using the observed states at the leaves and Camin-Sokal parsimony rules. Specifically, we can deduce the character state for a character i for the LCA p of any group of

leaves $\{l_1, \dots, l_n\}$ as so:

$$s_i(p) = \begin{cases} s_i(l_1) & \text{if } l_1 = \dots = l_n \\ 0 & \text{otherwise} \end{cases}$$

We can repeat this procedure for every character in every ancestral node, where the leaves considered are the leaves in the subtree rooted at the ancestral node.

Once each character is inferred for every node, we can identify edges that separate nodes u and v with identical character states and remove them, connecting $parent(u)$ to v . This process is repeated until every edge in the tree separates nodes with different character states. This collapsed tree is referred to as $\tilde{\mathcal{T}}_R$. Then, the Triplets Accuracy was calculated between the ground truth topology \mathcal{T} and the reconstructed, collapsed tree $\tilde{\mathcal{T}}_R$.

Part II

***In vivo* Lineage Tracing in Cancer Models**

Chapter 4

**Single-cell lineages reveal the rates, routes,
and drivers of metastasis in cancer
xenografts**

4.1 Abstract

Detailed phylogenies of tumor populations can recount the history and chronology of critical events during cancer progression, such as metastatic dissemination. We applied a Cas9-based, single-cell lineage tracer to study the rates, routes, and drivers of metastasis in a lung cancer xenograft mouse model. We report deeply resolved phylogenies for tens of thousands of cancer cells traced over months of growth and dissemination. This revealed stark heterogeneity in metastatic capacity, arising from preexisting and heritable differences in gene expression. We demonstrate that these identified genes can drive invasiveness and uncovered an unanticipated suppressive role for *KRT17*. We also show that metastases disseminated via multidirectional tissue routes and complex seeding topologies. Overall, we demonstrate the power of tracing cancer progression at subclonal resolution and vast scale.

4.2 Introduction

Cancer progression is governed by evolutionary principles [reviewed in [263]], which leave clear phylogenetic signatures on every step of this process [188, 260], from early acquisition of oncogenic mutations [i.e., the relationships between normal and malignantly transformed cells [245]] to metastatic colonization of distant tissues [i.e., the relationship between a primary tumor and metastases [26]] and finally to adaptation to therapeutic challenges [i.e., the relationship between drug-sensitive or -resistant populations [19]]. Metastasis is a particularly critical step in cancer progression to study because it is chiefly responsible for cancer-related mortality [35]. Yet, because metastatic events are

intrinsically rare, transient, and stochastic [152, 171], they have proved challenging to monitor in real time. Analogous to the cell fate maps that have deepened our understanding of organismal development and cell-type differentiation [248, 100], accurately reconstructed phylogenetic trees of tumors and metastases can reveal key features of this process, such as the clonality, timing, frequency, origins, and destinations of metastatic seeding [21].

Lineage-tracing techniques allow one to map the genealogy of related cells, providing a crucial tool for exploring the phylogenetic principles of biological processes such as cancer progression and metastasis. Classical lineage tracing strategies can infer tumor ancestry from the pattern of shared sequence variations across tumor subpopulations (e.g., naturally occurring mutations, such as single-nucleotide polymorphisms or copy-number variations) [285, 223]. These “retrospective” tracing approaches are particularly valuable for studying the subclonal dynamics of cancer in patient-derived samples, such as elucidating which mutations contribute to metastasis and when they occur [123, 116, 88, 227]. However, the resolution of these approaches is limited by the number of distinguishing natural mutations, and the conclusions can be confounded by incomplete or impure bulk tumor sampling [113], sequencing artifacts [208], varying levels of intratumor heterogeneity, and non-neutral mutations [263, 26]. Alternatively, so-called “prospective” lineage-tracing approaches—wherein cells are marked with a static label, such as a genetic barcode or fluorescent tag—can measure gross population dynamics at clonal resolution [284] but cannot resolve important and fine subclonal features of cancer biology, such as evolution and the rate, order, and directionality of metastatic events.

The recent development of Cas9-enabled lineage-tracing techniques with single-cell RNA se-

quencing (scRNA-seq) readouts [206, 37, 79, 4] provides the potential to explore cancer progression at vastly larger scales and finer resolution than has been previously possible with classical prospective or retrospective tracing approaches. These methods rely on similar technical principles [reviewed in [15, 269]]. Briefly, Cas9 cuts a defined genomic locus (hereafter “target site”), resulting in a stable insertion or deletion (indel) “allele” that is inherited over subsequent generations; as the cells divide, they accrue more Cas9-induced indels at additional sites that further distinguish successive clades of cells (**Figure 4.1A** and **Figure 4.8**). At the end of the lineage-tracing experiment, the indel alleles are collected from individual cells by sequencing and paired with single-cell expression profiles of the cell state [206, 37]. Then, as in retrospective tracing approaches, computational approaches [216, 29, 247, 70, 127, 293] can reconstruct a phylogenetic tree that best models subclonal cellular relationships (e.g., by maximum parsimony) from the observed shared or distinguishing alleles. Thus far, Cas9-enabled tracing has been successfully applied to study the cellular progenitor landscape in early mammalian embryogenesis [37, 134], hematopoiesis [24], and neural development in zebrafish [206]. Additionally, resources now exist for studying other phylogenetic processes in murine models [37, 134], and analytical tools are available for computationally reconstructing and benchmarking trees from large lineage-tracing datasets [127, 178].

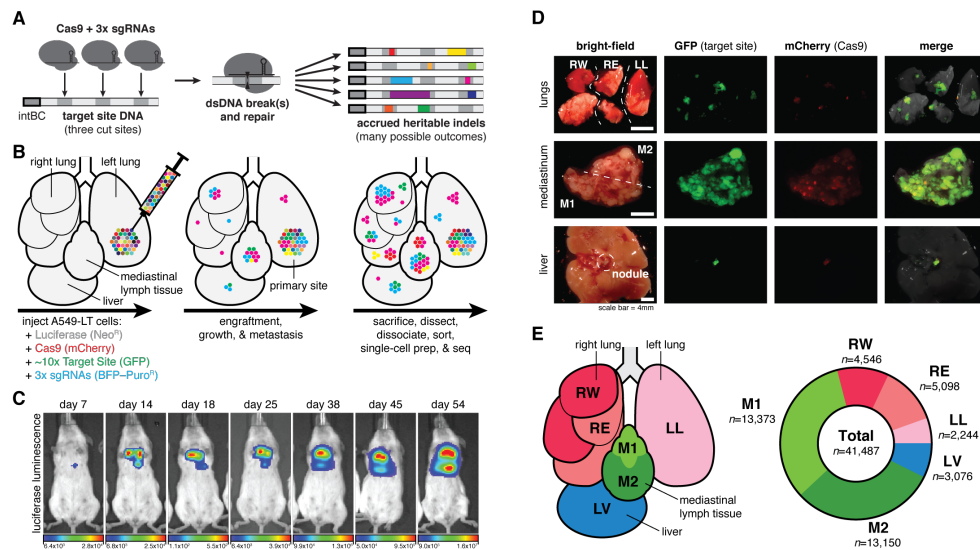


Figure 4.1: Lineage tracing in a lung cancer xenograft model in mice. (A) Our Cas9-enabled lineage-tracing technology. Cas9 and three sgRNAs bind and cut cognate sequences on genomically integrated target sites, resulting in diverse indel outcomes (multicolored rectangles), which act as heritable markers of lineage. dsDNA, double-strand DNA. (B) Xenograft model of lung cancer metastasis. About 5000 A549-LT cells were surgically implanted into the left lung of immunodeficient mice. The cells engrafted at the primary site, proliferated, and metastasized within the five lung lobes, mediastinal lymph, and liver. GFP, green fluorescent protein; BFP, blue fluorescent protein. (C) In vivo bioluminescence imaging of tumor progression over 54 days of lineage recording, from early engraftment to widespread growth and metastasis. (D) Fluorescent imaging of collected tumorous tissues. The white dashed lines indicate bulked tissue samples. (E) Anatomical representation of the six tumorous tissue samples (left) and the number of cells collected with paired single-cell transcriptional and lineage datasets (right). LL, left lung; LV, liver; M1, mediastinum 1; M2, mediastinum 2; RE, right lung E; RW, right lung W.

4.3 Results

4.3.1 Tracing metastasis in a mouse xenograft model

Here, we apply lineage tracing to explore the subclonal dynamics of metastatic dissemination in a mouse cancer model [189]. We used a human KRAS-mutant lung adenocarcinoma line (A549 cells) in an orthotopic xenograft model in mice because this system is characterized by aggressive metastases [189] and orthotopic xenografting experiments are useful for modeling cancer progression in vivo [77]. We engineered A549 cells with a refined version of our molecular recorder technology [37] [Figure 4.9 and Methods]. Specifically, the engineered cells contained (i) luciferase for live imag-

ing; (ii) Cas9 for generating heritable indels; (iii) 10 uniquely barcoded copies of the target site for recording lineage information, which can be captured as expressed transcripts by scRNA-seq; and (iv) triple–single guide RNAs (sgRNAs) to direct Cas9 to the target sites, thereby initiating lineage recording (**Figure 4.1A** and **Figure 4.92, A to C**). To enable tracing over long time scales, we designed the sgRNAs with nucleotide mismatches to the target sites, thereby decreasing their affinity [25, 129] and tuning the lineage recording rate [37, 131]. About 5000 engineered cells (“A549-LT”) were then embedded in matrigel and surgically implanted into the left lung of an immunodeficient (C.B-17 SCID) mouse (**Figure 4.1B**). We followed bulk tumor progression by live luciferase-based imaging (**Figure 4.1C**): The early bioluminescent signal was modest and restricted to the primary site (left lung), consistent with engraftment; with time, the signal progressively increased and spread throughout the thoracic cavity, indicating tumor growth and metastasis. After 54 days, the mouse was sacrificed and tumors were identified in the five lung lobes, throughout the mediastinal lymph tissue, and on the liver (**Figure 4.1D**), in a pattern consistent with this model [189]. From these tumorous tissues, we collected six samples, including one from the left lung (i.e., including the primary site; **Figure 4.1E**, left). The tumor samples were dissociated, fluorescence-sorted to exclude normal mouse cells, and finally processed for scRNA-seq. To simultaneously measure the transcriptional states and phylogenetic relationships of the cells, we prepared separate RNA expression and target site amplicon libraries, respectively, resulting in 41,487 paired single-cell profiles from six tissue samples [**Figure 4.1E**, right; **Figure 4.10**; and Methods].

In addition to the mouse described above (hereafter “M5k”), we also performed lineage tracing in three other mice (called “M10k,” “M100k,” and “M30k”), using A549-LT cells engineered with slightly

different versions of the lineage-tracing technology [**Figure 4.11** and Methods]. Unless otherwise noted, we focus our primary discussion of the results on mouse M5k because it yielded the richest lineage-tracing dataset with the most cells and distinct lineages.

4.3.2 Distinguishing clonal cancer populations

Our lineage recorder target site [37] carries two orthogonal units of lineage information: (i) a static 14–base pair–randomer barcode (“intBC”) that is unique and distinguishes between multiple integrated target site copies within each cell and (ii) three independently evolving Cas9 cut-sites per target site that record heritable indel alleles and are used for subclonal tree reconstruction (**Figure 4.1A**). Each target site is expressed from a constitutive promoter, allowing it to be captured by scRNA-seq. After amplifying and sequencing the target site mRNAs, the reads were analyzed using the Cassiopeia processing pipeline [127]. Briefly, this pipeline leverages unique molecular identifier (UMI) information and redundancy in sequencing reads to confidently call intBCs and indel alleles from the lineage data, which inform subsequent phylogenetic reconstruction [**Figure 4.8** and Methods].

We determined the number of clonal populations (i.e., groups of related cells that descended from a single clonogen at the beginning of the xenograft experiment) that are each associated with a set of intBCs. Importantly, the A549-LT cells were prepared such that clones carry distinct intBC sets. By sampling the A549-LT cells before implantation, we estimate that the implanted pool of 5000 cells initially contained 2150 distinguishable clones (**Figure 4.9D**). From these intBC sets, we assigned

most of the cancer cells collected from the mouse (97.7%) to 100 clonal populations (**Figure 4.12, A and B**), ranging in size from >11,000 (clone 1, "CP001") to ~30 cells (CP100) (**Figure 4.12C**). Though there were some smaller clonal populations, we focused on these largest 100 because lineage tracing in small populations is less informative. Furthermore, despite initially implanting ~2150 distinct clones, only 100 clones successfully engrafted and proliferated, suggesting that only a small minority of cells were competent for engraftment and survival in vivo (**Figure 4.9D**). Moreover, we find minimal correlation between initial (preimplantation) and final (postsacrifice) clonal population size [Spearman's correlation coefficient (ρ) = -0.026 ; **Figure 4.9E**], suggesting that clone-intrinsic characteristics that confer greater fitness in vitro do not necessarily confer greater fitness in the in vivo environment [176, 105].

Features that influence the lineage recording capacity and tree reconstructability differed between clonal populations, such as the copy number of target sites, the percentage of recording sites bearing indel alleles, and allele diversity (**Figures 4.13, A to C, and 4.14**). Although most clonal populations exceeded parametric standards for confident phylogenetic reconstruction, some had slow recording kinetics or low allele diversity and failed to pass quality-control filters (17 clones, 7.3% of total cells in mouse M5k; **Figures 4.13D and 4.14B**); these clones were excluded from tree reconstruction and downstream analyses (see Methods).

We observed that the clonal populations exhibited distinct distributions across the six tissues (**Figure 4.15, A to C**), ranging from exclusively residing in the primary site (e.g., CP029, CP046) to being overrepresented in a tissue (e.g., CP003, CP020) or to being distributed broadly over all sampled tissues (e.g., CP002, CP013). The level of tissue dispersal is logically a consequence

of metastatic dissemination and thus can inform on the frequency of past metastatic events. To quantify the relationship between tissue distribution and metastatic dissemination, we defined a statistical measure of the observed-versus-expected tissue distributions of cells [termed "tissue dispersion score"; Methods] to operate as a coarse, tissue-resolved approximation of the dissemination frequency. Across the 100 clonal populations in this mouse, we observed a wide range of tissue dispersion scores (**Figure 4.15D**), suggesting broad metastatic heterogeneity across the tumor populations. We next explored this heterogeneity more directly and at far greater resolution using the evolving lineage information.

4.3.3 Single-cell–resolved cancer phylogenies

The key advantage of our lineage tracer is not in following clonal lineage dynamics (i.e., from cells' static intBCs, as described above) but rather in reconstructing subclonal lineage dynamics (i.e., from cells' continuously evolving indel alleles). As such, we reconstructed high-resolution phylogenetic trees using the Cassiopeia suite of phylogenetic inference algorithms [127] with parameters tailored to this dataset's complexity and scale (see Methods). Each of the resulting trees comprehensively describes the phylogenetic relationships between all cells within the clonal population and summarizes their history of metastatic dissemination between tissues (**Figure 4.2**). The trees are intricately complex (mean tree depth of 7.25; **Figure 4.13E**) and highly resolved [consisting of 37,888 cells with 33,266 (87.8%) distinct lineage states; **Figure 4.13C**].

To illustrate the intricate complexity of the trees in this dataset, we present the reconstructed phy-

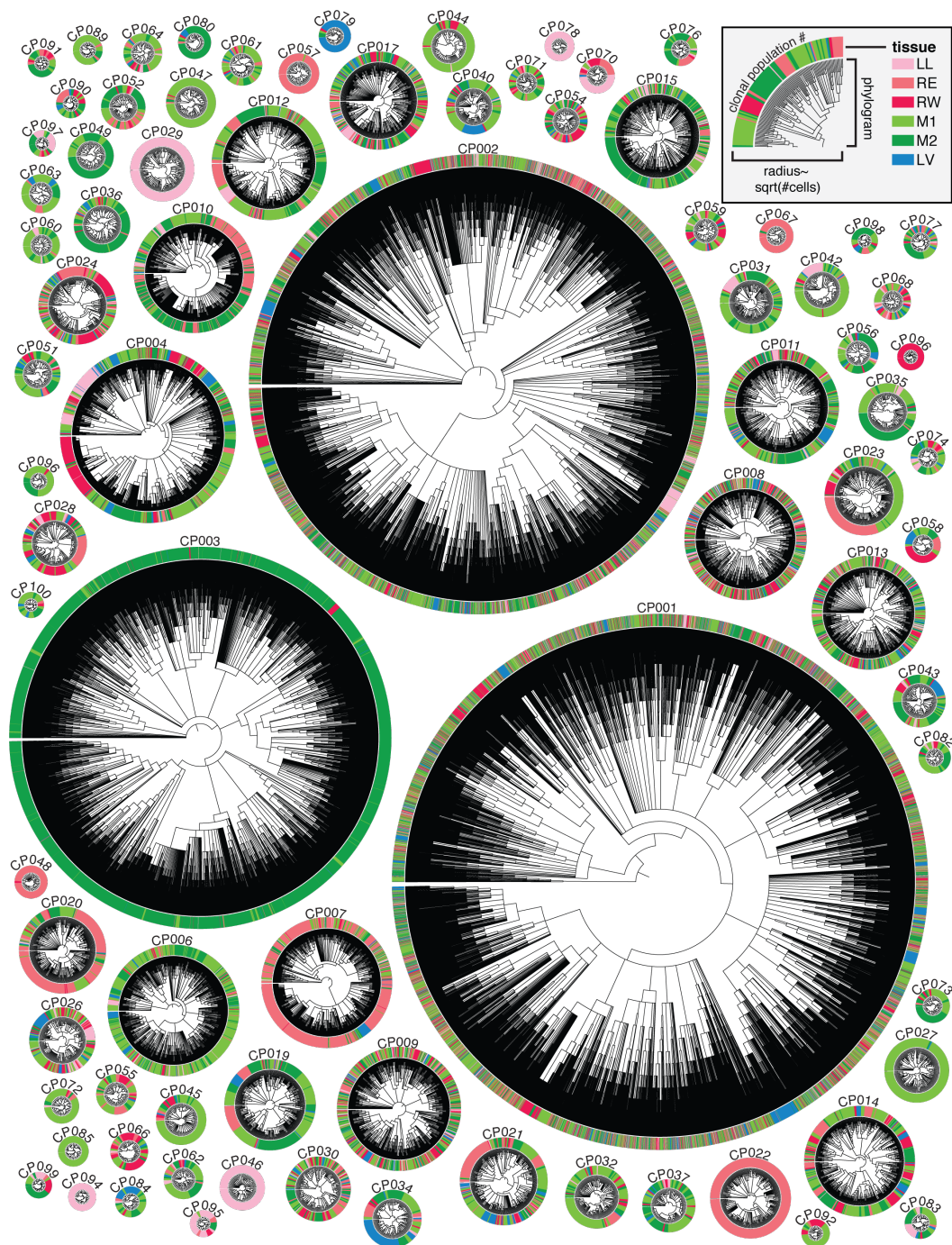


Figure 4.2: High-resolution phylogenetic trees capture the histories of clonal cancer populations. Highly detailed phylogenetic reconstructions for each clonal population, represented as radial phylograms. Each cell is represented along the circumference and colored by tissue, as in **Figure 4.1E** and the legend. Trees differ in size, tissue distribution, and frequency of tissue transitions. Each tree is scaled by the square root ($\sqrt{\text{#cells}}$) of the number of cells.

logram and lineage alleles for a representative clonal population of 5616 cells (CP003; **Figure 4.3A**) with 99.0% (5560) distinct cell lineage states, mean tree depth of 10.0, and maximum tree depth of 20. Intuitively, cells that are more closely related to one another ought to share more lineage alleles, which is evident from the patterns of shared alleles within clades and distinguishing alleles between clades (zoomed inlays in **Figure 4.3A**). Indeed, we find systematic agreement between phylogenetic distance (i.e., the distance between two cells in the tree) and allelic distance (the difference between two cells' lineage alleles) for this example (**Figure 4.3B**) and across all other trees (**Figure 4.17**). The high diversity of distinguishable Cas9-induced indels (9936 unique alleles across all M5k cells; evident in the array of distinct allele colors in **Figure 4.3A**) also reduces the probability of homoplasy, an issue that complicates tree reconstruction and impairs tree accuracy ([127, 218]). Altogether, these features indicate that the reconstructed trees accurately model the true phylogenetic relationships between cells.

4.3.4 Inferring and quantifying past metastatic events from phylogenies

A notable feature revealed by the reconstructed phylogenies is the varying extent to which closely related cells reside in different tissues (**Figure 4.2**), patterns that directly result from ancestor cells having physically transited from one tissue to another in the past (i.e., metastatic seeding). Varying rates of metastasis produce different patterns of concordance between phylogeny and tissue (**Figure 4.4A**). For example, nonmetastatic populations result in all clades remaining within a single tissue (**Figure 4.4, A and B**, left); conversely, highly metastatic populations result in closely related

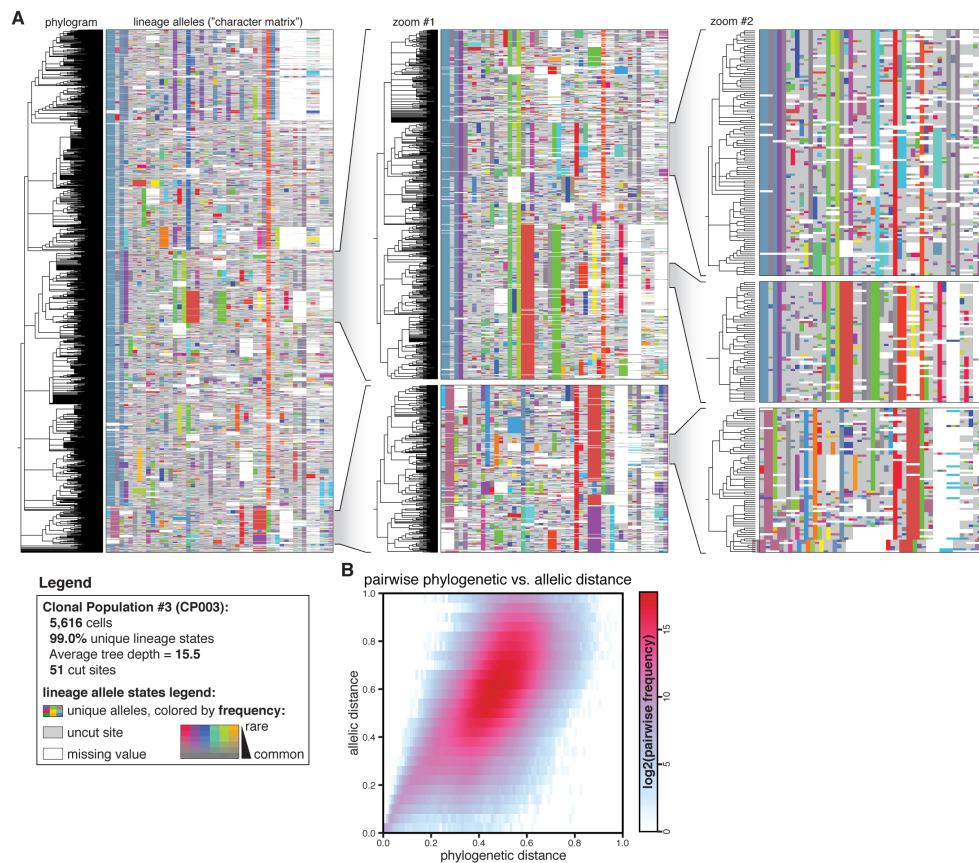


Figure 4.3: **Phylogenetic reconstructions are detailed and accurate.** (A) Phylogenetic tree and lineage alleles of one clonal population (CP003; N = 5616 cells). The phylogram (left) represents cell-cell relationships, and the matrix (right) represents the lineage alleles for each cell. Alleles are assigned distinct colors, where saturation indicates allele rarity (legend). Nested zooms of individual clades [inlays of (A)] show the patterns of shared and distinguishing indel alleles and highlight indel diversity, tree depth, and tree complexity. (B) Correspondence between phylogenetic distance (the normalized pairwise tree distance between two cells) and allelic distance (the normalized pairwise difference in alleles between two cells) for CP003, indicating that the tree accurately models phylogenetic relationships.

cells residing in different tissues (Figure 4.4, A and B, right). Finally, intermediate levels of metastasis can similarly lead to a dispersed tissue distribution as in the highly metastatic regime, though with fewer metastatic transitions, thus supporting the need to reconstruct trees to distinguish such cases (Figure 4.4, A and B, middle).

To quantitatively study the relationship between metastatic phenotype and phylogenetic topology, we used the Fitch-Hartigan maximum parsimony algorithm ([72, 104]). Our implementation

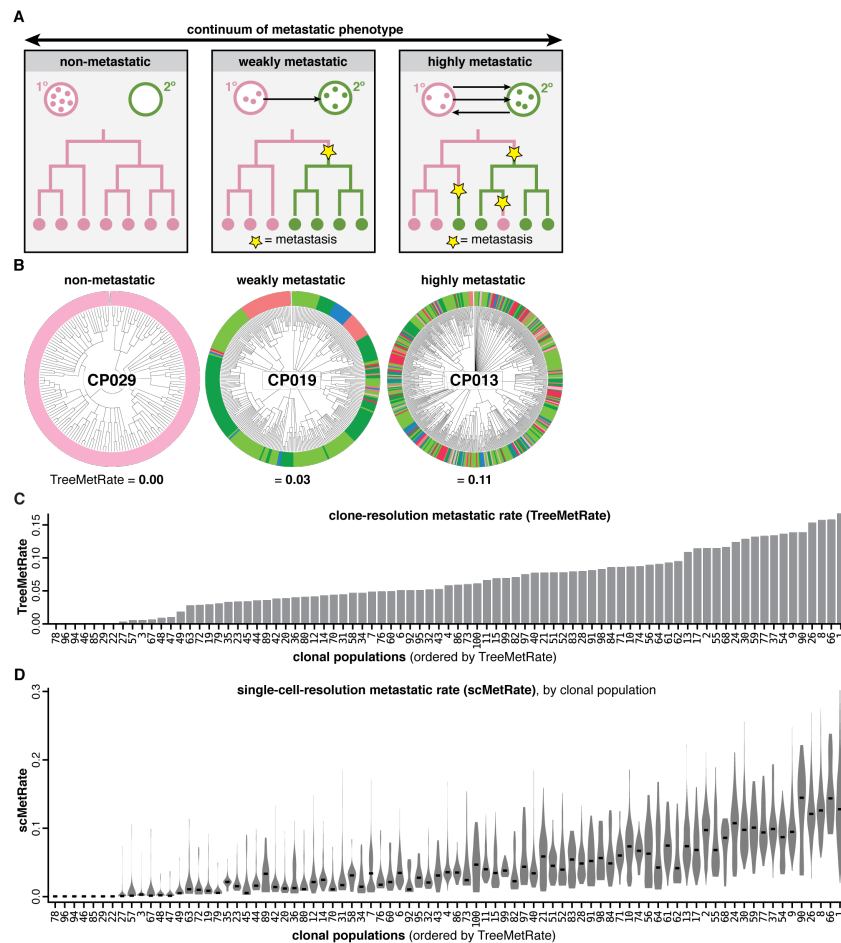


Figure 4.4: Quantifying the diverse metastatic phenotypes of clonal populations directly from cell lineages. (A) Theoretical continuum of metastatic phenotypes, spanning nonmetastatic (never exiting the primary site) to highly metastatic (frequently transitioning between tumors; arrows). Ancestral metastatic events between tissues leave clear phylogenetic signatures (yellow stars). (B) Example clonal populations that illustrate the wide range of metastatic phenotypes observed: a nonmetastatic population that never exits the primary site (CP029), a moderately metastatic population that infrequently transitions between different tissues (CP019), and a frequently metastasizing population with closely related cells residing in different tissues (CP013). Cells are colored by tissue as in **Figure 4.1E**; metastatic phenotypes were scored by the TreeMetRate. (C) The distribution of TreeMetRates for each clonal population. (D) The distributions of single-cell-resolution metastatic phenotypes (scMetRates) for each clonal population, rank ordered by TreeMetRate, with median scMetRate indicated in black.

of this algorithm provides the minimal number of ancestral (i.e., not directly observed) metastatic transitions that are needed to explain the final (i.e., observed) tissue location of each cell in a given tree. We defined a score of the metastatic potential (termed “TreeMetRate”) by dividing the inferred minimal number of metastatic transitions by the total number of possible transitions (i.e., edges in

the tree). Empirically, we observe a distribution of clonal populations that spans the full spectrum of metastatic phenotypes between low (nonmetastatic) and high (very metastatic) TreeMetRates (**Figure 4.4, B and C**). The TreeMetRate is stable across bootstrapping experiments in simulated trees (**Figure 4.16, E and F**) and when using an alternative phylogenetic reconstruction method [neighbor joining [216]] on empirical data (**Figure 4.18A**; Pearson's $\rho = 0.94$), indicating that the TreeMetRate is a robust measurement of metastatic behavior—though, notably, Cassiopeia trees are more parsimonious than those reconstructed by neighbor joining (**Figure 4.18B**). Empirically, the tissue dispersal score agrees with the TreeMetRate at low metastatic rates (**Figure 4.19, A and C**); however, the TreeMetRate more accurately captures the underlying metastatic rate over a broad range of simulated metastatic rates because it can distinguish between moderate and high metastatic rates (**Figure 4.16D**), which both result in broad dispersion across tissues (**Figure 4.4A**), whereas the tissue dispersal score saturates at intermediate metastatic rates (**Figure 4.16B**). Furthermore, the TreeMetRate agrees with an alternate measure that does not depend on tree reconstruction [termed “AlleleMetRate”; see Methods; **Figure 4.19, B and D**], though, again, simulations indicate that the TreeMetRate best reflects the underlying metastatic rate (**Figure 4.16, A to D**).

We further extended our parsimony-based approach to quantify the metastatic phenotype at the resolution of individual cells (termed the “scMetRate”) by averaging the TreeMetRate for all subclades containing a given cell (see Methods). This measurement is sensitive to subclonal differences in metastatic behavior (**Figure 4.4D**) and highlighted intriguing bimodal metastatic behavior for clone CP007 (discussed below). Additionally, we find that the scMetRate is uncorrelated to clonal population size, proliferation signatures [16, 215], or cell cycle stage [278] (**Figure 4.20**), indicating that it

can measure metastatic potential uncoupled from proliferative capacity. Overall, these results indicate that cancer cells in this dataset exhibit diverse metastatic phenotypes both between and within clonal populations, which can be meaningfully distinguished and quantified by virtue of the lineage tracer but would have otherwise been hidden from classical barcoding approaches.

4.3.5 Transcriptional drivers of differences in metastatic phenotype

We next explored the extent to which single-cell transcriptional states underlie metastatic capacity [34]. By comparing the paired transcriptional and lineage datasets, we found that different metastatic behaviors corresponded to differential expression of genes, many with known roles in metastasis. First, after filtering and normalizing the scRNA-seq data, we applied *Vision* [58], a tool for assessing the extent to which the variation in cell-level quantitative phenotypes can be explained by transcriptome-wide variation in gene expression. Although we found little transcriptional effect attributable to clonal population assignment, we observed a modest association between a cell's transcriptional profile and its tissue sample or metastatic rate (**Figure 4.21**). We next performed pairwise differential expression analyses comparing cells from completely nonmetastatic clonal populations (i.e., four clones that never metastasized from the primary tissue in the left lung, such as CP029) to metastatic clones in the same tissue (**Figure 4.22**). This clone-resolution analysis identified several genes with significant expression changes that were also consistent across each nonmetastatic clone [\log_2 fold change >1.5 , false discovery rate <0.01], such as *IFI6*. These initial results suggested that differences in metastatic phenotype may manifest in characteristic differences in gene

expression and motivated deeper analysis.

Next, we sought to comprehensively identify genes that are associated with metastatic behavior by regressing single-cell gene expression against the scMetRates [over all observed cells, clonal populations, and tissues; **Figure 4.5A** and Methods], thereby leveraging both the scRNA-seq dataset and the single-cell phylogenies. Many of the identified positive metastasis-associated candidates [i.e., genes with significantly higher expression in highly metastatic cells; see Methods] have known roles in potentiating tumorigenicity (**Figure 4.5B**, top), such as *IFI27* [271, 161], *REG4* [96, 249], and *TNNT1* [103]. Reciprocally, many negative metastasis-associated candidates have known roles in attenuating metastatic potential (**Figure 4.5B**, bottom), such as *NFKBIA* [27], *ID3* [38], and *ASS1* [205]. The gene whose expression we identified as most strongly and significantly anticorrelated with metastatic capacity, *KRT17* (Methods), has paradoxically been implicated in promoting invasiveness in lung adenocarcinoma [162], and its overexpression has been associated with poor prognosis in some cancers [111]; we follow up on this unexpected finding below. Additionally, many of the identified genes were significantly reproduced across every mouse in this study (**Figure 4.55, C and D**, and **Figure 4.24**) (Methods). And more generally, the gene-level expression trends are broadly supported by significant correlation (Methods) between the TreeMetRate and several gene expression signatures [246], such as cancer invasiveness [7] and epithelial-mesenchymal transition [125] (**Figure 4.23**).

Although we identified many interesting and reproducible gene candidates in our regression analysis, it was unclear whether they were directly driving the metastatic phenotype or were merely associated with it. To address this point, we next explored the functional impact on metastatic be-

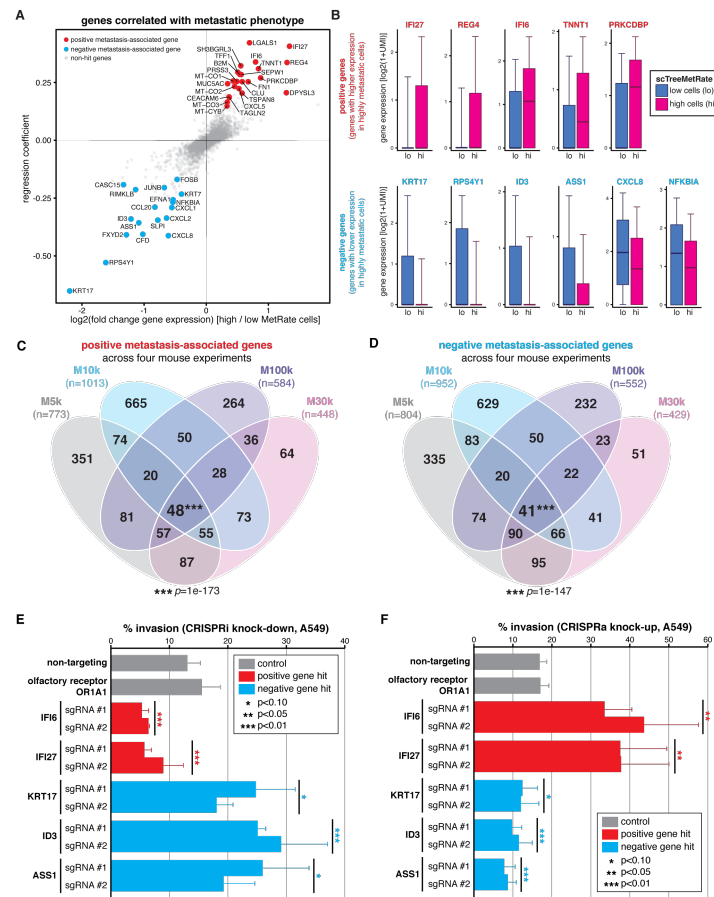


Figure 4.5: Divergent metastatic phenotypes are driven by differences in gene expression. (A) Poisson regression analysis of single-cell gene expression and scMetRate for all cells and all tissues, with fold change and coefficient of regression shown. The strongest and most significant positive and negative genes are annotated (red and blue, respectively; see Methods). (B) Expression level of several positive and negative metastasis-associated gene candidates (top and bottom rows, respectively) in cells with low or high scMetRate (blue and magenta box plots, respectively). Boxes represent first, second, and third quartiles; whiskers represent 9th and 91st percentiles of expression distribution. (C and D) Overlap of identified positive and negative metastasis-associated genes, respectively, from the four mouse experiments, with the number of genes indicated. Four-way intersections between gene sets are significant by SuperExactTest [272] multiset intersection test. (E and F) In vitro transwell invasion assays after CRISPRi or CRISPRa gene perturbation, respectively, in A549 cells. Perturbations were performed using two independent sgRNAs per gene. Differences in invasion phenotype relative to two negative control guides (nontargeting and olfactory receptor) were significant by two-tailed Student's t test; error bars show standard deviation across triplicates.

havior of modulating the expression of five high-scoring gene candidates (*IFI6*, *IFI27*, *KRT17*, *ID3*, and *ASS1*). First, we engineered A549 cells to enable CRISPR-inhibition (CRISPRi) or CRISPR-activation (CRISPRa) perturbations (activity validated in **Figure 4.25, C and D**) and then increased or decreased expression, respectively, of the five gene targets using two independent sgRNAs per

gene. Finally, we measured the perturbed cells' invasion phenotype in vitro using a transwell invasion assay [Figure 4.5, E and F, and Methods]. As hypothesized, CRISPRi knockdown resulted in decreased invasiveness for positive metastasis-associated genes (*IFI6* and *IFI27*; $p = 0.001$ and 0.005 , respectively, by two-tailed Student's t test) and increased invasiveness for negative metastasis-associated genes (*KRT17*, *ID3*, and *ASS1*; $p = 0.054$, 0.003 , and 0.062 , respectively; Figure 4.5E). Conversely, we found that increasing candidate gene expression by CRISPRa produced the exact opposite results (Figure 4.5F), indicating that the invasion phenotype can be quantitatively altered by both increased or decreased expression for each of the five candidate genes tested, including, notably, *KRT17*, in agreement with the results of the lineage tracing experiments. We confirmed that the modulation of expression of each of these genes strongly and significantly modulated invasiveness ($p < 0.01$, by two-tailed Student's t test) in a separate human lung cancer cell line (H1299 cells, which are KRAS wild type and TP53 mutant and harbor endogenous *NRAS*^{Q61K}; Figure 4.25, A and B), though, for two of the genes (*IFI27* and *IFI6*), CRISPRa had a significant effect ($p < 0.01$), whereas CRISPRi did not. Taken together, these results indicate that (i) the lineage tracer can meaningfully identify metastasis-associated genes in vivo, (ii) some of these gene candidates are sufficient to drive differences in metastatic phenotype, and (iii) these genes' roles in mediating invasiveness extend beyond the one A549 cancer model and across different oncogenic backgrounds.

4.3.6 Heterogeneity and heritability of metastatic behavior in preimplantation cells

We next used the positive and negative metastasis-associated genes identified above (**Figure 4.5A**) to define a de novo transcriptional signature (hereafter, “metastasis signature”; **Figure 4.6A** and **Figure 4.26A**). Even before implantation into the mice, the cells already exhibited meaningful heterogeneity in the metastatic signature (**Figure 4.6B**), and metastasis-associated genes such as *ID3* and *TNNT1* were similarly heterogeneously expressed preimplantation (**Figure 4.6C**). Next, we used the lineage barcodes to map cells from the in vitro preimplantation pool to the clonal populations that engrafted in vivo (**Figure 4.26B**). We then segregated these mapped cells into the top and bottom halves by their corresponding TreeMetRate and queried their preimplantation metastatic signatures. We found that cells from more metastatic clones in the mouse had modestly, yet significantly, higher metastatic signatures before implantation, and vice versa (**Figure 4.26C**). This indicates that the preimplantation transcriptional signature is mildly predictive of in vivo metastatic phenotype (**Figure 4.26D**), though the distinction becomes more amplified in vivo (**Figure 4.26, C and D**). This result suggests that even before cells were xenografted into the mouse, they were primed for greater or lesser metastatic capacity in vitro.

Although the preexisting transcriptional heterogeneity in the preimplantation cells was noteworthy, it remained unclear whether these differences were stochastic or intrinsic properties of the cells that could be robustly propagated in vitro and *in vivo*. One way to address this question is by implanting two cells from the same clone into two distinct mice and querying how well their

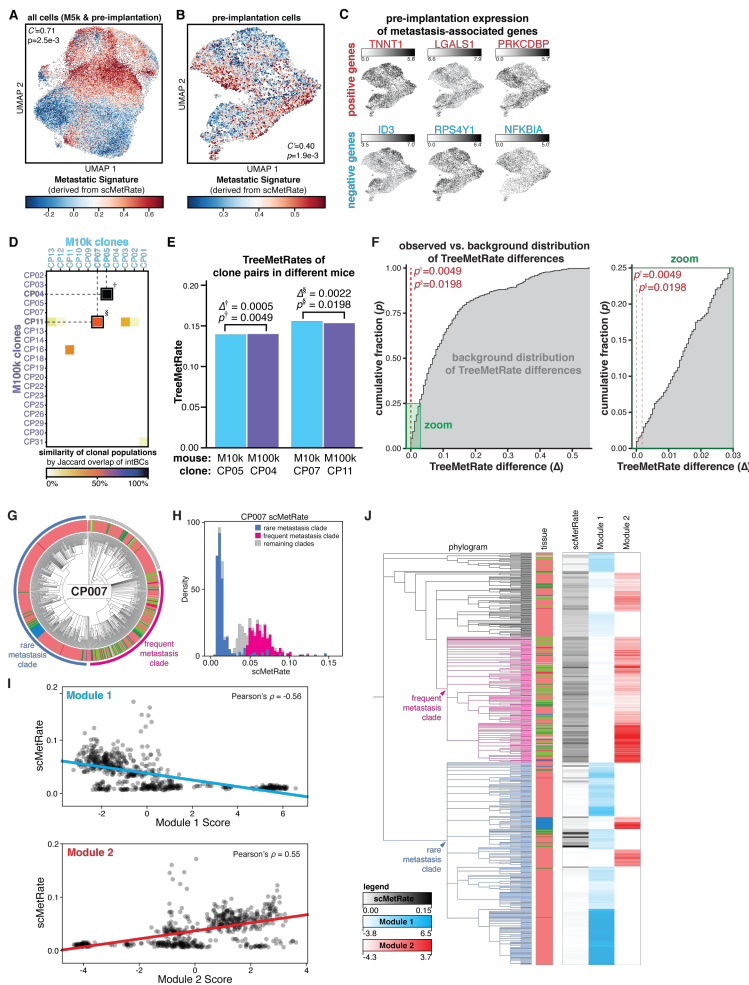


Figure 4.6: Metastatic phenotype is predetermined, heritable, and reproducible. (A and B) Projections of transcriptional states of M5k cancer cells and preimplantation cells (A) or preimplantation cells alone (B), colored by metastatic signature. Association between transcriptional state and metastatic signature is measured by inverted Geary's C^* and significance by false discovery rate (p). (C) Preimplantation cells exhibit heterogeneity in expression of metastasis-associated genes. The color scale bars indicate normalized log gene expression. (D) Jaccard overlap of intBC sets between clonal populations in M10k and M100k mice. Two pairs of clonal populations (indicated by † and §) were related between the two mouse experiments (Jaccard overlap > 50%). (E) Comparison of TreeMetRates from related clones implanted in M10k and M100k, showing minimal difference in metastatic rate (δ) between clone pairs, and the empirical probability (p) that the δ for a randomly selected pair of clones will be smaller than the observed δ . (F) Cumulative distribution plot of the background distribution of all possible pairwise TreeMetRate differences between M10k and M100k clones (gray), with a zoom to show the low- δ regime; red dashes indicate p . (G) Divergent subclonal metastatic behavior exhibited in the phylogenetic tree of clonal population 7, with annotated subclades; cells are colored by tissue as in **Figure 4.1E**. (H) The bimodal distribution of scMetRates for cells in CP007, with cells from the divergent subclades indicated. (I) Comparison of single-cell metastatic phenotype and Hotspot transcriptional module scores. (J) Overlay illustrating concordance between CP007 phylogeny, scMetRates, and Hotspot module scores.

metastatic phenotype is reproduced. Using the cells' intBCs, which stably mark clones, we identified two such instances where cells from the same clonal population seeded tumors in two different mice (**Figure 4.6D** and **Figure 4.27**). Notably, for each of the two pairs of clonal populations, the TreeMetRates were nearly identical (**Figure 4.6E**). Indeed, one of these pairs had the most similar TreeMetRates across all pairs of clones in the two mouse experiments [$\delta(\text{TreeMetRate}) = 0.0005$; **Figure 4.6F**]. Taken together, these results indicate that (i) the diverse metastatic phenotypes *in vivo* are determined before implantation (also **Figure 4.6, B and C**), (ii) the metastatic phenotype is reproducible over generations and is thus heritable (**Figure 4.6, E and F**, and **Figure 4.26, C and D**), and (iii) our analytical approaches for quantifying the metastatic rate, including reconstruction of the phylogenies, are experimentally robust (**Figure 4.6E**).

4.3.7 Evolution of metastatic phenotype

Though we have thus far discussed how metastatic phenotype is clone-intrinsic and stably inherited, we identified a clear example within the dataset that was the exception to this general rule. Specifically, clone 7 (CP007) exhibited distinct subclonal metastatic behaviors, wherein one clade metastasized frequently to other tissues while another clade remained predominantly in the right lung (**Figure 4.6G**). This distinction is reflected in a bimodal distribution of scMetRates (**Figure 4.4D** and **4.6H**). We used the Hotspot [57] algorithm to explore the relationship between subclonal structure and gene expression and identified two modules of correlated genes that exhibit heritable expression programs (**Figure 4.28A**). Notably, the cumulative expression of genes in module 1 is correlated with

lower metastatic rates, whereas the opposite holds for module 2 (**Figure 4.6I** and **Figure 4.28, B and C**). Consistently, the two modules broadly correspond to the two clades with diverging metastatic phenotypes (**Figure 4.6J**). This result is reproduced even in a control analysis of CP007 cells from the right lung only (**Figure 4.28, D to G**), indicating that these differences in gene expression indeed reflect differences in metastatic phenotype rather than tissue-specific effects. This example illustrates that although the metastatic rate is stably inherited, it can also evolve—albeit rarely—within a clonal population, alongside concordant changes in transcriptional signature. Importantly, this finding could only be appreciated by virtue of the subclonal resolution of the lineage tracer.

4.3.8 Tissue routes and topologies of metastasis

The phylogenetic reconstructions also made it possible to describe detailed histories about the tissue routes and the directionality of metastatic seeding. For example, the phylogenetic tree for CP095 reveals five distinct metastatic events from the left lung to different tissues, in a paradigmatic example of simple primary seeding (**Figure 4.7, A and B**). Other phylogenies revealed more complicated trajectories, such as CP019, wherein early primary seeding to the mediastinum was likely followed by intramediastinal transitions and later seeding from the mediastinum to the liver and right lung (**Figure 4.7, C and D**). To more systematically characterize the tissue transition routes revealed by the phylogenetic trees, we extended the Fitch-Hartigan algorithm ([72, 104]) to infer the directionality of each tissue transition (i.e., the origin and destination of each metastatic event) along a clonal population's ancestry. Our algorithm, called FitchCount, builds on other ancestral inference algorithms

such as MACHINA [135] by scaling to large inputs and providing tissue transition frequencies that are aggregated across all ancestries that satisfy the maximum parsimony criterion (see Methods and Appendix). Through simulation, we show that FitchCount can accurately recover underlying transition probabilities better than a naïve application of the Fitch-Hartigan algorithm [Figure 4.16, G and H, and Methods], likely because the naïve approach summarizes only a single optimal assignment solution, whereas FitchCount summarizes all optimal solutions. The resulting conditional probabilities of metastasis to and from each tissue are summarized in a tissue transition probability matrix (Figure 4.7, E and F). Notably, we found that these transition matrices are varied and distinct to each clone (Figure 4.7G and Figure 4.29).

We next used principal components analysis (PCA) to stratify clones by their transition matrices (Figure 4.7H) and identified descriptive features that capture differences in the metastatic tissue routes traversed by each clone (Figure 4.7I and Figure 4.30). These descriptive features include primary seeding from the left lung (as in CP095; Figure 4.7, A and B), metastasis from and within the mediastinum (CP098; Figure 4.7G, left), or metastasis between lung lobes (CP070; Figure 4.7G, middle) and may reflect intrinsic differences in tissue tropism. From this feature analysis, we also note that many clones primarily metastasized via the mediastinal lymph tissue (Figure 4.7, H and I), suggesting that the mediastinum may act as a nexus for seeding in this mouse model, perhaps because the mediastinal lymph is a favorable niche with extensive tissue connections [196]. This observation is consistent with previous experiments in this model [189], bulk live imaging during tumor progression in this experiment wherein tumors appear to quickly colonize the mediastinum (Figure 4.1C), and the terminal disease state wherein the mediastinum harbors the majority of the

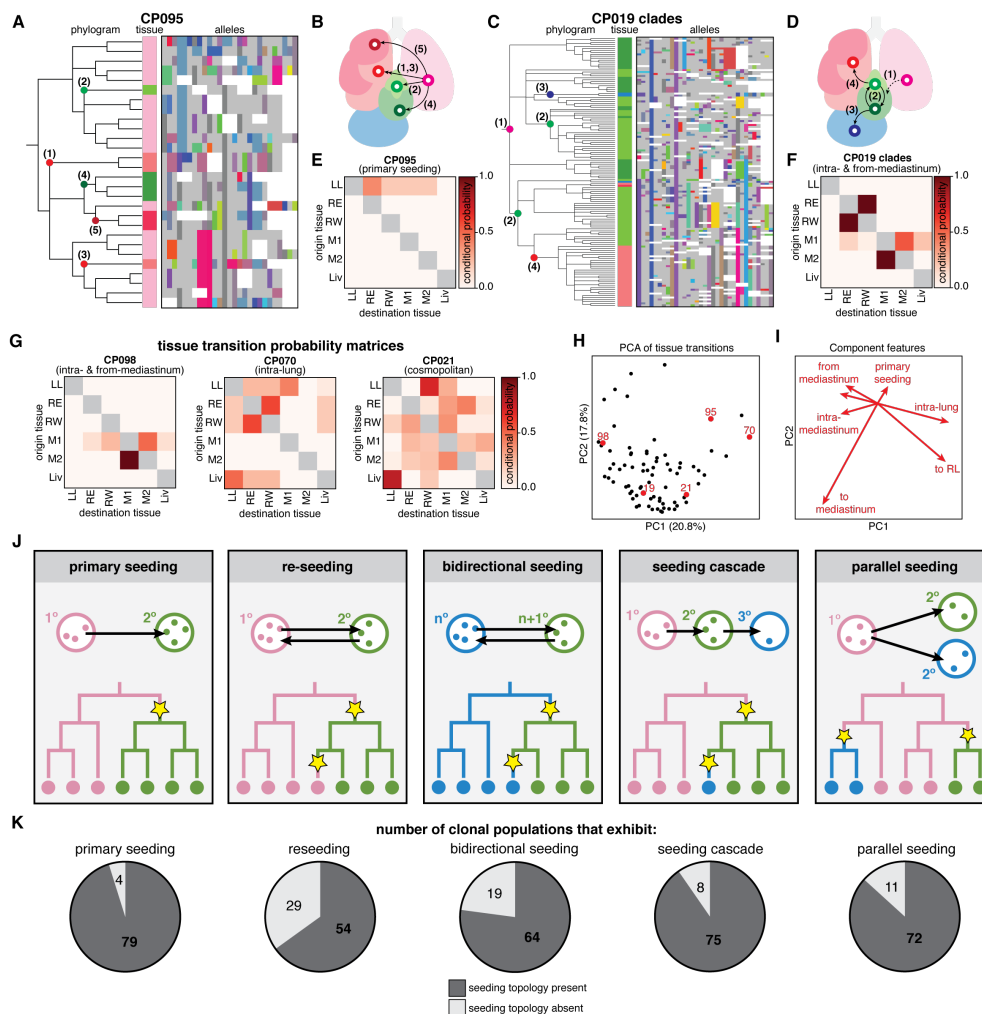


Figure 4.7: **Metastases were seeded via complex tissue routes and multidirectional topologies.** (A and C) Phylogenetic trees and lineage alleles for clonal population CP095 and CP019 clades, respectively. (B and D) Notable metastatic events are annotated in the phylogram and represented graphically as arrows. Cells are colored by tissue as in **Figure 4.1**, lineage alleles are colored as in **Figure 4.3A**, and the dashed arrow indicates an assumed transition. (E and F) Tissue transition matrices representing the conditional probability of metastasizing from and to tissues, defining the tissue routes of metastasis for each clonal population. CP095 solely exhibits primary seeding from the left lung, whereas CP019 shows more complex seeding routes. (G) Tissue transition matrices illustrating the diversity of tissue routes, including metastasis from and within the mediastinum (left), between the lung lobes (middle), or amply to and from all tissues (right). (H) PCA of tissue transition probabilities for each clonal population. Displayed clones are annotated in red; the percentages of variance explained by components are indicated on the axes. (I) Component vectors of PCA with descriptive features. (J) Possible phylogenetic topologies of metastatic seeding, represented as in **Figure 4.4A**. (K) Number of clonal populations that exhibit each metastatic seeding topology.

tumor burden (**Figure 4.15**). This illustrates how the lineage tracer can capture subtle differences in tissue tropism for different tumor populations.

Many models of metastatic seeding topology (i.e., the sequence and directionality of metastatic transitions) have been described in cancer [263]—including reseeding, seeding cascades, parallel seeding, and others—and each is characterized by a distinct phylogenetic signature (**Figure 4.7J**). These different metastatic topologies can critically influence the progression, relapse, and treatment of cancers ([171, 155, 71, 190]; for example, reseeding of metastatic cells returning to the primary tumor site can contribute genetic diversity, resistance to treatment, and metastatic potential to tumors [108, 48]. Within this single dataset, we find numerous examples of all of these topologies (**Figure 4.7K**); indeed, we most often observe examples of all topologies within every clone (**Figure 4.31**), as well as more complex topologies that defy simple classifications (e.g., **Figure 4.7, D and G**, right), further underscoring the aggressive metastatic nature of A549 cells in this xenograft model. Extending beyond this model, these findings suggest that metastatic seeding patterns can be highly complex or patient-specific.

4.4 Discussion

By applying our next-generation, Cas9-based lineage tracer to a mouse model of metastasis, we observed meaningful features of metastatic biology that were only apparent by virtue of subclonal lineage information. Among these key insights were the broad range of metastatic rates for different tumor populations, the preexistence and stable heritability of these heterogeneous metastatic phe-

notypes, and the complex, multidirectional tissue routes by which cancer cells disseminate in this model.

The heterogeneity we observed may have intriguing implications for understanding the biology of cancer metastasis. First, rather than being a simple binary process, there appear to exist multiple distinct cell states that have characteristic and graded differences in metastatic potential, and these differences are orthogonal to proliferative potential. Second, there are characteristic transcriptional differences underlying the different metastatic states, and multiple genes involved in these differences are individually sufficient to modulate the degree of cell invasion; this suggests that coherent transcriptional programs drive these different metastatic states. Third, although these transcriptional differences can be detected *in vitro*, they are muted in that context and are amplified *in vivo*, suggesting an interplay between tissue environment and cell phenotype. Finally, these phenotypes are stably inherited over cell generations but are capable of evolution, as we document in one clear example. Understanding the genetic and/or epigenetic bases for these phenotypic differences—how they arise, how they change, and how they affect cell biology—could broadly inform our understanding of how cancer disseminates and progresses.

As a first report, this work by necessity focuses on a single model of metastasis. Nonetheless, multiple distinct steps underlie the metastatic process—including extravasation, transit between tissues, intravasation, and colonization—and the approaches described here can be broadly applied to study each of these steps and indeed other aspects of cancer progression in future work. The lineage-tracing approach could be applied to models of inducible tumor initiation [62] or patient-derived xenografts [297, 109], which we anticipate may provide a window into earlier stages of cancer

progression, such as slower or less complex metastatic dynamics than the aggressively metastatic behavior observed here by A549 cells. Lineage tracing in syngeneic cancer lines or autochthonous models of cancer could chart how an intact immune system may influence cancer progression [93, 8, 20]. It will also be of interest to investigate the roles that other gene candidates identified here play in metastasis, as well as to elucidate the molecular mechanism by which *KRT17* suppresses metastatic phenotype in vitro and in vivo—an unexpected role that this work uncovered. Merging lineage tracing with recent high-resolution spatial sequencing approaches ([79, 212, 239, 12]) would enable the exploration of cancer biology at higher spatial resolution (e.g., resolving individual tumors, rather than resolving tumorous tissues as we did here) to distinguish the clonality of micrometastases, monophyletic versus polyphyletic dissemination [21], intercellular interactions between cancer cells and the microenvironment, and the spatial constraints of tumor growth and metastasis.

Our work establishes that it is now possible to uniquely distinguish tens of thousands of cells over several months of growth in vivo, reconstruct deeply resolved and accurate cell phylogenies, and then interpret them to identify rare, transient events in the cells' ancestry (here, metastasis), revealing otherwise unapparent distinctions in cellular phenotypes. Extending beyond metastasis, this approach can inform many other facets of cancer biology, such as the timing or order of genetic mutations during malignant transformation, adaptation to different tumor microenvironments, or the origin and mechanism by which tumor cells acquire resistance to therapeutic agents. And beyond cancer, our approach has the potential to empower the study of the phylogenetic foundations of biological processes that transpire over many cell generations at unprecedented resolution and scale.

4.5 Methods

Cell culture

. A549 cells (human lung adenocarcinoma line, American Type Culture Collection (ATCC) CCL-185) and H1299 cells (ATCC CRL-5803) were maintained in Dulbecco's modified eagle medium (DMEM, Gibco) supplemented with 10% (v/v) fetal bovine serum (FBS; VWR Life Science Seradigm), 2 mM glutamine, 100 units/mL penicillin, and 100 μ g/mL streptomycin (hereafter "complete DMEM"). Cells were cultured at 37°C in a humidified 5% (v/v) CO₂ atmosphere. Cells were split into fresh culture medium every two to three days by trypsinization with TrypLE reagent (Gibco) quenched with complete DMEM, and maintained at cell density as recommended by ATCC.

Plasmid design and cloning

The triple-sgRNA-BFP-PuromycinR lentivector, PCT62 (to be made available on Addgene), was constructed using four-way Gibson assembly (NEB) as previously described [130], and expresses three sgRNA cassettes driven by distinct U6 promoters, with constitutive BFP and puromycin-resistance markers for selection. The three sgRNAs are complementary to the three cut-sites in the Target Site (PCT48), except for precise single base-pair mismatches that decrease their avidity for the cognate cut-sites and subsequently slow lineage recording kinetics [91]. Guide RNAs for CRISPRi and CRISPRa experiments were chosen from the human CRISPRi/a v.2 libraries (Addgene #83969 and #83978, respectively) and were cloned into the pLG1 lentiviral backbone (Addgene #84832) using the annealing and ligation as previously described [114].

Lentivirus preparation and infection method

Lentivirus was produced by transfecting HEK293T cells with standard 4th generation packaging vectors delivered by TransIT-LTI transfection reagent (Mirus) as described in [127] and [3]. Target Site (PCT48) lentiviral supernatant was concentrated 10-fold using Lenti-X Concentrator (Takara Bio) according to manufacturer's instructions. Viral preparations were filtered and frozen prior to infection. Triple-sgRNA lentiviral preparation (PCT62) was titered and diluted to a concentration to yield approximately 50% infection rate. All infections were performed in 6-well plates with 2 mL media and 2×10^5 adhered cells per well; 1 mL of titered lentivirus was added to the culture medium with $8 \mu\text{g/mL}$ of polybrene and incubated at 37°C overnight, after which media was replaced and cells were expanded to larger culture volumes as needed.

Cell line engineering and selection strategies

Two separate lineage tracing-competent A549 cell lines (hereafter, "A549-LT1" and "A549-LT2") were engineered separately and by different methods, primarily distinguished by (i) the method of high copy-number transduction of the Target Site (piggyBac transposition and high-titer lentiviral infection, respectively) and (ii) the version of the "molecular recorder" technology used (the original components described in [37] and the improved components described in [127], respectively). For both A549-LT1 and -LT2, the cells were first infected with Luciferase-NeomycinR lentivirus; two days following infection, neomycin(+) cells were selected by treatment with $800 \mu\text{g/mL}$ G418 Geneticin (Thermo Fisher) every second day for 10 days, expanding the cells as necessary. Next, the

A549+Luciferase cells were infected with either Cas9-P2A-BFP or Cas9-P2A-mCherry lentivirus, respectively; three days following infection, BFP(+) or mCherry(+) cells were collected by fluorescence-activated cell sorting (FACS) using the BD FACS Aria II at the UCSF Center for Advanced Technology core. Next, the A549+Luciferase+Cas9 cells were serially transduced with different versions of the GFP-TargetSite vector (transposon-based PCT17 or lentivector-based PCT48, respectively). For the A549-LT1 line, the transposon transduction was performed by electroporation using an Amaxa Nucleofector (Lonza) with 100 ng of PiggyBac transposase plasmid (SBI System Biosciences) and 500 ng of PCT17 transposon plasmid. For the A549-LT2 line, the high-titer PCT48 lentiviral infections were performed as above, but in triplicate and pooled to further maximize diversity. Following GFP-TargetSite transduction, GFP(+) cells were collected by FACS. These steps were repeated two additional times for a total of three serial transductions of the Target Site, fluorescence-sorting cells with progressively higher GFP fluorescence after each transduction (**Figure 4.9**). Finally, four days prior to implantation, A549+Luciferase+Cas9+TargetSite cells were transduced by titrated triple-sgRNA (with BFP-PuromycinR markers; PCT61) lentivirus; 36 hours following infection, BFP(+) cells were collected by FACS and returned to in vitro culture until final preparation for implantation, thus producing lineage tracing-competent A549-LT1 and -LT2 cell lines

Mouse care

Mouse experiments were performed as in [189]. Six- to eight-week-old female SCID C.B-17 mice (*C.B-Igh* – *1^b/IcrTac-Prkdc^{scid}*; Taconic) were maintained in specific-pathogen-free conditions in

facilities approved by the American Association for Accreditation of Laboratory Animal Care. Surgical procedures were reviewed and approved by the UCSF Institutional Animal Care and Use Committee (IACUC), Protocol #AN107889-03C.

Orthotopic lung xenografts

To prepare A549-LT cell suspensions for implantation, cells were collected from culture by trypsinization and quenched with complete DMEM. Cells were then washed in cold PBS and resuspended with cold Matrigel matrix (BD Bioscience) at the appropriate final concentration (500, 1000, 3000, and 10,000 cells/ μ L for mice M5k, M10k, M30k, and M100k, respectively). The Matrigel cell suspensions were gently mixed and transferred into a 1-mL syringe and remained on ice until implantation. Orthotopic implantations were performed as in [189]: mice were placed in the right lateral decubitus position and anesthetized with 2.5% inhaled isoflurane. A 1-cm surgical incision was made along the posterior medial line of the left thorax. Fascia and adipose tissue layers were dissected and retracted to expose the lateral ribs, the intercostal space, and the left lung parenchyma. Upon recognition of left lung respiratory variation, a 30-gauge hypodermic needle was used to advance through the intercostal space approximately 3 mm into the lung tissue. Care was taken to inject 10 μ L (5,000, 10,000, 30,000, or 100,000 cells, respectively) of cell suspension directly into the left lung. Mice were observed post-procedure for 1–2 hours, and their body weights and wound healing were monitored weekly. Experiments were performed across two mouse cohorts: A549-LT1 cells were implanted in the first mouse cohort (including mice M10k and M100k); A549-LT2 cells were

implanted in the second cohort (including mice M5k and M30k).

Bioluminescence imaging

Mice were imaged with the Xenogen IVIS 100 bioluminescent imaging system at the UCSF Pre-clinical Therapeutics Core. Bioluminescence monitoring of the tumor engraftment and metastatic progression was performed biweekly. Before imaging, mice were anesthetized with inhaled isoflurane and injected intraperitoneally with 200 μ L of D-Luciferin at a dose of 150 mg/kg body weight. For each mouse, bioluminescent signal was measured from the thoracic cavity in the supine position and calculated automatically using Living Image Software; radiance units are photons/s/cm²/steradian. Mice were sacrificed after bioluminescent signal was anatomically extensive throughout the thorax but before the mice exhibited labored breathing (53 to 80 days post-implantation; varied by mouse). Mice were injected with D-Luciferin as before and subsequently killed; the heart and lungs were resected en bloc, the heart was removed, and the right and left lung lobes were separated. Tumorous tissues were imaged ex vivo by either bioluminescence (performed as before) or fluorescence using a stereo fluorescent microscope (Nikon).

CRISPRi/a perturbations and invasion assays

CRISPRi/a cell lines were first generated by infecting parental cell lines with lentivirus carrying either the CRISPRi (dCas9-BFP-KRAB; Addgene #46911) or CRISPRa (dCas9-XTEN-VPR-GFP [251]) genetic component, respectively; stably transduced cells were selected to purity by FACS. CRISPRi

and CRISPRa activity was confirmed by perturbing the gene expression of the cell-surface marker genes CD81 and CD151 (for CRISPRi) and CXCR4 (for CRISPRa); perturbations were performed by infecting with lentivirus carrying sgRNA against the target marker genes and transduced cells were selected for 4 days with puromycin as described above. Seven days after transduction, the perturbed cells were collected and stained with APC-labelled antibodies against CD81, CD151, and CXCR4 (BioLegend), and the fluorescence of >103 individual cells was measured by flow cytometry (Attune NxT, Thermo Fisher Scientific). For invasion assays, CRISPRi and CRISPRa cell lines were treated with lentivirus carrying sgRNAs against the targeted metastasis-associated gene candidates. We selected five candidate genes of interest to assay based on the following criteria: the strongest positive (*IFI27*) or negative (*KRT17*) genes identified from mouse M5k; in the same gene family as *IFI27* (interferon-induced; *IFI6*); or previous evidence of modulating invasion phenotype in similar models (*ASS1*, *ID3*) [38, 205]. Cells transduced with sgRNAs against these genes were selected and cultured for one week, as above. For the invasion assays, DMEM supplemented with 10% FBS was added to the bottom chamber of a 24-well trans-well plate. The perturbed cells were counted, and 1.5×10^4 cells were resuspended in serum-free media and added to the top chamber of 8- μ m pore matrigel-coated (invasion) or non-coated (migration) trans-well inserts (Corning BioCoat). After 16 hours, non-invading cells on the apical side of inserts were scraped off and the trans-well membrane was fixed in methanol for 15 minutes and stained with Crystal Violet for 30 minutes. The basolateral surface of the membrane was visualized with a Zeiss Axioplan II immunofluorescence microscope at 10 \times . Each trans-well insert was counted manually at 10 \times and performed in triplicate. Invasion phenotype was calculated as the number of invading cells through the matrigel membrane

divided by the mean number of cells migrating through control insert. Differential invasiveness was assessed using a two-tailed t-test comparing the invasion rate of the perturbation to both negative controls.

Library Sequencing

Sequencing libraries from each tissue sample were pooled to yield approximately equal coverage per cell per sample; gene expression libraries and Target Site amplicon libraries were pooled in an approximately 10:1 molar ratio to yield more RNA reads than Target Site reads. Libraries were further pooled with approximately 5% PhiX genomic DNA library added for quality-control. The libraries were sequenced using a custom sequencing strategy on the NovaSeq S2 platform (Illumina) in order to read the full-length Target Site amplicons. Sample identities were read as dual indices (I1 and I2: 8 cycles each); the 10X cell barcode and unique molecular identifier (UMI) sequences were read first (R1: 26 cycles) and the Target Site sequence was read second (R2: >250 cycles). Over 4.70 billion sequencing clusters passed Illumina QC filters and were processed as described below. All raw and processed data has been made available on GEO accession no. GSE161363. Only the first 98 bases per read were used for analysis in the RNA expression libraries to mask the longer reads required to sequence the Target Sites.

TargetSite sequencing data processing

Raw Target Site sequencing data was processed using the Cassiopeia processing pipeline as defined [127]. Briefly, reads with identical cellBC and UMI were collapsed into a single, error-corrected consensus sequence representing a single expressed transcript. Consensus sequences with poor quality or sequencing coverage were removed, according to pipeline-defined thresholds. Each consensus sequence was aligned to the wild-type reference Target Site sequence, and the intBC and indel alleles were called from the alignment. These data are summarized in a molecule table which records the cellBC, UMI, intBC, indel allele, read depth, and other relevant information.

Calling clonal populations

Collected cells were grouped into “clonal populations”, defined as populations of cells which descended from a single engineered clone and which therefore shared the same set of intBCs. This was accomplished using an iterative strategy of defining de novo sets of intBCs and assigning cells to clonal populations based on the sets, and repeating this process to further refine the assignments, similar to the strategy described in [127], as follows: (1) The most frequently observed intBC (“top intBC”) was identified from the molecule table. (2) A “clustered intBC set” was defined as the intBCs present in >20% of cells containing the top intBC. (3) Cells with >25% of their Target Site UMIs pertaining to the clustered intBC set were then collected into a “cluster” and removed from the molecule table. (4) This process of identifying the top intBC, clustered intBC set, and cell clusters was iterated until at least one of two stopping criteria was met: (i) the returned cluster of cells was empty,

indicating that the clustering strategy had exhausted well defined clusters, or (ii) the total fraction of clustered cells exceeded 98% of all cells in the molecule table. (5) Next, a "clonal population intBC set" was defined for each cell cluster, defined as intBCs that were present in >20% of cells in the cluster. Some clusters were manually redefined based on overlapping clonal population intBC sets likely resulting from the serial transduction strategy of cell line engineering; i.e., some clusters were expected to share intBCs because they were clonally related during serial integration of the Target Site (see **Figure 4.9B** for an example). (6) Finally, cells from the initial molecule table were assigned to a clonal population if >60% of its UMIs pertained to a single clonal population intBC set. Clonal populations with fewer than 25 cells were not considered, as lineage tracing over small cell numbers is less informative.

In the pre-implantation sample, we assigned cells to clonal populations observed in M5k by evaluating the proportion of TargetSite UMIs that corresponded to a clonal population intBC set. As above, we summed together the UMI proportions of each clonal population intBC set for each cell, thus yielding a similarity score that we could use to assign cells to clones. Here, we assigned a cell to a clone if at least 75% of its UMIs pertained to that clonal population intBC set.

Filtering cells and assembly of allele tables

Cells with poor Target Site capture (defined as cellBCs with <10 Target Site UMIs) were removed due to poor representation. Cells with ambiguous allele states were removed if >10% of the UMIs had conflicting allele states (i.e., intBC-distinguished Target Sites with more than one allele state), likely

resulting from PCR errors or cell doublets (wherein both cells are in the same clonal population; so-called “intra-doublets”). Cells with ambiguous clonal population assignment were removed if <60% of its UMIs pertained to a single clonal population’s set of intBCs; this likely results from cell-free Target Site transcripts, PCR errors, or cell doublets (wherein cells are from different clonal populations; so-called “inter-doublets”) Finally, the filtered molecule table with clonal population assignments is collapsed into an allele table, which summarizes the cellBC, intBC, indel allele, number of UMIs, assigned clonal population, and other relevant information.

Calculation of Tissue Dispersal Score

The Tissue Dispersal Score is the inverted Cramér’s $V(1 - V)$, a statistical measure of the association between two variables derived from the chi-squared test, ranging from 0 (no deviation from the background) to 1 (complete deviation from the background). For a given clonal population, we first perform a chi-squared test by forming a contingency table X over summarizing the number of cells found in each tissue for the clonal population, and the number of cells found in each tissue aggregated across all other clonal populations (referred to as the “background”). Importantly, the number of cells found in each tissue in the background are scaled such that the sums for both columns in X are equal. After performing a chi-squared test on the $r \times k$ contingency table, X , with a total of N counts across both columns, we derive the bias-corrected Cramér’s V test [18] statistic:

$$V = \sqrt{\frac{\phi'}{\min(\hat{k} - 1, \hat{r} - 1)}}$$

where $\phi' = \max(0, \frac{\phi}{N} - \frac{(k-1)(r-1)}{N-1})$, $\hat{k} = \frac{(k-1)^2}{N-1}$, and $\hat{r} = \frac{(r-1)^2}{N-1}$. Finally, this statistic is inverted to obtain the Tissue Dispersal Score.

Filtering of clonal populations for tree reconstruction

We removed a minority of clonal populations that exhibited suboptimal lineage tracing parameters, as defined by <15% of cut-sites bearing indels (i.e., an estimate of the lineage recording kinetics) and <66.7% of unique cell allele states (i.e., an estimate of the lineage diversity and information content). Phylogenetic reconstruction in such poor parameter regimes resulted in low information trees (indicated by the shallow tree depth of filtered trees in Figs. S6E and S7C) that would not be interpretable for sensitive downstream analysis, such as calculating the inferred rate of metastasis.

Assembly of character matrices for phylogenetic tree reconstruction.

To reconstruct lineages, we created “character matrices” from the allele tables of each clonal population using the Cassiopeia software. Specifically, we summarized the indels observed in each of the N cells in a clonal population across the M cut-sites to form an $N \times M$ matrix where each entry is an integer-representation of the indel observed in that cell at that cut-site. These entries are referred to as “character-states” or “states”; the columns of this matrix are referred to as “characters”; and the rows are referred to as “cells” or “samples”, interchangeably. Missing data was specified as the string ‘-’, and uncut sites were specified as ‘0’.

Cassiopeia-build pipeline overview

We reconstructed each clonal population's phylogeny using the Cassiopeia-Hybrid module, as described in [127]. Briefly, Cassiopeia-Hybrid takes as input the $N \times M$ character matrix and first uses the Cassiopeia-Greedy heuristic to split cells into small groups based on the presence, or absence, of states that occurred early in the phylogeny (referred to as "character-splits"); then, each subproblem is solved precisely with the Steiner-Tree-based Cassiopeia-ILP module; finally, the completed subproblems are merged together to form the final tree. All clonal populations were reconstructed with a maximum neighborhood size of 10,000 (`-max_neighborhood_size 10000`) and a maximum time to convergence of 3.5 hr (`-time_limit 12600`).

While missing data is handled in Cassiopeia-ILP by considering all possible assignments, Cassiopeia-Greedy handled missing data with two heuristics: first, cells with missing data in a character-split were classified based on their 10 closest neighbors by a modified Hamming distance normalized by the number of overlapping characters two cells shared (using the "knn" greedy missing data mode in Cassiopeia: `-greedy_missing_data_mode knn -num_neighbors 10`); second, characters with greater than 30% missing data were not selected as character-splits (i.e., `-greedy_max_missing_rep 0.3`).

We determined prior-probabilities for each indel by using the proportion of times it appeared, independently, on an intBC in a clonal population (accounting for all clonal populations in our data). These priors were used in two ways: first, they were used to select indels as character-splits that likely occurred early in the phylogeny (i.e. they appeared in several cells, but had a relatively low prior)

for Cassiopeia-Greedy; second, we used the priors to weight edges in the Steiner-Tree optimization within Cassiopeia-ILP (invoked with the `-weighted_ilp` argument to Cassiopeia's command line interface). Priors were provided to Cassiopeia's reconstruction algorithm using the `-mutation_map` argument.

To transition from Cassiopeia-Greedy to -ILP, we introduced a modified criteria based on the maximum distance from a cell population's Latest Common Ancestor (LCA) to any given cell in the population. Specifically, a new Cassiopeia-ILP subproblem is spawned if the distance to an LCA of a group of cells is below a user-defined threshold (`-cutoff <lca_cutoff>`). An appropriate LCA distance threshold was determined, iteratively, such that no Cassiopeia-ILP subproblem exceeded the user-defined maximum neighborhood size. The LCA transitioning criteria was invoked with `-hybrid_lca_mode`.

Neighbor-Joining reconstructions

For neighbor-joining reconstructions, we used the `-neighbor_joining_weighted` reconstruction option in the Cassiopeia package, which uses a scikit-bio implementation of neighbor-joining (version 0.5.5). To define distances between cells as input, we utilized a version of the modified Hamming distance function that incorporated prior-probabilities (prior-probabilities are passed in via the `-mutation_map` argument). Specifically, we defined $d'(a, b)$ as the sum of all the pairwise $h'(a, b)$ values across the cut-sites in a clonal population:

$$h'_p(a_i, b_i) = \begin{cases} -\log(p(a_i)) - \log(p(b_i)) & \text{if } a_i \neq b_i \text{ and } a_i, b_i \text{ are mutated} \\ -\log(p(a_i)) & \text{if } a_i \text{ mutated, } b_i \text{ unmutated} \\ -\log(p(b_i)) & \text{if } b_i \text{ mutated, } a_i \text{ unmutated} \\ \log(p(a_i)) + \log(p(b_i)) & \text{if } a_i = b_i \text{ and } a_i, b_i \text{ are mutated} \\ 0 & \text{otherwise} \end{cases}$$

Finally, distances $d'(a, b)$ were normalized by the number of cut-sites overlapping between cells a and b .

Allelic vs. phylogenetic distance calculations

The concordance between allelic and phylogenetic distances were used to assess the agreement between a tree's phylogenetic structure and the observed mutations. To quantify allelic distances, we defined a modified Hamming distance between cells a and b ,

$$d(a, b) = \sum_{i \in M} h'(a_i, b_i)$$

Where M is the number of cut-sites in the clonal population that cells a and b come from, a and b_i are the states observed at the i^{th} character, and h' is defined as follows:

$$h'(a_i, b_i) = \begin{cases} 2 & \text{if } a_i \neq b_i \text{ and } a_i \neq 0 \text{ and } b_i \neq 0 \\ 1 & \text{if } (a_i == 0 | b_i == 0) \text{ and } a_i \neq b_i \\ 0 & \text{otherwise} \end{cases}$$

Importantly, these values were normalized by $2 \times M$, the maximum distance for a pair of cells.

The phylogenetic distance was calculated for all pairs of cells as the number of non-zero length edges that separated the two cells in the tree. The phylogenetic distances were normalized by the “diameter” of the phylogeny, i.e., the maximum distance between any pair of cells.

Derivation of the AlleleMetRate

The AlleleMetRate is provided as a “tree-agnostic” measurement of a clonal population’s intrinsic metastatic potential. Intuitively, the rate measures the proportion of cells that do not reside in the same tissue as their closest relative (as determined by the modified Hamming distance between two cells’ character-states). Importantly, if a cell has more than one closest relative, each of their votes are normalized by the number of relatives this cell has. More formally, the contribution a cell has to the AlleleMetRate is

$$m(c) = \frac{1}{K} \sum_{i \in \text{Neighbors}(c)} I(\text{tissue}(i) \neq \text{tissue}(c))$$

where K is the number of closest relatives a cell has, and $I(\cdot)$ is an indicator function that equals 1 if the tissue of cell i is different from the tissue of cell c . The AlleleMetRate reported is

$$M_a = \frac{1}{|L|} \sum_{i \in L} m(i)$$

where L is the set of all leaves in the tree.

Derivation of the TreeMetRate

The TreeMetRate is derived from the Fitch-Hartigan maximum parsimony algorithm [72, 104]. Briefly, the Fitch-Hartigan algorithm takes in a rooted tree with labelings at the leaves (in our case labels indicate which tissue the cell was obtained from) and in our case reports two items: (a) a distribution of labels at the internal nodes that minimize the number of transitions between tissues (i.e., achieves maximum parsimony); and (b) the minimum number of transitions between tissues (referred to as the “parsimony score”). The Fitch-Hartigan algorithm is an efficient (scaling with $n \times k$ where n is the number of cells and k is the number of tissues) and common algorithm for solving the “Small Parsimony Problem”, as opposed to the ‘Large Parsimony Problem” which attempts to find the tree of maximum parsimony.

The Fitch-Hartigan algorithm operates in two phases: first, by propagating up from the leaves the set of possible labelings at each internal node; second, by traversing in depth-first order from the root of the phylogeny and selecting one state for each internal node from its set of optimal labels. Conveniently, the parsimony score can be obtained from this second phase. To finally report the TreeMetRate, we simply normalize this parsimony score by the number of edges in the tree.

Derivation of the single-cell MetRate

The scMetRate for a given cell is defined as the average of all TreeMetRates for the clades that contain that cell. Specifically, we employ the following algorithm:

```

1: function COMPUTE_SCMETRATE(phylogeny = tree, cell = x)
2:   rate  $\leftarrow$  0
3:   num_ancestors  $\leftarrow$  0
4:   for all  $n \in \text{depth\_first\_search}(\text{tree}, x)$  do
5:     rate  $\leftarrow$  rate + compute_tree_metrage(n)
6:     num_ancestors  $\leftarrow$  num_ancestors + 1
7:   rate  $\leftarrow$  rate / num_ancestors
8:   return rate

```

where `compute_tree_metrage(n)` is a function that computes the TreeMetRate for the subtree at an internal node n (as described above).

scRNA-seq preprocessing and normalization

RNA-seq libraries were quantified using CellRanger version 2.1.1 with the GRCh38 genome build for M5k and M30k; and CellRanger version 2.0.0 with the hg19 genome build for M10k and M100k. Cells not found in the Target Site Library were filtered out. All RNA-seq datasets were normalized identically – we first normalized the UMI counts in every cell to the median number of UMIs found in each library (referred to hereafter as “UMI-normalized” counts) and then each matrix of UMI-normalized counts was log-transformed (after adding a 1-pseudocount to preserve 0’s in the data; hereafter referred to as “log-transformed” counts). In the analysis presented in **Figure 4.6A** we applied scVI (version 0.6.5) to the raw counts of the M5k and the pre-implantation cells to obtain a

shared, batch-corrected latent space (learning rate of 1e-3, 400 epochs, 10 latent dimensions). This space was used for downstream projections and Vision analyses.

***VISION* analysis on M5k**

Vision (version 2.1) was applied to the UMI-normalized counts for the 35,006 filtered and clonal population-assigned cells that overlapped between M5k's scRNA-seq and TargetSite libraries. Genes were filtered out using the Fano-filter procedure in Vision, leaving 4,005 genes. To provide a latent space to Vision, we computed a reduced dimension embedding using scVI [164] on the filtered gene matrix of raw counts, using a learning rate of 1e-3 and 40 epochs to produce 10 latent components. No batch correction was used. Signatures were obtained from MSigDB [246].

Differential expression of left lung samples

We used the UMI-normalized counts from M5k to perform differential expression between cells in the left lung, stratified by whether or not they were from a clonal population that metastasized from the left lung at any point in the experiment. All cells found in the left lung that were from clonal populations that metastasized are labeled as "Metastatic". We performed 5 differential expression tests for the cells in each group: CP029 vs. "Metastatic"; CP036 vs. "Metastatic"; CP078 vs. 'Metastatic'; CP094 vs. 'Metastatic'; and 'Metastatic' vs. all cells in CP029, CP036, CP078, and CP094. Tests were performed using the Wilcoxon rank sums test as implemented in Scanpy [282].

Differential expression of single-cell MetRate by Poisson regression

Genes differentially expressed between highly and lowly metastatic cells were determined by using a Poisson regression scheme (as implemented in the GLM package in Julia, version 1.3.7). Cells were first segmented into “Low” and “High” groups (referred to as m) based on the scMetRate. Then, the following model was employed on the UMI-normalized counts:

$$H_0 : (1 + expr) \sim 1 + SizeFactor$$

$$H_a : (1 + expr) \sim 1 + SizeFactor + m$$

Where *SizeFactor* was defined as the number of genes detected in that cell and *expr* refers to the UMI-normalized expression count of a particular gene. A Likelihood Ratio Test (LRT) was used to determine the significance of the alternative hypothesis (in particular, the significance of the model fit improvement due to the variable m). Log2 fold-changes (Log2FC's) were calculated by comparing the mean expression of a gene in the “High” group to that of the “Low” group of the UMI-normalized counts with a pseudocount of 0.01 added to account for zero counts. P-values were adjusted using the Benjamini-Hochberg false discovery rate procedure [17]. Genes were considered significant if their FDR-adjusted p-value was less than 0.01.

Cells with greater than 20% counts attributed to the mitochondrial genome were filtered out and genes observed in fewer than 10% of cells were filtered. We proceeded with 35,005 cells and 4,993 genes for M5k; 1,492 cells and 11,171 genes for M10k; 17,175 cells and 7,620 genes for M30k; and 2,105 cells and 10,371 genes for M100k. Analysis was performed similarly for the M10k, M30k, and

M100k mice, with considerations of their underlying scMetRate distributions taken into account. To ensure an accurate reflection of High and Low metastatic groups, cells were stratified according to the median in M10k, the 25th percentile in M100k; and in M30k because the scMetRate was not bimodal, we stratified cells into groups below the 25th and above the 75th percentiles.

Positive and negative gene “hits” in the M5k analysis were determined using a “discriminant” score defined as the absolute value of $\log_2(\text{fold-change})$ times the negative $\log_{10}(\text{FDR})$ of the gene. Genes with a discriminant score greater than 600 were annotated as “hits”. For the analysis in Fig 4.5C,D we identified significant gene sets as those with an FDR-corrected p-value < 0.01 and $\log_2\text{FC} > 0$ or $\log_2\text{FC} < 0$ for positive and negative sets, respectively. Significance of gene set overlap was assessed with the R package SuperExactTest [272], version 1.0.7) Finally, genes were considered “reproducible” in the M10k, M30k, and M100k analyses if they were found to be significant (FDR < 0.01) and their effect was in the same direction as in M5k.

A consensus “Metastatic Signature” was formed from the top genes in the positive and negative direction from each mouse. Specifically, we first ranked each list by the discriminant score described above. Then, for each direction in each mouse as measured by $\log_2\text{FC}$, we selected the top 30 genes. We then took the unique genes in each direction across all mice and formed a directional signature from these two lists. Scores for this signature were obtained for each cell with Vision.

Identifying gene signatures correlated with TreeMetRates

Gene signature scores for single cells were calculated with Vision, and pseudo-bulked by clonal population by taking the mean signature score for each signature across the cells in a clonal population. We then calculated the Spearman correlation using the scipy Python package (version 1.2.2) between the pseudo-bulked gene signature score and the TreeMetRate for a given clonal population.

Hotspot analysis for CP007

Hotspot (version 0.9.0) was acquired from Github (<https://github.com/YosefLab/Hotspot>) and run on the UMI-normalized counts for the 603 cells in CP007 and all genes expressed in at least 10% of cells. We used the UMI-adjusted negative binomial model ("danb") as the background gene expression model and distances between cells were computed from the CP007 tree reconstructed with Cassiopeia. We used 40 neighbors for the Hotspot analysis. Modules of at least 120 genes were identified from all genes with an FDR < 0.1 , and the Hotspot module scores were used to annotate the tree. The right lung (E) only ("RE-only") control was performed identically, except for first removing cells that were not from the RE tissue sample in CP007 (437 cells).

Inference of Tissue Transition Matrices with FitchCount

To infer the relative propensities of a clonal population's cells to transition between any two tissues, we developed an efficient algorithm for aggregating together optimal solutions proposed by the Fitch-Hartigan algorithm (see Appendix for algorithm and proof). Briefly, FitchCount is a dynamic

programming algorithm that operates on a rooted phylogeny. It begins by performing the “bottom-up” phase of the Fitch-Hartigan algorithm to obtain a distribution of optimal labels over each internal node and then employs a computationally efficient algorithm for computing the number of times a given transition occurs in all optimal solutions proposed by the Fitch-Hartigan algorithm. Finally, this algorithm returns a square matrix, M , where each value $m_{i,j}$ indicates the number of times a transition from tissue i to tissue j was observed in the tree over all optimal solutions.

In **Figure 4.7** and **Figure 4.29**, we report transition matrices for the probability of a cell metastasizing from one tissue to another, given that the cell metastasizes; i.e., $P(m_{i,j} | i \neq j)$. To obtain these conditional probability tables, P , we first set $diag(M)$ to 0 (indicating that the probability of self-transition is 0) and re-normalize each row to sum to 1.

Feature selection and Principal Component Analysis (PCA) of tissue

transition matrices

To identify the trends in the tissue transition matrices presented in Fig 4.7, we performed dimensionality reduction using Principal Component Analysis (PCA) on the flattened tissue transition matrices. Beyond all the conditional probability of transitioning between tissue samples summarized in the tissue transition matrix M , we included additional features which we hypothesized would aggregate important signals. In particular, the following were added by deriving statistics from P , the conditional probability matrix, and M , the unnormalized tissue transition matrix (recall that we observed 6 tissue samples in M5k: left lung [LL], right lung W [RW], right lung E [RE], mediastinum 1 & 2 [M1 & M2]

and liver [Liv]):

- The Left Lung reseeding rate (defined as the sum of the probabilities in the LL-column of P)
- The Mediastinum tropism rate (defined as the sum of the probabilities in the M1 or M2 columns of P)
- The Liver tropism rate (defined as the sum of the probabilities in the Liver column of P)
- The Right Lung tropism rate (defined as the sum of the probabilities in the Right Lung column of P)
- The Left lung to Right Lung seeding rate (defined as the sum of the probabilities and) $p_{LL, RW}$, $p_{LL, RE}$
- The Right Lung to Left Lung reseeding rate (defined as the sum of the probabilities and) $p_{RW, LL}$, $p_{RW, RL}$
- The “intra lung” metastatic rate (defined as the sum of the probabilities leading from the LL to either RL sample and vice versa in P)
- The “intra mediastinum” metastatic rate (defined as the sum of probabilities going between M1 and M2 samples in P)
- The Primary Seeding density (defined as the density of transitions observed in the LL row of T).
- The M1 seeding density (defined as the density of transitions in the M1 row of T).

- The M2 seeding density (defined as the density of transitions in the M2 row of T).
- The Mediastinum seeding density (defined as the sum of densities of transitions in the M1 and M2 rows of T).

This resulted in a set of 48 features used for dimensionality reduction. To perform PCA on this matrix, we concatenated the 48 features for each clonal population, standardized the features, and used PCA as implemented in the scikit-learn Python package (version 0.21.3)

Classification of Seeding Topologies

To detect seeding topologies, as presented in Fig4.7J and K, we devised simple algorithms from the tree structure and the conditional probability tissue transition matrices, P .

- To identify clonal populations that exhibited primary seeding, we evaluated if there existed some $p_{LL,x} > 0$ for some $x \in \{RW, RE, M1, M2, Liv\}$.
- To identify clonal populations that experienced reseeding to the left lung, we evaluated if there existed some $p_{x,LL} > 0$ for some $x \in \{RW, RE, M1, M2, Liv\}$
- To identify clonal populations that exhibited bidirectional seeding (i.e. seeding to any tissue that previously served as a source for a metastatic event), we evaluate whether or not there exist two metastatic events, the second of which returns to the source of the first metastatic event. To do so, we sampled 100 solutions from the Fitch-Hartigan top-down procedure (which assigns labels to internal nodes) and evaluated whether or not we observe a path from the

root to any leaf that follows a bidirectional seeding pattern (not necessarily on consecutive edges).

- To identify clonal populations that exhibited a seeding cascade, we evaluated whether or not there existed any tissue transition in a clonal population from $s_i \rightarrow s_j$ such that $s_i \neq s_j \neq LL$. Even if we do not observe any cells in the left lung of a given tumor, we know that the pattern previously described is evidence of a seeding cascade because all tumors began in the left lung of the mouse
- To identify clonal population that exhibited parallel seeding, we evaluated whether or not there existed two conditional probabilities in P , p_{x,y_1} and p_{x,y_2} , that were non-zero from any tissue x to any pair of tissues y_1 and y_2 such that $y_1 \neq y_2$.

Model framework, parameters, and assumptions of metastasis simulator

To simulate metastatic events, we built on the simulation framework presented in Cassiopeia as previously described in [127]. The framework simulates a series of D binary splits over cells with M characters and S states per character; mutations are introduced every generation for each cell according to a per-character mutation rate, m , and dropout is simulated at the end according to the dropout rate, d .

To simulate metastatic processes, we introduced three new parameters: a cell-division rate (α) metastatic rate (μ), and a probability map of transitioning between tissues should a metastatic event occur (P) that follows the same structure as the conditional tissue transition matrices discussed

above. We assume that every generation, a cell first has the opportunity to divide (by evaluating whether $Unif(0, 1) < \alpha$ and then if so, each of its offspring has the opportunity to metastasize (by evaluating $Unif(1) < \mu$). < If the daughter cell metastasizes, a tissue is chosen randomly according to the probabilities specified in P . To note, in this new simulation framework, we choose to control the size of a clonal population randomly with α , instead of subsampling at a predetermined rate as described previously.

Importantly, we can also apply this simulation framework on top of a tree by simply traversing the edges of the tree and simulating metastasis without cell divisions. For the results presented in this study, we used the M5k mouse to parameterize the simulations - especially in regards to the number of tissues that cells can metastasize to (6) and the distribution of metastatic rates (between 0 and 0.3).

Assessing accuracy of the TreeMetRate

We used the simulation framework to evaluate how well the Tissue Dispersion Score, AlleleMetRate, and TreeMetRate measurements were able to capture the underlying metastatic rate μ of a simulation. To do, we simulated trees of variable depth D and parameterized with some doubling rate, $\alpha \sim Unif(0, 0.7)$, and metastatic rate, $\mu \sim Unif(0, 0.3)$. We simulated 1000 such trees, with $D \in \{10, 14\}$ (90% of trees were simulated with $D = 10$, and 10% of trees were simulated with $D = 14$, to roughly reflect the distribution of trees we see in our data). Target Sites were simulated using empirically relevant parameters: 40 characters, 40 states, a dropout rate of 18%, and a muta-

tion rate of 2.5%. For the purposes of estimating the TreeMetRate, P contained uniform probabilities of transitioning to any other tissue from a given tissue.

For each tree, the AlleleMetRate and TreeMetRate were calculated directly from the cells or tree; the Tissue Dispersion Score was calculated with a background derived from the tissue distribution across the 1,000 trees. Performance was assessed by the agreement between a specific score and the underlying rate of metastasis for that tree, as in **Figure 4.16A-D**.

Bootstrapping analysis of TreeMetRate stability

Bootstrapping analysis was performed on a set of 10 simulated trees from [127]. For each simulated tree, we sampled with replacement the cut-sites of the character matrices 100 times, resulting in 100 "bootstrapped character matrices" for each tree. We reconstructed each of these bootstrapped character matrices with Cassiopeia-ILP (maximum_neighborhood_size = 10,000, time_limit = 12,600). This left us with 1,000 reconstructions where each reconstruction corresponded to one of 10 simulated trees.

Then, for each simulated tree, we overlaid 50 metastatic processes on to the tree (here, because the tree was already defined, it was not necessary to use α). Each time, we transferred the labels to each of the 100 reconstructions for that simulated tree and evaluated the mean TreeMetRate, as well as the standard error, defined as:

$$\hat{m}ean(TMR) = \frac{1}{B} \sum \text{compute_tree_metrate}(\text{tree})$$

$$\hat{se}(TMR) = \frac{1}{B} \sum \text{compute_tree_metrate}(\text{tree}) - \text{mean}(TMR)$$

where B is the number of bootstrapped samples for a given metastatic process (here, $B = 100$). The coefficient variation for each metastatic process was defined as the ratio $\frac{\hat{se}}{\text{mean}}$.

Assessing accuracy of the tissue transition matrices

We evaluated the accuracy of conditional tissue transition matrix inference by utilizing a similar approach above: we simulated trees ($\alpha \sim \text{Unif}(0, 0.7)$, $\mu \sim \text{Unif}(0, 0.3)$) with $D \in \{10, 14\}$ that allowed cells to move between 6 tissues (these probabilities were specified in the conditional probability matrix P). For each simulation, we evaluated how well a particular statistic was able to capture the underlying *conditional* tissue transition matrix P' (i.e. the non-diagonal probabilities, corresponding to the probability that a cell will move to some tissue, given it is found in a particular tissue and is metastasizing) calculated from the ground-truth internal labels.

Conditional tissue transition matrices were simulated according to two models: one where the rates of metastasizing to any other tissue were uniform (hereafter referred to as the "Uniform" model) and one where rates could be biased to some tissues, i.e., experience various tissue tropisms (hereafter referred to as the "Biased" model). To model both scenarios, we sampled each tissue's conditional transition probabilities from a Dirichlet distribution, parameterized with a length 5 array $\bar{\alpha}$ (recall that we are simulating conditional tissue transitions, and are not concerned with the probability that a cell remains in place) that essentially provides the "density" towards any outcome. To

simulate the "Uniform" model, $\hat{\alpha}$ was parameterized as [50, 50, 50, 50, 50] - yielding roughly uniform probabilities across the 5 tissues. To simulate the "Biased" model, we used a "flat Dirichlet distribution", corresponding to $\bar{\alpha}$ parameterized as [1, 1, 1, 1, 1], which yielded potentially very biased probability distributions. We simulated 500 trees per model.

We used three algorithms for inferring the conditional tissue transition matrix P for a given tree:

1. *FitchCount* (as described above, and in the Appendix).
2. Naive-Fitch (in which the internal nodes were assigned from a single solution drawn from the Fitch-Hartigan top-down algorithm and used to infer P').
3. Majority-Vote (in which the internal nodes were assigned the label that appeared at the greatest frequency at the leaves below the node and these assignments were used to infer P').

In both the Naive-Fitch and Majority-Vote case, after internal nodes were assigned a label we performed a depth-first traversal on the tree and counted the number of times a parent transitioned to a child. Ground-truth conditional transition matrices were found similarly, using the simulated ground-truth labels for internal nodes.

To evaluate accuracy, we computed the Spearman correlation (using Python's *scipy* library, version 1.2.2) between the flattened matrix of the ground-truth conditional transition matrix and the transition matrices inferred by one of the three algorithms described above (*FitchCount*, Naive-Fitch, and Majority-Vote).

4.6 Appendix

Deriving transition matrices from phylogenetic trees

In this document, we describe our approach for solving the following problem: Consider a phylogenetic tree \mathcal{T} rooted at vertex r over V vertices and E edges which we denote as \mathcal{T}^r . In this tree, each leaf l is assigned a state $state(l)$, where states are drawn from a state space Σ (i.e. $\forall v, state(v) \in \Sigma$). Here, leaves are single cells derived from a single-cell lineage tracing experiment [37] studying cancer metastasis, the tree describes the cells' evolutionary history (as inferred by Cassiopeia [127]), and the states represent the tissues from which the cells were obtained. Our task is to derive summary statistics obtained by assigning states to the internal nodes of the tree (i.e., ancestral cells that were not observed in the study). Specifically, the summary statistics we are interested at are: (1) the overall number of metastatic events (i.e., transition between tissues) that occurred in a clone's history and (2) the frequency (i.e. number) of transitions between each pair of tissues s_i and s_j .

This class of problems typically requires what is known as "ancestral state reconstruction" [132, 232, 179], which in essence attempts to assign an ancestral states to every node in a given tree that minimizes some function. Here, we use parsimony as our objective function. Our work relies on classical algorithms [72, 104] and is also inspired by recent algorithms using these principles to infer metastatic histories like MACHINA [135].

As noted by El-Kebir et al [135], while there exist several algorithms for effectively inferring clone trees from DNA samples of metastatic cancers [209, 56, 137], none of them except for MACHINA

propose an ancestral node labeling and subsequent classification of metastatic topologies. This is because a metastatic history does not follow uniquely from a tree structure, and metastasis itself is not necessarily unidirectional (i.e. there exist polyclonal & reseeding events that may introduce cyclic topologies). Though MACHINA represents a significant advance, we build on it by reporting summary statistics over *all* optimal solutions rather than one, and additionally circumvent the computationally-intensive Integer Linear Programming (ILP) optimization routine in favor for a dynamic programming approach.

Below, we describe our algorithmic strategy (and prove it) for inferring both summary statistics. We begin by describing how the overall number of transitions can be derived from the Fitch-Hartigan algorithm [72, 104]. Next, we introduce *FitchCount*, an algorithm for counting the number of transitions between any two tissues in a given phylogeny over all optimal solutions to the Fitch-Hartigan algorithm.

Algorithmic strategy

The first summary statistic that we are interested in is the minimal possible number of state transitions in the tree that is sufficient to explain the state assignment to the leaves (which is given as an input). In other words, out of all possible assignment of tissue labels to the ancestral cells (which can be exponentially many), consider the assignments that entail the minimum number of cases in which a parent node and a child node come from a different tissue (i.e., state transition). Our first goal is to retrieve the number of state transitions in these optimal assignments, but not the assignments

themselves (note that the number of transition is the same [i.e., minimum possible] in all optimal assignments). In our second goal, we are interested in the number of specific transitions across all optimal assignments, which means that we would have to also look at the optimal assignments themselves.

While our first goal can be readily addressed by the classical algorithm of Fitch [72] and Hartigan [104] or using another algorithm by Sankoff [219], the second goal requires an additional procedure. The reason for this is that the existing algorithms are able to retrieve only one specific optimal assignment in each run through the tree. However, we would ideally like to base our summary statistics on the space of all possible optimal assignments. Since there can be exponentially many optimal assignments, we needed to find an efficient way to extract the summary statistic without actually enumerating all algorithms.

Notably, the Fitch [72] algorithm was originally designed for binary trees. An important property of this algorithm is that the optimal assignments that it produces guarantee optimality even if we consider every sub-tree in isolation. This is different from a scenario where we allow state assignments that may not be optimal when we are considering only a certain sub-tree but become optimal due to compensation elsewhere in the tree. The Hartigan algorithm extends it to non-binary trees and can also be modified such that its optimal assignments remain optimal in every sub-tree. We refer to this modification as the Fitch-Hartigan algorithm. This algorithm operates in a time linear in the input size (i.e., it scales proportionally to $n \cdot k$ where n is the number of cells and k is the number of possible states [tissues, in our case]). The Sankoff algorithm [219] uses a more involved formulation with a slower run time (it scales proportionally to $n \cdot k^2$) that can account for different penalties to

different state transitions. It also returns all possible optimal solutions, including solutions that may not be optimal for every sub-tree, if it is considered in isolation. Here, we employ the Fitch-Hartigan approach due to its simplicity and speed and since we reasoned that it is desirable that our solutions remain optimal, not just for the entire clone, but also for every sub-clone individually.

Finding the minimal number of transitions

The Fitch-Hartigan algorithm begins with a "bottom-up" procedure in which labels at the leaves are propagated up to internal nodes in the tree. This "bottom-up" phase assigns a set of labels (tissues) $opt[v]$ to each node v in the tree that satisfy the optimality demands (namely, maximum parsimony). Specifically, for every $s \in opt[v]$ there exists at least one state assignment to the nodes in T^v (the tree rooted by v) where the state of v is s and that is optimal for T^v and for every sub-tree of T^v . Furthermore, the set $opt[v]$ includes all such states. For completeness, we provide a proof for these claims in the appendix (Claim 3).

In addition to generating the sets opt the algorithm can also count for every node v the minimum possible number of state transitions $n(v)$ required in the sub-tree rooted by v (Observe that $n(r)$ for the tree rooted at r corresponds to the number of transitions across the entire tree). The "bottom-up" procedure opt is applied in **post-order traversal** (evoked by applying it on the root node) with the following pseudocode:

```

1: function OPT(node = v)
2:   if is_leaf(v) then
3:     return {state(v)}    ▷ for leaves return their assignment, which was provided as input
4:    $\forall s \in \Sigma, c(s) = \#\{u \in \text{child}(v) \text{ s.t. } s \in \text{opt}(u)\}$ 
5:    $k = \max_{s \in \Sigma} c(s)$ 
6:    $n(v) = |\text{child}(v)| - k + \sum_{u \in \text{child}(v)} n(u)$ 
7:   return {s ∈  $\Sigma$  s.t.  $c(s) = k$ }

```

Note that in the original formulation by Hartigan, an additional complication is added to increase the number of optimal assignments that can be retrieved by the algorithm. This is done by accumulation of an additional set of states $\text{opt}_2(v)$ for every node v that can be used to derive solutions that are globally optimal, but are not optimal in the sub-tree rooted by v . We therefore excluded this part of the algorithm (see appendix for proof [Claim 3]).

Inferring the frequency of different state transition events

The second part of the Fitch-Hartigan algorithm is a top-down procedure for finding one (out of potentially many) optimal solution, i.e., a labeling $\text{state} : V \rightarrow \Sigma$ of each node $v \in V$ in the tree with a state $s \in \Sigma$. It starts by randomly selecting a state for the root node r out of the set $\text{opt}(r)$ and then continues to select legal (see Definition 2) states for child nodes, based on the value assigned to their parent.

```

1: function STATE-ASSIGNMENT(node = v)
2:   if is_root(v) then
3:     state(v) = random selection out of opt(v)
4:   else
5:     if state(parent(v)) ∈ opt(v) then state(v) = state(parent(v))
6:     else state(v) = random selection out of opt(v)

```

The above procedure can face many ties during its execution, and can thus potentially return all optimal solutions (provided that they remain optimal for every sub-tree) if it is applied many times. When we compare Fitch-Hartigan to FitchCount in the main text, we use one top-down round (i.e., consider one optimal solution) for the former.

The *FitchCount* procedure was designed to provide a comprehensive evaluation of state transition frequencies, by basing its estimation on the space of all possible optimal state assignments, instead of only a single or few optimal assignments. Compared to the naive approach to enumerate all possible optimal state assignments given by the Fitch-Hartigan algorithm (which may require exponential number of executions), *FitchCount* performs in $\mathcal{O}(n \cdot k^3)$ time for a tree with n leaf nodes and k possible states (assuming each internal node has at least two child nodes, which is the case in our work).

The algorithm

We define several arrays for storing necessary information:

1. $opt[v]$: The set of optimal assignments for a node v given by the Fitch-Hartigan bottom up approach procedure (defined in the algorithm *opt*).

2. $N[v, s]$: The number of optimal solutions below the node v given that it takes on the state s .
3. $C[v, s, s_i, s_j]$: The number of transitions from state s_i to state s_j in all optimal solutions of the tree rooted at v , given that v takes on the state s .
4. $M[i, j]$: The number of transitions between s_i and s_j observed across all optimal solutions to the Fitch-Hartigan algorithm.

Our overall objective is to fill in the dynamic programming matrix M , which will subsequently require knowledge of the other dynamic programming arrays. Note that in the following we refer to the arrays using either rounded parenthesis or rectangular parenthesis. The former denotes a function call and the latter denotes a retrieval of an already-computed entry (which we assume to get populated automatically after the respective function call and available globally to the algorithms).

Our Main function is:

```

1: function MAIN( $tree = T, states = \Sigma$ ).
2:    $r = \text{root of } T$ 
3:   Call  $\text{opt}(r)$             $\triangleright$  Note that  $\text{opt}(r)$  fills out the array  $\text{opt}$  in post-order from the root
4:   for all  $s \in \text{opt}[r]$  do
5:     Call  $N(r, s)$ 
6:   for all  $s \in \text{opt}[r]$  do
7:     for all  $\{s_i, s_j\} \in \Sigma^2$  do
8:       Call  $C(r, s, s_i, s_j)$ 
9:   for all  $\{s_i, s_j\} \in S^2$  do
10:     $M[s_i, s_j] = \text{sum}(C[r, :, s_i, s_j])$ 
11:  return  $M$ 

```

The following algorithms for filling in specific entries to $N[v, s]$, and $C[v, s, s_i, s_j]$:

```

1: function  $N(\text{node} = v, \text{state} = s)$ 
2:   if  $\text{is\_leaf}(v)$  then
3:     return 1
4:    $A = []$   $\triangleright$  an array storing the number of solutions in each subtree below  $v$  given its state  $s$ 
5:   for all  $u \in \text{child}(v)$  do
6:      $LS = \emptyset$   $\triangleright$  The set of legal states for node  $u$  given  $\text{state}(v) = s$ 
7:     if  $s \in \text{opt}[u]$  then
8:        $LS = \{s\}$ 
9:     else
10:       $LS = \text{opt}[u]$ 
11:     $A[u] = \sum_{s' \in LS} N(u, s')$ 
12:  return  $\prod_{u \in \text{child}(v)} A[u]$ 

12: function  $C(\text{node} = v, \text{state} = s, \text{from} = s_i, \text{to} = s_j)$ 
13:   if  $\text{is\_leaf}(v)$  then
14:     return 0
15:    $K = []$   $\triangleright$  A temporary array to store the number of transitions observed for each child
16:   for all  $u \in \text{child}(v)$  do
17:      $LS[u] = \emptyset$   $\triangleright$  The set of legal states for node  $u$  given  $\text{state}(v) = s$ 
18:     if  $s \in \text{opt}[u]$  then
19:        $LS[u] = \{s\}$ 
20:     else
21:        $LS[u] = \text{opt}[u]$ 
22:      $K[u] = \sum_{s' \in LS[u]} C(u, s', s_i, s_j)$ 
23:     if  $(s_i == s) \wedge (s_j \in LS[u])$  then
24:        $K[u] += N[u, s_j]$ 
25:   return  $\sum_{u \in \text{child}(v)} \left( K[u] \prod_{u' \in \text{child}(v) \setminus \{u\}} \left( \sum_{s' \in LS[u']} N[u', s'] \right) \right)$ 

```

Proof

We prove correctness of the dynamic programming matrices N , and C .

Claim 1. For any node v and state $s \in \text{opt}[v]$, $N[v, s]$, is precisely the number of optimal solutions in $T^{(v)}$ where $\text{state}(v) = s$.

Proof. To prove the correctness of the dynamic programming array N , we'll proceed by induction over the height of the tree $height(T) = h$. For convenience, we'll also make use of a temporary dynamic programming array A which stores the number of solutions for each child $c \in child(v)$, aggregated across each possible state of that child given the parent's state s .

Base Case #1, $h = 0$. $N[l, state(l)] = 1$. This relation is trivially true for the case where $h = 0$ and there exists a single leaf l_1 in which case the only solution consists of $state(l) = s$.

Base Case #2, $h = 1$. Consider a tree $T^{(v)}$ rooted at node v , where $child(v) = \{l_1, \dots, l_m\}$. We know that each leaf l_i has a single assignment $state(l_i)$ from the definition of the small-parsimony problem. Thus, the number of possible solutions for this tree with $state(v) = s$ is always one, namely with each leaf taking on their only state. Specifically, in this base case, we observe that $A[u] = 1 \forall u \in \{l_1, \dots, l_m\}$. To show this, we consider two cases:

- If $s \notin opt(l_i)$: $A[l_i] = \sum_{s' \in opt[l_i]} N[l_i, s'] = N[l_i, state(l_i)] = 1$ since $height(T^{(l_i)}) = 0$.
- If $s \in opt[l_i]$: $s == state(l_i)$ and $A[l_i] = N[l_i, s] = 1$ since $height(T^{(l_i)}) = 0$.

and the relation

$$N[v, s] = \prod_{u \in child(v)} A[u] = A[l_1] * \dots * A[l_m] = 1$$

thus is correct.

Inductive Hypothesis. For a tree $T^{(v)}$ of height h , and some $s \in \text{opt}[v]$, $N[v, s]$ exactly stores the number of optimal solutions in the tree rooted at v .

Inductive Step. Consider a tree $T^{(v)}$ of height $h + 1$ and some state $s \in \text{opt}[v]$. We will show that both the array A correctly stores the number of solutions for the child u given $\text{state}(v) = s$ and that the relation $N[v, s] = \prod_{u \in \text{child}(v)} A[u]$ is correct.

First, we note that for the tree to be globally optimal, for each $u \in \text{child}(v)$, $\text{state}(u)$ must be s if $s \in \text{opt}[u]$; else, any state from $\text{opt}[u]$ can be assigned to u as each incurs a cost of 1 to the overall parsimony of the tree (see Claim 3). These choices for "optimal" states are stored in the array $LS[u]$.

Second, we know from our inductive hypothesis that $N[u, s']$ is correct for any child $u \in \text{child}(v)$ and any state $s' \in \text{opt}[u]$ as the tree $T^{(u)}$ has a height h . Thus, it is clear that

$$A[u] = \sum_{s' \in LS[u]} N[u, s']$$

correctly returns the number of solutions in the subtree rooted at u over all possible legal states that u can take on.

Finally, we observe that given $\text{state}(v) = s$, each child can be treated independently as we consider global solutions that in the tree $T^{(v)}$ with $\text{state}(v) = s$. Because of this, the number of such solutions is the size of the permutation of all optimal sub-trees rooted at each $u \in \text{child}(v)$ - i.e. the product of all $A[u]$. To show this, consider v has m children. Let the set of optimal internal labellings to $T^{(u_j)}$ be denoted as $\tau_j = \{t_i^{(j)}\}_{i=1}^{A[u_j]}$ where $t_i^{(j)}$ is the i^{th} solution for the tree rooted

at u_j given $state(v) = s$. Then, the possible set of solutions is the Cartesian Product between

$\tau_1, \tau_2, \dots, \tau_m$:

$$\tau_1 \times \dots \times \tau_k = \left\{ \{t_1^{(1)}, t_1^{(2)}\}, \{t_2^{(1)}, t_1^{(2)}\}, \dots, \{t_{A[u_1]}^{(1)}, t_1^{(2)}\}, \dots, \{t_1^{(m-1)}, t_2^{(m)}\}, \dots, \{t_{A[u_{m-1}]}^{(m-1)}, t_{A[u_m]}^{(m)}\} \right\}$$

This Cartesian product has a size $|\tau_1| \times |\tau_2| \dots \times |\tau_m| = A[u_i] \times \dots \times A[u_k] = \prod_{u \in child(v)} A[u]$.

Thus, this relation holds $T^{(v)}$ where $height(T^{(v)}) = h + 1$. \square

Claim 2. For any node v and state $s \in opt[v]$ assigned to v and $\{s_i, s_j\} \in 2^\Sigma$, the array $C[v, s, s_i, s_j]$ correctly stores the number of transitions from $s_i \rightarrow s_j$ in $T^{(v)}$.

Proof. We will prove by induction over the height of the tree, h , that for a node v , a state $state(v) = s$, and some $(s_i, s_j) \in 2^\Sigma$ both $K[u] \forall u \in child(v)$ and

$$C[v, s, s_i, s_j] = \sum_{u \in child(v)} \left(K[u] \prod_{u' \in child(v) \setminus \{u\}} \left(\sum_{s' \in LS[u']} N[u', s'] \right) \right)$$

are correct. Here $K[u]$ is the number of transitions from s_i to s_j that exist by considering child u of node v given $state(v) = s$ and $LS[u]$ is a function that finds the set of legal (Definition 2) assignments to u given the parent's state is s .

Base Case #1, $h = 0$. The relation trivially holds for a tree of height 0 as there cannot exist any transitions for a tree without edges. As calculated, $C[v, s, s_i, s_j] = 0$ for all leaves and thus the relation holds.

Base Case #2, $h = 1$. Consider a tree of height 1, $T^{(v)}$, where $child(v) = \{l_1, \dots, l_m\}$ and that $N[l_i, state(l_i)] = 1$. We can count the number of transitions by considering for every edge the following:

- $s \neq s_i$ then $C[v, s, s_i, s_j]$ is necessarily 0.
- $s == s_i$, then $C[v, s, s_i, s_j]$ is the number of leaves that have state s_j .

By construction, for some child l_i , $C[v, s, s_i, s_j]$ must be

$$K[l_i] = 1[s == s_i \wedge s_j \in LS[u]] == 1[s == s_i \wedge s_j == state(l_i)]$$

Then, $C[v, s, s_i, s_j] = \sum_{l \in child(v)} K[l]$. We'll prove that this is equal to the relation described above:

$$\begin{aligned} C[v, s, s_i, s_j] &= \sum_{l \in child(v)} \left(K[l] \prod_{l' \in child(v) \setminus \{l\}} \left(\sum_{s' \in LS[l']} N[l', s'] \right) \right) \\ &= \sum_{l \in child(v)} \left(K[l] \prod_{l' \in child(v) \setminus \{l\}} 1 \right) \\ &= \sum_{l \in child(v)} K[l] \end{aligned}$$

Thus, our relation holds for $h = 1$.

Inductive Hypothesis. Assume for a tree $T^{(v)}$ of height h where $state(v) = s$, $C[v, s, s_i, s_j]$ correctly computes the number of transitions from $s_i, s_j \in 2^\Sigma$ for the tree.

Induction Step. Now consider a tree of height $h + 1$ rooted at v where $state(v) = s$. We'll show that both $K[u]$ is correct for all $u \in child(v)$ and that the relation for calculating $C[v, s, s_i, s_j]$ holds.

We'll first show that $K[u]$ is correct $\forall u \in child(v)$. As defined above, $K[u]$ is the number of $s_i \rightarrow s_j$ transitions that are due to the node u given $state(v) = s$. We know that given our inductive hypothesis, for the subtree rooted at u , $T^{(u)}$, $C[u, s', s_i, s_j]$ for any state $s' \in opt[u]$ is correct. Then, the number of $s_i \rightarrow s_j$ transitions under u , given $state(v) = s$, is equal to the sum of all $C[u, s', s_i, s_j]$ for those $s' \in LS[u]$ as those are the only solutions that would be considered by the Fitch-Hartigan algorithm (note that we are guaranteed to have optimal state assignments to choose from for $LS[u]$ as we prove in Claim 3), plus the transition (if it exists) from v to u .

Now, we'll show that the relation for $C[v, s, s_i, s_j]$ holds. For the tree $T^{(v)}$ let's assume that v has m children: $child(v) = \{u_i\}_{i=1}^m$. As above, we'll maintain the notation that the set of legal assignments given that $state(v) = s$ for u_i to be $LS[u_i]$. Furthermore, let the set of $s_i \rightarrow s_j$ transitions underneath u_j be $\rho_j = \{r_i^{(j)}\}_{i=1}^{K[u_j]}$ and the set of trees that are legal, optimal assignments under u_j be $\tau_j = \{t_i^{(j)}\}_{i=1}^{\Lambda(u_j)}$ where $\Lambda(u_j) = \sum_{s' \in LS[u_j]} N[u_j, s']$, assuming $state(v) = s$. We can see then that the total number of transitions from $s_i \rightarrow s_j$ is

$$\begin{aligned}
& \left\{ \{r_1^{(1)}, t_1^{(2)}\}, \{r_2^{(1)}, t_1^{(2)}\}, \dots, \{r_{K[u_1]}^{(1)}, t_1^{(2)}\}, \right. \\
& \left. \{r_1^{(1)}, t_2^{(2)}\}, \dots, \{r_{K[u_1]}^{(1)}, t_{\Lambda(u_2)}^{(2)}\}, \dots, \{r_{K[u_1]}^{(1)}, t_{\Lambda(u_m)}^{(m)}\}, \right. \\
& \left. \{r_1^{(2)}, t_1^{(1)}\}, \dots, \{r_{K[u_2]}^{(2)}, t_{\Lambda(u_m)}^{(m)}\}, \right. \\
& \left. \{r_1^{(m)}, t_1^{(1)}\}, \dots, \{r_{A[u_m]}^{(1)}, t_1^{(2)}\}, \{r_{A[u_m]}^{(m)}, t_{\Lambda(u_{m-1})}^{(m-1)}\} \right\}
\end{aligned}$$

Which is equal to the sum of the following Cartesian Products:

$$\rho_1 \times \{(\tau_2, \dots, \tau_m)\} + \rho_2 \times \{(\tau_1, \tau_3, \dots, \tau_m)\} + \dots + \rho_m \times \{(\tau_1, \dots, \tau_{m-1})\}$$

where the cardinality of this set is

$$K[u_1] \prod_{i \in 2..m} \Lambda(u_i) + K[u_2] \prod_{i \in 1,3,..m} \Lambda(u_i) + \dots + K[u_m] \prod_{i \in 1,..,m-1} \Lambda(u_i)$$

which can be further simplified to

$$\sum_{u \in \text{child}(v)} K[u] \prod_{u' \in \text{child}(v) \setminus \{u\}} \sum_{s' \in LS[u']} N(u', s')$$

Thus the relation is correct and $C[v, s, s_i, s_j]$ is correct by induction.

□

Supplementary Definitions and Proofs

Definition 2. (*Legal Assignment*). An assignment $state(v) = s$ is **legal** for a node v and given $state(parent(v)) = s'$ if either $s == s'$ or $s' \notin opt(v)$. Observe that only legal assignments are explored in the Fitch-Hartigan algorithm, and are guaranteed to be optimal in the sub-tree rooted at v .

Claim 3. Consider any node v and let $T(v)$ denote the sub-tree rooted at v . The bottom up procedure above returns a set $opt(v)$ such that for every $s \in opt(v)$: (1) there exists a solution (state assignment) that is optimal for $T(v)$ and every sub-tree of $T(v)$, in which the state of v is s ; and (2) there does not exist a solution that is optimal for $T(v)$ and for every sub-tree of $T(v)$, in which the state of v is some $s' \notin opt(v)$

Proof. Proof by induction on tree height h (max length from root to any leaf).

Base Case #1, $h = 0$. In this case the tree consists of a single leaf node, for which state assignment is already fixed and the claim follows trivially.

Base Case #2, $h = 1$. In this case, we have one internal node v with n leaves as its immediate descendants. Here, we define $opt(v)$ as the set of all states that are found in k out of n descendants, where k is maximal. Clearly, the value of the optimal solution in this case has $n - k$ state transitions, which can be obtained by assignment of any state in $opt(v)$ to v . Furthermore, the solution is trivially optimal for every sub-tree (i.e., singleton), thus proving the first part of the claim.

Furthermore, assignment of any $s' \notin \text{opt}(v)$ to v will necessarily entail strictly more than $n - k$ state transitions, thus proving the second part of the claim.

Inductive step. Assume an internal node v whose corresponding sub-tree is of height h . Let C denote the set of its child nodes. From the induction, we assume that the claim holds for every node $u \in C$. $\text{opt}(v)$ is defined as the set of all states that are found in k out of the $m = |C|$ child nodes, where k is maximal. First let us denote by $\text{optval}(v)$ the value (number of state transitions) of the optimal solution for $T(v)$. From the assumption of the induction, it is easy to see that $\text{optval}(v) = \sum_{u \in C} \text{optval}(u) + m - k$. This value can be reached following the top-down procedure of the Fitch- Hartigan algorithm: (i) assign v with some state $s \in \text{opt}(v)$; (ii) assign s to all child nodes u where $s \in \text{opt}(u)$ (iii) assign each remaining child node u' with some other state from $\text{opt}(u')$ (iv) consider some optimal solution for each of the sub-trees that are rooted by the child nodes. Note that these optimal solutions must exist due to the assumption of our induction. Clearly, the solution that we built satisfies the first part of our claim. For the second part of our claim, assume by contradiction that there exists a state assignment in which the state of v is $s' \notin \text{opt}(v)$ that achieves optimality for $T(v)$ and all of its sub-trees. However, from the assumption of the induction we must choose an assignment for every child u out of its set $\text{opt}(u)$. It therefore follows that the value of any such solution must be at least $\sum_{u \in C} \text{optval}(u) + m - k + 1$. We note that in the original paper by Hartigan, a solution is possible where for one or more child nodes u we select an assignment that is identical to the state of the parent v but is not from $\text{opt}(u)$. However, while this solution reaches optimality for $T(v)$, it will not be optimal for $T(u)$.

□

4.7 Supplementary Figures

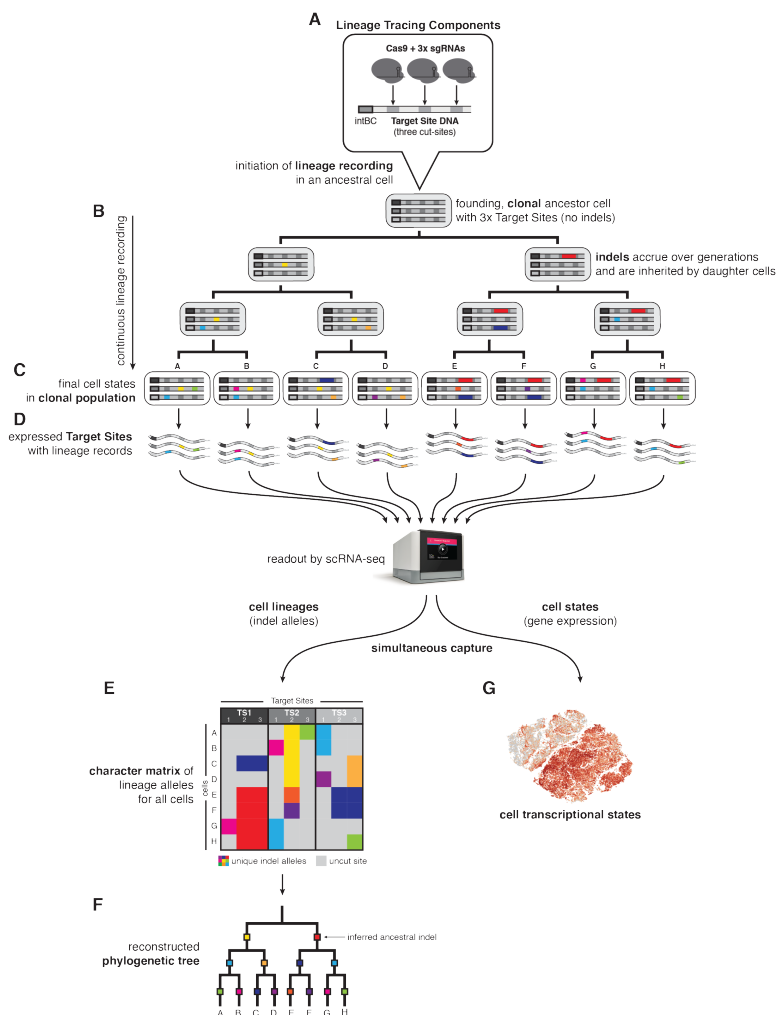


Figure 4.8: Detailed schematic of lineage tracing methodology. (A) Cells are genetically engineered with the lineage tracing components: (i) Cas9, (ii) multiple copies of the Target Site, and finally (iii) three sgRNAs that are complementary to three cut-sites on each Target Site. Multiple copies of the Target Site per cell are distinguished by unique integration barcodes (intBCs). Cas9-induced double-stranded breaks at cut-sites are repaired with high-diversity insertions or deletions (indels), which act as stable, heritable markers of cell lineage. (B) Upon initiation of lineage recording in a founding cell (i.e., one “clone”), indels continuously accrue on the Target Sites (colored boxes), which are inherited by descendants over subsequent generations. (C) At the end of the lineage recording experiment, the final population of descendent cells (i.e., one “clonal population”) are collected and (D) their expressed Target Site mRNAs are captured by single-cell RNA-sequencing (e.g., by the 10X Genomics Chromium platform) alongside transcriptome-wide expressed genes. (E) The indel allele information is read from the Target Site sequences and summarized in a “character matrix” of indel allele states (values) for each cut-site (columns) in each cell (rows). (F) From the pattern of cells’ shared and distinguishing indel alleles, a tree reconstruction algorithm builds a phylogenetic model that best captures cell–cell relationships (e.g., by maximizing parsimony), thus producing a detailed map of cell lineage. (G) Simultaneously, single-cell RNA-sequencing captures the transcriptional profiles of the observed cells, allowing for direct comparisons between cell lineage and cell state.

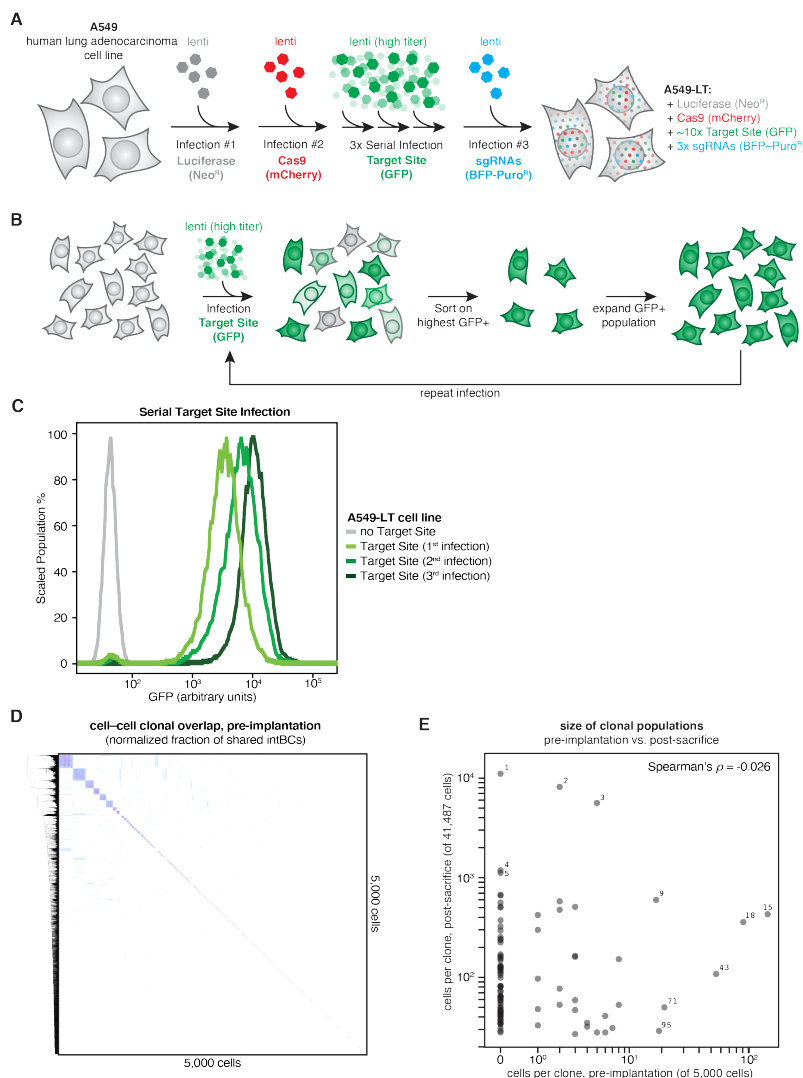


Figure 4.9: Cell line engineering strategy and estimation of clonal diversity. (A) Human lung adenocarcinoma (A549) cells were genetically engineered with the lineage tracing components by lentiviral transduction with (1) Luciferase-NeomycinR and antibiotic-selected, (2) Cas9-mCherry and fluorescence-sorted, (3) serial, high-titer TargetSite-GFP and fluorescence-sorted, and finally (4) triple-sgRNA BFP-PuromycinR and fluorescence-sorted, thus producing lineage tracing-competent A549-LT cells. (B) Serial, high-titer TargetSite-GFP lentiviral transduction strategy to achieve high copy-number. (C) Cells with high-shifted GFP fluorescence after successive TargetSite-GFP infections, indicating increasing copy-number of the Target Site. (D) Sample of 5,000 A549-LT cells prior to injection to estimate initial clonal diversity. Shown is a heat-map of the fraction of shared intBCs in all cell-cell comparisons. On average, approximately 22,000 unique, high-quality intBCs were identified per random sample of 5,000 cells; thus, we estimate approximately 2,150 distinct clones per 5,000 cells (assuming 10.3 intBCs per clonal population; Fig 4.13A) at the beginning of the experiment. (E) Comparison of the size of each clonal population observed in mouse M5k in a pre-implantation sample of 5,000 cells (in vitro) and post-sacrifice (in vivo). There is no correlation between the in vitro and in vivo population sizes (Spearman's $\rho = -0.026$).

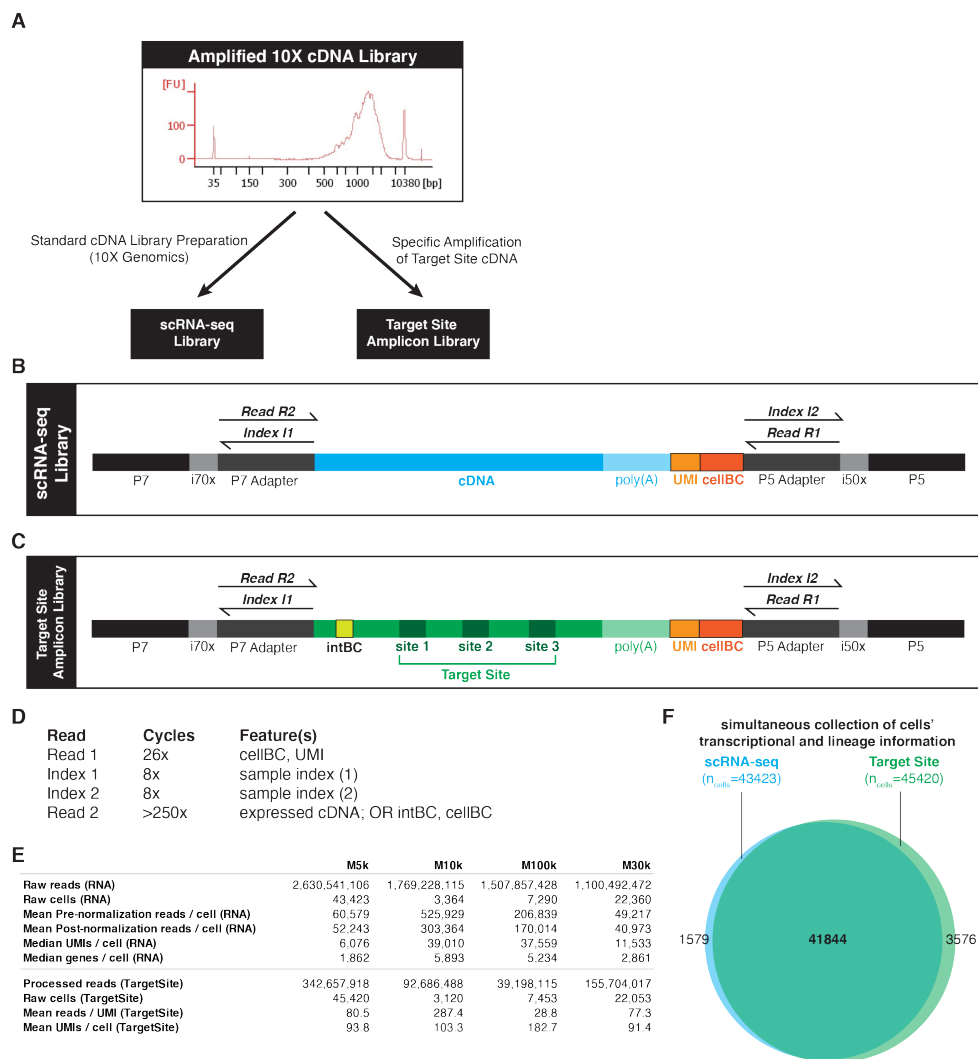


Figure 4.10: **Sequencing library construction and metrics.** (A) The amplified cDNA (shown here as a BioAnalyzer trace) from the Chromium 3' Single Cell V2 kit (10X Genomics) serves as the template for both (B) the single-cell gene expression ("RNA") library and (C) the single-cell Target Site amplicon library (Methods). (D) The cellBC and unique molecular identifier (UMI) are sequenced from Read R1, the sample identities are sequenced from Indices I1 and I2, and the expressed cDNA or the Target Site amplicon (including the intBC and cut-sites 1, 2, and 3) are sequenced from Read R2. (E) Library sequencing metrics for TargetSite and RNA libraries for each mouse. (F) There is vast overlap in the cells identified from the gene expression and lineage sequencing datasets, as shown for mouse M5k.

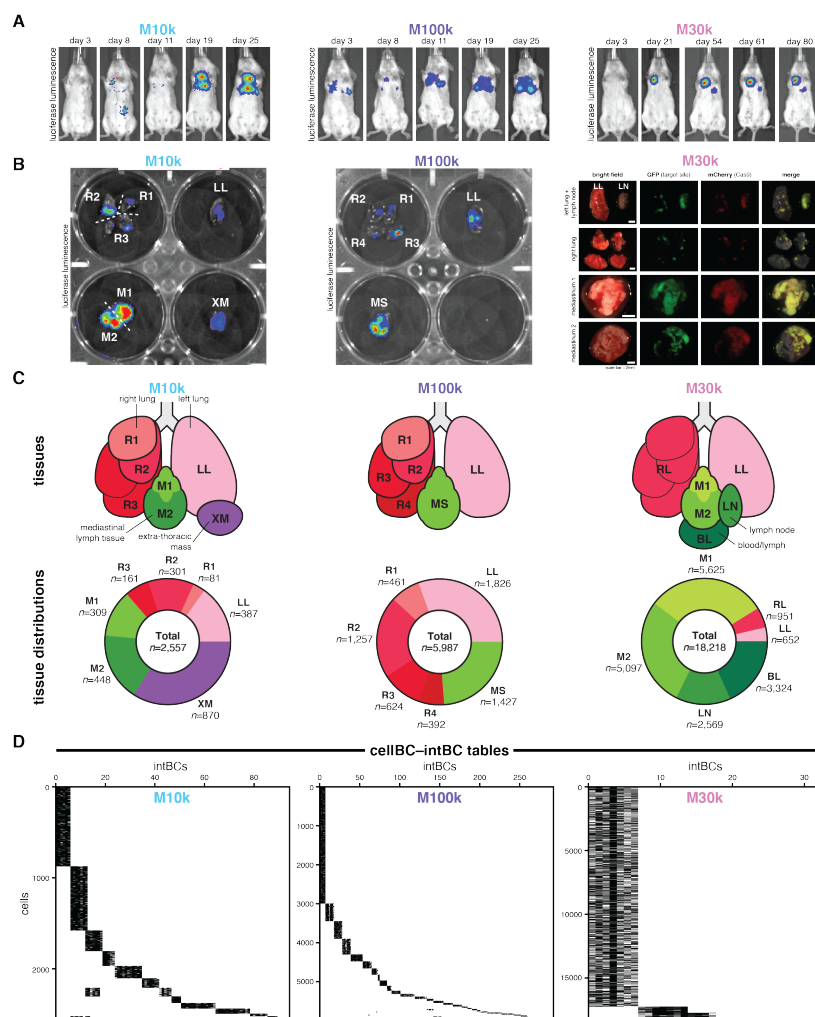


Figure 4.11: Tracing the cell lineages of metastatic progression in three additional mice. In addition to mouse M5k discussed in throughout the main text, we also traced the lineages of metastatic dissemination in three additional mice orthotopically xenografted with 10,000 (M10k, left), 100,000 (M100k, middle), and 30,000 (M30k, right) A549-LT cells in two cohort experiments (first cohort, A549-LT1: M10k and M100k; second cohort, A549-LT2: M5k and M30k). (A) In vivo bioluminescence imaging of cancer cell engraftment and metastatic spread at indicated times post-implantation. The mice were sacrificed 53, 67, and 80 days post-implantation, respectively. (B) Ex vivo imaging of tumorous tissues by bioluminescence (left and middle) or fluorescence (right) imaging with the tissue samples indicated. In addition to extensive tumors in the lungs and mediastinum, M10k had one large solid tumor located ventral to the left lung and embedded in the ribcage (called here an “extra-thoracic mass”; XM). (C) Anatomical representation of collected tumorous tissues (top) and the number of cells collected for each tissue and each mouse (bottom). Notably, M30k had a large lymph node (LN) on the left lung, as well as diffuse bloody lymph (BL) in the thoracic cavity upon sacrificing. (D) CellBC-intBC tables showing clonal populations of cancer cells in each mouse, as in **Figure 4.12A,B**. Some intBCs are shared between some clones (most notably in M10k) which likely resulted from cells that were clonally related at the stage of serial Target Site transduction during cell line engineering (Methods).

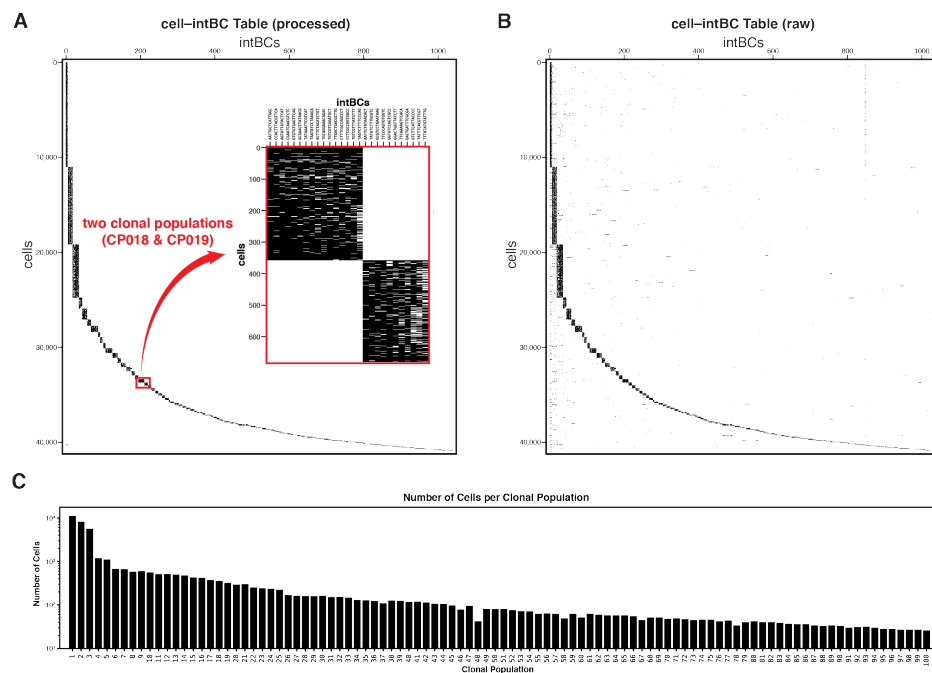


Figure 4.12: Identifying clonal populations by shared integration barcodes (intBCs). (A and B) Tables representing the >1,000 unique integration barcodes (intBCs; columns) and >40,000 cells (rows) observed in mouse M5k, after processing (A) and before processing (B). Because intBCs are clonally inherited, cells that share identical intBCs are grouped into clonal populations (here, black blocks). (A; inset) The set of intBCs is generally exclusive to a single clonal population, such as those observed in clonal populations #18 and #19 (CP018 and CP109). (B) In the raw, unprocessed cell-intBC table, cells may be associated with intBCs that are not in their defined clonal set (indicated here by density outside of the black blocks). Through the processing pipeline, these conflicting intBCs (and/or the cells to which they pertain) are identified as cell doublets, cell-free transcripts in emulsion droplets, or sequencing artifacts and removed (Methods). (C) The number of cells per clonal population, generally numbered by descending population size.

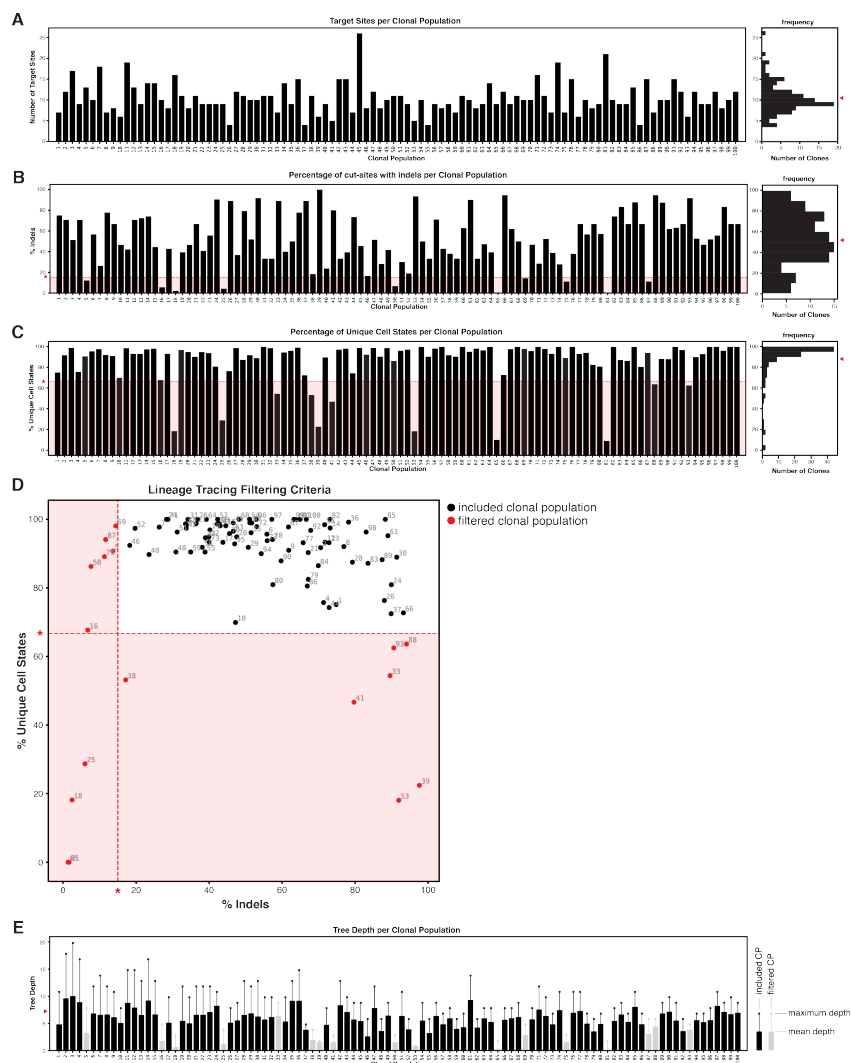


Figure 4.13: Characteristics of the lineage tracer and quality-control of clonal populations. (A) The copy-number of integrated Target Sites per clonal population, as determined by the number of unique intBCs. (A, right) The distribution of Target Site copy-number per clonal population; mean copy-number of Target Sites indicated (red arrowhead). (B) The percentage of cut-sites bearing lineage indel alleles per clonal population. Clonal populations with < 15% indels are excluded (red asterisk, red underlay). (B, right) The distribution of indel-bearing cut-sites per clonal population; mean percentage indicated (red arrowhead). (C) The percentage of unique cell lineage states per clonal population (i.e., lineage diversity). Clonal populations with < 66.7% diversity were excluded (red asterisk, red underlay). (C, right) The distribution of lineage diversity per clonal population; mean percentage indicated (red arrowhead). (D) Comparison of lineage tracing characteristics (% indels and % unique cell states) to define quality-control filtering criteria. Clonal populations removed by the filter (red closed circles) and filtering thresholds (red asterisks, red underlay) are indicated. (E) The depth of the reconstructed phylogenetic trees for each clonal population. Mean tree depths are shown as closed bars; maximum tree depths are shown as whiskers. Clonal populations that were excluded due to suboptimal lineage tracing characteristics (gray) and the average tree depth across all clonal populations (red arrowhead) are indicated.

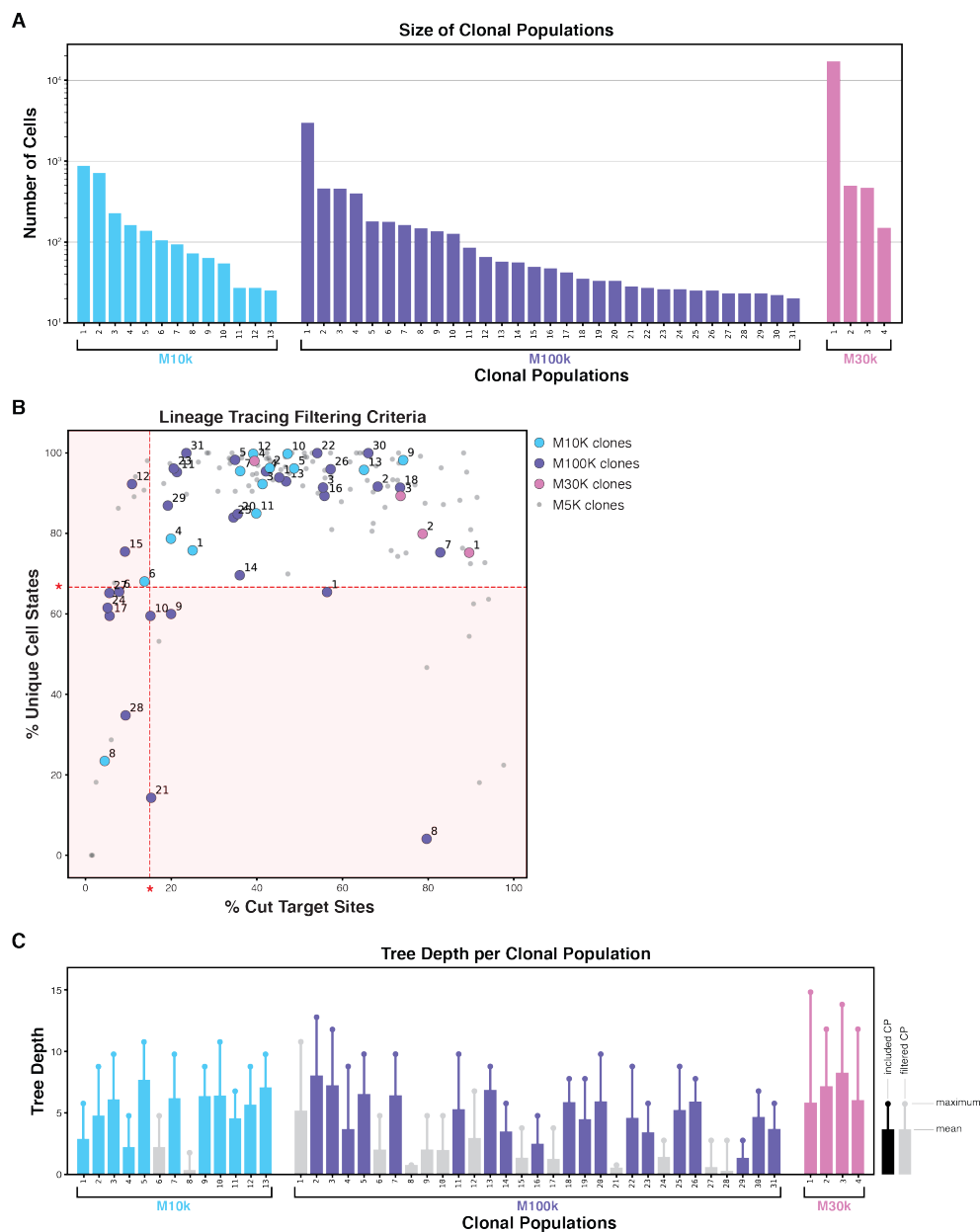


Figure 4.14: **Lineage tracing characteristics of clonal populations in additional mice.** (A) Number of cells in each clonal population in M10k, M100k, and M30k mice. (B) Scatter plot of the percentage of cut-sites bearing indels and the percentage of unique cell states per clonal population, which are characteristics of the lineage tracer that influence tree reconstructability. Some clonal populations exhibited suboptimal parameters (red asterisks and red field) and were excluded from reconstruction and downstream analyses. (C) The mean and maximum depths of the reconstructed phylogenetic trees for each clonal population in additional mice, as in Fig. 4.13E.

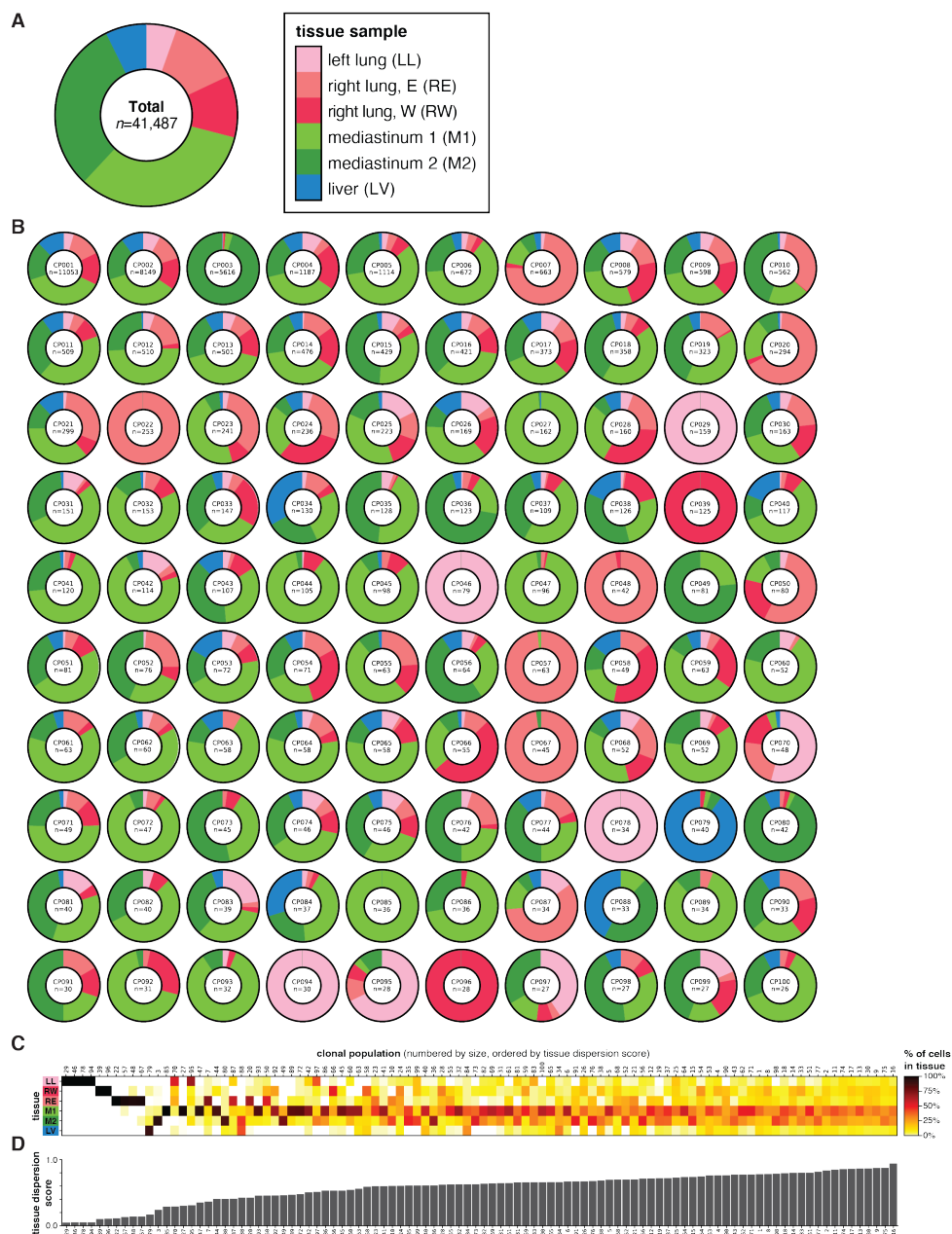


Figure 4.15: Clonal populations exhibit distinct tissue distributions. (A) The bulk distribution of all collected cells across the six tissue samples. (B) The distributions of cells from each clonal population across the six tissue samples. Some clonal populations were exclusive to the primary tissue (e.g., clonal population CP046), whereas some clones exhibited biased tissue distributions (e.g., CP003) and many others were observed broadly distributed across all tissues (e.g., CP011). (C) Tissue distributions of the largest 100 clonal populations. (D) The Tissue Dispersion Score is a statistical measurement of the distribution across tissues for each clonal population; x-axis shared with (C).

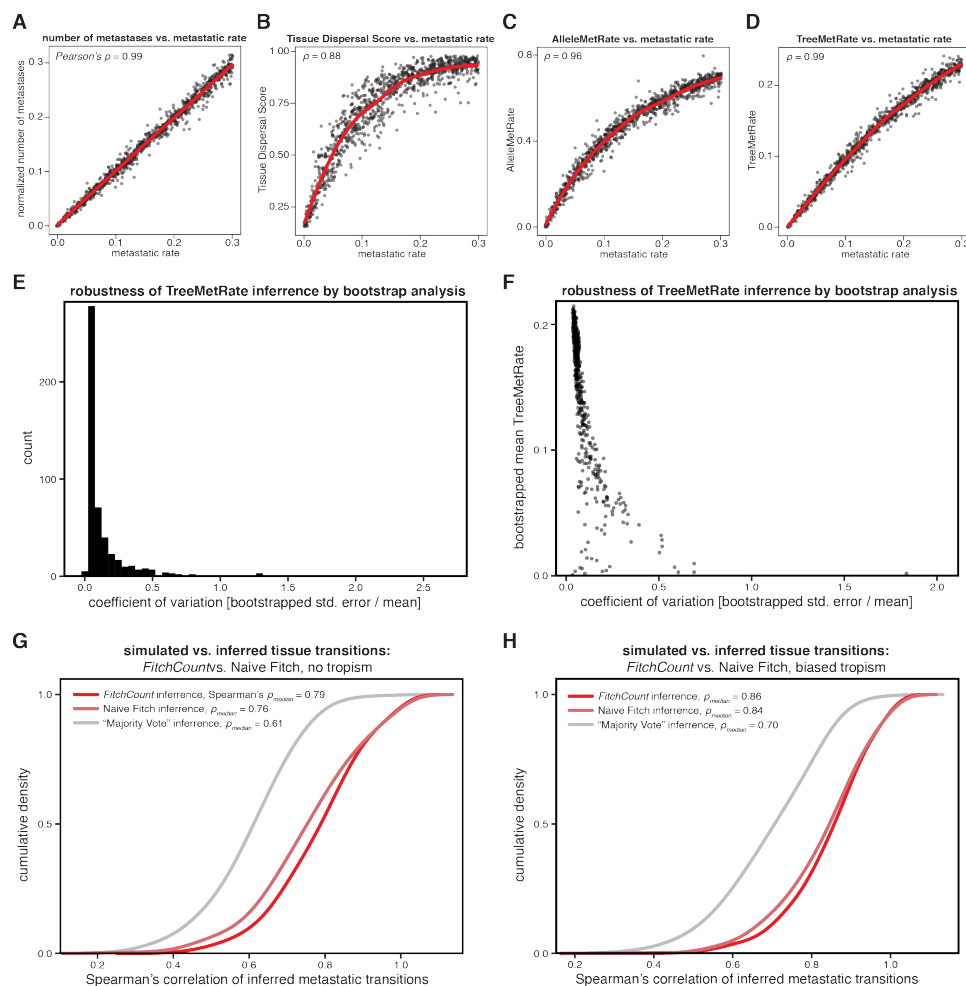


Figure 4.16: Assessing the accuracy of different measurements of metastatic rate and inference of tissue transitions using simulated lineages. (A–D) Comparison between the simulated metastatic rate and various lineage tracer-derived statistics, with the correlation (Pearson's ρ) indicated; red lines represent moving average. (A) The normalized count of simulated metastatic transitions is very well correlated with the simulated metastatic rates, and serves as a ground-truth benchmark. (B) The Tissue Dispersal Score, which is a statistical measure of how closely a clone's tissue distribution matches the background tissue distributions, is correlated with the metastatic rate, but saturates at intermediate metastatic regimes. (C) The AlleleMetRate, or the proportion of cells whose closest relative by allele similarity is in a different tissue, is better correlated with metastatic rate. (D) The TreeMetRate, or the proportion of inferred metastases in a reconstructed phylogeny, is the best lineage-derived measurement of metastatic phenotype by correlation. (E and F) Robustness of the TreeMetRate inference by bootstrap analysis. (E) The distribution of the coefficients of variation of the TreeMetRate across 50,000 simulated, bootstrapped phylogenies (Methods). The small coefficient of variation indicates that the TreeMetRate is a robust measurement. (F) The TreeMetRate coefficient of variation is only relatively large when the mean TreeMetRate is small. (G and H) Cumulative density plots assessing the accuracy of the FitchCount strategy for inferring ancestral tissue transition from simulated phylogenies with or without simulated biased tissue transitions (thus approximating tropism; Methods: "Assessing accuracy of the tissue transition matrices"). FitchCount outperforms other inference approaches, as measured by the Spearman correlation between the inferred and the ground-truth conditional tissue transition probability matrices. FitchCount was benchmarked against two other inference methods: (i) a "naive" single-solution implementation of the Fitch-Hartigan maximum parsimony algorithm or (ii) "Majority Vote", which infers ancestral tissue location as the tissue in which the majority of cells below each clade-level reside. Conditional probabilities for each algorithm were obtained by row-normalizing the count matrix with respect to the non-diagonal counts in each row.

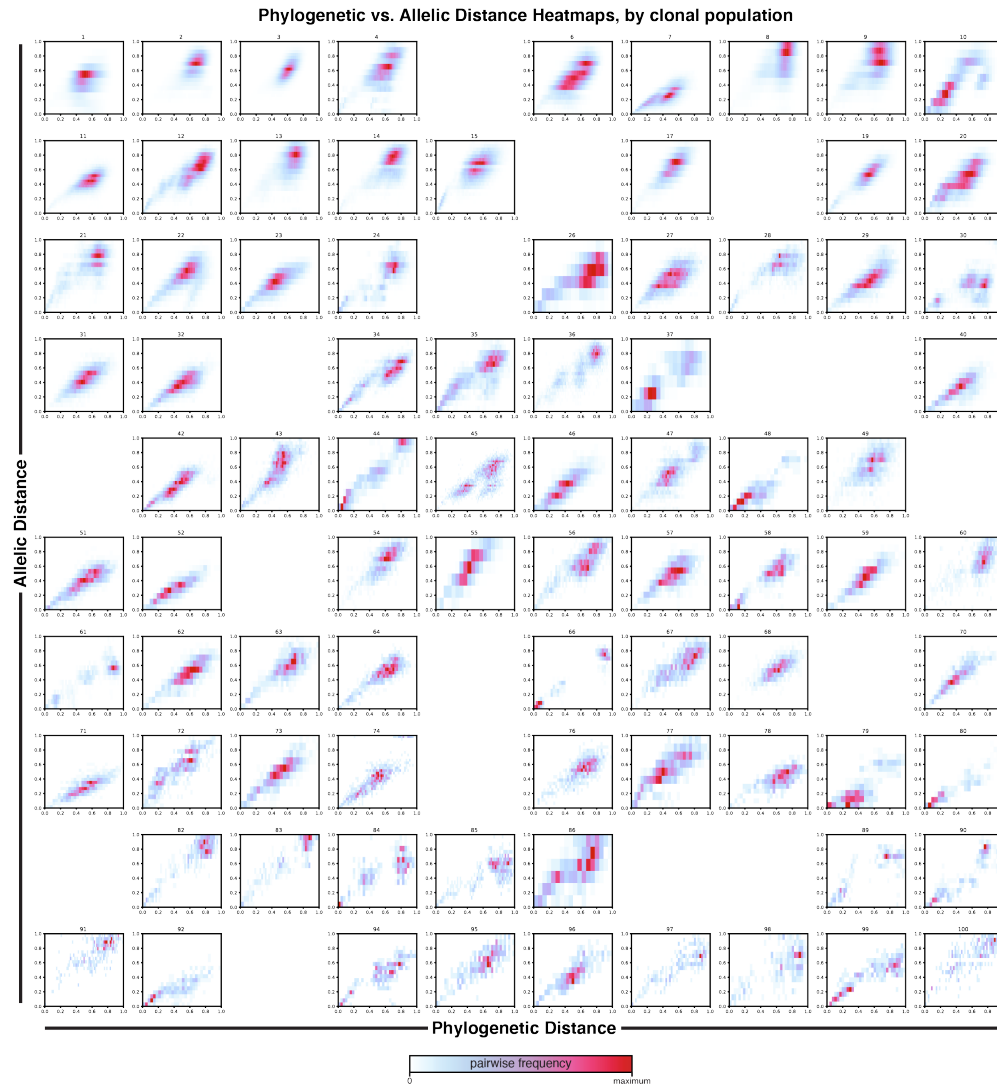


Figure 4.17: **Relationship between phylogenetic distance and allelic distance for each clonal population.** Density heat-maps comparing phylogenetic distance (i.e., the normalized tree branch distance between two cells) and allelic distance (i.e., the normalized difference in lineage allele state between two cells) for all pairwise cell–cell relationships and for each clonal population, as in Fig 4.2B. As expected, phylogenetic and allelic distances are correlated, suggesting that the reconstructed trees are a good phylogenetic model of cell–cell relationships. Excluded clonal populations are not shown.

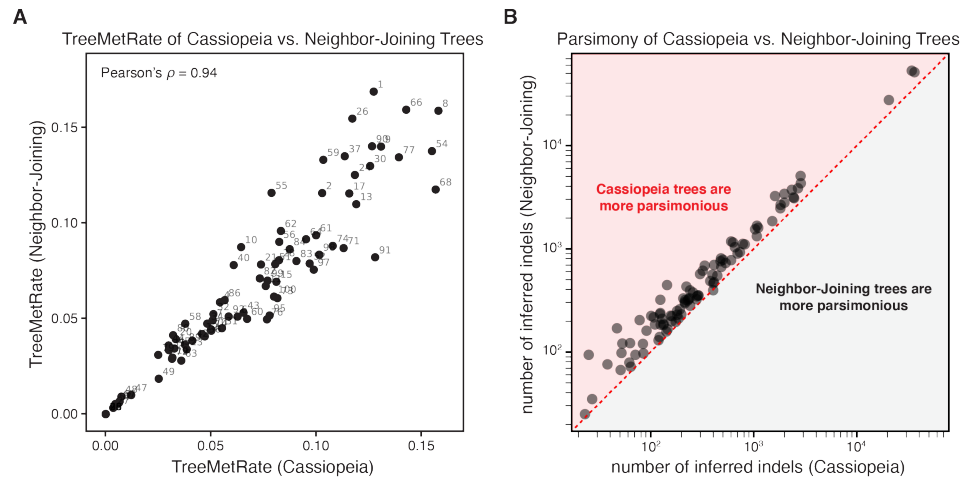


Figure 4.18: **The TreeMetRate is stable across tree reconstruction algorithms (Cassiopeia versus Neighbor-Joining).** (A) Comparison of the TreeMetRates for each clonal population from Cassiopeia trees and Neighbor-Joining trees. The TreeMetRates are correlated for both the Cassiopeia and Neighbor-Joining trees (Pearson's $\rho=0.94$). (B) Comparison of the parsimony of Cassiopeia trees and Neighbor-Joining trees, defined as the number of inferred indels in each tree. Notably, the Cassiopeia trees are more parsimonious than the Neighbor-Joining trees (i.e., they have fewer inferred indels; red overlay).

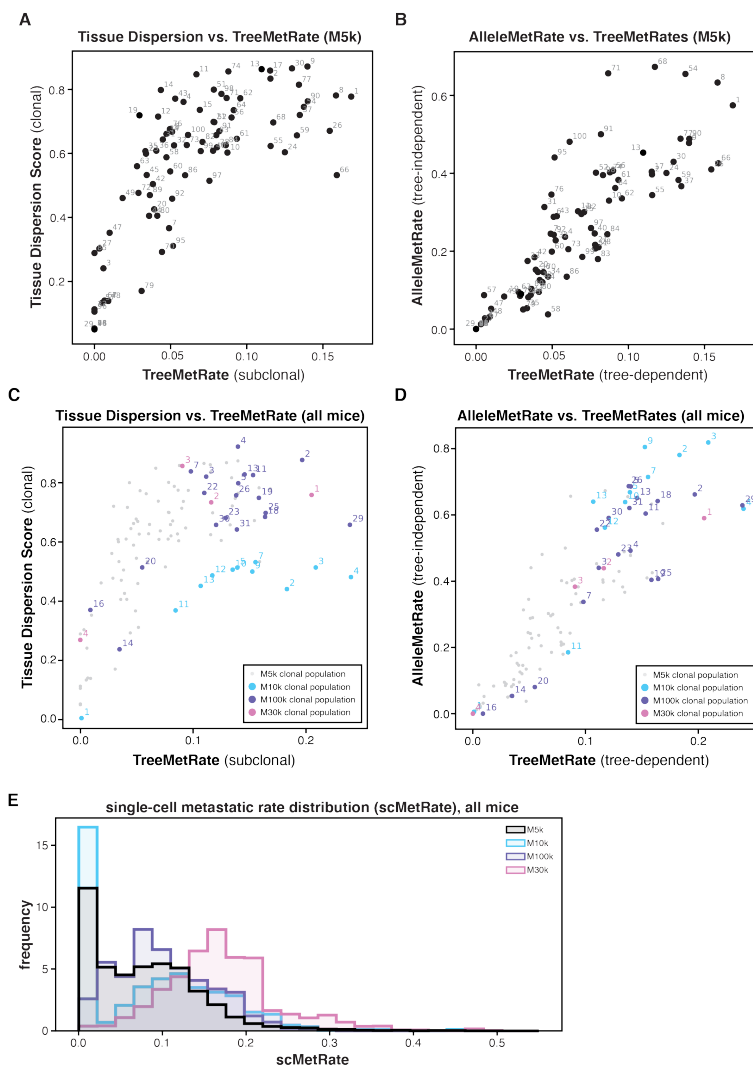


Figure 4.19: **Clonal populations exhibit broad metastatic phenotypes, measured by Tissue Dispersal Score, AlleleMetRate, and TreeMetRate.** (A–D) We evaluated the metastatic phenotype of each clonal population from mouse M5k (A, B) and the additional mice M10k, M100k, and M30k (C, D) using our three lineage-derived measurements: Tissue Dispersal Score (as in Fig 4.15D), AlleleMetRate, and TreeMetRate (as in Figure 4.4C). These three measurements follow similar relative trends across all four mouse experiments. First, the clonal populations exhibit a broad range of metastatic phenotypes. Second, all three measurements are correlated with one another, though simulations indicate that the TreeMetRate is the most accurate for estimating the underlying metastatic rate (Fig 4.169). Third, Tissue Dispersal Score saturates at intermediate metastatic regimes (A and C). (E) The distribution of single-cell-resolution metastatic rates (scMetRates) across all cells for each mouse (as in Fig 4.4D). Though all mice have broad distributions of metastatic phenotypes, mouse M5k (black) is particularly well represented by cells in low-to-intermediate metastatic regimes whereas mouse M30k (pink) has very few cells in the low metastatic regime.

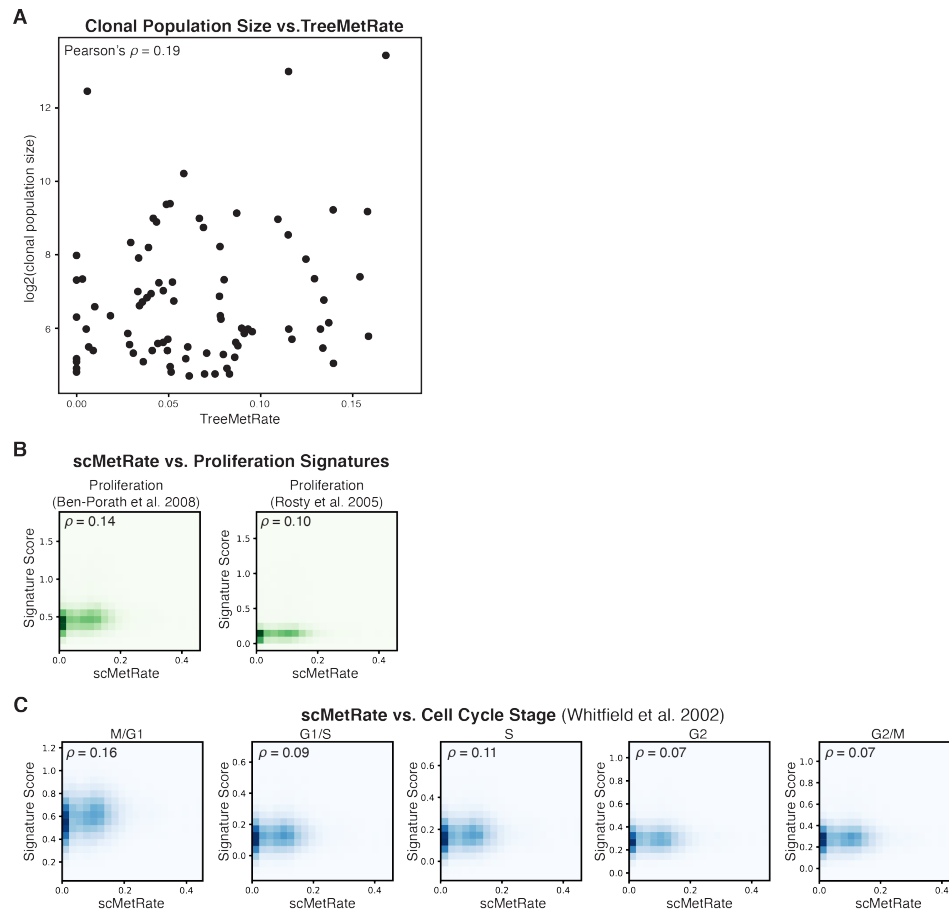


Figure 4.20: The scMetRate measures metastatic potential decoupled from proliferative capacity. (A) There is poor correlation between the scMetRate and the log₂-transformed clonal population size, a proxy for clonal fitness. (B and C) Density heat-maps comparing the scMetRate and various transcriptional signatures. Notably, the scMetRate is poorly correlated with transcriptional signatures of proliferation (B) nor stages of the cell cycle (C) [16, 278, 215]. Pearson's correlations (ρ) are indicated for each subplot.

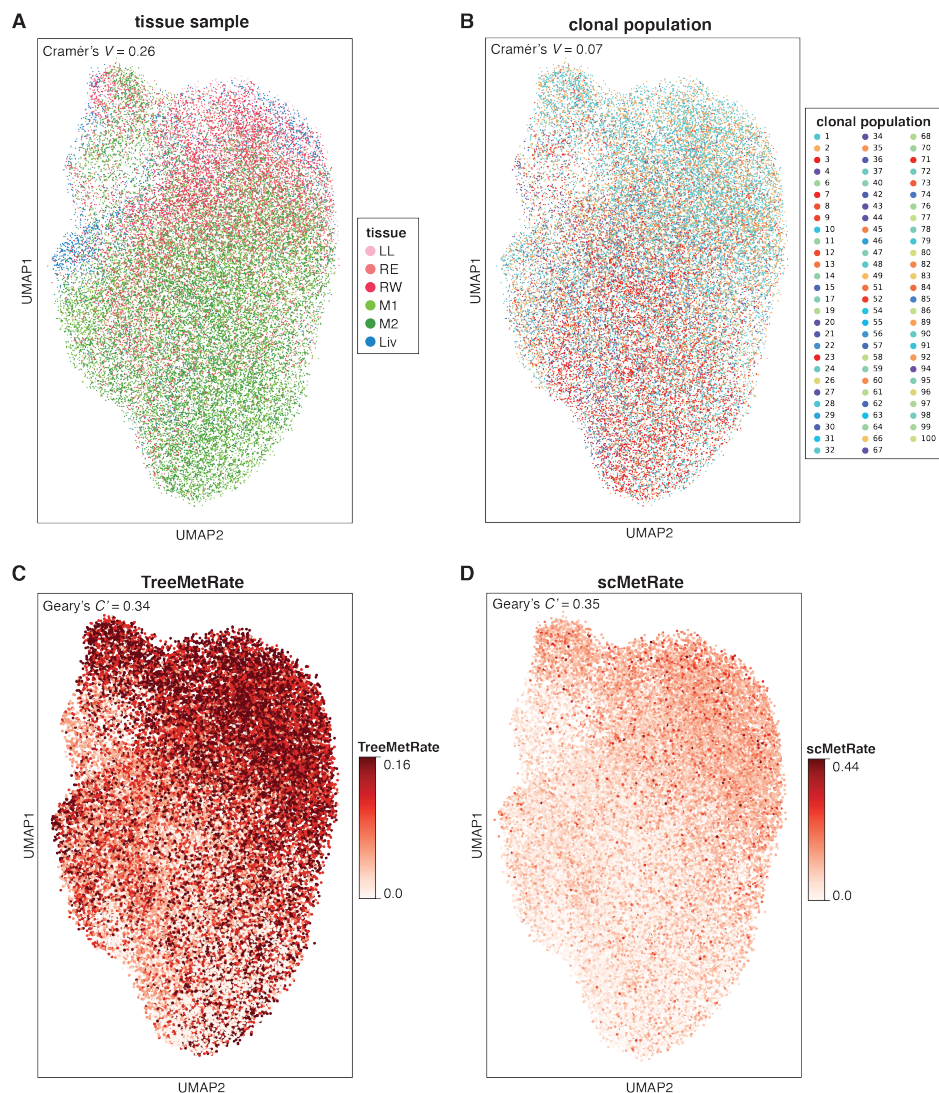


Figure 4.21: **The scMetRate measures metastatic potential decoupled from proliferative capacity.** To identify global trends in the gene expression data, we used Vision [58] to statistically assess the transcriptional effect of four features: (A) tissue sample, (B) clonal population identity, (C) TreeMetRate, and (D) scMetRate. The transcriptional states are represented here as a two-dimensional projection using Uniform Manifold Approximation and Projection (UMAP; B). Distinctions in transcriptional state are not predominantly explained by the clonal population (B; by Cramér's V), though there is modest association between transcriptional state and both tissue sample and metastatic phenotype (by Cramér's V and inverted Geary's C' , respectively).

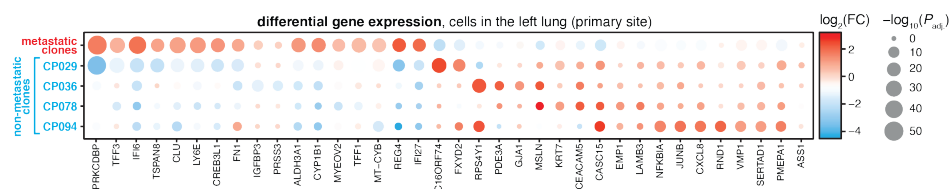


Figure 4.22: **Differential expression between non-metastatic and metastatic clonal populations in the primary tissue.** Differential gene expression analysis comparing four non-metastatic clonal populations (CP029, 36, 78, and 94) and all metastatic clonal populations in the primary tumor tissue (i.e., all other cells in the left lung). Significantly differentially expressed genes are colored by the \log_2 -transformed fold-change in gene expression and scaled by the adjusted Wilcoxon rank-sum test P -value.

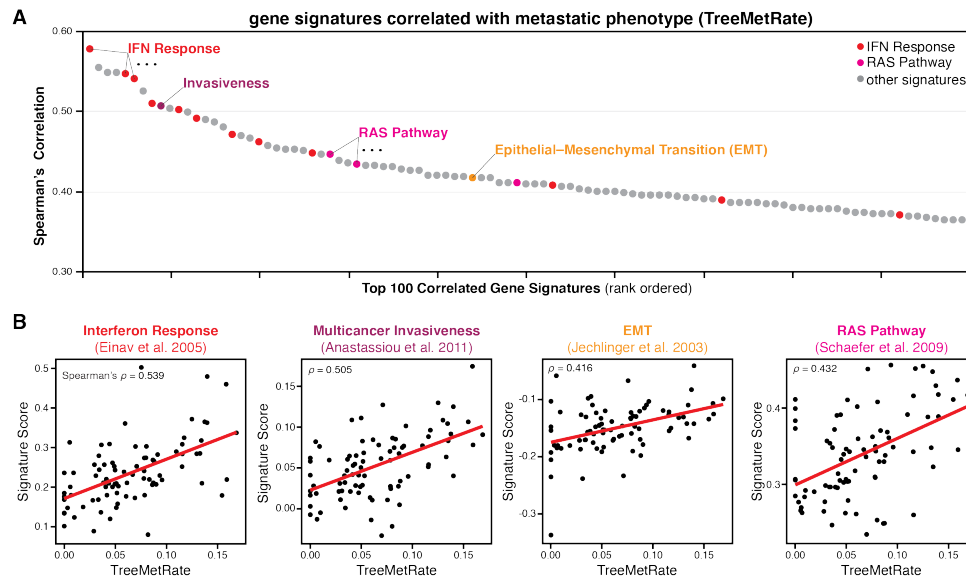


Figure 4.23: **Metastasis-related gene signatures are correlated with metastatic potential.** (A) Rank-ordered gene signatures that are the most positively correlated with TreeMetRate, including many related to interferon response (red) and RAS pathways (magenta), as well as other metastasis-related signatures. (B) Scatter plots showing the correlation between TreeMetRate and noted gene signature scores per clonal population; Spearman's correlation (ρ) indicated [65, 7, 125, 221].

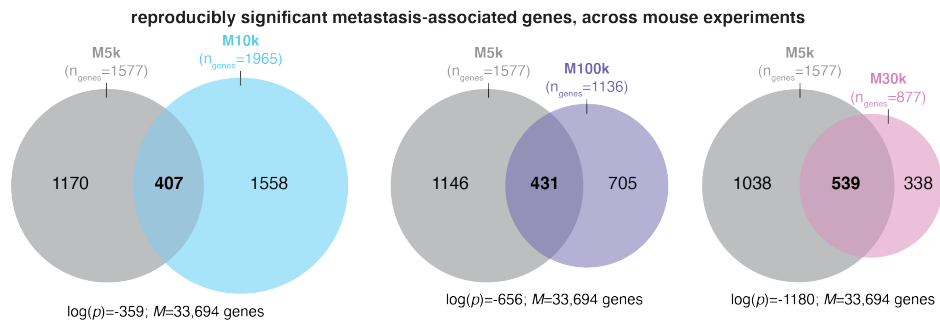


Figure 4.24: **Many of the same genes are associated with metastatic phenotype across all mice.** Using the same regression strategy as in the analysis of mouse M5k, we found many genes with expression that is significantly associated with high or low scMetRates. The number of significant genes for each mouse (FDR < 0.01) and their overlap in the same direction with mouse M5k (gray) are shown (n_{genes}). In all cases, the overlap between mouse M5k and each additional mouse is significant by hypergeometric test (p -value and M indicated).

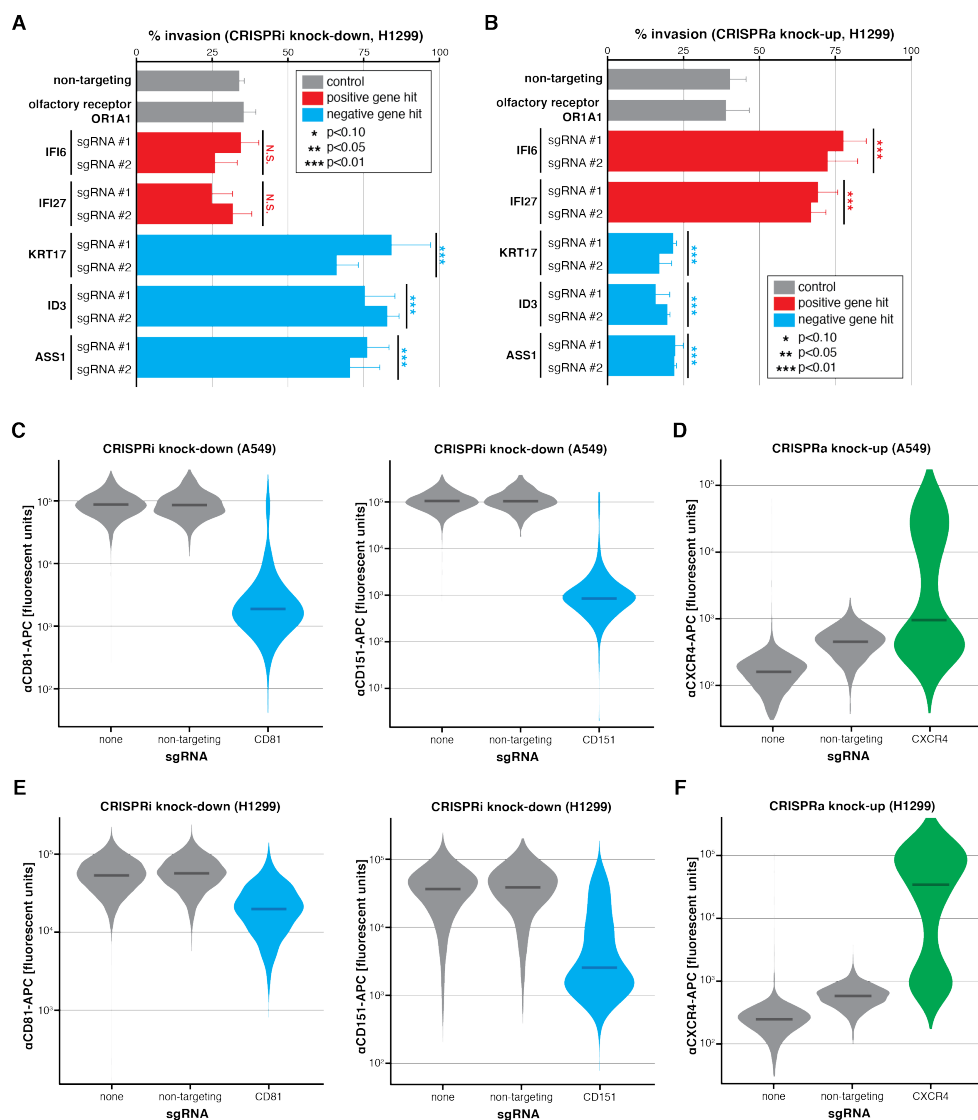


Figure 4.25: Functional validation of five gene candidates in a different cell line (H1299s) and validation of CRISPRi and CRISPRa activity. (A and B) *In vitro* transwell invasion assays following CRISPRi or CRISPRa gene perturbation, respectively, in H1299 cells; as in Fig 4.5E, F. Perturbation of positive and negative metastasis-associated gene candidates were performed in triplicate using two independent sgRNAs per gene. Differences in invasion phenotype relative to two negative control guides (non-targeting and olfactory receptor) were significant by two-tailed t-test. N.S., not significant; error bars show standard deviation. (C, E) A549-CRISPRi cells or H1299-CRISPRi cells, respectively, were treated with no sgRNA, non-targeting sgRNA, or sgRNAs against either CD81 or CD151, two highly expressed cell-surface markers. One week following treatment, the cells were collected, stained with APC-labelled anti-CD81 or anti-CD151 antibodies and their fluorescence was measured by flow cytometry. Shown here is substantial knock-down of CD81 or CD151 gene expression relative to no sgRNA or non-targeting sgRNA controls. (D, F) The same validation experiment as conducted in C and E, but for A549-CRISPRa cells or H1299-CRISPRa cells, respectively, treated with an sgRNA against CXCR4, a lowly expressed cell-surface marker. Shown here is substantially increased CXCR4 gene expression relative to no sgRNA or non-targeting sgRNA controls. Violin plots show distribution of fluorescent signal for each cell identified by flow cytometry; median marked by dark bar.

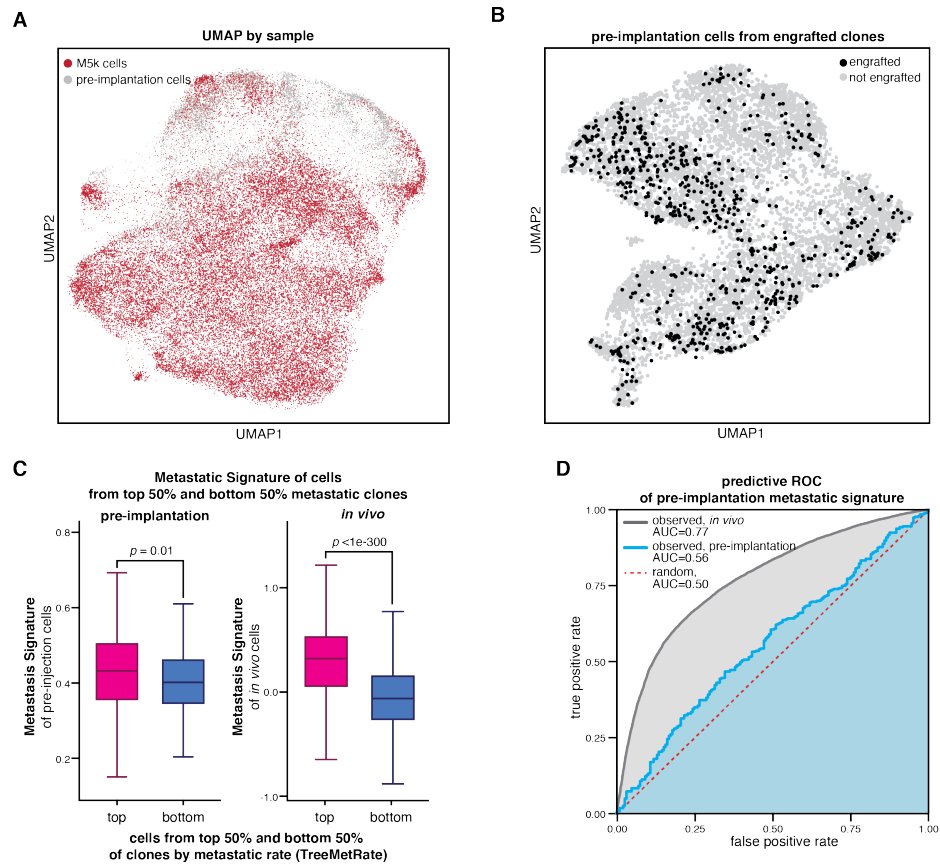


Figure 4.26: **Functional validation of five gene candidates in a different cell line (H1299s) and validation of CRISPRi and CRISPRa activity.** (A) Projection of transcriptional states of M5k and pre-implantation cells, colored by sample, as in Fig 4.6A. (B) Some of the cells from the pre-implantation pool could be assigned to the 100 clonal populations that engrafted and proliferated in mouse M5k based on their clonal barcodes (i.e., intBCs). Shown are the pre-implantation cells that could be assigned to an engrafted clone (black) on a projection of pre-implantation transcriptional states, as in Fig 4.6B,C. (C, left) Pre-implantation cells from the top 50% (most) metastatic clones *in vivo* have higher Metastatic Signature scores than pre-implantation cells from the bottom 50% (least) metastatic clones *in vivo* (Mann-Whitney U p -value=0.01). (C, right) The Metastatic Signature of the most and least metastatic clones is more pronounced *in vivo* than in the pre-implantation cells (p -value<1e-300). (D) For the pre-implantation cells, the difference in Metastatic Signature scores between the most and least metastatic clones is modest, yet significant by ROC (receiver operator characteristic) analysis of false positives vs. true positives (area under the curve, AUC=0.56), indicating that the Metastatic Signature score pre-implantation is a modest predictor of *in vivo* metastatic phenotype. The predictive power for the *in vivo* population of cells is greater (AUC=0.77).

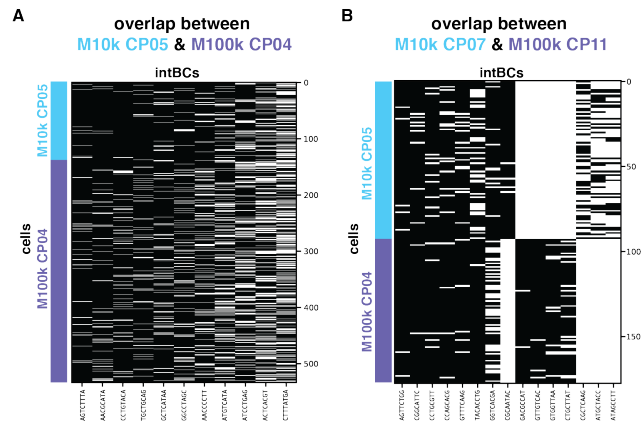


Figure 4.27: **Two pairs of clonal populations from mice M10k and M100k are related, enabling an experiment to determine the robustness and reproducibility of metastatic phenotype across independent mouse experiments.** Each intBC (columns) observed for each cell (rows) from (A) M10k CP05 and M100k CP04 and (B) M10k CP07 and M100k CP11. Cells from M10k are shown in light blue; M100k in purple. The clonal populations in each of these pairs are related to one another based on their shared sets of intBCs, as in Fig 4.6D. The paired clonal populations here are the most closely related between the two mouse experiments.

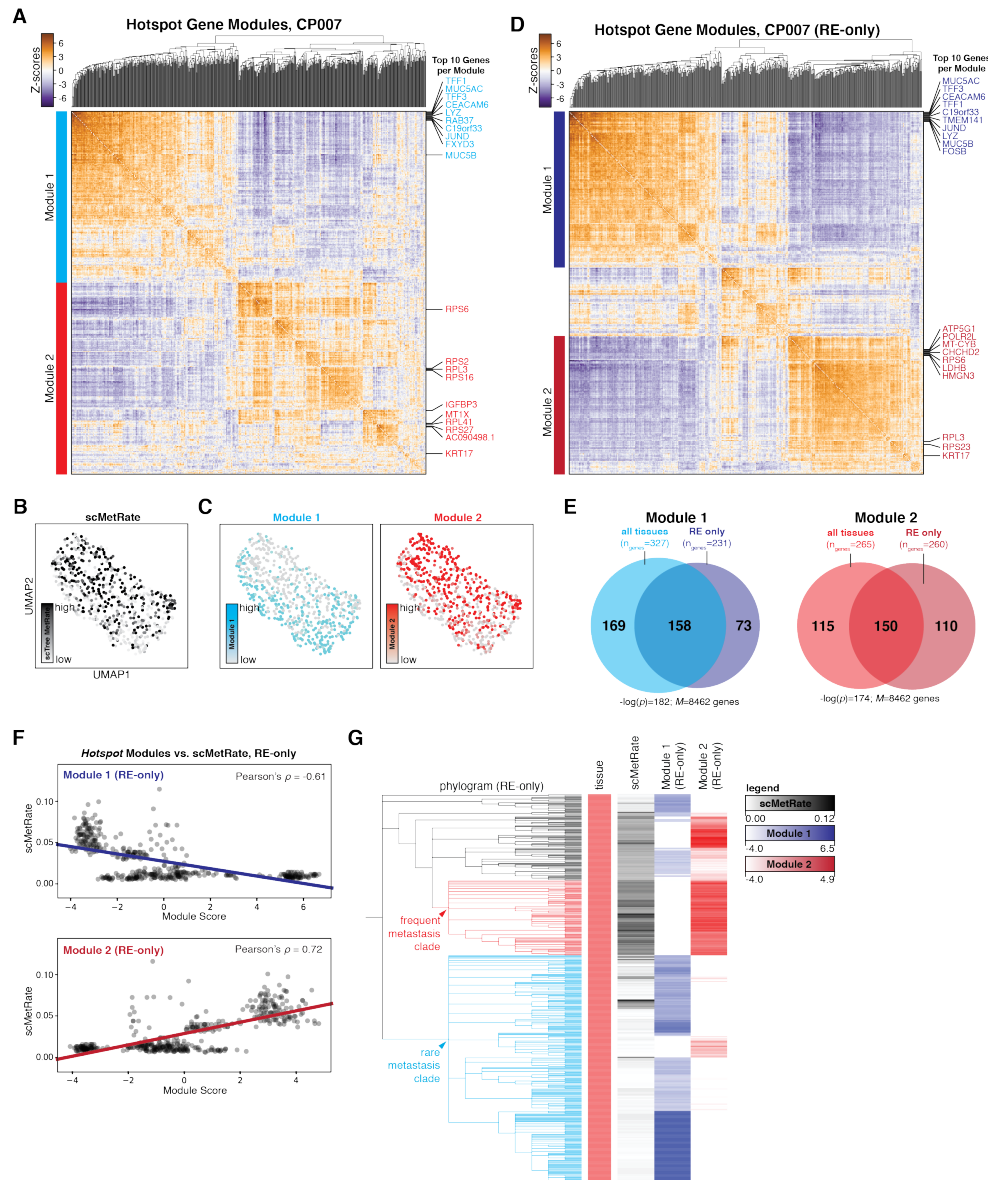


Figure 4.28: Distinct transcriptional modules underlie distinct clade-specific metastatic behaviors in Clone #7. Hotspot analysis identifies two gene modules that have heritable expression patterns in CP007 (Modules 1 and 2; indicated in cyan and red, respectively). Pairwise local correlations of genes (with FDR < 0.1) calculated with Hostspot are shown by heat-map; the top 10 most significant genes each for Modules 1 and 2 are annotated in cyan and red, respectively. (B and C) The transcriptional states for all cells in CP007 represented in a two-dimensional projection (UMAP) and colored by scMetRate (B) or Hotspot Module scores (C). (D) When restricting Hotspot analysis to only cells from the “RE” tissue sample in CP007, two heritable gene modules are identified (dark blue and dark red). As before, the top 10 genes from each module are annotated. (E) The gene modules identified from all cells or from only RE-only cells in CP007 overlap significantly by hypergeometric test. Number of genes in each set (n_{genes}), number of expressed genes in at least 10% of cells in CP007 (M), and p -value of the hypergeometric test are indicated. (F) RE-only Modules 1 and 2 are negatively and positively associated with the scMetRate, respectively, as in the analysis for all cells from CP007 (Fig 4.6I). (G) Overlay of the phylogram for RE-only cells from CP007, scMetRate, and Hotspot module scores (RE-only), showing concordance between the frequently metastasizing clade (red) and Module 2 and the rarely metastasizing clade (cyan) and Module 1, as in Fig 4.6J.

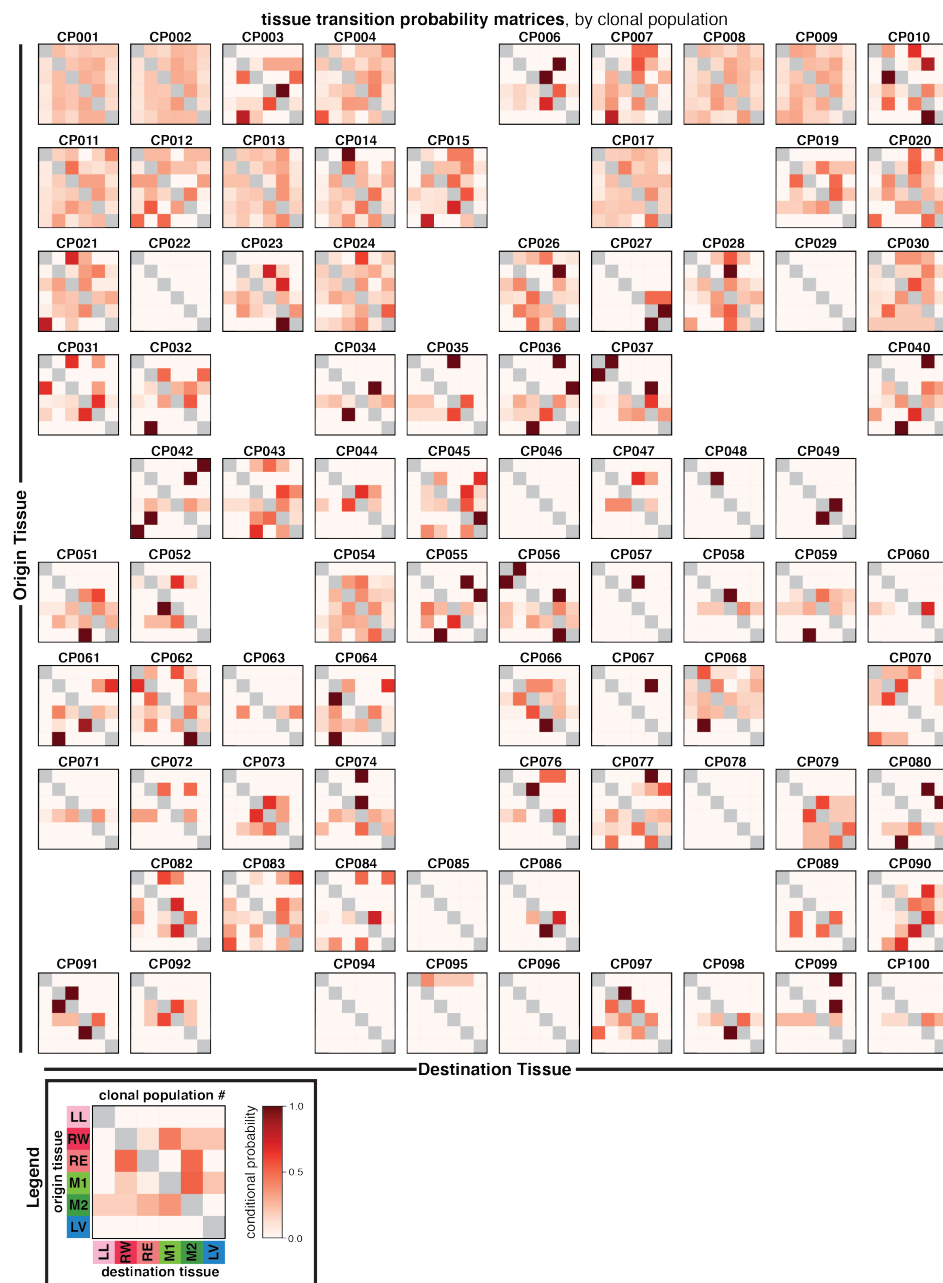


Figure 4.29: **Tissue transition probability matrices for each clonal population.** The conditional probability of transition from and to each tissue inferred from the phylogenetic trees (calculated with *FitchCount*) of each clonal population, thus summarizing the most probable tissue routes of metastasis. Legend (lower left) indicates the color bar showing conditional probability and the tissue labels, as in Fig 4.1E. Notably, the transition matrices are varied and distinct to each clonal population.

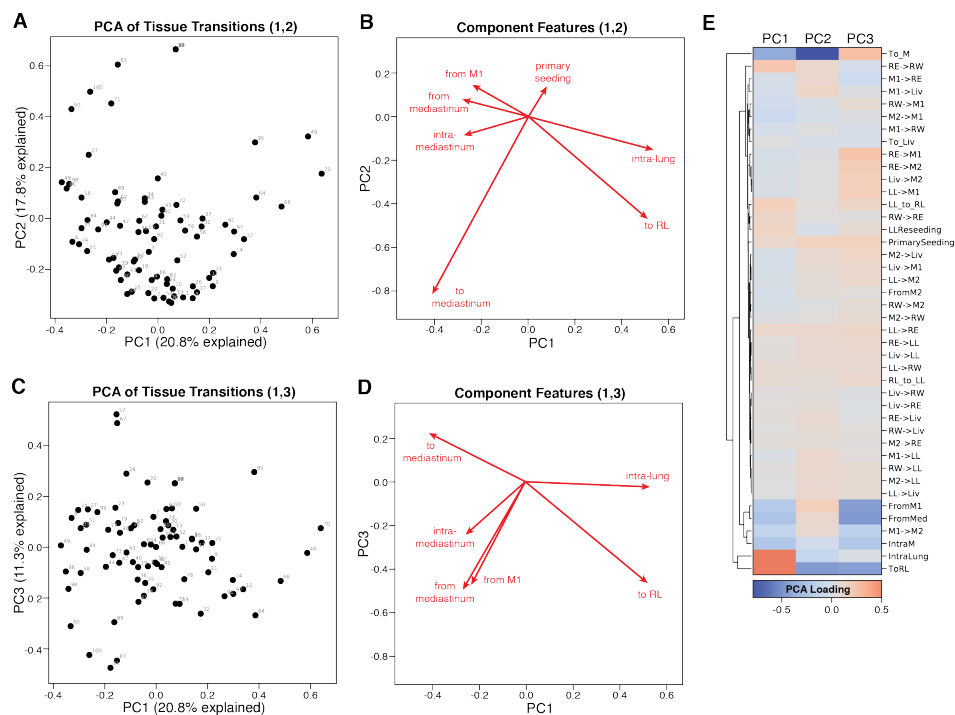


Figure 4.30: **Describing the principal features of metastatic seeding routes.** (A, C) PCA projections of the metastatic tissue transitions for each clonal population (annotated). The percentage of the variance explained by each component is indicated on the axes for the first and second (A) or first and third (C) components. (B, D) Biplot vectors representing the most explanatory features of the first, second, and third principal components, annotated by descriptive features of metastatic transitions. The length and angle of the vector describe the scale and direction, respectively, of each descriptive feature. (E) The PCA loadings of the metastatic transition features for each principal component.

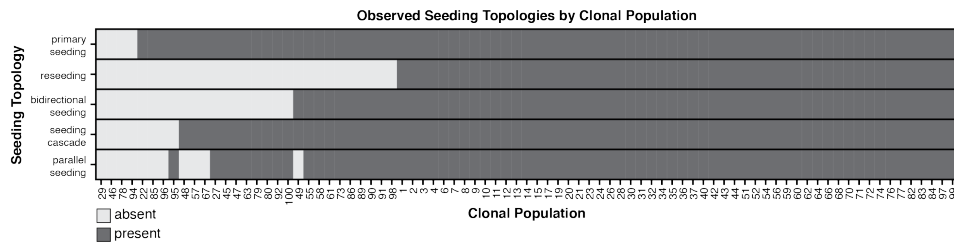


Figure 4.31: **Seeding topologies observed in each clonal population.** A table describing classified seeding topologies (rows) that are present or absent (dark or light gray, respectively) in each clonal population (columns). The majority of clonal populations exhibit examples of all seeding topologies.

Chapter 5

Lineage Tracing Reveals the

Phylodynamics, Plasticity and Paths of

Tumor Evolution

5.1 Abstract

Tumor evolution is driven by the progressive acquisition of genetic and epigenetic alterations that enable uncontrolled growth, expansion to neighboring and distal tissues, and therapeutic resistance. The study of phylogenetic relationships between cancer cells provides key insights into these processes. Here, we introduced an evolving lineage-tracing system with a single-cell RNA-seq readout into a mouse model of *Kras;Trp53*(KP)-driven lung adenocarcinoma which enabled us to track tumor evolution from single transformed cells to metastatic tumors at unprecedented resolution. We found that loss of the initial, stable alveolar-type2-like state was accompanied by transient increase in plasticity. This was followed by adoption of distinct fitness-associated transcriptional programs which enable rapid expansion and ultimately clonal sweep of rare, stable subclones capable of metastasizing to distant sites. Finally, we showed that tumors develop through stereotypical evolutionary trajectories, and perturbing additional tumor suppressors accelerates tumor progression by creating novel evolutionary paths. Overall, our study elucidates the hierarchical nature of tumor evolution, and more broadly enables the in-depth study of tumor progression.

5.2 Introduction

Cancer is an evolutionary process characterized by the dynamic interplay of cellular subpopulations, each driven by progressive genetic and epigenetic changes [188]. Throughout this process, cancer cells can acquire phenotypic heterogeneity that increases fitness by enabling them to grow more aggressively, invade neighboring tissues, evade the immune system and therapeutic challenges,

and metastasize to distant sites ([101, 176, 267]). Interrogating the molecular bases of subclonal selection and metastatic seeding, the origins of and transitions between transcriptional states, as well as the identities and genetic determinants of evolutionary paths that tumors undergo will not only illuminate fundamental principles governing tumor evolution, but also have immediate clinical implications [22]. To fully understand these processes, it is essential to study the evolutionary dynamics giving rise to a tumor in its native setting, preferably in experimentally defined conditions [6].

Tumor phylogenetic analysis, the study of lineage relationships among the cells comprising the tumor population descended from a single transformed progenitor, can provide key insights into the dynamics of tumor progression. Classically, phylogenies have been constructed using naturally-occurring somatic genomic variations (mutations or copy-number variations [CNVs]) as natural lineage tracers, and have yielded key insights into the evolutionary dynamics of human tumor development [268, 230, 223, 166, 82, 89, 235]. These efforts have illuminated several key evolutionary processes underpinning tumor development, including the acquisition of critical subclonal genetic or epigenetic changes [87, 279, 185], the timing and routes of metastatic dissemination [263, 115], and the development of therapeutic resistance [172, 200, 1, 141, 217]. While progress has been enabled by innovative computational methods [199, 136, 169, 220], these studies are limited by the inherent variation in naturally-occurring somatic mutations, incomplete or low cell sampling, and other confounding variables (e.g. environmental exposures and genetic background), and are not amenable to further perturbations or functional studies.

Genetically engineered mouse models (GEMMs) of cancer provide a critical tool for modeling

tumor progression as they allow one to study tumor evolution in its native microenvironment and experimentally defined conditions [102, 78]. The $Kras^{LSL-G12D/+}; Trp53^{fl/fl}$ (KP) model of lung adenocarcinoma allows tumor initiation via viral delivery of Cre recombinase to a small number of lung epithelial cells, leading to activation of oncogenic *Kras*, homozygous deletion of the *p53* tumor suppressor gene, and clonal tumor outgrowth. It faithfully models the major steps of tumor evolution from nascent cell transformation to aggressive metastasis, recapitulating human lung adenocarcinoma progression both molecularly and histopathologically [120, 121, 280]. Moreover, recent work has revealed that substantial transcriptomic and epigenomic heterogeneity emerges during tumor evolution in this model [170, 151], consistent with human tumors [157]. The tractability of this model provides an appealing opportunity to probe several unanswered, but crucial tumor evolutionary-related questions: how a single transformed cell expands into an aggressive tumor, how various cell states relate to one another and contribute to tumor evolution, how different transcriptional states transition to each other, and how metastases and primary tumors are evolutionarily related.

Approaches that permit simultaneous measurements of cell lineage and cell state information have the potential to provide unique insights into these questions [252, 269, 238]. While previous studies have used synthetic "static" barcoding techniques to study clonal relationships [19, 163, 153, 195, 60, 222], studying the evolution of individual tumors at subclonal resolution remains challenging. This limitation is in large part due to the low mutational burden in GEMM tumors, thus offering little lineage resolution within individual tumors [277, 174]. The recent development of high resolution CRISPR/Cas9 evolving lineage tracing paired with single-cell RNA-seq (scRNA-seq) readouts overcomes these limitations. Generally, this continuous lineage-tracing technology leverages Cas9-

induced DNA cleavage and subsequent repair to progressively generate heritable insertions and deletions (“indels”) at synthetic DNA target sites engineered into the genomes of living cells [178, 79, 134, 37, 177]. Importantly, these DNA target sites are transcribed into poly-adenylated mRNAs, allowing them to be captured and profiled along with all other cellular mRNAs using droplet-based scRNA-seq. In doing so, this approach makes it possible to directly link the current cell state (as measured by scRNA-seq) with its past lineage history (as captured by the lineage tracer), and to do so on a massive scale [4, 236, 206, 37, 24]. Recently, this technology has been introduced into cancer cell lines before transplanting them into mice to track metastatic behaviors in vivo [228, 203, 295].

Here, we have developed an autochthonous “KP-Tracer” mouse model which allows us to simultaneously initiate an engineered lineage tracing system and induce *Kras* and *Trp53* oncogenic mutations in individual lung epithelial cells. This enabled continuous and comprehensive monitoring of the processes by which a single cell harboring oncogenic mutations evolves into an aggressive tumor. The resulting tumor phylogenies reveal that rare but consequential subclones drive tumor expansion by adopting distinct fitness-associated transcriptional programs. By integrating lineage and transcriptome data, we uncovered changes in cancer cell plasticity and parallel evolutionary paths of tumor evolution in this model, which could be profoundly altered by perturbing additional tumor suppressor genes commonly mutated in human tumors. We have also identified the subclonal origins, spatial locations and cellular state of metastatic progression. Collectively, this technology allowed us to reconstruct the lifespan of a tumor from a single transformed cell to a complex and aggressive tumor population at unprecedented scale and resolution.

5.3 Results

5.3.1 KP-Tracer mouse enables continuous and high-resolution lineage tracing of tumor initiation and progression

To generate high-resolution tumor phylogenies, we developed a lineage-tracing competent mouse model of lung adenocarcinoma capable of months-long continuous cell lineage tracing (Figure 5.1A). Specifically, we engineered mouse embryonic stem cells (mESCs) harboring the conditional alleles *Kras*^{LSL-G12D/+} and *Trp53*^{fl/fl} (KP) to additionally encode conditional SpCas9 and mNeonGreen fluorophore at the Rosa26 locus; *Rosa26*^{LSL-Cas9-P2A-mNeonGreen} (KPCas9). We then engineered these mESCs with a refined version of our lineage tracing technology [37, 203]. Specifically, we introduced a library of piggyBac transposon-based lineage tracing vector containing two essential components: first, target sites for lineage tracing, consisting of three cut sites positioned within the 3' UTR of a mCherry fluorescent reporter and a 14-base-pair randomer integration barcode ("intBC") to distinguish individual copies; and second, three constitutively expressed single-guide RNAs (sgRNAs) for directing Cas9 to each of the three individual cut-sites within the target sites, thereby generating indels for lineage tracing (**Figure 5.8A**). A key enabling feature is that the speed of tracing (i.e., indel generation kinetics) can be tuned to match the tumor developmental timescale by engineering mismatches between sgRNAs and target sites [37, 203]. We isolated engineered mESC clones by fluorescence activated cell sorting (FACS) based on high mCherry expression (**Figure 5.8B-C**) and selected clones with 10-30 integrated target sites by quantitative PCR (qPCR)

and DNA sequencing (Figure 5.8D-E). Finally, we generated chimeric mice (hereafter “KP-Tracer” mice) from five validated mESC clones to ensure evolutionary behavior was not idiosyncratic to a specific clone [299, 201].

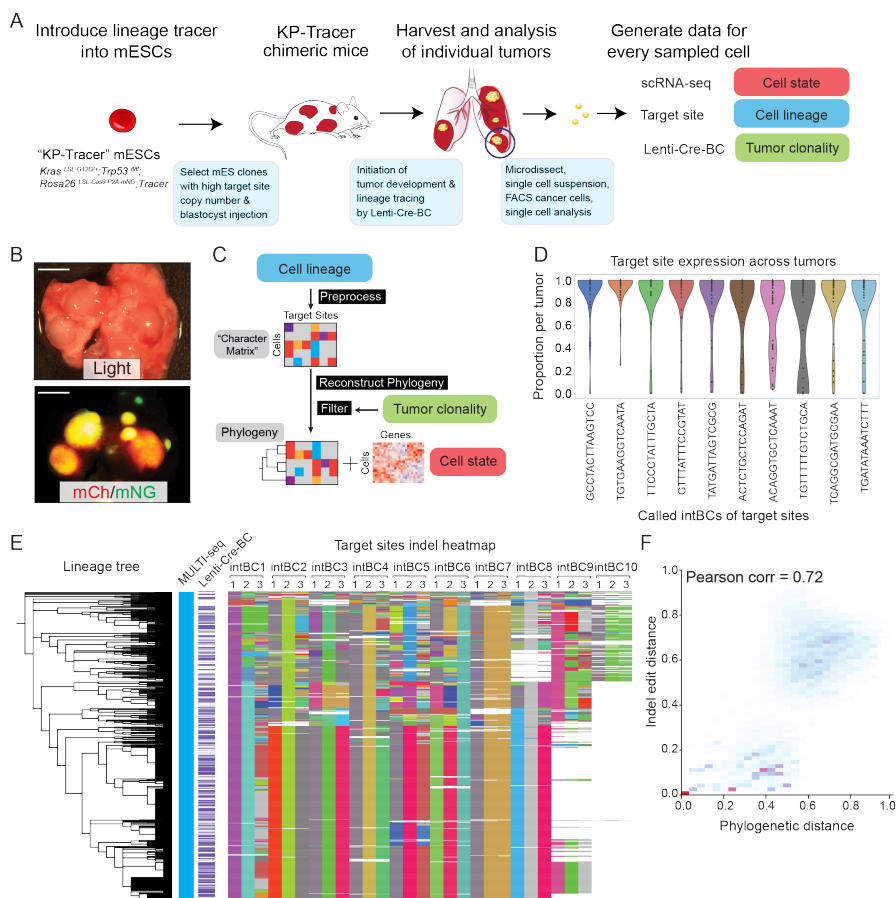


Figure 5.1: **KP-Tracer mouse enables continuous and high-resolution lineage tracing of tumor initiation and progression.** (A) Generation of the KP-Tracer chimeric mouse and initiation of KP-Tracer tumors (see Methods). Five to six months after tumor initiation, individual tumors are dissociated into single cell suspension and single cell sequencing libraries are prepared. (B) Representative images of tumors from KP-Tracer mouse. Tumors are positive for *mCherry* and *mNeonGreen*. Scale bars = 5 mm. (C) Tumor lineage reconstruction data analysis pipeline. (D) Target site capture efficiency across tumors from mice generated from one representative mESC clone (2E1). Dots represent the average capture rate of a specific target site in a tumor. (E) Phylogeny with MULTI-seq, lenti-Cre-BC, and target site information for an example tumor. Each row represents a single cell and each column indicates barcode or target site information (ordered by the percentage of target sites detected across cells). Unique colors represent unique barcodes or indels, uncut sites are shown in light-gray, and missing data is indicated in white. (F) Comparison of phylogenetic distance (from the reconstructed tree) and allele edit distance (from target sites) for the example tumor in (E).

In KP-Tracer mice, intratracheal administration of lentivirus expressing Cre recombinase simul-

taneously initiates lung tumors by activating conditional oncogenic alleles and lineage tracing by inducing the expression of Cas9 which together with the expressed sgRNAs causes accumulation of indels in the target sites [61]. Previous static lineage tracing studies, using lentiviral barcoding or multi-color reporters, have shown that KP tumors induced with this strategy are clonal and homogeneously contain oncogenic *Kras;p53* mutations [44, 32]. To validate tumor clonality, we induced tumors with a barcoded lentiviral-Cre construct (lenti-Cre-BC) providing a unique clonal barcode for each tumor [3].

Individual tumors with strong mCherry and mNeonGreen expression (indicating target site and Cre, respectively) and clear boundary separation from adjacent tumors were harvested 5-6 months after tumor initiation, microdissected, and dissociated completely to ensure unbiased cell sampling (**Figure 5.1B**). After being labeled with Multiplexing Using Lipid-Tagged Indices for scRNA-seq (MULTI-seq) [175] and purified by FACS (see Methods), cancer cells were subjected to scRNA-seq analysis to measure cell state, lineage, sample identity, and tumor clonality. After integrating all four datasets for each cell (**Figure 5.1C**; see Methods), we proceeded with paired lineage and transcriptome measurements for 40,386 cells with a median of 9,680 UMIs and 2,877 genes detected across 35 tumors (29 primary tumors and 6 metastases; a median of 511 cells were detected per primary tumor). Importantly, target sites were consistently expressed across tumors (**Figure 5.1D, 5.8F-G**).

After preprocessing target site data based on lineage-tracing sequencing quality control and ensuring tumor clonality with lenti-Cre-BC information (**Figure 5.1C**; see Methods), we reconstructed phylogenies for each tumor with Cassiopeia [127]. **Figure 5.1E** displays the inferred phylogeny and its corresponding indel status (summarized in an “allele heatmap”) of a single representative tumor,

consisting of 772 cells. The resulting tree revealed a rich subclonal structure and deep lineage relationships, with a median depth of 12 and maximum depth of 15. As a validation of the integrity of our lineage reconstruction, we observed strong correlations between phylogenetic and allelic distances across our trees (**Figure 5.1F**). With these high-resolution tumor phylogenies, we next turned to studying the relationship between subclonal dynamics and cellular state as determined by gene expression.

5.3.2 Rare subclones expand during tumor progression, marked by increased DNA copy number variation, cell cycle score, and fitness score

A key question in tumor evolution is how subclonal selection, based on the acquisition of growth-promoting genetic or epigenetic changes, and the resulting population dynamics lead to the expansion of aggressive subclones relative to other parts of the same tumor [188, 176, 53, 235]). To examine the subclonal dynamics in KP tumors, we adapted a statistical test that compares the relative size of each subclone to what would be expected in a “neutral” model of evolution where no subclone is under selection [94, 237](see Methods). Using this method on a high-quality subset (21/29) of primary tumors (**Figure 5.8H**; see Methods), we found examples of tumors that appeared to be neutrally evolving (i.e., with no evidence for positive selection) and tumors with subclones showing clear signs of positive selection (**Figure 5.2A**). Tumors predominantly had one or sometimes two subclones undergoing expansion, and across tumors there was a broad distribution in the proportion

of cells within expansions (**Figure 5.2B**). The proportion of expanding cells in each tumor was poorly explained by individual technical covariates, including the age of the tumor ($R^2 = 0.25 \pm 0.14$), the depth of the tumor phylogeny ($R^2 = 0.23 \pm 0.15$), the number of cells in the tumor ($R^2 = 0.09 \pm 0.07$), and the proportion of unique cell lineage states ($R^2 = 0.28 \pm 0.15$, **Figure 5.9A-D**); though an additive linear model with all of these covariates was a stronger predictor ($R^2 = 0.52$).

Several lines of evidence support the accuracy of the inferred phylogenies and subclonal dynamics. First, lineage trees inferred by an alternative phylogenetic reconstruction algorithm, Neighbor Joining, revealed consistent subclonal expansion proportions [216] (Pearson's $\rho = 0.87$, **Figure 5.9E**). Second, copy number variation (CNV) - a common feature for inferring subclonal structure in tumors [256] - corroborated tumor subclonal structure. Specifically, despite the low-resolution lineages inferred from detected CNVs, in the majority of tumors (20/21) the relationships from subclonal CNVs were significantly similar to the relationships inferred from our Cas9 lineage-tracing trees (**Figure 5.9G-I**; Permutation Test; see see Methods). Furthermore, expanding subclones were significantly enriched for CNVs (Mann-Whitney U Test $p < 0.0001$, **Figure 5.2C-D** and **Figure 5.9J**) and independent subclonal expansions from the same tumor could harbor distinct CNV patterns (**Figure 5.9K**). Third, cancer cells in expansions had significantly higher expression of cell-cycle genes (Mann-Whitney U test; **Figure 5.2E**, **5.9F**; see Methods). Together with our tumor spatial-lineage analysis (see below), these orthogonal data strongly support the fidelity of our tumor phylogeny and expansion calling and indicate the aggressive nature of subclonal expansions.

In population genetics, the relative "fitness" of a sample can be defined as the growth advantage of an individual compared to the rest of the population [279]. The fine-scale structure of our lineages

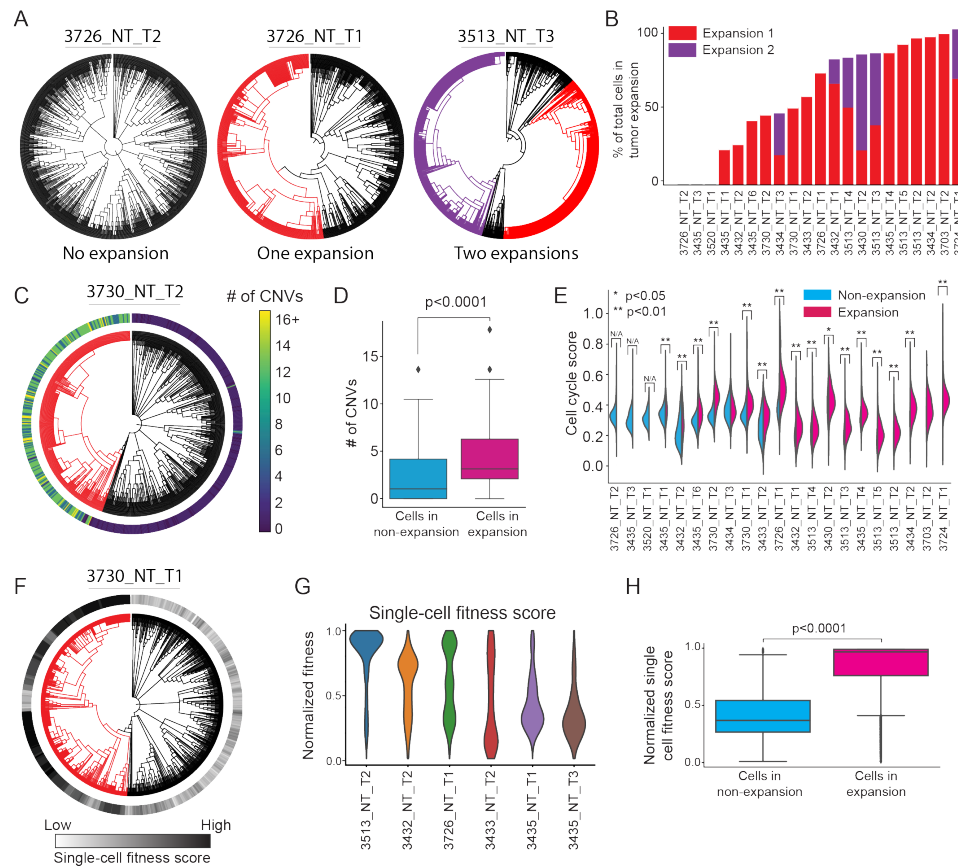


Figure 5.2: are subclones expand during tumor progression, marked by increased DNA copy number variation, cell cycle score, and fitness score. (A) Example tumor phylogenies with expansions highlighted with red or purple branches. (B) The number of expansions and percentage of expanding cells across tumors. Tumors are ranked by the total percentage of cells in expanding subclones. (C) CNV numbers per cell (outer bar) in expanding (red) versus non-expanding (black) cells of an example tumor. (D) Comparison of CNV number per cell in expansions versus non-expansions (Permutation test, $p < 0.0001$). (E) Comparison of cell cycle transcriptional scores of cells from the expanding and non-expanding subclones (two-sided Mann-Whitney U test, $*p < 0.05$, $**p < 0.01$). Tumors without expansions are labeled as N/A. (F-H) Phylogenetic single-cell fitness scores in expansions. (F) A representative tumor phylogeny with single-cell fitness scores overlaid. (G) Single cell fitness scores in representative tumors. (H) Cancer cells from expansions have significantly higher single-cell fitness scores (two-sided Mann-Whitney U test, $p < 0.0001$).

offers us the opportunity to predict fitness at single-cell resolution [186] (Figure 5.9F; see Methods).

This analysis revealed a spectrum of intratumoral fitness distributions across tumors (Figure 5.2G)

with expanding cells consistently having higher single-cell fitness scores (Mann-Whitney U Test $p <$

0.0001, Figure 5.9F-H). Overall, these results argue that we can quantitatively infer the relative

fitness of individual cells within a tumor and that cell fitness is consistent with the subclonal dynamics revealed by the tumor phylogeny.

5.3.3 Integration of phylodynamics and transcriptome uncovers

fitness-associated gene programs for KP tumors

With quantitative measurements of single-cell fitness in each tumor, we next sought to identify the molecular features consistently associated with subclonal expansions. Consistent with KP tumor progression being driven largely by epigenetic rather than genetic changes [151, 11, 170], we observed that CNV profiles within expansions were largely inconsistent across tumors (**Figure 5.9L**). We therefore examined the transcriptomic differences underpinning expansion. By integrating the scRNA-seq data across tumors, we detected 15 distinct subpopulations characterized by marker genes consistent with previous work in the KP model: spanning from an early-stage Alveolar type 2 (AT2)-like population, characterized by expression of *Lyz2* and *Sftpc*, to late-stage Epithelial-Mesenchymal transition (EMT)-related clusters characterized by expression of *Vim*, *Twist1*, and *Zeb2* [170, 151] (**Figure 5.3A**, **5.10A**). Notably, while normal AT2 cells appeared similar to the tumor AT2-like state, the transcriptome of cancer cells could be clearly distinguished from normal AT2 cells (**Figure 5.10B**; see Methods). Together, the agreement of transcriptomic states observed here and in previous studies implies that the continuous lineage tracing system did not strongly perturb tumor progression.

Combining the aforementioned single-cell fitness scores with single-cell transcriptomes for each

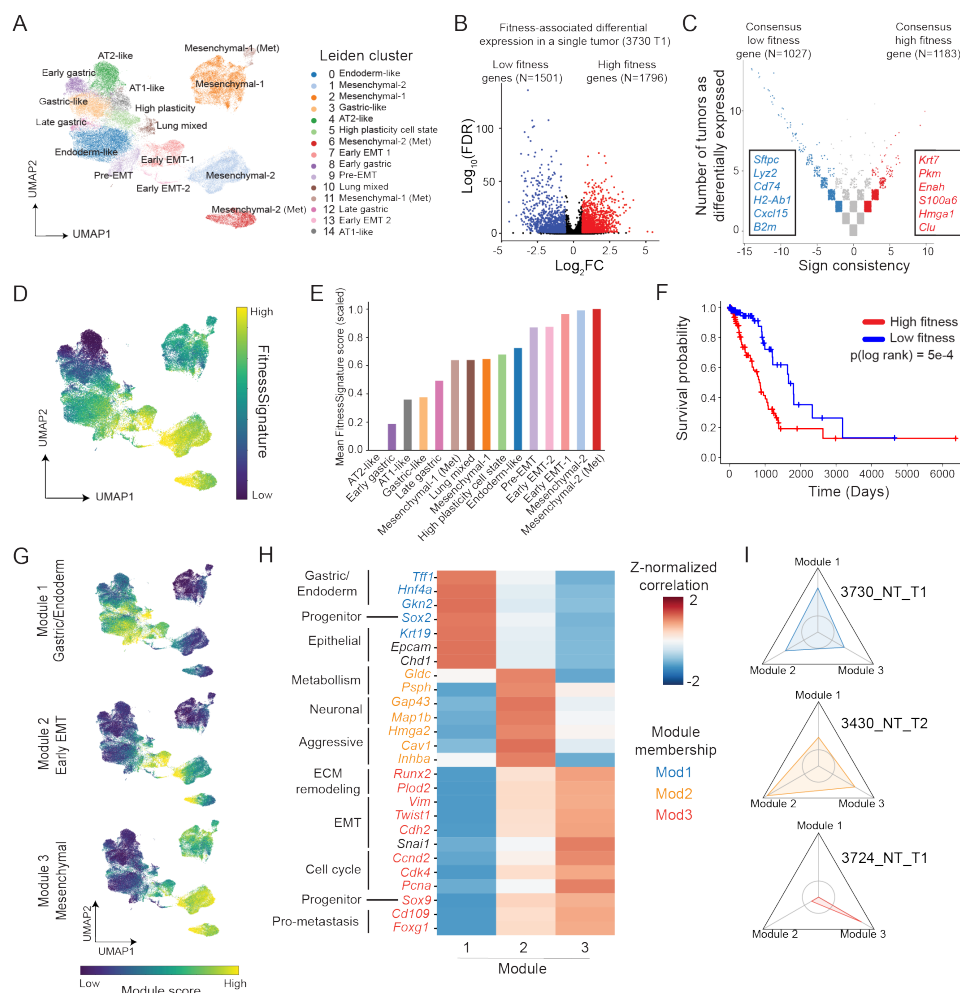


Figure 5.3: Integration of phylogenetics and transcriptome uncovers fitness-associated gene programs for KP tumors. (A) Gene expression UMAP and clustering of cancer cells from KP-Tracer tumors. (B-C) Identification of a transcriptional FitnessSignature. (B) Differential expression analysis identifies genes positively (red) and negatively (blue) associated with single-cell fitness (C) Meta-analysis of fitness-associated genes across all KP tumors. (D) Gene expression UMAP annotated by individual cells' single cell FitnessSignature scores (normalized to a 0-1 scale). (E) Average FitnessSignature scores of each Leiden cluster (normalized to 0-1). Colors reflect the Leiden clusters in (A). (F) Kaplan-Meier survival analysis of TCGA lung adenocarcinoma patients (n=495) stratified into high (red) and low (blue) groups based on gene expression of the derived transcriptional FitnessSignature (Log-rank test, $p = 5e - 4$). (G) Gene expression UMAP annotated with transcriptional scores of the three fitness gene modules. (H) Heatmap of Z-normalized Pearson's correlations between marker gene expression and fitness module scores for selected differentially expressed genes with manual annotations. Genes are colored by assigned fitness gene module; genes in black indicate helpful markers that did not appear in a fitness module. (I) Personality plots of three representative tumors displaying the fold change in fitness module scores of individual expansions compared to the non-expanding regions. Vertices indicate individual fitness modules. Axes are normalized to 0.4 - 2.2-fold change observed across tumors. Inner circle represents a fold change of 1 (no change) and values greater than 1 indicate the cells in expansions exhibiting enriched usage of the particular fitness gene module. Colors (see (H)) reflect the module a tumor expansion is characterized by.

tumor, we next identified genes associated with changes in fitness for each tumor (**Figure 5.3B**; see Methods). We then utilized a majority-vote meta-analysis of differentially expressed genes across tumors to find genes consistently associated with fitness differences (**Figure 5.3C**; see Methods). The resulting consensus genes associated with elevated fitness revealed broad transcriptomic changes and were enriched for gene sets associated with ribosome biogenesis, stem cell differentiation, and wound healing. The genes detected in our majority-vote meta-analysis represented a transcriptional program (hereafter referred to as the "FitnessSignature") consistently associated with tumor expansions that could be used to describe state trajectories underlying tumor evolution. Indeed, the AT2-like cluster had the lowest FitnessSignature score while the Mesenchymal clusters scored highest (**Figure 5.3D-E**; see Methods). Interestingly, the ranking of Leiden clusters in between these extremes suggested that an increase in FitnessSignature was concomitant with dedifferentiation from the AT2-like state through various Gastric, Endoderm-like, or Lung Mixed states to an eventual Mesenchymal state (**Figure 5.3D-E**). Importantly, the FitnessSignature scores were significantly associated with poor prognosis in lung adenocarcinoma patients from The Cancer Genome Atlas [47](TCGA; **Figure 5.3F**; see Methods).

Consistent with previous studies showing increased transcriptional heterogeneity during KP tumor evolution [170], we observed that tumors occupied qualitatively different transcriptional states (**Figure 5.10E**). This progression could be categorized into three non-overlapping gene modules decomposed from the FitnessSignature (**Figure 5.10F-G**; see Methods): Module 1 contained genes enriched for gastric and endoderm signatures (*Tff1*, *Hnf4a*, *Gkn2*), Module 2 contained a subset of EMT marker genes and some neuronal genes (*Hmga2*, *Inhba*, *Gap43*) and Module 3 contained clas-

sical mesenchymal and pro-metastasis genes (*Vim*, *Twist1*, *Cdh2*, *Cd109*, *Runx2*) (**Figure 5.3G-H**). Additionally, tumor subclonal expansions could preferentially employ a particular module, though some expansions exhibited co-expression of multiple modules (**Figure 5.3I**, **5.10I-J**; see Methods). Importantly, the expression of each of these modules was predictive of worse patient survival in the TCGA lung adenocarcinoma cohort (**Figure 5.10H**; see Methods). Collectively, these results argue that increased cell fitness in lung adenocarcinoma can be achieved via at least three distinct transcriptional modules.

5.3.4 Intratumoral transcriptional heterogeneity is driven by transient increases in plasticity of cell states

We next investigated the dynamics of intratumoral transcriptional diversity, as such behavior is can be a driver of tumor aggressiveness and therapeutic resistance [[194](#), [207](#), [225](#), [141](#), [170](#), [172](#)]). In our model, tumors varied widely in the transcriptional states they occupied, rarely being dominated by a single state. While tumors with low FitnessSignature scores were enriched for the AT2-like state, increases in the Fitness score were associated with Gastric-like, Lung Mixed, and Mesenchymal states (**Figure 5.11A**). Moreover, tumors had generally similar levels of transcriptional state heterogeneity, as measured by Shannon's Entropy Index [[170](#), [151](#)](**Figure 5.11B**).

How is this intratumoral diversity established and maintained? In principle, this diversity reflected by the entropy index can be achieved either by rare transitions and stable commitment to distinct states or by frequent transitions between these states. Lineage tracing is uniquely positioned to dis-

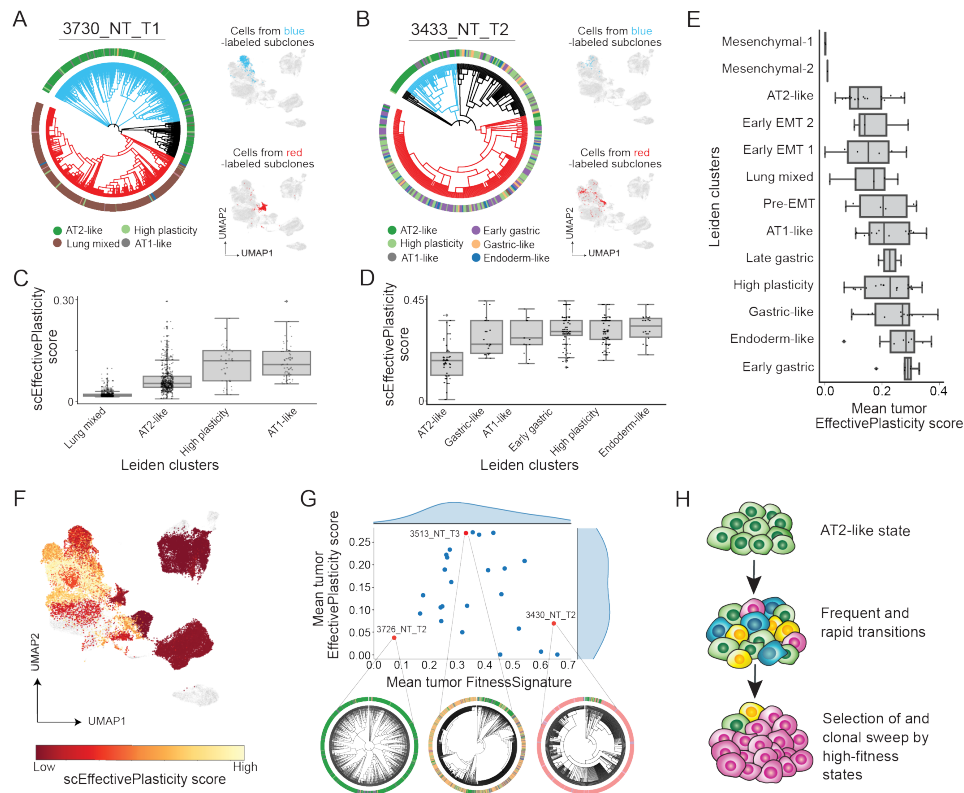


Figure 5.4: Intratumoral transcriptional heterogeneity is driven by transient increases in plasticity of cell states. (A-B) Representative tumors with (A) low EffectivePlasticity and (B) high EffectivePlasticity. Outer bar indicates the Leiden cluster of single cells (as in **Figure 5.3A**). Selected clades are highlighted on the gene expression UMAP to the right of phylogenies. (C-D) Quantification of scEffectivePlasticity for each transcriptional state (Leiden cluster) for tumors in (A) and (B). Each dot represents a single cell's EffectivePlasticity. (E) Distribution of mean EffectivePlasticity scores for each Leiden cluster across KP tumors. Each dot represents a Leiden cluster's mean EffectivePlasticity within a tumor. Leiden clusters are ranked by the mean of the distribution across tumors. (F) scEffectivePlasticity score overlaid onto the gene expression UMAP. Cells marked in grey are from metastases and not included. (G) Relationship between tumor average FitnessSignature and EffectivePlasticity. Three representative phylogenies are displayed with Leiden cluster annotations (outer circle). (H) A model describing changes of transcriptome heterogeneity and EffectivePlasticity following tumor progression.

tinguish these two models as it directly reports how intermixed transcriptomic states are in subclonal lineages, thus providing a measure of effective plasticity. Interestingly, tumor subclones exhibited varying amounts of plasticity: some tumor subclones were dominated by a single transcriptomic state, suggesting strong stability (**Figure 5.4A**), while others were characterized by strong mixing between transcriptomic states (**Figure 5.4B**). Using tumor phylogenies, we estimated the frequency

of cellular state changes for each tumor to create an empirical measurement of the tree plasticity (hereafter referred to as the "EffectivePlasticity" score) and extended this measure to a single-cell statistic ("scEffectivePlasticity") by averaging together the EffectivePlasticity scores for all the subclades that contained a particular cell [203] (see Methods). Importantly, this scEffectivePlasticity statistic was consistent with alternative approaches that quantified the effective plasticity by comparing transcriptional states between cells with similar indel states (without relying on trees; **Figure 5.11C-E**) or by computing dissimilarity in gene expression profiles between nearest neighbors on the phylogeny (**Figure 5.11F-H**; see Methods).

In two representative tumors, we observed that cells from the AT2-like state exhibited consistently low scEffectivePlasticity, whereas other states like the Gastric- and AT1-like state had elevated scEffectivePlasticity scores (**Figure 5.4C-D**). To systematically quantify the relative effective plasticity of different cell states, we averaged scEffectivePlasticity scores for each Leiden cluster on a tumor-by-tumor basis (**Figure 5.4E**). Mesenchymal (Leiden clusters 1 & 2) and AT2-like clusters (Leiden cluster 4) represented the most stable states, while the previously reported "High Plasticity Cell State" [170] (Leiden cluster 5), Gastric-Like (Leiden clusters 3, 8, 12) and Endoderm-like states (Leiden cluster 0) exhibited high EffectivePlasticity (**Figure 5.4F**).

We next investigated the relationship of tumor plasticity, as measured by EffectivePlasticity, and aggressiveness, as measured by the FitnessSignature. While previous studies have indicated that transcriptional heterogeneity is a hallmark of tumor progression [170], we found that the average EffectivePlasticity score was maximized when the FitnessSignature score was in the intermediate regime and minimized when the FitnessSignature was on the low or high extremes (**Figure 5.4G**

and **5.11I-J**). Taken together, these findings support a model of tumor progression whereby loss of AT2-like state unlocked high plasticity enabling rapid, parallel transitions to generate high transcriptomic heterogeneity, which permitted selection of increasingly stable states with higher-fitness and ultimately resulted in subclonal expansion and tumor progression (**Figure 5.4H**).

5.3.5 Mapping the phylogenetic relationships between cell states reveals common paths of tumor evolution

In principle, the observed cellular plasticity and subsequent transcriptional heterogeneity in the KP model could arise from either random or structured evolutionary paths through transcriptional states. To investigate the consistency of evolutionary paths across tumors, we developed a statistic termed "Evolutionary Coupling", which extends a clonal coupling statistic [275, 270] to quantify the phylogenetic distance between pairs of cell states (see Methods).

Applying this approach to individual tumors uncovered distinct coupling patterns between transcriptomic states. In one example tumor, the Lung Mixed state was more closely related to the High Plasticity state than to the AT2-like state (**Figure 5.5A-B**). In another tumor, the Gastric-like and High Plasticity states clustered together, while the AT1-like and Early Gastric states clustered together (**Figure 5.5C-D**). Relationships for these two tumors were consistent with alternative definitions for inter-state coupling, inferred directly from the indel information (without relying on trees; **Figure 5.12A-B**; see Methods) or based on local neighborhoods on the tree (**Figure 5.12C-D**; see Methods); these statistics were generally consistent across trees (**Figure 5.12E**).

A data-driven hierarchical clustering of the full set of tumors based on their transcriptional state occupancy and Evolutionary Couplings revealed that tumors could be classified into three distinct groups ("Fate Clusters"; **Figure 5.5E** and **5.12F**; see Methods). While some transcriptional states were shared between Fate Cluster 1 and 2 (including the AT2-like, AT1-like, and High-Plasticity states), Fate Cluster 1 was predominantly distinguished by couplings that include the Gastric-like (Leiden clusters 3, 8, and 12) and Endoderm-like states (Leiden cluster 0; **Figure 5.5F**, left, **Figure 5.12G**) and Fate Cluster 2 by evolution towards the Lung Mixed state (Leiden cluster 10; **Figure 5.5F**, middle, **Figure 5.12G**). Fate Cluster 3 was more difficult to interpret as it lacked couplings with the AT2-like state and instead was dominated by high-fitness states, such as early EMT (Leiden clusters 7 and 13) and Mesenchymal states (Leiden cluster 1 and 2; **Figure 5.5F**, right, **Figure 5.12G**).

We thus hypothesized the majority of differences between tumors was driven by tendencies towards Fate Cluster 1 or 2. Indeed, Principal Component Analysis (PCA) on Evolutionary Couplings and state composition revealed that the first two principal components explained a substantial amount of the observed variance (32%; **Figure 5.12H**) and couplings involving the Gastric & Endoderm states (Fate Cluster 1; Leiden clusters 3, 8, 0) or the Mixed Lung state (Fate Cluster 2; Leiden cluster 10) were among the strongest features distinguishing tumors (**Figure 5.12I**). Taken together, these distinct coupling patterns argue that tumor progression from the initial AT2 state preferentially follows one of two non-overlapping evolutionary paths, characterized by Fate Clusters 1 and 2, to aggressive states like those found in Fate Cluster 3.

To characterize the transcriptional changes that underlie these two alternative fates (Fate Clus-

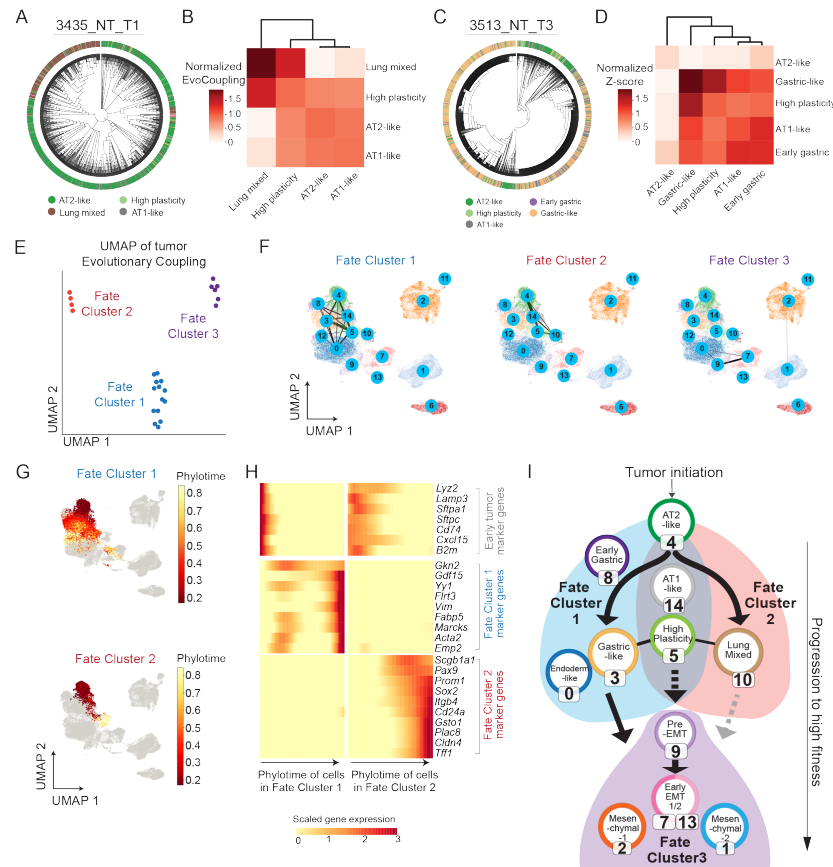


Figure 5.5: Mapping the phylogenetic relationships between cell states reveals common paths of tumor evolution. (A-D) Transcriptional state relationships of representative tumors are quantified with Evolutionary Couplings. (A, C) Phylogenies of tumors 3435_NT_T1 and 3513_NT_T3 with overlaid Leiden cluster annotations (colors from **Figure 5.3A**). (B, D) Corresponding normalized Evolutionary Couplings between Leiden clusters in each tumor. (E) UMAP projection of KP tumor Evolutionary Couplings annotated by identified "Fate Clusters" (see **Figure 5.12F**). Dots correspond to tumors. (F) Aggregated Evolutionary Couplings between transcriptional states of tumors from each Fate Cluster visualized on the gene expression UMAP. Thickness of bars reflect the average magnitude of couplings across tumors in a Fate Cluster. (G) Gene expression UMAP annotated by Phylotime of single cells from tumors in Fate Cluster 1 (top) and 2 (bottom) (normalized to 0-1). Cells from tumors that do not appear in the Fate Cluster of interest are shown in gray. (H) Significant gene expression changes along Phylotime for Fate Cluster 1 and 2 across Phylotime quantiles. Genes are annotated by their assigned Fate Cluster. Colors in heatmap are library-normalized gene expression, Z-normalized across quantiles of both Fate Clusters. (I) Summary of major paths of KP tumor progression. Solid lines indicate direct evidence of Evolutionary Couplings; dotted lines indicate couplings likely involving unobserved intermediate states; gray lines indicate couplings that are supported by rare examples.

ter 1 & 2), we developed "Phylotime": a single-cell statistic that quantifies the evolutionary distance between an individual cell and cells in the progenitor, AT2-like state (see Methods). Importantly, estimates of Phylotime were consistent with different metrics for approximating distances on the tree: either by the absolute number of mutations or the number of mutation-bearing edges (**Figure 5.12J-K**). Integrating Phylotimes from tumors within Fate Clusters 1 and 2 confirmed two separate evolutionary routes (**Figure 5.5G**) and highlighted distinct transcriptional changes associated with Phylotime along each route (**Figure 5.5H**; see Methods). Specifically, while expression of early markers like *Lyz2* and *Sftpc* were shared in early Phylotime of both Fate Clusters, late Phylotime in Fate Cluster 1 was enriched for gastric and endoderm markers like *Gkn2*, whereas late Phylotime in Fate Cluster 2 was characterized by markers of airway progenitors, such as *Sox2* and *Scgb1a1*, and markers of tumor propagating cells, like *Cd24a* and *Itgb4*. Although Fate Cluster 3 tumors generally had poor couplings with earlier states, our data suggest that tumors can evolve from either the Fate Cluster 1 or Fate Cluster 2 into an EMT state and progress to late-stage Mesenchymal states (**Figure 5.12L**). Overall, our analysis provides evidence that KP tumors could evolve predominantly through one of two major paths with one towards Gastric-like and Endoderm-like state, and the other through the Mixed-Lung state, with distinct transcriptional changes associated with each evolutionary trajectory (summarized in **Figure 5.5I**).

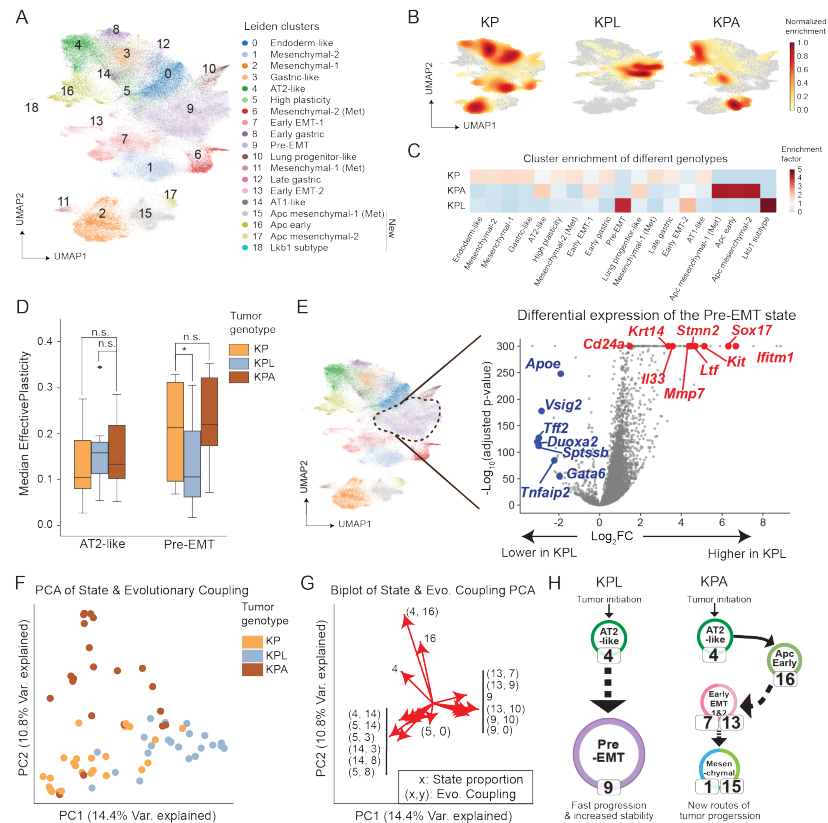


Figure 5.6: Loss of tumor suppressors alters tumor transcriptome, plasticity and evolutionary trajectory. (A) Batch corrected and integrated gene expression UMAP of all cancer cells from KP, KPL and KPA tumors annotated by 19 Leiden clusters (see Methods). (B) Density plots of cancer cells from KP, KPL and KPA tumors on the UMAP. (C) Enrichment of genotypes in each Leiden cluster. Enrichments below 1 are colored blue; enrichments above 1 are colored red. (D) Median EffectivePlasticity scores in selected Leiden clusters across genotypes (one-sided Mann-Whitney U Test, $*p \leq 0.05$, n.s. = not significant). (E) Genes up-regulated (red) and down-regulated (blue) in the Pre-EMT state of KPL tumors compared to KP and KPA tumors combined. (F) PCA of Evolutionary Coupling and transcriptional state proportion vectors for all tumors analyzed across genotypes. Each dot represents a tumor. (G) Biplot of top 10 features per principal component from PCA analysis shown in (F). Evolutionary Couplings are shown as tuples (x, y) ; transcriptional state proportions are shown as a single number x indicating Leiden cluster ID. (H) Summary of major evolutionary paths in KPL and KPA tumors. Solid lines indicate direct evidence of Evolution Couplings between transcriptome states, dotted lines indicate couplings that likely involve unobserved intermediate cell states.

5.3.6 Loss of tumor suppressors alters tumor transcriptome, plasticity and evolutionary trajectory

Tumor suppressor genes regulate diverse cellular activities and their loss is associated with increased tumor aggressiveness [274, 226]; however, it remains unclear how these genes affect tumor evolutionary dynamics in vivo. Here, we combined genetic perturbations with our quantitative phylodynamic approaches to interrogate how additional oncogenic mutations altered KP tumor evolutionary trajectories.

We focused on two frequently mutated tumor suppressors in human lung adenocarcinoma, *LKB1* and *APC* [59, 47, 231]. Both genes have been studied extensively in both human and mouse models and appear to regulate progression through distinct mechanisms [126, 31, 187, 112, 253, 183, 139, 193]. We engineered our lenti-Cre-BC vector to carry an additional sgRNA targeting *Lkb1* or *Apc*, such that delivery of this vector simultaneously initiated tumor induction, lineage tracing, and disruption of the targeted tumor suppressor gene. With this system, we collected data from 18,321 cells across 57 KP tumors with *Lkb1* knockout (24 primary and 33 metastatic tumors; referred to as KPL tumors), and 13,825 cells across 35 KP tumors with *Apc* knockout (23 primary and 12 metastatic tumors; referred to as KPA tumors). Targeting of either *Lkb1* or *Apc* increased tumor burden [214], but did not appear to alter the number and relative size of subclonal expansions (**Figure 5.13A-B**). Yet, genes associated with tumor fitness were largely distinct across genetic backgrounds (**Figure 5.13C**).

To examine whether perturbations alter the transcriptional landscape of KP tumors, we integrated

transcriptional states of KPL and KPA tumors with the prior KP dataset. While many cells could be classified into existing Leiden clusters identified in the KP analysis, the additional perturbations also created four new transcriptional states (**Figure 5.6A**; see Methods). As expected from *Apc*'s role as a negative regulator of *Wnt* signaling [14], *Axin2* expression was high in the three KPA-specific clusters, indicative of elevated *Wnt* signaling (**Figure 5.13D**), as was the expression of *Wnt* antagonists such as *Notum* and *Nkd1* which were recently reported to increase the ability of cancer cells to compete with the neighboring niche in human *APC* mutant colon tumors [74, 184](**Figure 5.13D**). Moreover, targeting of *Lkb1* or *Apc* resulted in changes to the relative occupancies of transcriptomic states: KPL tumors were primarily enriched in the Pre-EMT state (Leiden cluster 9), while KPA tumors were enriched in *Apc*-specific early, mesenchymal, and metastatic states (Leiden clusters 15, 16, and 17; **Figure 5.6B-C** and **5.13E**).

Interestingly, although most cell states had comparable EffectivePlasticity across tumor genotypes (**Figure 5.13F**), the Pre-EMT state (Leiden cluster 9) in KPL tumors had significantly less EffectivePlasticity, indicating stabilization of this cell state ($p < 0.05$, Mann-Whitney U Test; **Figure 5.6D**). We next identified genes differentially expressed in cells from KPL tumors in the Pre-EMT cluster (**Figure 5.6E**; see Methods), which included gene programs that can promote pro-metastatic chromatin remodeling (*Sox17* [197]), tumor progression (*Ifitm1* and loss of *Gata6*; [288, 41]), metastatic ability (*Mmp7* [106]), and tumor fitness by modulating cancer-immune cell interaction (*Cd24a*, *I133*, and loss of *ApoE* [229, 160, 257]). These together potentially explain why the Pre-EMT state was uniquely stabilized in KPL tumors.

To examine how loss of tumor suppressors altered evolutionary trajectories, we performed PCA

on the transcriptional state occupancy and Evolutionary Couplings of individual tumors and found that tumors broadly segregated according to their genotypes (**Figure 5.6F**; see Methods). Specifically, KPA tumors created a unique trajectory including a coupling between the AT2-like and the Apc-early states (Leiden clusters 4 and 16), while KPL tumors were characterized by couplings between the Pre-EMT state and nearby states (**Figure 5.6G**).

In summary, although the targeting of the tumor suppressors *Lkb1* or *Apc* both increased tumor growth, their effects on cell states, plasticity and paths of evolution varied substantially. Specifically, KPL tumors quickly progressed to and became stabilized in the Pre-EMT state, while KPA tumors largely exploited a distinct path through new Apc-specific states (**Figure 5.13G** and summarized in **Figure 5.6H**). Together, our analyses highlight how lineage tracing offers rich information for dissecting the multifaceted role of tumor suppressors in tumor evolution.

5.3.7 Metastases originate from spatially localized, expanding subclones of primary tumors

Metastases account for 90% of cancer mortality yet remain difficult to study because of their spatially and temporally sporadic nature [81]. An outstanding question is how metastases originate from the primary tumor. Here we integrated lineage tracing with spatial and transcriptomic information to investigate the subclonal origins and evolution of metastases.

We first focused on a single primary tumor, which consisted of two independent subclonal expansions (3724_NT_T1; **Figure 5.2B**), and its four related metastases (three in liver and one in soft tis-

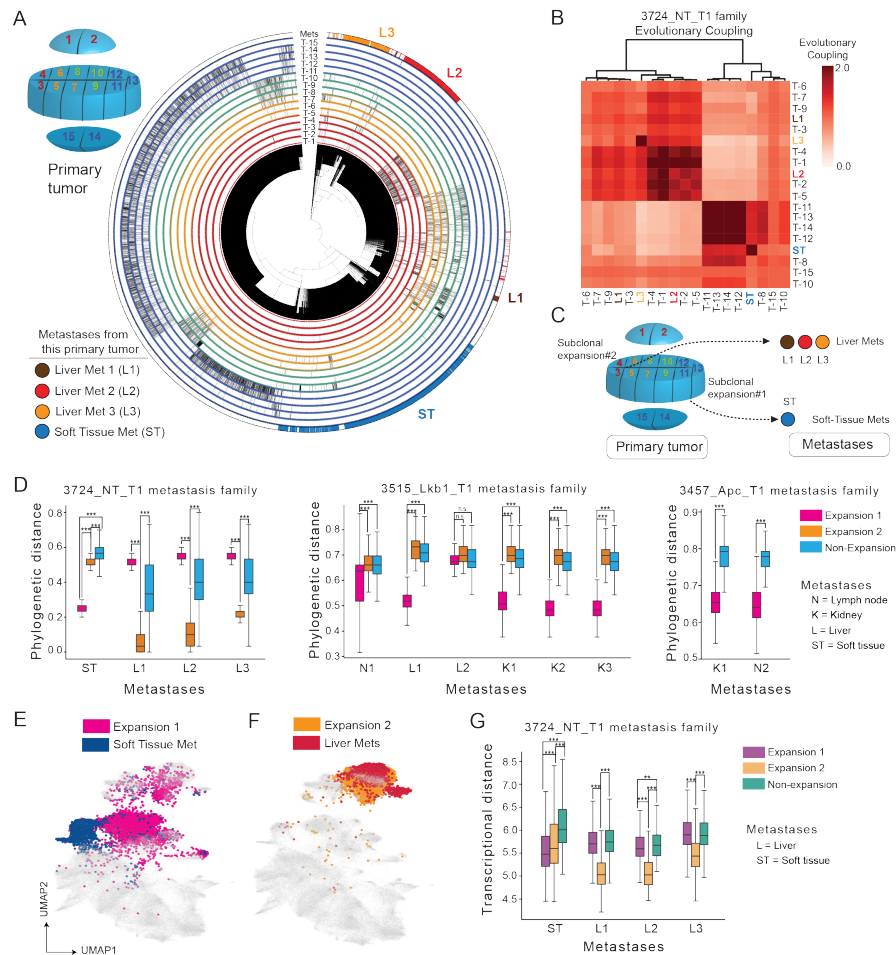


Figure 5.7: Metastases originate from spatially localized, expanding subclones of primary tumors. (A) Multi-region analysis of tumor-metastasis family 3724_NT_T1. Top left inset showed the relative spatial location of tumor pieces. The phylogeny of the primary tumor and metastases is annotated via peripheral radial tracks for each color-coded region of the tumor (matching the inset) and four metastases. (B) Heatmap of Evolutionary Couplings of primary tumor pieces (black) and 4 related metastases (matching colors in (A)) from the 3724_NT_T1 tumor-metastasis family. (C) Summary of the spatial-phylogenetic relationship of the tumor-metastasis family 3724_NT_T1. (D) Single-cell phylogenetic distance of each metastasis to the non-expanding and expanding subclones in its related primary tumor. Each box represents the distribution of phylogenetic distances from a metastasis to a defined region of its related primary tumor (one-sided Mann-Whitney U test are indicated: *** $p < 0.0001$, n.s. = not significant). (E-F) Gene expression UMAP annotated by metastases and their original subclones in 3724_NT_T1. Cells that are not relevant to the comparison in each panel are shown in gray. (G) Transcriptional distances between expanding regions of 3724_NT_T1 and its four metastases (one-sided Mann-Whitney U test are indicated: * $p < 0.001$, ** $p < 0.0001$).

sue; **Figure 5.7A, 5.14A**). We performed multi-regional analysis of the primary tumor (**Figure 5.7A**, inset) and inferred a combined phylogeny relating all cells in the primary tumor and metastases. Integrating lineage-spatial information revealed that individual metastases originated from distinct spatial locations (**Figure 5.7A-C**; see Methods), and phylogenetically originated from specific subclonal expansions in the primary tumor (**Figure 5.7C-D**).

To investigate the consistency of these results, we extended this phylogenetic analysis to five other tumor-metastasis families, across KP, KPL, and KPA backgrounds. Importantly, metastases were consistently more closely related phylogenetically to specific subclonal expansions regardless of tumor genotype (**Figure 5.7D** and **Figure 5.14D**). Collectively, our results argue that metastases generally originated from subclonal expansions within primary tumors. Independent metastases from the same primary tumor could arise from spatially and phylogenetically distinct subclones.

We next evaluated to what degree metastases preserved the transcriptional state of their origins in the primary tumor. Analysis of metastases arising from an example primary tumor (3724_NT_T1) revealed that liver metastases were more similar to the subclone from which they originated, whereas the soft tissue metastasis evolved to a new transcriptional state (**Figure 5.7E-F**). This was further quantified by measurements of total transcriptional distance between each metastasis and the subclonal expansions in the metastatic primary tumor (**Figure 5.7G**). Liver metastases were significantly more similar to its originating subclonal expansion ($p < 0.0001$, one-sided Mann-Whitney U Test), while the soft tissue metastasis did not clearly resemble its subclonal origin (**Figure 5.7G**; see Methods). Consistently, metastases from KP, KPL, and KPA mice were significantly more similar, as measured by transcriptional state, to their respective expanding subclades in the primary tumor

as compared to non-expanding regions, further suggesting that progression at the primary site is a prerequisite for metastasis [151] (**Figure 5.14E**).

In addition, our high-resolution lineage tracing offered evidence of complex metastatic behaviors, including multi-subclonal seeding from a primary tumor to the lymph node, and cross-seeding from one metastatic primary tumor to another primary tumor, or from one metastasis to another (**Figure 5.14A-C**). Collectively, these results highlight the ability of phylogenetic analysis to trace the origins and evolution of metastases.

5.4 Discussion

In this study, we have developed a genetically engineered mouse model of lung adenocarcinoma that allows Cre-inducible initiation of oncogenic mutations and simultaneous continuous in vivo lineage tracing of tumor development over many months, paired with a single-cell transcriptomic readout. This model system enabled us to track at an unprecedented resolution the recurring patterns of tumor evolution from activation of oncogenic mutations in single cells as they grow into large, aggressive, and ultimately metastatic tumors. Three principles emerged from our study, linking together tumor phylodynamics, fitness, plasticity, parallel evolutionary trajectories, origins of metastasis, and genetic determinants of tumor evolution.

First, tumors were driven by rare subclonal expansions that utilized distinct fitness-associated transcriptional programs and enabled both tumor progression at the primary site and metastasis to distant tissues. The expansions identified by tree topology argue for subclonal selection, distinct

from evolutionary models lacking selective sweeps observed in other cancer types [235]. The identification of gene expression states associated with tumor fitness revealed a set of transcriptional fitness modules underlying KP-Tracer tumor development. Importantly, these signatures of aggressive tumors found in our mouse model were predictive of the outcome of human disease. Despite the higher somatic mutation burden and longer developing timescales of human tumors [30, 124, 89, 110], our data uncovered critical fitness gene programs that are conserved in both mouse and human lung adenocarcinomas. Notably, we found that metastases consistently originated from expanding subclones, regardless of additional loss of *Lkb1* or *Apc*. They often retained the same transcriptional state as their original subclones but could further adopt distinct transcriptional states. This underscored the importance of tumor progression at the primary site in enabling metastasis [32, 263, 115, 151], and argues against alternative models in which metastases arise early during tumor evolution [119, 198, 147, 210, 235].

Second, our analysis revealed that tumor progression is accompanied by transient increases in lineage plasticity. This period of high plasticity is followed by clonal sweeps of subclones with aggressive cell states that can remain stable even following metastasis to new environments. Our ability to monitor how often cells are transitioning between transcriptomic states also allowed us to untangle the relationship between intratumoral heterogeneity and lineage plasticity, and shed light on the dynamics of the transcriptomic heterogeneity observed in the KP mouse model and human NSCLC [170, 157]. The finding that KP tumors progress via parallel, rapid transitions between cell states is consistent with previous work suggesting that epigenetic instability is a major driver of tumor progression in this model [151, 170]. Given the essential role of cellular plasticity in tumor

progression and therapeutic resistance [36, 63, 85, 75, 292, 204], the ability of our lineage tracing system to quantitatively explore plasticity provides a critical tool for understanding the role that cell state plasticity plays in various aspects of tumor evolution.

Third, tumors evolved through stereotypical trajectories and introduction of additional oncogenic mutations increased the speed of tumor evolution by creating new evolutionary trajectories. Traditionally, while cellular trajectories inferred by pseudotemporal approaches have proved to be a versatile tool for scRNA-seq datasets [261, 150], they make the inviolable assumption that transcriptional similarity indicates developmental relationship [262]. Overcoming this, our measurement of cell state coupling directly from phylogenies enabled the discovery of two distinct evolutionary paths that are substantiated by transcriptional differences. Moreover, CRISPR targeting of tumor suppressors *Lkb1* and *Apc* altered the cellular plasticity and observed evolutionary paths in a genotype-specific way, which can be explained by alterations in transcriptional landscape. Collectively, our approach offers an orthogonal and more quantitative evaluation of the multifaceted role genes play in tumor evolution as compared to traditional growth-based fitness analysis. Future studies combining the KP-Tracer model and high-throughput *in vivo* functional genomics will be foundational in assessing the evolutionary consequences of any genes of interest in lung adenocarcinoma progression [281].

In summary, our results represent the first report of tracing the evolutionary history of a tumor from a single transformed cell to an aggressive tumor using a CRISPR-based lineage tracer in an autochthonous mouse model. The continuous and high-resolution tumor lineage tracing in this setting offers a major advance in tumor evolution modeling by enabling quantitative inference of fitness landscapes, cellular plasticity, evolutionary paths, origins of metastases, and the role of tumor

suppressors in altering all these facets of tumor development. With the expanding lineage tracing toolkit and integration of other emerging data modalities, we expect that the experimental and computational framework presented here will greatly improve future efforts at building high-dimensional, quantitative, and predictive models of tumor evolution, thus shedding light on new therapeutic strategies.

Limitations of the study

Our findings highlight several opportunities for future efforts. First, we were limited in our ability to describe the directionality of transitions or to rule out the possibility of unobserved intermediates. This issue could be resolved experimentally by harvesting samples from multiple time points of tumor development, or expanding our lineage-tracing technology to develop multichannel molecular recorders for simultaneous recording of marker gene expression of intermediate states [79, 254]. Alternatively, enhancing the interpretability of branch lengths by engineering a “molecular clock” or probabilistic models of Cas9 editing [192] could aid in the reconstruction of unobserved intermediate states [191]. Second, our fitness-inference approach assumes that evolution occurs via small effect size mutations, which may overlook the impact of mutations with large impact such as CNVs in other tumor models [186]. Third, future integration of emerging data modalities with lineage tracing, such as combined genomic, multiomic and spatial analysis [182, 167, 158, 158, 242, 43], will illuminate how genetic and epigenetic changes and the tumor microenvironment influence tumor evolution.

5.5 Methods

Chimeric Lineage Tracing Mouse Model

All mouse experiments described in this study were approved by the Massachusetts Institute of Technology Institutional Animal Care and Use Committee (IACUC) (institutional animal welfare assurance no. A-3125-01). The engineered and selected mESC clones were injected into blastocysts from albino B6 or CD1 background for chimera making as previously described [299]. We chose to use the chimeric mice strategy because the multiple, random integration of lineage tracing target sites in the genome makes it challenging for breeding stable strains. Mice with more than 10% chimerism based on coat color were used in this study. Tumors were initiated by intratracheal infection of mice with lentiviral vectors expressing Cre recombinase [61]. Five total mESC clones were used in this study to avoid idiosyncrasy in clonal behavior and analyses were performed on all tumors combined. Lenti-Cre-BC vector was co-transfected with packaging vectors (delta8.2 and VSV-G) into HEK-293T cells using polyethylenimine (Polysciences). The supernatant was collected at 48h post-transfection, ultracentrifuged at 25,000 r.p.m. for 90 min at 4C, and resuspended in phosphate-buffered saline (PBS). 8-12-week-old chimeras were infected intratracheally with lentiviral vectors, including lenti-Cre-BC-sgNT (2x10⁷ PFU) or lenti-Cre-BC-sgLkb1 (4x10⁶ PFU) or lenti-Cre-BC-sgApc (1x10⁷ PFU) to achieve similar aging time after tumor initiation.

Lenti-sgRNA-Cre-Barcode vector

The lenti-sgRNA-Cre-barcode vector was derived from a previously described Perturb-seq lentiviral vector [3], pBA439, with the following changes: the two loxP sites were removed by site-directed mutagenesis (SDM) using oDYT001 and oDYT002 followed by oDYT009 and oDYT010; the Puro-BFP was removed using restriction sites NheI and PacI and was replaced by Cre that was PCR amplified using oDYT003 and oDYT004 via Gibson assembly; a ubiquitous chromatin opening element (UCOE) that was PCR amplified using oDYT005 and oDYT006 was introduced using restriction sites NsiI and NotI via Gibson assembly. oDYT007 and oDYT008 (containing EcoRI and SbfI sites for subsequent barcode cloning) were then annealed and ligated using restriction sites BclI and PacI. Three different sgRNAs of interest were then cloned into the resulting vector using pairs of top and bottom strand sgRNA oligos: sgNT (non-targeting) (oDYT011 and oDYT012), sgLkb1 (oDYT013 and oDYT014), and sgApc (oDYT015 and oDYT016) were each annealed and ligated using restriction sites BlnI and BstXI to form pDYT003, pDYT004, and pDYT005 respectively. These sgRNAs have been used and validated previously [213, 214]. Finally, a whitelist barcode oligo pool consisting of 249,959 unique 16-nucleotide barcodes where every barcode has a Levenshtein distance of >3 from every other barcode was designed. The whitelist barcode library was PCR amplified then introduced at the 3'UTR region of Cre in each of the three constructs using restriction sites EcoRI and SbfI.

Lineage tracer vector (Target site & triple sgRNAs)

The lineage tracer vectors pDYT001 and pDYT002 were derived from previously described target site plasmids, PCT 60-62 [37, 203, 127]. A loxP site was first removed from both PCT61 and PCT62 using oDYT017 and oDYT018 via site-directed mutagenesis. The triple sgRNA cassettes driven by distinct U6 promoters in PCT61 and PCT62 were then PCR amplified using oDYT019 and oDYT020 and introduced into the PCT60 backbone using restriction sites XbaI and NotI via Gibson assembly. Finally, the target site barcode library was PCR amplified from a previously described gene fragment from PCT48 [127], using oDYT021 and oDYT022 and introduced into the two resulting vectors above using restriction sites PacI and HpaI to form pDYT001 and pDYT002, which contain the triple guide cassette from PCT61 and PCT62 respectively. The target site library consists of a 14-bp random integration barcode and three target sites (*ade2*, *bri1*, *whiB*), which are complementary to the three sgRNAs.

Lineage tracing embryonic stem cell engineering

KP*17 is an embryonic stem (ES) cell line derived from C57BL/6-129/Sv F1 background engineered with conditional alleles *Kras*^{LSL-G12D/+}; *p53*^{fl/fl}. ES cells were maintained with JM8 media (500mL: 82.9% Knockout DMEM (Gibco Cat#10829-018), 15% FBS (Hyclone Cat#SV30014), 1% GlutaMax (Gibco Cat#35050-061), 1% Non-essential amino acids (Thermo Fisher Scientific Cat#11140050), 0.1% 2-mercaptoethanol (Sigma Cat#M-7522), 500,000U Recombinant Mouse LIF Protein (Millipore Cat#ESG1107)) with feeders. KP*17 was first targeted using CRISPR-assisted

HDR to generate *Rosa26^{LSL-Cas9-P2A-mNeonGreen}* which was validated for correct targeting by PCR and southern blot and validated for Cas9 activity. The lineage tracing transposon vectors were then introduced together with transposase vector (SBI) by transfection. Three passages after transfection, mESCs were purified by FACS based on mCherry expression and expanded as individual clones.

Target site integration number was quantified as the following: We first used fluorescence-based readout to examine mCherry expression of each ES cell clone in 96 well format, which allowed us to narrow down the ES clone candidates with relatively high expression of mCherry (the reporter of lineage tracer library). Then we used quantitative genomic PCR to count the number of lineage tracer genome integration in each ES cell clone by amplifying the target site regions (oDYT062 and oDYT063) and normalized to a 2N locus, β -actin, in the genome (oDYT060 and oDYT061). Samples were run in triplicates and the reactions were performed on a QuantStudio 6 Flex Real-Time PCR System. In this study, we used the following ES clones in the tumor analysis due to a combination of high chimeric rate and good target site capture: 1D5, 2E1, 1C4, 2F4 and 2H9. Clones 1D5, 1C4 were engineered with pDYT001 and clones 2E1, 2F4 and 2H9 were engineered with pDYT002. All five clones were used independently for generating chimeric mice in this study and no major difference in their lineage tracing performance was observed.

Sample preparation and purification of cancer cells

Tumors were harvested and single cell suspension was prepared as described in [44, 54]. Primary tumors and metastases were dissociated using a digestion buffer (DMEM/F12, 5mM HEPES, DNase, Collagenase IV, Dispase, Trypsin-EDTA) and incubated at 37°C for 30 min. After dissociation, the samples were quenched with twice the volume of cold quench solution (L-15 medium, FBS, DNase). The cells were then filtered through a 40um cell strainer, spun down at 1000rpm for 5 min, resuspended in 2mL ACK Lysing Buffer, and incubated at room temperature for 1-2 min. Lysis was then stopped with the addition of 10mL DMEM/F12 followed by the spinning down and resuspending of the samples in 1mL FACS buffer. Cells within the pleural fluid were collected immediately after euthanasia by making a small incision in the ventral aspect of the diaphragm followed by introduction of 1 ml of PBS. Cells were stained with antibodies to *CD45* (30-F11, Biolegend Cat#103112), *CD31* (390, Biolegend Cat#102410), *F4/80* (BM8, Biolegend Cat#123116), *CD11b* (Biolegend Cat#101212) and *Ter119* (Biolegend Cat#116212) to exclude cells from the hematopoietic and endothelial lineages. DAPI was used to stain dead cells.

Cells were then labeled by MULTI-seq [175] in 10 μ l PBS buffer containing 5 μ l lipid anchor (50 μ M) and 2.5 μ l of barcode oligos (100 μ M) for 10 min on ice and then 6 μ l co-anchor (50 μ M) 10 min on ice. Cells were washed and resuspended with ice-cold FACS buffer to prevent aggregation. DAPI was used to exclude dead cells. FACS Aria sorters (BD Biosciences) were used for cell sorting. Live cancer cells were sorted based on positive expression of mCherry and mNeonGreen as well as negative expression of hematopoietic and endothelial lineage markers (mCherry+,

mNeonGreen+, CD45-, CD31-, Ter119-, F4/80-, DAPI-). High purity of the resulting cancer cells has been confirmed in previous studies using similar fluorescent reporter systems [32, 44, 151]. Live normal lung cells were sorted based on negative expression of mNeonGreen, and hematopoietic and endothelial lineage markers. Datasets were further filtered for normal cells analytically via gene expression analyses (see section below "Single-cell transcriptome processing for KP-Tracer NT data") and by removing cells with low editing efficiencies (see section below "Single-cell lineage tracing preprocessing pipeline and quality control filtering").

Single-cell RNAseq library preparation

Single-cell RNA-seq libraries were prepared using 10X_3'_V2 kit according to the 10X user guide, except for the following modification. After cDNA amplification, the cDNA pool is split into two fractions. Half of the cDNA pool are used for scRNA-seq library construction and proceed as directed in the 10X user guide.

Target site library preparation

To prepare the Target Site libraries, the amplified cDNA libraries were further amplified with Target Site-specific primers containing Illumina-compatible adapters and sample indices (oDYT023-oDYT038,

forward:5'-CAAGCAGAAGACGGCATACGAGATNNNNNNNGTCTCGTGGGCTCGGAGATGTGTA

TAAGAGACAGAATCCAGCTAGCTGTGCAGC;

reverse:5'-AATGATACGGCGACCACCGAGATCTACACNNNNNNNTCTTTCCCTACACGACGC
TCTTCCGATCT;

"N" denotes sample indices) using Kapa HiFi ReadyMix (Roche), as described in [127]. Approximately 30 fmol of template cDNA was used per sample, divided between four identical reactions to avoid possible PCR induced library biases. PCR products were purified and size-selected using SPRI magnetic beads (Beckman) and quantified by BioAnalyzer (Agilent).

MULTI-seq library preparation

The MULTI-seq libraries were prepared as described in [175], using a custom protocol based on the 10x Genomics Single Cell V2 and CITE-seq workflows. Briefly, the 10x workflow was followed up until complementary DNA amplification, where $1\mu\text{l}$ of $2.5\mu\text{M}$ MULTI-seq additive primer (oDYT039) was added to the cDNA amplification master mix. After amplification, MULTIseq barcode and endogenous cDNA fractions were separated using a 0.6X solid phase reversible immobilization (SPRI) size selection. To further purify the MULTI-seq barcode, we increased the final SPRI ratio in the barcode fraction to 3.2X reaction volumes and added 1.8X reaction volumes of 100% isopropanol (Sigma-Aldrich). Eluted barcode cDNA was then quantified using QuBit before library preparation PCR using primers oDYT040 and oDYT041-oDYT048 (95°C, 5'; 98°C, 15'; 60°C, 30'; 72°C, 30');

eight cycles; 72°C, 1'; 4°C hold).

TruSeq RPIX: 5'-CAAGCAGAAGACGGCATAACGAGATNNNNNNGTGACTGGAGTTCCTT
GGCACCCGAGAATTCCA-3'

TruSeq P5 adaptor: 5'-AATGATACGGCGACCACCGAGATCTACACTCTTTCCCTACACGAC
GCTCTTCCGATCT-3'

Following library preparation PCR, the library was size-selected by a 1.6X SPRI clean-up prior to sequencing.

Lenti-Cre-BC library preparation

The Lenti-Cre-BC library amplification protocol was adapted from the Perturb-seq protocol [3]. 4 parallel PCR reactions were constructed containing 30ng of final scRNA-seq library as template, oDYT049, and indexed oDYT050-oDYT059, and amplified using KapaHiFi ReadyMix according to the following PCR protocol: (1) 95C for 3 min, (2) 98C for 15 s, then 70C for 10 s (16-24 cycles, depending on final product amount). Reactions were re-pooled during 0.8X SPRI selection, and then fragments of length 390bp were quantified by bioanalyzer. Lenti-Cre-BC libraries were sequenced as spike-ins alongside the parent RNA-seq libraries.

Sequencing

Sequencing libraries from each sample were pooled to yield approximately equal coverage per cell per sample; scRNA gene expression libraries, Target Site amplicon libraries, MULTI-seq amplicon libraries and Lenti-Cre-BC amplicon libraries were pooled in an approximately 10:3:1:1 molar ratio for sequencing, aiming for at least 70,000 total reads per cell. The libraries were sequenced using a custom sequencing strategy on the NovaSeq platform (Illumina) in order to read the full-length Target Site amplicons. Sample identities were read as indices (I1: 8 cycles, R1: 26 cycles, R2: 290 cycles). Only the first 98 bases per read were used for analysis in the RNA expression libraries to mask the longer reads required to sequence the Target Sites.

Single-cell lineage tracing preprocessing pipeline and quality control

filtering

Each cell was sequenced in four sequencing libraries: a MULTI-seq library (for identifying sample identity), a target site library (for reconstructing phylogenies), an RNA-seq library (for measuring transcriptional states), and a Lenti-Cre-BC library (for verifying clonal identity). First, the scRNA-seq was processed using the 10X CellRanger pipeline (version 2.1.1) with the mm10 genome build. Then, each cell barcode identified from the 10X pipeline was assigned to a sample using the MULTI-seq library, which was processed with the deMULTIplex R package (version 1.0.2 [175]). Cells identified as doublets or without a discernible MULTI-seq label were filtered out from downstream analysis.

Next, we processed the Target Site library using the previously described Cassiopeia preprocessing pipeline [127, 203]. Briefly, reads with identical cellBC and UMI were collapsed into a single, error-corrected consensus sequence representing a single expressed transcript. Consensus sequences were identified within a cell based on a maximum of 10 high-quality mismatches (PHRED score greater than 30) and an edit distance less than 2 (default pipeline parameters). UMIs within a cell reporting more than one consensus sequence were resolved by selecting the consensus sequence with more reads. Each consensus sequence was aligned to the wild-type reference Target Site sequence using a local alignment strategy, and the intBC and indel alleles were called from the alignment. Cells with fewer than 2 reads per UMI on average or fewer than 10 UMIs overall were filtered out. These data are summarized in a molecule table which records the cellBC, UMI, intBC, indel allele, read depth, and other relevant information. Cells that were assigned to Normal lung tissue via a MULTI-seq barcode or had more than 80% of their TargetSites uncut were assigned as "Normal" and not used for downstream lineage reconstruction tasks.

Lenti-Cre-BC libraries were processed using a custom pipeline combining Cassiopeia transcript collapsing, filtering, and quantification and a probabilistic assignment strategy based on the Perturb-seq gRNA calling pipeline [3]. First, sequencing reads were collapsed based on a maximum sequence edit distance of 2 and 3 high-quality sequences mismatches and then cells with fewer than 2 average reads per UMI or 2 UMIs overall were filtered out. Then, Lenti-Cre-BC sequencing reads were compared to the reference sequence and barcode identities were extracted and error-corrected by comparing each extracted barcode to a whitelist of Lenti-Cre-BC sequences, allowing for an edit distance of 3. Then, the count distributions for each unique Lenti-Cre-BC were inspected to re-

move barcodes that represented background noise. Next, a Lenti-Cre-BC coverage matrix was formed, summarizing the ratio between reads and number of UMIs for each barcode in each cell. Cell coverages were normalized to sum to the median number of coverages across the matrix and log₂-normalized. Finally, with this matrix we adapted the Perturb-seq gRNA calling pipeline to assign barcode identity to cells [3]. To do so, we fit a Gaussian kernel density function to the coverage distribution for each barcode and then determined a threshold separating "foreground" from "background" based on the relative extrema of the distribution (after removing the 99th percentile of the coverage distribution). Cells whose coverage values fell above the threshold were assigned that particular Lenti-Cre-BC. Cells that received more than one assignment or no assignment at all were marked as ambiguous.

After pre-processing each of these libraries, we called clonal populations, created character matrices, and reconstructed phylogenies for each clonal population (see sections below "Tree Reconstruction with Cassiopeia" and "Calling clonal populations and creating character matrices"). In this, we removed cells that contained few edited sites as this could indicate normal cell contamination (i.e. inactivity of Cas9) and identified consensus sets of intBCs per mES Clone (see section below "Creating a consensus intBC set for mESC clones") that were used for tree reconstruction. After tree reconstruction, we used the Lenti-Cre-BC data to remove cells within each tumor that contained strong evidence of different clonal origin (see section below "Cell Filtering with Lenti-Cre-BC"). Finally, we computed important clone-level quality-control statistics used for identifying clones with sufficient information for phylodynamic analysis (see section below "Tree Quality Control for Fitness Inference").

Across all three datasets (KP, KPL and KPA), this pipeline left us with 72,328 cells with high-quality Target Site information.

Calling clonal populations and creating character matrices

In this study, each clonal population corresponded to a primary tumor or metastatic family. Tumors were identified with two approaches: first, by deconvolution with MULTI-seq (and filtering with Lenti-Cre-BC information; see below in section “Cell Filtering with Lenti-Cre-BC”); and second, by separating cells based on differing intBC sets. In the second approach, we used Cassiopeia to identify non-overlapping intBC sets and classify cells using the “call-lineages” command-line tool. Once clonal populations were identified, consensus intBC sets were identified (see “Creating a consensus intBC set for mESC clones” below). All summarized molecular information for a given cell (cellBC, number of UMI, intBC, indel allele, read depth, etc) along with the assigned clonal identity were summarized in an allele table. Then, character matrices were formed for each clonal population, summarizing mutation information across the N cells in a population and their M cut-sites. Characters (i.e., cut-sites) with more than 80% missing information or containing a mutation that was reported in greater than 98% of cells were filtered out for downstream tree reconstruction.

Creating a consensus intBC set for mESC clones

Given that each mouse is generated from a specific mESC clone, we expected tumors from each mouse would maintain the same set of intBCs as the parental mESC clone. To identify this con-

sensus set of intBCs, we stratified tumors based on which mESC clone they originated from, and within these groups computed the proportion of tumors that reported a given intBC in at least 10% of cells. We determined cutoffs separating reproducible intBCs from irreproducible intBCs for each mES clone separately. These consensus intBC sets were used for downstream reconstruction of phylogenies.

Tree Reconstruction with Cassiopeia

Trees for each clonal population (see “Calling clonal populations and creating character matrices” above) were reconstructed with Cassiopeia-Hybrid [127]. Briefly, Cassiopeia-Hybrid infers phylogenies by first splitting cells into clusters using a “greedy” criterion (Cassiopeia-Greedy) until a user-defined criteria is met at which point each cluster of cells is reconstructed using a near-optimal Steiner-Tree maximum-parsimony algorithm (Cassiopeia-ILP). We compared the parsimony of trees generated using two different greedy criterions - both criterions employed work by first identifying a mutation and subsequently splitting cells based on whether or not this mutation was observed in a cell. First, we used the original Cassiopeia-Greedy criterion, which identifies mutations to split cells on by using the frequency and probability of mutations. Second we applied a compatibility-based criterion which prioritizes mutations based on character-compatibility (see section “Compatibility-based greedy heuristic for tree reconstruction” below). We proceeded with the more parsimonious tree. In one specific case, (3515_Lkb1_T1), we observed that the lineage tracing alleles were not adequately captured with phylogenetic inference of the primary tumor alone. To handle this, we

rebuilt the tree of the tumor-metastasis family and then subset the phylogeny to consist of only the cells from the primary tumor - resulting in a clonal phylogeny that appeared to be better supported by allelic information.

In most inferences, we used indel priors computed with Cassiopeia to select mutations with a Cassiopeia-Greedy algorithm as well as weight edges during the Steiner-Tree search with Cassiopeia-ILP. Generally, we used an LCA-based cutoff to transition between Cassiopeia-Greedy and Cassiopeia-ILP as previously described [203]. Clone-specific parameters are reported in Table S1.

Compatibility-based greedy heuristic for tree reconstruction

A rare, but simple case for phylogenetic inference is that of perfect phylogeny in which every character (or cut-site) is binary (that is, can be cut or uncut) and mutates at most one time. In this regime, every pair of characters is “compatible” – that is, given two binary characters i and j , the sets of cells that report a character i as mutated are non-overlapping with the set of cells that report character j as mutated, or one set of cells is completely contained within the other.

In this approach, we used a heuristic, called the compatibility index, to measure how far a pair of characters is from compatibility. To do so, we first “binarized” our character matrices by treating each unique (cut-site, mutation) pair as a binary character. (To note this binarization procedure is possible because of the irreversibility of Cas9 mutations and discussed in our previous work [127].) Then, we found the character that had deviated the least from perfect phylogeny, that is violated compatibility the least. To find this character, we first built a directed “compatibility-graph”, where individual nodes

represented characters and edges between nodes represented deviations from compatibility. Each edge from character i to j was weighted as follows:

$$w_{i,j} = -n_j \log(p_j)$$

where i and j are two incompatible characters, n_j is the number of cells reporting character j , and p_j is the prior probability of character j mutating. For the purposes of building this compatibility matrix, missing data was ignored (this is, no node in the graph corresponded to a missing state). A character c' to split cells with was identified by minimizing the sum of weights emitted from the node:

$$c' = \arg \min_{c \in X} \sum_{j \in Out(c)} w_{c,j}$$

where $Out(c)$ denotes the set of edges with c as a source. This process was repeated until the tree was resolved completely, or a criterion was reached as in Cassiopeia-Hybrid.

Cell Filtering with Lenti-Cre-BC

After performing tree reconstruction for each clonal population, leaves were annotated with Lenti-Cre-BC information and evaluated manually for filtering. Specifically, in tumors with adequate Lenti-Cre-BC information, we identified subclades (defined here as clades that joined directly to the root) that clearly had divergent Lenti-Cre-BC information. This combined Lenti-Cre-BC and lineage analysis helped minimize the influence of lenti-Cre-BC dropout in single-cell experiments. These subclades were subsequently removed and cells were filtered out from the phylogenetic analysis. In

one case (3513_NT_T4 and 3513_NT_T5), two tumor populations were split from a parental tumor (3513_NT_N2), reconstructed, and used in downstream analyses.

CNV analysis

Chromosomal copy number variations (CNV) were inferred with the InferCNV R package (version 1.2.1), which predicts CNVs based on single-cell gene expression data. InferCNV was run in 'sub-clusters' analysis mode using 'random_trees' as the subclustering method. Genes with less than one cell were filtered with the 'min_cells_per_gene' option, and no clipping was performed on centered values ('max_centered_threshold' set to 'NA'). The cutoff for the minimum average read count per gene among reference cells was set to 0.1, per software recommendation for 10X data. CNV prediction was performed with the 'i6' Hidden Markov Model, whose output CNV states were filtered with the included Bayesian mixture model with a threshold of 0.2 to find the most confident CNVs. All other options were set to their default values.

Each tumor sample was processed independently with normal lung cells (identified solely from the MULTI-seq deconvolution pipeline) as the reference cells. The number of CNVs for each cell was computed by counting the number of CNV regions predicted. We filtered cells with CNV counts greater than three standard deviations away from the mean of each tumor, in addition to cells with greater than or equal to 20 predicted CNVs. When comparing CNV counts of cells in expansions against those of cells in non-expansions, statistical significance was computed with a one-sided permutation test and the Mann-Whitney U-test, both of which yielded the same results.

We applied hierarchical clustering with a euclidean distance metric and the “ward” linkage to identify CNV clusters of cells within each tumor. For each clustering induced by cutting the hierarchical clustering dendrogram at different heights, we computed the probability that a cell and its nearest neighbor on the Cassiopeia tree were in the same hierarchical cluster (“nearest neighbor probability”). These clusters ranged from most coarse-grained (low cutoff height) to the most fine-grained (high cutoff height). When there were multiple nearest neighbors, pseudocounts were used by taking the fraction of nearest neighbors that were in the same cluster. We performed nonparametric Permutation Tests for each unique clustering by shuffling the cluster assignments of the cells and computing the nearest neighbor probability using these assignments.

Tree Quality Control for Fitness Inference

Trees were subjected to quality control before identifying subclones under positive selection and single-cell fitness inference. We employed two quality control metrics: first, a measure of subclonal diversity known as “percent unique indel states”, defined as the proportion of cells that reported a unique set of character states (i.e., mutations). Second, we also filter lineage trees based on the level of “unexhausted target sites” defined as the proportion of characters (i.e., specific cut sites) that were not dominated by a single mutation (i.e, more than 98% of cells contained the same mutation). These metrics describe the diversity and depth of the lineage trees, and enable filtering out tumors with poor lineage tracing quality (i.e., lineage tracing capacity became saturated too early during tumor development). Using these two metrics, we filtered out tumors that had less than 10% unique indel

states or less than 20% unexhausted target sites. Additionally, tumors with too few cells recovered (fewer than 100 cells) were ignored for this analysis because of a lack of power to confidently quantify subclonal behavior.

Identifying subclonal selection (i.e., expansions)

Subclones undergoing positive selection were identified by comparing the number of cells contained in the subclone to its direct “sisters” (i.e. branches emanating directly from the parent of a subclone of interest) and computing a probability of this observation with a coalescent model. Specifically, consider a node v in a particular tree with k children stored in the set C . Let n_c denote the number of leaves below a particular node c (and observe that $N = n_v = \sum_{c \in \text{children}(v)} n_c$). Under the coalescent model, we can compute a probability indicating how likely a subclone c under v would have exactly n_c leaves given v had N total leaves as follows:

$$p_{N,k}(n_c) = \frac{\binom{N-n_c-1}{k-2}}{\binom{N-1}{k-1}}$$

Finally, we computed the probability that a subclone c under v would have at least n_c leaves given v had N total leaves is:

$$p_{\hat{N},k}(n_c) = \sum_{n=n_c}^{N-k+1} p_{N,k}(n)$$

Nodes with probabilities $p_{\hat{N},k}(n_c) < 0.01$, at least a depth of 1 from the root, and containing subclades with at least 15% of the total tree population were annotated as undergoing an “expansion”.

In the analysis presented in this study, we additionally filtered out nodes annotated as “expanding” if they contained another node in their subtree that was also expanding. Expansion proportions were calculated as the fraction of the tree consisting of cells residing in any subclade called as “expanding”.

Inferring single-cell fitness

To compute single-cell fitness, we used the “infer_fitness” function from the jungle package (publicly available at <https://github.com/felixhorns/jungle>) which implements a previously described probabilistic method for inferring relative fitness coefficients between samples in a clonal population [186]. Because some trees contained exhausted lineages (i.e., those in which all target sites were saturated with edits), after filtering out trees that did not pass quality control (see section “Tree quality control for fitness inference” above), we pre-processed branch lengths on each phylogeny such that branches had a length of 0 if no mutations separated nodes and 1 if not. In essence, this collapses uninformative edges in the fitness inference and helps control for lineage exhaustion. After this procedure, we were left with fitness estimates for each leaf in a phylogeny, normalized to other cells within the phylogeny.

Tumor fitness differential expression

Genes differentially expressed along the fitness continuum within each tumor were identified with a linear regression approach. Specifically, given a cell i , we can compare two models for the ex-

pression of some gene j according to the cell's fitness score f_i and its number of total UMIs, s_i , as follows:

$$H_0 : \log(1 + e_{i,j}) \sim f_i$$

$$H_a : \log(1 + e_{i,j}) \sim f_i + s_i$$

Where $e_{i,j}$ is the count-normalized expression of gene j in cell i (we used the median number of UMI counts across the dataset to normalize expression level). Only genes appearing in more than 10 cells were retained for differential expression analysis. Linear models were fit using julia's 'GLM' package function, and p-values were computed via a likelihood ratio test between the null and alternative hypothesis likelihoods. P-values were FDR corrected using the Benjamini-Hochberg procedure [17]. Log₂-fold-changes were computed by comparing the average expression of a gene in the top vs bottom 10th percentile of fitness scores.

Meta-analysis and derivation of the FitnessSignature

The transcriptional FitnessSignature was derived from the results of individual tumor fitness differential expressions with a majority-vote meta-analysis. This approach ranks genes based on the number of times that a gene is differentially expressed (FDR < 0.05 and |log₂FC| > log₂(1.5)) and the consistency of its direction. We used the MetaVolanoR R package (version 1.0.1) to perform this majority-vote analysis, which computed both of these values. We identified consistently differentially

expressed genes for our transcriptional FitnessSignature if a gene appeared to show up at least 2 times in the same direction, and if the ratio between frequency and consistency was greater than 0.5.

Fitness module identification

We determined transcriptional fitness gene modules using the Hotspot package (version 0.9.0 [57]). To do so, we first subset our processed single-cell expression matrix (see section below “Single-cell transcriptome analysis for KP-Tracer data”) to contain only the 1,183 genes in the FitnessSignature that were positively associated with fitness. Then, using Hotspot we identified fitness-related genes that were significantly autocorrelated with the scVI latent space using the “danb” observation model and 211 neighbors (the square-root of the number of cells in the expression matrix). After this procedure, genes with an FDR of less than 0.05 were retained for downstream clustering. We then computed pairwise local autocorrelations with Hotspot and clustered genes using these pairwise statistics with the “create_modules” function in Hotspot (minimum gene threshold of 100, FDR threshold of 0.05, core_only=False). This procedure identified three modules that were used for downstream analysis.

Single-cell transcriptome processing for KP-Tracer NT data

The scRNA-seq was processed using the 10X CellRanger pipeline (version 2.1.1) with the mm10 genome build. Cells were assigned to a sample using the MULTI-seq pipeline described above (see

section "Single-cell preprocessing pipeline"). After quantification, informative genes were identified using the Fano filtering process implemented in VISION [58], and raw counts were batch-corrected (using the batch-harvest data, indicating when a batch of mice were sacrificed as the batch variable) and projected into a shared latent space of 10 dimensions with scVI [164]. Cells were initially clustered with the Leiden algorithm as implemented in Scanpy [282], and two clusters dominated by cells annotated as normal and cells that could not be confidently mapped to a tumor via MULTI-seq or Lenti-Cre-BC analysis (see section "Single-cell preprocessing pipeline" and "Cell Filtering with Lenti-Cre-BC" above) were removed from downstream analysis. Clusters were then manually re-clustered to obtain segmentations that aligned with gene expression patterns. After this process, we were left with a total of high-quality 58,022 cells with single-cell transcriptomic profiles from KP mouse tumors. Single-cell counts were normalized by the median UMI count across cells and logged to obtain log-normalized data. Gene markers for each Leiden cluster were identified using the Wilcox rank-sums test on the log-normalized gene counts with the Scanpy package.

Integration of normal lung epithelium transcriptomes

scRNA-seq data of cells obtained from various tissues in sample L46 were quantified using the 10X CellRanger pipeline (version 2.1.1) with the mm10 genome build. Cells were assigned to a sample (one of 4 tissues) using the CellRanger multi procedure. After quantification and sample assignment, cells with fewer than 200 UMIs and genes appearing in fewer than 1% of cells were filtered out. This left us with 14,424 high-quality cells. A low-dimensional embedding was inferred

using scVI on the dataset with the 4000 most highly-variable genes (using the "seurat_v3" flavor of Scanpy's `highly_variable_genes` function). Transcriptional clusters were identified using the Leiden community detection algorithm. One cluster of 329 cells consisted of normal lung cells and expressed gene markers *Nkx2-1*, *Sftpc*, and *Scgb1a1*; we isolated and annotated this cluster as normal lung epithelial cells (primarily AT2 and club cells).

This dataset of 329 normal lung epithelial cells (isolated from the L46 sample, as described above) was integrated into the scRNA-seq dataset of KP tumors (see section "Single-cell transcriptome processing for KP-Tracer NT data") using scVI. Specifically, we used scVI to batch-correct these two datasets and project all cells into a common coordinate system. Then, we visualized this scVI batch-corrected embedding with UMAP.

Differential expression analysis of Chuang et al

TPM-normalized RNA-seq data were downloaded from GEO accession GSE84447. Samples were split into early and late-stage tumor groups based on the author annotations: tumors annotated with "KPT-E" were assigned to the early stage group and tumors with "TnonMet" or "TMet" annotations were assigned to the late group. Then, we log-normalized the TPM counts and used the limma R package (version 3.36.3) to infer differentially expressed genes with the "eBayes" function. Genes passing an FDR threshold of 0.05 and log₂-fold-change threshold of 1 (in either direction) were called differentially expressed and used for comparison with the FitnessSignature described in this study.

FitnessSignature analysis of Marjanovic et al

Raw expression count matrices were downloaded directly from GEO, accession number GSE152607. Gene counts were normalized to transcript length, to account for read depth artifacts in the Smart-Seq2 protocol. VISION [58] was used to compute FitnessSignature scores (using the FitnessSignature gene set described in our study) for each cell in the dataset and scores were averaged within time points of KP mice.

Survival analysis with TCGA lung adenocarcinoma tumors

The fitness signature genes including 1183 up-regulated genes and 1027 down-regulated genes from mice experiments were converted to corresponding genes from the H. sapiens genome (build hg19), resulting in 1126 up- and 970 down-regulated human genes, respectively. FitnessSignature with only up-related genes was denoted as FSU, FitnessSignature with only down-related genes was denoted as FSD. TCGA Lung adenocarcinoma cohort with RNAseq data (n=495) were stratified into FSU-High, FSU-Low, FSD-High, and FSD-Low according to median expression of sum of FitnessSignature genes, then, patients harboring genes with FSU-High and FSD-Low formed a group, patients containing FSU-Low and FSD-High gene expression formed another group. Subsequently, these two groups were used for survival analysis using the survival package in R (version 3.2.11). The survival analysis was invoked with the call "survfit(Surv(Time, Event) ~ Group)" where "Group" is the FitnessSignature-based stratification. Kaplan–Meier curve is shown with a log-rank statistical test. For fitness gene module 1, 2, and 3 analyses, patients were divided into module

gene expression of High and Low based on the median of the sum of gene expression, followed by survival analysis.

Fitness Module Enrichment

Each of the three fitness gene module scores (computed with VISION) were normalized to the range [0, 1] across all NT cells. All NT cells in non-expansions were defined as the background cells, and the background module scores were calculated by averaging the normalized module scores of these cells. Additionally, the module scores of cells in each expansion were averaged to obtain the pseudo-bulk module score for each expansion. These module scores were divided by the background module scores, yielding the module enrichment score (i.e. fold-change versus background) per fitness module. These scores were plotted on a personality plot for visualization. Every expansion was assigned (non-exclusively) to the three fitness modules using a permutation test to test whether the cells in the expansion exhibited a significant increase in fitness module score compared to non-expanding background cells ($p < 0.05$).

Calculation of single-cell and Leiden cluster EffectivePlasticity

EffectivePlasticity for each tumor was computed by first calculating a normalized parsimony score for the tumor tree, with respect to the Leiden cluster identities at the leaves, using the Fitch-Hartigan algorithm [72, 104]. Briefly, this procedure begins by assigning cluster identities to the leaves of the tree, and then calculates the minimum number of times a transition between cluster identities must

have happened ancestrally in order to account for the pattern observed at the leaves. To compare scores across trees, we normalize these parsimony scores by the number of edges in the tree, thus giving the EffectivePlasticity score. In all analyses, we filtered out cells that were part of Leiden clusters that were represented in less than 2.5% of the total size of the tree.

In order to generate single-cell EffectivePlasticity ("scEffectivePlasticity"), we computed the EffectivePlasticity for each subtree rooted at a node on the path from the root to a leaf and averaged these scores together. This score thus represents the average EffectivePlasticity of every subtree that contains a single-cell.

To generate average EffectivePlasticity for each Leiden cluster, we first stratified cells in each tumor according to the Leiden cluster. Then, we averaged together scores within each tumor for each Leiden cluster, thus providing a distribution of EffectivePlasticity for each Leiden cluster.

Calculation of the Allelic EffectivePlasticity score

The Allelic EffectivePlasticity score provided a "tree-agnostic" measurement of a cell's effective plasticity. Qualitatively, the score measures the proportion of cells that are found in a different Leiden cluster than their closest relative (as determined by the modified edit distance between two cells' character states; see section "Allelic Coupling" for the definition of this distance metric). Importantly, if a cell has more than one closest relative, each of their votes are normalized by the number of equally close relatives this cell has. More formally, the single-cell Allelic EffectivePlasticity was defined as:

$$a(i) = \frac{1}{|K|} \sum_{k \in K} I(\text{leiden}(k) == \text{leiden}(i))$$

where K indicates the set of a cell's closest relatives, as measured by modified edit distance, $\text{leiden}(i)$ indicates the Leiden cluster that cell i resides in, and $I(\cdot)$ is an indicator function that is 1 if the two Leiden clusters are the same and 0 otherwise. The Allelic EffectivePlasticity of a tumor is the average of these scores:

$$A(\text{tumor}) = \frac{1}{|L|} \sum_{l \in L} a(l)$$

where L is the set of all leaves in the tumor.

Calculation of the L2 EffectivePlasticity score

The L2 EffectivePlasticity score served as an alternative tree-based score that accounted for random noise at the boundary between two Leiden clusters, as opposed to treating each Leiden Cluster as a point. As with the EffectivePlasticity score, we first found nearest-neighbors of each cell i using the phylogenies and considered neighbors found in a different Leiden cluster than i . Yet, in contrast to the EffectivePlasticity score, we distinctly used an L2-distance in the 10 dimensional scVI latent space to obtain a measure of how distinct the neighbor was. Mathematically, the single-cell L2 EffectivePlasticity score was defined as:

$$l_2(i) = \frac{1}{|K|} \sum_{k \in K} \|x_i - x_k\|_2$$

Where K indicates the set of a cell's closest relatives, as found with the phylogeny, and x_i indicates the 10-dimensional embedding of cell i 's single-cell expression profile in scVI space. The L2 EffectivePlasticity of a tumor was defined as the average across all leaves in the tumor.

Evolutionary Coupling

Evolutionary Coupling is the normalized phylogenetic distance between any pair of variables on a tree. Mathematically, given two states M and K that can be used to label a subset of the leaves of the tree, we compute the average distance between these states:

$$D(M, K) = \frac{1}{n_m n_k} \sum_{m \in \{M\}, k \in \{K\}} d_T(m, k)$$

where n_M is the number of leaves with state M , $\{M\}$ denotes the set of cells in set M , and $d_T(i, j)$ denotes the phylogenetic distance between leaves. There are multiple ways to score $d_T(i, j)$, and here we used the number of mutated edges for our analysis (i.e., the number of edges separating two leaves i and j that carried at least one mutation). To normalize these distances, we compare $D(M, K)$ to a random background generated by shuffling the leaf assignments 2,000 times. Then, to obtain background-normalized scores, we Z-normalize to the random distribution D_R :

$$D'(M, K) = \frac{D(M, K) - E[D_R(M, K)]}{SD[D_R(M, K)]}$$

This score is obtained for all pairs of states in a tumor that pass a 2.5% proportion threshold (i.e.,

we filter out cells in states that fall below this threshold). Then, from the matrix of all background-normalized phylogenetic distances, P (such that $P_{M,K}$ is equal to $D'(M, K)$), we compute the Evolutionary Couplings between two states M and K by Z-normalizing P :

$$E(M, K) = \frac{P_{M,K} - E[P]}{SD[P]}$$

Evolutionary Couplings presented in **Figure 5.5B** and **5.5D** are normalized as:

$$E(\hat{M}, K) = \exp\left(-\frac{E(M, K)}{\max(abs(E))}\right)$$

Where E denotes all the Evolutionary Couplings between states in a given tumor.

Allelic Coupling

We used modified edit distances between cells to compute an Allelic Coupling score that could be used to assess consistency of the Evolutionary Coupling results. Here, we used a modified edit distance, $h'(a_i, b_i)$, that scored the distance between sample a and b at the i^{th} character:

$$h'_p(a_i, b_i) = \begin{cases} -\log(p(a_i)) - \log(p(b_i)) & \text{if } a_i \neq b_i \text{ and } a_i, b_i \text{ are mutated} \\ -\log(p(a_i)) & \text{if } a_i \text{ mutated, } b_i \text{ unmutated} \\ -\log(p(b_i)) & \text{if } b_i \text{ mutated, } a_i \text{ unmutated} \\ \log(p(a_i)) + \log(p(b_i)) & \text{if } a_i = b_i \text{ and } a_i, b_i \text{ are mutated} \\ 0 & \text{otherwise} \end{cases}$$

The allelic distance between two samples a and b is $\sum_{i \in X} h'(a_i, b_i)$. We used these distances instead of phylogenetic distances to compute the coupling statistic described in the section above entitled "Evolutionary Coupling" and called this new coupling statistic "Allelic Coupling".

K-nearest-neighbor (KNN) Coupling

K-nearest-neighbor (KNN) coupling was computed by using d_T as the distance to the k^{th} neighbor in the Evolutionary Coupling statistic. We used the same phylogenetic distance described in the section entitled "Evolutionary Coupling" to compute the k^{th} neighbor and used $k = 10$ for the analysis.

Fate clustering

To identify separate fates in the KP-Tracer dataset, we first computed Evolutionary Couplings in each tumor for all pairs of states. To remove noise intrinsic to the clustering, we filtered out clusters that accounted for less than 2.5% of the tumor. As a phylogenetic distance metric, we used the number of mutated edges (i.e., any edge that contained at least one mutation was given a weight of 1 and

otherwise the edge was weighted as 0). Before computing Evolutionary Couplings, we preprocessed the lineages such that each leaves with the same Leiden cluster were grouped together (see section entitled “Preprocessing lineages with respect to states”).

After calculating the Evolutionary Coupling for all pairs of states within each tumor, we concatenated all vectors of Evolutionary Coupling together into a matrix. We additionally converted Evolutionary Couplings to similarities by exponentiating these values (i.e, $E'(M, K) = \exp(-E(M, K))$). As additional features for this clustering, we also added Leiden cluster proportions to each tumor’s vector of couplings. Then we Z-normalized across features to compare tumors and clustered this transformed matrix using a hierarchical clustering approach in the python scipy package (version 1.6.1). We used a Euclidean metric and the “ward” linkage method. We identified three clusters from this hierarchical clustering, corresponding to our three Fate Clusters. These three Fate Clusters were visualized using Uniform Manifold Approximation and Projection (UMAP) on the Evolutionary Coupling and Leiden cluster proportion concatenated matrix. Important couplings were identified using Principal Component Analysis on the same Evolutionary Coupling concatenated matrix.

Preprocessing lineages with respect to states

In some lineages, we observed that polytomies (or non-bifurcating) subclades were created at the very bottom of the tree due to the saturation of target site edits. Because this could artificially appear to make cellular states more closely related than they actually were, we took a conservative approach to making conclusions about cellular relationships between leaves in such polytomies. Specifically,

we first assigned states from a state space Σ to each leaf in a tree according to some function $s(l) \rightarrow \sigma \in \Sigma$ for all l leaves in the tree. Then, for all polytomies that contained at least unique states or more, we created extra splits in the tree for each unique state. More formally:

```

1: function PREPROCESS-LINEAGE(phylogeny = Tree)
2:   for all  $n \in \text{Tree}$  do
3:     states = []
4:     if len(children( $v$ ) < 3 then
5:       continue
6:     for all  $c \in \text{children}(v)$  do
7:       if is_leaf( $c$ ) then
8:         states.append( $c$ )
9:     if len(unique(states)) > 2 then
10:      for all state in unique(states) do
11:        Tree.add_edge( $v$ , 'new-node-{state}')
12:      for all  $c \in \text{children}(v)$  do
13:        if  $\sigma(c) == \text{state}$  then
14:          Tree.add_edge('new-node-{state}',  $c$ )
15:          Tree.remove_edge( $v$ ,  $c$ )
16:   return Tree

```

Aggregating Evolutionary Coupling across Fate Cluster

To create a consensus Evolutionary Coupling map across the tumors in a Fate Cluster, we first computed the average Evolutionary Coupling between all pairs of states in a tumor as described previously. Then, we computed an average Evolutionary Coupling for each pair of states, normalizing by the number of tumors that this pair appeared in above the requisite 2.5% threshold. Critically, we removed patterns that were driven by a small proportion of cells, we only considered states that appeared in at least 2.5% of the total number of cells across all tumors in a Fate Cluster.

Phylotime

Phylotime was defined as the distance to the first ancestor that could have been a particular state. To approximate the Phylotime in this study, we defined the initial AT2-like state (Leiden cluster 4) as the ground state, and inferred the sets of states for each ancestor with the Fitch-Hartigan bottom-up algorithm [72, 104]. Then, in each tumor, we computed the phylogenetic distance separating each cell from its closest ancestor that could have been an AT2-like cell, as determined with the Fitch-Hartigan bottom-up algorithm. Phylogenetic distances were defined as the number of non-zero-length branches (though we compare the consistency of Phylotime to a distance metric that uses the number of mutations along each edge in **Figure 5.12J,K**). Here, the tree structure is advantageous in modeling divergence times from the AT2-like state because it can account for homoplasy (i.e., the same mutation occurring independently) and convergent evolution (i.e., the same transcriptomic state being reached separately) events. Thus, it is preferable, in principle, to comparing the mutation states directly between a leaf and all AT2-like cells. Phylotime within each tumor was normalized to a 0-1 scale. Once every tumor was analyzed this way, Phylotime across tumors was merged by performing an average-based smoothing across the transcriptional space: specifically, for each cell, we found the 5 closest neighbors in transcriptional space (in the low-dimensional scVI latent space) and averaged Phylotimes within this neighborhood. After integrating together Phylotime in this manner, the final distribution across tumors was normalized once again to a 0-1 scale.

Phylotime differential expression

Genes associated with Phylotime in each Fate Cluster were identified using the Tradeseq package [264]. Specifically, for each Fate Cluster, lowly-expressed genes were filtered if they were detected in fewer than 10% of cells and high-variance genes were identified with the Fano filtering procedure implemented in VISION [58]. Then, in each cluster, expression models were fitted with the “fitGAM” function and genes associated with a specific segment of Phylotime were identified with the “associationTest” function. P-values were FDR corrected using the Benjamini-Hochberg procedure [17], and significant genes were retained if they had an FDR below 0.05 and a mean log₂-fold-change above 0.5. Smoothed expression profiles were predicted with the Tradeseq package using the models fit from the fitGAM procedure and genes were subsequently clustered into those expressed early and late. Gene set enrichment analysis was performed using the enrichR R package (version 3.0) after converting gene names from mm10 to GRCh38. We used the Biological Process gene ontology, ChEA, and MsigDB Hallmark gene sets. Informative genes were manually selected from the set of genes passing the significance and effect-size thresholds, and manually clustered for display in **Figure 5.5**.

Integrating transcriptomes of KPL and KPA data

The scRNA-seq data was processed using the 10X CellRanger pipeline (version 2.1.1) with the mm10 genome build. Cells were assigned to a sample using the MULTI-seq pipeline as described above (see section “Single-cell preprocessing pipeline”) to form a raw count matrix consisting of

cells from KP, KPL, or KPA mice. Cells with fewer than 200 genes detected, greater than 15% of mitochondrial reads, or greater than 7000 genes detected were filtered out. Cells were batch-corrected and projected into 20 latent dimensions using scVI (Lopez et al. 2018) with 2 hidden layers and the library batch as a batch covariate on the top 4000 most variable genes, as detected with Scanpy's 'highly_variable_genes' function with the "seruat_v3" flavor [282]. Clusters were identified with the Leiden algorithm [259] with manual parameter selection to obtain an acceptable resolution. All normal cells and seven additional clusters with high proportions of normally-annotated cells (as with MULTI-seq or via the lineage-tracing data) were filtered out for downstream analysis (a total of 2,209 cells in the entire dataset).

To perform label transfer from the KP-Tracer dataset, we first labeled all KP cells in the integrated dataset with previous annotations and labeled all new cells with "Unknown". Then, we used scANVI [287] to predict labels of cells from KPL and KPA mice using 40 latent dimensions, 2 hidden layers, and a dropout rate of 0.2. Upon inspecting predictions, we elected to keep predictions made by scANVI for the majority of cells, with the exception of 5 new Leiden clusters identified by clustering the scVI latent space. Additionally, we elected to merge one new Leiden cluster with the Pre-EMT state because key gene expression markers across these two states were consistent. After this process, we were left with a total of 104,197 high-quality cell transcriptomes.

Differential expression analysis of Pre-EMT state

The single-cell RNA count matrix was first count-normalized to the median number of UMI counts across cells and log-transformed. Then, cells assigned to the Pre-EMT state were separated into three non-overlapping sets according to their genotype (KP, KPL, or KPA). Differentially expressed genes in the KPL subset of cells in the Pre-EMT cluster were identified by comparing these cells to all other cells with Scanpy using a t-test on log-normalized count matrix with the top 5000 most variable genes. Highlighted genes were selected from the set genes passing an FDR cutoff of 0.05 and a log2FC cutoff of 1.

Evolutionary Trajectory Analysis of KPL and KPA Tumors

The evolutionary trajectories from KPL and KPA mice were analyzed identically to the KP tumors as described in the previous section entitled "Fate Clustering". Briefly, each tumor was described as a vector of Leiden cluster proportions and exponentiated Evolutionary Couplings (i.e, $E'(M, K) = \exp(-E(M, K))$). Vectors were concatenated together and Z-normalized across features. The resulting matrix was decomposed and analyzed using Principal Component Analysis (PCA) and informative features were identified by evaluating the features with highest principal component loadings.

Evolutionary Coupling of 3724_NT_T1 Tumor-Metastasis Family

Using the tumor-metastasis family tree for 3724_NT_T1 and associated metastases, we computed the Evolutionary Couplings between each microdissected piece of the primary tumor (T1-15) and

each metastasis (the statistic is described in the section entitled “Evolutionary Coupling”). Normalized Evolutionary Couplings (E) were computed as described previously.

Phylogenetic distances on Tumor-Metastasis Family trees

In each of the tumor-metastasis families (defined as a tumor containing both a primary tumor and a large enough metastatic population) analyzed in **Figure 5.7** and **5.14**, we first reconstructed trees encompassing all cells in the primary and metastatic tumors (referred to as a “tumor-metastasis family” tree). Then, we stratified cells in the primary tumor by the expansions called with our expansion-calling statistic (see above, “Identifying subclonal selection”). If a cell was not part of an expansion, it was labeled as “non-expansion”. Then, for each cell in a metastatic tumor, we computed the average modified phylogenetic distance to all primary tumor cells in the tumor-metastasis family tree. The modified phylogenetic distance was computed as the sum of branch lengths, where each branch length was defined as the number of mutations separating each node from one another (as inferred using Camin-Sokal parsimony - i.e., irreversibility of mutations).

Transcriptional distances on Tumor-Metastasis Family trees

Tumor-metastasis family trees were inferred and stratified as described above (see “Phylogenetic distances on Tumor-Metastasis Family trees”) and Euclidean distance was used to measure transcriptomic differences between metastatic cells and primary tumor subpopulations.

5.6 Supplementary Figures

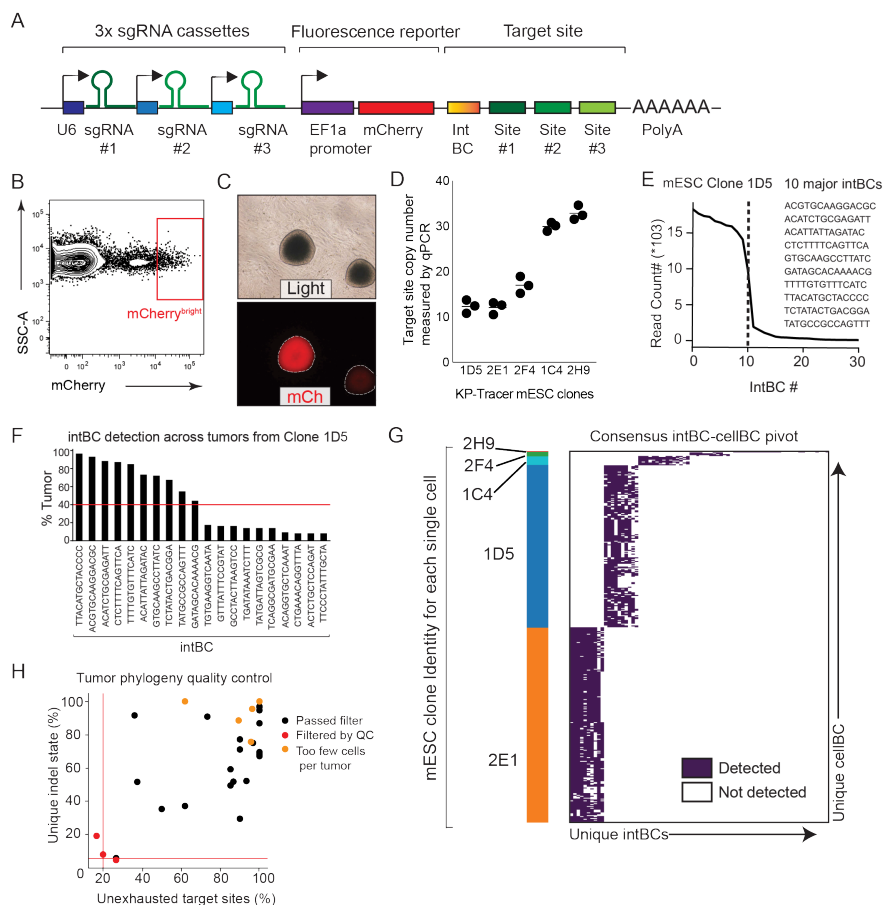


Figure 5.8: KP-Tracer mouse genetic components, validation, and quality-control. (A) The piggyBac transposon-based lineage tracing vector libraries used to engineer the KP-Tracer mice contained (1) a triple-guideRNA cassette and (2) a target site library cassette with a 14bp integration barcode (“intBC”) and three CRISPR/Cas9 cut sites on the 3’ UTR of an mCherry reporter gene. (B) Enrichment of mESC population with high lineage-tracer expression based on high mCherry expression (a reporter indicating lineage tracer expression). These cells are then single-cell cloned before generating chimeric KP-Tracer mice. (C) Representative images of specific mCherry positive mESC clones that express the lineage tracing vectors. (D) Copy number of lineage tracing vectors across 5 mouse embryonic stem cell (mESC) clones used in this study measured by genomic qPCR are shown. (E-F) Detection of unique lineage tracing target site intBCs for a representative mESC clone (1D5) using (E) DNA-sequencing and (F) scRNA-seq. A consensus set of target sites intBCs for each mESC clone was determined by selecting intBCs detected in at least 40% of all tumors derived from that mESC clone. (G) The consensus intBC pivot table across all five mESC clones used in this study to generate KP-Tracer mice. Each row is a single cell and is annotated with which mESC clone it came from. Each column is a unique intBC. Colors in the heatmap indicate whether or not an intBC was detected in a given cell. (H) Quality-control filtering of tumor phylogenies for subclonal expansion analyses. Quality of lineage-tracing data was assessed with two metrics: first, the percentage of cells that contained a unique set of mutations (“% unique indel state”; Methods); and second, the percentage of target sites that had to be filtered because of low-diversity (“target site saturation”; Methods). Tumors with less than 5% overall unique indel state, greater than 80% target site saturation, or fewer than 100 cells were filtered out.

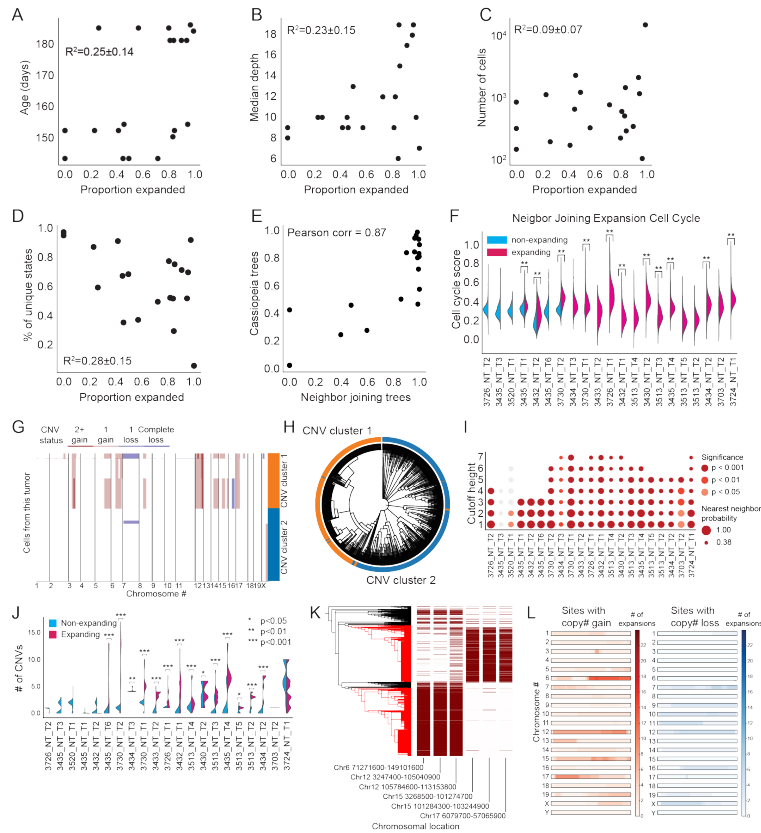


Figure 5.9: Characterization of KP-Tracer tumor subclonal expansions. (A-D) Phylogenetic features of tumor lineages and their predictiveness (as measured with R^2) on the expansion proportion of a tumor. Features evaluated were (A) age, (B) median tree depth, (C) size measured in the number of cells, and (D) proportion of unique cells. (E) Expansion proportion of tumors measured from Neighbor-Joining trees versus Cassiopeia trees. The percentage of cells in expansions were highly consistent between these two tree reconstruction strategies (Pearson's correlation = 0.87). (F) Comparison of cell-cycle scores inferred from transcriptomic profiles in expanding versus non-expanding tumor subclones, identified from Neighbor-Joining trees ($** p < 0.01$). (G-H) Representative example of comparison between hierarchical clustering of CNVs and Cassiopeia-reconstructed phylogeny. (G) The inferred CNVs are shown for the representative tumor, with the largest two clusters, identified via hierarchical clustering, indicated by the colorbar. (H) These two clusters are also indicated with unique colors on the Cassiopeia-reconstructed tumor phylogeny. The good correlation between CNV status and tumor phylogeny indicates the accuracy of tree reconstruction. (I) Heatmap displaying the probabilities that a cell and its nearest neighbor on the Cassiopeia-reconstructed phylogeny are in the same CNV cluster (size of circles). These probabilities were calculated for each tumor at various depths of the CNV hierarchical clustering dendrogram. The depth that yielded the most coarse-grained clusters were set to have a cutoff height of 1, with higher cutoff heights indicating finer clusters. The majority of Cassiopeia-reconstructed phylogenies were significantly consistent with CNV clusters (color of circles; Permutation Test) at all clustering resolutions. (J) A comparison of CNV counts in expanding versus non-expanding portions of tumors ($* p < 0.05$, $** p < 0.01$, $*** p < 0.001$). (K) An example of distinct CNV regions of cells from a single tumor. This tumor underwent two independent clonal expansions (red branches; left), each of which exhibited distinct CNV patterns (red bars; right). (L) An aggregated view of the CNV "hotspots" across subclonal expansions from all tumors. Each horizontal bar represents a chromosome, and the intensity of color indicates the number of subclonal expansions exhibiting a CNV in a region (see Methods). Regions that more often exhibited copy number gains are indicated in red (left); genomic regions that more often exhibited copy number losses are indicated in blue (right).

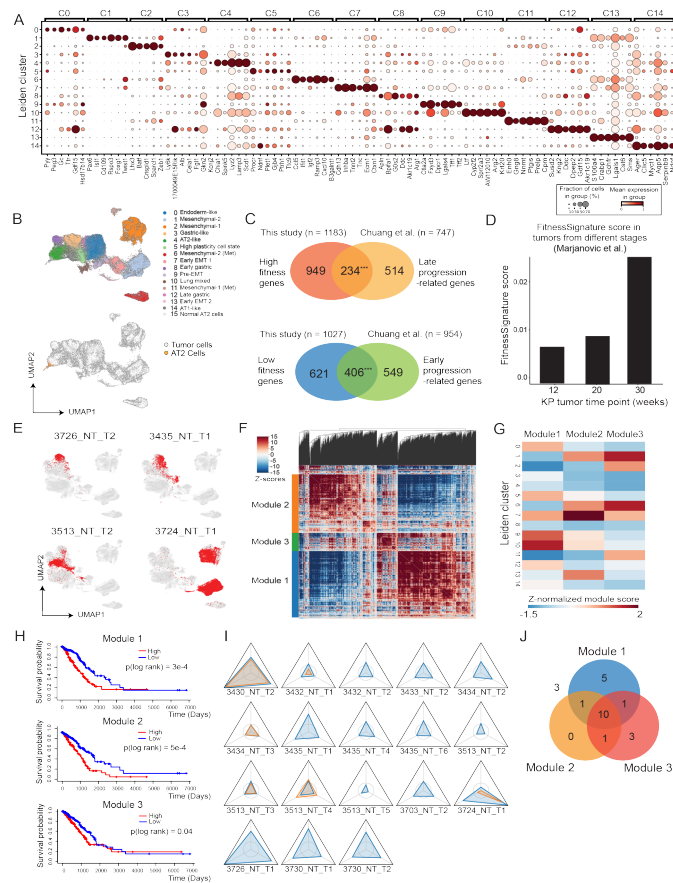


Figure 5.10: Characterization of KP-Tracer transcriptomic fitness landscape. (A) Gene markers for each Leiden cluster identified in the processed scRNA-seq latent space. Dot size indicates the percent of cells expressing the marker. Color indicates mean expression level. (B) Integration of normal lung epithelial cells with KP-Tracer dataset. Normal lung epithelial cells were isolated from an independent dataset and integrated with KP-Tracer tumors using scVI (Methods). Leiden cluster annotations from analysis of KP-Tracer tumors are shown (top) and normal cells are highlighted against tumor cells (bottom). Gene set enrichment analysis of genes associated with high fitness using the biological process (BP) gene sets. Selected sets passing an FDR cutoff of 0.2 are shown. Dot size indicates the number of genes that appear in the gene set of interest. (C) Gene set comparison between the FitnessSignature described in this study and KP tumor progression-associated genes described in [44]. Overlap significance assessed with a hypergeometric test ($*** = p < 1e - 5$). (D) Average transcriptional FitnessSignature score in KP tumors harvested at 12-week, 20-week, and 30-week timepoints from [170]. (E) Representative examples of tumors occupying distinct regions of the transcriptional space. Cells from the tumor of interest are shown in red, and all other cells are shown in gray. (F) Hotspot autocorrelation heatmap and clustering of genes that appear in the FitnessSignature and are positively associated with fitness. Gene modules are identified by distinct color strips on the left. Values in the heatmap are Z-normalized pairwise autocorrelation scores between genes. The dendrogram linking genes is shown for the columns. (G) Z-normalized mean fitness gene module signature scores of each Leiden cluster. (H) Kaplan-Meier plots for TCGA human lung adenocarcinoma patients with respect to genes in each fitness module. Curves are shown comparing overall survival of patient groups whose tumors have high (red) versus low (blue) expression of individual fitness gene modules, as determined by the median fitness module score. P -values from a log-rank test are indicated. (I) Fitness module enrichment personality plots. Each corner of the triangle represents the fold enrichment of an expansion's fitness module expression over expectation (non-expanding background). Independent expansions in each tumor are shown in unique colors (blue or orange). (J) Venn diagram illustrating the classification of expansions to gene modules based on a p -value threshold of 0.05 using a permutation test against non-expanding background.

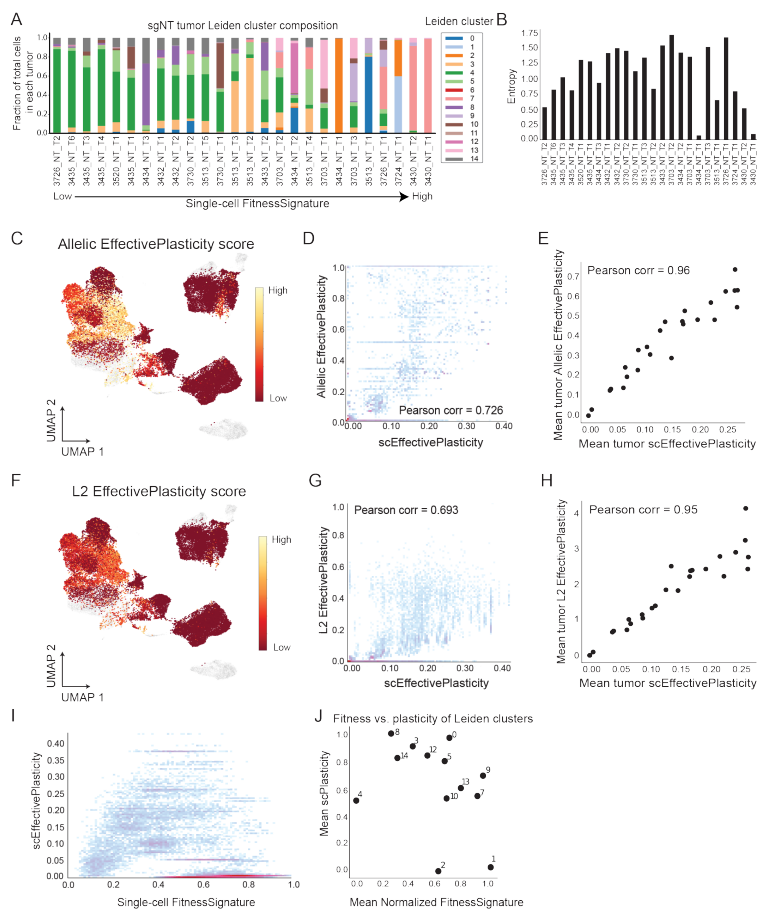


Figure 5.11: **Validation of KP-Tracer EffectivePlasticity score and comparison to FitnessSignature.** (A) Leiden cluster proportions for each KP-Tracer tumor. The fraction of cells in each Leiden cluster is shown for each tumor in a stacked bar plot, where each Leiden cluster is indicated by the unique color introduced in Fig 5.3A. Tumors are ordered by mean FitnessSignature score. (B) Shannon's Entropy statistic for each tumor, computed with the Leiden cluster proportions; tumors are ordered by mean FitnessSignature score. (C) Allelic EffectivePlasticity score overlaid onto two-dimensional gene expression UMAP is shown. Allelic EffectivePlasticity is an alternative way to quantify EffectivePlasticity by comparing transcriptional states between cells with similar lineage tracing indel states without using lineage trees. (D) Comparison of Allelic EffectivePlasticity to scEffectivePlasticity (Pearson's correlation = 0.73). Each point represents a single cell. (E) Comparison of mean tumor Allelic EffectivePlasticity to tumor EffectivePlasticity (Pearson's correlation = 0.96). Each point represents a tumor. (F) L2 EffectivePlasticity score overlaid onto two-dimensional gene expression UMAP is shown. L2 EffectivePlasticity is another alternative way to quantify EffectivePlasticity by computing dissimilarity in gene expression profiles between nearest neighbors on the phylogeny. (G) Comparison of single-cell L2 EffectivePlasticity to scEffectivePlasticity (Pearson's correlation = 0.69). Each point represents a single cell. (H) Comparison of mean tumor L2 EffectivePlasticity to mean tumor EffectivePlasticity (Pearson's correlation = 0.95). Each point represents a tumor. (I) Comparison of scEffectivePlasticity to single-cell FitnessSignature scores. Each point represents a single cell. (J) Weighted mean EffectivePlasticity vs mean FitnessSignature for each transcriptional state (Leiden cluster). The weighted Mean EffectivePlasticity for each Leiden cluster was determined by first computing the mean scEffectivePlasticity for each Leiden cluster in a tumor, and then averaging these values together. Each point represents a tumor.

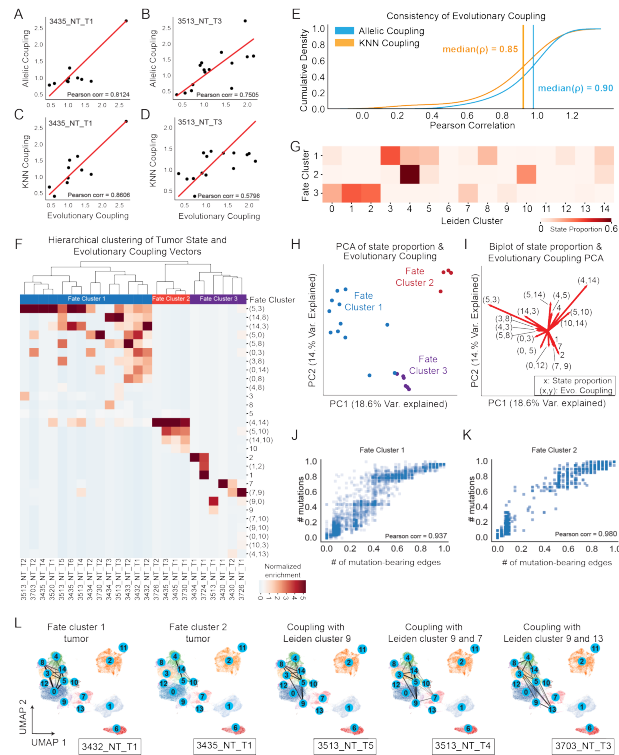


Figure 5.12: Validation of KP-Tracer Evolutionary Coupling and Fate clustering. (A-D) Two alternative statistics measuring couplings between states from lineage tracing data are used to corroborate the Evolutionary Coupling results for the representative tumors 3435_NT_T1 and 3513_NT_T3 shown in Figure 5.5A-D. The comparisons between Allelic Coupling and Evolutionary Coupling for (A) 3435_NT_T1 and (B) 3513_NT_T3 are consistent (Pearson's correlation = 0.94 and 0.99, respectively). The comparisons between KNN Coupling and Evolutionary Coupling for (C) 3435_NT_T1 and (D) 3513_NT_T3 are consistent (Pearson's correlation = 0.97 and 0.86, respectively). Red line indicates the symmetrical $y = x$ relationship. (E) Cumulative density function for Pearson's correlation of Allelic Coupling and KNN Coupling statistics with Evolutionary Couplings for all KP-Tracer tumors. Median correlations are indicated with vertical bars and annotated with the median correlation value. (F) Clustering of tumors based on Evolutionary Coupling and Leiden cluster proportion statistics reveals features that distinguish different Fate Clusters. Three clusters are identified by unbiased clustering, corresponding to Fate Clusters 1, 2, and 3. Fate Cluster is annotated on top of each unique color in the first row of the heatmap. Values/colors in the heatmap are normalized across tumors, and each row corresponds to a feature (either an Evolutionary Coupling or Leiden cluster proportion). Evolutionary couplings are indicated by a tuple of the form (x, y) and Leiden cluster proportions are indicated by a single number of the form x . We focus on showing features that distinguish different clusters, and uninformative features, identified as non-significant by a Mann-Whitney U test ($p > 0.1$), are not shown. (G) Heatmap of state proportions for each Fate Cluster across Leiden clusters. The value of the i^{th} row and j^{th} column indicate the fraction of cells found in the j^{th} Leiden Cluster across all tumors in the i^{th} Fate Cluster. (H) Principal Component Analysis (PCA) of tumor Evolutionary Coupling and Leiden cluster proportion vectors. Each dot is a tumor. Tumors are colored by their Fate Cluster, as identified with the hierarchical clustering shown in **Figure 5.12E**. The percent of variance explained is indicated on each axis. (I) Biplot of PCA of Evolutionary Coupling and Leiden cluster composition vectors, where each arrow indicates the loading of the feature with respect to the first two principal components. The top 10 features for the first two principal components are shown; arrows are annotated with the feature label. The percent of variance explained is indicated on each axis. Features of the form (x, y) represent Evolutionary Couplings between state x and state y ; features of the form x represent the proportion of cells found in Leiden cluster x . (J-K) Comparison of Phylotime statistics computed using weighted and binary tree branch lengths for (J) Fate Cluster 1 and (K) Fate Cluster 2 (Methods). Correlations are strong for both Fate Clusters (Pearson's correlation = 0.94 and correlation = 0.98, respectively). (L) Selected Evolutionary Couplings of individual tumors displayed on gene expression UMAP illustrating connections between transcriptional states (Leiden clusters) of interest. From left: the first plot shows the Evolutionary Couplings within a representative tumor in Fate Cluster 1. The second plot shows the Evolutionary Couplings within a representative tumor in Fate Cluster 2. The third plot shows couplings between Fate Cluster 1 (Leiden clusters 3 and 5) and Late stage transcriptome states (Leiden cluster 9). The fourth plot shows couplings between Fate Cluster 1 (Leiden clusters 3 and 5) and high fitness transcriptome states (Leiden cluster 7 and 9). The last plot shows couplings between Fate Cluster 1 (Leiden clusters 3, 5 and 14) and high fitness transcriptome states (Leiden cluster 9 and 13). These results offer evidence of potential transition from early, low fitness to late, high fitness transcriptome states during tumor evolution.

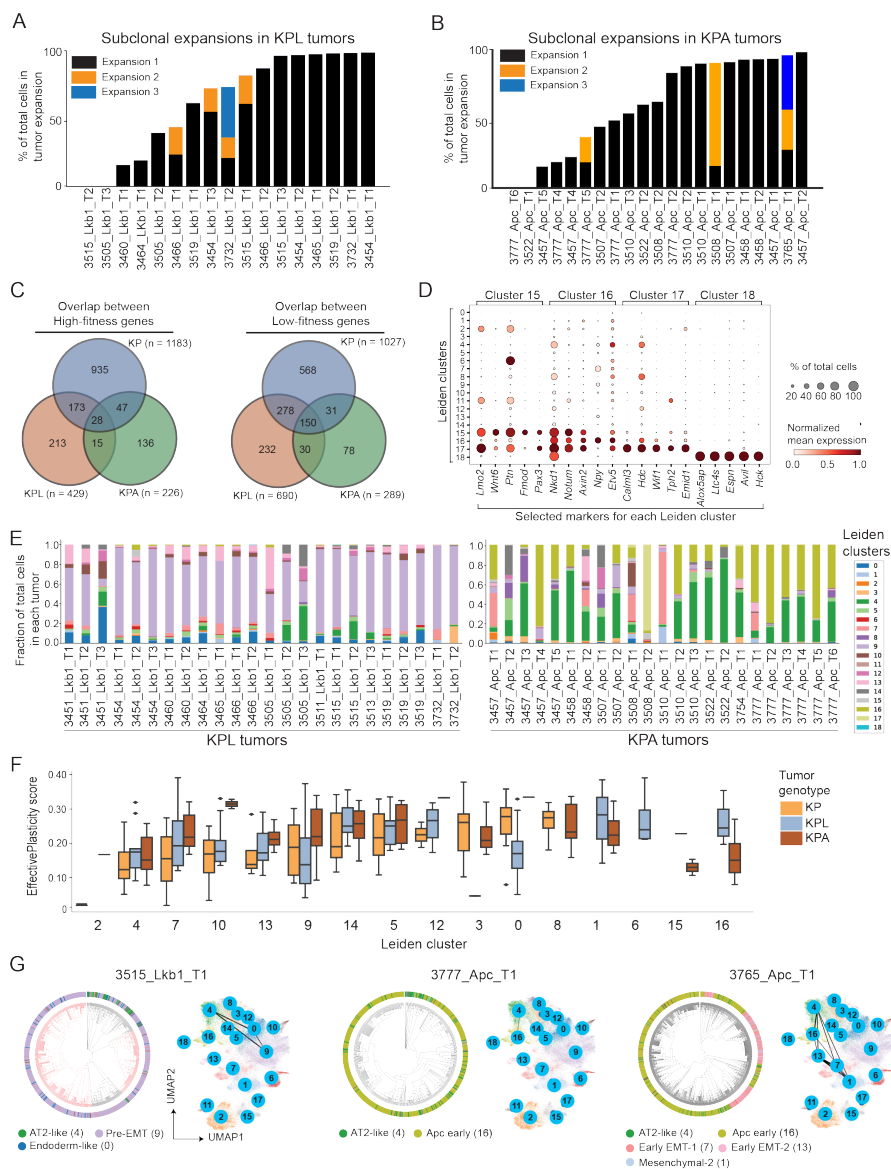


Figure 5.13: Genetic perturbations shift the transcriptional fitness and plasticity landscape of KP-Tracer tumors. (A-B) Subclonal expansion dynamics of (A) KPL and (B) KPA tumors. Independent expansions are colored with black, orange or blue and measured with the percentage of cells in the expanding subclone. (C) Overlap of genes associated with high and low fitness for KP, KPL and KPA tumors. (D) Gene markers for newly identified Leiden clusters in the KP, KPL and KPA integrated analysis. Dots are sized by the fraction of cells expressing a marker and colored by the mean expression of the gene marker in a Leiden cluster. (E) Leiden cluster proportions for each KPL (left) and KPA (right) tumor. (F) Distribution of the mean EffectivePlasticity for each Leiden cluster, averaged within each tumor, compared across genotypes. Leiden clusters 6, 11, 17, 18 are not shown because they lacked enough tumors across genotypes to make comparisons. (G) Evolutionary Couplings of different transcriptional states in three representative tumors reveals evolutionary paths in KPL and KPA tumors. Transcriptional states that are represented by at least 2.5% of cells in each tumor are used. 3515_Lkb1_T1 is a representative KPL tumor. The left plot shows the lineage relationship of transcriptional states in this KPL tumor and the right plot summarizes Evolutionary Couplings on the gene expression UMAP illustrating connections between Leiden clusters 4, 0 and 9. 3777_Apc_T1 is a representative KPA tumor. The left plot shows the lineage relationship of transcriptional states in this KPL tumor and the right plot summarizes Evolutionary Couplings on the gene expression UMAP illustrating connections between Leiden clusters 4 and 16. 3765_Apc_T1 is another representative KPA tumor. The left plot shows the lineage relationship of transcriptional states in this KPL tumor and the right plot summarizes Evolutionary Couplings on the gene expression UMAP illustrating connections between Leiden clusters 4, 16, 13, 7 and 1.

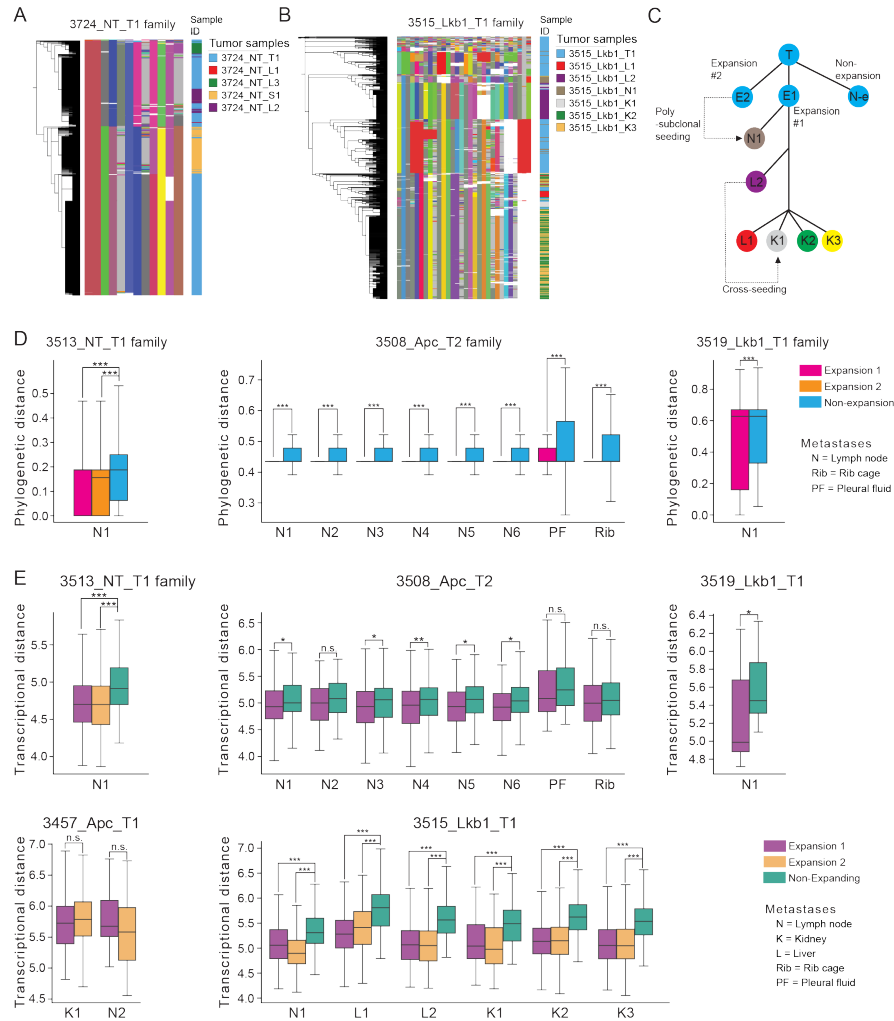


Figure 5.14: Lineage tracing illuminates the metastatic routes and origins in KP-Tracer tumors. (A) Lineage indel heatmap of the 3724_NT_T1 tumor-metastasis family, summarizing the allelic information (indels) from the target sites confirming the separate origin of the soft tissue and liver metastatic tumors. In the Lineage indel heatmap, each row represents a single cell and each column represents a cut site of the lineage tracer. Unique indels are shown in unique colors, uncut target sites are indicated in gray, and missing data is indicated in white. The reconstructed lineage based on the accumulated indel patterns using Cassiopeia are shown on the left. The corresponding sample ID for each cell is labeled on the right. (B-C) Subclonal origin and the metastatic routes for 3515_Lkb1_T1 tumor-metastasis family. (B) Lineage indel heatmap of 3515_Lkb1_T1 tumor-metastasis family, indicating indel alleles supporting the subclonal origins, the relative order and the routes of metastases and (C) a model summarizing these metastatic behaviors. (D) More supporting examples of expanding subclones giving rise to metastases across genotypes for 3513_NT_T1 (left), 3508_Apc_T2 (center), and 3519_Lkb1_T1 (right). (E) Comparison of transcriptional distance between metastatic tumors and cells in non-expanding and expanding regions of the primary tumor phylogeny for 3513_NT_T1, 3508_Apc_T2, 3519_Lkb1_T1, 3457_Apc_T1, and 3515_Lkb1_T1 metastasis families. All significances are indicated from a one-sided Mann-Whitney U test: *** indicates $p < 0.001$, ** indicates $p < 0.01$, and * indicates $p < 0.05$.

Part III

Conclusions

Chapter 6

Conclusion

I have purposely broken up this thesis into two parts, roughly mirroring my path through graduate school: first, computational methods and technology development and second, applications to cancer biology. In these concluding remarks, I offer some perspectives and future directions building on my thesis and how these two themes - computation and mouse modeling - might continue to complement one another.

Developing the next generation of lineage tracers

Already, tremendous progress has been made in single-cell lineage tracing. From the beginnings of optically tracking progenitor fields and individual cell divisions [284], the field has generated a spectrum of approaches for labeling several progenitors and multiplexing these tracing approaches with cutting-edge single-cell assays [269]. As illustrated in the previous chapters, our group has had success in applying CRISPR/Cas9 lineage tracers to mouse models of lung cancer.

Still, there are several challenges intrinsic to these technologies that motivate future work. First and foremost, some of the current technologies - especially CRISPR/Cas9-based technologies - induce high levels of cellular stress by virtue of their incessant double-strand breakage. While a good number of processes are robust to this [37], this stress has precluded the adoption of these technologies for tracing in more sensitive populations like stem cells. Fortunately, there are several alternatives to this variant of Cas9 editing [9]. Specifically, there exist alternative Cas9-based approaches that do not rely on double-strand breaks - such as base editors [149, 127] or prime editors [10] - that could be used to improve to cytotoxicity of labeling.

Second, we've found through our simulations that the rate of homoplasy (i.e., the number of mutations that are induced more than once, independently) greatly affects the complexity of tree inference. This is specifically the case for the Cassiopeia-Greedy algorithm that performs substantially better when homoplasy is minimal [127] (see Chapter 2). Motivated by these findings, we believe that the next generation of lineage tracers should focus on controlling this homoplasy rate. While one way to achieve this is by increasing the number of state outcomes (see Chapter 3), another way is to improve the entropy of the state distribution. As it is, while there are thousands of different indels that are possible, the Cas9-based indel distribution is dominated by simple insertions and deletions [127]. Might new technologies be able to guide this state distribution to a more evenly-balanced distribution and thereby control homoplasy? One such avenue for this would be the use of prime-editing, which necessarily controls the possible states and can be modified such that the state outcomes are well balanced (for example with computational design [143] or by studying the genetic determinants of prime editing efficiency [39]).

Third, missing data often plagues inference of the single-cell lineages. Based on the observations of our work and that of others, we have hypothesized that missing data comes from two sources: stochastic dropout from the single-cell assay and heritable removal of the target cassette. While stochastic dropout, will likely remain consistent in the near future due to the sensitivity of commercial scRNA-seq platforms, we believe there are ways of improving on the heritable dropout problem with new algorithms or lineage tracing designs. Computationally, we believe that improvement can be achieved by first delineating between stochastic and heritable dropout and then imputing the most likely identity of missing data using unsupervised learning algorithms. In certain aspects, this

problem set up is similar to that of “recommender algorithms” that attempt to predict behaviors like shopping preferences from incomplete observations. Technologically, advances can also be made by reducing the chances of resection events that remove target sites. Naturally, this is largely a problem with technologies that leverage Cas9-induced double-strand breaks for lineage tracing and can be avoided with alternative approaches like the base editing and prime editing systems discussed above.

In all of these technological improvements, we offer our Cassiopeia2.0 codebase for efficiently experimenting and prioritizing new lineage tracing design regimes (see Chapter 3). Because developing new technologies is a costly and time-consuming process, screening promising engineering regimes with cheap and realistic simulations is a desirable feature of the improved codebase.

Probing the spatial determinants of tumor progression

As we’ve discussed in Chapter 5, the value of studying the progressive genetic and epigenetic changes underlying tumor progression cannot be understated as they have led to key insights into how tumors progress and how they might be targeted by specific therapies [188, 87, 13]. More recently, the impact of the spatial context of tumors has become increasingly appreciated in its role in facilitating or suppressing cancer growth [20]. Indeed, it is a critical question how tumors evolve in the presence of selective pressure from the microenvironment [265] - either from immune cell or other stromal subsets - and is a question that lineage tracing is well poised to address.

An immediate direction that can be pursued is studying how “normal” cell subsets and tumor

cells co-evolve in the tumor niche. In Chapter 5, we introduce a mouse model capable of tracing lung cancer tumor progression in the native lung microenvironment and provide statistical measures for quantifying the selection dynamics and evolutionary trajectories. With this KP-Tracer model, one might ask if certain immune cell subsets are associated with certain fitness-associated transcriptional programs or with specific evolutionary trajectories. We anticipate that applying statistical models from studying co-evolution in the field of ecology [180, 294] will be helpful here.

Beyond measuring co-evolution from associations in abundance, future investigations might probe how specific cell-cell interactions are impacting tumor progression. While some studies have been able to infer cell-cell interactions either computationally [64] or from FACS-related artifacts [90], a more promising direction for this task would be to leverage the burgeoning field of spatial-transcriptomics technologies [239, 212, 242]. With these approaches, one could associate each cell not only with its phylogenetic properties (e.g., fitness) and transcriptomic profile, but also with its neighborhood of interactions. This could provide answers to several questions of interest - for example, are interactions with specific cell-subsets associated with increases in fitness? Do metastases arise from specific immune cell niches?

Naturally, these technological improvements would necessitate simultaneous computational advances. For example, methods will have to be developed to handle the non-single-cell-resolution of these technologies. Once again, we offer Cassiopeia2.0 as an extensible and flexible software package for implementing such processing pipelines as well as simulating the effects of varying resolution on lineage tracing performance.

Predicting evolution from lineages

From the outset, I have looked towards how one might be able to develop models for predicting how a cell came to its current state and how its lineage will evolve in the future. There exist sophisticated models for inferring these properties from transcriptomic data alone - most notably, pseudotime methods like Monocle [261] and velocity methods [150] - but these contain several limitations around the assumptions they make about the data [262]. We believe that harnessing phylogenies would provide a more principled approach for predicting the evolution of the populations.

Above, I am using the term “evolution” loosely - but of course, this can refer to several different properties. For example, we have leveraged previous work that proposes a model for predicting the future *fitness* of extant samples [186] (see Chapter 5). While this previous work showed that this property could be accurately predicted in certain regimes, and we were able to successfully apply it in our data, it is still unclear to what degree evolution is predictable [156]. This is because true evolution relies on random variability and thus to answer this question one tends to have to speak in generalizations.

Still, certain systems are more amenable to predicting the fate of extant populations. For example, in reproducible systems like development, evolution is far less stochastic as one can appreciate that there are a finite set of fates for each progenitor cell type. Thus, the problem becomes more constrained - given the cell type of some observed cell, what is the likelihood of it becoming another cell type? Additionally, when there is strong selection pressure towards a certain phenotype, evolution once again becomes constrained - in this context, evolution is most likely to obey the *fitness land-*

scape imposed by selection pressures [156]. This observation prompts other questions - namely, if selection pressure can shape evolution, to what degree can evolution be guided artificially?

It is our belief that understanding a cell's lineage history is just as important as where it might go. In developmental contexts, for example, researchers have long been interested in inferring whether or not there are multiple lineages that can converge on a specific physiological function. Computationally, we have developed deep-learning methods that can predict the ancestral, or unobserved, cellular states from the observed leaves of a phylogeny [191]. However, this is a notoriously difficult problem because of the intrinsic uncertainty of what unobserved intermediate states looked like (see Chapter 5). Technologically, we believe that this problem can be improved by engineering "molecular recorders" that are capable of inducing marks if a specific gene is expressed (for example, if *Wnt* is expressed, as in [79]). In this way, one can deduce from final observations if particular cells developed through lineages defined by a gene set of interest. Together, it is likely that knowing where a cell came from will only aid in the prediction of where it will go.

While predicting the fates of single cells is of general interest, we are particularly motivated with how this applies to cancer. Already, others are working on this idea - for example, the concept of using selective pressures to guide evolution is an exciting direction under investigation in cancer treatment as groups have attempted to "evolutionarily steer" cancer populations towards drug sensitivity [2]. There are still several questions left to be answered around the reproducibility of tumor evolution and the identification of features that allow one to accurately predict its future. In all of this, we believe that phylogenetic models and the lineage tracing technologies described in this thesis will prove to be irreplaceable in these studies going forward.

Extending lineage tracing beyond model systems

A key caveat to all the cancer biology work described above is that it has been performed in model systems such as cell lines (Chapter 4) and mouse models (Chapter 5). While these systems have the benefit of being tractable and are known to recapitulate several aspects of human disease, there has historically been a major bottleneck in applying lessons learned from model systems to human patients. We believe that this is motivation to extend the computational methodology and ideas expressed in this thesis to human samples to the best of our abilities.

While it is currently infeasible to be tracing tumors using the synthetic lineage tracing systems described in this thesis, there are several opportunities to perform single-cell lineage tracing. For example, as mentioned in Chapter 4 and 5, copy-number variations (CNVs) can be used to trace relationships between samples. Beyond this, another area of exciting method development is that of mitochondrial sequencing (mito-seq [166]). With a relatively small genome of $\sim 16\text{kb}$, the mitochondria is a compact source of variation that can be observed using standard single-cell assays. From this variation, phylogenetic inference can be performed. However, as of date, the resolution of these mito-seq approaches is limited and motivates other work for resolving lineages. Towards this, other heritable endogenous markers can be exploited - for example, short tandem repeats [255], methylation marks [80], or extrachromosomal DNA [154]. Beyond using individual modalities, we believe that integrating each of these together - perhaps accessible through long-read sequencing - would greatly improve our ability to infer lineages in human cases. Of course, as above, we offer Cassiopeia as the development environment for such processing pipelines and algorithmic development.

Concluding Remarks

Working on interdisciplinary teams, I have been left with an appreciation for how science will be done in the 21st century, where the lines discriminating between computation and biology will fade away and larger teams of complementary skillsets will become the norm. And so, the past five years have been a rich experience for me as I've had the extraordinary opportunity to work on both cutting-edge algorithms and technologies. I am sure that developing the ability to interface between biologists, technologists, and computer scientists will serve me well and for that I am eternally indebted to those who had the patience to teach me these trades in the beginning.

As this conclusion has discussed, there is still much work to be done. As Sydney Brenner once said, "Progress in science depends on new techniques, new discoveries and new ideas, probably in that order". I believe that in this thesis, I have offered new techniques and discoveries, and perhaps a few new ideas of interest about how tumors progress. I am hopeful that as we continue to develop the technologies proposed here, discoveries and ideas will follow. At the very least, I look forward to building more bridges and helping to fill in the missing pieces of how tumor evolution works and can be predicted.

References

- [1] Christopher Abbosh et al. “Phylogenetic ctDNA analysis depicts early-stage lung cancer evolution”. In: *Nature* 545.7655 (2017), pp. 446–451.
- [2] Ahmet Acar et al. “Exploiting evolutionary steering to induce collateral drug sensitivity in cancer”. In: *Nature communications* 11.1 (2020), pp. 1–14.
- [3] Britt Adamson et al. “A Multiplexed Single-Cell CRISPR Screening Platform Enables Systematic Dissection of the Unfolded Protein Response”. In: *Cell* 167.7 (2016), 1867–1882.e21. ISSN: 0092-8674. DOI: <https://doi.org/10.1016/j.cell.2016.11.048>.
- [4] Anna Alemany et al. “Whole-organism clone tracing using single-cell sequencing”. In: *Nature* 556.7699 (2018), pp. 108–112.
- [5] Felicity Allen et al. “Predicting the mutations generated by repair of Cas9-induced double-strand breaks”. In: *Nature Biotechnology* 37 (2018), 64 EP.
- [6] Nabil Amirouchene-Angelozzi, Charles Swanton, and Alberto Bardelli. “Tumor evolution as a therapeutic target”. In: *Cancer discovery* 7.8 (2017), pp. 805–817.

- [7] Dimitris Anastassiou et al. "Human cancer cells express Slug-based epithelial-mesenchymal transition gene expression signature obtained in vivo". In: *BMC cancer* 11.1 (2011), pp. 1–9.
- [8] Mihaela Angelova et al. "Evolution of metastases in space and time under immune selection". In: *Cell* 175.3 (2018), pp. 751–765.
- [9] Andrew V Anzalone, Luke W Koblan, and David R Liu. "Genome editing with CRISPR–Cas nucleases, base editors, transposases and prime editors". In: *Nature biotechnology* 38.7 (2020), pp. 824–844.
- [10] Andrew V Anzalone et al. "Search-and-replace genome editing without double-strand breaks or donor DNA". In: *Nature* 576.7785 (2019), pp. 149–157.
- [11] Anna Arnal-Estapé et al. "Tumor progression and chromatin landscape of lung cancer are regulated by the lineage factor GATA6". In: *Oncogene* 39.18 (2020), pp. 3726–3737.
- [12] Amjad Askary et al. "In situ readout of DNA barcodes and single base edits facilitated by in vitro transcription". In: *Nature biotechnology* 38.1 (2020), pp. 66–75.
- [13] Chris Bailey et al. "Tracking cancer evolution through the disease course". In: *Cancer discovery* 11.4 (2021), pp. 916–932.
- [14] Nick Barker et al. "Crypt stem cells as the cells-of-origin of intestinal cancer". In: *Nature* 457.7229 (2009), pp. 608–611.

- [15] Chloé S Baron and Alexander van Oudenaarden. “Unravelling cellular relationships during development and regeneration using genetic lineage tracing”. In: *Nature reviews molecular cell biology* 20.12 (2019), pp. 753–765.
- [16] Ittai Ben-Porath et al. “An embryonic stem cell–like gene expression signature in poorly differentiated aggressive human tumors”. In: *Nature genetics* 40.5 (2008), pp. 499–507.
- [17] Yoav Benjamini and Yosef Hochberg. “Controlling the false discovery rate: a practical and powerful approach to multiple testing”. In: *Journal of the Royal statistical society: series B (Methodological)* 57.1 (1995), pp. 289–300.
- [18] Wicher Bergsma. “A bias-correction for Cramér’s V and Tschuprow’s T”. In: *Journal of the Korean Statistical Society* 42.3 (2013), pp. 323–328.
- [19] HC Bhang. “Ruddy D a, Krishnamurthy Radhakrishna V, Caushi JX, Zhao R, Hims MM, et al. Studying clonal dynamics in response to cancer therapy using high-complexity barcoding”. In: *Nat Med* 21.5 (2015).
- [20] Mikhail Binnewies et al. “Understanding the tumor immune microenvironment (TIME) for effective therapy”. In: *Nature medicine* 24.5 (2018), pp. 541–550.
- [21] Nicolai J Birnbak and Nicholas McGranahan. “Cancer genome evolutionary trajectories in metastasis”. In: *Cancer cell* 37.1 (2020), pp. 8–19.
- [22] James RM Black and Nicholas McGranahan. “Genetic and non-genetic clonal diversity in cancer evolution”. In: *Nature Reviews Cancer* 21.6 (2021), pp. 379–392.

- [23] Hans L. Bodlaender and Tandy J. Fellows Mike R. an Warnow. “Two strikes against perfect phylogeny”. In: *Automata, Languages and Programming*. Ed. by W. Kuich. Berlin, Heidelberg: Springer Berlin Heidelberg, 1992, pp. 273–283. ISBN: 978-3-540-47278-0.
- [24] Sarah Bowling et al. “An engineered CRISPR-Cas9 mouse line for simultaneous readout of lineage histories and gene expression profiles in single cells”. In: *Cell* 181.6 (2020), pp. 1410–1422.
- [25] Evan A Boyle et al. “High-throughput biochemical profiling reveals sequence determinants of dCas9 off-target binding and unbinding”. In: *Proceedings of the National Academy of Sciences* 114.21 (2017), pp. 5461–5466.
- [26] A. Rose Brannon et al. “Comparative sequencing analysis reveals high genomic concordance between matched primary and metastatic colorectal cancer lesions”. In: *Genome Biology* 15.8 (2014), p. 454. ISSN: 1474-760X. DOI: [10.1186/s13059-014-0454-7](https://doi.org/10.1186/s13059-014-0454-7).
- [27] Markus Bredel et al. “NFKBIA deletion in glioblastomas”. In: *New England Journal of Medicine* 364.7 (2011), pp. 627–637.
- [28] Jason D Buenrostro et al. “Single-cell chromatin accessibility reveals principles of regulatory variation”. In: *Nature* 523.7561 (2015), pp. 486–490.
- [29] Joseph H. Camin and Robert R. Sokal. “A Method for Deducing Branching Sequences in Phylogeny”. In: *Evolution* 19.3 (1965), pp. 311–326. ISSN: 00143820, 15585646.
- [30] Brittany B Campbell et al. “Comprehensive analysis of hypermutation in human cancer”. In: *Cell* 171.5 (2017), pp. 1042–1056.

- [31] Julian Carretero et al. “Integrative genomic and proteomic analyses identify targets for Lkb1-deficient metastatic lung tumors”. In: *Cancer cell* 17.6 (2010), pp. 547–559.
- [32] Deborah R Caswell et al. “Obligate progression precedes lung adenocarcinoma dissemination”. In: *Cancer discovery* 4.7 (2014), pp. 781–789.
- [33] L. L. Cavalli-Sforza and A. W. F. Edwards. “Phylogenetic Analysis: Models and Estimation Procedures”. In: *Evolution* 21.3 (1967), pp. 550–570.
- [34] Toni Celià-Terrassa and Yibin Kang. “Distinctive properties of metastasis-initiating cells”. In: *Genes & development* 30.8 (2016), pp. 892–908.
- [35] Christine L Chaffer and Robert A Weinberg. “A perspective on cancer cell metastasis”. In: *science* 331.6024 (2011), pp. 1559–1564.
- [36] Christine L Chaffer et al. “Poised chromatin at the ZEB1 promoter enables breast cancer cell plasticity and enhances tumorigenicity”. In: *cell* 154.1 (2013), pp. 61–74.
- [37] Michelle M. Chan et al. “Molecular recording of mammalian embryogenesis”. In: *Nature* 570.7759 (2019), pp. 77–82. ISSN: 1476-4687. DOI: [10.1038/s41586-019-1184-5](https://doi.org/10.1038/s41586-019-1184-5).
- [38] FF Chen et al. “Effects of upregulation of Id3 in human lung adenocarcinoma cells on proliferation, apoptosis, mobility and tumorigenicity”. In: *Cancer Gene Therapy* 22.9 (2015), pp. 431–437.
- [39] Peter J Chen et al. “Enhanced prime editing systems by manipulating cellular determinants of editing outcomes”. In: *Cell* 184.22 (2021), pp. 5635–5652.

- [40] Wei Chen et al. “Massively parallel profiling and predictive modeling of the outcomes of CRISPR/Cas9-mediated double-strand break repair”. In: *bioRxiv* (2018). DOI: [10.1101/481069](https://doi.org/10.1101/481069).
- [41] William KC Cheung et al. “Control of alveolar differentiation by the lineage transcription factors GATA6 and HOPX inhibits lung adenocarcinoma metastasis”. In: *Cancer cell* 23.6 (2013), pp. 725–738.
- [42] Benny Chor and Tamir Tuller. “Maximum likelihood of evolutionary trees: hardness and approximation”. In: *Bioinformatics* 21 Suppl 1 (2005), pp. i97–106.
- [43] Ke-Huan K Chow et al. “Imaging cell lineage with a synthetic digital recording system”. In: *Science* 372.6538 (2021), eabb3099.
- [44] Chen-Hua Chuang et al. “Molecular definition of a metastatic lung cancer state reveals a targetable CD109–Janus kinase–Stat axis”. In: *Nature medicine* 23.3 (2017), pp. 291–300.
- [45] Francesca D. Ciccarelli et al. “Toward Automatic Reconstruction of a Highly Resolved Tree of Life”. In: *Science* 311.5765 (2006), pp. 1283–1287. ISSN: 0036-8075. DOI: [10.1126/science.1123061](https://doi.org/10.1126/science.1123061).
- [46] Jonathan D.W. Clarke and Cheryll Tickle. “Fate maps old and new”. In: *Nature Cell Biology* 1.4 (Aug. 1999), E103–E109. DOI: [10.1038/12105](https://doi.org/10.1038/12105).
- [47] Eric A. Collisson et al. “Comprehensive molecular profiling of lung adenocarcinoma”. In: *Nature* 511.7511 (2014), pp. 543–550. ISSN: 1476-4687. DOI: [10.1038/nature13385](https://doi.org/10.1038/nature13385).

- [48] Elizabeth Comen and Larry Norton. "Self-seeding in cancer". In: *Minimal Residual Disease and Circulating Tumor Cells in Breast Cancer* (2012), pp. 13–23.
- [49] The Tabula Muris Consortium and colleagues. "Single-cell transcriptomics of 20 mouse organs creates a Tabula Muris". In: *Nature* 562.7727 (Oct. 2018), pp. 367–372. doi: [10.1038/s41586-018-0590-4](https://doi.org/10.1038/s41586-018-0590-4).
- [50] The Tabula Sapiens Consortium and Stephen R Quake. "The Tabula Sapiens: a multiple organ single cell transcriptomic atlas of humans". In: *bioRxiv* (2021). doi: [10.1101/2021.07.19.452956](https://doi.org/10.1101/2021.07.19.452956).
- [51] Forrest W Crawford, Lam Si Tung Ho, and Marc A Suchard. "Computational methods for birth-death processes". In: *Wiley Interdisciplinary Reviews: Computational Statistics* 10.2 (2018), e1423.
- [52] Douglas E. Critchlow, Dennis K. Pearl, and Chunlin Qian. "The Triples Distance for Rooted Bifurcating Phylogenetic Trees". In: *Systematic Biology* 45.3 (1996), pp. 323–334. ISSN: 10635157, 1076836X.
- [53] Alexander Davis, Ruli Gao, and Nicholas Navin. "Tumor evolution: Linear, branching, neutral or punctuated?" In: *Biochimica et Biophysica Acta (BBA) - Reviews on Cancer* 1867.2 (2017), pp. 151–161. ISSN: 0304-419X. doi: <https://doi.org/10.1016/j.bbcan.2017.01.003>.
- [54] Sarah K Denny et al. "Nfib promotes metastasis through a widespread increase in chromatin accessibility". In: *Cell* 166.2 (2016), pp. 328–342.

- [55] U Deppe et al. “Cell lineages of the embryo of the nematode *Caenorhabditis elegans*”. In: *Proceedings of the National Academy of Sciences* 75.1 (1978), pp. 376–380. ISSN: 0027-8424. DOI: [10.1073/pnas.75.1.376](https://doi.org/10.1073/pnas.75.1.376).
- [56] Amit G Deshwar et al. “PhyloWGS: reconstructing subclonal composition and evolution from whole-genome sequencing of tumors”. In: *Genome biology* 16.1 (2015), p. 35.
- [57] David DeTomaso and Nir Yosef. “Hotspot identifies informative gene modules across modalities of single-cell genomics”. In: *Cell Systems* 12.5 (2021), 446–456.e9. ISSN: 2405-4712. DOI: <https://doi.org/10.1016/j.cels.2021.04.005>.
- [58] David DeTomaso et al. “Functional interpretation of single cell similarity maps”. In: *Nature communications* 10.1 (2019), pp. 1–11.
- [59] Li Ding et al. “Somatic mutations affect key pathways in lung adenocarcinoma”. In: *Nature* 455.7216 (2008), pp. 1069–1075.
- [60] Gregory Driessens et al. “Defining the mode of tumour growth by clonal analysis”. In: *Nature* 488.7412 (2012), pp. 527–530.
- [61] Michel DuPage, Alison L Dooley, and Tyler Jacks. “Conditional mouse lung cancer models using adenoviral or lentiviral delivery of Cre recombinase”. In: *Nature protocols* 4.7 (2009), pp. 1064–1072.
- [62] Michel DuPage and Tyler Jacks. “Genetically engineered mouse models of cancer reveal new insights about the antitumor immune response”. In: *Current opinion in immunology* 25.2 (2013), pp. 192–199.

- [63] Hariharan Easwaran, Hsing-Chen Tsai, and Stephen B Baylin. “Cancer epigenetics: tumor heterogeneity, plasticity of stem-like states, and drug resistance”. In: *Molecular cell* 54.5 (2014), pp. 716–727.
- [64] Mirjana Efremova et al. “CellPhoneDB: inferring cell–cell communication from combined expression of multi-subunit ligand–receptor complexes”. In: *Nature protocols* 15.4 (2020), pp. 1484–1506.
- [65] Uri Einav et al. “Gene expression analysis reveals a strong signature of an interferon-induced pathway in childhood lymphoblastic leukemia as well as in breast and ovarian cancer”. In: *Oncogene* 24.42 (2005), pp. 6367–6375.
- [66] Weixiang Fang et al. “Quantitative fate mapping: Reconstructing progenitor field dynamics via retrospective lineage barcoding”. In: *bioRxiv* (2022). doi: [10.1101/2022.02.13.480215](https://doi.org/10.1101/2022.02.13.480215).
- [67] James S. Farris. “Methods for Computing Wagner Trees”. In: *Systematic Zoology* 19.1 (1970).
- [68] J Felsenstein. “PHYLIP (Phylogeny Inference Package)”. In: *Distributed by the author. Department of Genome Sciences, University of Washington, Seattle* ().
- [69] Joseph Felsenstein. “Evolutionary trees from DNA sequences: A maximum likelihood approach”. In: *Journal of Molecular Evolution* 17.6 (1981), pp. 368–376. ISSN: 1432-1432. doi: [10.1007/BF01734359](https://doi.org/10.1007/BF01734359).
- [70] Jean Feng et al. “Estimation of cell lineage trees by maximum-likelihood phylogenetics”. In: *bioRxiv* (2019). doi: [10.1101/595215](https://doi.org/10.1101/595215).

- [71] Isaiah J Fidler and Margaret L Kripke. “The challenge of targeting metastasis”. In: *Cancer and Metastasis Reviews* 34.4 (2015), pp. 635–641.
- [72] Walter Fitch. “Toward Defining the Course of Evolution: Minimum Change for a Specific Tree Topology”. In: *Systematic Zoology* 20.4 (1971).
- [73] Walter M. Fitch and Emanuel Margoliash. “Construction of Phylogenetic Trees”. In: *Science* 155.3760 (1967), pp. 279–284. ISSN: 0036-8075. DOI: [10.1126/science.155.3760.279](https://doi.org/10.1126/science.155.3760.279).
- [74] Dustin J Flanagan et al. “NOTUM from Apc-mutant cells biases clonal competition to initiate cancer”. In: *Nature* 594.7863 (2021), pp. 430–435.
- [75] William A Flavahan, Elizabeth Gaskell, and Bradley E Bernstein. “Epigenetic plasticity and the hallmarks of cancer”. In: *Science* 357.6348 (2017), eaal2380.
- [76] Les R Foulds and Ronald L Graham. “The Steiner problem in phylogeny is NP-complete”. In: *Advances in Applied mathematics* 3.1 (1982), pp. 43–49.
- [77] Giulio Francia et al. “Mouse models of advanced spontaneous metastasis for experimental therapeutics”. In: *Nature Reviews Cancer* 11.2 (2011), pp. 135–141.
- [78] Kristopher K Frese and David A Tuveson. “Maximizing mouse cancer models”. In: *Nature Reviews Cancer* 7.9 (2007), pp. 654–658.
- [79] Kirsten L Frieda et al. “Synthetic recording and in situ readout of lineage information in single cells”. In: *Nature* 541.7635 (2017), pp. 107–111.

- [80] Calum Gabbutt et al. “Fluctuating methylation clocks for cell lineage tracing at high temporal resolution in human tissues”. In: *Nature biotechnology* (2022), pp. 1–11.
- [81] Karuna Ganesh and Joan Massagué. “Targeting metastatic cancer”. In: *Nature medicine* 27.1 (2021), pp. 34–44.
- [82] Ruli Gao et al. “Delineating copy number and clonal substructure in human tumors from single-cell transcriptomes”. In: *Nature biotechnology* 39.5 (2021), pp. 599–608.
- [83] O Gascuel and M Steel. “Neighbor-Joining Revealed”. In: *Molecular Biology and Evolution* 23.11 (2006), pp. 1997–2000. DOI: [10.1093/molbev/msl072](https://doi.org/10.1093/molbev/msl072).
- [84] Nicole M. Gaudelli et al. “Programmable base editing of A*T to G*C in genomic DNA without DNA cleavage”. In: *Nature* 551 (2017), 464 EP.
- [85] Yejing Ge et al. “Stem cell lineage infidelity drives wound repair and cancer”. In: *Cell* 169.4 (2017), pp. 636–650.
- [86] Jason M. Gehrke et al. “An APOBEC3A-Cas9 base editor with minimized bystander and off-target activities”. In: *Nature Biotechnology* 36 (2018), 977 EP.
- [87] Marco Gerlinger et al. “Cancer: evolution within a lifetime”. In: *Annual review of genetics* 48 (2014), pp. 215–236.
- [88] Marco Gerlinger et al. “Intratumor heterogeneity and branched evolution revealed by multi-region sequencing”. In: *N Engl J Med* 366 (2012), pp. 883–892.

- [89] Moritz Gerstung et al. “The evolutionary history of 2,658 cancers”. In: *Nature* 578.7793 (2020), pp. 122–128.
- [90] Amir Giladi et al. “Dissecting cellular crosstalk by sequencing physically interacting cells”. In: *Nature Biotechnology* 38.5 (2020), pp. 629–637.
- [91] Luke A. Gilbert et al. “Genome-Scale CRISPR-Mediated Control of Gene Repression and Activation”. In: *Cell* 159.3 (2014), pp. 647–661. ISSN: 0092-8674. DOI: <https://doi.org/10.1016/j.cell.2014.09.029>.
- [92] Wuming Gong et al. “Benchmarked approaches for reconstruction of in vitro cell lineages and in silico models of *C. elegans* and *M. musculus* developmental trees”. In: *Cell Systems* 12.8 (Aug. 2021), 810–826.e4. DOI: [10.1016/j.cels.2021.05.008](https://doi.org/10.1016/j.cels.2021.05.008).
- [93] Hugo Gonzalez, Catharina Hagerling, and Zena Werb. “Roles of the immune system in cancer: from tumor initiation to metastatic progression”. In: *Genes & development* 32.19-20 (2018), pp. 1267–1284.
- [94] R.C. Griffiths and Simon Tavaré. “The age of a mutation in a general coalescent tree”. In: *Communications in Statistics. Stochastic Models* 14.1-2 (1998), pp. 273–295. DOI: [10.1080/15326349808807471](https://doi.org/10.1080/15326349808807471).
- [95] M Grotschel, A Martin, and R Weismann. “The Steiner Tree packing problem in VLSI design”. In: *Mathematical Programming* 78 (1997), pp. 265–281.
- [96] Ying Guo et al. “RegIV potentiates colorectal carcinoma cell migration and invasion via its CRD domain”. In: *Cancer genetics and cytogenetics* 199.1 (2010), pp. 38–44.

- [97] LLC Gurobi Optimization. *Gurobi Optimizer Reference Manual*. 2018. URL: <http://www.gurobi.com>.
- [98] Dan Gusfield. "Efficient algorithms for inferring evolutionary trees". In: *Networks* 21.1 (1991), pp. 19–28. DOI: [10.1002/net.3230210104](https://doi.org/10.1002/net.3230210104).
- [99] Dan Gusfield. "The Multi-State Perfect Phylogeny Problem with Missing and Removable Data: Solutions via Integer-Programming and Chordal Graph Theory". In: *Journal of Computational Biology* 17.3 (2010), pp. 383–399. DOI: [10.1089/cmb.2009.0200](https://doi.org/10.1089/cmb.2009.0200).
- [100] Xiaoping Han et al. "Mapping the mouse cell atlas by microwell-seq". In: *Cell* 172.5 (2018), pp. 1091–1107.
- [101] Douglas Hanahan and Robert A Weinberg. "Hallmarks of cancer: the next generation". In: *cell* 144.5 (2011), pp. 646–674.
- [102] Byron Hann and Allan Balmain. "Building 'validated' mouse models of human cancer". In: *Current Opinion in Cell Biology* 13.6 (2001), pp. 778–784.
- [103] Yun-He Hao et al. "TNNT1, a prognostic indicator in colon adenocarcinoma, regulates cell behaviors and mediates EMT process". In: *Bioscience, Biotechnology, and Biochemistry* 84.1 (2020), pp. 111–117.
- [104] John A Hartigan. "Minimum mutation fits to a given tree". In: *Biometrics* (1973), pp. 53–65.
- [105] Aaron N Hata et al. "Tumor cells can follow distinct evolutionary paths to become resistant to epidermal growth factor receptor inhibition". In: *Nature medicine* 22.3 (2016), pp. 262–269.

- [106] Weiling He et al. "CTHRC1 induces non-small cell lung cancer (NSCLC) invasion through upregulating MMP-7/MMP-9". In: *BMC cancer* 18.1 (2018), pp. 1–14.
- [107] Gaelen T. Hess et al. "Directed evolution using dCas9-targeted somatic hypermutation in mammalian cells". In: *Nature Methods* 13 (2016), 1036 EP.
- [108] Alexander Heyde et al. "Consecutive seeding and transfer of genetic diversity in metastasis". In: *Proceedings of the National Academy of Sciences* 116.28 (2019), pp. 14129–14137.
- [109] Manuel Hidalgo et al. "Patient-derived xenograft models: an emerging platform for translational cancer research". In: *Cancer discovery* 4.9 (2014), pp. 998–1013.
- [110] William Hill, Deborah R Caswell, and Charles Swanton. "Capturing cancer evolution using genetically engineered mouse models (GEMMs)". In: *Trends in cell biology* 31.12 (2021), pp. 1007–1018.
- [111] RP Hobbs et al. "Loss of Keratin 17 induces tissue-specific cytokine polarization and cellular differentiation in HPV16-driven cervical tumorigenesis in vivo". In: *Oncogene* 35.43 (2016), pp. 5653–5662.
- [112] Pablo E Hollstein et al. "The AMPK-related kinases SIK1 and SIK3 mediate key tumor-suppressive effects of LKB1 in NSCLC". In: *Cancer discovery* 9.11 (2019), pp. 1606–1627.
- [113] Woo Suk Hong, Max Shpak, and Jeffrey P Townsend. "Inferring the origin of metastases from cancer phylogenies". In: *Cancer research* 75.19 (2015), pp. 4021–4025.

- [114] Max A Horlbeck et al. “Compact and highly active next-generation libraries for CRISPR-mediated gene repression and activation”. In: *elife* 5 (2016), e19760.
- [115] Zheng Hu and Christina Curtis. “Looking backward in time to define the chronology of metastasis”. In: *Nature Communications* 11.1 (2020), pp. 1–4.
- [116] Zheng Hu et al. “Quantitative evidence for early metastatic seeding in colorectal cancer”. In: *Nature genetics* 51.7 (2019), pp. 1113–1122.
- [117] John P. Huelsenbeck and Fredrik Ronquist. “MRBAYES: Bayesian inference of phylogenetic trees”. In: *Bioinformatics* 17.8 (2001), pp. 754–755. DOI: [10.1093/bioinformatics/17.8.754](https://doi.org/10.1093/bioinformatics/17.8.754).
- [118] John P. Huelsenbeck et al. “Bayesian Inference of Phylogeny and Its Impact on Evolutionary Biology”. In: *Science* 294.5550 (2001), pp. 2310–2314. ISSN: 0036-8075. DOI: [10.1126/science.1065889](https://doi.org/10.1126/science.1065889).
- [119] Yves Hüsemann et al. “Systemic spread is an early step in breast cancer”. In: *Cancer cell* 13.1 (2008), pp. 58–68.
- [120] Erica L Jackson et al. “Analysis of lung tumor initiation and progression using conditional expression of oncogenic K-ras”. In: *Genes & development* 15.24 (2001), pp. 3243–3248.
- [121] Erica L Jackson et al. “The differential effects of mutant p53 alleles on advanced murine lung cancer”. In: *Cancer research* 65.22 (2005), pp. 10280–10288.

- [122] Ariel Jaffe et al. “Spectral neighbor joining for reconstruction of latent tree models”. In: (2020). DOI: [10.48550/ARXIV.2002.12547](https://doi.org/10.48550/ARXIV.2002.12547).
- [123] Mariam Jamal-Hanjani et al. “Tracking genomic cancer evolution for precision medicine: the lung TRACERx study”. In: *PLoS biology* 12.7 (2014), e1001906.
- [124] Mariam Jamal-Hanjani et al. “Tracking the evolution of non–small-cell lung cancer”. In: *New England Journal of Medicine* 376.22 (2017), pp. 2109–2121.
- [125] Martin Jechlinger et al. “Expression profiling of epithelial plasticity in tumor progression”. In: *Oncogene* 22.46 (2003), pp. 7155–7169.
- [126] Hongbin Ji et al. “LKB1 modulates lung cancer differentiation and metastasis”. In: *Nature* 448.7155 (2007), pp. 807–810.
- [127] Matthew G Jones et al. “Inference of single-cell phylogenies from lineage tracing data using Cassiopeia”. In: *Genome biology* 21.1 (2020), pp. 1–27.
- [128] Matthew G. Jones, Yanay Rosen, and Nir Yosef. “PhyloVision: Interactive Software for Integrated Analysis of Single-Cell Transcriptomic and Phylogenetic Data”. In: *bioRxiv* (2021). DOI: [10.1101/2021.09.13.460142](https://doi.org/10.1101/2021.09.13.460142).
- [129] Stephen K Jones et al. “Massively parallel kinetic profiling of natural and engineered CRISPR nucleases”. In: *Nature Biotechnology* 39.1 (2021), pp. 84–93.

- [130] Marco Jost et al. "Combined CRISPRi/a-Based Chemical Genetic Screens Reveal that Rigosertib Is a Microtubule-Destabilizing Agent". In: *Molecular Cell* 68.1 (2017), 210–223.e6. ISSN: 1097-2765. doi: <https://doi.org/10.1016/j.molcel.2017.09.012>.
- [131] Marco Jost et al. "Titrating gene expression with series of systematically compromised CRISPR guide RNAs". In: *bioRxiv* (2019). doi: [10.1101/717389](https://doi.org/10.1101/717389).
- [132] Jeffrey B. Joy et al. "Ancestral Reconstruction". In: *PLOS Computational Biology* 12.7 (July 2016), pp. 1–20. doi: [10.1371/journal.pcbi.1004763](https://doi.org/10.1371/journal.pcbi.1004763).
- [133] Reza Kalhor, Prashant Mali, and George M. Church. "Rapidly evolving homing CRISPR barcodes". In: *Nature Methods* 14 (2016), 195 EP.
- [134] Reza Kalhor et al. "Developmental barcoding of whole mouse via homing CRISPR". In: *Science* 361.6405 (2018). ISSN: 0036-8075. doi: [10.1126/science.aat9804](https://doi.org/10.1126/science.aat9804).
- [135] Mohammed El-Kebir, Gryte Satas, and Benjamin J Raphael. "Inferring parsimonious migration histories for metastatic cancers". In: *Nature genetics* 50.5 (2018), pp. 718–726.
- [136] Mohammed El-Kebir et al. "Inferring the mutational history of a tumor using multi-state perfect phylogeny mixtures". In: *Cell systems* 3.1 (2016), pp. 43–53.
- [137] Mohammed El-Kebir et al. "Reconstruction of clonal trees and tumor composition from multi-sample sequencing data". In: *Bioinformatics* 31.12 (2015), pp. i62–i70.
- [138] David G Kendall. "On the generalized "birth-and-death" process". In: *The annals of mathematical statistics* 19.1 (1948), pp. 1–15.

- [139] Samuel A Kerk et al. “Metabolic networks in mutant KRAS-driven tumours: tissue specificities and the microenvironment”. In: *Nature reviews Cancer* 21.8 (2021), pp. 510–525.
- [140] Lennart Kester and Alexander van Oudenaarden. “Single-Cell Transcriptomics Meets Lineage Tracing”. In: *Cell Stem Cell* 23.2 (2018), pp. 166–179. ISSN: 1934-5909. DOI: <https://doi.org/10.1016/j.stem.2018.04.014>.
- [141] Charissa Kim et al. “Chemoresistance evolution in triple-negative breast cancer delineated by single-cell sequencing”. In: *Cell* 173.4 (2018), pp. 879–893.
- [142] Hui Kwon Kim et al. “Deep learning improves prediction of CRISPR-Cpf1 guide RNA activity”. In: *Nature Biotechnology* 36 (2018), 239 EP.
- [143] Hui Kwon Kim et al. “Predicting the efficiency of prime editing guide RNAs in human cells”. In: *Nature Biotechnology* 39.2 (2021), pp. 198–206.
- [144] Junhyong Kim and Michael J Sanderson. “Penalized likelihood phylogenetic inference: bridging the parsimony-likelihood gap”. In: *Systematic biology* 57.5 (2008), pp. 665–674.
- [145] Motoo Kimura. “The Number of Heterozygous Nucleotide Sites Maintained in a Finite Population Due to Steady Flux of Mutations”. In: *Genetics* 61.4 (), pp. 893–903.
- [146] Allon M. Klein et al. “Droplet Barcoding for Single-Cell Transcriptomics Applied to Embryonic Stem Cells”. In: *Cell* 161.5 (May 2015), pp. 1187–1201. DOI: [10.1016/j.cell.2015.04.044](https://doi.org/10.1016/j.cell.2015.04.044).

- [147] Christoph A Klein. “Parallel progression of primary tumours and metastases”. In: *Nature Reviews Cancer* 9.4 (2009), pp. 302–312.
- [148] Bryan Kolaczkowski and Joseph W. Thornton. “Performance of maximum parsimony and likelihood phylogenetics when evolution is heterogeneous”. In: *Nature* 431.7011 (2004), pp. 980–984. ISSN: 1476-4687. DOI: [10.1038/nature02917](https://doi.org/10.1038/nature02917). URL: <https://doi.org/10.1038/nature02917>.
- [149] Alexis C. Komor et al. “Programmable editing of a target base in genomic DNA without double-stranded DNA cleavage”. In: *Nature* 533 (2016), 420 EP.
- [150] Gioele La Manno et al. “RNA velocity of single cells”. In: *Nature* 560.7719 (2018), pp. 494–498.
- [151] Lindsay M LaFave et al. “Epigenomic state transitions characterize tumor progression in mouse lung adenocarcinoma”. In: *Cancer cell* 38.2 (2020), pp. 212–228.
- [152] Arthur W Lambert, Diwakar R Pattabiraman, and Robert A Weinberg. “Emerging biological principles of metastasis”. In: *Cell* 168.4 (2017), pp. 670–691.
- [153] Xiaoyang Lan et al. “Fate mapping of human glioblastoma reveals an invariant stem cell hierarchy”. In: *Nature* 549.7671 (2017), pp. 227–232.
- [154] Joshua T Lange et al. “Principles of ecDNA random inheritance drive rapid genome change and therapy resistance in human cancers”. In: *bioRxiv* (2021).

- [155] Robert R Langley and Isaiah J Fidler. “The seed and soil hypothesis revisited—The role of tumor-stroma interactions in metastasis to different organs”. In: *International journal of cancer* 128.11 (2011), pp. 2527–2535.
- [156] Michael Lässig, Ville Mustonen, and Aleksandra M Walczak. “Predicting evolution”. In: *Nature ecology & evolution* 1.3 (2017), pp. 1–9.
- [157] Ashley M Laughney et al. “Regenerative lineages and immune-mediated pruning in lung cancer metastasis”. In: *Nature medicine* 26.2 (2020), pp. 259–269.
- [158] Je Hyuk Lee et al. “Highly multiplexed subcellular RNA sequencing in situ”. In: *Science* 343.6177 (2014), pp. 1360–1363.
- [159] F. Lemoine et al. “Renewing Felsenstein’s phylogenetic bootstrap in the era of big data”. In: *Nature* 556.7702 (2018), pp. 452–456. ISSN: 1476-4687. DOI: [10.1038/s41586-018-0043-0](https://doi.org/10.1038/s41586-018-0043-0).
- [160] Amy Li et al. “IL-33 signaling alters regulatory T cell diversity in support of tumor development”. In: *Cell reports* 29.10 (2019), pp. 2998–3008.
- [161] Shuqin Li et al. “Interferon alpha-inducible protein 27 promotes epithelial–mesenchymal transition and induces ovarian tumorigenicity and stemness”. In: *journal of surgical research* 193.1 (2015), pp. 255–264.
- [162] Jianbo Liu et al. “Keratin 17 promotes lung adenocarcinoma progression by enhancing cell proliferation and invasion”. In: *Medical Science Monitor: International Medical Journal of Experimental and Clinical Research* 24 (2018), p. 4782.

- [163] Jean Livet et al. “Transgenic strategies for combinatorial expression of fluorescent proteins in the nervous system”. In: *Nature* 450.7166 (2007), pp. 56–62.
- [164] Romain Lopez et al. “Deep generative modeling for single-cell transcriptomics”. In: *Nature methods* 15.12 (2018), pp. 1053–1058.
- [165] Chin Lung Lu, Chuan Yi Tang, and Richard Chia-Tung Lee. “The full Steiner tree problem”. In: *Theoretical Computer Science* 306.1 (2003), pp. 55–67. ISSN: 0304-3975. DOI: [https://doi.org/10.1016/S0304-3975\(03\)00209-3](https://doi.org/10.1016/S0304-3975(03)00209-3).
- [166] Leif S Ludwig et al. “Lineage tracing in humans enabled by mitochondrial mutations and single-cell genomics”. In: *Cell* 176.6 (2019), pp. 1325–1339.
- [167] Sai Ma et al. “Chromatin potential identified by shared single-cell profiling of RNA and chromatin”. In: *Cell* 183.4 (2020), pp. 1103–1116.
- [168] Evan Z. Macosko et al. “Highly Parallel Genome-wide Expression Profiling of Individual Cells Using Nanoliter Droplets”. In: *Cell* 161.5 (May 2015), pp. 1202–1214. DOI: [10.1016/j.cell.2015.05.002](https://doi.org/10.1016/j.cell.2015.05.002).
- [169] Salem Malikic et al. “Integrative inference of subclonal tumour evolution from single-cell and bulk sequencing data”. In: *Nature communications* 10.1 (2019), pp. 1–12.
- [170] Nemanja Despot Marjanovic et al. “Emergence of a high-plasticity cell state during lung cancer evolution”. In: *Cancer cell* 38.2 (2020), pp. 229–246.

- [171] Joan Massagué and Anna C Obenauf. “Metastatic colonization by circulating tumour cells”. In: *Nature* 529.7586 (2016), pp. 298–306.
- [172] Ashley Maynard et al. “Therapy-induced evolution of human lung cancer revealed by single-cell RNA sequencing”. In: *Cell* 182.5 (2020), pp. 1232–1251.
- [173] Katie McDole et al. “In toto imaging and reconstruction of post-implantation mouse development at the single-cell level”. In: *Cell* 175.3 (2018), pp. 859–876.
- [174] David G McFadden et al. “Mutational landscape of EGFR-, MYC-, and Kras-driven genetically engineered mouse models of lung adenocarcinoma”. In: *Proceedings of the National Academy of Sciences* 113.42 (2016), E6409–E6417.
- [175] Christopher S. McGinnis, Lyndsay M. Murrow, and Zev J. Gartner. “DoubletFinder: Doublet Detection in Single-Cell RNA Sequencing Data Using Artificial Nearest Neighbors”. In: *Cell Systems* 8.4 (2019), 329–337.e4. ISSN: 2405-4712. DOI: [10.1016/j.cels.2019.03.003](https://doi.org/10.1016/j.cels.2019.03.003).
- [176] Nicholas McGranahan and Charles Swanton. “Clonal heterogeneity and tumor evolution: past, present, and the future”. In: *Cell* 168.4 (2017), pp. 613–628.
- [177] Aaron McKenna and James A. Gagnon. “Recording development with single cell dynamic lineage tracing”. In: *Development* 146.12 (2019). ISSN: 0950-1991. DOI: [10.1242/dev.169730](https://doi.org/10.1242/dev.169730).
- [178] Aaron McKenna et al. “Whole organism lineage tracing by combinatorial and cumulative genome editing”. In: *Science* (2016). ISSN: 0036-8075. DOI: [10.1126/science.aaf7907](https://doi.org/10.1126/science.aaf7907).

- [179] Andrew McPherson et al. “Divergent modes of clonal spread and intraperitoneal mixing in high-grade serous ovarian cancer”. In: *Nature genetics* 48.7 (2016), p. 758.
- [180] Lauren MF Merlo et al. “Cancer as an evolutionary and ecological process”. In: *Nature reviews cancer* 6.12 (2006), pp. 924–935.
- [181] R Mihaescu, D Levy, and L Pachter. “Why Neighbor-Joining Works”. In: *arXiv* (2006). DOI: [arXiv:cs/0602041v3](https://arxiv.org/abs/cs/0602041v3).
- [182] Eleni P Mimitou et al. “Scalable, multimodal profiling of chromatin accessibility, gene expression and protein levels in single cells”. In: *Nature biotechnology* 39.10 (2021), pp. 1246–1258.
- [183] Christopher W Murray et al. “An LKB1–SIK axis suppresses lung tumor growth and controls differentiation”. In: *Cancer discovery* 9.11 (2019), pp. 1590–1605.
- [184] Sanne M van Neerven et al. “Apc-mutant cells act as supercompetitors in intestinal tumour initiation”. In: *Nature* 594.7863 (2021), pp. 436–441.
- [185] Cyril Neftel et al. “An integrative model of cellular states, plasticity, and genetics for glioblastoma”. In: *Cell* 178.4 (2019), pp. 835–849.
- [186] Richard A Neher, Colin A Russell, and Boris I Shraiman. “Predicting evolution from the shape of genealogical trees”. In: *Elife* 3 (2014), e03568.
- [187] Don X Nguyen et al. “WNT/TCF signaling through LEF1 and HOXB9 mediates lung adenocarcinoma metastasis”. In: *Cell* 138.1 (2009), pp. 51–62.

- [188] Peter C. Nowell. “The Clonal Evolution of Tumor Cell Populations”. In: *Science* 194.4260 (1976), pp. 23–28. DOI: [10.1126/science.959840](https://doi.org/10.1126/science.959840).
- [189] Ross A Okimoto et al. “Inactivation of Capicua drives cancer metastasis”. In: *Nature genetics* 49.1 (2017), pp. 87–96.
- [190] Thordur Oskarsson, Eduard Batlle, and Joan Massagué. “Metastatic stem cells: sources, niches, and vital pathways”. In: *Cell stem cell* 14.3 (2014), pp. 306–321.
- [191] Khalil Ouardini et al. “Reconstructing unobserved cellular states from paired single-cell lineage tracing and transcriptomics data”. In: *bioRxiv* (2021).
- [192] Jihye Park et al. “Recording of elapsed time and temporal information about biological events using Cas9”. In: *Cell* 184.4 (2021), pp. 1047–1063.
- [193] Marie J Parsons, Tuomas Tammela, and Lukas E Dow. “WNT as a Driver and Dependency in Cancer”. In: *Cancer Discovery* 11.10 (2021), pp. 2413–2429.
- [194] Anoop P Patel et al. “Single-cell RNA-seq highlights intratumoral heterogeneity in primary glioblastoma”. In: *Science* 344.6190 (2014), pp. 1396–1401.
- [195] Weike Pei et al. “Polylox barcoding reveals haematopoietic stem cell fates realized in vivo”. In: *Nature* 548.7668 (2017), pp. 456–460.
- [196] Ethel R Pereira et al. “Lymph node metastases can invade local blood vessels, exit the node, and colonize distant organs in mice”. In: *Science* 359.6382 (2018), pp. 1403–1407.

- [197] Sarah E Pierce et al. “LKB1 inactivation modulates chromatin accessibility to drive metastatic progression”. In: *Nature Cell Biology* 23.8 (2021), pp. 915–924.
- [198] Katrina Podsypanina et al. “Seeding and propagation of untransformed mouse mammary cells in the lung”. In: *Science* 321.5897 (2008), pp. 1841–1844.
- [199] Nicola E Potter et al. “Single-cell mutational profiling and clonal phylogeny in cancer”. In: *Genome research* 23.12 (2013), pp. 2115–2125.
- [200] Thomas Powles et al. “ctDNA guiding adjuvant immunotherapy in urothelial carcinoma”. In: *Nature* 595.7867 (2021), pp. 432–437.
- [201] Prem K Premsrirut et al. “A rapid and scalable system for studying gene function in mice using conditional RNA interference”. In: *Cell* 145.1 (2011), pp. 145–158.
- [202] Morgan N. Price, Paramvir S. Dehal, and Adam P. Arkin. “FastTree: Computing Large Minimum Evolution Trees with Profiles instead of a Distance Matrix”. In: *Molecular Biology and Evolution* 26.7 (2009), pp. 1641–1650. DOI: [10.1093/molbev/msp077](https://doi.org/10.1093/molbev/msp077).
- [203] Jeffrey J Quinn et al. “Single-cell lineages reveal the rates, routes, and drivers of metastasis in cancer xenografts”. In: *Science* 371.6532 (2021), eabc1944.
- [204] Álvaro Quintanal-Villalonga et al. “Lineage plasticity in cancer: a shared pathway of therapeutic resistance”. In: *Nature reviews Clinical oncology* 17.6 (2020), pp. 360–371.
- [205] Shiran Rabinovich et al. “Diversion of aspartate in ASS1-deficient tumours fosters de novo pyrimidine synthesis”. In: *Nature* 527.7578 (2015), pp. 379–383.

- [206] Bushra Raj et al. “Simultaneous single-cell profiling of lineages and cell types in the vertebrate brain”. In: *Nature Biotechnology* 36 (2018), 442 EP.
- [207] Philipp Rathert et al. “Transcriptional plasticity promotes primary and acquired resistance to BET inhibition”. In: *Nature* 525.7570 (2015), pp. 543–547.
- [208] Johannes G Reiter et al. “Reconstructing metastatic seeding patterns of human cancers”. In: *Nature communications* 8.1 (2017), pp. 1–10.
- [209] Johannes G. Reiter et al. “Reconstructing metastatic seeding patterns of human cancers”. In: *Nature Communications* 8.1 (2017), p. 14114. ISSN: 2041-1723. DOI: [10.1038/ncomms14114](https://doi.org/10.1038/ncomms14114).
- [210] Andrew D Rhim et al. “EMT and dissemination precede pancreatic tumor formation”. In: *Cell* 148.1-2 (2012), pp. 349–361.
- [211] D.F. Robinson and L.R. Foulds. “Comparison of phylogenetic trees”. In: *Mathematical Biosciences* 53.1 (1981), pp. 131–147. ISSN: 0025-5564. DOI: [https://doi.org/10.1016/0025-5564\(81\)90043-2](https://doi.org/10.1016/0025-5564(81)90043-2).
- [212] Samuel G Rodriques et al. “Slide-seq: A scalable technology for measuring genome-wide expression at high spatial resolution”. In: *Science* 363.6434 (2019), pp. 1463–1467.
- [213] Zoë N Rogers et al. “A quantitative and multiplexed approach to uncover the fitness landscape of tumor suppression in vivo”. In: *Nature methods* 14.7 (2017), pp. 737–742.
- [214] Zoë N Rogers et al. “Mapping the in vivo fitness landscape of lung adenocarcinoma tumor suppression in mice”. In: *Nature genetics* 50.4 (2018), pp. 483–486.

- [215] Christophe Rosty et al. "Identification of a proliferation gene cluster associated with HPV E6/E7 expression level and viral DNA load in invasive cervical carcinoma". In: *Oncogene* 24.47 (2005), pp. 7094–7104.
- [216] N Saitou and M Nei. "The neighbor-joining method: a new method for reconstructing phylogenetic trees." In: *Molecular Biology and Evolution* 4.4 (1987), pp. 406–425. DOI: [10.1093/oxfordjournals.molbev.a040454](https://doi.org/10.1093/oxfordjournals.molbev.a040454).
- [217] Sohrab Salehi et al. "Clonal fitness inferred from time-series modelling of single-cell cancer genomes". In: *Nature* 595.7868 (2021), pp. 585–590.
- [218] Irepan Salvador-Martínez et al. "Is it possible to reconstruct an accurate cell lineage using CRISPR recorders?" In: *eLife* 8 (2019), e40292. ISSN: 2050-084X. DOI: [10.7554/eLife.40292](https://doi.org/10.7554/eLife.40292).
- [219] David Sankoff. "Minimal Mutation Trees of Sequences". In: *SIAM Journal on Applied Mathematics* 28.1 (1975), pp. 35–42. DOI: [10.1137/0128004](https://doi.org/10.1137/0128004).
- [220] Gryte Satas et al. "Scarlet: Single-cell tumor phylogeny inference with copy-number constrained mutation losses". In: *Cell systems* 10.4 (2020), pp. 323–332.
- [221] Carl F Schaefer et al. "PID: the pathway interaction database". In: *Nucleic acids research* 37.suppl_1 (2009), pp. D674–D679.
- [222] Arnout G Schepers et al. "Lineage tracing reveals Lgr5+ stem cell activity in mouse intestinal adenomas". In: *Science* 337.6095 (2012), pp. 730–735.

- [223] Russell Schwartz and Alejandro A Schäffer. “The evolution of tumour phylogenetics: principles and practice”. In: *Nature Reviews Genetics* 18.4 (2017), pp. 213–229.
- [224] Gur Sevillya, Zeev Frenkel, and Sagi Snir. “Triplet MaxCut: a new toolkit for rooted supertree”. In: *Methods in Ecology and Evolution* 7.11 (2016), pp. 1359–1365. DOI: [10.1111/2041-210X.12606](https://doi.org/10.1111/2041-210X.12606).
- [225] Sydney M Shaffer et al. “Rare cell variability and drug-induced reprogramming as a mode of cancer drug resistance”. In: *Nature* 546.7658 (2017), pp. 431–435.
- [226] Charles J Sherr. “Principles of tumor suppression”. In: *Cell* 116.2 (2004), pp. 235–246.
- [227] David JH Shih et al. “Genomic characterization of human brain metastases identifies drivers of metastatic lung adenocarcinoma”. In: *Nature genetics* 52.4 (2020), pp. 371–377.
- [228] Kamen P Simeonov et al. “Single-cell lineage tracing of metastatic cancer reveals selection of hybrid EMT states”. In: *Cancer Cell* 39.8 (2021), pp. 1150–1162.
- [229] Ansam Sinjab et al. “Resolving the spatial and cellular architecture of lung adenocarcinoma by multiregion single-cell sequencing”. In: *Cancer Discovery* 11.10 (2021), pp. 2506–2523.
- [230] Tobias Sjöblom et al. “The Consensus Coding Sequences of Human Breast and Colorectal Cancers”. In: *Science* 314.5797 (2006), pp. 268–274. DOI: [10.1126/science.1133427](https://doi.org/10.1126/science.1133427).
- [231] Ferdinandos Skoulidis et al. “Co-occurring genomic alterations define major subsets of KRAS-mutant lung adenocarcinoma with distinct biology, immune profiles, and therapeutic vulnerabilities”. In: *Cancer discovery* 5.8 (2015), pp. 860–877.

- [232] Montgomery Slatkin and Wayne P Maddison. “A cladistic measure of gene flow inferred from the phylogenies of alleles.” In: *Genetics* 123.3 (1989), pp. 603–613.
- [233] Sagi Snir and Satish Rao. “Using max cut to enhance rooted trees consistency”. In: *IEEE/ACM transactions on computational biology and bioinformatics* 3.4 (2006), pp. 323–333.
- [234] Robert R Sokal. “A statistical method for evaluating systematic relationships.” In: *Univ. Kansas, Sci. Bull.* 38 (1958), pp. 1409–1438.
- [235] Andrea Sottoriva et al. “A Big Bang model of human colorectal tumor growth”. In: *Nature Genetics* 47.3 (2015), pp. 209–216. ISSN: 1546-1718. DOI: [10.1038/ng.3214](https://doi.org/10.1038/ng.3214).
- [236] Bastiaan Spanjaard et al. “Simultaneous lineage tracing and cell-type identification using CRISPR-Cas9-induced genetic scars”. In: *Nature Biotechnology* 36 (2018), 469 EP.
- [237] Leo Speidel et al. “A method for genome-wide genealogy estimation for thousands of samples”. In: *Nature Genetics* 51.9 (2019), pp. 1321–1329. ISSN: 1546-1718. DOI: [10.1038/s41588-019-0484-x](https://doi.org/10.1038/s41588-019-0484-x).
- [238] Tanja Stadler, Oliver G Pybus, and Michael PH Stumpf. “Phylodynamics for cell biologists”. In: *Science* 371.6526 (2021), eaah6266.
- [239] Patrik L Ståhl et al. “Visualization and analysis of gene expression in tissue sections by spatial transcriptomics”. In: *Science* 353.6294 (2016), pp. 78–82.

- [240] Michael Steel. “The complexity of reconstructing trees from qualitative characters and subtrees”. In: *Journal of Classification* 9.1 (1992), pp. 91–116. ISSN: 1432-1343. DOI: [10.1007/BF02618470](https://doi.org/10.1007/BF02618470).
- [241] Michael Steel. “The complexity of reconstructing trees from qualitative characters and subtrees”. In: *Journal of classification* 9.1 (1992), pp. 91–116.
- [242] Robert R. Stickels et al. “Highly sensitive spatial transcriptomics at near-cellular resolution with Slide-seqV2”. In: *Nature Biotechnology* 39.3 (2021), pp. 313–319. ISSN: 1546-1696. DOI: [10.1038/s41587-020-0739-1](https://doi.org/10.1038/s41587-020-0739-1).
- [243] Marlon Stoeckius et al. “Cell Hashing with barcoded antibodies enables multiplexing and doublet detection for single cell genomics”. In: *Genome Biology* 19.1 (2018), p. 224. ISSN: 1474-760X. DOI: [10.1186/s13059-018-1603-1](https://doi.org/10.1186/s13059-018-1603-1).
- [244] Marlon Stoeckius et al. “Simultaneous epitope and transcriptome measurement in single cells”. In: *Nature Methods* 14.9 (July 2017), pp. 865–868. DOI: [10.1038/nmeth.4380](https://doi.org/10.1038/nmeth.4380).
- [245] Michael R. Stratton, Peter J. Campbell, and P. Andrew Futreal. “The cancer genome”. In: *Nature* 458.7239 (2009), pp. 719–724. ISSN: 1476-4687. DOI: [10.1038/nature07943](https://doi.org/10.1038/nature07943).
- [246] Aravind Subramanian et al. “Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles”. In: *Proceedings of the National Academy of Sciences* 102.43 (2005), pp. 15545–15550.
- [247] Ken Sugino and Tzumin Lee. “Robust reconstruction of CRISPR and tumor lineage using depth metrics”. In: *bioRxiv* (2019), p. 609107.

- [248] J.E. Sulston et al. “The embryonic cell lineage of the nematode *Caenorhabditis elegans*”. In: *Developmental Biology* 100.1 (1983), pp. 64–119. ISSN: 0012-1606. DOI: [https://doi.org/10.1016/0012-1606\(83\)90201-4](https://doi.org/10.1016/0012-1606(83)90201-4).
- [249] Si Sun et al. “REG4 is an indicator for KRAS mutant lung adenocarcinoma with TTF-1 low expression”. In: *Journal of Cancer Research and Clinical Oncology* 145.9 (2019), pp. 2273–2283.
- [250] Fumio Tajima. “Infinite-allele model and infinite-site model in population genetics”. In: *Journal of Genetics* 75.1 (1996), p. 27. DOI: [10.1007/BF02931749](https://doi.org/10.1007/BF02931749).
- [251] Y Esther Tak et al. “Inducible and multiplex gene regulation using CRISPR–Cpf1-based transcription factors”. In: *Nature methods* 14.12 (2017), pp. 1163–1166.
- [252] Tuomas Tammela and Julien Sage. “Investigating tumor heterogeneity in mouse models”. In: *Annual review of cancer biology* 4 (2020), pp. 99–119.
- [253] Tuomas Tammela et al. “A Wnt-producing niche drives proliferative potential and progression in lung adenocarcinoma”. In: *Nature* 545.7654 (2017), pp. 355–359.
- [254] Weixin Tang and David R Liu. “Rewritable multi-event analog recording in bacterial and mammalian cells”. In: *Science* 360.6385 (2018), eaap8992.
- [255] Liming Tao et al. “Retrospective cell lineage reconstruction in humans by using short tandem repeats”. In: *Cell reports methods* 1.3 (2021), p. 100054.

- [256] Maxime Tarabichi et al. “A practical guide to cancer subclonal reconstruction from DNA sequencing”. In: *Nature methods* 18.2 (2021), pp. 144–155.
- [257] Masoud F Tavazoie et al. “LXR/ApoE activation restricts innate immune suppression in cancer”. In: *Cell* 172.4 (2018), pp. 825–840.
- [258] Jeffrey P. Townsend. “Profiling Phylogenetic Informativeness”. In: *Systematic Biology* 56.2 (2007), pp. 222–231. DOI: [10.1080/10635150701311362](https://doi.org/10.1080/10635150701311362).
- [259] Vincent A Traag, Ludo Waltman, and Nees Jan Van Eck. “From Louvain to Leiden: guaranteeing well-connected communities”. In: *Scientific reports* 9.1 (2019), pp. 1–12.
- [260] “Tracing the tumor lineage”. In: *Molecular Oncology* 4.3 (2010), pp. 267–283. ISSN: 1574-7891. DOI: <https://doi.org/10.1016/j.molonc.2010.04.010>.
- [261] Cole Trapnell et al. “The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells”. In: *Nature biotechnology* 32.4 (2014), pp. 381–386.
- [262] Sophie Tritchler et al. “Concepts and limitations for learning developmental trajectories from single cell genomics”. In: *Development* 146.12 (2019), dev170506.
- [263] Samra Turajlic and Charles Swanton. “Metastasis as an evolutionary process”. In: *Science* 352.6282 (2016), pp. 169–175. DOI: [10.1126/science.aaf2784](https://doi.org/10.1126/science.aaf2784).
- [264] Koen Van den Berge et al. “Trajectory-based differential expression analysis for single-cell sequencing data”. In: *Nature communications* 11.1 (2020), pp. 1–13.

- [265] Roberto Vendramin, Kevin Litchfield, and Charles Swanton. “Cancer evolution: Darwin and beyond”. In: *The EMBO Journal* 40.18 (2021), e108389.
- [266] Michel Verleysen and Damien François. “The Curse of Dimensionality in Data Mining and Time Series Prediction”. In: (2005). Ed. by Joan Cabestany, Alberto Prieto, and Francisco Sandoval, pp. 758–770.
- [267] Bert Vogelstein et al. “Cancer genome landscapes”. In: *science* 339.6127 (2013), pp. 1546–1558.
- [268] Bert Vogelstein et al. “Genetic alterations during colorectal-tumor development”. In: *New England Journal of Medicine* 319.9 (1988), pp. 525–532.
- [269] Daniel E Wagner and Allon M Klein. “Lineage tracing meets single-cell omics: opportunities and challenges”. In: *Nature Reviews Genetics* 21.7 (2020), pp. 410–427.
- [270] Daniel E. Wagner et al. “Single-cell mapping of gene expression landscapes and lineage in the zebrafish embryo”. In: *Science* 360.6392 (2018), pp. 981–987. ISSN: 0036-8075. DOI: [10.1126/science.aar4362](https://doi.org/10.1126/science.aar4362).
- [271] Hong Wang et al. “Knockdown of IFI27 inhibits cell proliferation and invasion in oral squamous cell carcinoma”. In: *World Journal of Surgical Oncology* 16.1 (2018), pp. 1–7.
- [272] Minghui Wang, Yongzhong Zhao, and Bin Zhang. “Efficient test and visualization of multi-set intersections”. In: *Scientific reports* 5.1 (2015), pp. 1–12.

- [273] Robert Wang et al. “Theoretical Guarantees for Phylogeny Inference from Single-Cell Lineage Tracing”. In: *bioRxiv* (2021).
- [274] Robert A Weinberg. “Tumor suppressor genes”. In: *Science* 254.5035 (1991), pp. 1138–1146.
- [275] Caleb Weinreb et al. “Lineage tracing on transcriptional landscapes links state to fate during differentiation”. In: *Science* 367.6479 (2020), eaaw3381.
- [276] J. F. Weng, I. Mareels, and D. A. Thomas. “Probability Steiner trees and maximum parsimony in phylogenetic analysis”. In: *Journal of Mathematical Biology* 64.7 (2012), pp. 1225–1251. ISSN: 1432-1416. DOI: [10.1007/s00285-011-0442-4](https://doi.org/10.1007/s00285-011-0442-4).
- [277] Peter MK Westcott et al. “The mutational landscapes of genetic and chemical models of Kras-driven lung cancer”. In: *Nature* 517.7535 (2015), pp. 489–492.
- [278] Michael L Whitfield et al. “Identification of genes periodically expressed in the human cell cycle and their expression in tumors”. In: *Molecular biology of the cell* 13.6 (2002), pp. 1977–2000.
- [279] Marc J Williams et al. “Quantification of subclonal selection in cancer from bulk sequencing data”. In: *Nature genetics* 50.6 (2018), pp. 895–903.
- [280] Monte M Winslow et al. “Suppression of lung adenocarcinoma progression by Nkx2-1”. In: *Nature* 473.7345 (2011), pp. 101–104.

- [281] Ian P Winters, Christopher W Murray, and Monte M Winslow. “Towards quantitative and multiplexed in vivo functional cancer genomics”. In: *Nature Reviews Genetics* 19.12 (2018), pp. 741–755.
- [282] F Alexander Wolf, Philipp Angerer, and Fabian J Theis. “SCANPY: large-scale single-cell gene expression data analysis”. In: *Genome biology* 19.1 (2018), pp. 1–5.
- [283] Samuel L. Wolock, Romain Lopez, and Allon M. Klein. “Scrublet: Computational Identification of Cell Doublets in Single-Cell Transcriptomic Data”. In: *Cell Systems* 8.4 (2019), 281–291.e9. ISSN: 2405-4712. DOI: <https://doi.org/10.1016/j.cels.2018.11.005>.
- [284] Mollie B Woodworth, Kelly M Girskis, and Christopher A Walsh. “Building a lineage from single cells: genetic techniques for cell lineage tracking”. In: *Nature Reviews Genetics* 18.4 (2017), pp. 230–244.
- [285] Szu-Hsien Sam Wu, Ji-Hyun Lee, and Bon-Kyoung Koo. “Lineage tracing: computational reconstruction goes beyond the limit of imaging”. In: *Molecules and cells* 42.2 (2019), p. 104.
- [286] Zhenxiang Xi, Liang Liu, and Charles C. Davis. “The Impact of Missing Data on Species Tree Estimation”. In: *Molecular Biology and Evolution* 33.3 (Nov. 2015), pp. 838–860. ISSN: 0737-4038. DOI: [10.1093/molbev/msv266](https://doi.org/10.1093/molbev/msv266).
- [287] Chenling Xu et al. “Probabilistic harmonization and annotation of single-cell transcriptomics data with deep generative models”. In: *Molecular systems biology* 17.1 (2021), e9620.

- [288] Jun Yan et al. "Inhibiting of proliferation, migration, and invasion in lung cancer induced by silencing interferon-induced transmembrane protein 1 (IFITM1)". In: *BioMed Research International* 2019 (2019).
- [289] Dian Yang et al. "Lineage Recording Reveals the Phylodynamics, Plasticity and Paths of Tumor Evolution". In: *bioRxiv* (2021). DOI: [10.1101/2021.10.12.464111](https://doi.org/10.1101/2021.10.12.464111).
- [290] Hui Yang et al. "Base editing generates substantial off-target single nucleotide variants". In: *bioRxiv* (2018). DOI: [10.1101/480145](https://doi.org/10.1101/480145).
- [291] Ziheng Yang and Bruce Rannala. "Molecular phylogenetics: principles and practice". In: *Nature Reviews Genetics* 13 (2012), 303 EP.
- [292] Salina Yuan, Robert J Norgard, and Ben Z Stanger. "Cellular plasticity in cancer". In: *Cancer discovery* 9.7 (2019), pp. 837–851.
- [293] Hamim Zafar, Chieh Lin, and Ziv Bar-Joseph. "Single-cell Lineage Tracing by Integrating CRISPR-Cas9 Mutations with Transcriptomic Data." In: *bioRxiv* (2019). DOI: [10.1101/630814](https://doi.org/10.1101/630814).
- [294] Nastaran Zahir et al. "Characterizing the ecological and evolutionary dynamics of cancer". In: *Nature genetics* 52.8 (2020), pp. 759–767.
- [295] Weijie Zhang et al. "The bone microenvironment invigorates metastatic seeds for further dissemination". In: *Cell* 184.9 (2021), pp. 2471–2486.

- [296] Xiannian Zhang et al. “Comparative Analysis of Droplet-Based Ultra-High-Throughput Single-Cell RNA-Seq Systems”. In: *Molecular Cell* 73.1 (Jan. 2019), 130–142.e5. doi: [10.1016/j.molcel.2018.10.020](https://doi.org/10.1016/j.molcel.2018.10.020).
- [297] Xiaomei Zhang et al. “A renewable tissue resource of phenotypically stable, biologically and ethnically diverse, patient-derived human breast cancer xenograft models”. In: *Cancer research* 73.15 (2013), pp. 4885–4897.
- [298] Grace X. Y. Zheng et al. “Massively parallel digital transcriptional profiling of single cells”. In: *Nature Communications* 8.1 (Jan. 2017). doi: [10.1038/ncomms14049](https://doi.org/10.1038/ncomms14049).
- [299] Yinghui Zhou et al. “Chimeric mouse tumor models reveal differences in pathway activation between ERBB family–and KRAS-dependent lung adenocarcinomas”. In: *Nature biotechnology* 28.1 (2010), pp. 71–78.
- [300] Leonid Zosin and Samir Khuller. “On Directed Steiner Trees”. In: *Proceedings of the Thirteenth Annual ACM-SIAM Symposium on Discrete Algorithms*. SODA '02. San Francisco, California: Society for Industrial and Applied Mathematics, 2002, pp. 59–63. ISBN: 0-89871-513-X.

Publishing Agreement

It is the policy of the University to encourage open access and broad distribution of all theses, dissertations, and manuscripts. The Graduate Division will facilitate the distribution of UCSF theses, dissertations, and manuscripts to the UCSF Library for open access and distribution. UCSF will make such theses, dissertations, and manuscripts accessible to the public and will take reasonable steps to preserve these works in perpetuity.

I hereby grant the non-exclusive, perpetual right to The Regents of the University of California to reproduce, publicly display, distribute, preserve, and publish copies of my thesis, dissertation, or manuscript in any form or media, now existing or later derived, including access online for teaching, research, and public service purposes.

DocuSigned by:

Matthew Jones

CF69FCA6064546C...

Author Signature

5/23/2022

Date