Elucidating Cancer Evolution using Single-Cell Sequencing and Comparative Genomics

A dissertation presented by

Robert Aboukhalil

to

The Watson School of Biological Sciences at
Cold Spring Harbor Laboratory

In partial fulfillment of the requirements for
the degree of **Doctor of Philosophy**
in **Biological Sciences**

Cold Spring Harbor Laboratory
March 2016

Table of Contents

List of Tables and Figures

List of Abbreviations

BED	Browser Extensible Data file
BAM	Binary SAM file
CBS	Circular Binary Segmentation
CNV	Copy-Number Variation
DOP-PCR	Degenerate oligonucleotide-primed PCR
FACS	Fluorescence Activated Cell Sorting
GMM	Gaussian Mixture Models
GO	Gene Ontology
IOD	Index Of Dispersion
LOWESS	Locally WEighted Scatterplot Smoothing
MAD	Median Absolute Deviation
MALBAC	Multiple Annealing and Looping Based Amplification Cycles
MCMC	Markov Chain Monte Carlo
MDA	Multiple Displacement Amplification
NGS	Next-Generation Sequencing
SAM	Sequence Alignment/Map file
scDNA-seq	Single-Cell DNA Sequencing
SNP	Single Nucleotide Polymorphism
SoS	Sum of squares
SRA	Sequence Read Archive
TSG	Tumor Suppressor Gene
WGA	Whole Genome Amplification

Acknowledgments

> ❝ *The scientist is not a person who gives the right answers; he's one who asks the right questions*
> *– Claude Lévi-Strauss*

I would like to sincerely thank my advisors Mickey Atwal and Michael Wigler for all their mentorship, support and guidance—and for teaching me that at least 900% of all science is about sanity checking your numbers. To all my committee members, Josh Dubnau, Mikala Egeblad, Alex Krasnitz and Bud Mishra, thank you very much for your valuable suggestions and insightful comments. I also want to thank Mike Schatz, for being a wonderful collaborator and unofficial-third-advisor. I am also grateful to the WSBS staff for all their help throughout the years; in particular, many thanks to Alyson Kass-Eisler, Kim Geer, Kim Creteur, Carrie Cowan and Alex Gann.

During my time at CSHL, I had the utmost pleasure to work with, and learn from, many great colleagues. Thank you to Tyler Garvin, Jude Kendall, Bernard Fendler, Ravi Kandasamy, Joan Alexander, Jim Hicks, Timour Baslan, Dan Levy, Vijay Kumar and Peter Andrews. I am also grateful to everyone who supported my extra-curricular and entrepreneurial ventures during my PhD. Many thanks to Jimmy Chan, for being a great business partner and fantastic friend; to Charla Lambert and David Stewart, for their help and support with launching CSHL's *Current Exchange* magazine; and to Bruce Stillman and David Spector, for supporting the Bioscience Enterprise Club.

My PhD life was also enriched by the many interactions I had with colleagues such as John Inglis, Jeffrey Nagel, Mickie Nagel, Charles Ryan, Dill Ayres, Hillary Sussman, and Anne Churchland. Thank you all for your inspiration and support; you have been great mentors and role models. Furthermore, my time at CSHL would not have been the same without all the great friendships that were formed over the years. Thank you in particular to my classmates of 2011 for making this a fun journey!

Many thanks are also due to my family, who will henceforth be subjected to a lifetime of *"No, not that kind of doctor"*. In particular, thank you to my parents George and Sylvia for all the sacrifices they made, and for their support and guidance throughout my life—none of this would have been possible without them. I also wish to thank my siblings, Anton and Darleen, for all the fun times and for always cheering me on. In addition, I would like to acknowledge my turtle Python, thanks to whom local grocery stores seem amused by my sizeable consumption of lettuce.

Last but most certainly not least, my heartfelt thanks to the unparalleled Maria Nattestad for her love, kindness and support. Many thanks are also due for her invaluable intellectual contributions to my life—I have truly enjoyed our discussions about bioinformatics, software engineering and entrepreneurship, and I look forward to more.

CHAPTER 1

Introduction

1.1. Cancer Genomics

Cancer is a family of diseases characterized by an accumulation of mutations. These genetic and epigenetic alterations impart the cell with properties of uncontrolled growth and proliferation, along with the potential for malignancy (Hanahan and Weinberg 2011). Beyond this shared property of genomic unrest, most cancers—and most patients—generally present vastly different mutation patterns, mechanisms of tumor growth, and response to treatment.

Cancer is a process that can be studied from multiple vantage points. From the single-cell perspective of an individual tumor, the heterogeneity and somatic evolution of the cancer can be dissected. From a comparative genomics perspective, the presence, location, function, and mechanism of cancer genes gives us clues as to how the cancer genome has morphed through germline evolution over millions of years. From both viewpoints, a valuable approach for studying cancer is to probe the underlying DNA to help identify these alterations. Advances in biotechnology, in particular Next-Generation Sequencing (NGS), have provided powerful tools to query the genome at base-pair resolution. As a result, sequencing has become a ubiquitous and affordable tool to study cancer in the research community, and has made important headways in clinical settings (Park et al. 2013).

The rise of NGS technologies has also resulted in new data types that allow us to address different biological questions, which in turn require the development of new algorithms and reproducible computational frameworks. Although much insight into cancer has been gleaned through the use of sequencing technologies (Meyerson et al. 2010), analyzing the genome of tumors remains a significant challenge due to intra-tumor heterogeneity.

1.2. The challenges of tumor heterogeneity

Tumors are heterogeneous populations of cells consisting of both cancer cells and normal cells (Pietras 2011; Marusyk et al. 2012). Within the cancerous population, cells exhibit a wide diversity of genetic and phenotypic properties. The concept of tumor heterogeneity gained ground in the 1980's through several lines of evidence, including early work using histological and cytogenetic approaches that characterized tumors as consisting of several clonal subpopulations (Shapiro et al. 1981; Yung et al. 1982). For example, cytogenetic analysis of breast carcinomas using G-banding identified tumor regions with distinct chromosomal abnormalities (Pandis et al. 1995; Teixeira et al. 1995). Other studies used fluorescence *in situ* hybridization (FISH) to detect copy-number events in a few loci, and identified clones with differing copy number profiles (Sauter et al. 1995; Pantou et al. 2005).

From a clinical perspective, deconstructing the heterogeneity of tumors is of critical importance as it severely impacts: **(1)** Accurate diagnosis and prognosis, since the most abundant clones are not necessarily the most malignant ones; **(2)** The course of patient treatment, as unidentified clones can cause resistance to chemotherapy (Fisher et al. 2013);

and **(3)** Drug development initiatives, as heterogeneity can blur the results of drug target validation studies (Murugaesu et al. 2013).

Two widely accepted hypotheses attempt to explain tumor heterogeneity: the clonal evolution model (Nowell 1976) and the cancer stem cell (CSC) hypothesis (Pierce and Speers 1988). The clonal evolution model, introduced in 1976 by Nowell, applies the ideas of natural selection to tumors. Starting from a cell of origin, tumor cells that acquire more mutations will be at a selective advantage, which will lead to one or more clonal expansions and the formation of clones, groups of cells with shared genomic profiles. In 1988, Pierce and Speers introduced the CSC hypothesis, suggesting that tumor growth is steered by a subpopulation of cancer stem cells that produces differentiated tumor cells. The CSC hypothesis is appealing since it could help explain heterogeneity (although not clonal heterogeneity) and, if CSCs are rare, it could explain why relapse is common after some patients show signs of remission. The CSC hypothesis would also suggest a different strategy for cancer treatment that specifically targets the CSC population instead of the entire tumor.

1.3. Unraveling tumor heterogeneity using single-cell sequencing

1.3.1. Single-cell sequencing

Until recently, most studies of tumor evolution were done in bulk, using technologies that probe millions of cells. By studying several tumors of the same type across hundreds of patients, it is possible to identify shared mutation patterns and stratify patients by subtypes (Perou et al. 2000; Hicks et al. 2006; Pathare et al. 2009). Another approach is to study the sub-clones present within a tumor by e.g. identifying copy-number variations (CNVs) in bulk tumor samples. To account for heterogeneity, these studies have used techniques such as: **(1)** Laser-capture micro-dissection, to extract different tumor cell subpopulations (Glöckner et al. 2002); **(2)** FACS sorting based on the presence of surface receptors previously associated with certain cell subpopulations (Shipitsin et al. 2007); **(3)** FACS sorting to separate cell subpopulations that exhibit different ploidy (Navin et al. 2010); and **(4)** Deep-sequencing, to identify clones and study population substructure (Campbell et al. 2008).

However, bulk sequencing or microarray analyses cannot generally account for heterogeneity since they average out the signals of the cancer and normal sub-populations present in the tumor. Although there are algorithms that attempt to de-convolve bulk DNA sequencing into sub-populations (Carter et al. 2012b; Gusnanto et al. 2012; Chen et al. 2013; Oesper et al. 2013; Deshwar et al. 2015), some only support one population of cancer cells and one population of normal cells (Carter et al. 2012b; Gusnanto et al. 2012), whereas others do not scale easily beyond 2 to 3 populations (Chen et al. 2013; Oesper et al. 2013). Furthermore, some scenarios are very difficult to de-convolve; for example, a heterozygous mutation in a uniform tumor cannot be distinguished from a homozygous mutation in a tumor where half the cells are cancerous and half are normal, since both scenarios result in the same allele frequency at that locus.

The most accurate approach towards studying heterogeneous tumors is to study the genomic profiles at the level of individual cells. The first single-cell DNA sequencing approach developed for studying human tumor evolution is single-nucleus sequencing (SNS), a single-cell technique pioneered by Wigler and colleagues (Navin et al. 2011).

Briefly, single nuclei are isolated by flow cytometry, followed by whole-genome amplification (WGA) using DOP-PCR and sequencing (Baslan et al. 2012). SNS is typically done at low-coverage (<1X depth of coverage, <10% coverage of the genome), which is enough to detect large-scale copy-number variations.

Recent advances made possible by single-cell sequencing (Shapiro et al. 2013; Navin 2015) suggest that scDNA-seq will likely become an indispensible tool in the near future to study the genomic variability of complex biological systems. However, since the field is still in its early beginnings and the analysis required to identify CNV events is quite complex, it is difficult for most researchers to effectively use these data. To broaden the reach of this new technology, we developed Ginkgo (see **Chapter 2**), a cloud-based platform for the interactive analysis, quality assessment, and visualization of single-cell CNV data.

1.3.2. Whole-genome amplification methods

Another important challenge in using single-cell sequencing is the limited quantity of starting material. A single human diploid cell contains ~6pg of genomic DNA (Milo et al. 2010), yet current sequencing technologies require input material on the order of micrograms. As such, amplifying the whole genome of a cell—Whole Genome Amplification (WGA)—is an essential step during single-cell sequencing. The most commonly used WGA methods for single-cell DNA sequencing employ exponential amplification and are mainly differentiated by whether they employ isothermal conditions, temperature cycling, or a combination of the two approaches (Table 1.1).

The most commonly used isothermal WGA method is Multiple Displacement Amplification (MDA) (Dean et al. 2001). MDA uses random primers, extended by a ϕ29 DNA polymerase. ϕ29 is an enzyme with high levels of strand displacement activity, which leads to branching patterns of exponential amplification. In contrast, Degenerate Oligonucleotide Primed PCR (DOP-PCR) (Telenius et al. 1992) uses PCR-based amplification and degenerate primers for random priming. Although MDA covers a greater fraction of the genome, DOP-PCR generally exhibits more even coverage (Zong et al. 2012). Finally, some WGA approaches, such as MALBAC (Multiple Annealing and Looping Based Amplification Cycles) (Zong et al. 2012), employ both isothermal and thermal cycling conditions. The primers used in MALBAC have a 5′ end with a known fixed sequence of 27nt, such that amplicons form hairpins due to complementary ends, and prevent further amplification. This isothermal step is then followed by several cycles of PCR amplification.

WGA method	Temperature condition	Primers
MDA	Isothermal (Phi29 polymerase)	6nt random
DOP-PCR	Thermal cycling (PCR)	6nt degenerate (3′), fixed (5′)
MALBAC	Both (Bst polymerase + PCR)	8nt random (3′), 27nt fixed (5′)

Table 1.1: Comparison of commonly used Whole-Genome Amplification (WGA) methods.

Despite the effectiveness of WGA, going from picogram to microgram levels of DNA requires a large amount of amplification, which is bound to cause errors. A major source of amplification bias includes random fluctuations during WGA, the uneven levels of GC content, and varying efficiencies of priming and extension along the genome. Errors due to WGA also include allelic drop out and the formation of chimeric DNA, both of which can bias downstream analyses. Since each WGA approach exhibits different classes of errors for different classes of genetic variants, it is important to compare these methods to determine which generally results in better data quality and less bias. We address this question for copy-number alterations in the latter part of **Chapter 2**, where we use Ginkgo to investigate the data quality of the three commonly used WGA approaches listed in Table 1.1, and conclude that DOP-PCR is best suited for copy-number analysis at single-cell resolution.

1.4. Genomic variation at single-cell resolution

Following the study by (Navin et al. 2011), other studies have applied scDNA-seq to study tumor evolution (Leung et al. 2015; Malhotra et al. 2015), circulating tumor cells (Ni et al. 2013; Dago et al. 2014), mosaicism in the brain (McConnell et al. 2013), and identify recombination and crossover sites in sperm and oocytes (Lu et al. 2012; Wang et al. 2012a; Hou et al. 2013; Kirkness et al. 2013). These studies have successfully identified genomic variation in individual cells at the level of single nucleotides (SNPs), and large-scale CNVs at resolutions ranging from 50kb to 1Mb.

For studies of tumor evolution, CNVs constitute an important class of mutations that can be probed via low-coverage single-cell sequencing. In healthy individuals, germline CNVs play a large role in genetic diversity amongst humans and cover a significant fraction of the genome, with estimates ranging from ~1% (Sebat et al. 2004) to ~11% (Redon et al. 2006). In several cancers, somatic CNVs are a key source of alteration that can contribute to cancer initiation and progression (Henrichsen et al. 2009; Shlien and Malkin 2009a). CNVs can cause the amplification and deletion of important cancer genes (or even whole chromosomes), or impact their levels of expression. Although CNVs belong to a larger class of variants termed Structural Variants (SVs), very few studies have investigated copy-neutral SVs such as translocations and inversions at single-cell level (Voet et al. 2013). Due to low coverage and noise, algorithms that use paired-end reads or split-read information to call SVs are not effective. Furthermore, whole genome amplification is also known to create chimeras (Lasken and Stockwell 2007), artificial DNA segments joining together distant regions in the genome that are misinterpreted as SVs. For these reasons, most single-cell sequencing datasets analyzed in this thesis **(Chapter 2, 3, 4)** are performed at low coverage (< 1X), and target large-scale (50kb to 500kb) copy-number alterations.

1.4.1. Algorithms for CNV analysis at single-cell resolution

The genome-wide copy-number state of a healthy human cell is—for the most part—2 copies (ignoring sex chromosomes and inherited CNVs). Copy-number variations are defined as events that amplify or delete the number of copies of a region in the genome. Three major mechanisms have been proposed to explain the formation of CNVs (Redon et al. 2006; Gu et al. 2008; Hastings et al. 2009a; Hastings et al. 2009b; Zhang et al. 2009): **(1)** Non-Allelic Homologous Recombination (NAHR), where two distant stretches of DNA with high sequence similarity undergo crossover during cell division; **(2)** Errors during Non-Homologous End Joining (NHEJ), a process used by the cell to repair double-strand breaks in DNA; and **(3)** Stalling of the DNA replication fork, followed by template switching (also known as FoSTeS).

Before identifying CNVs from scDNA-seq data, the reference human genome must first be sub-divided into bins (e.g. ~50kb or ~500kb bins). Since cells are sequenced at low depth of coverage, piling up reads into large bins reduces the fluctuations present in the signal; at the same time, however, using larger bins reduces the resolution at which CNVs can be identified. Although fixed-size bins are sometimes used in scDNA-seq experiments (Zong et al. 2012; Ni et al. 2013), this approach introduces bias in the analysis. For example, regions of low mappability along the genome will not yield many reads, which could be misinterpreted as deletions. Therefore, the use of variable-size bins that take mappability into account is highly recommended (Navin et al. 2011; Baslan et al. 2012). Once the genome binned, raw sequencing reads from each cell are mapped to the genome (Langmead and Salzberg 2012), placed into bins, and corrected for GC-bias.

Next, the read depth is used to estimate the copy-number state at each bin, assuming that read depth is proportional to copy-number state. For example, if the average bin in a cell contains 100 reads, observing contiguous bins with 200 reads are likely due to an amplification, whereas neighboring bins with 50 reads are likely due to a deletion. In practice, WGA artifacts and noise require a more sophisticated approach. In recent single-cell sequencing data, two major approaches were used to identify CNVs from noisy read-depth profiles: **(1)** Segmentation, where neighboring regions of common read depth are joined (Navin et al. 2011; McConnell et al. 2013); and **(2)** Hidden Markov Models (HMMs), where the hidden states correspond to discrete copy-numbers (Zong et al. 2012; Ni et al. 2013).

1.4.1.1. Circular Binary Segmentation

The most commonly used segmentation algorithm in single-cell studies is Circular Binary Segmentation (CBS) (Olshen et al. 2004; Venkatraman and Olshen 2007), an algorithm that recursively splits chromosomes into segments of equal copy numbers using a t-statistic.

Specifically, CBS is based on the binary segmentation approach of (Sen and Srivastava 1975), an algorithm that uses a t-statistic to test for the presence of a single "change-point", a location along the segment where a shift in the mean value is observed. If a change-point is identified along a segment, the two (hence the 'binary') resulting sub-segments are tested recursively; the recursion ends when no more change-points are detected. However, since the t-statistic only tests for the presence of a single change-point, this approach may fail for more complex patterns observed in CNV profiles, e.g. small events occurring within other CNVs (Olshen et al. 2004).

To address this issue, CBS extends the idea of binary segmentation by considering each chromosome as a circle (hence the name). For a given segment with bin coordinates $[S, E]$, instead of looking for individual change-points, the goal is to identify the pair of change-points i, j that maximize the t-statistic T_{ij}. This statistic tests whether the mean of the values found in bins $[i, j]$ differs from the mean of values in bins $[S, i) \cup (j, E]$, where $S \leq i < j \leq E$; in other words, it tests whether the two arcs of the segment circle defined by i, j have different means. If such change-points exist, the algorithm is applied recursively on the three resulting sub-segments $[S, i), [i, j],$ and $(j, E]$, until no further change-points are identified. Since the data is generally not normally distributed, CBS assesses the significance of the t-statistic $T = \max_{i,j} |T_{ij}|$ by repeating the procedure thousands of times on permuted data and calculating the t-statistic T^* at each iteration, so as to estimate the underlying reference distribution and infer a p-value. Once the CBS segmentation is complete, the segment boundaries across all bins are determined, and the counts for all bins within each segment are set to the median bin count value within that segment.

Although this procedure has a prohibitive $O(N^2)$ running time (N = number of bins) due to the pairwise comparisons, the most recent version of CBS infers p-values in linear time by approximating the tail probability, which the authors have shown to be highly accurate (Venkatraman and Olshen 2007). In Table 1.2 and Figure 1.1, we briefly investigate the speed of CBS at commonly used bin sizes, using scDNA-seq data from a cancer cell sequenced at ~4.1M reads (Navin et al. 2011). For each binning scheme, we bin the reads, perform GC correction, and estimate the average running time of CBS in R over 100 iterations. As expected, the running time grows linearly with increasing number of bins (or decreasing bin size).

Average bin size	Number of bins	Running Time
800kb	3,337	1.70 ± 0.008
500kb	5,363	1.90 ± 0.009
200kb	13,466	2.83 ± 0.013
100kb	26,970	4.79 ± 0.023
50kb	53,977	10.14 ± 0.176
25kb	107,995	17.75 ± 0.046

Table 1.2: Average running time of CBS as a function of bin size (using a variable-sized binning strategy).

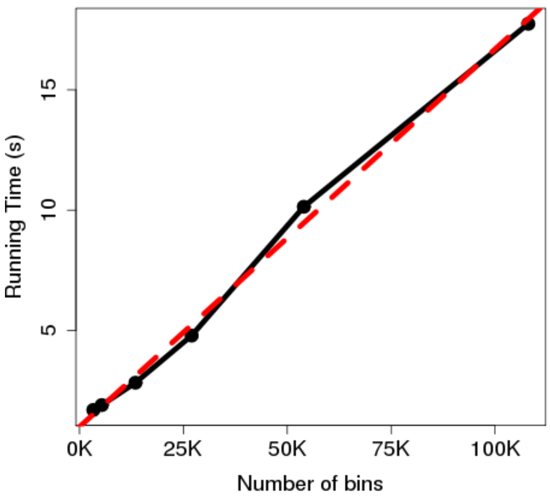


Figure 1.1: The running time of CBS for one cell (black line) grows linearly with number of bins. The line of best fit is plotted in red.

1.4.1.2. Hidden Markov Models

Another commonly used approach for single-cell CNV calling is the use of HMMs. As discussed in (Zong et al. 2012), both cancer cells and known normal cells are first normalized by their total read depth, and each cancer cell is described as a binary vector of size N (number of bins), where “1” indicates that the cancer cell has higher coverage than the normal cell and “0” indicates otherwise. Using a HMM with three hidden states (diploid, amplification, deletion) and the binary vectors as the observed sequence, the Viterbi algorithm is used to infer the state path. To obtain integer copy-number calls, a second HMM with 6 hidden states (0, 1, 2, 3, 4, 5 copy number) is used. For each cancer cell, the number of reads per bin is normalized by that of a normal cell, and fed in to the HMM as the observed sequence. Besides CBS and HMMs, (Daruwala et al. 2004) have also proposed a segmentation approach that uses a Maximum a posteriori (MAP) estimation, a Bayesian approach similar to Maximum Likelihood that includes a prior distribution.

1.4.1.3. CBS vs. HMMs

Although the HMM approach above has been used in several single-cell sequencing studies (Zong et al. 2012; Hou et al. 2013; Ni et al. 2013), this approach has several drawbacks. As used in these studies, the HMMs are typically modeled such that copy-number alterations of >5 copies cannot be not detected accurately. This is an issue for cancers where very high levels of copy-number are observed, e.g. in a single-cell sequencing study of a triple negative breast cancer patient (Navin et al. 2011), several cells were observed to share a 50-fold amplification of KRAS, an important oncogene. Furthermore, the HMM approach above requires a transition matrix built using *a priori* estimates of the expected rate per bin of copy-number aberration start and end (which in (Zong et al. 2012) are arbitrarily specified as 0.01 and 0.1 respectively).

Since CBS requires fewer assumptions, and performs better or as well as other approaches (Lai et al. 2005; Willenbrock and Fridlyand 2005; Knouse et al. 2016), we make use of CBS for CNV calling in this thesis.

1.4.2. Optimizing scDNA-seq for CNV analysis and tumor heterogeneity

From a practical standpoint, a major drawback of single-cell sequencing is the prohibitive cost incurred for sequencing a large number of cells. Recently, through improvements to WGA, library preparation and the use of multiplexed sequencing, (Baslan et al. 2015) present an optimized approach that reduces costs down to $30 per cell, and yields accurate CNV profiling. Despite these advances, scDNA-seq remains expensive for sequencing thousands of cells.

Given a fixed budget, there exists a trade-off between the number of cells that can be sequenced and the depth of coverage at which each cell is sequenced, and it is unclear which approach yields greater biological insight into tumor heterogeneity and population structure. Although guidelines exist to guide investigators to better decide which sequencing depth if appropriate for bulk sequencing of cancer samples (Griffith et al. 2015), no such guidelines exist for single-cell CNV analysis.

In **Chapter 3**, we use simulations and statistical analyses to help drive experimental design choices. Using single-cell sequencing data from 3,446 cells, we explore the space of sequencing parameters for CNV analysis at single-cell resolution. Using millions of *in silico* sub-sampling experiments, we identify the critical read depth thresholds needed to ensure accurate **(1)** Reconstruction of CNV profiles; **(2)** Inference of phylogeny from tumor cells; and **(3)** Identification of known clonal groups of cells. Applying these simulations on varying levels of read depth and cells, we further **(4)** Evaluate the tradeoffs between sequencing at lower depth and sequencing more cells.

We find that the optimal strategy is to sequence as many cells as possible, but at very low depth of coverage—especially for very heterogeneous tissues. The guidelines we highlight allow us to accurately identify the major features of a sample's population structure at a first pass, while capturing as much of its heterogeneity as possible. Subsequent studies can then target sub-populations of interest and examine them at greater depth.

Another challenge in single-cell sequencing is that noisy CNV data can hinder the dissection of tumor heterogeneity at single-cell resolution due to the presence of signal from spurious breakpoints. To address some of these issues, **Chapter 4** introduces an approach for identifying the informative CNV breakpoints that define clones of cells with shared genomic events. In tumors characterized by only one small clone, we demonstrate enhanced power to hone in on these clones. In tumors with several sub-clones, we introduce an approach that enhances the reconstruction of the population structure.

1.5. Comparative genomics of genome topology

1.5.1. Co-localization of gene clusters

Section §1.5.1 has been reproduced with modifications from:

> Aboukhalil R, Fendler B, and Atwal GS. **Kerfuffle: a web tool for multi-species gene colocalization analysis**. *BMC Bioinformatics* 14:22 (2013).

Shifting the focus from somatic to germline evolution of cancer, advances in genomics and DNA sequencing technology have fueled growing interest in the large-scale physical and functional organization of chromosomes. Several studies have shown that genomes of many disparate species may have chromosome regions containing clusters of functionally related genes (Hurst et al. 2004a; Petkov et al. 2005a; Xue et al. 2012b). It is well known that operons, ubiquitous in prokaryotes, allow multiple genes to be transcribed at once into a polycistronic mRNA. The extent to which genes co-localize in eukaryotes and the extent to which gene clusters are conserved across species are largely unknown. In eukaryotes, operons are rare (Blumenthal 2004); however, there is evidence to suggest that genes within the same biological pathway may be clustered more so than expected by random rearrangements, possibly because of co-regulation (Lee and Sonnhammer 2003). For example, the *Hox* genes are tandem duplicate genes organized into clusters, playing a pivotal role in defining the body plan of organisms. Further, the order of the genes within a *Hox* cluster defines the sequence in which these genes are expressed (Carroll 1995). While these examples rely on positional clustering, other mechanisms may also lead to gene clusters. For example, clustered genes could be co-regulated because **(1)** their promoters are bound to by the same transcription factors; **(2)** they share regulatory elements such as bidirectional promoters (Trinklein et al. 2004); and **(3)** the transcription of a gene can change local chromatin accessibility for its neighbors.

Between evolutionary distinct species, we expect to find random genomic rearrangements that do not conserve gene clusters, unless co-localization is beneficial to the organism. It is possible that co-localization is acted upon by natural selection, conserving the gene clusters across large evolutionary time scales, although it remains unclear what structural, regulatory, and functional factors are responsible for the co-localization (Lercher et al. 2002; Hurst et al. 2004a; Singer et al. 2005). A recent study found that the genome of a number of different species was arranged into neighborhoods of functionally-related genes that were not necessarily orthologous (Al-Shahrour et al. 2010). If functionally related genes cluster for mechanistic purposes, then it is expected that those clustered genes would co-localize in other species as well. However, few of the bioinformatics tools currently available allow for a systematic study of gene co-localization across several, evolutionarily distant species. Furthermore, most tools require the user to input manually curated lists of gene position information, DNA sequences or gene homology relations between species. With the growing number of sequenced genomes, there is a need to provide new comparative genomics tools that can address the analysis of multi-species gene co-localization.

In **Chapter 5**, we introduce Kerfuffle, a web tool designed to help discover, visualize, and quantify the physical organization of genomes by identifying significant gene co-localization and conservation across the assembled genomes of available species (currently up to 47, from humans to worms). Kerfuffle only requires the user to specify a list of human genes and the names of other species of interest. Without further input from the user, the software queries the Ensembl BioMart server to obtain positional information and discovers homology relations in all genes and species specified. Using this information, Kerfuffle performs a multi-species clustering analysis, presents downloadable lists of clustered genes, performs Monte Carlo statistical significance calculations, estimates the level of conservation of gene clusters across species, plots histograms and interactive graphs, allows users to save their queries, and generates a downloadable visualization of the clusters using the Circos software. These analyses may be used to further explore the functional roles of gene clusters by interrogating the enriched molecular pathways associated with each cluster.

1.5.2. Synteny of tumor suppressor genes

Section §1.5.2 has been reproduced with modifications from:

> Fendler B* and Aboukhalil R*, Xue R, Esposito D, Powers S, Lowe SW and Atwal GS. **Tumor Suppressive Genes are Conserved in Syntenic Clusters.** *In preparation*.

Of particular interest for cancer genomics is the collection of tumor suppressor genes (TSGs), which are central to our understanding of human tumorigenesis. Currently, our knowledge of the physical distribution of TSGs throughout the genome, and the implications this has for tumor development, is not well developed. Tumor suppressor genes (TSGs) play a pivotal role in preventing tumorigenesis, as their partial or complete inactivation through germline or somatic mutations contributes to human cancer. Since the identification of RB1, in 1986 (Friend et al. 1986), many other TSGs have since been identified, adding to a growing list of genes that sustain loss-of-function mutations in tumorigenesis. Despite ongoing efforts to identify these genes, little attention has been paid to their physical organization in the genome and the functional constraints imposed upon their order. This lack of understanding is conspicuous in light of observations that genetic lesions in cancer frequently involve large genomic deletions that span many contiguous genes (Beroukhim et al. 2010a). Interestingly, anecdotal evidence of human loci containing clustered TSGs have been reported in current topological investigations (Zender et al. 2008a; Scuoppo et al. 2012), including a study that demonstrates direct evidence that the coattenuation of the genes in the syntenic Ink4a/Arf locus has a profound cooperative effect on tumorigenesis in mice (Krimpenfort et al. 2007).

Consistent with these results, Solimini et al. demonstrated evidence, from array CGH (comparative genomic hybridization) data, for the so-called "cancer gene-island model," in which recurrently deleted regions of the genome are enriched in growth preventative genes (Solimini et al. 2012). Further, recent arguments have been made for an increased role of happloinsufficiencies in tumorigenic growth (Solimini et al. 2012; Xue et al. 2012a). In particular, Xue et al demonstrated that the *8p* locus, which contains the putative *DLC1* TSG, harbors other neighboring candidate TSGs. Subsequent RNAi knockdown in a hepatocellular carcinoma model demonstrated an increase in growth and validation of happloinsufficient candidates. In light of these investigations, if physically linked TSGs are common across the genome, then an increased role for happloinsufficiencies in cancer confers the possibility of tumorigenic susceptibility due to the increased likelihood of altering multiple genes in large deletions.

Over the last decade, several studies have demonstrated, across many eukaryotic species, that genomes contain chromosomal regions in which functionally related genes physically cluster (Hurst et al. 2004b; Petkov et al. 2005b). While it is well known that operons, ubiquitous in prokaryotes, allow multiple genes to be transcribed at once into a polycistronic mRNA, operons are rare in eukaryote (Blumenthal 2004). However, there is evidence to suggest that genes within the same biological pathway may be clustered more so than expected by random rearrangements, possibly because of co-regulation (Akashi et al. 2003), although the extent to which genes co-localize in eukaryotes is largely unknown. Combining evidence for clusters of functionally related genes, along with anecdotal evidence of clustered TSGs, and the implications this has for tumorigenesis, there is a pressing need to investigate TSG topology.

In **Chapter 6**, we use Kerfuffle to perform a co-localization analysis of known human TSGs. To explore whether selection forces are at play to maintain these clusters across evolutionary time-scales, we carry out a comparative genomics co-localization analysis across 46 different species, ranging from worms to mammals. We find that neighboring TSGs co-localize in syntenic clusters. Overall, our results demonstrate that the conserved germline evolution of the physical distribution of TSGs has constrained the physical organization of the cancer genome and bears significantly on the risk of cancer development.

CHAPTER 2

Ginkgo: Interactive analysis and assessment of single-cell copy-number variations

This chapter has been reproduced with modifications from:

> Garvin T* and Aboukhalil R*, Kendall J, Baslan T, Atwal GS, Hicks J, Wigler M, Schatz MC. **Interactive analysis and assessment of single-cell copy-number variations.** *Nature Methods* 12: 1058-1060 (2015).

2.1. Introduction

Single-cell DNA sequencing (scDNA-seq) is a powerful tool for probing complex biological systems, and has previously been used to unravel the population structure of heterogeneous tumors (Navin et al. 2011), study the genomic profiles of rare circulating tumor cells (Ni et al. 2013; Dago et al. 2014), identify mosaicism in the brain (McConnell et al. 2013), and detect genome-wide recombination/crossover sites in gametes (Lu et al. 2012; Wang et al. 2012a; Hou et al. 2013; Kirkness et al. 2013). For these applications, bulk sequencing is inadequate since it averages out the signal over millions of cells. One important application of scDNA-seq is to identify large-scale copy-number variations (CNVs), which play important roles in several cancers (Shlien and Malkin 2009b) and neurological disorders (Malhotra and Sebat 2012).

Given the insights made possible by single-cell sequencing, many researchers are now interested in applying the technology to study diverse biological systems and species. However, the downstream analysis is complex. Although many approaches and computational tools exist for CNV analysis of bulk samples (Alkan et al. 2011) there are currently no fully automated tools that address the unique challenges of single-cell sequencing data: **(1)** extremely low depth of sequencing coverage (< 1X) makes for noisy profiles and makes split-read, paired-end, or SNP density approaches ineffective; **(2)** whole-genome amplification (WGA) biases markedly distort read counts, including failure to amplify entire segments (Baslan et al. 2012); **(3)** badly assembled regions of the genome (e.g. centromeres) lead to the artificial inflation of read counts ("bad bins") (Baslan et al. 2012); **(4)** the need for new algorithms for calling copy numbers at single-cell, integer levels; and **(5)** the fact that current tools for exploring population structure are not built for single-cell data. In addition, several sources of cell-specific experimental errors, including GC content and other sequencing biases, need to be addressed. While ad hoc methods have been developed for individual studies, there is currently no easy-to-use, open-source software available to execute this pipeline automatically.

Here we introduce Ginkgo, a suite of software tools for the interactive analysis and quality assessment of single-cell copy-number alterations. Ginkgo automates and standardizes the computation required to go from mapped reads to copy-number profiles of individual cells, to phylogenetic trees of cell populations. Ginkgo also enables users to navigate within a cell's copy number profile, zoom into regions of interest, annotate profiles and export tracks to the UCSC browser for further inspection. Ginkgo is available online as a web application at http://qb.cshl.edu/ginkgo, and as a stand-alone software package at http://github.com/robertaboukhalil/ginkgo.

To validate Ginkgo, we reproduce the major findings of five recent single-cell studies. These datasets address vastly different scientific questions, were collected from a variety of tissue types, and make use of different experimental and computational approaches at different institutions. Next, we use Ginkgo's quality assessment tools to examine the data characteristics of three commonly used single-cell amplification techniques (MDA, MALBAC, and DOP-PCR) through comparative analysis of 9 different single-cell datasets. We find that both MALBAC and DOP-PCR outperform MDA in terms of data quality. As previously reported, MDA displays poor coverage uniformity and low signal-to-noise ratios. Coupled with high GC biases, MDA is unreliable for accurately determining CNVs compared to the other two techniques. Furthermore, while both DOP-PCR and MALBAC data can be used to generate CNV profiles and identify large variants, we find that DOP-PCR data exhibits lower coverage dispersion and smaller GC biases when compared to MALBAC data. Given the same level of coverage, our results indicate that data prepared using DOP-PCR can reliably call CNVs at higher resolution with better signal-to-noise ratios.

2.2. Results

2.2.1. Ginkgo: an interactive software suite for single-cell CNV analysis

Ginkgo's user-friendly web interface guides users through every aspect of the analysis, from uploading data to visualization and exploration of the single-cell copy-number profiles (Figure 2.1). Ginkgo takes, as input, mapped sequencing reads in the form of tab-separated .BED files, one for each cell to be analyzed (Figure 2.2A). Each .BED file contains mapping information about the reads from that cell, including chromosome number and nucleotide position. Although the more standard .BAM file format could in principle be supported, Ginkgo requires .BED files since they contain only the information required for the downstream read depth analysis and copy-number calling. As such, they are much more condensed (~5-10X smaller in size when gzip-compressed), which greatly speeds up the uploading process and reduces the burden on the Ginkgo servers.

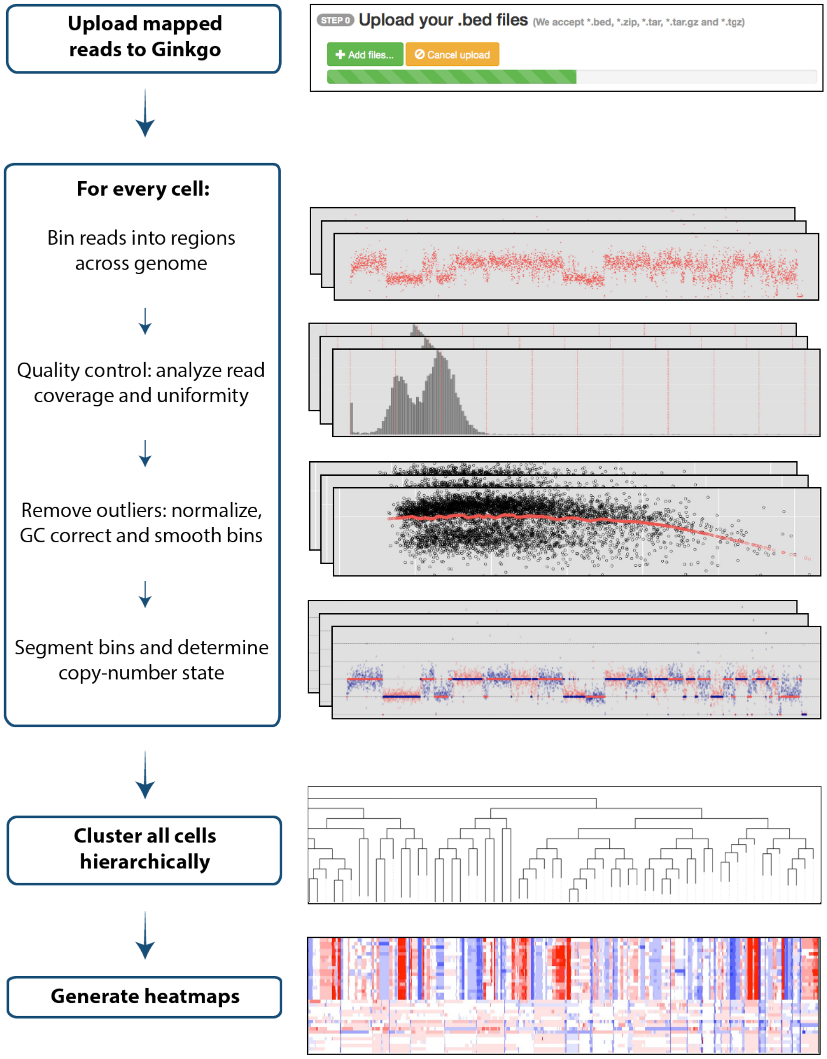

Upload mapped reads to Ginkgo
STEP 0 Upload your .bed files (We accept *.bed, *.zip, *.tar, *.tar.gz and *.tgz)
Add files...
Cancel upload
For every cell:
Bin reads into regions across genome
Quality control: analyze read coverage and uniformity
Remove outliers: normalize, GC correct and smooth bins
Segment bins and determine copy-number state
Cluster all cells hierarchically
Generate heatmaps

Figure 2.1: The Ginkgo flowchart for single-cell copy-number analysis. Starting from mapped sequencing reads, Ginkgo places the reads into variable-sized bins along the genome, and performs GC correction. Following segmentation of the copy-number profiles, Ginkgo generates phylogenetic trees and heatmaps to help elucidate population structure.

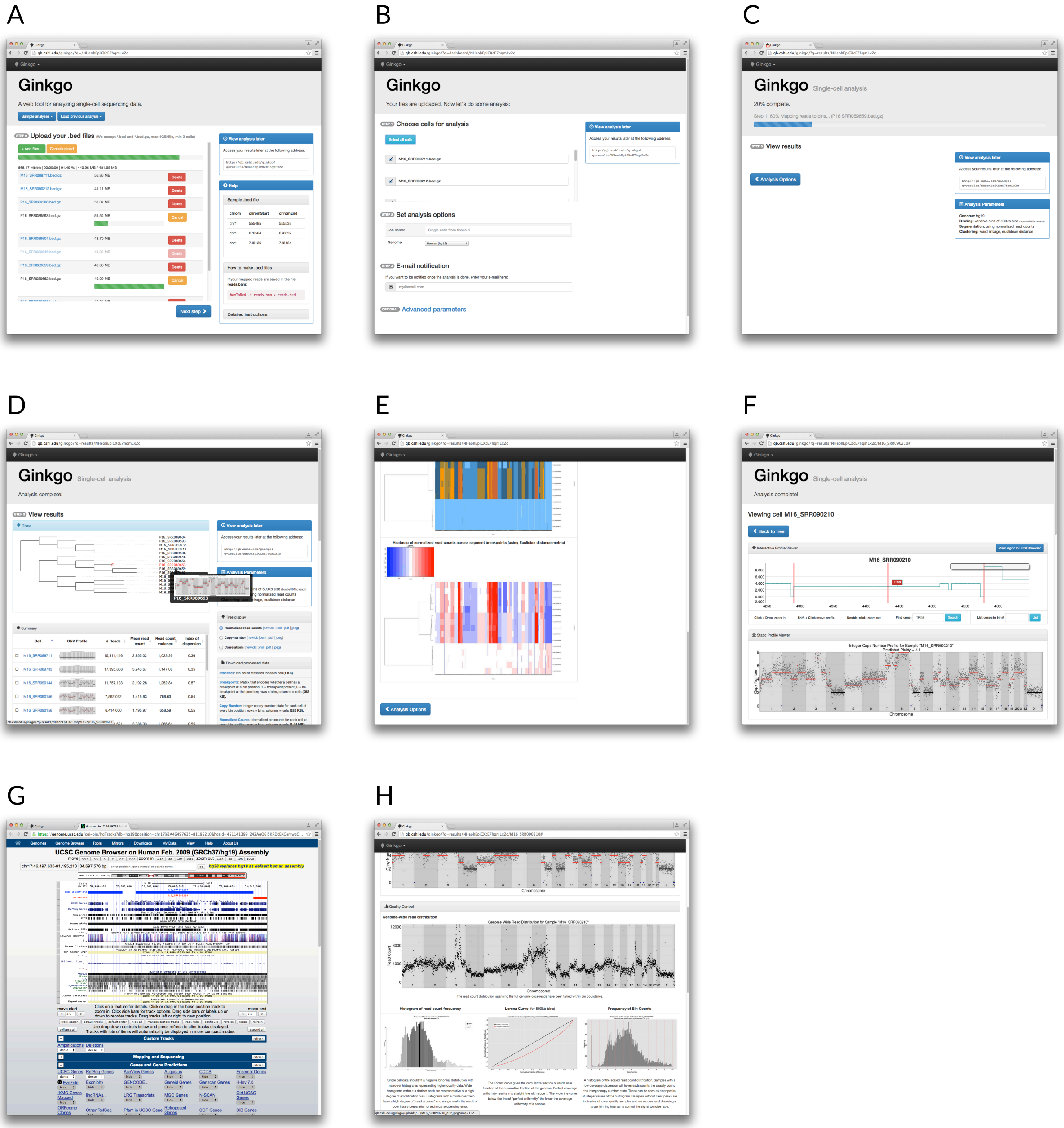

A
B
C
D
E
F
G
H
Ginkgo
A web tool for analyzing single-cell sequencing data.
Upload your .bed files
Next step
Your files are uploaded. Now let's do some analysis:
Choose cells for analysis
Set analysis options
E-mail notification
Advanced parameters
View analysis later
Help
Single-cell analysis
20% complete.
View results
Analysis Options
Analysis Parameters
Analysis complete!
Tree
Summary
Heatmap of normalized read counts across segment breakpoints (using Euclidean distance metric)
Viewing cell M16_SRR090210
Back to tree
M16_SRR090210
Static Profile Viewer
Chromosome
UCSC Genome Browser on Human Feb. 2009 (GRCh37/hg19) Assembly
Custom Tracks
Mapping and Sequencing
Genes and Gene Predictions
Quality Control
Genome-wide read distribution
Histogram of read count frequency
Lorenz Curve
Frequency of Bin Counts

Figure 2.2: Screenshots of the Ginkgo software, illustrating major steps in the analysis and visualization. (A) Users are asked to upload mapped sequencing read data in .BED (or .BED.GZ) format. **(B)** Many parameters of the analysis can be tweaked as necessary. **(C)** Once the analysis launched, Ginkgo will inform the user of the progress in real-time. **(D-E)** Once the analysis complete, Ginkgo provides tools to view the results at a glance, including a phylogenetic tree and heatmaps. **(F-H)** Each cell has a dedicated page with information about its copy-number profile and quality control graphs, with links to automatically export amplification/deletion tracks to the UCSC Genome Browser for further inspection.

Once a user selects analysis parameters (Figure 2.2B), sequencing reads from each cell are binned by chromosome position, normalized for GC bias and other amplification artifacts (Methods), and segmented to identify chromosome regions with consistent copy-number states using the Circular Binary Segmentation algorithm (Olshen et al. 2004; Venkatraman and Olshen 2007); Methods). Integer copy-number state is assigned to each segment, which allows Ginkgo to build phylogenetic trees and heat maps from the copy-number or breakpoint profiles of the collection of cells. Throughout the analysis, Ginkgo displays the progress of each step (Figure 2.2C). Since the analysis may take a few hours depending on the number of cells and sequencing depth, the user can also choose to be notified by e-mail once the analysis is done (at a depth of 2M reads and 500kb bin resolution, the analysis generally requires ~20s per cell).

Once the analysis is complete, Ginkgo displays an overview of the data in a sortable data table, an interactive phylogenetic tree (Smits and Ouverney 2010) of all cells used in the analysis and a set of heat maps detailing the CNVs that drove the clustering results (Figure 2.2D-E). Clicking on a cell in the phylogenetic tree or data table allows the user to view an interactive plot of the genome-wide copy-number profile of that cell (Figure 2.2F), search for genes of interest and link out to a custom track of amplifications and deletions in the UCSC genome browser (Figure 2.2G). Ginkgo also outputs several quality-assessment graphs for each cell (Figure 2.2H): a plot of read distribution across the genome, a histogram of read-count frequency per bin and a Lorenz curve for assessing coverage uniformity (Zong et al. 2012). The Lorenz curve is obtained by sorting bin counts from lowest to highest and plotting the cumulative fraction of reads as a function of the cumulative fraction of the genome covered by these reads. Subsets of cells can also be selected by the user for direct comparison of copy-number profiles, Lorenz curves, GC bias and coverage dispersion.

All plots, statistical measurements and clustering results can be downloaded in publication-quality figures or as tab-delimited text files. The results are saved on our servers for several months, allowing the user to return to their results at a future date and run different analyses with the same data. A unique URL is generated for each project, allowing researchers to easily share the displays with collaborators of their choosing while maintaining security of their data. Alternatively, we provide and document all the software necessary for hosting the web tool on a local server for extended analysis using Docker containers (Methods).

2.2.2. Ginkgo reproduces the results of previous single-cell studies

To validate Ginkgo, we set out to reproduce the major findings of several recent single-cell sequencing studies (Navin et al. 2011; Lu et al. 2012; Hou et al. 2013; McConnell et al. 2013; Ni et al. 2013). These studies address vastly different scientific questions and originate from a variety of tissue types: breast tumors, lung circulating tumor cells, neurons, sperm and oocytes. Furthermore, these conclusions reported in these studies were obtained using different computational approaches (HMMs vs. segmentation approaches), and different whole-genome amplification methods: MDA (Dean et al. 2001), MALBAC (Zong et al. 2012) and DOP-PCR (Telenius et al. 1992; Blainey 2013). Using Ginkgo, we replicate most published CNVs, with the exception of one cell from a study by (Hou et al. 2013). We believe that this failed replication was due to mislabeling in the National Center for Biotechnology Information (NCBI) Sequence Read Archive (SRA). Moreover, as shown below, we successfully reproduce the distinct clonal subpopulations in the two data sets from (Navin et al. 2011) and the patient clustering results from (Ni et al. 2013) that were generated from inferred CNVs.

2.2.2.1. *Navin et al.*

This work profiled breast cancer in two separate studies. The first (dataset T10) examined heterogeneity in a polygenomic breast tumor. CNV analysis and hierarchical clustering of 100 single-cells revealed three distinct clonal subpopulations present in the tumor. The second study (datasets T16M/P) examined a monogenomic breast tumor and its suspected liver metastasis. CNV analysis and hierarchical clustering of 100 cells revealed that a single clonal expansion formed the primary breast tumor and seeded the metastasis. In the polygenomic breast tumor analysis, Ginkgo clusters all 100 samples into the same four distinct subpopulations of the original study, replicating the published population structure (Figure 2.3A). In the monogenic breast tumor and its associated liver metastasis analysis, Ginkgo clusters all 100 samples into the same three distinct subpopulations as the original publication, linking the primary tumor to its metastasis (Figure 2.3B).

2.2.2.2. *McConnell et al.*

This study profiled CNV events in human hiPSC-derived fibroblasts and 110 frontal cortex neurons. McConnell *et al.* found a wide degree of mosaic copy-number variation in neurons and discovered that a subset of neurons have highly aberrant genomes. McConnell *et al.* identified a total of 148 CNVs across 45 of the 110 sequenced cortical neurons. They further present detailed information for the 148 CNV calls, including their genomic coordinates, the copy number assignments of the CNVs expressed as the median of the segment values, the genome-wide median segment value of diploid regions, and the median absolute deviation (MAD) score of the calls.

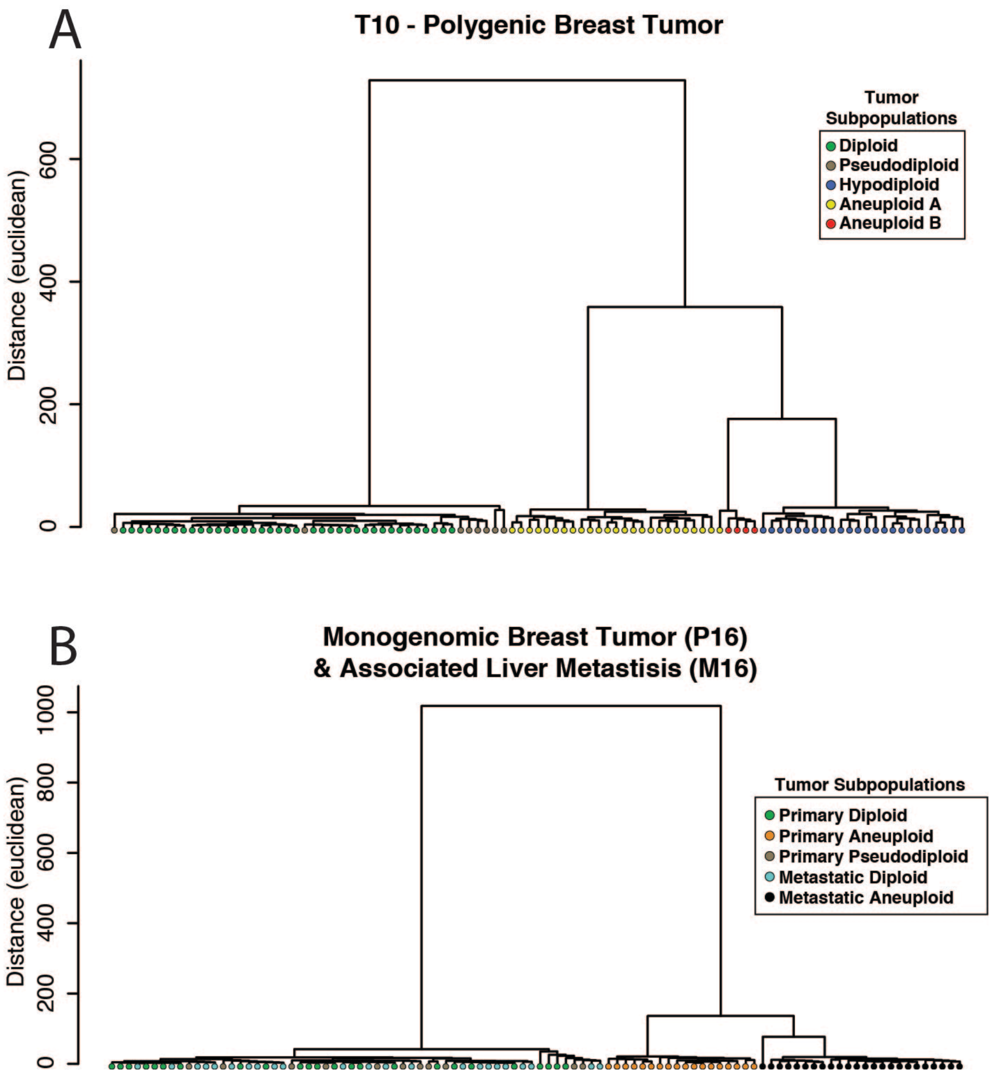


Figure 2.3: Phylogenetic trees generated through hierarchical clustering by copy-number using (A) 100 polygenomic breast tumor samples (T10) and **(B)** 52 monogenomic breast tumor (T16P) and 48 liver metastasis (T16M) samples. These results match the clonal structure published in the original study.

Using this information, we investigated the concordance between the CNVs Ginkgo reports to those reported by McConnell. To do so, we matched the parameters used by McConnell as closely as possible by using 500kbp variable length bins and requiring a minimum of 6 bins for a CNV (note that McConnell used a minimum of 5 bins for a CNV, but their bin size was on average 686kb wide after accounting for mappability). We do not expect perfect concordance, as the two methods use different strategies and technical choices for identifying CNVs such as different strategies for mappability, normalization, and thresholding CNVs. Nevertheless, we find that the concordance is extremely high, with 99.7% bin-level concordance with Ginkgo reporting 127 (85.8%) of the 148 CNVs identified by McConnell plus 116 additional CNV calls. McConnell et al. identifies CNVs whenever the median segment value is more than 2 MAD scores above or below the genome wide median. We investigated this relationship and found very strong correlation (R^2=0.996) between Ginkgo's median segment values and McConnell's over these regions (Figure 2.4). Note that the bin boundaries do not exactly coincide, due to different binning strategies and techniques to account for mappability, which introduces some variability in the segment coordinates. We investigated the most incongruent of those segments and observed that they were at the very beginning or very end of chromosomes in highly repetitive telomeric sequences (Figure 2.5 top). This suggests the differences were largely due to the details of how the reads were mapped and the bin boundaries were determined, especially since McConnell used default BWA parameters, while Ginkgo aggressively controls for multi-mapping reads and quality scores in the analysis. In other cases, the median segment values of the discordant calls were virtually indistinguishable and yet not classified by Ginkgo as a CNV (Figure 2.5 bottom). In particular, Ginkgo and McConnell agree on all CNV calls when their segment MAD calls are greater than 2.35, but below that cutoff there are slight variations depending on the specific context of the segment (Figure 2.6). Finally, we speculate the additional 116 calls made by Ginkgo were just below the McConnell's thresholds for reporting a CNV (slightly below a MAD of 2.0), although the data are not available to directly compare.

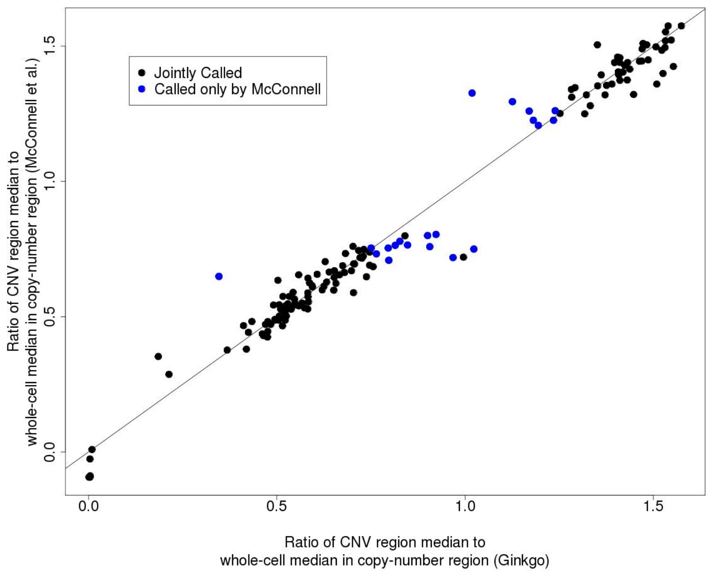


Figure 2.4: Comparison of segment medians between Ginkgo and McConnell. Scatter plot showing correlation between events called by McConnell et al. and Ginkgo. Each data point represents, for a given CNV region in McConnell et al., the ratio of the segment median in that region to the segment median of the entire cell. Points labeled in blue correspond to the 20 segments that were called as CNVs by McConnell but called diploid by Ginkgo.

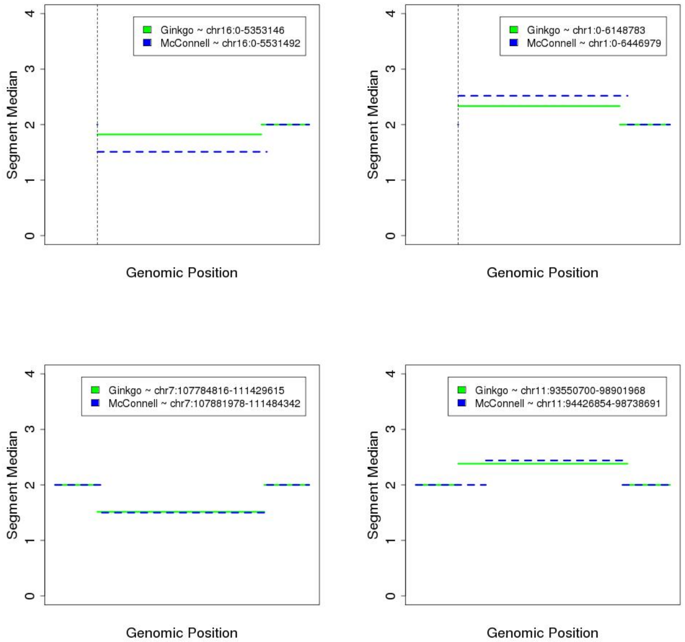


Figure 2.5: Comparing discordant segments between Ginkgo and McConnell. (Top) The two regions with the most discordant median segment values that were called by McConnell but not Ginkgo. These CNVs are located at the start of chromosomes 1 and 16, in highly repetitive telomeric regions. **(Bottom)** The two regions with the most concordant median segment values that were called by McConnell but not Ginkgo. Although small differences are clearly detected by Ginkgo in these regions, they are not marked as copy-number events due to differences between Ginkgo and McConnell's CNV calling thresholds.

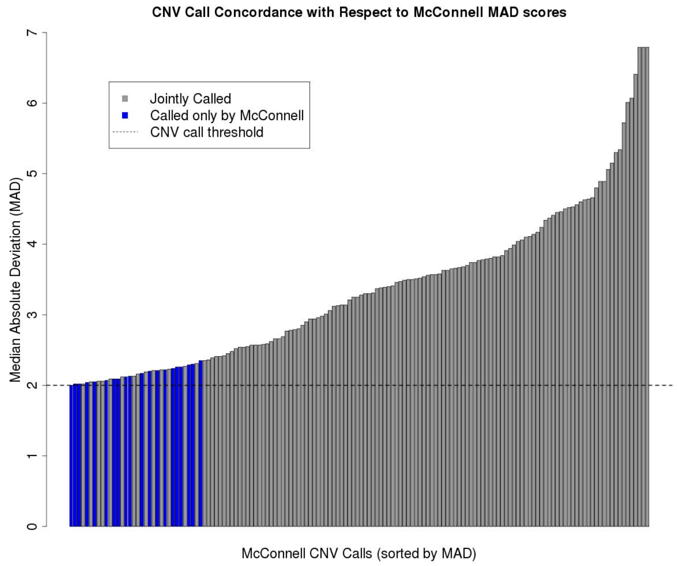


Figure 2.6: An analysis of discordant calls with respect to McConnell MAD values. The 148 segments called by McConnell as CNVs rank-sorted by their MAD values. All 128 segments that are also called as CNVs by Ginkgo have higher MAD values (>2.35). All of the calls made exclusively by McConnell (colored in blue) lie right near their threshold for detection.

2.2.2.3. *Ni et al.*

This study explored SNPs and CNVs in circulating tumor cells (CTCs) in patients with lung cancer. Through CNV analysis and hierarchical clustering of 29 CTCs across 7 patients with lung adenocarcinoma (ADC) or small-cell lung cancer (SCLC), Ni et al. discovered that CNVs appear specific to cancer types and are reproducible from cell to cell and from patient to patient. Using default settings in Ginkgo, we generate similar CN profiles for all 29 samples and can reproduce the published clustering results (Figure 2.7A). However, careful consideration of gender must be given when analyzing patients from mixed populations, as the combined set of the X and Y-chromosomes make up a large fraction of the human genome that can distort the clustering results. When we examined the Ni *et al*. dataset with Ginkgo with sex chromosomes masked, we could still discriminate between individual patient's tumors, but we could no longer discriminate between ADC and SCLC (Figure

2.7B); the SCLC patients were exclusively female and ADC patients were almost entirely male. Ginkgo comes prepackaged with the ability to mask sex chromosomes to prevent gender biases from dominating the clustering.

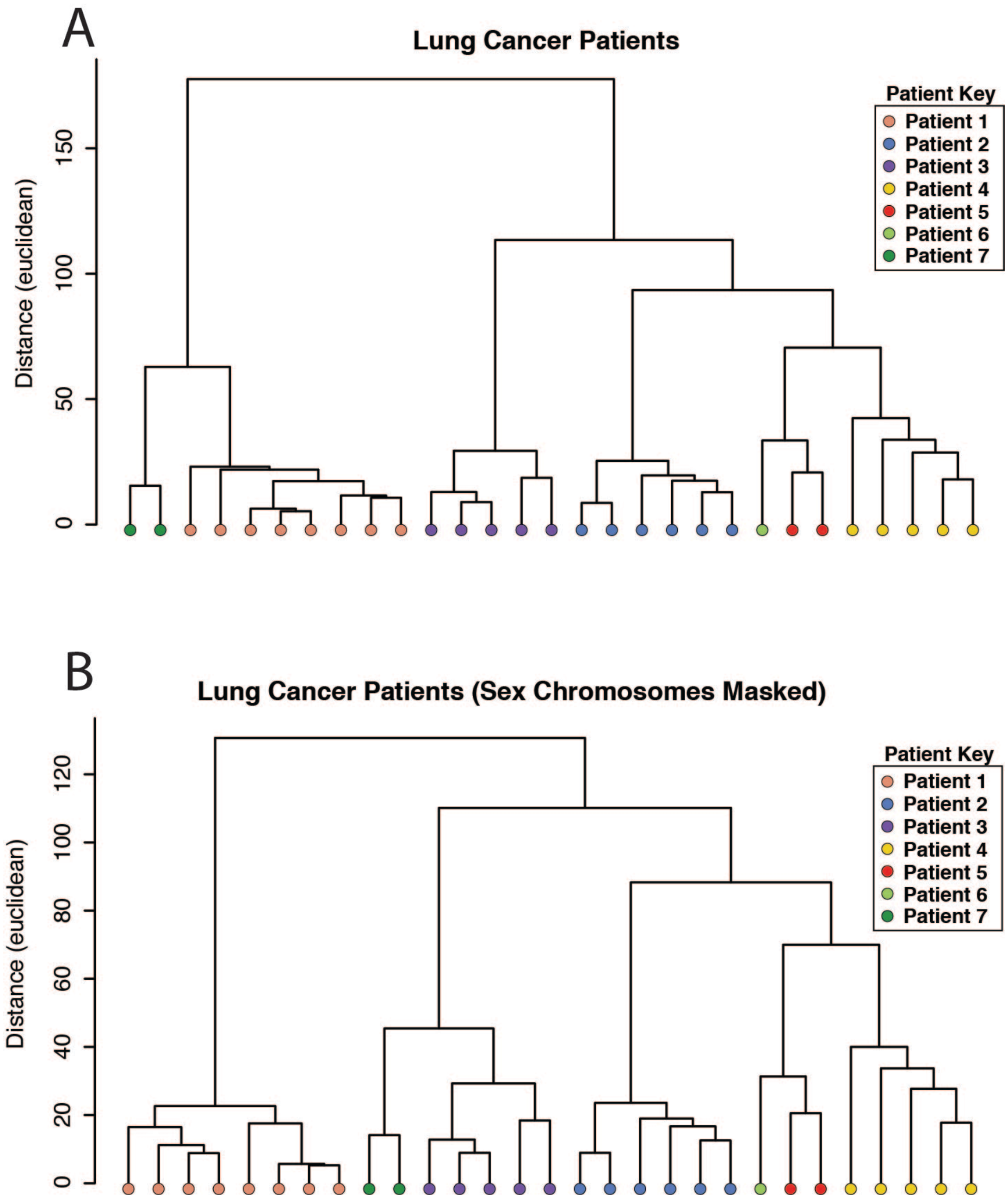


Figure 2.7: Ginkgo patient clustering of Ni et al. CTC data. (A). Hierarchical clustering by Ginkgo of 29 samples derived from 7 different patients with either adenocarcinoma (patients 2-6) or small cell lung cancer (patients 1, 7), matching the results published by Ni et al. **(B)** When sex chromosomes are masked, cells still cluster by patient, but patients no longer cluster by cancer subtype. In particular, after masking sex chromosomes, patient 3 is intermixed between patients 1 and 7 and there is no clear association between cancer types.

2.2.2.4. Hou et al.

This study sequenced several oocytes in order to phase their genomes and determine their crossover maps and frequency. Additionally, genome-wide CN profiles were generated to explore aneuploidy in each sample. The authors identified a total of 47 CNVs in 25 aneuploid cells across 5 patients. We could replicate these results as Ginkgo uncovered 45 of the 47 CNVs in 23 of the 25 identified aneuploid cells. One sample, S0808 (containing the missing two cells/CNVs), did not have CNV events matching the published results. We believe this was due to accidental mislabeling of sample IDs upon being deposited to NCBI.

2.2.2.5. Lu et al.

In this study, single-cell sequencing was used to study meiotic recombination and aneuploidy in 99 sperm cells from an Asian male; of the 99 cells, 5 were aneuploid. Using Ginkgo, we uncovered the same chromosomal aberrations in the 5 aneuploid cells, and successfully separated the X- and Y- bearing chromosomes (Figure 2.8), with the exception of 2 cells that clustered separately due to poor coverage uniformity and high read drop-out.

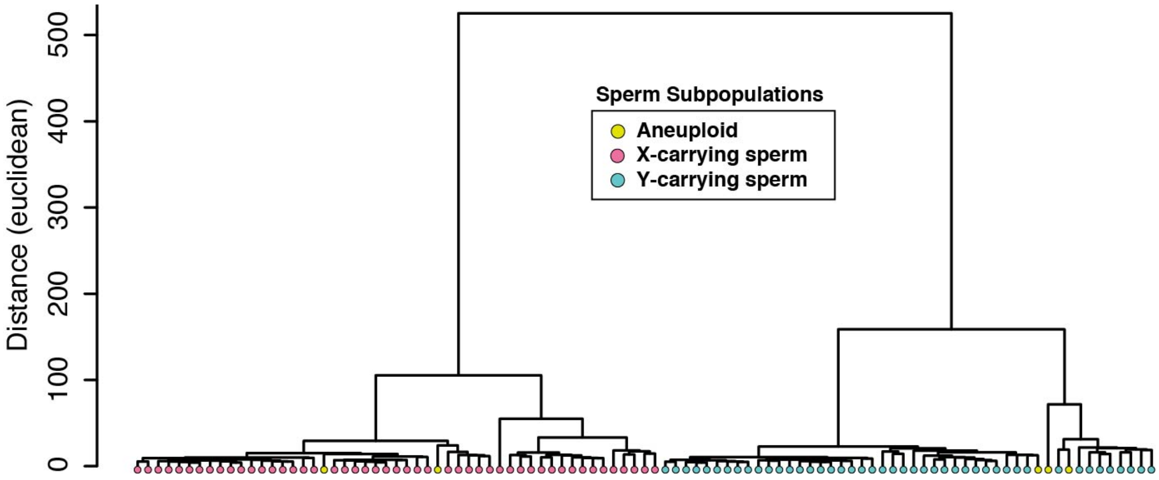


Figure 2.8: Ginkgo clusters the Lu et al. sperm samples. The major populations are defined by X- and Y-carrying sperm. Ginkgo identifies the same variants found by the original study in 5 aneuploid cells (in yellow).

2.2.2.6. *Simulations*

To further test the accuracy of the copy number and clustering analysis by Ginkgo, we simulated single cell sequencing of 90 cells with 100 total copy-number events per cell. We modeled the cells after a population comprised of 9 distinct clonal populations, with 10 cells per population (Figure 2.9A). We began by generating 3 primary clonal populations by introducing 80 copy-number events compared to the parent diploid cell. Next, for each of the 3 primary clones, we generated 3 sub-clonal populations by introducing an additional 20 non-overlapping copy-number events to the original clones. Overall, this resulted in 9 distinct sub-clones belonging to 3 larger clonal populations with a total of 100 CNVs with respect to the human reference genome (hg19). The genome positions of CNVs were non-overlapping and generated from a uniform random distribution across the genome. The lengths of CNVs were generated from an exponential distribution with a mean of 5Mb and bounded between the range of 200kb and 20Mb to approximate the CNVs observed in the genuine data. The copy-number states of the CNVs were generated from a Poisson distribution with a mean of 2.5 excluding the value 2. We generated 10 cells from each of the 9 subclones (90 cells in total) by simulating reads from the subclone reference sequences generated above. For each cell, we simulated 200k, 101bp, single-end reads from the subclone reference sequence using dwgsim (https://github.com/nh13/DWGSIM) (`dwgsim -n 101 -z -1 -e .01 -d 1 -r 0 -1 101 -2 0`). For each cell, the simulated reads were then mapped to the hg19 human reference genome using the command `bowtie hg19.fa –S -t -m --best -strata` and filtered for only uniquely mappable high scoring reads (quality > 25). The SAM output was then converted to BED format and all 90 cells were uploaded and analyzed directly within Ginkgo with variable length 50kb bins.

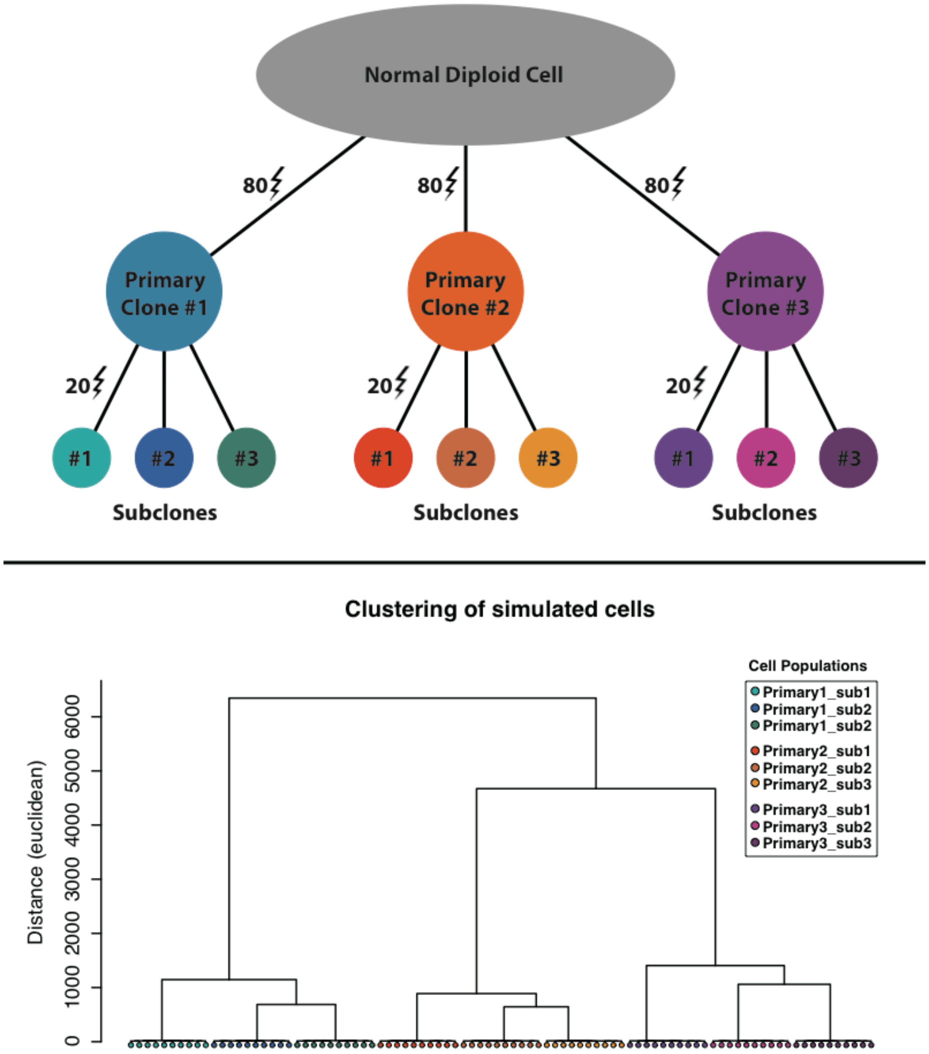


Figure 2.9: (A) Model representation of the 9 distinct subclones generated by simulation of 100 copy number events with respect to the reference. **(B)** Hierarchical clustering of the 90 samples by Ginkgo. Ginkgo perfectly recovers the underlying subclonal population structure.

As shown in Figure 2.9B, Ginkgo accurately reproduces the population structure through hierarchical clustering. In addition, we examined Ginkgo's ability to call CNVs by examining the false negative and false positive rates for all 90 cells at three different read counts (2M, 1.5M, 1M) across three different bin sizes (100kb, 50kb, 25kb). As shown in Table 2.1, we find that Ginkgo has a 0.15% false negative rate and a 0.08% false positive rate, excluding those bins that are partially spanned by a copy number alteration. When the entire genome is considered, including partially spanned bins, Ginkgo still has only an ~2%

false negative and ~1.2% positive rate. Hence, as expected, errors are almost exclusively concentrated at the boundaries of CNVs where the precise end of the event cannot be determined due to the extremely low coverage available or partially spanning of a bin.

We compared these results to the widely used CNVnator algorithm (Abyzov et al. 2011) for bulk sequencing CNV analysis, and find that Ginkgo performs CNV calls with higher accuracy (Table 2.1). Furthermore, CNVnator and other bulk sample analysis programs do not attempt to assign integer copy number states. In this analysis, we measured Ginkgo's accuracy with this stricter requirement while for CNVnator, we could only evaluate if an amplification or deletion had been identified. Ginkgo also has numerous features for evaluating population-wide CNV relationships (heatmaps, phylogenetic trees, multi-sample GC and Lorenz plots) that are also not present in CNVnator. From a practical sense, we also find Ginkgo to be substantially faster than CNVnator for the 90 cell evaluation, requiring a few hours via a simple web-interface rather than several days.

Simulated reads (M)	Mapped reads (M)	Mean bin length (kb)	False Negative Rate (%)			False Positive Rate (%)		
			Ginkgo Complete	Ginkgo	CNVnator	Ginkgo Complete	Ginkgo	CNVnator
2.0	1.64	100	0.15	2.03	6.37	0.08	1.28	0.69
2.0	1.64	50	0.18	1.29	5.86	0.07	1.20	0.5
2.0	1.64	25	0.26	1.63	6.01	0.05	1.16	0.54
1.5	1.23	100	0.22	2.22	6.46	0.10	1.34	0.75
1.5	1.23	50	0.28	1.67	5.99	0.07	1.21	0.66
1.5	1.23	25	0.39	2.37	6.1	0.08	1.21	0.6
1.0	0.82	100	0.33	2.47	6.42	0.17	1.41	0.94
1.0	0.82	50	0.50	2.17	6.23	0.13	1.24	1.03
1.0	0.82	25	0.75	3.82	6.03	0.14	1.24	0.68

Table 2.1: Simulation accuracy. False negative and false positive rates for genomes with 100 simulated copy number events at varying read depths and bin sizes. "Ginkgo complete" represents only the segments of copy number variants that fully overlap bin boundaries.

2.2.3. New algorithms for single-cell CNV analysis

The Ginkgo pipeline builds on previous single-cell sequencing work (Navin et al. 2011; Baslan et al. 2012) and contains several novel features that we describe below: **(1)** an algorithm for determining absolute copy-number state from the segmented raw read depth; and **(2)** a method for controlling quality issues in the reference assembly.

2.2.3.1. Absolute copy-number state algorithm

Since we are analyzing single-cell data, we expect every genomic locus to have an integer copy number (CN) value. Furthermore, the quantized nature of single-cell data means that the same number of reads per bin should separate every sequential CN state, e.g., ~50 reads for CN 1, ~100 reads for CN 2, ~150 reads for CN 3, etc. While biological and technical noise prevent read counts from segregating perfectly into distinct CN states, read counts should still be centered around integer CN states. The most direct approach for determining the CN state of each cell is available for users that have *a priori* knowledge of the ploidy of each sample. For example, cells that are DAPI-stained prior to cell sorting can be gated based on their fluorescence activity, and ploidy can be determined by comparing its fluorescence activity to that of a reference cell with a known CN state. With these data, Ginkgo determines the copy number state of each sample by scaling the segmented bin counts such that the mean bin count is equal to the ploidy of the sample. Finally bin counts are rounded to integer copy number values. Advances in fluorescence activated cell sorting (FACS) will make this copy number prediction even more accurate in time, although cells that are incorrectly sorted and placed into wells with more than one cell will show much higher fluorescence activity and will have an incorrectly inferred copy number state.

Since FACS data is not always available for analysis and has potential for error, Ginkgo provides an alternative to determine the copy number of each sample. As discussed earlier, before determining the CN state of a cell, the cell is binned, normalized, and segmented. This copy number profile with a mean of one is referred to as the raw copy number profile (RCNP). If the true genome-wide copy number of a sample were equal to X, the scaled copy number profile (SCNP) would then be the product of RCNP and X, and the final *integer* copy number profile (FCNP) would be the rounded value of the SCNP so all segments contain an integer value.

With these relationships, Ginkgo infers the genome-wide copy number X using numerical optimization (Pseudocode 1). For a given cell, Ginkgo first determines the SCNP and FCNP for all possible values of X in the set [1.50, 1.55, …, 5.95, 6.00]. Ginkgo then computes the sum of square (SoS) error between the SCNP and the RCNP for each value of X and selects the value of X with the smallest SoS error. Once the multiplier is identified and applied, the scaled bins are rounded to generate the final integer copy number profile for each sample. Intuitively, this is equivalent to finding the copy number multiplier that causes the normalized segmented bin counts to best align with integer copy number values. Sample runs of the algorithm are shown in Figure 2.10.

```
allMult = 1.5 : 0.05 : 5.5                                                           Define all possible multipliers

allErrors = $mean\ [\ \left(CNV_{seg} * allMult - round(CNV_{seg} * allMult)\right)^2\ ]$   Get average SoS error for each multiplier

multiplier = allMult[ which.min(CNerror) ]                                           Use multiplier that yields smallest error
```

Pseudocode 1: Algorithm used to find best multiplier, defined as factor that minimizes the sum of squares error between the scaled segmented profile and the rounded scaled segmented profile.

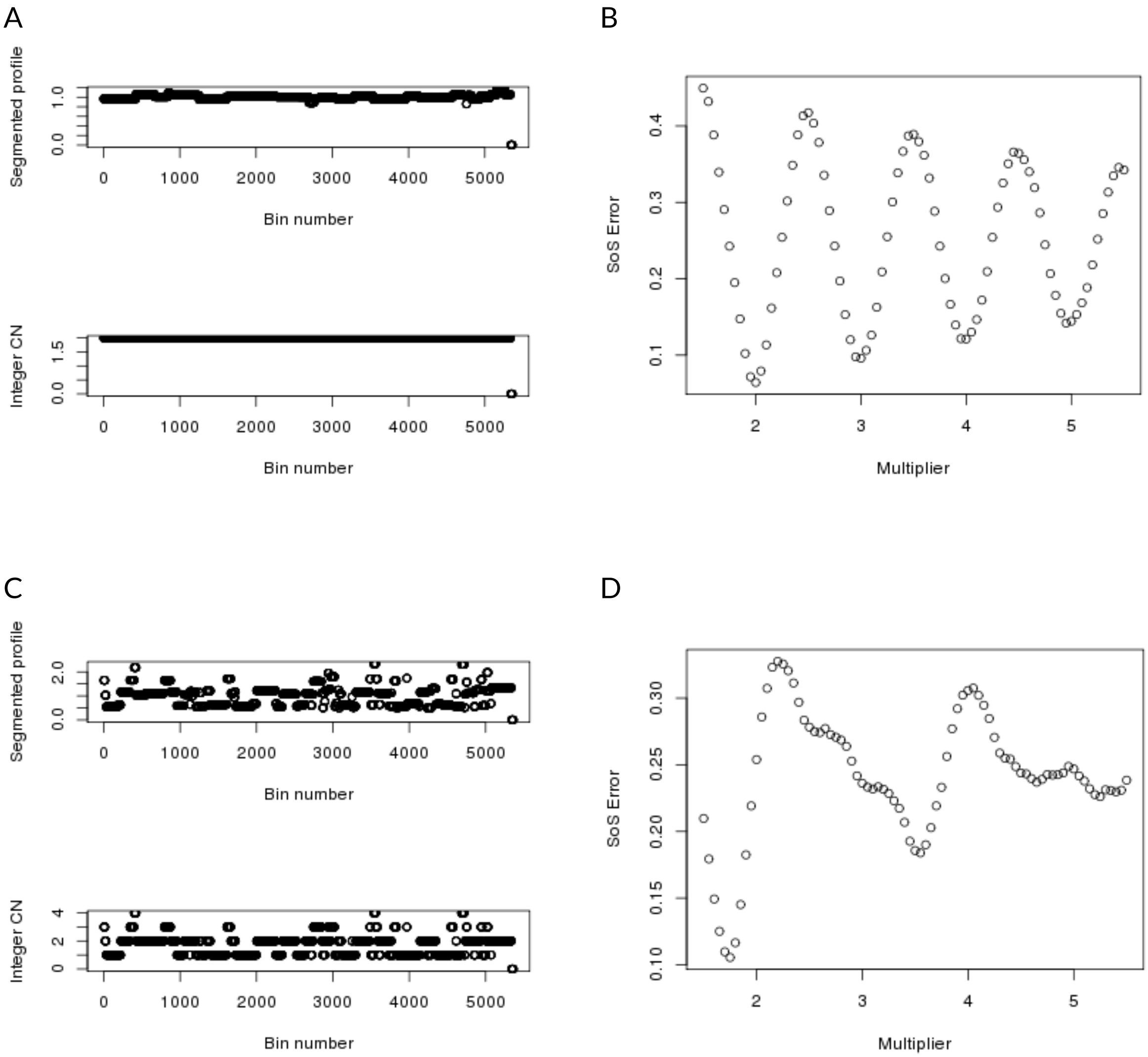


Figure 2.10: Sample runs of the integer copy-number inference algorithm using **(A-B)** a diploid cell, and a **(C-D)** breast tumor cell. Note that in the plots of Sum-of-Squares errors (B, D), the minimum peak is far away from other peaks, thereby simplifying the choice of multiplier.

Despite a lot of success with this approach, there are rare occasions, especially at low number of reads, where two potential multipliers exhibit very similar Sum-of-Squares errors (Figure 2.11A), which could lead the algorithm to choose the wrong multiplier (Figure 2.11B). One way to address this issue would be to use ploidy information obtained from staining during FACS sorting. However, when that information is not available, we apply the following heuristic to emulate a human decision: if there are two peaks that are at very similar heights, choose the peak corresponding to the smallest multiplier (Pseudocode 2). For example, if there are two peaks at multipliers 2 and 3, the algorithm will choose 2 (Figure 2.11C).

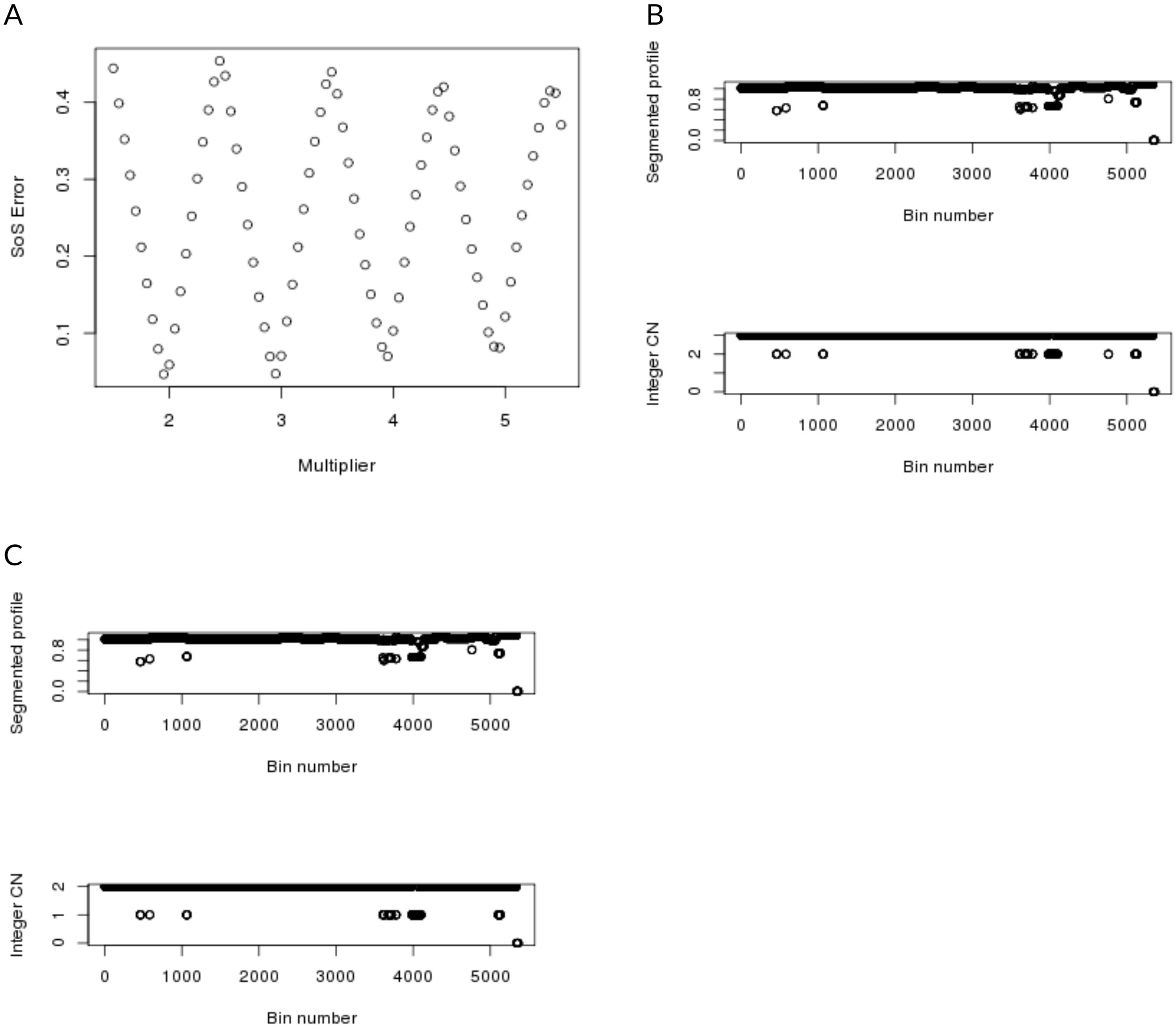


Figure 2.11: (A) Sample situation where the peaks occur at very similar Sum-of-Square errors. **(B)** Without the proposed heuristic, the algorithm could choose the wrong multiplier. **(C)** Using the proposed heuristic, the algorithm detects two peaks that are very close to each other and chooses the smallest multiplier among them.

```
allPeaksMin = which( diff(sign(diff( allErrors ))) == 2 )        Compute  discrete 2nd derivative to find
allPeaksMax = which( diff(sign(diff(allErrors ))) == -2 )           all peaks at local minima and maxima


naivePeak = min(allErrors[allPeaks])                             Naïve peak occurs at global minimum
closestPeak = closest(allErrors[allPeaks], naivePeak)            Find peak closest to naïve peak


$\Delta = \frac{|\text{ naivePeak} - \text{closestPeak }|}{\text{max(allErrors[allPeaksMax])} - \text{min(allErrors[allPeaks])}}$
if(Δ < 0.05)                                                     If the two peaks are very similar, choose
    CNmult = min( CNgrid[ c(naivePeak, closestPeak) ] )             the smallest multiplier
```

Pseudocode 2: This heuristic will detect situations where two multipliers have very similar sum-of-square errors, and will choose the smallest multiplier of the two.

2.2.3.2. Controlling for quality issues in the reference assembly

As previously demonstrated (Navin et al. 2011), fixed-length bins introduce mappability issues in highly repetitive regions such as centromeres. These regions are often "dead zones" that remain empty even if the overall coverage is high, and will bias segmentation algorithms into identifying nonexistent breakpoints. While Ginkgo supports normalizing copy-number profiles by a diploid cell, this can introduce noise in bins with low counts. We recommend using a variable-binning strategy in which variable-length bins with equal mappability are used, allowing uniform mapping of reads. When neglected, for example, in Ni *et al.*, dead zones are present at chromosome boundaries and centromeres. In comparison, the same profiles generated by Ginkgo are less noisy and free of dead zones (see Ginkgo website). We note, however, that in using the variable-length bin strategy, certain regions of the genome, specifically near the centromeres of several chromosomes, have consistently higher read depth than expected whether profiling bulk DNA or single cells. This problem also occurs in fixed bins but is less severe due to the substantial read drop out caused by using fixed intervals. These peaks may be the result of occasional mis-mapping of highly repetitive sequences from elsewhere in the genome to unique but similar sequences located near the peri-centromeric DNA. This is also likely influenced by imperfect reference assemblies that do not fully represent the correct genetic sequences. We have termed these “bad bins” and have provided an option in Ginkgo to mask them for human (hg19) for simplicity of presentation.

Using data from 54 normal individual diploid cells, these bins (designated as “bad bins”) were determined in the human reference genome (hg19) as follows. The bin counts were divided by the mean bin counts for each cell to normalize for differences between cells in total read count. For each chromosome, the mean of the bins over all cells is subtracted from each normalized bin count to normalize for differences between chromosomes. The mean and standard deviation of the autosomes is then used to compute an outlier threshold corresponding to a p-value of 1/N, where N is the number of bins used. These bins are masked from downstream copy number analysis.

2.2.4. DOP-PCR and MALBAC outperform MDA in data quality

Although Ginkgo corrects for many of the biases present in single-cell data, higher-quality data inevitably lead to higher-quality results. We set out to compare the biases and differences in coverage uniformity among the three most widely published WGA techniques—MDA, MALBAC and DOP-PCR—using three distinct data sets with each method. Raw sequencing reads downloaded from NCBI were mapped to the human genome and sub-sampled to match the sample with the lowest coverage. Aligned reads were then binned into variable-length intervals across the genome that averaged 500kb in length but contained the same number of uniquely mappable positions (Methods). We use these binned read counts to measure two key data-quality metrics: GC bias and coverage dispersion. Importantly, raw bin counts provide a view of data quality that is impartial to the different approaches to segmentation, copy-number calling and clustering.

GC content bias refers to preferential amplification of a given genomic region due to the local fraction of G and C nucleotides[17]. This bias introduces cell- and library-specific correlations between GC content and bin counts. In particular, when the GC content in a genomic region falls outside of a certain range (typically <0.4 or >0.6), read counts rapidly decrease (Methods). We found that the GC bias of MDA was very high compared with that of MALBAC or DOP-PCR (Figure 2.12A). Only 45.9% of MDA bin counts fell within the expected coverage range, compared with 94.0% of MALBAC bin counts and 99.6% of DOP-PCR bin counts. It is important to note that regardless of the WGA approach used, each cell has unique GC biases that must be individually corrected.

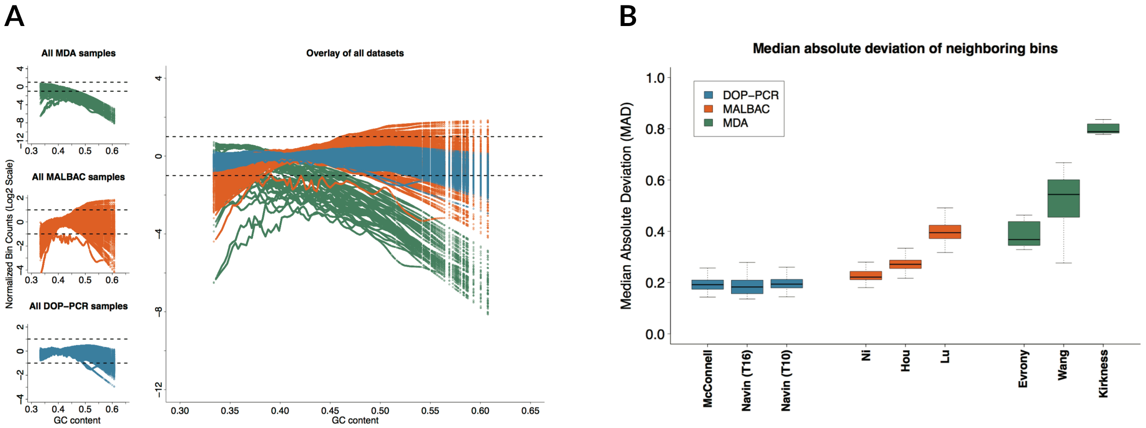


Figure 2.12: (A) LOWESS fit of GC content with respect to log-normalized bin counts for all samples in each of the 9 data sets analyzed: 3 for MDA (top left, green), 3 for MALBAC (center left, orange) and 3 for DOP-PCR (bottom left, blue). Each colored line in a plot corresponds to the LOWESS fit of a single sample. The upper and lower dashed lines in each plot mark twofold increased and decreased values with respect to the average observed coverage. Note that the MDA plot has a different y-axis scale because of large GC bias. **(B)** The MAD between neighboring bins. A single pairwise MAD value was generated for each sample in a given data set and is represented in the figure by a box and whisker. The bold line in the center of a box represents the mean, the box boundaries represent the quartiles and the whiskers represent the remaining data points. Names along the x-axis are the first authors of the referenced studies. T16 and T10 refer to types of breast cancer tumors as established by (Navin et al. 2011). The high biases present in the MDA data sets made it difficult to compare DOP-PCR and MALBAC samples. Supplementary Figure 3 shows this comparison more clearly.

As a further measure of data quality, we calculated the median absolute deviation (MAD) of all pairwise differences in read counts between neighboring bins for each sample, after normalizing the cells by dividing the count in each bin by the mean read count across bins. The MAD is resilient to outliers caused by copy-number breakpoints, as transitions from one copy-number state to another are relatively infrequent. Instead, pairwise MAD reflects the bin count dispersion due to technical noise. As expected on the basis of previous comparisons (Zong et al. 2012; Navin 2014), MDA data displayed high levels of coverage dispersion, with a mean MAD two to four times that of the DOP-PCR data sets (Figure 2.12B). In addition, the MALBAC and MDA data sets showed large differences in data quality between studies, whereas the DOP-PCR data sets showed consistently flat MAD across all three studies (Figure 2.13).

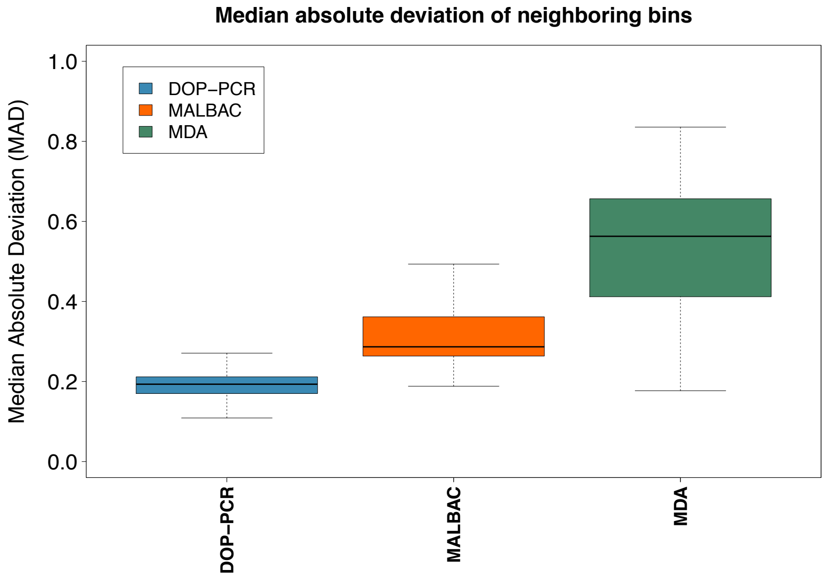


Figure 2.13: The median absolute deviation (MAD) of neighboring bins across 3 WGA approaches. A single pair-wise MAD value is generated for each sample in a given dataset and represented by a box and whisker plot. The DOP-PCR datasets show the lowest mean MDA as well as the lowest variance across samples. While certain MDA samples outperform the MALBAC dataset, they show much large variability in data quality than MALBAC.

2.2.5. DOP-PCR outperforms MALBAC for single-cell CNV analysis

Whole-genome amplification using MDA introduces a large degree of biases compared to MALBAC or DOP-PCR, limiting its applicability to CNV analysis. As such, we focused the scope of the remaining comparisons on the latter two WGA techniques.

For a fine-grained comparison of MALBAC and DOP-PCR, we compare the T10 dataset from Navin *et al*. 2011 and the CTC dataset from Ni *et al*. 2013 due to their similar biological and technical conditions and similar published analysis. Both datasets contain aneuploid cancer cells, were sequenced to similar depth (CTC mean read count: 4,133,466; T10 mean read count: 6,706,119), and were used to generate phylogenetic clusters of samples based on CNVs. We begin by comparing the coverage dispersion and investigate the minimum coverage and bin size needed to reproduce the published results.

Using the MAD criteria described above, the DOP-PCR-based T10 dataset shows markedly better bin-to-bin correlation than the MALBAC-based CTC dataset as judged by a lower MAD of adjacent and offset bin counts (Figure 2.14). For adjacent bins, the first quartile of the CTC MAD comparison (orange) is higher than the third quartile of the T10 MAD comparison (blue). As we increase the bin offset, greater variation is seen in the CTC data as show by the separation of the mean MAD between the T10 and CTC datasets. We interpret this to mean that there is more local trending in amplification efficiency in MALBAC than in DOP-PCR data.

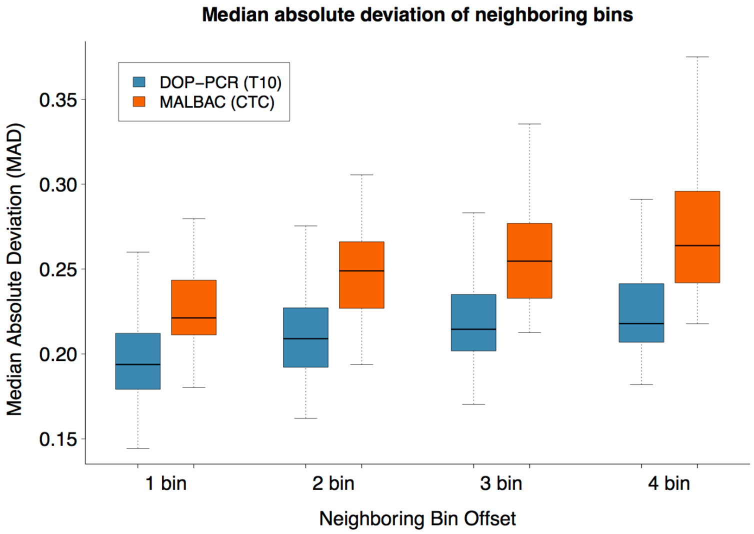


Figure 2.14: A comparison of MAD between the Navin *et al*. (T10) shown in blue and Ni *et al*. (CTC) shown in orange. As the bin offset increases the separation between the mean T10 MAD and mean CTC MAD grows.

To understand the effects of noise further, we evaluated each dataset to discriminate distinct copy number states. Because the copy-number states of individual cells are integer, we expect the data to be centered at integer values. If the data is highly uniform, read coverage per bin should tightly surround integer copy-number states. As bin count dispersion around copy-number states increases, or is influenced by local chromosomal trends, the distinction between copy-number states will blur.

To examine this, we generated a histogram of the normalized read count distribution for the CTC and T10 datasets In Figure 2.15, we show the distributions of bin counts for representative cells: excellent, typical, and lower quality cells as well as the highest quality population average. All T10 profiles have distinct peaks representative of integer copy-number values. While there are a few cells in the CTC dataset that have distinct peaks, many of the CTC profiles have considerably worse resolution with substantial blurring between CN states.

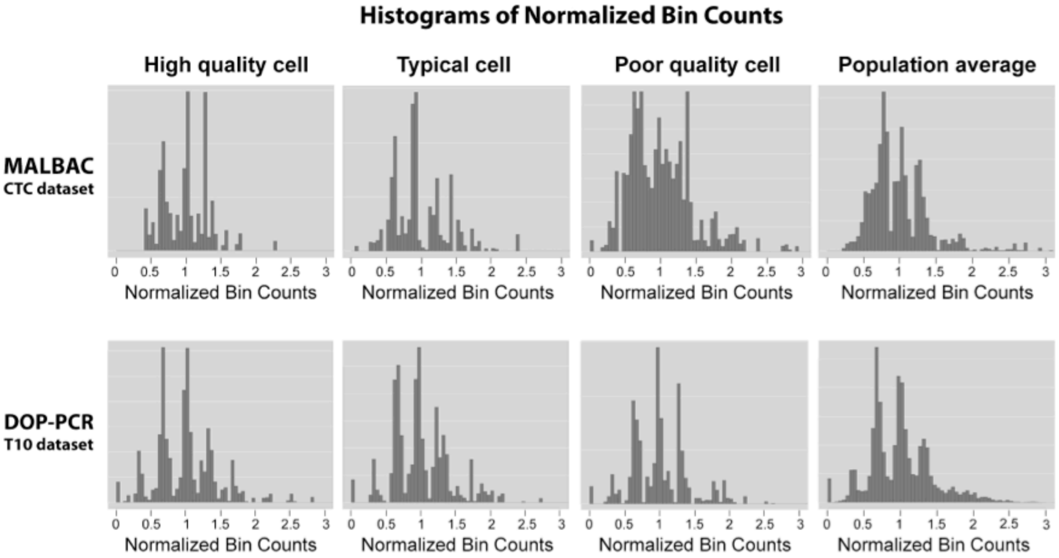


Figure 2.15: Histograms of normalized bin counts across the CTC and T10 datasets, for a high-, typical-, and poor-quality cell. The rightmost column contains histograms of high quality cell population averages. Distinct peaks are representative of clean data from which accurate copy number calls can be made.

2.3. Discussion

Single-cell sequencing has tremendous potential to shed light on genetically complex environments. Early applications have already discovered surprisingly high levels of heterogeneity and copy-number mosaicism in tumors, neurons, and sperm. The implications are profound and provide a new computational lens to observe, for example, the founding cancer cell population and trace its development through a tumor and metastasis. Many projects are now underway to apply the technology to diverse tissue and cell types. The experimental protocols are maturing, and with Ginkgo, a validated, open-source end-to-end pipeline is now available for researchers as well. The interactive visual analytics environment provides researchers with an intuitive platform to explore and understand their population of cells. It begins with a high-level overview of the population represented by dendrograms and heatmaps. It then gives researchers the ability to zoom in on the copy-number profile of individual cells, filter the analysis for subsets of cells, and inspect the details of copy-number state or read depth on demand. Throughout each stage of the analysis, statistical summaries and quality metrics guide researchers to the most significant and most accurate data. Figures and data tables are available for download to be embedded into presentations or publications.

Furthermore, we found that DOP-PCR outperformed both MALBAC and MDA in terms of data quality. As previously reported (Zong et al. 2012; Cai et al. 2014; Chen et al. 2014; de Bourcy et al. 2014; Navin 2014), MDA displayed poor coverage uniformity and low signal-to-noise ratios. These characteristics, coupled with overwhelming GC biases, make MDA unreliable for accurate determination of CNVs compared with the other two techniques examined. Furthermore, although both DOP-PCR data and MALBAC data can be used to generate CNV profiles and identify large variants, DOP-PCR data have substantially lower coverage dispersion and smaller GC biases than MALBAC data. Our results indicate that given the same level of coverage, data prepared using DOP-PCR can reliably call CNVs with better signal-to-noise ratios and are more reliable for accurate copy-number calls than are data obtained with MDA or MALBAC.

Finally, to guide researchers in their analysis, we highlight common computational pitfalls in single-cell analysis and discuss how Ginkgo corrects for them. Correctly calling CNVs from single-cell sequencing data is still an open problem and Ginkgo's wide array of parameters leaves it flexible to users' needs while remaining robust. As single-cell sequencing methods further develop, we anticipate it will become practical to analyze SNPs and other smaller mutations reliably. As this occurs, and as other algorithmic improvements are made, we will incorporate those new ideas into our toolbox. We are also exploring ideas for the analysis and visualization of single-cell RNA sequencing as those protocols become more widely available. Users are encouraged to customize and contribute back to Ginkgo's open-source code base hosted on GitHub.

2.4. Methods

2.4.1. Mapping reads to genome

Reads were mapped to hg19 using bowtie (Langmead et al. 2009) and only uniquely mapped reads (mapping quality score >= 25) were kept.

2.4.2. Binning reads

Copy number analysis begins with binning uniquely mapping reads into fixed-length or variable-length intervals across the genome. This aggregates read depth information into larger regions that are more robust to variable amplification and other biases. As discussed in the main text, fixed-length bins are generally discouraged as they lead to read drop out in regions that span highly repetitive regions, centromeres, and other complex genomic regions. To generate boundaries for variable-length bins, we use the method outlined in (Navin et al. 2011), where we sample 101bp stretches of the reference assembly at every position along the genome. These simulated reads are mapped back to the genome using Bowtie and only uniquely mapping reads are analyzed. For a given bin size, we assign reads into bins such that each bin has the same number of uniquely mappable reads. Consequently, intervals with higher repeat content and low mappability will be larger than intervals with highly mappable sequences, although they will both have the same number of uniquely mappable positions. Using variable-length bins with sufficient depth of coverage and consistent ploidy, high quality reads are expected to map evenly across the entire genome. Users are provided with a variety of bin sizes from which to choose, depending on the overall coverage available; if the mean coverage per bin is too low, we encourage users to use larger bins.

2.4.3. GC bias correction

Once reads are placed into bins, Ginkgo normalizes each sample and corrects for GC biases prior to segmentation. The normalization process begins by dividing the count in each bin by the mean read count across all bins. This centers the bin counts of all samples at 1.0. To identify and correct GC biases, Ginkgo computes a locally-weighted linear regression using the R function lowess (Cleveland 1981) (`smoother span = .5, iterations = 3, delta=0.1*range(x)`) to model the relationship between GC content and log-normalized bin counts. This lowess fit is then used to scale each bin such that the expected average log-normalized bin count across all GC values is zero. After the lowess fit, we monitor the bias of each cell by calculating the proportion of bins that fall outside an expected coverage of zero by +/- 1, log base 2.

2.4.4. Segmentation

Following GC bias correction, bin counts are segmented using Circular Binary Segmentation (CBS) to reduce fluctuations in noise across chromosomes and identify longer regions of equal copy number (Olshen et al. 2004). The key step during segmentation is selecting the right reference sample for comparison. Using a diploid sample to normalize bin counts can eliminate additional biases uncorrected by GC normalization. Although Ginkgo supports uploading data from such a cell, this is not always available so Ginkgo provides alternatives for segmenting samples: **(1)** Independent segmentation, where samples are segmented independently by their own normalized bin count profiles; and **(2)** Sample with lowest IOD, where Ginkgo selects the sample with the lowest index of dispersion (IOD - the ratio between the read coverage variance and the mean) and uses that sample as a reference for all other samples. The sample with the lowest index of dispersion will likely be among the most evenly balanced ploidy and highest quality of all submitted cells.

2.4.5. Clustering

Before visualization, the final step is to look outside the scope of individual cells and determine the overall population structure. Ginkgo first determines the distance (dissimilarity structure) between all cells. We provide six choices of distance metrics: Euclidean, $d(x,y) = \sqrt{\sum_i (x_i - y_i)^2}$, Manhattan, $d(x,y) = \sum_i |x_i - y_i|$, maximum, $d(x,y) = \max\{\,|x_1 - y_1|, \dots, |x_n - y_n|\,\}$, Canberra, $d(x,y) = \frac{\sum_i |x_i - y_i|}{|x_i| + |y_i|}$, and Minkowski, $d(x,y,p) = \sqrt[p]{\sum_i |x_i - y_i|^p}$ for $p \geq 1$. After computing the dissimilarity matrix, Ginkgo then computes a dendrogram by hierarchically clustering samples using one of four different agglomeration methods: single linkage, complete linkage, average linkage, and ward linkage. In addition, Ginkgo supports building a phylogenetic tree using the more robust Neighbor Joining algorithm (Saitou and Nei 1987).

2.4.6. Server

Ginkgo is hosted at Cold Spring Harbor Laboratory on a CentOS server with 24 CPUs, 64GB RAM and 7TB of total storage space. Most algorithms for data analysis were implemented in the R language, except the read-binning step, which is written in the C language to optimize running time. The user-facing interface was written in HTML, CSS and JavaScript, using the Twitter Bootstrap (http://getbootstrap.com) and jQuery libraries (http://jquery.com). The phylogenetic tree was built using jsPhyloSVG (Smits and Ouverney 2010), and the interactive copy-number profile viewer was based on the Dygraphs library (http://dygraphs.com). A PHP script manages all communication between the browser and the server, and between the server and the analysis pipeline. When a user launches an analysis, the PHP script launches a Bash script that executes R scripts in the required order. Every few seconds, the browser queries the back-end to retrieve the current progress of the analysis. Ginkgo is available open-source at http://github.com/robertaboukhalil/ginkgo.

To enable standalone installations of Ginkgo (for large-scale analyses on local servers), we also provide a Docker image that contains the Ginkgo source code, R dependencies, and all required server software (PHP, MySQL, Apache). The Docker image can be obtained from https://registry.hub.docker.com/u/robertaboukhalil/ginkgo/.

2.5. Contributions

- Ginkgo software and copy-number analyses were done by Tyler Garvin and myself.
- The algorithm in Pseudocode 1 was developed by Michael Wigler.
- Many thanks to Jude Kendall for guidance and assistance with bioinformatics algorithms.

Optimized single-cell sequencing strategies for copy-number analysis and tumor heterogeneity

This chapter has been reproduced with modifications from:

> Aboukhalil R, Alexander J, Kendall J, Baslan T, Wigler M, Atwal GS. **Optimized single-cell sequencing strategies for copy-number analysis and tumor heterogeneity**. *In preparation*.

3.1. Introduction

Single-cell DNA sequencing is an important tool for probing the underlying biology of heterogeneous tissues and rare cells, where genomic variability is obscured by bulk sequencing of millions of cells (Wigler 2012; Shapiro et al. 2013). One of the goals of single-cell sequencing is to identify large-scale (>10kb) copy-number variations (Baslan et al. 2012), which are known to play a critical role in cancer (Shlien and Malkin 2009b). In recent years, In recent years, single-cell sequencing was used to probe tumor evolution and metastasis in breast tumors (Navin et al. 2011), analyze circulating tumor cells to monitor disease progression and effectiveness of therapy (Dago et al. 2014), and identify subtype-specific CNV markers in lung cancer (Ni et al. 2013). Low coverage (<1X) single-cell DNA sequencing is an informative and cost-effective approach for studying diseases such as cancer, which are often characterized by widespread CNV events (Baslan et al. 2015). To automate the analysis of these datasets, software tools have recently been made available (Garvin et al. 2015).

Despite these advances, single-cell sequencing remains financially prohibitive for studying thousands of cells. As such, there exists a trade-off between the number of cells sequenced and the depth of coverage, and it is unclear which parameters yield greater biological insight into sample heterogeneity and population structure. Although guidelines exist to help investigators decide the appropriate sequencing depth for bulk sequencing of cancer samples (Griffith et al. 2015), no such guidelines exist for single-cell CNV analysis. It is therefore unclear which experimental parameters are necessary for accurate single-cell analysis, for both copy-number profiling and inference of population structure in clonal tumors.

To address this gap, we present a comprehensive analysis of single-cell data from 14 whole-genome sequencing datasets (Table 3.1). These datasets span a variety of cancer types, including triple-negative breast cancers (Navin et al. 2011; Wang et al. 2014), estrogen-receptor positive breast tumors (Wang et al. 2014; Baslan et al. 2015), a paired metastatic liver carcinoma (Navin et al. 2011), a highly rearranged HER2-amplified breast cancer cell line (Wang et al. 2014; Baslan et al. 2015), biopsies from prostates of different grades (Alexander 2016), and circulating tumor cells from lung cancer (Ni et al. 2013) patients. These data also make use of different whole-genome amplification methods—Degenerate Oligonucleotide Primed PCR (DOP-PCR) (Telenius et al. 1992; Blainey 2013) and Multiple Annealing and Looping Based Amplification Cycles (MALBAC) (Zong et al. 2012), allowing us to test the robustness of our results across differing amplification methods.

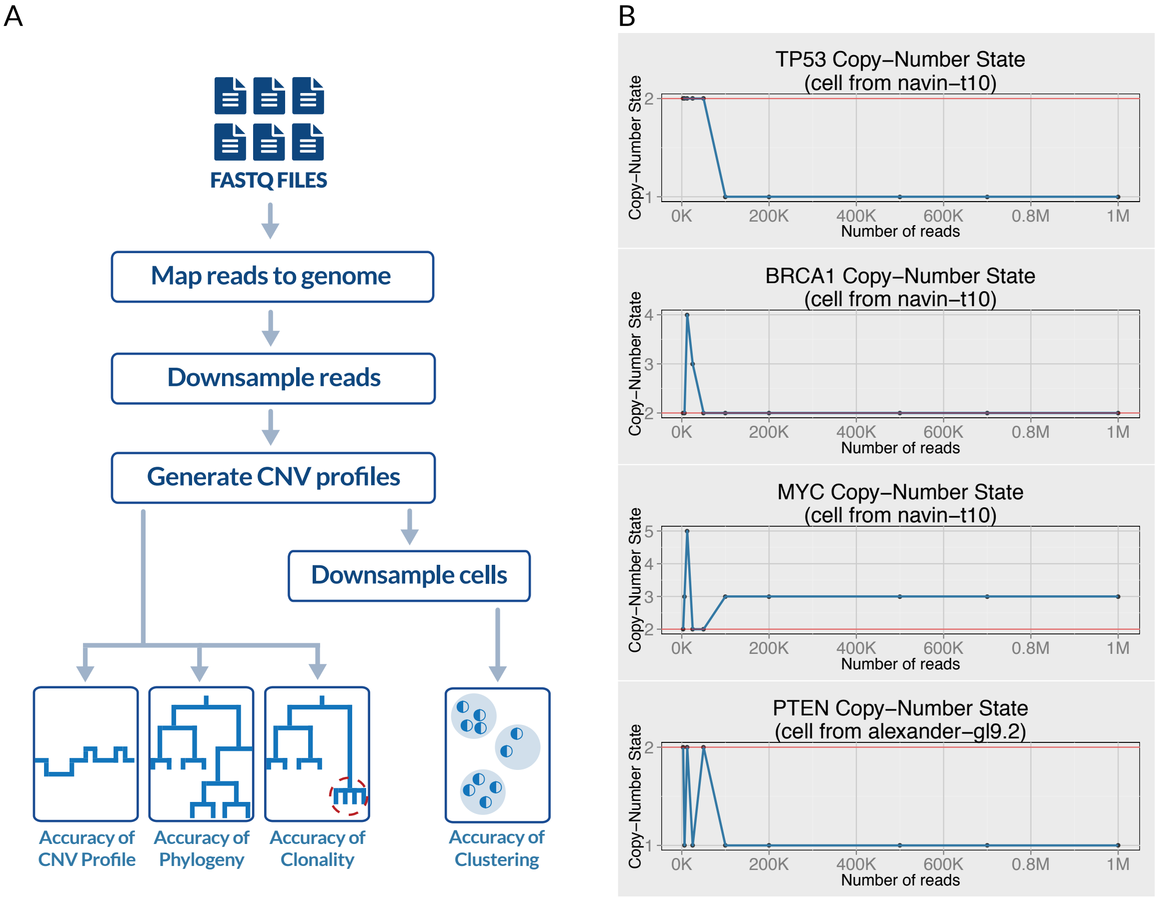


Figure 3.1: (A) The Drizzle pipeline automates all the steps required to perform the analyses highlighted in this study: (1) Given sequencing reads in FASTQ format, the reads from each cell are aligned to the reference genome; (2) Millions of subsampling experiments are carried out on the mapped reads. For each cell, we sample a random subset of reads at 15 different sampling depths, and 100 randomizations at each level; (3) At each downsampling depth, the reads are binned across the genome and copy-number profiles are inferred; (4) We evaluate the accuracy of reconstructing the copy-number profile, phylogenetic trees, and clonal subpopulations. Next, by downsampling cells, we evaluate the tradeoff between sequencing more cells and sequencing at greater depth. **(B)** Sequencing at the appropriate read depth has important clinical implications for the mutational status of tumor suppressor genes and oncogenes. Below a certain read depth, some copy-number events may no longer be detected (e.g. TP53, PTEN, MYC), while other events are incorrectly called as alterations (e.g. BRCA1).

By analyzing data from 2,826 single cells, we estimate the number of mapped reads per cell required to accurately recover a cell's copy-number profile, reconstruct phylogeny, and assess clinically relevant clonal subpopulations (Figure 3.1A). We also investigate the tradeoff between depth of coverage and the number of cells in a sample. Our work has important implications for future single-cell sequencing studies, as sequencing at lower depth can be an important cost-saving measure, especially for samples that are mainly characterized by large-scale copy-number alterations. To aid future investigators in the analysis and quality assessment of their own datasets, we developed Drizzle, an open-source package that automates the analyses presented here.

Study	Sample ID	Tissue	WGA method	# Cells
(Navin et al. 2011)	navin-t10	Breast cancer	DOP-PCR	100
	navin-t16p	Breast cancer	DOP-PCR	52
	navin-t16m	Liver metastasis	DOP-PCR	48
(Ni et al. 2013)	ni-lungctc	Lung CTCs	MALBAC	68
(Baslan et al. 2015)	baslan-pt31	Breast cancer	DOP-PCR	89
	baslan-pt41	Breast cancer	DOP-PCR	138
	baslan-skbr	SK-BR-3 cell line	DOP-PCR	94
	baslan-315A	315A cell line	DOP-PCR	95
(Wang et al. 2014)	wang-nucseq-skbr	SK-BR-3 cell line	DOP-PCR	50
	wang-nucseq-er	Breast cancer	DOP-PCR	50
	wang-nucseq-tnbc	Breast cancer	DOP-PCR	50
(Alexander et al. In preparation)	alexander-gl6.1	Prostate cancer	DOP-PCR	494
	alexander-gl7.1	Prostate cancer	DOP-PCR	739
	alexander-gl9.1	Prostate cancer	DOP-PCR	349
	alexander-gl9.2	Prostate cancer	DOP-PCR	505

Table 3.1: All datasets analyzed in this study, along with citation, tissue, whole-genome amplification (WGA) method, and number of cells.

To demonstrate the clinical implications of sequencing at the correct depth, consider Figure 3.1B, where we highlight examples of inferred copy-number profiles from single cells of breast and prostate tumors. Amplifications or deletions of important tumor suppressor genes and oncogenes, although correctly inferred at relatively low coverage, are miscalled below certain read depths.

3.2. Results

3.2.1. Accurate copy-number profiles require ~1M reads per cell

We re-analyzed 14 single-cell sequencing datasets (Table 3.1), for a total of 2,826 cells, and generated equally-mappable bins along the genome as discussed in (Baslan et al. 2012) to account for read mappability artifacts. For each cell and subsampling depth, we calculate the correlation between the sub-sampled CNV profile and the original profile inferred using all the reads. As shown in Figure 3.2A-B, reconstructing the copy-number profile with high accuracy (>0.9 Pearson correlation) generally requires ~1M mapped reads per cell using a 50kb binning scheme. We find that this result is robust for a variety of tumors types and grades, sequencing layouts (single-end, paired-end), short read lengths (30-101bp) and whole-genome amplification approaches (DOP-PCR and MALBAC). Interestingly, all cells seem to exhibit a critical threshold (or elbow) below which the signal is overwhelmed by sampling noise, and above which we observe diminishing returns in accuracy (Figure 3.2A-B and Supplementary Figure 3.1). Furthermore, high accuracy of copy-number profiling (>90% correlation) is observed at read depths well below the depth of coverage in the original studies (Figure 3.2C); on average, cells were sequenced at read depths ~8 fold higher than necessary. Despite low sequencing depth of coverage, it is therefore still possible to obtain accurate whole-genome copy-number profiling. A plausible concern is that the qualitative behavior of these curves would remain unchanged regardless of the starting number of reads. To therefore test the robustness of our analysis with respect to the initial starting number of reads, we performed downsampling experiments where we varied the starting number of reads per cell. For each starting point, we determine a quality score by calculating the average normalized area under the curve (Supplementary Figure 3.2). This quality score reflects whether the initial number of reads is sufficient to accurately generate the copy-number profiles. We recommend an area under the curve close to 1; for example, in the *navin-t10* tumor sample, 1M reads per cell yields scores > 0.95 (Figure 3.2D). Finally, we verify that the shape of the subsampling curve is not due to the segmentation algorithm having too few reads per bin. To that end, we repeat the analysis and adjust the bin size at each subsampling level so that we fix the number of reads per bin, and observe that the shape of the curve is maintained (Supplementary Figure 3.3 and Methods).

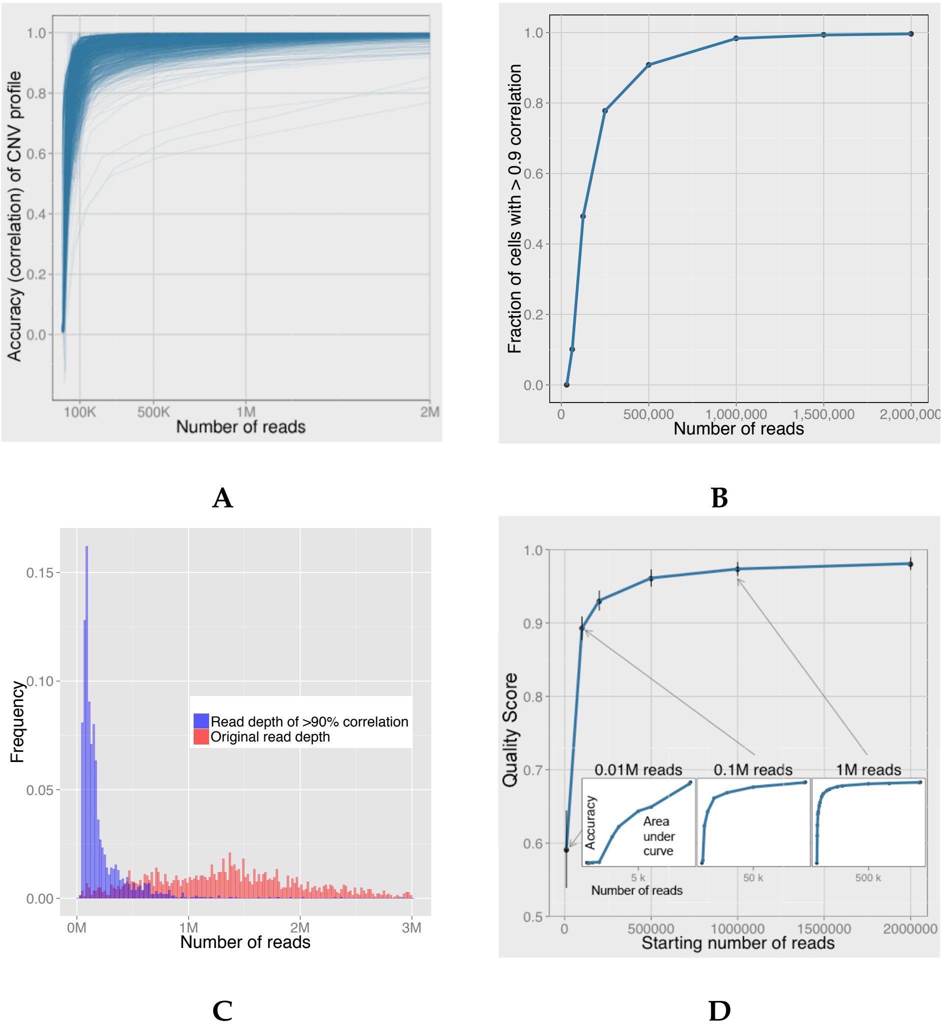


Figure 3.2: Generally, ~1M reads per cell are sufficient for accurate copy-number profile construction at 50kb bin resolution. **(A)** Correlation of the subsampled CNV profile of each cell at various read depths. **(B)** The vast majority of cells in this analysis exhibit >0.9 Pearson correlation at 1M reads per cell. **(C)** Cells are shown to have been sequenced at greater depth of coverage than necessary in the original studies. **(D)** The average normalized area under the cells' subsampling curves can be used as a measure of data quality. Insets illustrate a decrease in the area under the CNV accuracy curve as the starting read depth decreases; starting the subsampling analysis at 10K reads yields significantly worse CNV profile reconstruction than 1M reads (shown for *navin-t10* tumor sample).

3.2.2. Reconstructing population structure requires ~500K reads per cell

In addition to generating accurate copy-number profiles, single-cell copy-number studies often build phylogenetic trees to infer population structure, particularly in tumor samples where we expect to observe clonal subpopulations (Navin et al. 2011; Ni et al. 2013). In this section, we explore the effect of decreasing depth of coverage on the accuracy of phylogenetic tree construction. For this analysis, we chose the datasets in Table 3.1 that were obtained from patient tumors with >50 cells and only analyzed cells with >1M reads. This filtering resulted in the analysis of breast and prostate tumors spanning differing clinical grades and levels of intra-tumor heterogeneity. For each subsampling level, we build a Neighbor Joining tree (Saitou and Nei 1987) using each cell's breakpoint profile, and evaluate how well the tree was constructed using a tree similarity metric based on the Branch Score distance (Kuhner and Felsenstein 1994) (Methods).

At megabase pair resolution, all samples exhibit >90% tree similarity at 0.5M reads per cell (Figure 3.3A). Our results appear robust across various cancer types, tumor grade and level of tumor heterogeneity. Interestingly, the required number of reads does not vary drastically with larger bins (Figure 3.3B), which is expected since most cancers exhibit megabase-sized CNV events (Beroukhim et al. 2010b). To ensure that the shape of the curves is not an artifact of low bin counts during segmentation, we repeat the analysis while maintaining the number of reads per bin at each downsampling step and observe a similar trend (Supplementary Figure 3.4).

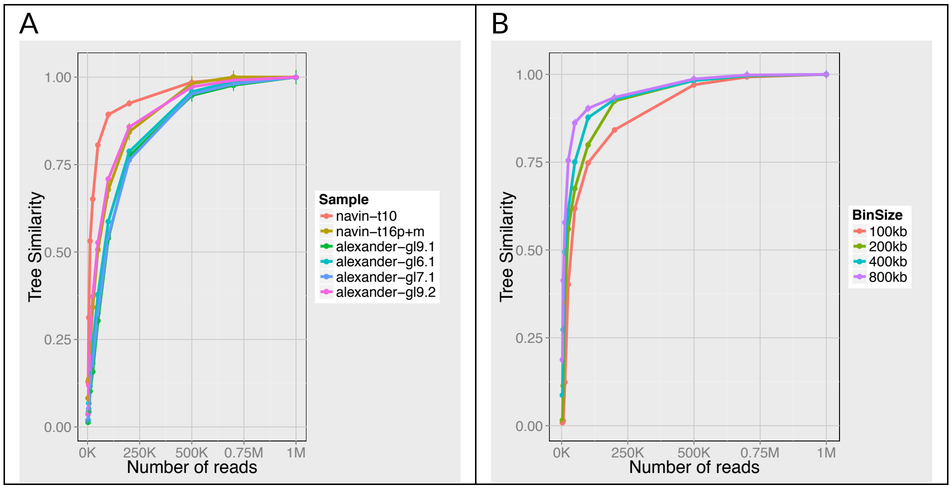


Figure 3.3: Approximately 500K reads per cell are required for accurate phylogenetic tree inference. **(A)** For the samples we analyzed, 0.5M reads per cell are sufficient for accurate phylogeny inference at 500kb bins. **(B)** Using smaller bins only changes the shape of the curves slightly but does not affect the required number of reads to attain > 90% accuracy (shown for *navin-t10* sample).

3.2.3. Accurate assessment of clonality requires ~50K reads

An important application of single-cell sequencing to early diagnostics in cancer is to identify the cells in a tumor that form clones (i.e. groups of cells with shared genomic alterations). Here, we assess the effect of reducing read depth on our ability to identify cells that form a clone. Using the prostate biopsy samples from (Alexander 2016), we downsampled the number of reads per cell and clustered cells by their shared profiles using Gaussian Mixture Models (GMM). To score our ability to identify the cells that are clonal, we use the following approach at each read depth level: for each set of cells S_i identified (algorithmically and by visual inspection) by (Alexander 2016) as being clonal, we find the corresponding GMM cluster T_i (Methods) and define the score as $|S_i \cap T_i| \ / \ |T_i|$, the ratio of the number of true clonal cells found in T_i divided by the total number of cells found in cluster T_i. In other words, the score improves if we identify the correct clonal cells, and decreases if the clonal cells we identify are within a cluster that contains many non-clonal cells. Surprisingly, for the purpose of identifying clonal cells, 50K reads per cell was sufficient for the samples we analyzed (Figure 3.4). Since this analysis does not require accurate CNV profiles or phylogeny structure, it is expected that we would need fewer reads per cell.

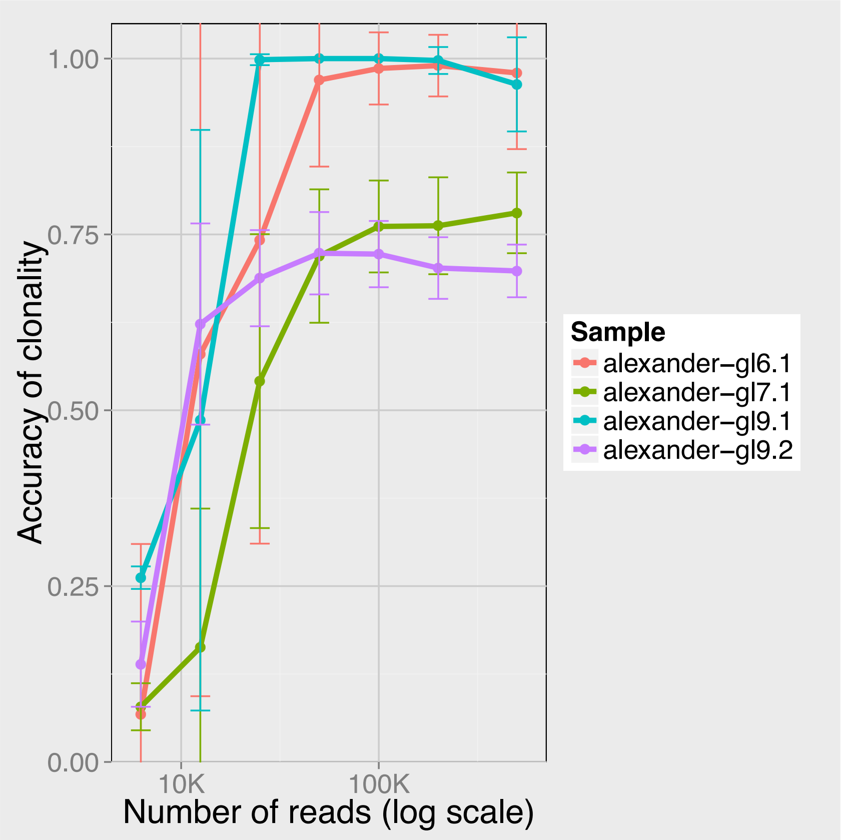
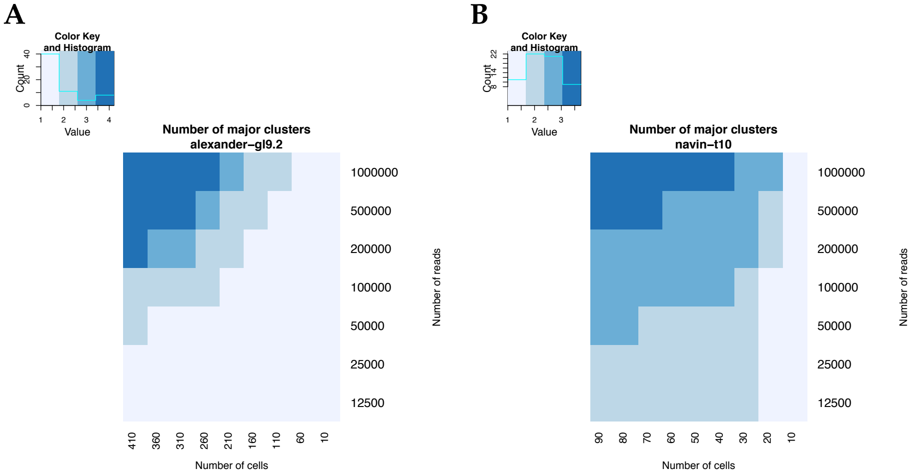


Figure 3.4: Generally, 50K reads per cell are sufficient to accurately identify clonal cells. Note that the *x* axis is on a log scale.

3.2.4. Tradeoff between sequencing more cells and sequencing at greater depth of coverage

An important parameter in single-cell sequencing experiments is the number of cells to sequence in a given sample. Here we assess the tradeoff between sequencing more cells and sequencing at greater depth of coverage, by varying both read depth and number of cells. At each downsampling depth, we generate 100 random subsets of reads; for each subset, we further generate 100 random subsets of cells—for a total of 10^4 randomizations for each combination. At every randomization, we evaluate our ability to accurately infer population structure, as measured by two statistics: (1) the number of major clusters as determined by a Gaussian Mixture Model and the Bayesian Information Criterion; (2) and the conservation of each cluster's content as determined by the Jaccard Index (Methods).

Across all tumor samples, we find that our ability to infer population structure is much more sensitive to the number of cells than the read depth (Figure 3.5A-B and Supplementary Figure 3.5). Whereas removing reads from a sample generally leads to an approximately logarithmic decrease in accuracy, removing cells exhibits an approximately linear decrease. From this analysis, we conclude that given a fixed budget and sufficient read depth, sequencing more cells is preferable to sequencing at greater depth, especially for heterogeneous tumor samples.



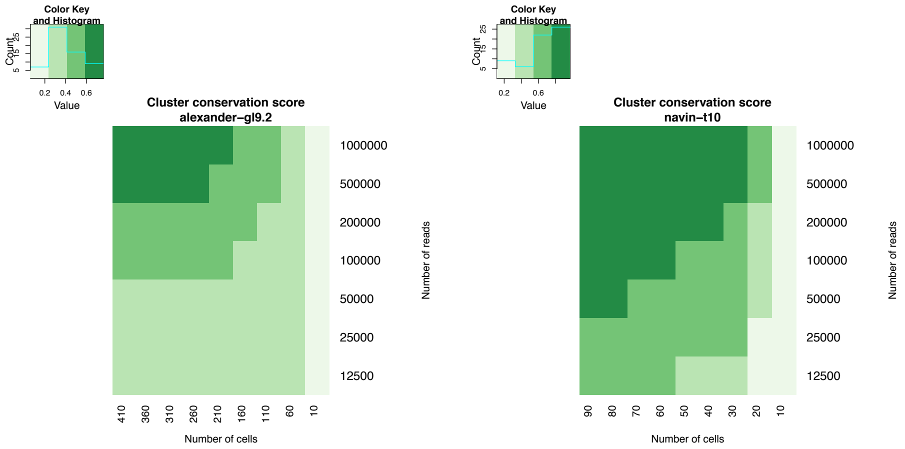


Figure 3.5: Approximate log-linear relationship in the tradeoff between sequencing more cells and at greater depth. **(A)** Heatmaps of two population-structure measures on a prostate tumor biopsy (*alexander-gl9.2*) illustrates that a heterogeneous tumor would benefit more from sequencing more cells than sequencing at greater depth. **(B)** Similar results were observed for a breast tumor (*navin-t10*).

3.2.5. Drizzle: a software tool for assessing single-cell data quality

We developed the software tool Drizzle to automate millions of subsampling experiments on single-cell sequencing data (Figure 3.1 and Methods). Drizzle takes as input sequencing reads in FASTQ format (one file per cell), maps reads to the reference genome (Langmead and Salzberg 2012), and removes PCR duplicates (Li et al. 2009) or reads with poor mapping quality. Drizzle also supports the analysis of data published in the Sequence Read Archive (SRA) database (Leinonen et al. 2011); given a list of accession IDs, Drizzle automatically downloads raw sequencing reads, and fetches important metadata (expected file size, read length, and single- or paired-end reads). Once the download is complete, Drizzle converts the SRA files into FASTQ format in memory, and proceeds with mapping the reads to the human reference genome. For each cell, Drizzle generates a segmented copy-number profile using variable-sized bins (Navin et al. 2011; Baslan et al. 2012) at 10 supported bin size configurations (25kb to 6.4Mb). Using the mapped reads from each cell, Drizzle randomly samples reads at 15 different levels of read depth. For each sampling depth, Drizzle estimates the accuracy of inferring a cell's copy-number profile, constructing phylogenetic trees using all cells from a sample, assessing divergent subclonal populations, and assessing population structure using various combinations of read depth and number of cells. Although we observe several trends across the different datasets, we anticipate that the exact range of optimal parameters may vary by tissue type, tumor grade, and whole-genome amplification approach. As such, here we provide Drizzle as a suite of open-source software tools to enable investigators to evaluate the data quality of pilot single-cell sequencing experiments.

3.3. Discussion

Here we present the first large-scale single-cell sequencing analysis of copy-number variation across a wide spectra of cancer sub-types and experimental procedures. Overall, we find that most cells exhibit a plateau behavior, where sequencing above a certain read depth yields diminishing returns, whereas the signal rapidly deteriorates below that threshold (Figure 3.2). We show that 1M reads per cell is generally sufficient to accurately recover a cell's copy-number profile at differing bin resolutions (Figure 3.2A-B). For most cells, this threshold is well below the original read depth (**Error! Reference source not found.**C), which suggests that using lower coverage sequencing would achieve similar results at reduced cost. Next, we introduce a quality control measure (average normalized area under the curve at different starting conditions) that can help readers assess whether the initial number of reads per cell is sufficient in their own preliminary data (**Error! Reference source not found.**D). Furthermore, we show that ~500K reads per cell is sufficient for accurate phylogeny construction in tumors of different tissue type (breast and prostate), heterogeneity, and clinical grade (Figure 3.3). To identify highly diverged clonal sub-populations, we find that ~50K reads is sufficient (Figure 3.4). Next, we explored the number of cells required for accurate population structure inference. For heterogeneous samples, we find that sequencing more cells at a sufficient depth yields greater insight into population structure than sequencing fewer cells at greater depth (Figure 3.5).

Our results have important implications for future single-cell sequencing studies that aim to study copy-number alterations. In cancer research, the single-cell CNV profile can be used in biopsies to detect malignancy and its evolution in the patient. In such studies, doing a first pass sequencing at low depth of coverage is a cost-effective approach for identifying the major clones in a tumor. Thereafter, a clone of interest can be studied in greater detail by pooling together the libraries of that clone and re-sequencing. This would allow for an in-depth characterization of the clone, including the identification of single nucleotide polymorphisms (SNPs) and smaller copy-number events.

Our results show a trend that is consistent across a variety of tissue types and whole genome amplification methods. Nevertheless, we recommend that investigators perform similar downsampling analyses outlined in this paper for new projects. To this end, we developed Drizzle, an open-source software that enables readers to assess the quality of their preliminary single-cell sequencing data for copy-number profiling and phylogeny. This pipeline is available open-source on GitHub.

3.4. Methods

3.4.1. Obtaining the data

We obtained the SRA accession codes from each of the published single-cell studies in Table 3.1. Using Drizzle, we generated a list of all cells to download, and manually removed cells that were exome-sequenced. In total, we analyzed 2,826 cells.

3.4.2. Drizzle

Drizzle is packaged as a collection of R and Bash scripts. Each module included in the package is listed below, along with a description. Also refer to the flowchart in Figure 3.1A.

1-download_sra.sh

The first module takes as input a list of accession IDs from the SRA database (supports SRX and SRP accession IDs). Using the NCBI E-utilities/EFetch API, it fetches metadata about each cell (download URL, total file size, and timestamp); as well as information about experimental design (paired-end or single-end, read length, etc.). To accelerate file download, our pipeline uses the Aspera protocol (instead of HTTP or FTP). In our experience, this resulted in ~5 fold improvement in download speeds (~50 MB/s instead of ~10 MB/s). Once each file downloaded, the pipeline verifies that the size and timestamp of the file on the local storage matches that in the database to ensure the download was successful. Finally, each cell's SRA file is decompressed into a FASTQ file (or 2 files for paired-end sequencing) containing a list of sequencing reads. Also included is an optional step to generate quality control reports for each SRA file using FastQC (Andrews 2010).

2-fastq_to_bed.sh

Reads in the FASTQ files are mapped to the human genome (hg19) using Bowtie2 (Langmead and Salzberg 2012) (using the correct parameters for whether a cell has paired-end or single-end data). The SAM format output from Bowtie2 is converted to BAM format and filtered using Samtools (Li et al. 2009) to remove low-quality reads (< 25 mapping quality) and PCR duplicates. The output from that step is then converted to a more compact BED file with only 2 columns (chromosome and start position), which is then compressed to BED.GZ using gzip compression. This script also supports basic read trimming options to remove barcodes.

2-sra_to_bed.sh

Similar as previous script, but also converts the SRA to FASTQ before launching read alignment. To speed up the process significantly, FASTQ files are not stored on disk but instead only maintained in memory when needed.

3-downsample.sh

We perform the downsampling analysis as follows: at each step, we remove a certain portion of a cell's reads and reconstruct the segmented copy-number profile using a streamlined version of the Ginkgo implementation. To account for biases in mappability across the genome, we bin the genome using variable-sized bins, as described previously (Navin et al. 2011). This step supports several options for bin sizes: (1) choose a constant bin size, e.g. 500kb; (2) use a bin size that yields specified average number of reads per bin, e.g. ~100 reads/bin; (3) number of reads per bin to maintain is defined by the specified start bin size.

4-fig2-cnvprofile.R

Using downsampled data from above, measure the accuracy of reconstructing a cell's CNV profile by calculating the correlation between the ground truth and the downsampled segmented CNV profiles. As with previous steps, the desired bin size and the number of reads per bin can be specified. This module also allows the user to reproduce all the plots shown in Figure 3.2, which includes: (1) correlation as a function of number of reads; (2) quality score as function of starting number of reads per cell; (3) accuracy as a function of bin size; (4) histogram of elbow for each cell; and (5) fraction of cells with > 0.9 correlation as a function of read depth.

5-fig3-phylogeny.R

Measures the accuracy of reconstructing the phylogenetic tree of a sample (useful for samples where clonal structure is expected such as tumors). Trees are built using Neighbor Joining (see below for details). As with previous steps, the desired bin size and number of reads per bin can be specified. Furthermore, the user can define which tree distance metric to use when building Neighbor Joining trees (euclidean, maximum, manhattan, canberra, binary or minkowski). This step produces the plots show in Figure 3.3 that illustrate the accuracy of tree construction as a function of read depth.

6-fig4-clonality.R

Launches the clonality analysis where our ability to accurately identify highly diverged clones is assessed at various read depths. This step also produces the plot shown in Error! Reference source not found.

7-fig5-tradeoffs.R

To evaluate the tradeoffs between sequencing more cells and sequencing at greater depth, this module progressively removes both reads and cells from the analysis, and calculates the resulting number of major clusters and the Jaccard Index, and plots the heatmaps shown in Figure 3.5.

3.4.3. Elbow

To identify the critical threshold of a curve (or "elbow"), we use the Kneedle algorithm (Satopää et al. 2011), as follows. First, we connect the first and last point from the curve and define a line D that passes through these two points. Next, we connect each data point to line D using a line E that is perpendicular to D; the elbow occurs at the data point that maximizes length of line E. When calculating the elbow for all cells, we discard cells that are diploid, which we define to be cells where >90% of autosomal bins are at copy-number 2.

3.4.4. Building phylogenetic trees

In the field of cancer genomics, a powerful application of single-cell sequencing is to identify copy-number alterations, and use those events to infer the phylogenetic history of a tumor (Navin et al. 2011; Ni et al. 2013). A common approach to single-cell phylogeny is to first calculate the pairwise distance (e.g. Euclidean, Manhattan) between all integer copy-number profiles and build the tree using algorithms such as hierarchical clustering and Neighbor Joining. However, using integer copy-number profiles can lead to additional bias, as discussed here. First, in the presence of very large events, such as a whole chromosome loss, the distance between any two cells will be largely a function of whether those events are observed, and will overlook smaller, possibly equally important events. Instead, starting from a cell’s segmented copy-number profile, we infer its integer copy-number state across the genome and identify the resulting “breakpoints”, i.e. bins that indicate a transition in the integer copy-number state. The resulting binary vector of each cell is further modified to remove breakpoints that occur at chromosome starts and ends (we don’t consider transitions between chromosomes as a breakpoint), and within sex chromosomes (in multi-patient phylogenetic trees, this ensures the clustering is based on shared events and not gender). Finally, we build phylogenetic trees by calculating the Manhattan distance between all pairs of cells and using the Neighbor Joining algorithm (Saitou and Nei 1987).

3.4.5. Comparing Trees

To calculate the effect of using fewer reads per cell on the topology of the tree, we generate a ground truth phylogenetic tree using a fixed number of reads per cell. For this analysis, we set this parameter to 1 million reads per cell; this threshold was chosen such that most cells have a read depth > 1M reads. At each read depth level, we built a tree and calculated the “distance” between the two trees using the Branch Length Score distance (See Choosing an appropriate tree distance metric below).

3.4.6. Estimating the number of major cell clusters

To assess the tradeoff between sequencing more cells and sequencing at higher depth, we evaluate the effect of using fewer cells on our ability to accurately reconstruct population structure. As a proxy for population structure, we use the number of major cell clusters and a score that estimates the conservation of cluster contents. Although we could in principle make use of similar tree comparison techniques as described below, this would require making comparisons between trees with different numbers of leaves, yet tree distances are only defined between trees that share the same leaves. Although one could prune extraneous leaves, this is likely to generate artifacts: when downsampling to low number of cells, it is much easier to accurately reconstruct a tree with 10 leaves than a tree with 300 leaves. Instead, to estimate the number of major clusters, we first trim the data from breakpoints that are seen in no cells, and remove cells that contain no breakpoints. Next, we perform a Principal Component Analysis on the cell breakpoint profiles to reduce the dimensionality of the data. To find clusters of cells with similar copy number profiles, we fit the data to a Gaussian Mixture Model of k clusters (Fraley and Raftery 2002; Fraley et al. 2012), where k ranges from 1 to 10. The best k is chosen as the one that minimizes the Bayesian Information Criterion. Finally, we evaluate cluster content preservation using the Jaccard Index. Please refer to **Chapter 4**'s Methods section for details on the Gaussian Mixture Model procedure.

3.4.7. Choosing an appropriate tree distance metric

3.4.7.1. Introduction

To calculate the effect of using fewer reads on the topology of the tree, we generate a phylogenetic tree before and after downsampling, and calculate the "distance" between the two trees. Although several measures have been developed for calculating the distance between two unrooted trees (Felsenstein and Felenstein 2004), there are certain pitfalls so care must be taken when choosing a distance measure, as discussed here.

3.4.7.2. Robinson-Foulds Distance (also known as: Symmetric Difference or Partition Metric)

The Robinson-Foulds distance measures the number of branches that are not shared between two trees T_1 and T_2 (Robinson and Foulds 1981). Specifically, each internal branch in a tree partitions its leafs into two groups; the distance is defined as the number of partitions that are seen in one tree but not in the other.

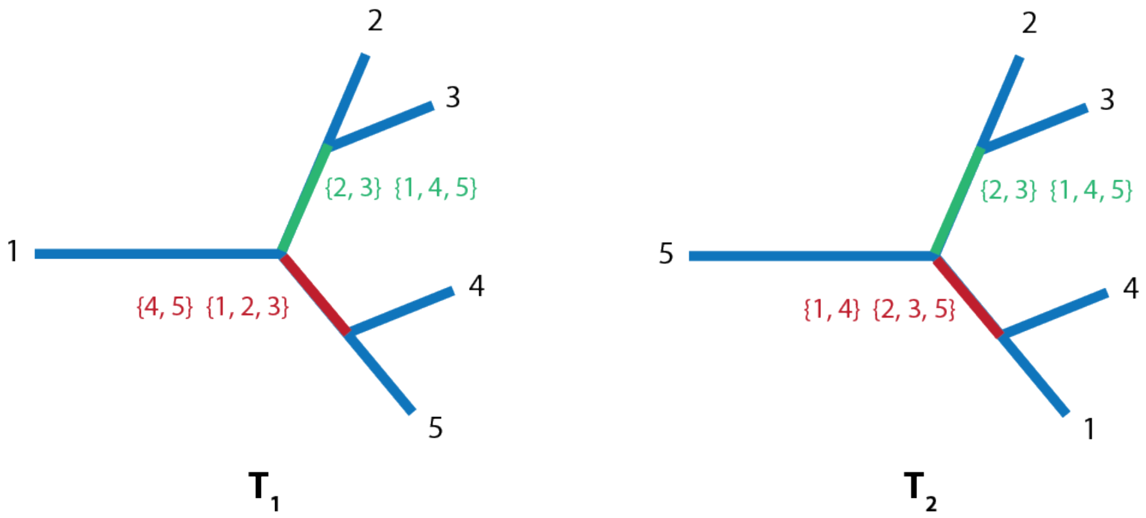


Figure 3.6: The Robinson-Foulds distance is defined as the number of inner branches not shared between two trees. In this example, the green internal branches are shared between the two trees, whereas the red internal branches are not, thus giving a Robinson-Foulds distance of 2; i.e. 50% of branches are shared.

For example, consider the trees shown in Figure 3.6. Out of a total of 4 inner branches, 2 branches are shared between the trees, whereas the remaining 2 are not shared. Hence, the Robinson-Foulds distance between T_1 and T_2 is 2; in other words, $1 - \frac{2}{4} = 50\%$ of internal branches are shared between the two trees. Algorithmically, to compare trees T_1 and T_2, we loop through each internal branch from both trees and add 1 to the distance if the branch is seen in one tree but not the other, and 0 if it is seen in both trees:

$$\mathrm{RF}(T_1, T_2) = \sum_{\substack{internal \\ branch\ b}} |I_b^{(T_1)} - I_b^{(T_2)}|$$

$$\text{where } I_b^T = \begin{cases} 1, & \text{branch } b \text{ found in tree } T \\ 0, & \text{otherwise} \end{cases}$$

This distance ranges from 0 to twice the number of internal branches (maximum distance is when every branch from T_1 is not found in T_2, and vice-versa). We define the Robinson-Foulds score as the percentage of internal branches maintained in both trees:

$$\mathrm{RFS}(T_1, T_2) = 1 - \frac{\mathrm{RF}(T_1, T_2)}{2\,(n-3)}$$

Note that the total number of internal branches in a tree is $n - 3$. As illustrated in Figure 3.7, an unrooted tree with $n = 3$ leafs has 0 internal branches, a tree with $n = 4$ leafs has 1 internal branch, and a tree with $n = 5$ leafs has 2 internal branches. By induction, we can show that a tree with n leafs has $n - 3$ internal branches, for $n \geq 3$.

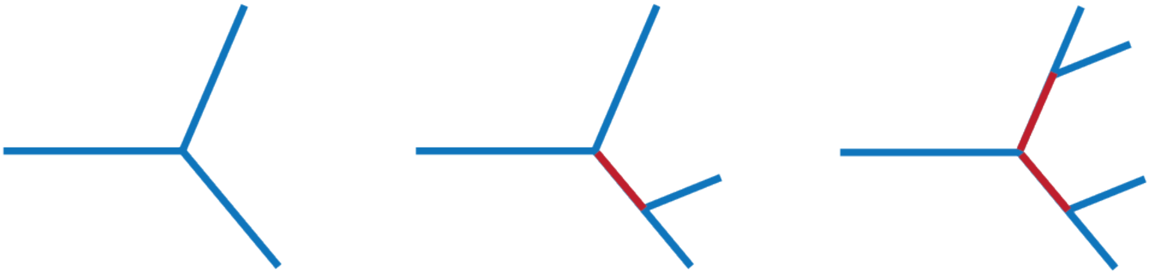


Figure 3.7: A tree with $\boldsymbol{n}$ leafs has $\boldsymbol{n-3}$ internal branches.

3.4.7.3. Branch Length Distance

The Branch Length distance is a generalization of the Robinson-Foulds distance that takes branch lengths into account and considers all branches, not only internal branches (Kuhner and Felsenstein 1994). Instead of a 0 penalty if a branch is present, the BL distance uses a penalty based on the squared differences in branch lengths:

$$\text{BL}(T_1, T_2) = \sqrt{\sum_{branch\ b} \left(T_1^{(b)} - T_2^{(b)}\right)^2}$$

Note that if branch lengths in both trees are 1, then the Branch Length Distance and the Robinson-Foulds distance are equal.

Next, we define the Branch Length Score by normalizing the Branch Length Distance by the maximum distance between the two trees, which is when every branch in T_1 is not found in T_2, and vice-versa:

$$\text{BLS}(T_1, T_2) = 1 - \frac{\text{BL}(T_1, T_2)}{\sqrt{\|T_1\|^2 + \|T_2\|^2}}$$

3.4.7.4. Quartet Distance

A quartet is defined as any set of four leafs in a tree. The quartet distance is defined as the proportion of quartet subtrees whose topology is not preserved between two trees. As shown in Figure 3.8, a quartet can have one of three possible topologies.

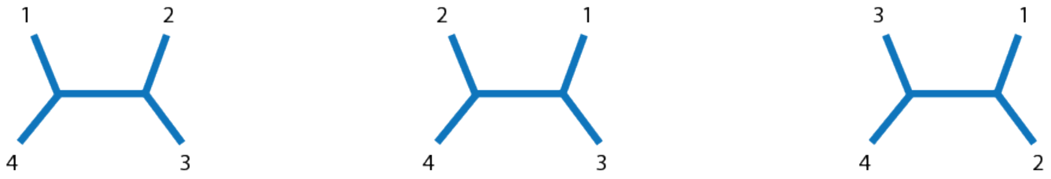


Figure 3.8: A quartet can have one of three tree topologies *(assuming all nodes in the tree have degree 3)*

For two trees T_1 and T_2 that share N leafs, the Quartet Distance is therefore defined as:

$$\text{QD}(T_1, T_2) = \sum_{quartet\ Q} \mathbf{1}_Q(T_1, T_2) \ / \ \binom{N}{4}$$

$$\text{where } \mathbf{1}_Q(T_1, T_2) = \begin{cases} 0, & \text{quartet } Q \text{ has same topology in both trees} \\ 1, & \text{otherwise} \end{cases}$$

3.4.7.5. Other tree distances

So far, we discussed the three most commonly used tree distances, although other ones have also been suggested, including the *Nearest-Neighbor Interchange Distance*—which calculates how many steps are needed to go from one tree topology to the other—and the *Path-Length-Difference Metric*—which measures the conservation of the number of branches that separate each pair of leafs. For an extensive discussion on tree distances, refer to (Felsenstein and Felenstein 2004).

3.4.7.6. Which distance is most appropriate?

In principle, many of the distances described above could be used to compare trees. Here we explore the usefulness of each, especially within the context of building phylogenies of single-cell copy-number data obtained from tumors.

1) The effect of minor differences in topology

Although one would expect the Robinson-Foulds score to be a good measure of topology difference between two trees, in practice we find it to be extremely sensitive to minor differences. To illustrate this issue, consider the example shown in Figure 3.9, where we wish to compare two trees that are almost identical, except for leaf #6, whose position is incorrectly deduced while building tree T_2.

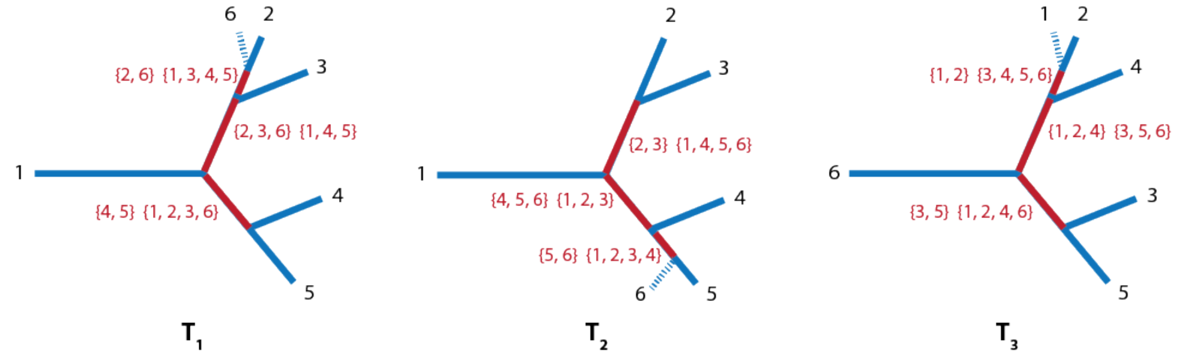


Figure 3.9: The Robinson-Foulds distance is very sensitive to minor changes. In this example, trees $\boldsymbol{T_1}$ and $\boldsymbol{T_2}$ have the maximum possible distance according to the Robinson-Foulds distance, despite their clear similarities.

Although the trees are clearly very similar, their Robinson-Foulds distance is 6, which is the maximum possible distance for those two trees! In other words, according to the Robinson-Foulds metric, the distance between T_1 and T_2 is the same as the distance between the two very dissimilar trees T_1 and T_3, where the contents of the main clusters are completely lost. In contrast, the Quartet Distance and Branch Length score are much less sensitive to this situation (see Table 3.2). Other issues with the Robinson-Foulds metric have also been raised previously. For instance, when randomly sampling trees from 11 leafs, (Penny et al. 1982) found that over 80% of tree pairs had the maximum possible Robinson-Foulds distance.

	$\boldsymbol{d(T_1, T_2)}$	$\boldsymbol{d(T_1, T_3)}$
Robinson-Foulds	100%	100%
Quartet Distance	67%	80%
Branch Length Score	46%	62%

Table 3.2: The Robinson-Foulds distance is very sensitive to small changes. Trees $\boldsymbol{T_1, T_2, T_3}$ refer to **Figure 3.9**.

2) *<u>The importance of branch lengths</u>*

Next, consider the case where a tumor is composed of several clusters of cells. Although each cluster may contain cells that share much of their copy-number profiles, minor differences—for example, due to errors during whole genome amplification, sequencing or copy-number calling—may affect the tree building process. As a result, although cells in a cluster will remain near each other after downsampling reads, the order they occur in the tree may vary every so slightly, as shown in Figure 3.10. For such situations, it is important to use a distance that incorporates branch lengths into the calculation. For example, not considering branch lengths would greatly increase the distance between the trees ($RF = 2/6, QD = 3/15$). However, if we use a distance that takes branch lengths into account—such as the Branch Length Score—the fact that the cells have remain near each other will be reflected in the smaller branch lengths, and therefore the distance between the trees will be much smaller.

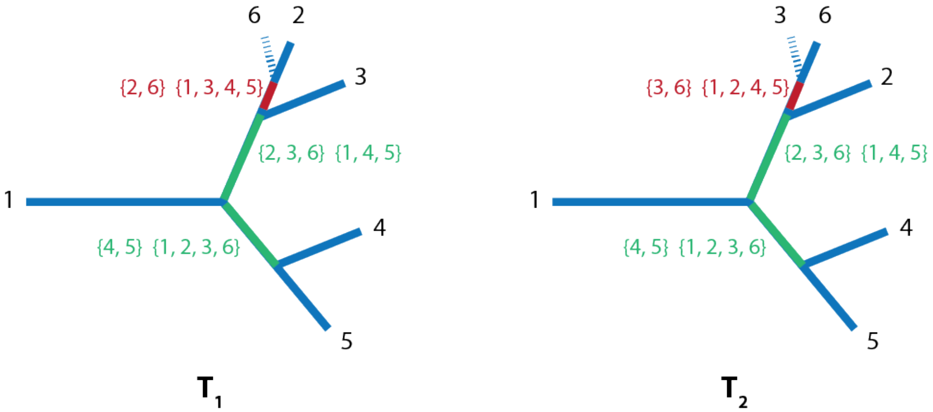


Figure 3.10: Minor changes in the order of cells within a cluster will affect the distance between the trees more significantly when branch lengths are not taken into account.

3) *<u>Trees with large number of leafs</u>*

In many of the samples we use in our analysis, trees can have >500 leafs, which exacerbates the two previous issues, especially for the Robinson-Foulds distance. In addition, although the Quartet Distance was robust to the situation highlighted in Figure 3.9, it would be as sensitive as the Robinson-Foulds distance if the tree harbored large clusters of size $\gg 4$. In contrast, since the Branch Length Score takes branch lengths into account, the small branch lengths within cell clusters indicate that those cells are very similar, and therefore the calculated distance will be smaller.

4) The most appropriate tree distance

Taken together, the most appropriate tree distance for our analysis is the Branch Length Score, given that it is robust to minor variation in the tree and takes branch lengths into account. We use the BLS implementation found in the `phangorn` R package (Schliep 2011), and normalize appropriately as discussed above. Note that the `ape` R package (Paradis et al. 2004) also implements the BLS, but only takes inner branch lengths into account (all bipartitions are of size $>$ 2), whereas phangorn takes all branches into account (bipartitions include those of size 1, i.e. the leafs of the tree).

3.4.8. Is the BLS a real "metric"?

In the previous sections, the words "distance" and "metric" were used loosely to mean a quantity that represents how far apart two trees are from each other. Here we formally prove that the BLS is a metric in the mathematical sense. There are four requirements for a function to be defined as a "metric":

1. *The distance between a tree and itself should be* 0*:*

$$\mathrm{BL}(T_1, T_1) = 0$$

2. *The function must be symmetric, i.e. the distance between tree T_1 and T_2 is the same as the distance between tree T_2 and T_1:*

$$\mathrm{BL}(T_1, T_2) = \mathrm{BL}(T_2, T_1)$$

3. *The distance between two trees must be non-negative:*

$$\mathrm{BL}(T_1, T_2) \geq 0$$

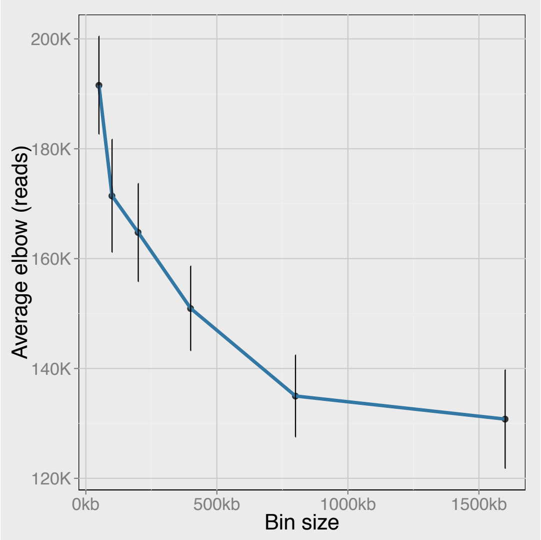
4. *The distance function must satisfy the triangle inequality:*

$$\mathrm{BL}(T_1, T_3) \leq \mathrm{BL}(T_1, T_2) + \mathrm{BL}(T_2, T_3)$$

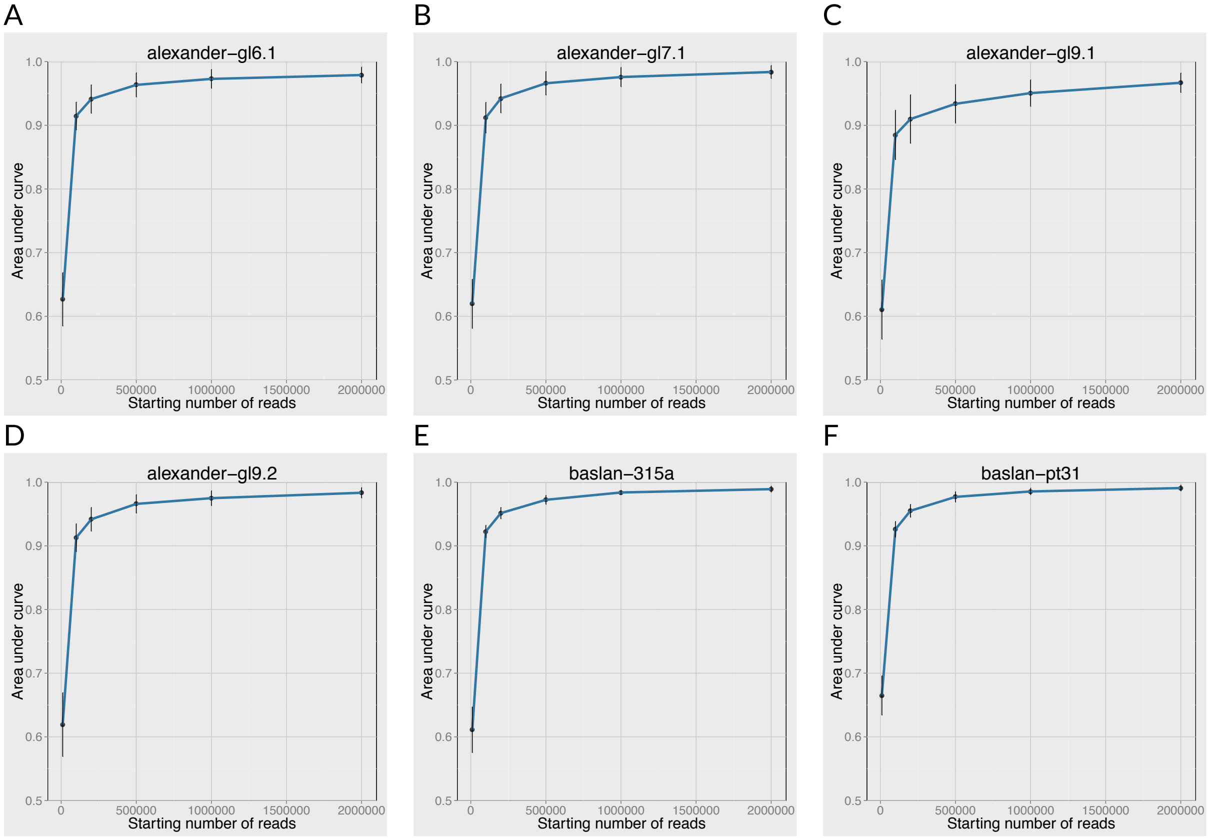
From the mathematical formulas shown in previous pages, the Branch Length Score always satisfies requirements #1 and #2. Next, requirement #3 will always be satisfied as long as the trees being compared contain the same leafs. As for requirement #4, it is satisfied by the following proof:

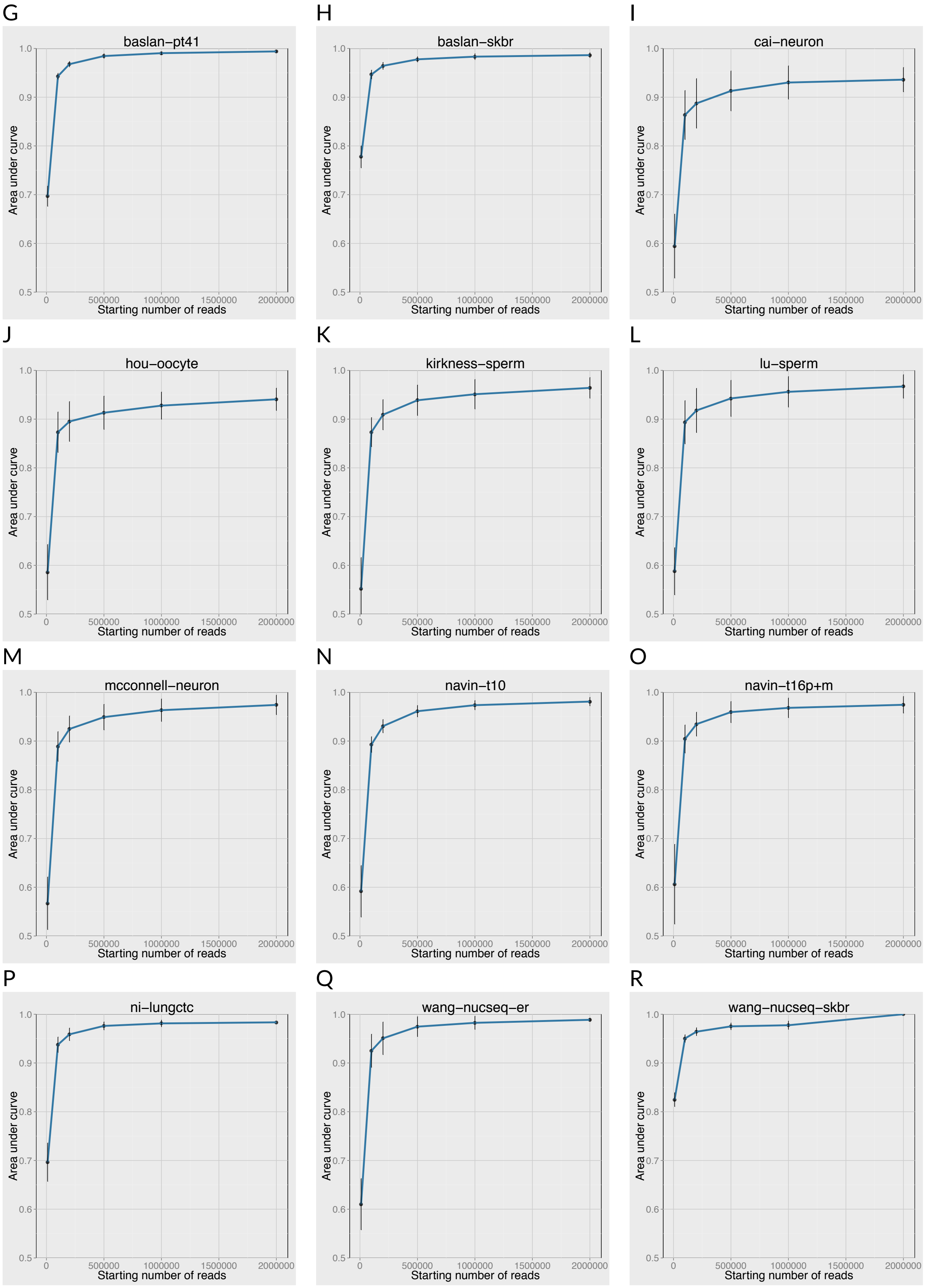
$$\begin{aligned}
\mathrm{BL}(T_1, T_2) + \mathrm{BL}(T_2, T_3) &= \sqrt{\sum_{branch\ b} \left(T_1^{(b)} - T_2^{(b)}\right)^2} + \sqrt{\sum_{branch\ b} \left(T_2^{(b)} - T_3^{(b)}\right)^2} \\
&\geq \sqrt{\sum_{branch\ b} \left(T_1^{(b)} - T_2^{(b)} + T_2^{(b)} - T_3^{(b)}\right)^2} \quad \text{by Cauchy Schwarz inequality} \\
&= \sqrt{\sum_{branch\ b} \left(T_1^{(b)} - T_3^{(b)}\right)^2} \\
&= \mathrm{BL}(T_1, T_3)
\end{aligned}$$

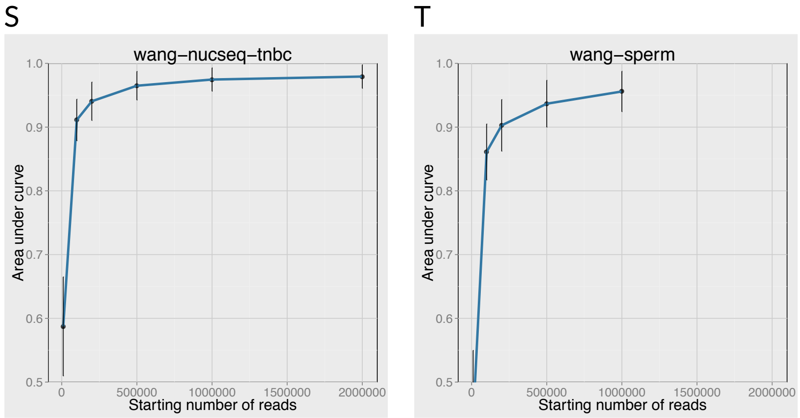
3.5. Supplementary Figures



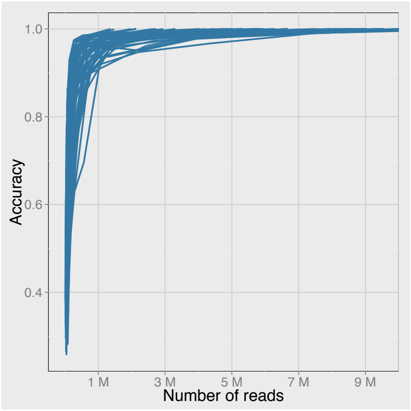
Supplementary Figure 3.1: The number of reads needed for the same level of accuracy as a function of bin size (shown for *navin-t10* sample). See **Online Methods** for description of elbow algorithm.



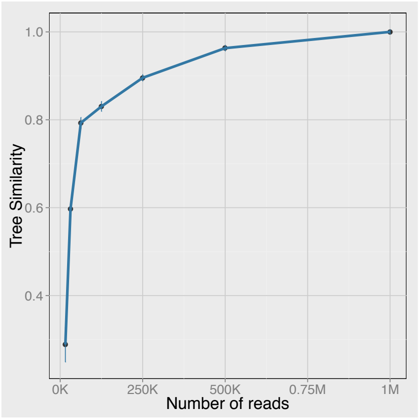

G
baslan−pt41
H
baslan−skbr
I
cai−neuron
J
hou−oocyte
K
kirkness−sperm
L
lu−sperm
M
mcconnell−neuron
N
navin−t10
O
navin−t16p+m
P
ni−lungctc
Q
wang−nucseq−er
R
wang−nucseq−skbr
Area under curve
Starting number of reads
0.5
0.6
0.7
0.8
0.9
1.0
0
500000
1000000
1500000
2000000



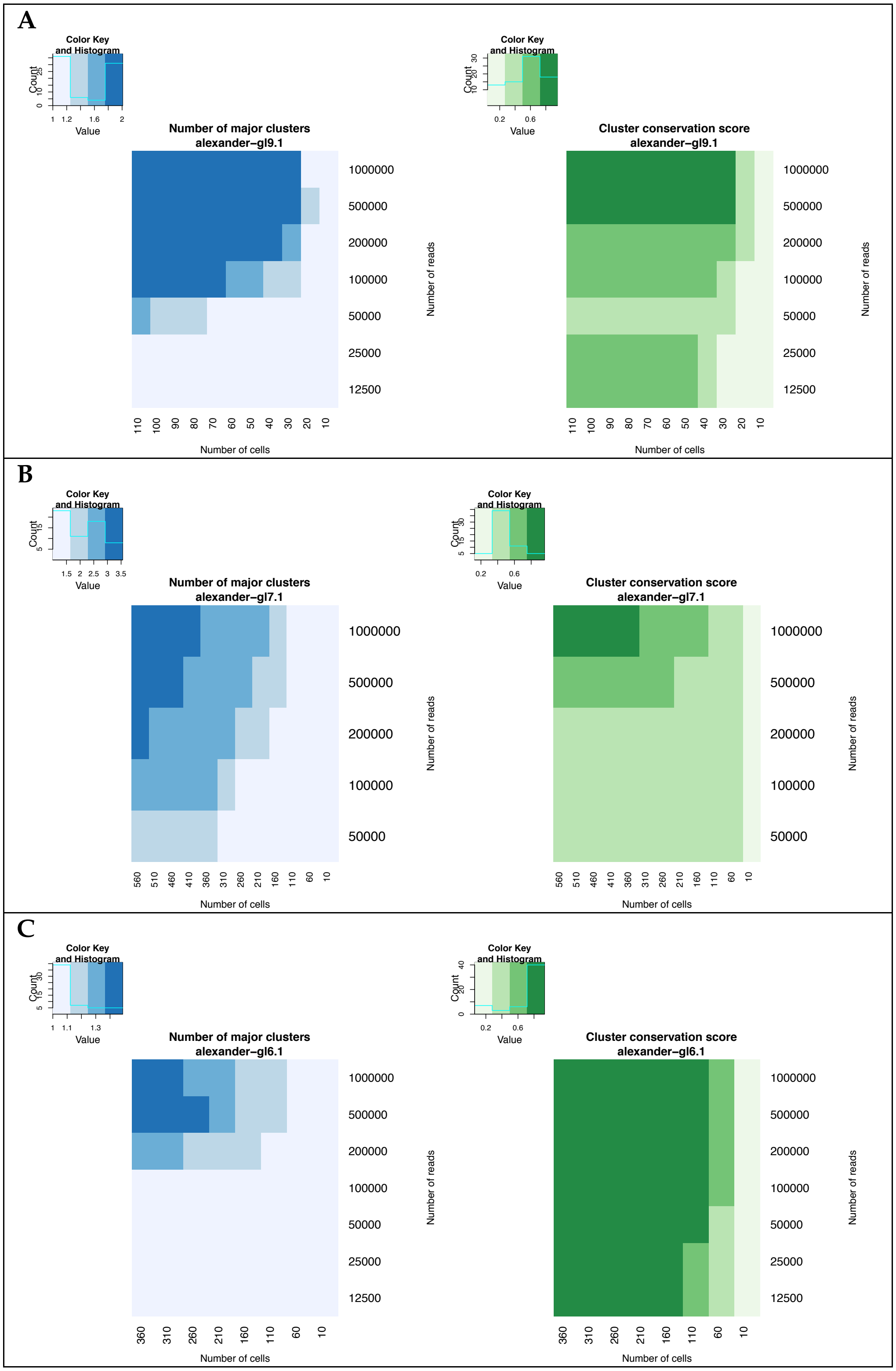
Supplementary Figure 3.2: Repeating the AUC analysis shown in **Figure 3.2F** for all samples in our study shows similar results across samples.



Supplementary Figure 3.3: To test that the plateau observation isn't due to a low number of reads per bin, we repeated the analysis shown in **Figure 3.3B**, but using a bin size of 100kb and, at each downsampling step, remove half the reads and use a bin size twice the size. The data shown here is from the *navin-t10* sample (diploids removed).



Supplementary Figure 3.4: To verify that the result in **Figure 3.3** is not due to low number of reads per bin during segmentation, we repeat the analysis while maintaining the same number of reads per bin at each sampling step, and observe a similar trend. The data shown here is from the *navin-t10* sample.



Supplementary Figure 3.5: We repeated the analysis shown in **Figure 3.5** on several other samples. Note that in *alexander-gl6.1*, the number of major clusters is close to 1 because most cells are not similar except for a sub-clone of 4 cells.

CHAPTER 4

Algorithms for single-cell copy-number phylogeny and clone detection

4.1. Introduction

Recent technological developments have paved the way for sequencing DNA at single-cell resolution for the study of complex biological systems (Navin and Hicks 2011; Wigler 2012; Shapiro et al. 2013; Navin 2014). In particular, single-cell DNA sequencing has enabled the analysis of heterogeneous tumors (Navin et al. 2011; Ni et al. 2013; Dago et al. 2014; Wang et al. 2014; Alexander et al. In preparation), whose population structure would otherwise be obscured by bulk sequencing of millions of cells.

Studying the population structure of tumors helps disentangle intra-tumor heterogeneity by identifying groups of cells with shared copy-number events (clones). Inferring the presence of early clones is a promising application of single-cell sequencing for early cancer diagnosis in the clinic (Alexander et al. In preparation). Furthermore, a better understanding of the genetic profile of the various clones present in a tumor can help direct future treatment options, as intra-tumor heterogeneity plays an important role in drug resistance (Saunders et al. 2012). By retracing the relationships between the observed clones, investigators can also uncover early mutations to better identify driver genes, and potentially predict future clinical outcome or tumor recurrence (Urbschat et al. 2011; McGranahan and Swanton 2015).

Despite the growing interest in single-cell cancer genomics (Shapiro et al. 2013), the question of which approach to use for building phylogeny from single-cell copy-number sequencing data remains unanswered. Our goal in this chapter is to develop tools to more accurately reconstruct tumor evolution. Specifically, this chapter will address current challenges in single-cell CNV phylogeny, and present a method for identifying the informative breakpoints in a tumor, finding the major clones in a tumor and building a phylogenetic tree from these clones.

4.2. The challenges of single-cell CNV phylogeny

Although several algorithms are available to infer phylogeny from SNP allele frequency obtained from single-cell exome sequencing (Hou et al. 2012; Li et al. 2012; Kim and Simon 2014), few algorithms exist for building phylogenetic trees of tumor cells from single-cell copy-number sequencing data. A common approach in single-cell CNV studies is to use distance-based methods such as Hierarchical Clustering or Neighbor Joining (Navin et al. 2011; Baslan et al. 2012; Ni et al. 2013). For example, (Ni et al. 2013) performed single-cell sequencing of circulating tumor cells from lung cancer patients, and constructed a phylogenetic tree by taking the pairwise Euclidean distance between integer copy-number profiles. However, in tumors with very large CNV events (e.g. chromosome arm deletions), such an approach would heavily bias the clustering by placing more weight on large CNV events and less weight on smaller, potentially equally important events.

For example, consider the copy-number profiles in Figure 4.1A, where cells 1 & 2 share three copy-number events while cells 3 & 4 share three different copy-number events. Building a phylogenetic tree from integer copy-number profiles places cells 2 & 3 together since the large deletion takes precedence in the Euclidean distance calculation (Figure 4.1B). Instead, using breakpoints—locations where there is a change in the copy-number state of the cell—reflects the number of events shared rather than their size (Figure 4.1C-D).

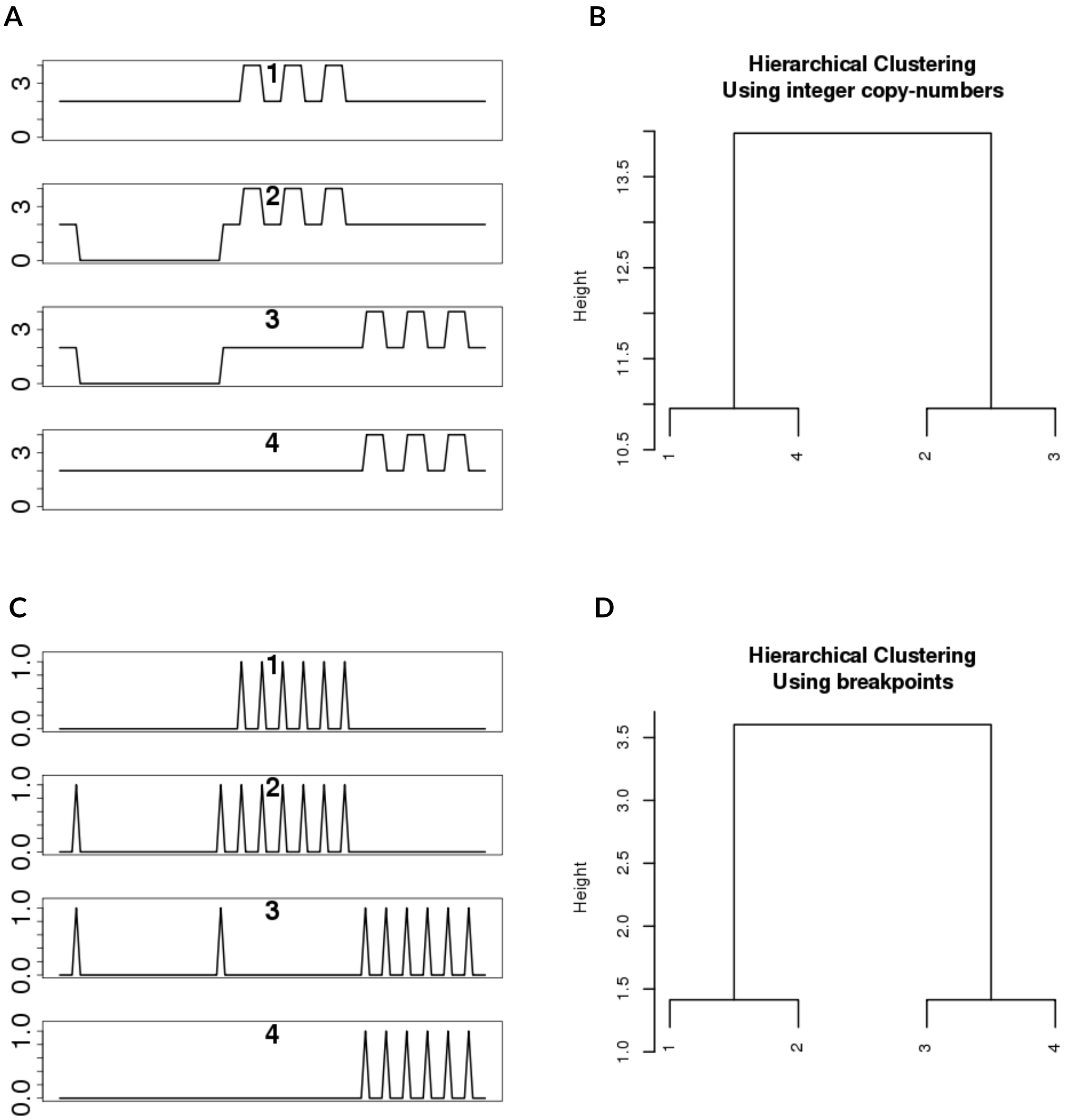
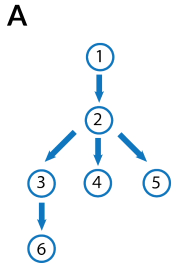


Figure 4.1: (A-B) Constructing a phylogenetic tree (average linkage, Euclidean distance) using the integer copy-number profiles places a lot more weight on large events and can miss small events that could represent events in important genes. **(C-D)** By contrast, using the breakpoint profiles to build the phylogenetic tree does not show the same CNV size bias.

However, the use of breakpoint profiles may not suffice for accurate population structure inference, depending on the choice of phylogenetic methods. Although hierarchical clustering approaches were used in previous single-cell studies (Gangnus et al. 2004; Ulmer et al. 2004; Mathiesen et al. 2012; Heitzer et al. 2013; Ni et al. 2013; Melchor et al. 2014), such approaches can fail for complex population structures. To demonstrate this issue, consider the population structure of 6 cells in Figure 4.2A, a simplified version of the population structure observed in the *GL9.2* prostate tumor biopsy sequenced at single-cell level by (Alexander et al. In preparation). In this example, Cell 1 is a diploid cell, Cell 2 is the precursor cell, Cells 3, 4, 5 are derived directly from the precursor, and Cell 6 was further derived from Cell 3. The breakpoints profiles of the 6 cells can be represented as a matrix of breakpoints (columns) by cells (rows), as shown in Figure 4.2B.

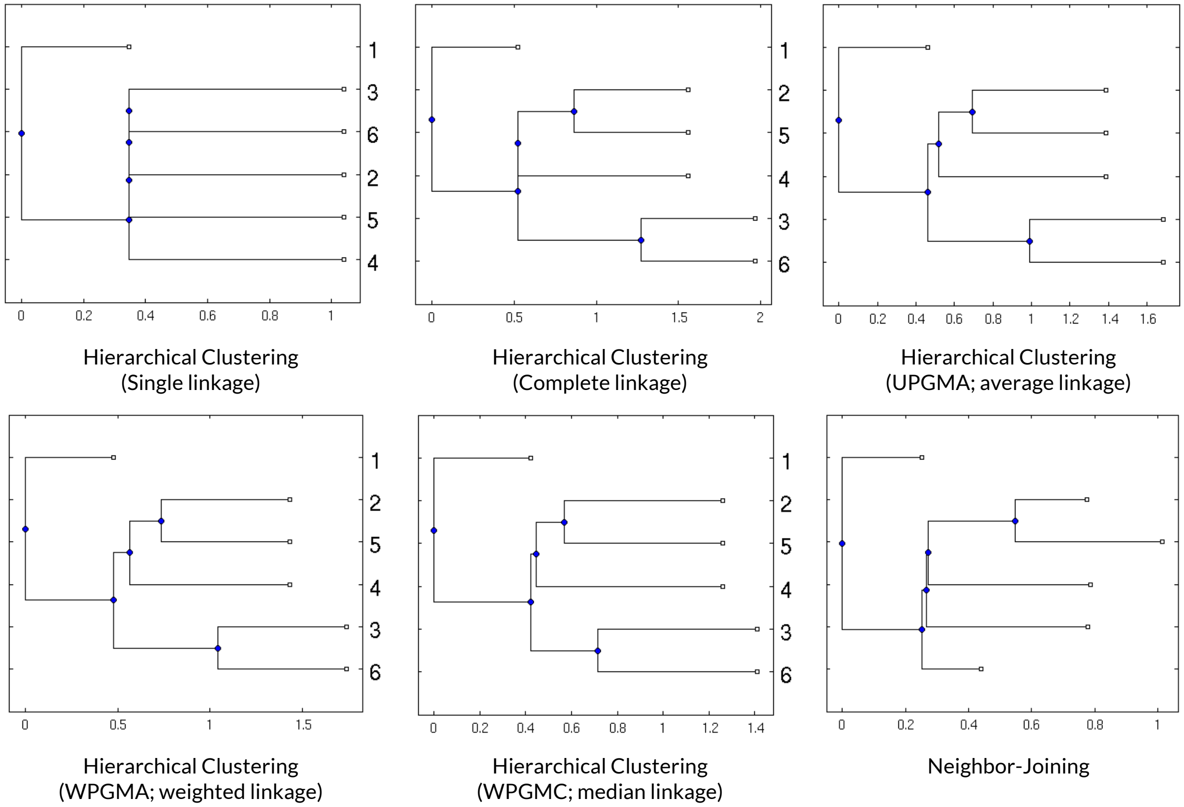


B

$$\begin{array}{cl} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{bmatrix} & \begin{array}{l} \text{Cell 1} \\ \text{Cell 2} \\ \text{Cell 3} \\ \text{Cell 4} \\ \text{Cell 5} \\ \text{Cell 6} \end{array} \end{array}$$

Figure 4.2: Sample population structure represented as a **(A)** tree and as a **(B)** matrix of breakpoint profiles.

Using several hierarchical clustering approaches and Neighbor-Joining, we built trees using the breakpoint information and re-rooted each tree by the diploid Cell 1. As shown in Figure 4.3, hierarchical clustering approaches (such as single linkage, complete linkage, UPGMA, WPGMA and WPGMC) are unable to reconstruct the tree of Figure 4.2A, whereas Neighbor-Joining comes closest. Although both Hierarchical Clustering and Neighbor-Joining are distance-based approaches, Neighbor-Joining does not assume a constant mutation rate across the branches of the tree, whereas methods such as UPGMA assume that the distance between the root and every leaf is the same (Felsenstein and Felenstein 2004).



Hierarchical Clustering (Single linkage)

Hierarchical Clustering (Complete linkage)

Hierarchical Clustering (UPGMA; average linkage)

Hierarchical Clustering (WPGMA; weighted linkage)

Hierarchical Clustering (WPGMC; median linkage)

Neighbor-Joining

Figure 4.3: Comparing different algorithms for reconstructing the tree shown in **Figure 4.2**A. In this example, Neighbor-Joining performs better than Hierarchical Clustering approaches. Note that the *x* axis is the *log* of the branch length, as several branches would otherwise overlap.

Non distance-based approaches for phylogeny also exist, such as Maximum Parsimony (MP), where the chosen tree is the one that minimizes the number of mutations required to explain the observed data. Studies comparing the accuracy of both methods concluded that Neighbor-Joining tends to be more accurate (Li et al. 1987; Sourdis and Nei 1988; Jin and Nei 1990). Other approaches include Maximum Likelihood (ML), where the tree maximizes the likelihood function based on a given model of evolution, and Bayesian approaches, where the tree instead maximizes the posterior probability (using MCMC to approximate the posterior distribution). Although such approaches outperform Neighbor-

Joining under high rates of divergence, Neighbor-Joining nonetheless provides accurate trees (Tateno et al. 1994; Kuhner and Felsenstein 1995; Kumar and Gadagkar 2000; Tamura et al. 2004; Mihaescu et al. 2007). Practically, Neighbor-Joining is a polynomial-time algorithm, which renders it very fast. In terms of running time, NJ far outperforms MP, ML, and Bayesian methods, sometimes requiring orders of magnitude less computation time, even for building trees with only 10 to 20 leafs (Kuhner and Felsenstein 1994; Williams and Moret 2003; Albright et al. 2014). This is a significant factor when building trees for hundreds of cells with information at thousands of sites across the genome (to detect 1Mb events, we need ~5,000 sites).

Tree building algorithms aside, inferring population structure from single-cell sequencing data is also challenging due to noise present in the data. This is in part caused by **(1)** Low sample quality due to complex sample preparation; **(2)** DNA contamination due to lysing of nuclei during cell isolation; **(3)** Wells accidentally containing multiple cells; **(4)** Uneven amplification during the whole-genome amplification step (Garvin et al. 2015); and **(5)** Fluctuations in coverage due to the low depth of coverage (< 1X) used in most single-cell CNV studies to reduce costs.

In tumors characterized by a founder clone comprising only a few cells (e.g. in early cancer diagnosis), these issues are exacerbated since the clone is much more difficult to identify. For example, consider the Neighbor-Joining tree obtained from single-cell sequencing of the *GL6.1* prostate tumor biopsy (Alexander et al. In preparation) plotted in Figure 4.4. Although nothing stands out strikingly, manual inspection of the ~500 copy-number profiles (Alexander et al. In preparation) reveals that cells mostly exhibit unique patterns of copy-number alteration, except for a small group of four cells (Figure 4.4 in red) from neighboring sectors of the tumor that share several copy-number events (Figure 4.4 Inset).

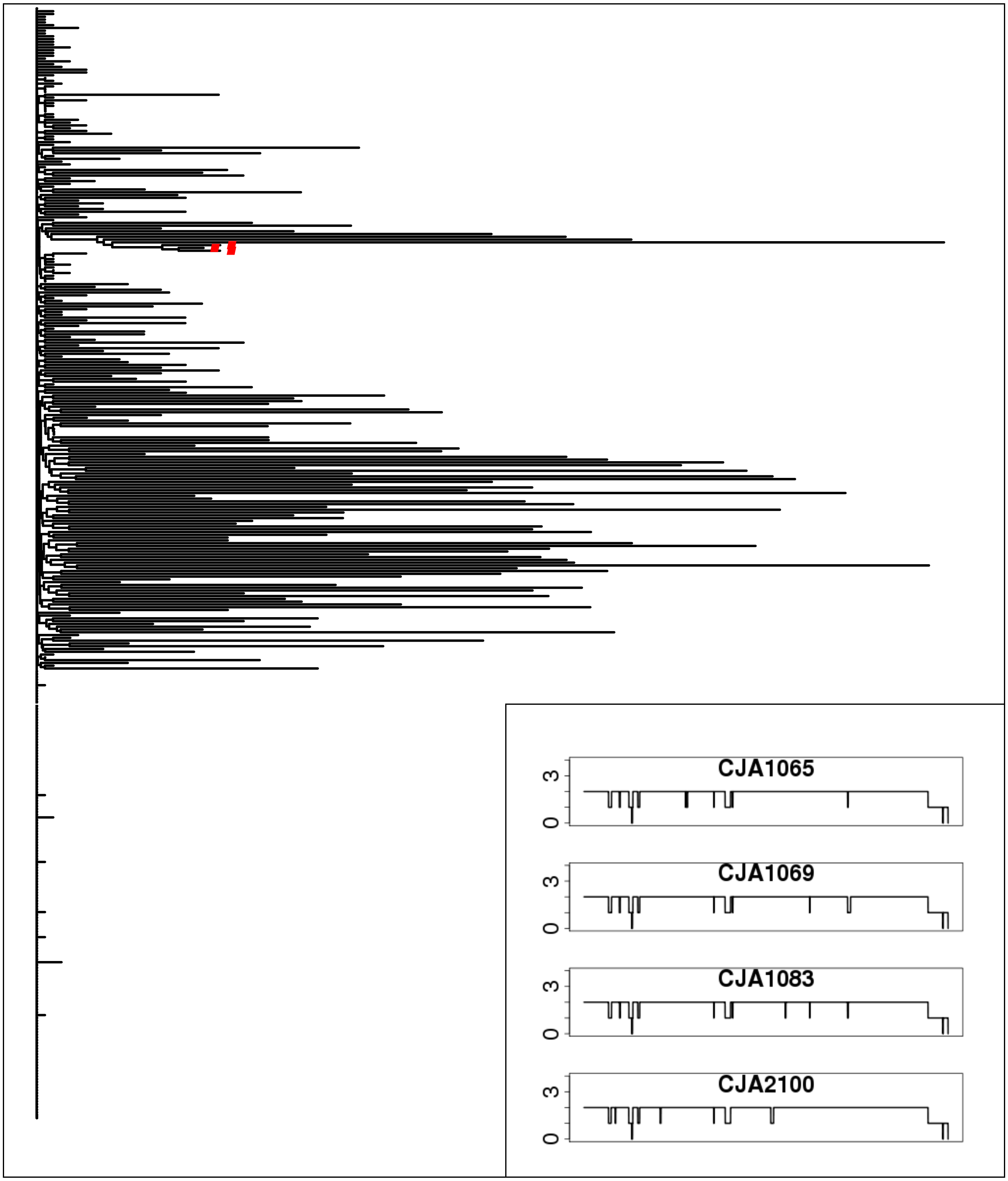


Figure 4.4: The Neighbor-Joining tree of the *GL6.1* sample, with a clone of cells in red. Inset: Integer copy-number profile of the cells present in the red clone.

From a clinical perspective, identifying such clones is important for early diagnosis. However, our analysis is overwhelmed by noisy, non-informative breakpoints that happen to be shared between a few cells due to chance. In this chapter, we address this issue by developing a statistical method for identifying the informative breakpoints that are important for defining clones.

4.3. Results

4.3.1. Single-cell CNV phylogeny using informative breakpoints

Following single-cell sequencing of a tumor, clones of cells can be identified by shared breakpoint patterns. However, for a large sample such as *GL6.1* (Figure 4.4), where 494 cells were sequenced, we expect to observe noisy breakpoints that occur in multiple cells. From a clinical perspective, this issue is further exacerbated by the fact that clones of interest could account for a small fraction of the tumor. For example, the clone in Figure 4.4 consists of only 0.8% of the total cells sequenced. To address this issue, we present a new statistical method for inferring informative breakpoints that are important for defining clones.

To distinguish informative breakpoints from noisy ones, we first discard bins where <2 cells exhibit a breakpoint. Next, we require informative breakpoints to “travel” in similar ways, as measured by the breakpoint-to-breakpoint covariance. Specifically, our input is a matrix of cells by bins, where a value of 1 denotes a cell that presents a breakpoint in a given bin, and 0 otherwise. Next, we compute the breakpoint-by-breakpoint covariances and denote a breakpoint as informative if the sum of its 3 largest covariances (not including selfcovariance) is significantly greater than expected by chance as follows. Significance is obtained by repeating the procedure on shuffled data (Methods and Pseudocode 4.2), where row and column sums are maintained. We use a p-value threshold of 0.01, normalized by the number of breakpoints to correct for multiple hypothesis testing (Pseudocode 4.1).

```
Data = $[d_{ij}]$                                $m \times n$ matrix of $m$ cells and $n$ breakpoints, where $d_{ij} = 1$
                                                    if cell $i$ has a breakpoint at bin $j$; $d_{ij} = 0$ otherwise.

// Perform randomizations
TopCov = [ ]
for i = 0 : N                                    $N$ is the desired number of randomizations.

    Obs = Data
    if i > 0                                     $i = 0$ uses original data; $i = 1 \rightarrow N$ shuffles the data
        Obs = shuffle(Data)                      Shuffle data (see Methods)

    Obs = Obs[ , which( colSums(Obs) < 2 ) ] = 0 Ignore breakpoints observed in only 1 cell (or no cells).
    Cov = covariance(Obs, diagonal = 0)          Calculate the covariance and set diagonal elements to 0.

    for j = 1 : nbBreakpoints
        TopCov[i, j] = Σ sort_desc(Cov[ , j]) [ 1 : 3 ]   For each breakpoint, find sum of 3 highest covariances.

// Find top breakpoints
TopBkpts = [ ]
for j = 1 : nbBreakpoints
    if pValue < 0.01 / nbBreakpoints             For each breakpoint, calculate the p-value, comparing
        TopBkpts[] = j                              the original data to the randomized data.

// Find informative breakpoints (close to highest "TopCov" score)
v = rep(0, nbBreakpoints)                        Vector where $v_i = 1$ if breakpoint $i$ is a top breakpoint.
v[TopBkpts] = 1

b = breakpoint for which $\sum_{1 \rightarrow b} v < \frac{1}{2}$   Require informative breakpoints to be some of the high
InformativeBkpts = which( v[1:b] == 1 )             scoring ones as determined by a running sum.
```

Pseudocode 4.1: Algorithm for finding most informative breakpoints.

Using single-cell sequencing data from eight prostate biopsy samples (Alexander et al. In preparation), we inferred the most informative breakpoints for each sample (Table 4.1). To validate our results, we compared the number of informative breakpoints found in each tumor to its severity. To grade the severity and prognosis of prostate cancers, pathologists use the *Gleason Score*, a grading system based on histological observations of the prostate biopsy. Specifically, the two most common patterns seen under the microscope are assigned a score between 1 and 5; the final score is a sum of both. Low Gleason scores (2–5) indicate that the tissue is normal whereas high Gleason scores (7–10) indicate that the tissue is cancerous. In principle, Gleason 6 tumors are cancerous but due to the risks of overtreatment and unnecessary radical prostatectomies, pathologists increasingly treat Gleason 6 tumors in a separate category, favoring active surveillance over treatment (Carter et al. 2012a; Nickel and Speakman 2012). As shown in Table 4.1, we find good concordance between the clinical grade of the tumor and the number of informative breakpoints.

Sample	Breakpoints	Clinical Grade
GL9.1	99	Gleason 9 (5 + 4)
GL9.2	39	Gleason 9 (4 + 5)
GL7.1	40	Gleason 7 (3 + 4)
GL7.2	48	Gleason 7 (3 + 4)
GL6.1	9	Gleason 6
GL6.2	3	Gleason 6
Pin.1	0	Prostatic Intraepithelial Neoplasia
Benign.1	0	Benign

Table 4.1: Number of informative breakpoint for each tumor, as determined by our algorithm in **Pseudocode 4.1**

Using only the informative breakpoints in the *GL6.1* prostate biopsy sample (Alexander et al. In preparation), we constructed a Neighbor-Joining tree. As shown in Figure 4.5B, the accuracy is improved tremendously compared to the tree built using all breakpoints (Figure 4.5A), as we are now able to identify the small clone that was previously identified via manual inspection.

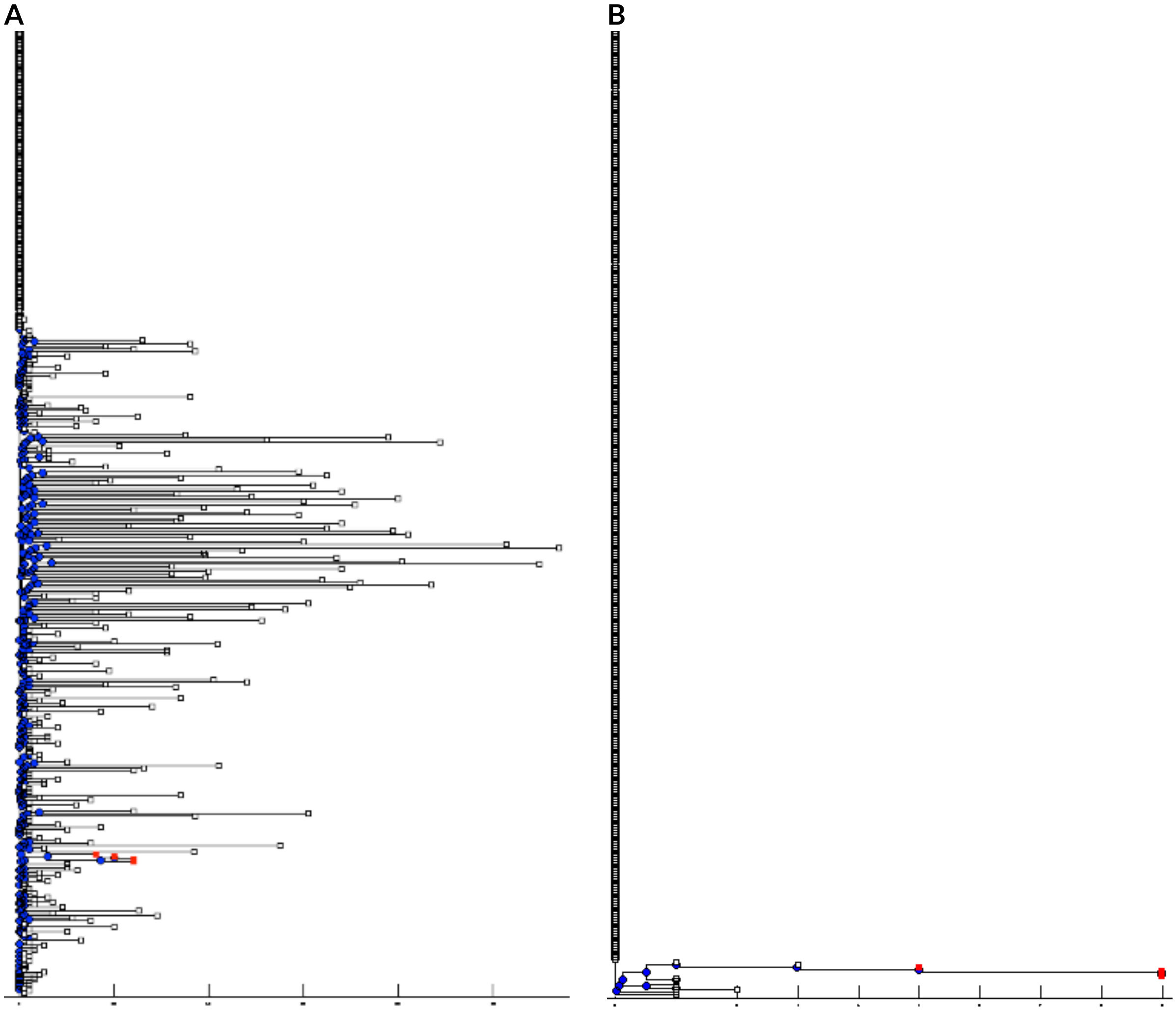


Figure 4.5: Neighbor-Joining tree of the *GL6.1* sample, with a clone of cells in red. **(A)** Tree built using all breakpoints. **(B)** Tree built using informative breakpoints, as determined by our algorithm in **Pseudocode 4.1**. Some of the diploid cells (top of image) were truncated for space considerations.

We repeated this procedure for *GL6*.2, another prostate sample with a clone of a few cells. Using only the top breakpoints, we are again able to build a tree that more easily identifies the clone (Figure 4.6); note that this is again the same clone previously identified by (Alexander et al. In preparation). This result is confirmed by plotting the integer copy-number profiles of the cells in that clone (Figure 4.7). Note that the phylogenetic tree accurately captures the relationships between these 8 cells, with 6 cells that have an additional copy-number event placed below the 2 cells they seem to derive from (Figure 4.6, Figure 4.7).

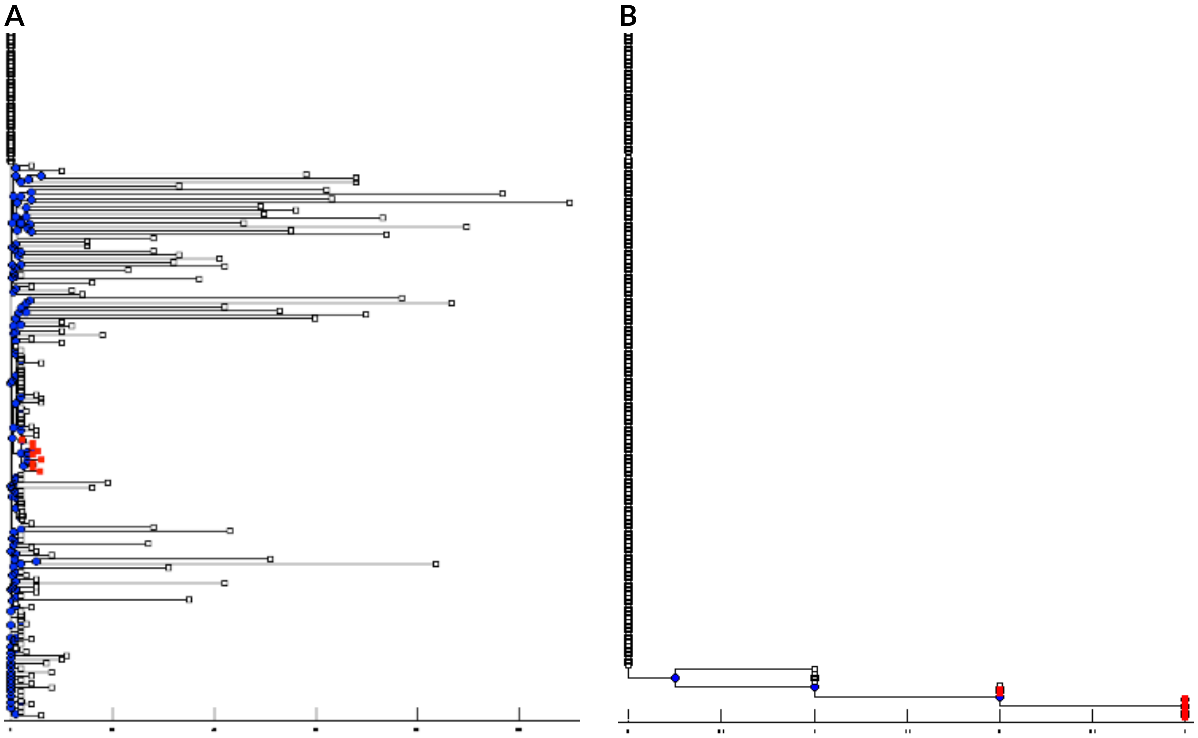


Figure 4.6: Neighbor-Joining tree of the *GL6.2* sample, with a clone of cells in red. **(A)** Tree built using all breakpoints. **(B)** Tree built using informative breakpoints, as determined by our algorithm in **Pseudocode 4.1**. Some of the diploid cells (top of image) were truncated for space considerations.

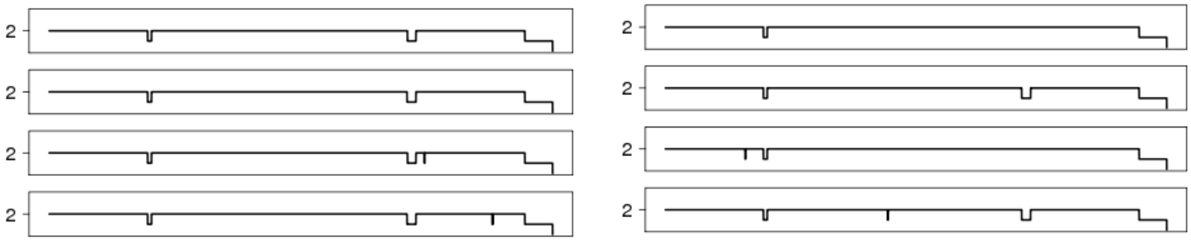


Figure 4.7: Integer copy-number profiles of cells in the clone highlighted in **Figure 4.6**.

We next repeated this procedure for the *GL9*.2 prostate biopsy (Alexander et al. In preparation), a higher-grade tumor with a much more complex population structure (manually inferred to be the structure shown in Figure 4.2). As shown in Figure 4.8, our algorithm allows us to better identify the clonal cells and infer their evolutionary history.

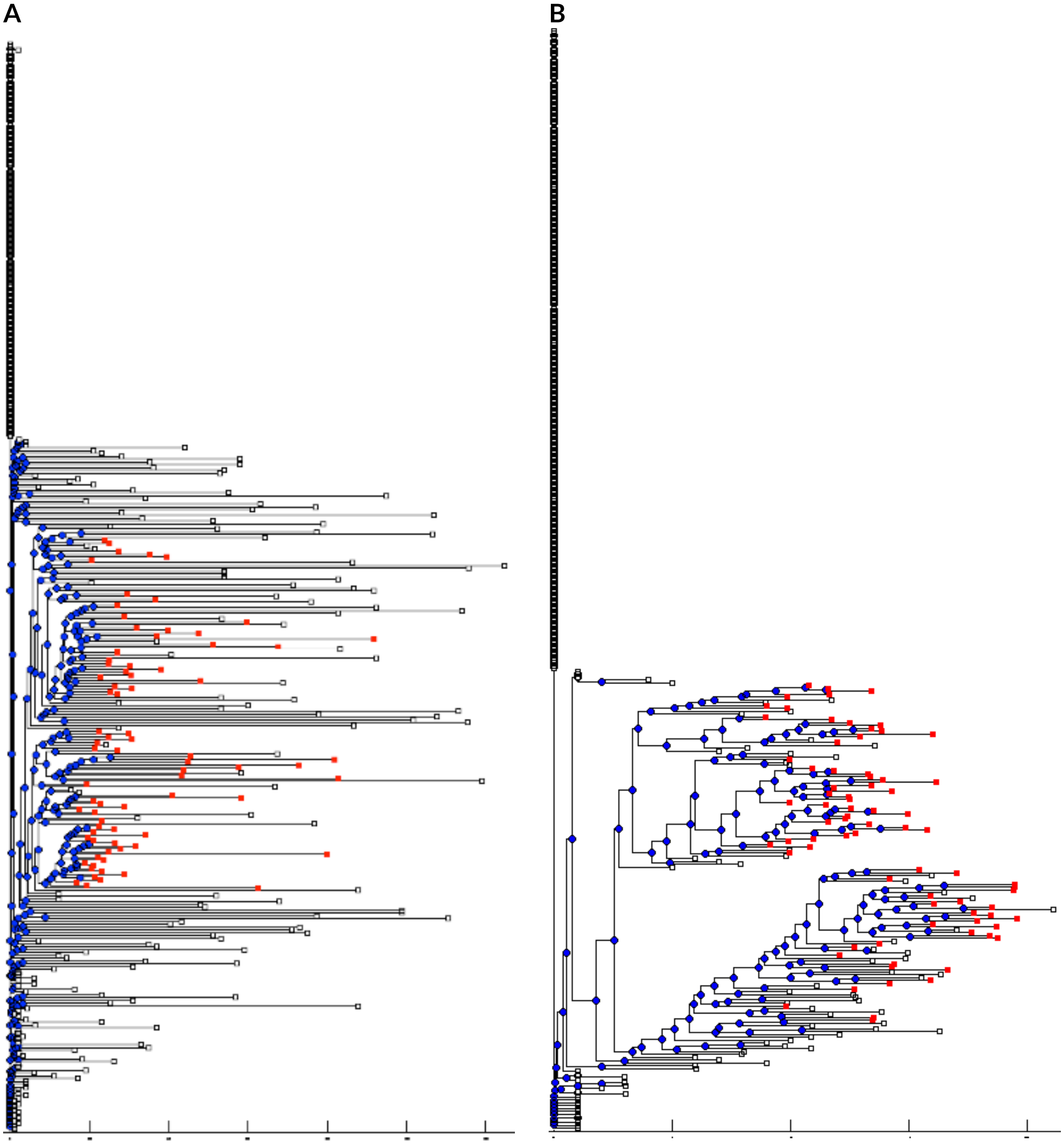


Figure 4.8: The Neighbor-Joining tree of the *GL9.2* sample, with clonal cells in red. **(A)** Tree built using all breakpoints. **(B)** Tree built using informative breakpoints, as determined by our algorithm in **Pseudocode 4.1**.

4.3.2. Inferring clonal evolution using informative breakpoints

Building phylogenetic trees using only the informative breakpoints allowed us to successfully identify—at a glance—the early clones present in the *GL6.1* (Figure 4.5) and *GL6.2* (Figure 4.6 and Figure 4.7) tumors, which are characterized by a single clone consisting of only a few cells. For more complex tumors such as *GL9.2*, although we're able to better retrace the history of the cells (Figure 4.8), it is not as trivial to infer the evolution of the clones themselves. In this section, we propose an approach to explicitly identify these clones by clustering cells based on shared patterns of informative breakpoints, with the end goal to build a tree of clones as in Figure 4.2.

Using Gaussian Mixture Models (GMMs), we identify clones by fitting our data (from only informative breakpoints) to a weighted sum of Gaussian distributions. We attempt to cluster the data using *G = 2 to 15* clusters and chose the clustering scheme that maximizes the Bayesian Information Criterion (Methods). Although here we use GMMs to cluster cells by similarity of informative-breakpoint profiles, it is also possible to cluster breakpoints to obtain a tree of breakpoint / mutation history. This clustering can also be used to identify the mutations that characterize each clone. For tree building purposes, we represent each cluster (or clone) as a single leaf, where the copy-number profile is the average breakpoint profile of all cells within that cluster. The idea here is to make use of the data from all cells in the cluster to average out the noise and construct a more robust CNV profile.

To validate our approach, we applied this methodology to the simulated single-cell CNV data from **Chapter 2**, where we simulated 3 major clones, each with 3 sub-clones of 10 cells for a total of 90 cells (Figure 4.9A). As shown in Figure 4.9B, our approach accurately reconstructs the simulated population structure.

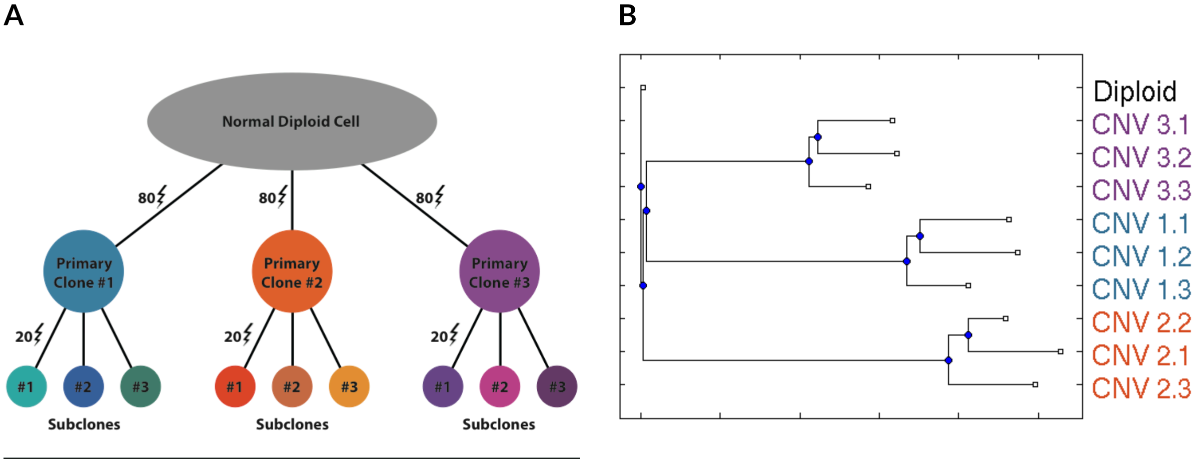


Figure 4.9: (A) Simulated population structure of 90 cells, as discussed in **Chapter 2**. **(B)** Neighbor-Joining tree built using average clone breakpoint profile accurately infers the expected population structure.

To further validate our approach within the context of non-simulated data, we attempted to reconstruct the population structure of a triple-negative breast tumor that was sequenced at single-cell resolution (Navin et al. 2011). As shown in Figure 4.10, we accurately reconstruct the expected tumor population structure (note that our approach further clusters the pseudo-diploid cells into a separate cluster).

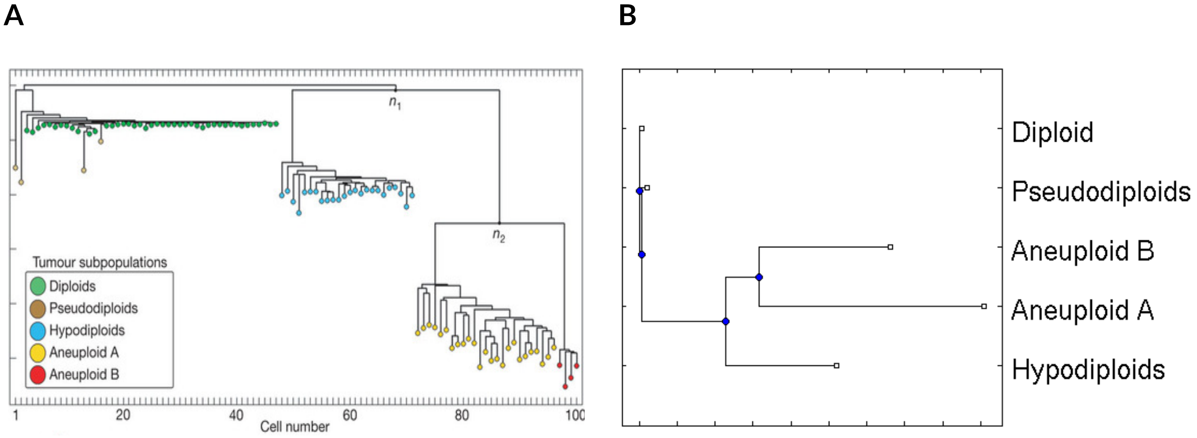


Figure 4.10: (A) Population structure of a triple negative breast cancer; figure reproduced from **(Navin et al. 2011)**. **(B)** Neighbor-Joining tree built using average clone breakpoint profile accurately recapitulates the clonal evolution. Note that the cluster names used in **B** are set according to where >90% of cells in that cluster originate.

Applying the same procedure to the *GL9.2* sample discussed in the last section, our algorithm identifies 5 clones. The Neighbor-Joining tree of those clones accurately infers the expected evolution of that sample (Figure 4.11).

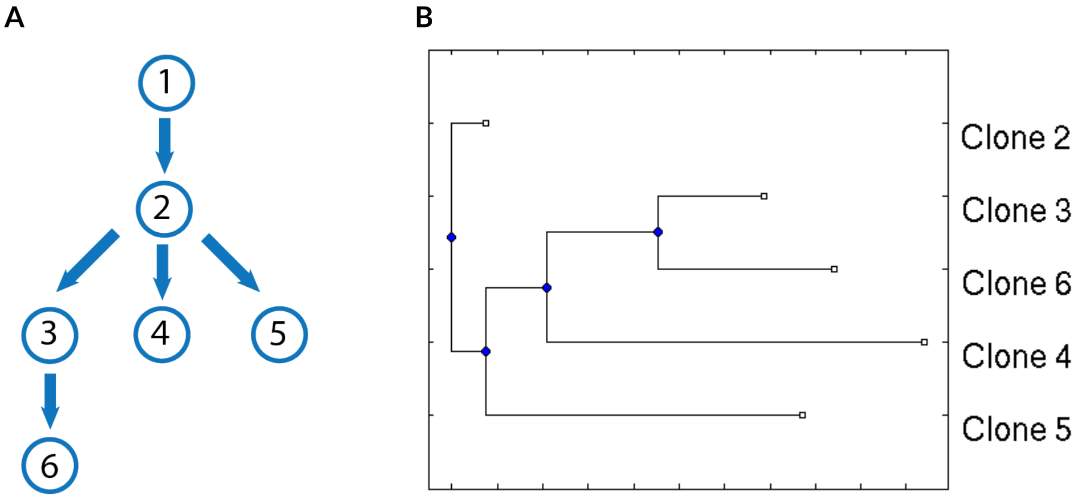


Figure 4.11: (A) Manually inferred clonal evolution of the *GL9.2* sample; reproduced from **Figure 4.2A** above. **(B)** The neighbor-Joining tree built using average clone breakpoint profiles accurately recapitulates the expected population structure.

4.4. Methods

4.4.1. Building trees

For each sample, Neighbor-Joining trees were constructed using MATLAB's `seqneighjoin()` function (Bioinformatics toolbox). Hierarchical clustering trees were built using MATLAB's `seqlinkage()` function.

4.4.2. Shuffling procedure

The shuffling of the breakpoint profiles is performed on a matrix of size $m \times n$ (m cells, n breakpoints). The simplest approach to randomizing the matrix would be to count the number of positions in the matrix where there is a breakpoint, and assign the same number of breakpoints at random positions within a new matrix of 0's. However, such a randomization would likely generate significance easily. A stricter approach would be to re-assign the breakpoints in the matrix such that row and column sums are preserved.

We implement this using a swapping approach (Pseudocode 4.2). For each pair of cells c_1 and c_2, we find the bins $\{b_i\}$ where cell c_1 has a breakpoint but where cell c_2 does not, and vice-versa. These breakpoints $\{b_i\}$ are then redistributed randomly to both cells. This ensures that we only perform 2 x 2 swaps that go from the sub-matrix configuration $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ to the configuration $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, thereby maintaining row and column sums in the overall matrix. By induction, it can be shown that for two binary matrices of same dimensions where row and column sums are fixed, there exists a finite number of 2 x 2 swaps that will transform one matrix into the other (Ryser 1987).

```
Data = $[d_{ij}]$                                      $m \times n$ matrix of $m$ cells and $n$ breakpoints, where $d_{ij} = 1$
                                                          if cell $i$ has a breakpoint at bin $j$; $d_{ij} = 0$ otherwise.

// Create list of breakpoints for each cell
bkpts = { }
for i = 0 : nbCells
    bkpts{$c_i$}.append( which(Data(i,:) == 1) )


// For each pair of cells, find breakpoints to swap
for each pair of cells $(c_1, c_2)$
    swaps12 = bkpts{$c_1$} \ bkpts{$c_2$}              The set difference $S_1 \setminus S_2 = \{x : x \in S_1$ and $x \notin S_2\}$.
    swaps21 = bkpts{$c_2$} \ bkpts{$c_1$}
    swaps = swaps12 ∪ swaps21                          Union of both sets defines breakpoint positions that can
                                                          safely be swapped in both cells.

    p = randperm(|swaps12| + |swaps21|)                Randomly assign these breakpoints between the 2 cells
    bkpts{$c_1$} = swaps[ p[1 : |swaps12| ] ]
    bkpts{$c_2$} = swaps \ bkpts{$c_1$}


// Generate shuffled data
DataShuffled = zeros(nbCells, nbBreakpoints)
for i = 0 : nbCells
    DataShuffled[ i, bkpts{$c_i$} ] = 1
```

Pseudocode 4.2: Algorithm used to randomize breakpoints in a matrix while maintaining the original row and column sums fixed.

Our algorithm also supports matrices with 0, +1, and -1, where the sign determines the direction of copy-number change at a breakpoint position. This is achieved simply by first performing swaps on the +1 breakpoints, followed by swaps on the -1 breakpoints. Since our procedure depends heavily on set operations, we use the MATLAB library *lightspeed* (Tom Minka, Microsoft Research) to speed up our calculations.

4.4.3. Computing covariances of large matrices

The data from (Alexander et al. In preparation) contains data for thousands of breakpoints across hundreds of cells. Calculating the breakpoint-by-breakpoint covariance matrices is therefore computationally prohibitive. Here we explored which language's implementation of the covariance function is most optimal. As a test case, we chose the scDNA-seq data from prostate sample *Pin.1* prostate sample (Alexander et al. In preparation), with 679 cells and 7611 breakpoints. For 100 iterations, we calculated the breakpoint-by-breakpoint covariance matrix. Overall, MATLAB gave the best performance, and was 1.3X faster than Python (numpy package) and 14X faster than R's built-in covariance function (Table 4.2).

Language	Average time (seconds)
R	28.18 ± 0.6119
Python	2.64 ± 0.0172
MATLAB	1.99 ± 0.0076

Table 4.2: MATLAB calculates breakpoint-by-breakpoint covariance matrices faster than R or Python, evaluated using a matrix with 679 cells and 7611 breakpoints (averaged over 100 trials).

4.4.4. Using Gaussian Mixture Models to cluster cells with shared breakpoint patterns

4.4.4.1. The math

To estimate the major clusters in a given sample, we fit our single-cell breakpoint profiles to a weighted sum of Gaussian distributions. The k-th Gaussian distribution has a weight of w_k, with mean vector $\boldsymbol{\mu_k}$, and covariance matrix $\boldsymbol{\Sigma_k}$. The goal is to find the mixture of Gaussians that maximizes the probability of sampling a D-dimensional point $\boldsymbol{x_i}$ from that distribution.

For a given cell i, the goal is to find the values of $\{ w_k, \boldsymbol{\mu_k}, \boldsymbol{\Sigma_k} \}$ that maximize the probability of observing the cell vector $\boldsymbol{x_i}$:

$$P(\boldsymbol{x_i} \mid \{w_k, \boldsymbol{\mu_k}, \boldsymbol{\Sigma_k}\}) = \sum_{k=1}^{G} w_k\, \mathcal{N}(\boldsymbol{x_i} \mid \boldsymbol{\mu_k}, \boldsymbol{\Sigma_k})$$

$$\text{where } \mathcal{N}(\boldsymbol{x_i} \mid \boldsymbol{\mu_k}, \boldsymbol{\Sigma_k}) = (2\pi)^{-D/2} |\boldsymbol{\Sigma_k}|^{-1/2} \exp\left\{-\frac{1}{2}(\boldsymbol{x_i} - \boldsymbol{\mu_k})^T \boldsymbol{\Sigma_k}^{-1} (\boldsymbol{x_i} - \boldsymbol{\mu_k})\right\}$$

Taking all cells together, the goal is to find the parameters that maximize the likelihood of observing the matrix $\mathbf{X}$:

$$P(\mathbf{X} \mid \{w_k, \boldsymbol{\mu_k}, \boldsymbol{\Sigma_k}\}) = \prod_{i=1}^{N} P(\boldsymbol{x_i} \mid \{w_k, \boldsymbol{\mu_k}, \boldsymbol{\Sigma_k}\}) = \prod_{i=1}^{N} \sum_{k=1}^{G} w_k\, \mathcal{N}(\boldsymbol{x_i} \mid \boldsymbol{\mu_k}, \boldsymbol{\Sigma_k})$$

For convenience, we maximize the log-likelihood:

$$\log \prod_{i=1}^{N} \sum_{k=1}^{G} w_k\, \mathcal{N}(\boldsymbol{x_i} | \boldsymbol{\mu_k}, \boldsymbol{\Sigma_k}) = \sum_{i=1}^{N} \log\left(\sum_{k=1}^{G} w_k\, \mathcal{N}(\boldsymbol{x_i} | \boldsymbol{\mu_k}, \boldsymbol{\Sigma_k})\right)$$

The best parameter values can be computed with the Expectation Maximization (EM) algorithm:

4.4.4.2. EM algorithm

At each iteration, the values of parameters $\{w_k, \boldsymbol{\mu_k}, \boldsymbol{\Sigma_k}\}$ are updated.

<u>After arbitrarily assigning each cell to a cluster, calculate initial parameter values</u>:

$$\widehat{w}_k = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}_{i\,\in\,k}$$

If cell 3 belongs to cluster 5: $\mathbf{1}_{3\,\epsilon\,5} = 1$

$$\widehat{\boldsymbol{\mu}}_{\boldsymbol{k}} = \frac{1}{\widehat{w}_k * N}\sum_{i=1}^{N} \boldsymbol{x_i} * \mathbf{1}_{i\,\in\,k}$$

$\widehat{w}_k * \mathrm{N} = \sum_{i=1}^{N} \mathbf{1}_{i\,\in\,k} = \#$ cells in cluster k

$$\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{k}} = \frac{1}{\widehat{w}_k * N}\sum_{i=1}^{N} (\boldsymbol{x_i} - \widehat{\boldsymbol{\mu}}_{\boldsymbol{k}})^T (\boldsymbol{x_i} - \widehat{\boldsymbol{\mu}}_{\boldsymbol{k}}) * \mathbf{1}_{i\,\in\,k}$$

<u>E-step</u>:

$$P(\boldsymbol{x_i} \in k) = \hat{\tau}_{ik} = \frac{\widehat{w}_k^{(t)} \;\mathcal{N}(\boldsymbol{x_i} \mid \widehat{\boldsymbol{\mu}}_{\boldsymbol{k}}^{(\boldsymbol{t})}, \widehat{\boldsymbol{\Sigma}}_{\boldsymbol{k}}^{(\boldsymbol{t})})}{\sum_{k=1}^{G} \widehat{w}_k^{(t)} \;\mathcal{N}(\boldsymbol{x_i} \mid \widehat{\boldsymbol{\mu}}_{\boldsymbol{k}}^{(\boldsymbol{t})}, \widehat{\boldsymbol{\Sigma}}_{\boldsymbol{k}}^{(\boldsymbol{t})})}$$

<u>M-step</u>:

$$\widehat{w}_k^{(t+1)} = \frac{1}{N}\sum_{i=1}^{N} \hat{\tau}_{ik}$$

$$\widehat{\boldsymbol{\mu}}_{\boldsymbol{k}}^{(\boldsymbol{t+1})} = \frac{1}{\widehat{w}_k^{(t+1)}} * \frac{1}{N}\sum_{i=1}^{N} \hat{\tau}_{ik}\, \boldsymbol{x_i}$$

$$\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{k}}^{(\boldsymbol{t+1})} = \frac{1}{\widehat{w}_k^{(t+1)}} * \frac{1}{N}\sum_{i=1}^{N} \hat{\tau}_{ik} \left(\boldsymbol{x_i} - \widehat{\boldsymbol{\mu}}_{\boldsymbol{k}}^{(\boldsymbol{t+1})}\right)^T \left(\boldsymbol{x_i} - \widehat{\boldsymbol{\mu}}_{\boldsymbol{k}}^{(\boldsymbol{t+1})}\right)$$

4.4.4.3. Noise term

In the analyses presented here, we make use of the *R* package *mclust* (Fraley et al. 2012), which also adds a first order Poisson noise term. Instead of maximizing:

$$\log P(\boldsymbol{X}|\,\{w_k,\ \mu_k, \mathbf{\Sigma_k}\}) = \sum_{i=1}^{N} \log\left(\sum_{k=1}^{G} w_k\, \mathcal{N}(\boldsymbol{x_i} \mid \mu_k, \mathbf{\Sigma_k}) \right)$$

it maximizes:

$$\log P(\boldsymbol{X}|\,\{w_k,\ \mu_k, \mathbf{\Sigma_k}\}) = \sum_{i=1}^{N} \log\left(\frac{w_0}{V} + \sum_{k=1}^{G} w_k\, \mathcal{N}(\boldsymbol{x_i} \mid \mu_k, \mathbf{\Sigma_k}) \right)$$

where the weights are selected so that $\sum_{k=0}^{G} w_k = 1$, and where V is the hypervolume of the dataset.

4.4.4.4. Choosing the best clustering regime

To choose the best clustering regime without user intervention, the procedure is performed using different number of clusters G, e.g. in the range 2 to 15 clusters. The best regime is the one that maximizes the Bayesian Information Criterion. For a solution with G clusters and with parameters estimated by EM to be $\{\widehat{w}_k,\ \boldsymbol{\widehat{\mu}_k}, \mathbf{\widehat{\Sigma}_k}\}$:

$$BIC = \log P\big(\mathbf{X} \mid \{\widehat{w}_k,\ \boldsymbol{\widehat{\mu}_k}, \mathbf{\widehat{\Sigma}_k}\}\big) - G \log{(N)}$$

Note that the $G \log(N)$ term penalizes solutions that contain too many clusters.

Tumor suppressive genes are conserved in syntenic clusters across the genome

This chapter has been reproduced with modifications from:

Aboukhalil R, Fendler B, and Atwal GS. **Kerfuffle: a web tool for multi-species gene colocalization analysis**. *BMC Bioinformatics* 14:22 (2013).

and

Fendler B* and Aboukhalil R*, Xue R, Esposito D, Powers S, Lowe SW and Atwal GS. **Tumor Suppressive Genes are Conserved in Syntenic Clusters.** *In preparation*.

5.1. Introduction

During the progression of tumors many large regions of the genome, encompassing multiple genes and regulatory sites, are either deleted or amplified (Beroukhim et al. 2011; Zack et al. 2013). Tumorigenesis is driven in part by the somatic copy number deletion and concomitant loss of function sustained at these loci, some of which may harbor one or more tumor suppressor genes (TSGs). This suggests that the physical distribution and synteny of TSGs throughout the genome may play an important role for the evolution of tumors, one that may be exploited by somatic genetic selection. This raises a number of questions: How are TSGs distributed across the human genome and is there any evidence of TSG colocalization? Are there evolutionary constraints on the physical organization of TSGs across other species? What are the implications for the pattern of somatic copy number deletions observed in human tumors?

To address these questions, there is a pressing need for computational tools that can overcome the onerous task of querying the growing list of available assembled genomes, analyzing the linear ordering of genes across the human genome to identify whether they form clusters, and assessing the conservation of these clusters across other species. To this end, we developed Kerfuffle, a web application that efficiently computes various summary statistics of gene clustering across most genomes in the Ensembl database (Kinsella et al. 2011), compares significance of clustering with shuffled null models, and graphically displays the results. The main advantage of Kerfuffle is that it only requires a user to specify human gene names and species of interest. In addition, orthologous gene searches are automated utilizing pre-computed homology from Ensembl servers, a relative statistic is used to quantify cluster conservation, and the online platform permits server-side saving of results for each registered user for later analysis. Furthermore, Kerfuffle can generate a visualization of the clusters using the Circos software (Krzywinski et al. 2009). This comprehensive platform is an important step in furthering our understanding of genome organization and its evolution.

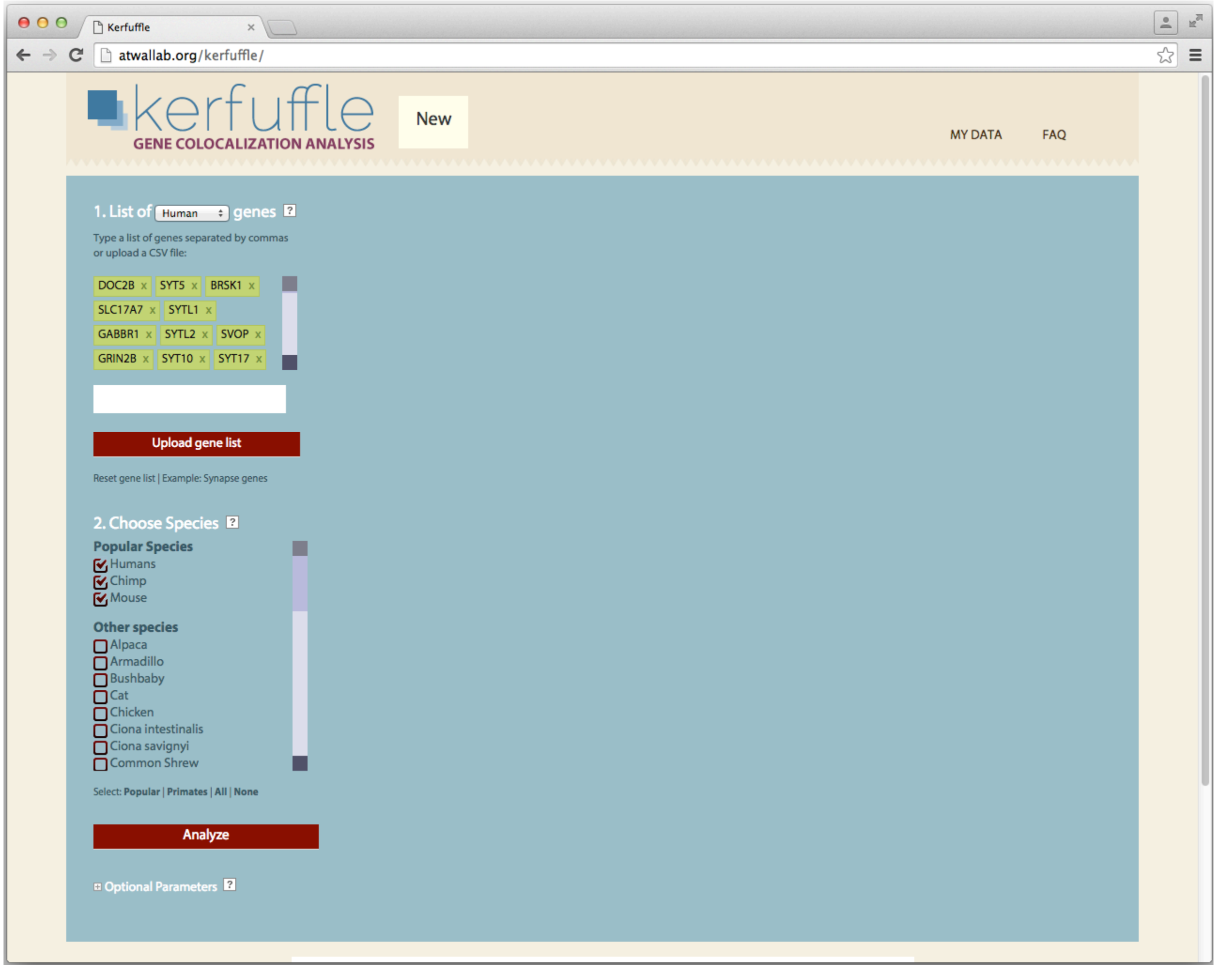
Next, we use Kerfuffle to carry out an integrative analysis of known TSGs and demonstrate significant co-localization of these genes into conserved syntenic clusters throughout the genome. Comparative genomics analysis provides evidence of evolutionary selection enforcing this co-localization across 46 eukaryotic species, ranging from worms to mammals.

5.2. Results

5.2.1. Kerfuffle: a web tool for multi-species co-localization analysis

Here we present Kerfuffle, a tool for automating the analysis of gene co-localization across multiple species. Although some tools exist to cluster genes, there are currently no tools available for efficiently verifying whether a given list of genes from one species forms clusters, and whether these clusters are conserved across other species. For example, although tools such as C-Hunter (Yi et al. 2007) cluster genes by genome position and GO category, they do not incorporate an analysis of conserved clustering across multiple species, and are not intended as a tool to query a general set of genes that don't share GO terms. Other tools, such as CGCV, allow for clustering across many species but require the user to input DNA sequences instead of gene names (Revanna et al. 2009); subsequently, the web tool performs BLAST searches to find orthologous genes, which adds significant overhead to run-time. There are related tools which identify regions of synteny, such as EnsemblCompara (Vilella et al. 2009), i-ADHoRe (Proost et al. 2012), MCScanX (Wang et al. 2012b), Cinteny (Sinha and Meller 2007), OrthoClusterDB (Ng et al. 2009) and Syntenator (Rödelsperger and Dieterich 2008). These tools are useful for identifying homologous genomic regions between species, but do not include an automated approach for evaluating gene clustering and its conservation across species.

As shown in Figure 5.1, Kerfuffle allows the user to specify a set of gene names, and select up to 47 species on which the analysis will be performed concurrently (we support the *Ensembl* and WikiGene naming standards). Default analysis parameters are provided, although customization is allowed; parameters include: **(1)** *d,* the maximum number of total intervening genes (or gaps) allowed in a cluster (Figure 5.8); **(2)** the maximum value on the *x*-axis of the histogram of distance between consecutive gene pairs (Figure 5.2A, C); **(3)** the maximum value on the *x*-axis of the histogram of cluster sizes (Figure 5.2B); and **(4)** the number of randomizations for p-value estimation.



1. List of Human genes ?

Type a list of genes separated by commas or upload a CSV file:

DOC2B x SYT5 x BRSK1 x
SLC17A7 x SYTL1 x
GABBR1 x SYTL2 x SVOP x
GRIN2B x SYT10 x SYT17 x

Upload gene list

Reset gene list | Example: Synapse genes

2. Choose Species ?

Popular Species
☑ Humans
☑ Chimp
☑ Mouse

Other species
☐ Alpaca
☐ Armadillo
☐ Bushbaby
☐ Cat
☐ Chicken
☐ Ciona intestinalis
☐ Ciona savignyi
☐ Common Shrew

Select: Popular | Primates | All | None

Analyze

⊞ Optional Parameters ?

Figure 5.1: Kerfuffle allows the user to input a list of genes and a set of species on which to launch the analysis.

Once the analysis launched, Kerfuffle queries the Ensembl BioMart database (Kinsella et al. 2011) and retrieves gene position information for all genes of interest. Simultaneously, Kerfuffle identifies the corresponding homologs and paralogs for each species of interest using EnsemblCompara (Vilella et al. 2009). Finally, the queried genes are grouped into clusters based on their co-locality (Methods). Once the analysis complete, the results are displayed to the user, including the distribution of distances between consecutive gene pairs (Figure 5.2A), a histogram of cluster sizes (Figure 5.2B) and the distribution of distances between consecutive gene pairs across various species (Figure 5.2C).

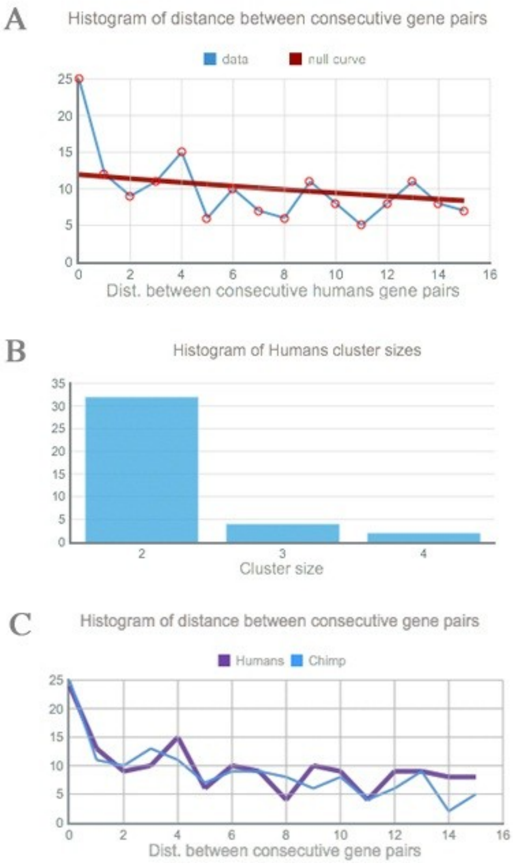


Figure 5.2: Sample Kerfuffle output using genes with ontology term 'synapse'. **(A)** The discovered (blue) and expected (red) human distance distribution. **(B)** Cluster size histogram for humans. **(C)** Distance distribution of two species, human (violet) and chimp (blue).

The plots in Figure 5.2A are interactive: hovering over each point in the plot of reveals its x and y coordinate, and clicking on the point will reveal all gene pairs separated by a distance x. To assess the significance of the clustering, we overlay a plot of the expected distribution under random gene shuffling, i.e. if gene co-localization were random. Deviation from the null distribution is also quantified as a p-value table generated using a permutation test. Note that the null distribution curve in the Figure 5.2A may appear to be linear, as opposed to the expected exponential, due to the significance of gene clustering—in other words, highly significant counts may overwhelm the null-curve. As a result, an option is available to generate an independent plot of this curve, demonstrating the decaying nature of the distribution. To ensure that p-value calculations does not slow down webpage usability, the calculations are performed in the background and appear in a table once complete.

In Figure 5.1 and Figure 5.2, the genes used are list of 477 functionally related synapse genes we obtained using the ontology term "synapse" in the AmiGO database (Carbon et al. 2009). Our analysis suggests that these genes are significantly more clustered than expected by chance (Figure 5.2, blue vs. red curve). Co-localization of these genes is supported by a study that demonstrates clustering of genes associated with GABAergic circuit assembly in the cerebellar cortex (Paul et al. 2012b).

For comparative genomics purposes, the user can also launch an analysis comparing human clusters to those of other species. To quantify the conservation of gene clusters in species T relative to those found in species S, we use the following conservation score:

$$Score(S,T) = \frac{1}{N_S} \sum_{i}^{N_S} \sum_{j}^{N_T} \frac{\left|S_i \cap T_j\right|}{\left|S_i\right|},$$

where S_i and T_i refers to the set of genes in cluster i in species S and T, respectively. N_X refers to the total number of clusters in species X. All clusters were chosen as size 2 or larger. The intersection between S_i and T_j is defined as the set of common genes between cluster i in species S and cluster j in species T. We normalize the size of the intersection by the size of the cluster S_i, hence calculating the score relative to species S. The inner sum increases if the genes found in cluster *j* of species T are also found in cluster *i* of species *S*, while the outer sum averages those scores over each cluster *i* in species S. Thus, *Score(S, T)* is a statistic which increases as the same clusters are observed and remain intact amongst the species investigated in T relative to S. Our default setting for this analysis sets *S* = *Human*. Once this analysis completed, Kerfuffle displays the degree of conservation of the clusters in each species relative to humans and plots the consecutive distance distribution for all species of interest (Figure 5.2C). For pathway analysis, we support querying the KEGG pathway database (Kanehisa et al. 2012) directly from Kerfuffle to assess whether genes in a cluster are enriched for belonging to a common pathway.

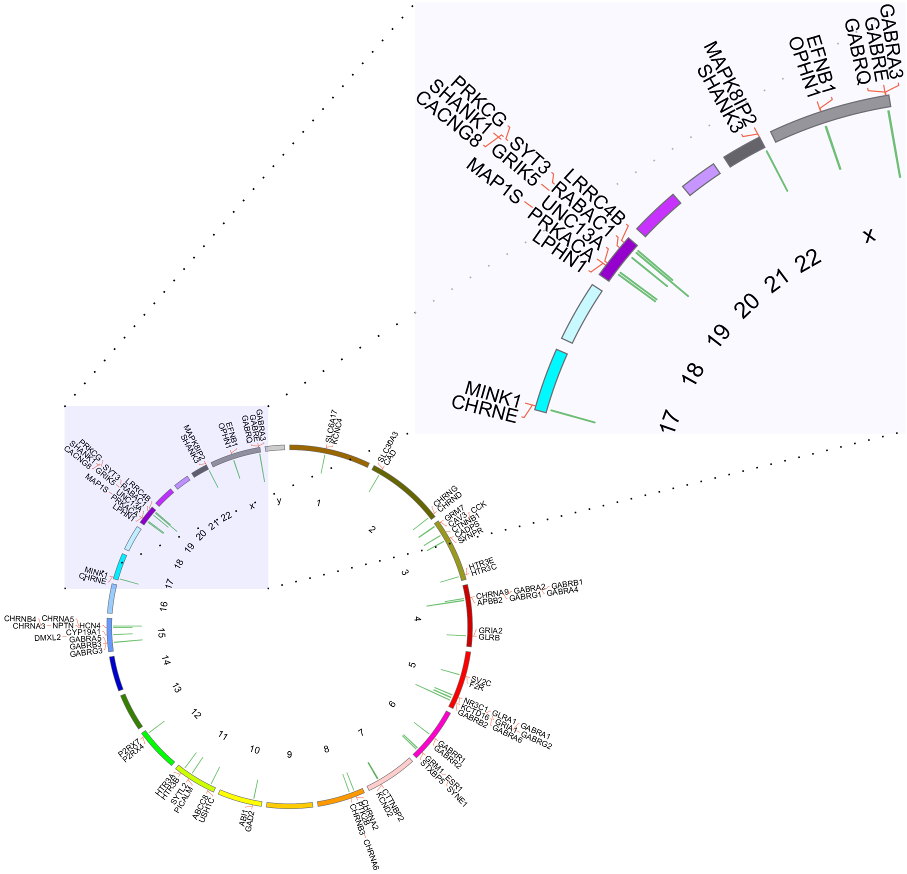


Figure 5.3: Circos output from the example clustered synapse genes. Kerfuffle outputs a Circos plot clustering the genes investigated by the user. The clusters are quantified by the green bars protruding inward in the Circos plot. The longer the bars, the more genes in the cluster. The output image also lists the colocalized genes.

To visualize gene clusters, Kerfuffle offers the option to generate a Circos plot. Figure 5.3 shows a Circos plot of the clustered genes from our synapse genes example. The sizes of the clusters are represented by a green histogram located at the appropriate genomic start and stop of the clustered genes, pointing radially inwards. We have attempted to optimize output for visualization of gene names (pointing radially outwards) while maintaining all genes on the image, however, some genes may run-off the Circos image because it is impossible to know *a priori* how many genes will sit next to each other in any given colocalization analysis.

To evaluate the performance of our tool, we ran several queries using gene sets of varying size and number of species (Figure 5.4). We find that a typical query of ~500 genes in 5 species completes in ~25 seconds (or ~3 minutes when querying all 47 species). Overall, for a given number of species, the running time increases exponentially with the number of input genes. However, even a query of 5,000 genes (an unusually high number of genes) in all 47 species completes in less than 10 minutes. Hence, our server is well suited to ensure that queries are handled expediently. Although there is no limit on how many genes a user can input, we recommend that users do not exceed 10,000 genes in order to maintain a reasonable running time, as well as the usefulness of results (too many genes increases the likelihood of finding clusters).

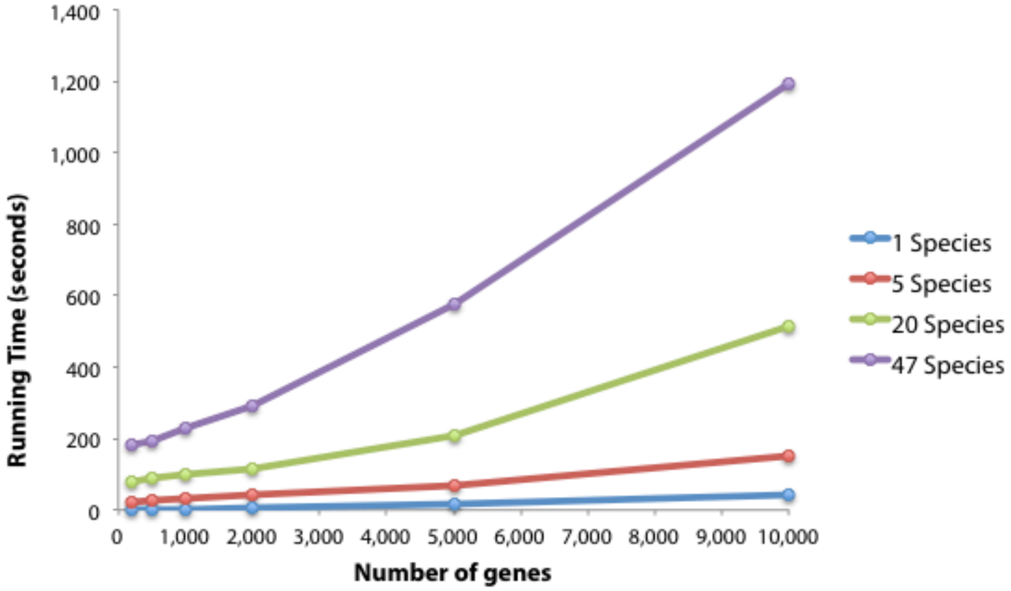


Figure 5.4: Running time of Kerfuffle as a function of the number of genes queried. For any given number of species, the running time increases exponentially with the number of input genes, but does not exceed 20 minutes for up to 10,000 genes in 47 species.

5.2.2. Tumor suppressor genes co-localize in syntenic clusters

To investigate the comparative genomic architecture of TSGs, we analyzed the assembled genomes of 46 different eukaryotes, spanning a wide range of taxonomic classes from Saccharomycetes to Mammalia (Supplementary Material). The TSGs used in this study were compiled by combining putative TSGs along with genes identified in the literature as promoting growth through whole genome RNAi-based screens of human cells and mouse tumor models (McClatchey et al. 1998; Salomoni and Pandolfi 2002; Yang and Fu 2003; Bench et al. 2004; Futreal et al. 2004; Sherr 2004; Ji et al. 2005; Westbrook et al. 2005; Bagchi et al. 2007; Lu et al. 2007; Hamaï et al. 2008; Rottmann et al. 2008; Xue et al. 2008; Zender et al. 2008b; Zhan et al. 2008; Bric et al. 2009; Gewinner et al. 2009; Veeriah et al. 2009; Zilfou and Lowe 2009; Chicas et al. 2010; Hsu et al. 2010; Park et al. 2010; Reimann et al. 2010;

Wajapeyee et al. 2010; Boehler et al. 2011; Bonilla et al. 2011; Isobe et al. 2011; Kong et al. 2011; Saha and Robertson 2011; Xu et al. 2012).

From this diverse set of literature we compiled a conservative list of 221 known and putative TSGs (Table 5.1) for the subsequent analyses in this study. In Figure 5.5, we represent the distribution of TSGs along the human genome. For other species, homologs were determined using the pipeline in (Flicek et al. 2012), and syntenic biases due to tandem duplications were addressed by removing all duplicated genes except one in each set of tandem duplicates (Ouedraogo et al. 2012).

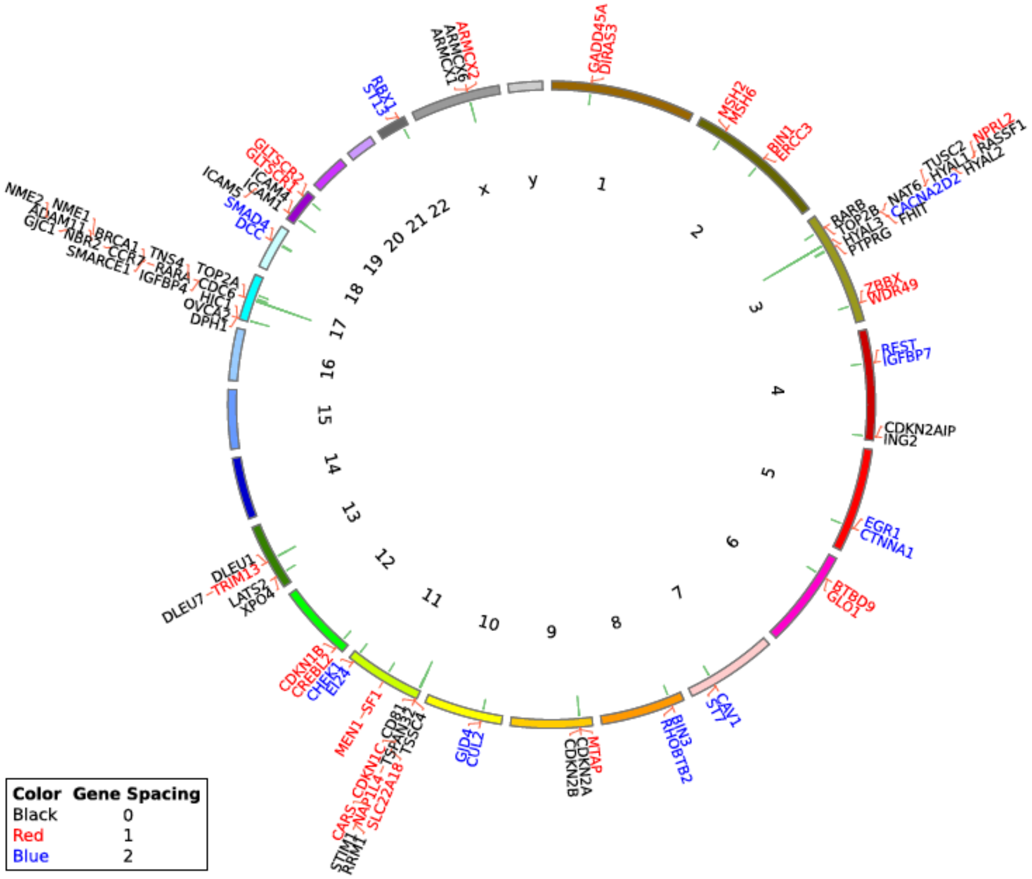


Figure 5.5: Distribution of TSGs in the Human Genome. The chromosomes are given by each color band and are labeled with the appropriate number or symbol. The genes are listed in genomic order. TSG Clusters of size $n \geq 2$ are marked with green inward-pointing bars, whose lengths denote the number of TSGs in a given cluster. The color of the gene name denotes the spacing of the cluster to which it belongs. Here we show duplicated genes although the duplicates were not involved in the co-localization analysis.

Using TSG localization data from all species, the distribution of TSG interval spacing is observed to approximately collapse on a characteristic curve when plotted on rescaled coordinates (Figure 5.6A), exhibiting an invariant global property of the TSG genome architecture across species.

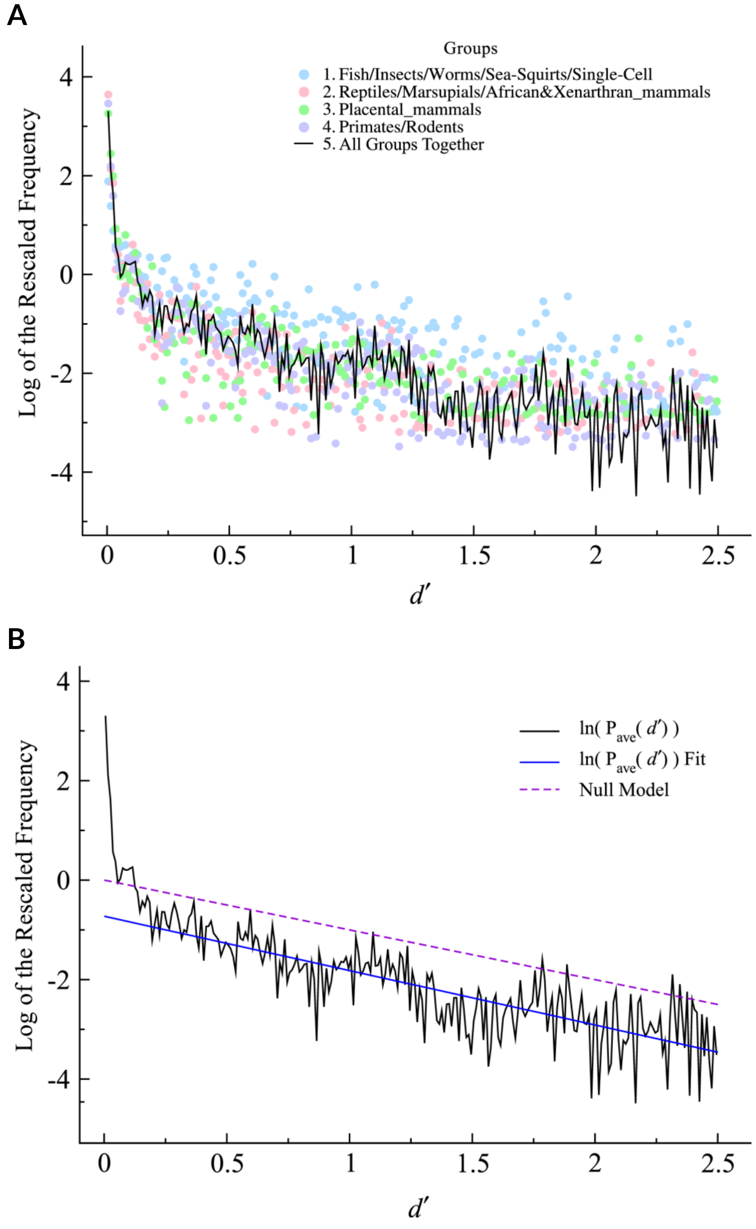


Figure 5.6: Frequency of TSG pairs as a function of normalized distance. **(A)** The black curve shows the average distribution over 46 different species. The four groups of related species are distributed about the average. **(B)** Shows the linear fit (solid blue) from $\boldsymbol{d'} \sim \mathbf{0.2-2.5}$ and the null model (dashed purple), both of which share similar slopes. Comparing the null model to the solid black curve, demonstrates that TSGs are significantly closer to each other than expected by chance ($\boldsymbol{p} < \mathbf{1\times10^{-9}}$ for human at $\boldsymbol{x} = \mathbf{0}$, for example); there are significantly more counts at small intergenic spacing than expected from the null model across most species.

We modeled the null distribution of TSG interval spacing as a Poisson process, subject only to the constraint that the average interval spacing for each species must be equal to the ratio of the number of genes in the genome to the number of TSG homologs (Methods). The resulting null distribution for a sufficiently large genome takes an exponential form, characteristic of a homogeneous Poisson process, $p(x) \approx A_i e^{-A_i x}$, where A_i is a constant dependent on species i, and x is the gene number spacing between two TSGs. When normalized for the varying number of TSG homologues in each species, the experimental data closely tracks the null model when averaged over all genomes for large intervals (Figure 5.6B).

For small intergenic intervals, we find significant departure from the null distribution for most species, for example, at $x = 0$ (i.e. situations where two TSGs are separated by no other genes), humans: $p < 10^{-9}$; mouse: $p < 10^{-9}$; zebrafish: $p \approx 6.0 \times 10^{-7}$. These results indicate significant co-localization of a subset of TSGs in which each consecutive TSG is typically separated by no more than 2 non-TSG genes (Table 5.1).

Together, these observations support a two-regime model for the correlations of TSG positioning: **(1)** below a critical interval length of about 2 genes, the TSGs are highly co-localized; and **(2)** above the critical interval length, the positioning of syntenic TSGs is uncorrelated and follows a random distribution. We note that re-parameterization of the intervals in terms of base pairs does not lend itself to the simple mathematical arguments used here due to gene length/interval variability, but qualitatively similar trends can be observed, i.e. the distributions of TSG interval base pair spacing for all species exhibits significant co-localization.

To quantify the extent to which inter-species chromosomal rearrangements may have disrupted synteny between clustered genes, we developed a cluster conservation score that quantifies the preservation of gene clusters between species. The conservation score between two species is a normalized count of homologous genes formed by the intersection between gene clusters found in the two genomes (Aboukhalil et al. 2013).

For every species, we calculated the scores separately for TSG homolog clusters and randomly chosen non-TSG gene clusters with equivalent cluster size distribution. Here we used the human genome as a reference in the calculation of the conservation score so that species with cluster profiles similar to humans would have scores closer to the maximal value 1 (Methods). As expected, species that share more recent common ancestry to humans—such as primates—tend to have larger conservation scores whereas more distant species such as fish and insects had lower scores. This observation was true for both TSG clusters and non-TSG clusters. Figure 5.7 summarizes the conservation scores for all analyzed taxonomic groups (see Figure 5.9 for a detailed breakdown of species-specific results).

Notably, we also observed a significant trend that the TSG scores tend to be higher than non-TSG scores across all species ($p = 3.5 \times 10^{-9}$), providing evidence that evolutionary selection has preferentially maintained the integrity of gene clusters associated with tumor suppression, compared to random clusters. The conserved physical linkage of TSGs argues for essential pleiotropic functionality of these genes beyond tumor suppression, since many tumor-associated deaths typically occur post reproduction age.

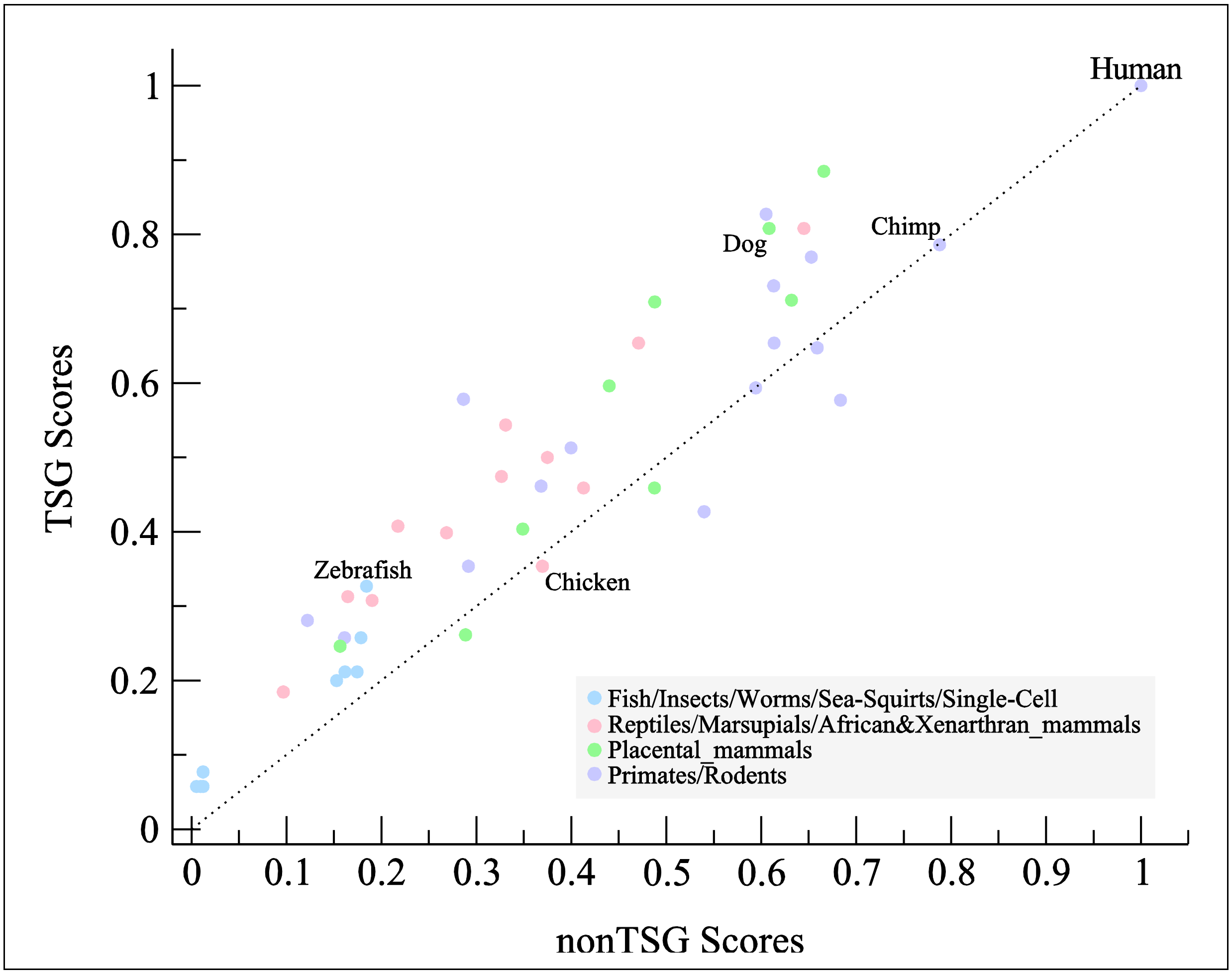


Figure 5.7: Conservation of TSG Clusters for Related Groups. Each species is depicted inside its respective taxonomic group. The human score is "1" because its clusters are completely conserved. Chimp's score is the closest to humans. The shift upward away from the $y = x$ line demonstrates that TSG clusters are, on average, conserved across species, more so than random gene clusters elsewhere in the genome ($p = 3.5 \times 10^{-9}$).

To determine whether other large functional groups of genes show similar conservation of clusters, we categorized gene groups of size at least 100 using the Gene Ontology (GO) database (Ashburner et al. 2000) and investigated their co-localization. We find 52 functional groups that cluster significantly in humans (Table 5.2). Using the same procedure, we find 20 functional groups that also show statistical evidence of conservation across all 47 eukaryotic genomes, e.g. genes involved in synaptic transmission ($p = 3.3 \times 10^{-7}$, Bonferroni-corrected). This suggests that functional clustering of related genes may be a shared feature across a wide eukaryotic clade, and not just mammals (Petkov et al. 2005a; Larkin et al. 2009; Dixon et al. 2012; Paul et al. 2012a).

5.3. Discussion

5.3.1. Kerfuffle

Kerfuffle is web analytics platform that provides tools for efficiently obtaining genomic organizational information about a set of user-defined functionally related genes. The software discovers clusters in each species of interest and determines the significance of those clusters while allowing for the interactive and visual exploration of genomic structure. Since it is expected that speciation would lead to differences in genomic organization, provided organization is random, we investigate relative cluster conservation between species using a measure we define as the *Score(S,T)*. Once the analysis is performed, the user may compare species and determine the degree of cluster conservation. The optional parameters make the investigations customizable and allow the user to optimize run-time. An account may also be created where all investigations may be saved for later use. Further, our website has an extensive FAQ section which may help guide the user.

Future developments will include increased investigative options, such as changing the type of genes investigated (currently set to protein-coding only) and incorporation of other gene name schemes (such as RefSeq IDs). Currently, our default conservation score sets humans as the relative species, i.e. for all calculations, S = Homo sapiens. Other features, such as the identification of common clusters in the species will be added, while other functionality will be included to improve our pathway investigations. Currently, we link to the KEGG website, a multi-gene pathway search. In later developments, our webpage will determine the similar pathways and display them along with the clusters. Finally, our current Circos implementation is limited to humans, mouse, rat, and drosophila; in future developments, we will make the visualization of clusters available for a much wider range of species.

5.3.2. Co-localization of tumor suppressor genes in synteny clusters

Here we demonstrate that TSGs are co-localized in the human genome, and are preserved in syntenic clusters across hundreds of millions of years of evolution. This co-localization was present significantly more than expected by chance, suggesting that a selective pressure is at play to maintain TSG gene order. To address potential biases in our downstream analysis, we removed tandem duplicate genes from our gene set (Methods). Tandem duplicates are genes that are co-localized following a gene duplication event (e.g. HOX genes), and could bias our analysis by exhibiting shared behavior such as co-expression due to sequence similarity. Furthermore, to account for the nature by which TSGs are often discovered (searching for novel TSGs in regions surrounding previously-known TSGs), our list was populated from studies that identified growth-promoting genes via genome-wide RNAi screens on human cells and mouse tumor models.

5.3.2.1. Mechanisms

A plausible mechanism for the enforced conservation is that the expression of TSGs is controlled at the chromatin level, i.e. clusters of neighboring TSGs are required to be co-expressed. It is known that species such as *S. cerevisiae*, *C. elegans*, and *D. melanogaster* demonstrate significant co-expression of co-localized, functionally related (Cho et al. 1998), and tissue-specific genes (Blumenthal et al. 2002; Boutanaev et al. 2002; Roy et al. 2002; Lercher et al. 2003). There is also evidence of clustered gene co-expression in mammals (Caron et al. 2001; Dempsey et al. 2001; Yang et al. 2002; Akashi et al. 2003).

Co-expression analyses by the Atwal Lab of published gene expression data suggest that co-expression between neighboring genes—irrespective of their function—is correlated across the genome; however, this trend was not confirmed for TSGs due to low statistical power (Fendler and Atwal, unpublished). This suggests that if co-expression is an important factor in maintaining conserved synteny of TSGs then it might only occur strongly during specific developmental stages. In fact, preliminary re-analysis of RNA-seq data obtained from the modENCODE project suggests that TSGs in *D. Melanogaster* may be more co-expressed than random sets of genes with the same cluster structure (Aboukhalil and Atwal, unpublished). However, this co-expression data was obtained from whole flies, yet TSG co-expression could be restricted to a select subset of tissues, and it is likely that fly cells exhibit significant heterogeneity during development. To address these issues, it would be necessary to study this process at single-cell resolution using a single-cell RNA-seq time-course. By studying gene expression levels along various developmental stages of model organisms such as flies and mice, we could determine if co-expression is present despite concerns over tissue-specificity and heterogeneity.

In addition to regulated co-expression, a few other mechanisms have been proposed to explain gene co-localization. For example, it has been suggested that co-localization could be driven by epistatic interactions between neighboring genes (Fisher 1930; Nei 1967). For example, compensatory mutations—where a mutation in gene A is counteracted by a mutation in a nearby gene B—would lead to reduced recombination events and hence synteny. In addition, it is possible that important groups of genes are co-localized in regions of the genome where very few structural variation mutations occur. This would suggest that studying known mutational cold spots could reveal further gene clusters.

5.3.2.2. Co-localization goes beyond TSGs

As shown above (§5.2.2), TSGs are not the only class of genes that are co-localized—colocalization analysis of other Gene Ontology (GO) groups reveals that at least a dozen more GO categories exhibit significant clustering across 46 species (Table 5.2). In future studies, our analysis could be extended; starting with sets of genes known to share function or phenotype, Kerfuffle could be used to automate the identification of additional co-localized groups of genes. Alternatively, from a bottom-up perspective, Kerfuffle could also be used to identify synteny blocks that are preserved across hundreds of millions of years, and once found, experimentally study the shared properties (e.g. co-expression, function) of these genes.

5.3.2.3. Implications of TSG co-localization

The conserved co-localization of TSGs is expected to have clinical consequences for the development of tumors, and may fuel a different mechanism of deletion-mediated tumorigenesis. The canonical model of tumor progression posits that it proceeds, in part, by sequential inactivation of single TSGs in a classic two-hit manner. However, recent studies suggest that happloinsufficient genes may play a more general role in tumorigenesis (Solimini et al. 2012; Xue et al. 2012a), and thus clusters of TSGs could potentially be prime genomic targets for lesions due to the selective pressure to suppress the activity of multiple genes. This new hypothesis predicts that larger TSG clusters, containing multiple weak tumor suppressive genes, are on average associated with greater fitness advantages for tumor growth, and should therefore be more frequently deleted in tumors.

Analysis by the Atwal Lab of 4,466 published and unpublished cancer sample patients across various tissue types suggests that the mean deletion probability of a particular TSG increases significantly with the number of TSGs in that cluster (Fendler and Atwal, unpublished). These results demonstrate a genome-wide pattern of mutations in which tumors are vulnerable to preferential deletion of regions enriched in TSGs and, conversely, focal deletions for isolated TSGs are not as frequent. This would suggest another mechanism for tumorigenesis other than the Knudson two-hit mechanism, i.e. the attenuation of multiple haploinsufficient genes can also drive the growth of tumors.

5.4. Methods

5.4.1. Kerfuffle

The Kerfuffle back-end runs on PHP 5.2 on an Apache server. The front-end was built primarily in HTML and JavaScript and two JavaScript libraries to enhance user experience: jsCharts for plotting graphs and jQuery to asynchronously query the server. Kerfuffle is also flexible in the way it accepts user input. The user may choose to input genes in a textbox one by one or alternatively, may upload a file that contains a list of genes, each of which is separated by a break line. However, we recognize that it is difficult for users to keep track of the dozen file formats they use. Thus, if the uploaded file is a comma- or tab-delimited file with multiple columns, Kerfuffle will ask the user to specify the column in which the gene names are found. To aid in recurring analyses, we recommend that users create a free

Kerfuffle account, in which their results and the queried genes will be saved in our databases. On the back-end, the query results obtained from Ensembl are temporarily stored in text files and purged every week, unless users decide to save their results to their account, in which case the results remain on the server until the users delete them.

5.4.2. Genomic Spacing Analysis

To download all available genomes, Kerfuffle interfaces with the Ensembl BioMart server. While there are 70 species available, some assemblies are not of high enough quality for the analysis presented here. Thus, only genomes that were at least in the "Scaffold" build stage were considered. If the genome's chromosomes contained too many "contig", unknown ("Un"), or "_random" elements, the species was not analyzed. A threshold of 75% of the chromosomes in the genome must be at least numeric or of "Scaffold" build. Other non-numeric chromosomes counted against the genome, such as, "reftig"s, "HG"s, "HSCHR"s, "Ultra"s, "GL"s, etc. The final list of species used can be found in the Supplementary Material. For every gene in the human genome, orthologs were obtained from Ensembl. If more than one ortholog was discovered, we used the larger of the "Query ID," which indicates the percentage of the queried sequence (human) that matches the ortholog belonging to the other species. We used this data to create a database of orthologs, and modified each genome to guarantee a conservative analysis; for each duplicated gene, we merged the duplicates with one gene in the duplicated group.

Clusters used in the conservation analysis are defined as in Figure 5.8. Namely, a set of ordered genes $G_1, G_2, \ldots, G_n$ is said to co-localize or form a cluster if the number of total intervening genes is less than or equal to the specified parameter d. Mathematically, if $x(G_i)$ is the positional order of gene G_i, then we require that $x(G_n) - x(G_1) - n + 1 \le d$. In Kerfuffle, the default value of the parameter d is 2. To optimize performance, this algorithm was written in C++.

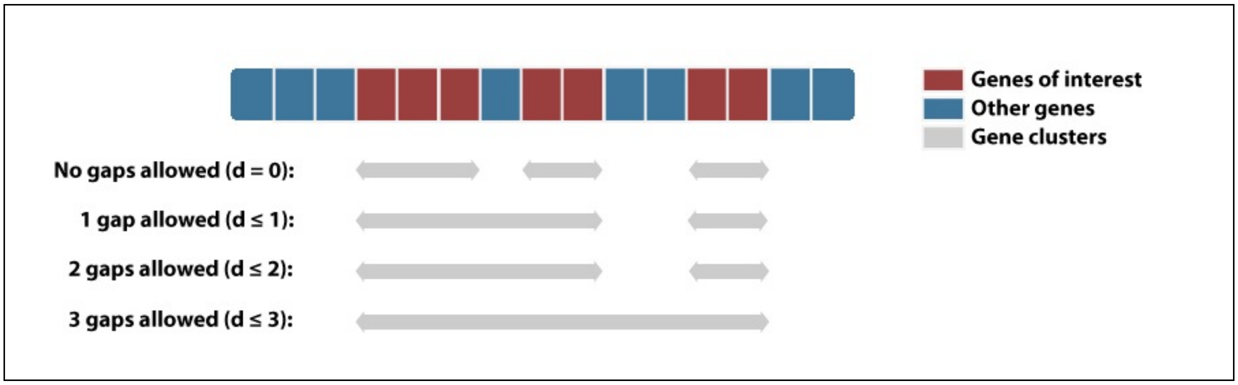


Figure 5.8: Examples of cluster definition. The clusters are defined by the parameter d, the maximum number of allowed gaps in a sequence of genes. Red boxes represent queried genes and blue boxes represents genes not queried.

5.4.3. Distribution of the random inter-TSG spacing

The probability distribution of the number of genes between consecutive TSGs (d) is governed by a Poisson process. For a given species i, the probability of observing two TSGs separated by d non-TSGs is:

$$P_i(d) = P(\text{TSG}) \cdot P(\text{no TSG})^d$$

$$P_i(d) = \frac{T_i}{G_i}\left(1 - \frac{T_i}{G_i}\right)^d$$

where T_i is the number of TSGs and G_i is the number of genes in the genome of species i. For convenience, we define the species-specific constant $A_i = T_i/G_i$:

$$P_i(d) = A_i(1 - A_i)^d$$

$$= A_i\left((1 - A_i)^{\frac{1}{A_i}}\right)^{dA_i}$$

$$\approx A_i \mathrm{e}^{-dA_i}$$

where we assume $A_i \ll 1$, allowing us to use the fact that $\lim_{n\to\infty}\left(1-\frac{1}{n}\right)^n = e^{-1}$. Note that we ignore boundary effects arising from the finite extent of chromosomes.

5.4.4. TSG Co-localization analysis

The genomic co-localization of each orthologous set of TSGs was investigated using Kerfuffle (Aboukhalil et al. 2013). While the significant regions ($d < 3$ genes) of the count distribution of each species were well sampled, larger distances did not offer sufficient statistics to demonstrate null model predictions. To solve these issues, we instead viewed each species as a sample of the same null model. Due to the model's simplicity, the empirical data can be normalized so that each "sample" follows the same null distribution. To perform the analysis across all species, with differing genome size and number of TSGs, we use the following normalization scheme:

$$P_i(d) = A_i e^{-dA_i}$$

$$\frac{P_i(d)}{A_i} = e^{-dA_i}$$

Setting $P_i'(d) = P_i(d)/A_i$ and $d_i' = dA_i$:

$$P_i{}'(d) = e^{-d_i{}'}$$

We then bin the $P_i'(d)$ distributions at distances $d_i{}'$. While the bins do not represent any real distance, each species is seen as a set of samples of the distribution $e^{-d'}$. To estimate the behavior across all m species, we average the m empirical distributions:

$$\langle P_i'(d')\rangle = \frac{1}{m}\sum_{i=1}^{m} P_i'(d') \approx e^{-d'}$$

If the count distribution follows our model prediction, then the log of this average will lead to a slope of -1. This is Figure 5.6 in the main text and has a slope of -1.09.

Reconstructing the probability distribution assumes that each realization is independent. This is not always the case since each common ancestor shares random genomic events prior to species divergence. However, we make the assumption that each species has diverged sufficiently to be treated as an independent sample. There are some species that are clearly not independent, for example, human, chimpanzee, and gorilla, which will lead to some over-fitting.

5.4.5. Conservation analysis

As described in §5.2.1, we quantify the conservation of gene clusters in species S relative to those found in the human genome H using the following conservation score:

$$\text{Score}(S,H) = \frac{1}{N_H}\sum_{i=1}^{N_H}\sum_{j=1}^{N_S}\frac{\left|H_i \cap S_j\right|}{|H_i|}$$

where S_i and H_i refer to cluster i of species S and in humans, respectively, and N_X refers to the total number of clusters in species X. To be included in the conservation score, clusters must be composed of two or more genes. The expression $\left|H_i \cap S_j\right|$ is the number of common (orthologous) genes between cluster i in humans and cluster j in species S. The expression is then normalized by the size of cluster i so that the score is relative to the human genome.

For each species S, we calculated $Score(H, S)$ and obtained two scores: **(1)** a score for TSG clusters; and **(2)** a score for randomly chosen non-TSG clusters that have the same cluster-size distribution as the TSG clusters. To determine the null distribution average score, we run 1,000 realizations for each species. Figure 5.9 shows each species' TSG score vs. the averaged non-TSG score for that species. It is expected that the null distribution would fall about the $y = x$ axis. This figure demonstrates that the TSG scores are generally greater than the non-TSG scores, suggesting that TSG clusters are more frequently conserved across species. Note that Figure 5.9 is the same as Figure 5.7 except all species are specified individually rather than in groups.

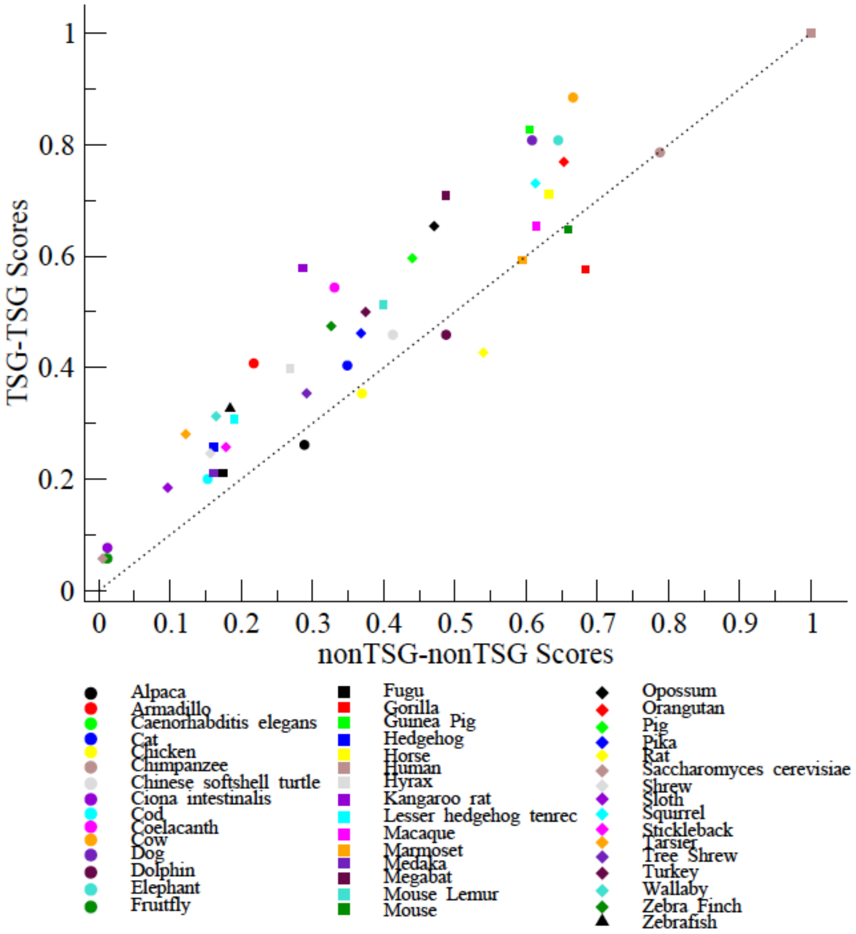


Figure 5.9: Conservation of TSG Clustering for each species. Each species is identified by a unique mark (combination of color and point-type).

Since the selection forces on each cluster are unknown, we cannot analytically model the null distribution for a significance test. We thus define the statistic R to measure how far the points deviate from $y = x$:

$$R = \frac{1}{N}\sum_{S} \frac{\text{Score\_TSG}(H,S)}{\text{Score\_nonTSG}(H,S)}$$

where N is the number of species. As R increases past "1", the more likely the selected set of genes are conserved across the species. An R less than "1" would suggest selective pressure against clustering. Once many R's are collected, we construct a null distribution. The R statistic measured for TSGs is significantly greater than the mean of the null distribution. Assuming a one-tailed Gaussian distribution, we find $p = 3.5 \times 10^{-9}$.

Next, we investigated whether the conservation phenomenon was limited to only TSGs. We collected GO (Gene Ontology) IDs that contained at least 100 genes. Those groups which showed significant clustering (52) were analyzed for conservation, as similarly done for TSGs. We find, after a Bonferroni correction that 20 GO groups showed significant co-localization and conservation of the clustered genes in that GO group. The GO groups and the significance results may be found in Table 5.2 (Supplementary Material).

5.5. Supplementary Material

5.5.1. Species used in the analysis

Overall, 46 species were used in the TSG analysis: *Vicugna pacos, Dasypus novemcinctus, Ciona intestinalis, Caenorhabditis elegans, Felis catus, Gallus gallus, Pan troglodytes, Pelodiscus sinensis, Gadus morhua, Latimeria chalumnae, Bos taurus, Canis familiaris, Tursiops truncatus, Loxodonta africana, Drosophila melanogaster, Takifugu rubripes, Gorilla gorilla, Cavia porcellus, Erinaceus europaeus, Equus caballus, Homo sapiens, Procavia capensis, Dipodomys ordii, Echinops* *telfairi, Macaca mulatta, Callithrix jacchus, Oryzias latipes, Pteropus vampyrus, Mus musculus, Microcebus murinus, Monodelphis domestica, Pongo abelii, Sus scrofa, Ochotona princeps, Rattus norvegicus, Saccharomyces cerevisiae, Sorex araneus, Choloepus hoffmanni, Ictidomys tridecemlineatus, Gasterosteus aculeatus, Tarsius syrichta, Tupaia belangeri, Meleagris gallopavo, Macropus eugenii, Taeniopygia guttata,* and *Danio rerio.*

5.5.2. TSGs used in the analysis

The tumor suppressor genes used in the co-localization analysis are listed in Table 5.1.

Chr	Gene names
1	TP73, CHD5, PRDM2, FABP3, CDKN2C, GADD45A; DIRAS3, CLCA2, PALMD, DDX20, ST7L, S100A2, BTG2, RASSF5
2	TSSC1, MSH2; MSH6, MXD1, CIAO1, STEAP3, BIN1; ERCC3, LRP1B, PMS1, IDH1
3	VHL, RARB; TOP2B, TGFBR2, MLH1, DLEC1, CSRNP1, RBM6, NAT6; HYAL1*; HYAL2*; HYAL3*; TUSC2; RASSF1; NPRL2; CACNA2D2, MAPKAPK3, PARP3, BAP1, FHIT; PTPRG, FOXP1, ATR, ZBBX; WDR49, TP63, DLG1
4	GAK, REST; IGFBP7, PTPN13, INPP4B, FBXW7, FSTL5, CDKN2AIP; ING2, FAT1
5	SKP2, DAB2, PIK3R1, RAD17, APC, MCC, CDC23, EGR1; CTNNA1, HDAC3
6	CDKN1A, BTBD9; GLO1, PERP, PLAGL1, LATS1
7	PMS2, PIK3CG, SLC26A3, CAV1; ST7
8	PINX1, DLC1, VPS37A, FGL1, BIN3; RHOBTB2, SFRP1, SCRIB
9	PTPRD, MTAP; CDKN2A*; CDKN2B*, RECK, GAS1, PTCH1, TGFBR1, SET, TSC1
10	WDR37, CUL2; GJD4, PTEN, FAS, LZTS2, TRIM8, MXI1, WDR11, DMBT1
11	TSPAN32; CD81; TSSC4; CDKN1C; SLC22A18; NAP1L4; CARS, STIM1; RRM1, ST5, MRVI1, WT1, SDHAF2, BAD, SF1; MEN1, ATM, CADM1, VWA5A, EI24; CHEK1, ST14
12	FGF6, ING4, CREBL2; CDKN1B, ANP32D, PAWR, BTG1, CDK2AP1
13	XPO4; LATS2, ATP8A2, BRCA2, STARD13, RB1, TRIM13, DLEU1; DLEU7, INTS6, ING1
14	SALL2, RPS29, NUMB, TRAF3
15	TP53BP1, SMAD3, PML
16	AXIN1, TSC2, TCEB2, DNAJA3, SOCS1, PALB2, RBL2, WWOX, WFDC1, GAS8
17	ABR, DPH1; OVCA2; HIC1, BCL6B, DLG4, EIF5A, TP53, MAP2K4, NF1, CDC6; RARA; TOP2A; IGFBP4; TNS4; CCR7; SMARCE1, BECN1, BRCA1; NBR2, ADAM11; GJC1, NME1*;

	NME2*, AXIN2
18	SMAD4; DCC
19	STK11, APC2, PIN1, ICAM1*; ICAM4*; ICAM5*, CDKN2D, SMARCA4, PIK3R2, CEBPA, ARHGAP35, BBC3, GLTSCR1; GLTSCR2, BAX
20	NRSN2, RBL1, L3MBTL1
22	BID, HIC2, SMARCB1, NF2, TIMP3, ST13; RBX1, ARHGAP8
23	RBBP7, SUV39H1, ARMCX1*; ARMCX6*; ARMCX2*, LDOC1, RPL10

Table 5.1: List of TSGs used in the co-localization analysis. Asterisks denote duplicated genes, and semicolons indicate clusters of TSGs.

5.5.3. Gene Ontology analysis

Gene Ontology groups used in the analysis and significance level is presented in Table 5.2.

Identifier	p-value	Description	Size
0010467	$< 1\times10^{-14}$	Gene expression	780
0005515	$< 1\times10^{-14}$	Protein binding	10084
0005524	$< 1\times10^{-14}$	ATP binding	1651
0005730	$< 1\times10^{-14}$	Nucleolus	704
0005737	$< 1\times10^{-14}$	Cytoplasm	5185
0005739	$< 1\times10^{-14}$	Mitochondrion	1543
0005829	$< 1\times10^{-14}$	Cytosol	2761
0005634	2.35×10^{-14}	Nucleus	5785
0044281	2.09×10^{-12}	Small molecule metabolic process	1526
0055085	7.56×10^{-12}	Transmembrane transport	778
0005654	4.13×10^{-11}	Nucleoplasm	1118
0006810	9.81×10^{-8}	Transport	554
0000139	1.22×10^{-7}	Golgi membrane	588
TSGs	1.82×10^{-7}	Tumor suppression	221
0046872	3.00×10^{-7}	Metal ion binding	2300
0007268	3.31×10^{-7}	Synaptic transmission	422

0006508	4.18×10^{-6}	Proteolysis	542
0003723	6.69×10^{-5}	RNA binding	720
0000122	0.001256324	Neg. regulation of transcription (Pol II promoter)	623
0006915	0.003827994	Apoptotic process	726
0000166	0.004391782	Nucleotide binding	648
0043565	0.008389023	Sequence-specific DNA binding	710
0005789	0.013259139	Endoplasmic reticulum membrane	763
0007165	0.039904931	Signal transduction	1429
0005794	0.041859028	Golgi apparatus	813
0005509	0.146089353	Calcium ion binding	780
0003700	0.34847182	Sequence-specific DNA binding TF activity	1071
0007596	0.472623809	Blood coagulation	507
0019048	0.968142871	Virus-host interaction	392
0003674	1	Molecular function	761
0003676	1	nucleic acid binding	1273
0003677	1	DNA binding	2279
0004930	1	GPCR activity	938
0005575	1	Cellular component	624
0005576	1	Extracellular region	1795
0005615	1	Extracellular space	946
0005622	1	Intracellular	1543
0005886	1	Plasma membrane	4052
0005887	1	Integral to plasma membrane	1210
0006351	1	Transcription, DNA-dependent	2044
0006355	1	Regulation of transcription, DNA-dependent	2084
0006954	1	Inflammatory response	360
0006955	1	Immune response	551
0007155	1	Cell adhesion	511

0007186	1	GPCR signaling pathway	1081
0007275	1	Multicellular organismal development	568
0008150	1	Biological process	708
0008270	1	Zinc ion binding	1358
0016020	1	Membrane	2467
0016021	1	Integral to membrane	5024
0042803	1	Protein homodimerization activity	649
0045893	1	Pos. regulation of transcription, DNA-dependent	586
0045944	1	Pos. regulation of transcription (RNA Pol II promoter)	811

Table 5.2: Gene Ontology groups and the (Bonferroni corrected) p-value of clustering in humans.

5.6. Contributions

- Analyses of co-localization, comparative genomics and gene ontology were performed by Bernard Fendler and myself.

CHAPTER 6

Conclusions and Perspectives

This concludes our brief foray into cancer evolution using single-cell sequencing and comparative genomics. In this section, we highlight our major contributions to the field and our outlook on future studies. As a tool for dissecting cellular heterogeneity, single-cell DNA sequencing (scDNA-seq) is increasingly becoming an indispensable tool. However, identifying copy-number alterations from scDNA-seq requires a complex pipeline to accurately infer integer copy-number states while accounting for sequencing and WGA biases. To render single-cell sequencing more accessible, we developed Ginkgo, a web analytics platform to aid in the analysis and visualization of single-cell CNV data (Chapter 2). Using Ginkgo, we evaluated the data quality of commonly used WGA methods and highlight their relative benefits.

In Chapter 3, we used simulations and statistical analyses to inform experimental design of single-cell sequencing studies. We identified the minimum read depth requirements for accurate copy-number analysis, tumor phylogeny, and assessment of clonality. We further explored the tradeoffs between sequencing a greater number of cells and sequencing at greater depth of coverage. In Chapter 4, we introduce a new approach for pruning non-informative CNV breakpoints from scDNA-seq data, and show its use for improving the inference of tumor evolution.

Finally, Chapter 5 explores the relationship between gene function and localization. To automate the analysis of gene co-localization at evolutionary time-scales, we built

Kerfuffle, a web platform that scans the human genome for evidence of gene co-localization, and carries out a comparative genomics analysis to detect whether this co-localization is preserved in other species. Using Kerfuffle, we show that Tumor Supressor Genes (TSGs)—genes whose inactivation contributes to cancer growth—are significantly co-localized in the human genome, and are conserved into syntenic groups over evolutionary time.

In the near future, improved and more cost-effective technologies for probing whole genome SNP and CNV mutations patterns will enhance our understanding of the mutational landscape of single cells. For example, such technologies can help answer questions about the relative timing of SNPs and CNVs in cancer. Preliminary single-cell investigations of breast and colon tumors have observed that most cells present either only CNV events, or both SNPs and CNVs, with no cells exhibiting SNPs alone (Wang et al. 2014; Huang et al. 2015), suggesting that large copy-number events may play critical driver roles in certain cancers.

Furthermore, advances in sequencing technologies will broaden the scope of single-cell studies through improved instrumentation, protocols, and read lengths. Although current sequencing technologies (e.g. Illumina, Ion Torrent, 454) feature read lengths on the order of hundreds of basepairs, Oxford Nanopore’s MinION and Pacific Biosciences’ RS II, produce reads spanning ~5kb and ~15kb respectively. Adapting instruments such as the RS II to single cell levels of input material could yield more accurate single-cell studies, since such technologies do not require amplification (hence no WGA bias), present virtually no bias in GC-rich or GC-poor regions (hence uniform coverage), and could simultaneously provide single-cell methylation patterns. Long-read technologies can also help span across repetitive regions in the genome, provide enhanced insight into large structural variations, and improve studies of genome phasing.

Another important area of single-cell biology is the sequencing of RNA from individual cells, a technique that has been applied to dissecting the heterogeneity of tumors (Patel et al. 2014), identifying novel cell types in complex tissues (Zeisel et al. 2015), and studying developmental patterns of gene expression (Xue et al. 2013). Recent advances have also allowed the sequencing of DNA and RNA from the same cell (Macaulay et al. 2015). As these technologies gain ground, improved population structure inference should be possible through the development of phylogeny approaches that combine SNP variation obtained from RNA along with CNV patterns obtained from DNA.

To elucidate the co-localization of TSGs, the use of technologies such as Hi-C (Van Berkum et al. 2010) can help probe the 3-dimensional architecture of the genome and identify potential long-range interactions between TSGs. Furthermore, although we did not observe evidence of co-expression of neighboring TSGs, further studies of gene expression are needed. In particular, since TSGs such as TP53 have been shown to play a critical role in fertility (Hu et al. 2008), time-course RNA-seq analyses could help shed light on whether TSGs are co-expressed at various stages of embryo development.

A recurring theme in this dissertation is the development of web applications to condense our complex multi-stage bioinformatics pipelines into accessible software tools for the broader community. Web-based platforms are valuable to experimental biologists and clinicians because they resolve issues of conflicting software dependencies, complex command-line interfaces, and provide a natural means for collaboration and sharing of results. With the increasing ubiquity of cloud services, web-based bioinformatics tools are likely to become more widespread in coming years.

References

Aboukhalil R, Fendler B, Atwal G. 2013. Kerfuffle: a web tool for multi-species gene colocalization analysis. *BMC Bioinformatics* **14**: 22.

Abyzov A, Urban AE, Snyder M, Gerstein M. 2011. CNVnator: an approach to discover, genotype, and characterize typical and atypical CNVs from family and population genome sequencing. *Genome Res* **21**: 974-984.

Akashi K, He X, Chen J, Iwasaki H, Niu C, Steenhard B, Zhang J, Haug J, Li L. 2003. Transcriptional accessibility for genes of multiple tissues and hematopoietic lineages is hierarchically controlled during early hematopoiesis. *Blood* **101**: 383-389.

Al-Shahrour F, Minguez P, Marqués-Bonet T, Gazave E, Navarro A, Dopazo J. 2010. Selection upon Genome Architecture: Conservation of Functional Neighborhoods with Changing Genes. *PLoS Comput Biol* **6**: e1000953.

Albright E, Hessel J, Hiranuma N, Wang C, Goings S. 2014. A comparative analysis of popular phylogenetic reconstruction algorithms. in *Proceedings of the Midwest Instruction and Computing Symposium (MICS)*.

Alexander J. 2016. *In preparation*.

Alexander J, Kendall J, Krasnitz A, Wigler M. In preparation. Single cell DNA analysis from prostate cancer biopsy. *In preparation*.

Alkan C, Coe BP, Eichler EE. 2011. Genome structural variation discovery and genotyping. *Nat Rev Genet* **12**: 363-376.

Andrews S. 2010. FastQC: a quality control tool for high throughput sequence data.

Ashburner M, Ball CA, Blake JA, Botstein D, Butler H, Cherry JM, Davis AP, Dolinski K, Dwight SS, Eppig JT et al. 2000. Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nature genetics* **25**: 25-29.

Bagchi A, Papazoglu C, Wu Y, Capurso D, Brodt M, Francis D, Bredel M, Vogel H, Mills AA. 2007. CHD5 is a tumor suppressor at human 1p36. *Cell* **128**: 459-475.

Baslan T, Kendall J, Rodgers L, Cox H, Riggs M, Stepansky A, Troge J, Ravi K, Esposito D, Lakshmi B et al. 2012. Genome-wide copy number analysis of single cells. *Nat Protoc* **7**: 1024-1041.

Baslan T, Kendall J, Ward B, Cox H, Leotta A, Rodgers L, Riggs M, D'Italia S, Sun G, Yong M et al. 2015. Optimizing sparse sequencing of single cells for highly multiplex copy number profiling. *Genome Res* **25**: 714-724.

Bench AJ, Li J, Huntly BJ, Delabesse E, Fourouclas N, Hunt AR, Deloukas P, Green AR. 2004. Characterization of the imprinted polycomb gene L3MBTL, a candidate 20q tumour suppressor gene, in patients with myeloid malignancies. *British journal of haematology* **127**: 509-518.

Beroukhim R, Mermel CH, Porter D, Wei G, Raychaudhuri S, Donovan J, Barretina J, Boehm JS, Dobson J, Urashima M. 2010a. The landscape of somatic copy-number alteration across human cancers. *Nature* **463**: 899-905.

Beroukhim R, Mermel CH, Porter D, Wei G, Raychaudhuri S, Donovan J, Barretina J, Boehm JS, Dobson J, Urashima M et al. 2010b. The landscape of somatic copy-number alteration across human cancers. *Nature* **463**: 899-905.

-. 2011. The landscape of somatic copy-number alteration across human cancers. *Nature* **463**: 899-905.

Blainey PC. 2013. The future is now: single-cell genomics of bacteria and archaea. *FEMS Microbiol Rev* **37**: 407-427.

Blumenthal T. 2004. Operons in eukaryotes. *Briefings in functional genomics & proteomics* **3**: 199-211.

Blumenthal T, Evans D, Link CD, Guffanti A, Lawson D, Thierry-Mieg J, Thierry-Mieg D, Chiu WL, Duke K, Kiraly M et al. 2002. A global analysis of Caenorhabditis elegans operons. *Nature* **417**: 851-854.

Boehler C, Gauthier LR, Mortusewicz O, Biard DS, Saliou JM, Bresson A, Sanglier-Cianferani S, Smith S, Schreiber V, Boussin F et al. 2011. Poly(ADP-ribose) polymerase 3 (PARP3), a newcomer in cellular response to DNA damage and mitotic progression. *Proc Natl Acad Sci U S A* **108**: 2783-2788.

Bonilla C, Hooker S, Mason T, Bock CH, Kittles RA. 2011. Prostate cancer susceptibility loci identified on chromosome 12 in African Americans. *PLoS One* **6**: e16044.

Boutanaev AM, Kalmykova AI, Shevelyov YY, Nurminsky DI. 2002. Large clusters of co-expressed genes in the Drosophila genome. *Nature* **420**: 666-669.

Bric A, Miething C, Bialucha CU, Scuoppo C, Zender L, Krasnitz A, Xuan Z, Zuber J, Wigler M, Hicks J. 2009. Functional identification of tumor-suppressor genes through an in vivo RNA interference screen in a mouse lymphoma model. *Cancer cell* **16**: 324-335.

Cai X, Evrony GD, Lehmann HS, Elhosary PC, Mehta BK, Poduri A, Walsh CA. 2014. Single-cell, genome-wide sequencing identifies clonal somatic copy-number variation in the human brain. *Cell Rep* **8**: 1280-1289.

Campbell PJ, Pleasance ED, Stephens PJ, Dicks E, Rance R, Goodhead I, Follows GA, Green AR, Futreal PA, Stratton MR. 2008. Subclonal phylogenetic structures in cancer revealed by ultra-deep sequencing. *Proceedings of the National Academy of Sciences* **105**: 13081-13086.

Carbon S, Ireland A, Mungall CJ, Shu SQ, Marshall B, Lewis S. 2009. AmiGO: online access to ontology and annotation data. *Bioinformatics* **25**: 288-289.

Caron H, van Schaik B, van der Mee M, Baas F, Riggins G, van Sluis P, Hermus MC, van Asperen R, Boon K, Voute PA et al. 2001. The human transcriptome map: clustering of highly expressed genes in chromosomal domains. *Science* **291**: 1289-1292.

Carroll SB. 1995. Homeotic genes and the evolution of arthropods and chordates. *Nature* **376**: 479-485.

Carter HB, Partin AW, Walsh PC, Trock BJ, Veltri RW, Nelson WG, Coffey DS, Singer EA, Epstein JI. 2012a. Gleason Score 6 Adenocarcinoma: Should It Be Labeled As Cancer? *J Clin Oncol* **30**: 4294-4296.

Carter SL, Cibulskis K, Helman E, McKenna A, Shen H, Zack T, Laird PW, Onofrio RC, Winckler W, Weir BA. 2012b. Absolute quantification of somatic DNA alterations in human cancer. *Nature biotechnology* **30**: 413-421.

Chen M, Gunel M, Zhao H. 2013. SomatiCA: identifying, characterizing and quantifying somatic copy number aberrations from cancer genome sequencing data. *PloS one* **8**: e78143.

Chen M, Song P, Zou D, Hu X, Zhao S, Gao S, Ling F. 2014. Comparison of multiple displacement amplification (MDA) and multiple annealing and looping-based amplification cycles (MALBAC) in single-cell sequencing. *PLoS One* **9**: e114520.

Chicas A, Wang X, Zhang C, McCurrach M, Zhao Z, Mert O, Dickins RA, Narita M, Zhang M, Lowe SW. 2010. Dissecting the unique role of the retinoblastoma tumor suppressor during cellular senescence. *Cancer cell* **17**: 376-387.

Cho RJ, Campbell MJ, Winzeler EA, Steinmetz L, Conway A, Wodicka L, Wolfsberg TG, Gabrielian AE, Landsman D, Lockhart DJ et al. 1998. A genome-wide transcriptional analysis of the mitotic cell cycle. *Molecular cell* **2**: 65-73.

Cleveland WS. 1981. Lowess - a Program for Smoothing Scatterplots by Robust Locally Weighted Regression. *Am Stat* **35**: 54-54.

Dago AE, Stepansky A, Carlsson A, Luttgen M, Kendall J, Baslan T, Kolatkar A, Wigler M, Bethel K, Gross ME et al. 2014. Rapid phenotypic and genomic change in response to therapeutic pressure in prostate cancer inferred by high content analysis of single circulating tumor cells. *PLoS One* **9**: e101777.

de Bourcy CF, De Vlaminck I, Kanbar JN, Wang J, Gawad C, Quake SR. 2014. A quantitative comparison of single-cell whole genome amplification methods. *PLoS One* **9**: e105585.

Dean FB, Nelson JR, Giesler TL, Lasken RS. 2001. Rapid amplification of plasmid and phage DNA using Phi 29 DNA polymerase and multiply-primed rolling circle amplification. *Genome Res* **11**: 1095-1099.

Dempsey AA, Pabalan N, Tang HC, Liew CC. 2001. Organization of human cardiovascular-expressed genes on chromosomes 21 and 22. *Journal of molecular and cellular cardiology* **33**: 587-591.

Deshwar AG, Vembu S, Yung CK, Jang GH, Stein L, Morris Q. 2015. PhyloWGS: reconstructing subclonal composition and evolution from whole-genome sequencing of tumors. *Genome Biol* **16**: 35.

Dixon JR, Selvaraj S, Yue F, Kim A, Li Y, Shen Y, Hu M, Liu JS, Ren B. 2012. Topological domains in mammalian genomes identified by analysis of chromatin interactions. *Nature* **485**: 376-380.

Felsenstein J, Felenstein J. 2004. Inferring phylogenies.

Fisher R, Pusztai L, Swanton C. 2013. Cancer heterogeneity: implications for targeted therapeutics. *Br J Cancer* **108**: 479-485.

Flicek P, Amode MR, Barrell D, Beal K, Brent S, Carvalho-Silva D, Clapham P, Coates G, Fairley S, Fitzgerald S et al. 2012. Ensembl 2012. *Nucleic acids research* **40**: D84-90.

Fraley C, Raftery AE. 2002. Model-based clustering, discriminant analysis, and density estimation. *Journal of the American statistical Association* **97**: 611-631.

Fraley C, Raftery AE, Murphy TB, Scrucca L. 2012. mclust Version 4 for R: normal mixture modeling for model-based clustering, classification. and Density Estimation Technical Report No.

Friend SH, Bernards R, Rogelj S, Weinberg RA, Rapaport JM, Albert DM, Dryja TP. 1986. A human DNA segment with properties of the gene that predisposes to retinoblastoma and osteosarcoma. *Nature* **323**: 643-646.

Futreal PA, Coin L, Marshall M, Down T, Hubbard T, Wooster R, Rahman N, Stratton MR. 2004. A census of human cancer genes. *Nat Rev Cancer* **4**: 177-183.

Gangnus R, Langer S, Breit E, Pantel K, Speicher MR. 2004. Genomic profiling of viable and proliferative micrometastatic cells from early-stage breast cancer patients. *Clin Cancer Res* **10**: 3457-3464.

Garvin T, Aboukhalil R, Kendall J, Baslan T, Atwal GS, Hicks J, Wigler M, Schatz MC. 2015. Interactive analysis and assessment of single-cell copy-number variations. *Nature methods* **12**: 1058-1060.

Gewinner C, Wang ZC, Richardson A, Teruya-Feldstein J, Etemadmoghadam D, Bowtell D, Barretina J, Lin WM, Rameh L, Salmena L. 2009. Evidence that inositol polyphosphate 4-phosphatase type II is a tumor suppressor that inhibits PI3K signaling. *Cancer cell* **16**: 115-125.

Glöckner S, Buurman H, Kleeberger W, Lehmann U, Kreipe H. 2002. Marked intratumoral heterogeneity of c-myc and cyclinD1 but not of c-erbB2 amplification in breast cancer. *Laboratory investigation* **82**: 1419-1426.

Griffith M, Miller CA, Griffith OL, Krysiak K, Skidmore ZL, Ramu A, Walker JR, Dang HX, Trani L, Larson DE et al. 2015. Optimizing cancer genome sequencing and analysis. *Cell Syst* **1**: 210-223.

Gu W, Zhang F, Lupski JR. 2008. Mechanisms for human genomic rearrangements. *Pathogenetics* **1**: 4.

Gusnanto A, Wood HM, Pawitan Y, Rabbitts P, Berri S. 2012. Correcting for cancer genome size and tumour cell content enables better estimation of copy number alterations from next-generation sequence data. *Bioinformatics* **28**: 40-47.

Hamaï A, Meslin F, Benlalam H, Jalil A, Mehrpour M, Faure F, Lecluse Y, Vielh P, Avril M-F, Robert C. 2008. ICAM-1 has a critical role in the regulation of metastatic melanoma tumor susceptibility to CTL lysis by interfering with PI3K/AKT pathway. *Cancer research* **68**: 9854-9864.

Hanahan D, Weinberg RA. 2011. Hallmarks of cancer: the next generation. *cell* **144**: 646-674.

Hastings P, Ira G, Lupski JR. 2009a. A microhomology-mediated break-induced replication model for the origin of human copy number variation. *PLoS genet* **5**: e1000327.

Hastings P, Lupski JR, Rosenberg SM, Ira G. 2009b. Mechanisms of change in gene copy number. *Nature Reviews Genetics* **10**: 551-564.

Heitzer E, Auer M, Gasch C, Pichler M, Ulz P, Hoffmann EM, Lax S, Waldispuehl-Geigl J, Mauermann O, Lackner C et al. 2013. Complex tumor genomes inferred from single circulating tumor cells by array-CGH and next-generation sequencing. *Cancer Res* **73**: 2965-2975.

Henrichsen CN, Chaignat E, Reymond A. 2009. Copy number variants, diseases and gene expression. *Human molecular genetics* **18**: R1-R8.

Hicks J, Krasnitz A, Lakshmi B, Navin NE, Riggs M, Leibu E, Esposito D, Alexander J, Troge J, Grubor V. 2006. Novel patterns of genome rearrangement and their association with survival in breast cancer. *Genome Research* **16**: 1465-1479.

Hou Y, Fan W, Yan L, Li R, Lian Y, Huang J, Li J, Xu L, Tang F, Xie XS et al. 2013. Genome analyses of single human oocytes. *Cell* **155**: 1492-1506.

Hou Y, Song L, Zhu P, Zhang B, Tao Y, Xu X, Li F, Wu K, Liang J, Shao D et al. 2012. Single-cell exome sequencing and monoclonal evolution of a JAK2-negative myeloproliferative neoplasm. *Cell* **148**: 873-885.

Hsu CM, Hsu YA, Tsai Y, Shieh FK, Huang SH, Wan L, Tsai FJ. 2010. Emodin inhibits the growth of hepatoma cells: finding the common anti-cancer pathway using Huh7, Hep3B, and HepG2 cells. *Biochem Biophys Res Commun* **392**: 473-478.

Hu W, Feng Z, Atwal GS, Levine AJ. 2008. p53: a new player in reproduction. *Cell Cycle* **7**: 848-852.

Huang L, Ma F, Chapman A, Lu S, Xie XS. 2015. Single-cell whole-genome amplification and sequencing: Methodology and applications. *Annual review of genomics and human genetics* **16**: 79-102.

Hurst LD, Pal C, Lercher MJ. 2004a. The evolutionary dynamics of eukaryotic gene order. *Nat Rev Genet* **5**: 299-310.

Hurst LD, Pál C, Lercher MJ. 2004b. The evolutionary dynamics of eukaryotic gene order. *Nature Reviews Genetics* **5**: 299-310.

Isobe T, Baba E, Arita S, Komoda M, Tamura S, Shirakawa T, Ariyama H, Takaishi S, Kusaba H, Ueki T et al. 2011. Human STEAP3 maintains tumor growth under hypoferric condition. *Exp Cell Res* **317**: 2582-2591.

Ji L, Minna JD, Roth JA. 2005. 3p21.3 tumor suppressor cluster: prospects for translational applications. *Future Oncol* **1**: 79-92.

Jin L, Nei M. 1990. Limitations of the evolutionary parsimony method of phylogenetic analysis. *Mol Biol Evol* **7**: 82-102.

Kanehisa M, Goto S, Sato Y, Furumichi M, Tanabe M. 2012. KEGG for integration and interpretation of large-scale molecular data sets. *Nucleic acids research* **40**: D109-D114.

Kim KI, Simon R. 2014. Using single cell sequencing data to model the evolutionary history of a tumor. *BMC Bioinformatics* **15**: 27.

Kinsella RJ, Kähäri A, Haider S, Zamora J, Proctor G, Spudich G, Almeida-King J, Staines D, Derwent P, Kerhornou A et al. 2011. Ensembl BioMarts: a hub for data retrieval across taxonomic space. *Database* **2011**.

Kirkness EF, Grindberg RV, Yee-Greenbaum J, Marshall CR, Scherer SW, Lasken RS, Venter JC. 2013. Sequencing of isolated sperm cells for direct haplotyping of a human genome. *Genome Res* **23**: 826-832.

Knouse KA, Wu J, Amon A. 2016. Assessment of megabase-scale somatic copy number variation using single cell sequencing. *Genome Research*: gr. 198937.198115.

Kong F, Tong R, Jia L, Wei W, Miao X, Zhao X, Sun W, Yang G, Zhao C. 2011. OVCA1 inhibits the proliferation of epithelial ovarian cancer cells by decreasing cyclin D1 and increasing p16. *Molecular and cellular biochemistry* **354**: 199-205.

Krimpenfort P, Ijpenberg A, Song JY, van der Valk M, Nawijn M, Zevenhoven J, Berns A. 2007. p15Ink4b is a critical tumour suppressor in the absence of p16Ink4a. *Nature* **448**: 943-946.

Krzywinski M, Schein J, Birol İ, Connors J, Gascoyne R, Horsman D, Jones SJ, Marra MA. 2009. Circos: an information aesthetic for comparative genomics. *Genome Research* **19**: 1639-1645.

Kuhner MK, Felsenstein J. 1994. A simulation comparison of phylogeny algorithms under equal and unequal evolutionary rates. *Mol Biol Evol* **11**: 459-468.

-. 1995. A Simulation Comparison of Phylogeny Algorithms under Equal and Unequal Evolutionary Rates (Vol 11, Pg 459, 1994). *Molecular Biology and Evolution* **12**: 525-525.

Kumar S, Gadagkar SR. 2000. Efficiency of the neighbor-joining method in reconstructing deep and shallow evolutionary relationships in large phylogenies. *J Mol Evol* **51**: 544-553.

Lai WR, Johnson MD, Kucherlapati R, Park PJ. 2005. Comparative analysis of algorithms for identifying amplifications and deletions in array CGH data. *Bioinformatics* **21**: 3763-3770.

Langmead B, Salzberg SL. 2012. Fast gapped-read alignment with Bowtie 2. *Nature methods* **9**: 357-359.

Langmead B, Trapnell C, Pop M, Salzberg SL. 2009. Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biol* **10**: R25.

Larkin DM, Pape G, Donthu R, Auvil L, Welge M, Lewin HA. 2009. Breakpoint regions and homologous synteny blocks in chromosomes have different evolutionary histories. *Genome Res* **19**: 770-777.

Lasken RS, Stockwell TB. 2007. Mechanism of chimera formation during the Multiple Displacement Amplification reaction. *BMC biotechnology* **7**: 1.

Lee JM, Sonnhammer ELL. 2003. Genomic Gene Clustering Analysis of Pathways in Eukaryotes. *Genome Research* **13**: 875-882.

Leinonen R, Sugawara H, Shumway M, International Nucleotide Sequence Database C. 2011. The sequence read archive. *Nucleic Acids Res* **39**: D19-21.

Lercher MJ, Blumenthal T, Hurst LD. 2003. Coexpression of neighboring genes in Caenorhabditis elegans is mostly due to operons and duplicate genes. *Genome Res* **13**: 238-243.

Lercher MJ, Urrutia AO, Hurst LD. 2002. Clustering of housekeeping genes provides a unified model of gene order in the human genome. *Nat Genet* **31**: 180-183.

Leung ML, Wang Y, Waters J, Navin NE. 2015. SNES: single nucleus exome sequencing. *Genome Biol* **16**: 55.

Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, Marth G, Abecasis G, Durbin R, Genome Project Data Processing S. 2009. The Sequence Alignment/Map format and SAMtools. *Bioinformatics* **25**: 2078-2079.

Li WH, Wolfe KH, Sourdis J, Sharp PM. 1987. Reconstruction of phylogenetic trees and estimation of divergence times under nonconstant rates of evolution. *Cold Spring Harb Symp Quant Biol* **52**: 847-856.

Li Y, Xu X, Song L, Hou Y, Li Z, Tsang S, Li F, Im KM, Wu K, Wu H et al. 2012. Single-cell sequencing analysis characterizes common and cell-lineage-specific mutations in a muscle-invasive bladder cancer. *Gigascience* **1**: 12.

Lu S, Zong C, Fan W, Yang M, Li J, Chapman AR, Zhu P, Hu X, Xu L, Yan L et al. 2012. Probing meiotic recombination and aneuploidy of single sperm cells by whole-genome sequencing. *Science* **338**: 1627-1630.

Lu T, Hano H, Meng C, Nagatsuma K, Chiba S, Ikegami M. 2007. Frequent loss of heterozygosity in two distinct regions, 8p23. 1 and 8p22, in hepatocellular carcinoma. *World journal of gastroenterology: WJG* **13**: 1090-1097.

Macaulay IC, Haerty W, Kumar P, Li YI, Hu TX, Teng MJ, Goolam M, Saurat N, Coupland P, Shirley LM. 2015. G&T-seq: parallel sequencing of single-cell genomes and transcriptomes. *Nature methods* **12**: 519-522.

Malhotra A, Wang Y, Waters J, Chen K, Meric-Bernstam F, Hall IM, Navin NE. 2015. Ploidy-Seq: inferring mutational chronology by sequencing polyploid tumor subpopulations. *Genome Med* **7**: 6.

Malhotra D, Sebat J. 2012. CNVs: harbingers of a rare variant revolution in psychiatric genetics. *Cell* **148**: 1223-1241.

Marusyk A, Almendro V, Polyak K. 2012. Intra-tumour heterogeneity: a looking glass for cancer? *Nature Reviews Cancer*.

Mathiesen RR, Fjelldal R, Liestol K, Due EU, Geigl JB, Riethdorf S, Borgen E, Rye IH, Schneider IJ, Obenauf AC et al. 2012. High-resolution analyses of copy number changes in disseminated tumor cells of patients with breast cancer. *Int J Cancer* **131**: E405-415.

McClatchey AI, Saotome I, Mercer K, Crowley D, Gusella JF, Bronson RT, Jacks T. 1998. Mice heterozygous for a mutation at the Nf2 tumor suppressor locus develop a range of highly metastatic tumors. *Genes & development* **12**: 1121-1133.

McConnell MJ, Lindberg MR, Brennand KJ, Piper JC, Voet T, Cowing-Zitron C, Shumilina S, Lasken RS, Vermeesch JR, Hall IM et al. 2013. Mosaic copy number variation in human neurons. *Science* **342**: 632-637.

McGranahan N, Swanton C. 2015. Biological and therapeutic impact of intratumor heterogeneity in cancer evolution. *Cancer Cell* **27**: 15-26.

Melchor L, Brioli A, Wardell CP, Murison A, Potter NE, Kaiser MF, Fryer RA, Johnson DC, Begum DB, Hulkki Wilson S et al. 2014. Single-cell genetic analysis reveals the composition of initiating clones and phylogenetic patterns of branching and parallel evolution in myeloma. *Leukemia* **28**: 1705-1715.

Meyerson M, Gabriel S, Getz G. 2010. Advances in understanding cancer genomes through second-generation sequencing. *Nature Reviews Genetics* **11**: 685-696.

Mihaescu R, Levy D, Pachter L. 2007. Why Neighbor-Joining Works. *Algorithmica* **54**: 1-24.

Milo R, Jorgensen P, Moran U, Weber G, Springer M. 2010. BioNumbers--the database of key numbers in molecular and cell biology. *Nucleic Acids Res* **38**: D750-753.

Murugaesu N, Chew SK, Swanton C. 2013. Adapting clinical paradigms to the challenges of cancer clonal evolution. *Am J Pathol* **182**: 1962-1971.

Navin N, Hicks J. 2011. Future medical applications of single-cell sequencing in cancer. *Genome Med* **3**: 31.

Navin N, Kendall J, Troge J, Andrews P, Rodgers L, McIndoo J, Cook K, Stepansky A, Levy D, Esposito D et al. 2011. Tumour evolution inferred by single-cell sequencing. *Nature* **472**: 90-94.

Navin N, Krasnitz A, Rodgers L, Cook K, Meth J, Kendall J, Riggs M, Eberling Y, Troge J, Grubor V. 2010. Inferring tumor progression from genomic heterogeneity. *Genome Research* **20**: 68-80.

Navin NE. 2014. Cancer genomics: one cell at a time. *Genome Biol* **15**: 452.

Navin NE. 2015. The first five years of single-cell cancer genomics and beyond. *Genome research* **25**: 1499-1507.

Ng M-P, Vergara I, Frech C, Chen Q, Zeng X, Pei J, Chen N. 2009. OrthoClusterDB: an online platform for synteny blocks. *BMC Bioinformatics* **10**: 192.

Ni X, Zhuo M, Su Z, Duan J, Gao Y, Wang Z, Zong C, Bai H, Chapman AR, Zhao J et al. 2013. Reproducible copy number variation patterns among single circulating tumor cells of lung cancer patients. *Proc Natl Acad Sci U S A* **110**: 21083-21088.

Nickel JC, Speakman M. 2012. Should We Really Consider Gleason 6 Prostate Cancer? *Bju Int* **109**: E16-646.

Nowell PC. 1976. The clonal evolution of tumor cell populations. *Science* **194**: 23-28.

Oesper L, Mahmoody A, Raphael BJ. 2013. THetA: inferring intra-tumor heterogeneity from high-throughput DNA sequencing data. *Genome Biol* **14**: R80.

Olshen AB, Venkatraman ES, Lucito R, Wigler M. 2004. Circular binary segmentation for the analysis of array-based DNA copy number data. *Biostatistics* **5**: 557-572.

Ouedraogo M, Bettembourg C, Bretaudeau A, Sallou O, Diot C, Demeure O, Lecerf F. 2012. The Duplicated Genes Database: Identification and Functional Annotation of Co-Localised Duplicated Genes across Genomes. *PloS one* **7**: e50653.

Pandis N, Jin Y, Gorunova L, Petersson C, Bardi G, Idvall I, Johansson B, Ingvar C, Mandahl N, Mitelman F. 1995. Chromosome analysis of 97 primary breast carcinomas: identification of eight karyotypic subgroups. *Genes, chromosomes and cancer* **12**: 173-185.

Pantou D, Rizou H, Tsarouha H, Pouli A, Papanastasiou K, Stamatellou M, Trangas T, Pandis N, Bardi G. 2005. Cytogenetic manifestations of multiple myeloma heterogeneity. *Genes, chromosomes and cancer* **42**: 44-57.

Paradis E, Claude J, Strimmer K. 2004. APE: Analyses of Phylogenetics and Evolution in R language. *Bioinformatics* **20**: 289-290.

Park J, Kang SI, Lee SY, Zhang XF, Kim MS, Beers LF, Lim DS, Avruch J, Kim HS, Lee SB. 2010. Tumor suppressor ras association domain family 5 (RASSF5/NORE1) mediates death receptor ligand-induced apoptosis. *J Biol Chem* **285**: 35029-35038.

Park JY, Kricka LJ, Fortina P. 2013. Next-generation sequencing in the clinic. *Nature biotechnology* **31**: 990-992.

Patel AP, Tirosh I, Trombetta JJ, Shalek AK, Gillespie SM, Wakimoto H, Cahill DP, Nahed BV, Curry WT, Martuza RL. 2014. Single-cell RNA-seq highlights intratumoral heterogeneity in primary glioblastoma. *Science* **344**: 1396-1401.

Pathare S, Schäffer AA, Beerenwinkel N, Mahimkar M. 2009. Construction of oncogenetic tree models reveals multiple pathways of oral cancer progression. *International journal of cancer* **124**: 2864-2871.

Paul A, Cai Y, Atwal GS, Huang ZJ. 2012a. Developmental Coordination of Gene Expression between Synaptic Partners During GABAergic Circuit Assembly in Cerebellar Cortex. *Frontiers in neural circuits* **6**: 37.

Paul A, Cai Y, Atwal GS, Huang ZJ. 2012b. Developmental coordination of gene expression between synaptic partners during GABAergic circuit assembly in cerebellar cortex. *Frontiers in Neural Circuits* **6**.

Penny D, Foulds LR, Hendy MD. 1982. Testing the theory of evolution by comparing phylogenetic trees constructed from five different protein sequences. *Nature* **297**: 197-200.

Perou CM, Sørlie T, Eisen MB, van de Rijn M, Jeffrey SS, Rees CA, Pollack JR, Ross DT, Johnsen H, Akslen LA. 2000. Molecular portraits of human breast tumours. *Nature* **406**: 747-752.

Petkov PM, Graber JH, Churchill GA, DiPetrillo K, King BL, Paigen K. 2005a. Evidence of a Large-Scale Functional Organization of Mammalian Chromosomes. *PLoS Genet* **1**: e33.

Petkov PM, Graber JH, Churchill GA, DiPetrillo K, King BL, Paigen K. 2005b. Evidence of a large-scale functional organization of mammalian chromosomes. *PLoS genetics* **1**: e33.

Pierce GB, Speers WC. 1988. Tumors as caricatures of the process of tissue renewal: prospects for therapy by directing differentiation. *Cancer research* **48**: 1996-2004.

Pietras A. 2011. Cancer stem cells in tumor heterogeneity. *Advances in cancer research* **112**: 256.

Proost S, Fostier J, De Witte D, Dhoedt B, Demeester P, Van de Peer Y, Vandepoele K. 2012. i-ADHoRe 3.0—fast and sensitive detection of genomic homology in extremely large data sets. *Nucleic acids research* **40**: e11-e11.

Redon R, Ishikawa S, Fitch KR, Feuk L, Perry GH, Andrews TD, Fiegler H, Shapero MH, Carson AR, Chen W. 2006. Global variation in copy number in the human genome. *nature* **444**: 444-454.

Reimann M, Lee S, Loddenkemper C, Dörr JR, Tabor V, Aichele P, Stein H, Dörken B, Jenuwein T, Schmitt CA. 2010. Tumor stroma-derived TGF-β limits myc-driven lymphomagenesis via Suv39h1-dependent senescence. *Cancer cell* **17**: 262-272.

Revanna KV, Krishnakumar V, Dong Q. 2009. A web-based software system for dynamic gene cluster comparison across multiple genomes. *Bioinformatics* **25**: 956-957.

Robinson D, Foulds LR. 1981. Comparison of phylogenetic trees. *Mathematical Biosciences* **53**: 131-147.

Rödelsperger C, Dieterich C. 2008. Syntenator: multiple gene order alignments with a gene-specific scoring function. *Algorithms Mol Biol* **3**: 14.

Rottmann S, Speckgens S, Luscher-Firzlaff J, Luscher B. 2008. Inhibition of apoptosis by MAD1 is mediated by repression of the PTEN tumor suppressor gene. *FASEB J* **22**: 1124-1134.

Roy PJ, Stuart JM, Lund J, Kim SK. 2002. Chromosomal clustering of muscle-expressed genes in Caenorhabditis elegans. *Nature* **418**: 975-979.

Ryser H. 1987. Combinatorial properties of matrices of zeros and ones. in *Classic Papers in Combinatorics*, pp. 269-275. Springer.

Saha A, Robertson ES. 2011. Functional modulation of the metastatic suppressor Nm23-H1 by oncogenic viruses. *FEBS letters* **585**: 3174-3184.

Saitou N, Nei M. 1987. The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Mol Biol Evol* **4**: 406-425.

Salomoni P, Pandolfi PP. 2002. The role of PML in tumor suppression. *Cell* **108**: 165-170.

Satopää V, Albrecht J, Irwin D, Raghavan B. 2011. Finding a" Kneedle" in a Haystack: Detecting Knee Points in System Behavior. in *Distributed Computing Systems Workshops (ICDCSW), 2011 31st International Conference on*, pp. 166-171. IEEE.

Saunders NA, Simpson F, Thompson EW, Hill MM, Endo-Munoz L, Leggatt G, Minchin RF, Guminski A. 2012. Role of intratumoural heterogeneity in cancer drug resistance: molecular and clinical perspectives. *EMBO Mol Med* **4**: 675-684.

Sauter G, Moch H, Gasser TC, Mihatsch MJ, Waldman FM. 1995. Heterogeneity of chromosome 17 and erbB‐2 gene copy number in primary and metastatic bladder cancer. *Cytometry* **21**: 40-46.

Schliep KP. 2011. phangorn: phylogenetic analysis in R. *Bioinformatics* **27**: 592-593.

Scuoppo C, Miething C, Lindqvist L, Reyes J, Ruse C, Appelmann I, Yoon S, Krasnitz A, Teruya-Feldstein J, Pappin D. 2012. A tumour suppressor network relying on the polyamine-hypusine axis. *Nature*.

Sebat J, Lakshmi B, Troge J, Alexander J, Young J, Lundin P, Månér S, Massa H, Walker M, Chi M. 2004. Large-scale copy number polymorphism in the human genome. *Science* **305**: 525-528.

Sen A, Srivastava MS. 1975. On tests for detecting change in mean. *The Annals of statistics*: 98-108.

Shapiro E, Biezuner T, Linnarsson S. 2013. Single-cell sequencing-based technologies will revolutionize whole-organism science. *Nat Rev Genet* **14**: 618-630.

Shapiro JR, Yung W-KA, Shapiro WR. 1981. Isolation, karyotype, and clonal growth of heterogeneous subpopulations of human malignant gliomas. *Cancer research* **41**: 2349-2359.

Sherr CJ. 2004. Principles of tumor suppression. *Cell* **116**: 235-246.

Shipitsin M, Campbell LL, Argani P, Weremowicz S, Bloushtain-Qimron N, Yao J, Nikolskaya T, Serebryiskaya T, Beroukhim R, Hu M. 2007. Molecular definition of breast tumor heterogeneity. *Cancer cell* **11**: 259-273.

Shlien A, Malkin D. 2009a. Copy number variations and cancer. *Genome medicine* **1**: 1-9.

Shlien A, Malkin D. 2009b. Copy number variations and cancer. *Genome Med* **1**: 62.

Singer GAC, Lloyd AT, Huminiecki LB, Wolfe KH. 2005. Clusters of Co-expressed Genes in Mammalian Genomes Are Conserved by Natural Selection. *Molecular Biology and Evolution* **22**: 767-775.

Sinha A, Meller J. 2007. Cinteny: flexible analysis and visualization of synteny and genome rearrangements in multiple organisms. *BMC Bioinformatics* **8**: 82.

Smits SA, Ouverney CC. 2010. jsPhyloSVG: a javascript library for visualizing interactive and vector-based phylogenetic trees on the web. *PLoS One* **5**: e12267.

Solimini NL, Xu Q, Mermel CH, Liang AC, Schlabach MR, Luo J, Burrows AE, Anselmo AN, Bredemeyer AL, Li MZ. 2012. Recurrent Hemizygous Deletions in Cancers May Optimize Proliferative Potential. *Science* **337**: 104-109.

Sourdis J, Nei M. 1988. Relative efficiencies of the maximum parsimony and distance-matrix methods in obtaining the correct phylogenetic tree. *Mol Biol Evol* **5**: 298-311.

Tamura K, Nei M, Kumar S. 2004. Prospects for inferring very large phylogenies by using the neighbor-joining method. *P Natl Acad Sci USA* **101**: 11030-11035.

Tateno Y, Takezaki N, Nei M. 1994. Relative efficiencies of the maximum-likelihood, neighbor-joining, and maximum-parsimony methods when substitution rate varies with site. *Mol Biol Evol* **11**: 261-277.

Teixeira MR, Pandis N, Bardi G, Andersen JA, Mitelman F, Heim S. 1995. Clonal heterogeneity in breast cancer: Karyotypic comparisons of multiple intra—and extra—tumorous samples from 3 patients. *International journal of cancer* **63**: 63-68.

Telenius H, Carter NP, Bebb CE, Nordenskjold M, Ponder BA, Tunnacliffe A. 1992. Degenerate oligonucleotide-primed PCR: general amplification of target DNA by a single degenerate primer. *Genomics* **13**: 718-725.

Trinklein ND, Aldred SF, Hartman SJ, Schroeder DI, Otillar RP, Myers RM. 2004. An abundance of bidirectional promoters in the human genome. *Genome Res* **14**: 62-66.

Ulmer A, Schmidt-Kittler O, Fischer J, Ellwanger U, Rassner G, Riethmuller G, Fierlbeck G, Klein CA. 2004. Immunomagnetic enrichment, genomic characterization, and prognostic impact of circulating melanoma cells. *Clin Cancer Res* **10**: 531-537.

Urbschat S, Rahnenfuehrer J, Henn W, Feiden W, Wemmert S, Linsler S, Zang KD, Oertel J, Ketter R. 2011. Clonal cytogenetic progression within intratumorally heterogeneous meningiomas predicts tumor recurrence. *Int J Oncol* **39**: 1601-1608.

Van Berkum NL, Lieberman-Aiden E, Williams L, Imakaev M, Gnirke A, Mirny LA, Dekker J, Lander ES. 2010. Hi-C: a method to study the three-dimensional architecture of genomes.

Veeriah S, Brennan C, Meng S, Singh B, Fagin JA, Solit DB, Paty PB, Rohle D, Vivanco I, Chmielecki J. 2009. The tyrosine phosphatase PTPRD is a tumor suppressor that is frequently inactivated and mutated in glioblastoma and other human cancers. *Proceedings of the National Academy of Sciences* **106**: 9435-9440.

Venkatraman ES, Olshen AB. 2007. A faster circular binary segmentation algorithm for the analysis of array CGH data. *Bioinformatics* **23**: 657-663.

Vilella AJ, Severin J, Ureta-Vidal A, Heng L, Durbin R, Birney E. 2009. EnsemblCompara GeneTrees: Complete, duplication-aware phylogenetic trees in vertebrates. *Genome Research* **19**: 327-335.

Voet T, Kumar P, Van Loo P, Cooke SL, Marshall J, Lin M-L, Esteki MZ, Van der Aa N, Mateiu L, McBride DJ. 2013. Single-cell paired-end genome sequencing reveals structural variation per cell cycle. *Nucleic acids research* **41**: 6119-6138.

Wajapeyee N, Serra RW, Zhu X, Mahalingam M, Green MR. 2010. Role for IGFBP7 in senescence induction by BRAF. *Cell* **141**: 746-747.

Wang J, Fan HC, Behr B, Quake SR. 2012a. Genome-wide single-cell analysis of recombination activity and de novo mutation rates in human sperm. *Cell* **150**: 402-412.

Wang Y, Tang H, DeBarry JD, Tan X, Li J, Wang X, Lee T, Jin H, Marler B, Guo H. 2012b. MCScanX: a toolkit for detection and evolutionary analysis of gene synteny and collinearity. *Nucleic acids research* **40**: e49-e49.

Wang Y, Waters J, Leung ML, Unruh A, Roh W, Shi X, Chen K, Scheet P, Vattathil S, Liang H et al. 2014. Clonal evolution in breast cancer revealed by single nucleus genome sequencing. *Nature* **512**: 155-160.

Westbrook TF, Martin ES, Schlabach MR, Leng Y, Liang AC, Feng B, Zhao JJ, Roberts TM, Mandel G, Hannon GJ et al. 2005. A genetic screen for candidate tumor suppressors identifies REST. *Cell* **121**: 837-848.

Wigler M. 2012. Broad applications of single-cell nucleic acid analysis in biomedical research. *Genome Med* **4**: 79.

Willenbrock H, Fridlyand J. 2005. A comparison study: applying segmentation to array CGH data for downstream analyses. *Bioinformatics* **21**: 4084-4091.

Williams TL, Moret BM. 2003. An investigation of phylogenetic likelihood methods. in *Bioinformatics and Bioengineering, 2003 Proceedings Third IEEE Symposium on*, pp. 79-86. IEEE.

Xu L, Li X, Chu ES, Zhao G, Go MY, Tao Q, Jin H, Zeng Z, Sung JJ, Yu J. 2012. Epigenetic inactivation of BCL6B, a novel functional tumour suppressor for gastric cancer, is associated with poor survival. *Gut* **61**: 977-985.

Xue W, Kitzing T, Roessler S, Zuber J, Krasnitz A, Schultz N, Revill K, Weissmueller S, Rappaport AR, Simon J. 2012a. A cluster of cooperating tumor-suppressor gene candidates in chromosomal deletions. *Proceedings of the National Academy of Sciences* **109**: 8212-8217.

Xue W, Kitzing T, Roessler S, Zuber J, Krasnitz A, Schultz N, Revill K, Weissmueller S, Rappaport AR, Simon J et al. 2012b. A cluster of cooperating tumor-suppressor gene candidates in chromosomal deletions. *Proceedings of the National Academy of Sciences*.

Xue W, Krasnitz A, Lucito R, Sordella R, VanAelst L, Cordon-Cardo C, Singer S, Kuehnel F, Wigler M, Powers S. 2008. DLC1 is a chromosome 8p tumor suppressor whose loss promotes hepatocellular carcinoma. *Genes & development* **22**: 1439-1444.

Xue Z, Huang K, Cai C, Cai L, Jiang C-y, Feng Y, Liu Z, Zeng Q, Cheng L, Sun YE. 2013. Genetic programs in human and mouse early embryos revealed by single-cell RNA [thinsp] sequencing. *Nature* **500**: 593-597.

Yang Y, Fu LM. 2003. TSGDB: a database system for tumor suppressor genes. *Bioinformatics* **19**: 2311-2312.

Yang YS, Song HD, Shi WJ, Hu RM, Han ZG, Chen JL. 2002. Chromosome localization analysis of genes strongly expressed in human visceral adipose tissue. *Endocrine* **18**: 57-66.

Yi G, Sze SH, Thon MR. 2007. Identifying clusters of functionally related genes in genomes. *Bioinformatics* **23**: 1053-1060.

Yung WK, Shapiro JR, Shapiro WR. 1982. Heterogeneous chemosensitivities of subpopulations of human glioma cells in culture. *Cancer Res* **42**: 992-998.

Zack TI, Schumacher SE, Carter SL, Cherniack AD, Saksena G, Tabak B, Lawrence MS, Zhang CZ, Wala J, Mermel CH et al. 2013. Pan-cancer patterns of somatic copy number alteration. *Nat Genet* **45**: 1134-1140.

Zeisel A, Muñoz-Manchado AB, Codeluppi S, Lönnerberg P, La Manno G, Juréus A, Marques S, Munguba H, He L, Betsholtz C. 2015. Cell types in the mouse cortex and hippocampus revealed by single-cell RNA-seq. *Science* **347**: 1138-1142.

Zender L, Xue W, Zuber J, Semighini CP, Krasnitz A, Ma B, Zender P, Kubicka S, Luk JM, Schirmacher P. 2008a. An oncogenomics-based in vivo RNAi screen identifies tumor suppressors in liver cancer. *Cell* **135**: 852-864.

Zender L, Xue W, Zuber J, Semighini CP, Krasnitz A, Ma B, Zender P, Kubicka S, Luk JM, Schirmacher P et al. 2008b. An oncogenomics-based in vivo RNAi screen identifies tumor suppressors in liver cancer. *Cell* **135**: 852-864.

Zhan L, Rosenberg A, Bergami KC, Yu M, Xuan Z, Jaffe AB, Allred C, Muthuswamy SK. 2008. Deregulation of scribble promotes mammary tumorigenesis and reveals a role for cell polarity in carcinoma. *Cell* **135**: 865-878.

Zhang F, Gu W, Hurles ME, Lupski JR. 2009. Copy number variation in human health, disease, and evolution. *Annual review of genomics and human genetics* **10**: 451.

Zilfou JT, Lowe SW. 2009. Tumor suppressive functions of p53. *Cold Spring Harb Perspect Biol* **1**: a001883.

Zong C, Lu S, Chapman AR, Xie XS. 2012. Genome-wide detection of single-nucleotide and copy-number variations of a single human cell. *Science* **338**: 1622-1626.