Statistical Modeling and Analysis for Biomedical Applications

by

Christine Ho

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Statistics

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Associate Professor Haiyan Huang, Chair
Associate Professor Elizabeth Purdom
Professor Lewis Feldman

Fall 2016

Statistical Modeling and Analysis for Biomedical Applications

Abstract

Statistical Modeling and Analysis for Biomedical Applications

by

Christine Ho

Doctor of Philosophy in Statistics

University of California, Berkeley

Associate Professor Haiyan Huang, Chair

This dissertation discusses approaches to two different applied statistical challenges arising from the fields of genomics and biomedical research. The first takes advantage of the richness of whole genome sequencing data, which can uncover both regions of chromosomal aberration and highly specific information on point mutations. We propose a method to reconstruct parts of a tumor's history of chromosomal aberration using only data from a single time-point. We provide an application of the method, which was the first of its kind, to data from eight patients with squamous cell skin cancer, in which we were able to find that knockout of the tumor suppressor gene TP53 occur early in that cancer type.

While the first chapter highlights what's possible with a deep analysis of data from a single patient, the second chapter of this dissertation looks at the opposite situation, aggregating data from several patients to identify gene expression signals for disease phenotypes. In this chapter, we provide a method for hierarchical multilabel classification from several separate classifiers for each node in the hierarchy. The first calls produced by our method improve upon the state-of-the-art, resulting in better performance in the early part of the precision-recall curve. We apply the method to disease classifiers constructed from public microarray data, and whose relationships to each other are given in a known medical hierarchy.

Wild Geese

You do not have to be good.
You do not have to walk on your knees
for a hundred miles through the desert repenting.
You only have to let the soft animal of your body love what it loves.
Tell me about despair, yours, and I will tell you mine.
Meanwhile the world goes on.
Meanwhile the sun and the clear pebbles of the rain
are moving across the landscapes,
over the prairies and the deep trees,
the mountains and the rivers.
Meanwhile the wild geese, high in the clean blue air,
are heading home again.
Whoever you are, no matter how lonely,
the world offers itself to your imagination,
calls to you like the wild geese, harsh and exciting
over and over announcing your place
in the family of things.

- Mary Oliver

Contents

List of Figures

List of Tables

Acknowledgments

My advisors, Haiyan Huang and Elizabeth Purdom, have been incredibly patient and supportive. I admire them for their research skills, resilience, and adaptability, and I am proud to have had them as my advisors.

Mom and Dad, I am here today because of your foresight and planning–you spent so many thankless 16-hour days in the restaurant and made countless sacrifices so I could have this education. Through your actions, you have taught me the meaning of dedication, faith, and generosity.

Within the Statistics department, I am grateful for the mentorship of Deborah Nolan, Terry Speed, Shobhana Stoyanov, and Bin Yu. I thank the administrative and computing staff of the department for working so hard to support their students and researchers. I would especially like to acknowledge the efforts of Ryan Lovett, Mary Melinn, Chris Paciorek, La Shana Porlaris, and Denise Yee.

I have been extremely lucky to have had the support of the following special people over the years: Sandra Backovich, Jaren Haber and his lovely family, Marla Johnson, Jeff Kitzes, Po-Ling Loh, Sanaz Mobasseri, and Priya Shimpi. Two communities also stand out in shaping my graduate experience: the AIDS Life/Cycle community a.k.a. "The Love Bubble" and the mindful people at the Empty Gate Zen Center. Finally, I would like to thank Jeremy Rom for always understanding, making me laugh, and giving me the encouragement I needed to make it through. I love you.

To those I failed to mention here (an oversight of mine, I'm sure), I thank you for being a part of the most transformative eight years of my life. It wouldn't have been the same without you!

Chapter 1

Introduction

This dissertation is split into two chapters, discussing two projects with applications to genomics and the biomedical sciences. In many ways, these projects represent two ends of a spectrum: the first project is about deeply mining sequencing data from a single timepoint in a tumor to piece together clues about its history. Based on the simple idea of using a region's mutation rate as a timestamp, the main challenges in statistical modeling were describing and accounting for the several sources of variability in sequencing experiments. On the other hand, the second project pulls away from a single patient view and is instead about extracting signal from data from hundreds of patients: in the application of our focus, we describe a hierarchical classifier for approximately 100 diseases based on public microarray gene expression data collected from 200 patients. The hierarchical structure comes from the relationships between the disease labels, where some disease labels are specific instances of a broader term. In this setting, the primary challenge is arriving at a hierarchically consistent class assignment based on the output of 100 separately trained classifiers for each disease, each with different performance characteristics.

The statistical methods proposed in these projects are unified by the common theme of ranking, whether that means estimating ranks as for the first project or standardizing incomparable data so that they can be ranked in the second project. In the first chapter, we are concerned with estimating a temporal ranking well in order to reconstruct tumor mutation history. The second chapter has a more methodological focus, and is centered around an algorithm for ranking classifier outputs to produce label assignments in hierarchical multilabel classification.

In the two sections that follow, we provide a high-level overview of the contents of each chapter. We avoid delving into the specifics of the data type, biological context, or mathematics in this overview, and refer the reader to the respective chapters for these details.

1.1 Temporal ordering of chromosomal aberrations in cancer using sequencing data

In this chapter, we use sequencing data on point mutations in regions of chromosomal aberration to reconstruct the temporal order in which the aberrations occurred. If we assume that the point mutations are acquired at a constant rate throughout the history of the tumor, we can establish a model linking the types of mutations in a region to its relative place in time in the history of the tumor: in very simple terms, this follows from reasoning that a region with more mutations than another would have required more time to accumulate them.

A model for temporal ordering is only possible because chromosomal aberrations affect point mutations by altering their allele frequency, or the fraction of copies that they appear on. The allele frequency provides key information on the mutation, like whether the point mutation occurred before or after the first event in the chromosomal aberration. Point mutations can thus be grouped by allele frequency, and regions of chromosomal aberration can be temporally ranked by careful accounting of its point mutations.

We develop a model relating the types of point mutations in a region to their relative time in tumor history, given that we can precisely identify the kind of aberration that has taken place. In practice, these mutations cannot be observed entirely: their allele frequencies are estimated from sequencing data, thus the "group" that each mutation belongs to must also be inferred.

A statistical challenge is modeling and accounting for this sequencing variability. We model the point mutation data, with different distributions that depend on the true allele frequency or group membership of the mutation, which we set as a latent variable. In practice, these allele frequencies are sensitive to tumor sample contamination by normal cells. Thus, we also model and provide an estimator for the degree of contamination, which we use to adjust the distributions of the point mutation data accordingly.

Although our method permits estimating the sequence of events in a region, typically it is only the time of the first event that is of greatest biological interest. Even so, the complexity of the data makes it difficult to obtain analytical results on the performance of the estimator we propose. We turn to simulations of Purdom et al. [2013], which shed light on the bias and variance of our estimates. We evaluate the coverage of bootstrap confidence intervals constructed for the estimates on the same metrics.

Since the method first appeared in Durinck et al. [2011] for the analysis of copy-neutral loss of heterozygosity (CNLOH) events in skin cancer samples, an extension and variant of the technique appeared in Greenman et al. [2012], which we refer to as partial maximum likelihood (MLE). Likewise, an extension of the CNLOH timing method to other regions was introduced in Purdom et al. [2013], to which we contributed. The simulations discussed compare the partial MLE technique against the full MLE technique, as well as a Bayesian estimation approach intended to mitigate the instability of our estimates for parameters at the boundary.

The chapter concludes with a discussion of the method applied to skin cancer in Durinck

et al. [2011], and an attempt to empirically validate the assumption of constant mutation rate, upon which our notion of mutation rate-based timing rests. Though we do not discuss it in the chapter, the method has also been applied to ovarian cancer in Purdom et al. [2013] and primary breast cancer (cell lines) in Greenman et al. [2012].

1.2 Hierarchical multilabel classification with local precision rates

In this chapter, we start with labels that are related to each other by a directed acyclic graph or tree. We assume that we have trained a classifier for each of these labels, and the classifiers produce a monotonic score as output. When the classifiers are each run separately, they may produce assignments that are not consistent with the hierarchy.

A hierarchical classifier built upon several classifiers for each label is typically called a local classifier (as opposed to a flat or global classifier) [Silla Jr and Freitas, 2011]. In this setting, the most common way to reconcile the decisions of these classifiers is to perform a Bayesian error correction, that is, finding the assignment that maximizes $P(Q|\hat{Q})$, where Q and $\hat{Q}$ represent the true and estimated class labels, respectively. However, the state of the art in hierarchical classification with multiple labels is global classification, wherein the labels are learned jointly during training. The most common approach in this case is based on decision trees.

We propose a method motivated by the challenges encountered with Bayesian error correction in the setting of disease diagnosis with public gene expression microarray data [Huang et al., 2010]. In this application, public gene expression data collected from close to 200 patients were used to train about 100 classifiers for different diseases. The diseases were related by the hierarchy given in the Unified Medical Language System.

In the non-hierarchical setting, Jiang et al. [2014] found that being able to account for differences in classifier quality explicitly improved the precision-recall curve. Their method involves using a new statistic, the local precision rate, in lieu of the classifier score. The local precision rate is a transformation of the classifier score, and estimating it also requires estimating the precision function for each classifier. Although their method is not intended for hierarchical data, their demonstrate improved overall precision-recall (pooling all assignments from all patients together) on the disease diagnosis data set.

We extend the results of Jiang et al. [2014] to the hierarchical setting, and find a precision-recall improvement over the Bayesian method of Huang et al. [2010] and the original method of Jiang et al. [2014] from aligning the assignments with the hierarchical. Simulations show that our method has comparable performance to state-of-the-art global classifiers in the early part of the precision-recall curve.

The proposed method consists of an algorithm for sorting the local precision rates that leaves the hierarchical relationships intact. The method is motivated by the theoretical optimality of the algorithm in a limited setting; when the assumptions for optimality are

not met, the method performs as well as the original non-hierarchical method of Jiang et al. [2014]. We also compare several methods for estimating the local precision rate and advise on the best method to use under varying levels of class imbalance and data size.

Chapter 2

Temporal ordering of chromosomal aberrations using point mutation data from sequencing

2.1 Introduction

Cancer is a disease characterized by tumors that develop as a result of dysfunction in the cells' regulatory processes. Depending on the type of cancer, a typical tumor contains tens to thousands of somatic and germline genomic abnormalities, such as chromosome-level changes in copy number, structural rearrangements (e.g., translocations, inversions, fusions), or single nucleotide mutations [Collisson et al., 2012]. Driver mutations are a subset of somatic mutations causally implicated in tumor progression, and thus preferentially selected for during oncogenesis. This term distinguishes them from "passenger" mutations which have no meaningful causal link [Hanahan and Weinberg, 2011]. Identifying driver mutations and the genes they tend to occur on is important for understanding cancer etiology and developing targeted treatments. For example, if a specialist can identify that a patient has a driver mutation on an oncogene, which is a gene known to have enhanced activity in tumors, then treatment targeted at inhibiting that gene could kill tumor cells that require its activation to survive and propagate [Vogelstein and Kinzler, 2004].

Because driver mutations are required for tumor progression, they are expected to be significantly represented in cancer. Therefore, efforts to identify driver mutations have centered around finding patterns of recurrence across patients. The advent of sequencing technology has allowed researchers to identify genes with a significantly elevated rate of somatic point mutations. Since 2006, the Sanger Institute has maintained a Catalogue of Somatic Mutations in Cancer (COSMIC) with the central aim of reporting their frequency, so as to distinguish possible driver mutations from 'passengers' [Forbes et al., 2006]. In another collaborative effort, The Cancer Genome Atlas (TCGA) project continues to collect several types of data on patients for use in integrative analyses, including identifying frequent

or significantly mutated genes [McLendon et al., 2008, Network et al., 2011]. Before the adoption of sequencing technology in cancer genomics, researchers were limited to examining somatic changes via microarray experiments or older techniques like fluorescence *in situ* hybridization (FISH). Nowadays, most methodological research is intended for data from sequencing experiments, but methods for microarray still remains an active area of research today [Brodeur et al., 1982, Huang et al., 2012, Newton et al., 1994, 1998, Newton and Lee, 2000, Taylor et al., 2008].

Through these approaches, researchers have identified several genes and chromosomal changes associated with cancer, but for many, their specific roles in tumor progression remain unclear. The order in which tumors acquire their changes can shed light on how they participate in oncogenesis: early mutation suggests that the gene helps initiate tumor growth, whereas a later mutation suggests its role is to sustain tumor progression. Additionally, knowing which driver mutations occur early is appealing from a clinical standpoint: the affected genes can be targeted for early treatment or used as markers for preventative care.

Previous approaches for determining the temporal ordering of mutations have either involved tracking a single patient over time [Frumkin et al., 2008, Nishizaki et al., 1997, Sasatomi et al., 2002], taking multiple samples from a single tumor [Campbell et al., 2008, Gerlinger et al., 2012, Navin and Hicks, 2010, Siegmund et al., 2009], or pooling data from several patients at stages of cancer for a cross-sectional approach [Attolini et al., 2010, Fearon et al., 1990, Desper et al., 2000, Simon et al., 2000, Newton, 2012, Beerenwinkel et al., 2005, 2006, Rahnenführer et al., 2005, Gerstung et al., 2009, Hjelm et al., 2006, Liu et al., 2009, Bilke et al., 2005].

The cross-sectional approach is the most common because it is the most feasible: identifying patients with early-stage cancer is not always easy, and following a single patient over time can be impractical; instead, it is more common to have samples of several different patients, which collectively represent different stages in cancer. The methodology for estimating a common temporal ordering based on this data type has been developed extensively since the first implementations by Vogelstein et al. [1988], Fearon et al. [1990], and Kinzler and Vogelstein [1996]. These earliest implementations sought a single, linear path to tumor progression. The next major development was the oncogenetic tree model of Desper et al. [2000], which cast the problem in a probabilistic framework and allowed for greater flexibility. These models were developed to describe the sequence of alterations in tumorigenesis without forcing tumors to fall into mutually exclusive categories, which occurs when estimating a common linear progression path. Extensions and generalizations of these models appear in Simon et al. [2000], Newton and Lee [2000], Beerenwinkel et al. [2005], Gerstung et al. [2009], Liu et al. [2009]. Attolini et al. [2010] takes a different approach with cross-sectional data: their algorithm RESIC models the accumulation of abnormalities in tumors via a probabilistic model of cellular growth and the fitness of mutations.

Cross-sectional approaches have the drawback that the mutation history cannot be directly observed, only inferred. Since pathways with multiple orderings can result in the same observed frequencies by stage, these inferred histories can be incorrect. In fact, Sprouffske et al. [2011] performed a simulation study to investigate this problem: they looked at the

reliability of tumor progression estimates based on cross-sectional data and found that the variability of tumor progression pathways produced misleading results.

An additional layer of nuance is that a gene's role can vary based on cancer type: for example, in breast, ovarian, and esophageal cancers, TP53 mutations have generally been thought to be early events [Bashashati et al., 2013, Shah et al., 2012, Weaver et al., 2014], but the opposite situation is true for clear cell renal cell carcinoma McGranahan and Swanton [2015]. Even within a single cancer type, research continues to reveal that there is a great deal of diversity in tumor progression pathways [de Bruin et al., 2014], underscoring a need for more precise or patient-specific methods.

The work presented here was the first to leverage the richness of genomic sequencing data to reconstruct the history of certain chromosomal aberrations from a tumor sample at a single time point. Specifically, it was the first to produce a patient-specific view of tumor evolution without requiring longitudinal data. Our method was first applied for the limited case of copy-neutral loss-of-heterozygosity (CNLOH) in Durinck et al. [2011] to squamous cell carcinoma samples to determine the timing of a loss-of-function mutation on TP53. Greenman et al. [2012] and Purdom et al. [2013] generalized a version of the method to chromosomal gains, and in Nik-Zainal et al. [2012] applied the technique to 21 breast cancer samples.

2.1.1 Advances since Durinck et al. [2011] and Purdom et al. [2013]

Since the publication of our method in Durinck et al. [2011] and Purdom et al. [2013], research on tumor evolution and the related subject of intra-tumor heterogeneity has grown. Intra-tumor heterogeneity refers to the presence of multiple cell populations with the tumor. At the time that Durinck et al. [2011] was published, one prevailing belief was that tumors were comprised primarily of one cell population, called the dominant clone, which had growth advantages over other lesser clonal populations. Our work in Durinck et al. [2011] attempts to order the mutations within the dominant clone using only information from the dominant clone itself. However, shortly after the publication of our work, research interest grew in using the subclonal populations to help infer the order in which the dominant clone had acquired mutations, since the subclonal populations were believed to have shared some of the same evolutionary path as the dominant clone. Additionally, researchers sought to characterize the degree of clonality present in tumours as well as whether the characteristics of the dominant clone were actually predictive of the properties of heterogeneous tumors [Marusyk et al., 2012]. An overview of heterogeneity in cancer genomes can be found in Yates and Campbell [2012]; a more recent review of heterogeneity in tumor evolution can be found in Alizadeh et al. [2015].

Methods for studying tumor evolution accounting for intra-tumour heterogeneity have been developed for three main data types [Davis and Navin, 2016]: first, there is multi-region sequencing, where multiple samples are taken from several spatially distinct regions

of a single tumour [de Bruin et al., 2014, Schwarz et al., 2014, 2015]. *Bitphylogeny* by Yuan et al. [2015] provides a method for reconstructing the phylogenetic tree from these kinds of studies, but can also accommodate other data types. The second method of inquiry is deep-sequencing, which involves sequencing a tumour sample at extremely high depth and computationally deconvoluting clonal populations by clustering mutation allele frequencies [Ding et al., 2012, Egan et al., 2012, Shah et al., 2012, Schuh et al., 2012, Beà et al., 2013, Landau et al., 2013, Bolli et al., 2014]. A well-known example of a computational method for analyzing this data type is *PyClone* [Roth et al., 2014], which uses a Bayesian clustering method to group deeply sequenced mutations.

The most direct evidence of tumor heterogeneity comes from the third data type, single-cell sequencing, in which each sample is comprised solely of DNA from a single cell, as the name suggests. This stands in contrast to the conventional sequencing employed in multi-region sequencing and deep-sequencing studies, where each sample is complex mixtures of different tumor cells. However, because the source of DNA is limited to a single cell, single-cell sequencing has high error rates and other limitations on what markers and genes can be queried [Qiao et al., 2014]. Presently, conventional sequencing still remains the most accessible and common data type for experiments. That said, much of the work on inferring tumor evolutionary history from cross-sectional data has been readily adapted for data from single-cell sequencing experiments: instead of looking across several patients for common mutational patterns, these models now yield the most probable and parsimonious tumor mutational history by analyzing the shared mutations among cells representing a tumor's different clonal populations. For example, *SCITE* (Single Cell Inference of Tumor Evolution) [Davis and Navin, 2016] and *OncoNEM* [Ross and Markowetz, 2016] are two methods for analyzing single-cell sequencing data from multiple cells within a tumor to infer the most likely evolutionary path, and are closely related to the phylogenetic or oncogenetic tree models mentioned earlier.

Finally, we mention *SubcloneSeeker*, developed by [Qiao et al., 2014], as an example of a method that reconstructs subclone structures and evolutionary histories through an integrative analysis of bulk somatic mutation data; it is distinctive in that it permits many types of somatic variant data, e.g., SNVs, copy number variation from sequencing or microarray.

Again, our work focuses on the narrow case of ranking mutations within a dominant clone. With the exception of the deep-sequencing methods, all of the methods mentioned above require multiple samples from a single patient. Our method is thus most contextually similar to deep-sequencing approaches in that it attempts to reconstruct an evolutionary history through deeply mining data from a single sample composed of a mixture of tumor cells. However, rather than looking to subclonal mutation frequencies to piece together the tumor's evolutionary past, we examine the copy number variations in the dominant clone to produce a putative timeline for chromosomal aberrations. As a result, the granularity of deep-sequencing approaches may be at the point mutation or gene level, whereas we only reconstruct large-scale chromosomal aberrations.

In what follows, we will begin by explaining the experimental data types used in the

full analysis of Durinck et al. [2011]. We will explain sequencing with greater detail since our method is based on modeling data from sequencing. While our method can be applied generally to CNLOH and chromosomal gains in principle, we will explain analytically and practically which chromosomal aberrations cannot be timed based on point mutation data. We then restrict our focus to the timing of a subset of events, and compare the accuracy of three methods based on the simulations of Purdom et al. [2013]: our method, the method of Greenman et al. [2012], and a Bayesian variant of our method. Finally, we review the original application of the method in Durinck et al. [2011] to squamous cell carcinoma.

2.2 Experimental data

In this section, we provide a brief overview of the experimental methods or techniques in cancer genomics used to study chromosomal aberrations and somatic point mutations, the two types of events that are the focus of this work. Before we do that, we review terminology used to discuss chromosomal aberrations and point mutations.

2.2.0.1 Preliminaries

Chromosomal aberrations refer to large-scale changes to regions of a chromosome, e.g. the deletion or duplication of several megabases at a time. These events often result in a change to the copy number of a region, or the number of copies of the region present. For example, if the entire p-arm of chromosome 6 has been duplicated twice, there will be three copies of this region. The copy number of the 6p arm would therefore be three. Copy number is always a nonnegative integer.

At the single nucleotide level, a variant refers to a position of the genome where one or both alleles do not match those of the reference human genome. Some variation occurs naturally in the population and is not a result of oncogenetic processes; these variants are referred to as single nucleotide polymorphisms (SNPs). When one allele does not match the reference, the variant is heterozygous, since it present on only one of two copies. When both alleles do not match the reference, the variant is homozygous. The same terminology also works for somatic single nucleotide changes, which are commonly known as point mutations or simply mutations for short, if there is no ambiguity. However, because chromosomal aberrations can interact with variation at the single nucleotide level, we will introduce allele frequency as a better way to describe the number of copies with the variant.

The allele frequency of a variant is the number of copies with the variant allele divided by the total number of copies of that locus. For example, the allele frequency of a heterozygous SNP is 0.5 because there are two copies of region the SNP appears on, but only one of the copies has the variant allele at the SNP location. In a duplication event which results in four total copies of a region, five allele frequencies for a variant are possible: trivially, an allele frequency of 0 is possible, which means that the variant does not appear; the variant could be present on 1 to 5 total copies. In the final case where it is present on all 5 copies, we say

that the variant is homozygous. More generally, in a region with S copies, the possible set of allele frequencies are always $\{0, 1/S, 2/S, \ldots, S/S\}$.

In a given region, somatic mutations are assumed to occur at random on any of the copies. Therefore, these somatic mutations will have allele frequency $1/S$ without additional changes induced by a chromosomal aberration. Chromosomal aberrations interact with point mutations by changing the number of copies that the variant appears on: for example, if the copy harboring the mutation were duplicated twice, then its allele frequency would change from $1/S$ to $3/(S+2)$, since the variant would now be present on three copies out of a total $S+2$.

2.2.1 Cytogenic techniques

Several experimental techniques are used for identifying chromosomal regions of somatic gain or loss. Comparative genomic hybridization (CGH) is one method, but the original technique involved hybridizing DNA to reference metaphase chromosomes [Kallioniemi et al., 1992] and the coarse resolution of the resultant data limited its usefulness [Shinawi and Cheung, 2008]. Since the invention of microarray technology, two techniques have emerged that offer finer resolution and are still widely used today: array CGH for copy number analysis of tumors; and single nucleotide polymorphism (SNP) arrays, which can identify allelic imbalance in addition to copy number.

2.2.1.1 CGH and SNP array

In array CGH, equal amounts of DNA extracted from the tumor and the reference (normal) sample are dyed with fluorescence of two complementary colors (usually red and green). The DNA is then competitively hybridized to a set of location-specific probes on a microarray. The probe intensities of each color are measured as the quantitative output from these experiments: more precisely, for the tumor and reference dye colors respectively, we have $\theta_{T,j}$ and $\theta_{R,j}$–nonnegative, continuous measurements of light intensity at probe locus j. The log ratio of the two, $\log \frac{\theta_{T,j}}{\theta_{R,j}}$ is proportional to the ratio of the copy numbers for tumor and reference and is commonly referred to as $\log R$. In practice, the intensity measurements are normalized before the log ratio is taken–statistical methods for processing the raw data from these arrays are beyond the scope of this discussion, and we refer the reader to Carvalho et al. [2007] for more detail on this topic. The underlying true copy numbers are integer-valued, therefore the ratio of tumor to reference copy number is theoretically discrete as well. However, the estimated log ratio is a noisy surrogate for the truth and is continuous and nonnegative. Intuitively, the ratio is close to zero when the tumor has the same copy number as the reference; and greater than one when there is a gain in the tumor. Because copy number variation also occurs naturally in the genome, cancer studies use neighboring normal tissue as the reference sample when possible to capture only somatic copy number alterations [Redon et al., 2006].

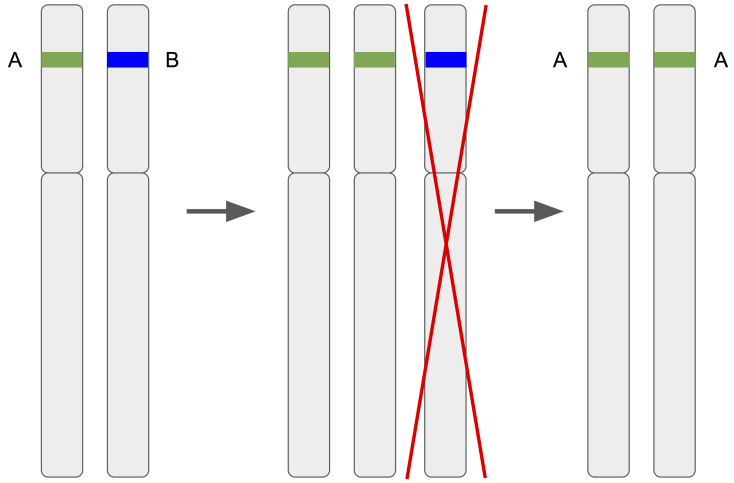
As the name suggests, SNP arrays have probes that can interrogate up to one million SNPs, i.e. sites of biallelic variation, depending on the array; there are two probes per SNP site–one per allele, so that genotype can be obtained in addition to copy number. SNP arrays are not a comparative hybridization technique, so determining somatic changes requires running both tumor and normal samples to array and normalizing the data from the tumor against the normal. As a result, in a paired tumor/normal experiment, we have for each probe locus j four intensity measurements – $\theta_{T,j,A}$, $\theta_{T,j,B}$, $\theta_{R,j,A}$, and $\theta_{R,j,B}$ – corresponding to two pairs of measurements for the SNP alleles A and B in the tumor and normal sample, respectively. As before, the intensities are usually normalized before being used, and $\log R$ can be calculated as an estimate of the copy number ratio of tumor to normal. In this case, R is the total normalized intensity ratio for tumor vs. normal: $\log R = \log(\frac{\theta_{T,j,A}+\theta_{T,j,B}}{\theta_{R,j,A}+\theta_{R,j,B}})$ [Staaf et al., 2008, Wang et al., 2007]. This $\log R$ is analogous to the $\log R$ from array CGH, and can be seen as a noisy surrogate for the true copy number. As a result, the SNP array data suffers from the same problem as array CGH in that the distribution of $\log R$ is continuous instead of discrete, and therefore requires additional analysis to produce discrete copy number calls.

In addition to copy number information, SNP arrays provide information on the frequency of each allele queried by a probe. We will refer to the alleles generically as A and B, but they actually correspond to the two nucleotides known to occur at that locus. The frequency of allele A is the fraction of copies with the A allele. Without loss of generality, an estimate for the A allele frequency at probe locus j can be obtained by taking $\frac{\theta_{T,j,A}}{\theta_{T,j,A}+\theta_{T,j,B}}$ [Staaf et al., 2008, Wang et al., 2007]. We illustrate the concept with two examples, one for a site with polymorphism, and another for a site without polymorphism. Samples usually will not have variation at all of the sites targeted by a SNP array. A sample with a polymorphism at locus j and no other chromosomal abnormalities in the region will be diploid with true alleles AB and true A and B allele frequencies of 0.5, since each allele appears on only one of the two copies. If the sample were to gain a copy of the chromosome with the A allele, then the true alleles would become AAB, resulting the A and B allele frequencies $\frac{2}{3}$ and $\frac{1}{3}$, respectively. Without loss of generality, both the normal and the tumor will have only the A allele in a location without polymorphism, resulting in an A allele frequencies of 1 and B allele frequency of 0.

Like other estimates based on probe intensity, allele frequency estimates are continuous on $[0, 1]$ although their true values are discrete. Additionally, without family data, it is unknown which allele is from the maternal or paternal copy; thus, in analyzing SNP data, the A and B alleles are exchangeable and, in regions of imbalance, the allele with the higher frequency is called the "major" allele while the other is called the "minor" allele. For the same reason, in exploratory plots of B allele frequency (BAF) by location, the designation of A or B allele is arbitrary [Zhang, 2010].

A key advantage to having allele frequency measurements is that they make it possible to detect regions of copy-neutral loss of heterozygosity (CNLOH). In this kind of aberration, the maternal or paternal copy is duplicated with simultaneous loss of the other copy, as

Figure 2.1: A diagram of a copy-neutral loss of heterozygosity (CNLOH) event affecting the entire chromosome. A SNP is shown in the p-arm of a generic chromosome, with allele A marked in green and B in blue. The chromosome copy with allele A is duplicated, while the copy with B is simultaneously deleted. This results in two copies of the allele A in the final result, with no net change to the copy number.



illustrated in Figure 2.1. As a result, the tumor carries two copies of the maternal or paternal copy rather than one of each. Several studies have suggested the importance of these events in cancer after identifying recurrent CNLOH involving known tumor suppressor genes [Purdie et al., 2007, Ross et al., 2007, Heinrichs and Look, 2007, Mao et al., 2007]. This event cannot be detected by CGH array because both tumor and normal still have the same number of copies–hence the designation CNLOH to distinguish it from LOH caused by deletion of a copy. Although CNLOH regions have diploid copy number, any polymorphic sites in the normal would appear as homozygous in the tumor as a result of the duplication, as shown in Figure 2.1. These events can therefore by identified by looking for regions with $\log R$ indicating no aberration in copy number but BAF corresponding to allele frequencies of 1 and 0.

2.2.1.2 Challenges in the analysis of CGH or SNP array data

Segmentation algorithms have been developed for processing SNP or CGH array data to identify breakpoints of regions of copy number or allelic aberration. The probe intensities are typically modeled as having a Gaussian distribution where the mean of the distribution depends on the region's true copy number. The problem can then be cast as one of changepoint detection, where the challenge is to determine positions where the mean of the distribution changes, i.e. changepoints. The most well-known approach is circular binary segmentation, which was first applied to the context of array data in [Olshen et al., 2004].

Many algorithms are different variations on this method: for a high-level overview, we refer the reader to Zhang [2010].

Many factors affect the accuracy of copy number and allele frequency estimated from array data: purity of the tumor DNA sample (i.e., the degree to which the sample contains only tumor cells of one type), the laboratory preparation, and biases inherent to the experimental platform itself (i.e., manufacturing differences) [Pinkel and Albertson, 2005]. As mentioned earlier, normalization of the raw intensities is usually performed as a pre-processing step to reduce unwanted variation due to the platform and laboratory preparation.

Heterogeneity in the tumor sample arises as a result of contamination with neighboring normal cells or the presence of subclonal cells that share some but not all of the mutations as the dominant tumor clone. If a region's copy number varies over the cells that comprise the sample, its log intensity measurement will reflect the average copy number across all cells. As a result, if the contaminated cells do not possess the same abnormality as the dominant clone, the effect on the region is dilution of the copy number toward 2 and the allele frequency toward 0.5. For example, if half of the cells have diploid copy number in a region but the other half has lost a copy from an LOH event, the true total copy number is $2(0.5)+1(0.5) = 1.5$. The overall effect on the copy number and allele frequency estimates are a narrower range of values, which makes it harder for segmentation algorithms to determine breakpoints, and even harder to estimate the true integer-valued copy number or discrete allele frequency.

Heterogeneity affects the entire sample and is impossible to correct without additional information or manual inspection to determine the degree of contamination. In many cases, contamination by normal cells is usually seen as unwanted variation, whereas the presence of subclonal populations in the tumor is not. It can be difficult to tease apart whether the signal dilution came from a genuine tumor subclones or the inclusion of neighboring normal tissue depending on the degree of subclonality or contamination, although it is theoretically possible to use the differential behavior of normal and subclonal cells–the effect of normal would be consistent across the entire genome, but the subclonal cells may affect only certain regions where they differ from tumor [Pinkel and Albertson, 2005, Zhang, 2010]. In practice, subtle changes like those implied by subclonal populations are difficult to distinguish from natural measurement variation. With heavy amounts of contamination, it is often necessary to combine array CGH or SNP array data with other sources of information (e.g., sequencing) to make definitive calls.

To illustrate the effect of tumor sample contamination, Figure 2.2 shows a chromosome with an LOH region after the centromere at 40 megabases and heavy signal dilution from contamination. This is noticeable from how the LOH is barely perceptible in the top plot with copy number estimates from SNP array. The fourth plot shows the SNP B allele frequency of tumor normalized against the normal sample, from which it is clear that the region after the centromere at 40 has allelic imbalance. Neither algorithms nor manual inspection can determine confidently what aberration had taken place based on these two plots from the SNP array data alone. However, a plot estimated mutation allele frequencies from sequencing shows the presence of only one allele frequency, a trait unique to regions of LOH. One can

see that the allele frequencies in this plot are also affected by normal contamination, which we explain further in the next section on sequencing data, Section 2.2.2.

Because integer-valued copy number must be inferred from array and these inferences are made more difficult by the presence of any contamination, researchers may seek alternative means of validating their findings. One possible means of validation is to use other cytogenic methods like fluorescence *in situ* hybridization (FISH). There are multiple techniques based around FISH, but they all involve dyeing targeted regions of a chromosome with different colors of fluorescence–from there, a technician views the chromosomes under a microscope and determines the copy number alteration that has taken place by visual inspection. By counting, they can also estimate the proportion of tumor cells possessing that alteration. Alternatively, 24-color FISH can produce a color-coded map of the genome for detecting chromosomal rearrangements, some of which cannot be caught by array [Bishop, 2010]. One example is a balanced translocation, which occurs when regions of two different chromosomes are swapped; this event does not change the copy number or produce allelic imbalance, so it would not be detected by the array methods, but is easily seen with imaging. A schematic representation of array CGH and spectral karyotyping (which produces similar output to FISH) taken from Bishop [2010] is shown in figure 2.3, and shows the difference in output the two kinds of experiments more clearly. The primary drawback to using FISH to validate array findings is how labor intensive the process is, as it requires manual inspection and is intended for interrogating specific, small regions of the genome at a time [Bishop, 2010].

2.2.2 Whole genome sequencing

2.2.2.1 Next generation sequencing

Sequencing produces data at the finest resolution possible: allelic information for each base along the genome. After The Human Genome Project kicked off in 1990, it took a little over a decade to assemble the first nearly complete sequence of the human genome [Venter et al., 2001]. Since that breakthrough nearly fifteen years ago, great leaps have been made in sequencing technology to more efficiently generate high quality data at lower costs–namely, the advent of massively parallel DNA sequencing or next generation sequencing has reduced the cost of sequencing experiments by over two orders of magnitude since 2005 [Shendure and Ji, 2008]. This push has further been driven by the desire for individual genome sequencing for use in personalized medicine [Ginsburg and Willard, 2009, Metzker, 2010]. Scientists have found widespread application for next generation sequencing, from investigating protein-DNA interactions, studying gene expression, genotyping SNPs, discovering copy number aberrations, to identifying somatic mutations in cancer. In this section, we provide an overview of next generation sequencing as it pertains to understanding somatic changes in cancer, the focus of our method. At a high-level, we will review the sequencing process and methods for analyzing the raw data output–alignment to the reference, mutation calling, and even copy number analysis.

The name "next generation sequencing" distinguishes it from conventional sequencing

Figure 2.2: The first four plots show processed SNP array estimates vs. position on chromosome 10 from matched tumor and normal samples for patient V07: the first contains copy number estimates; second, B allele frequency for the normal sample; third, B allele frequency for the tumor sample; and finally the allele frequency for tumor normalized against the normal sample to make clear regions of allelic imbalance with respect to the normal. Red color is used to highlight SNPs homozygous in the normal. The final plot shows allele frequencies for SNPs (slate blue) and mutations (orange) obtained from sequencing data.

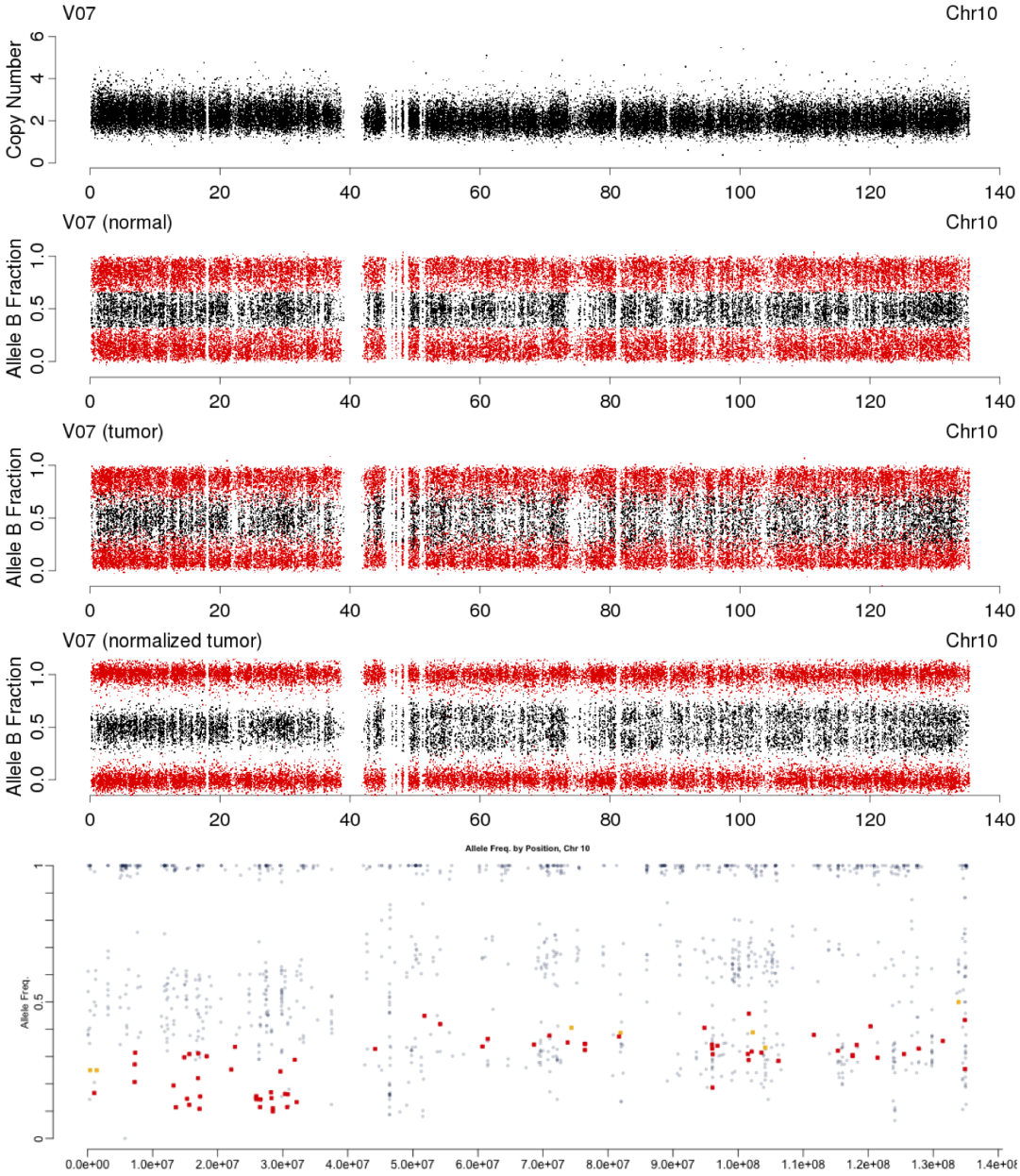
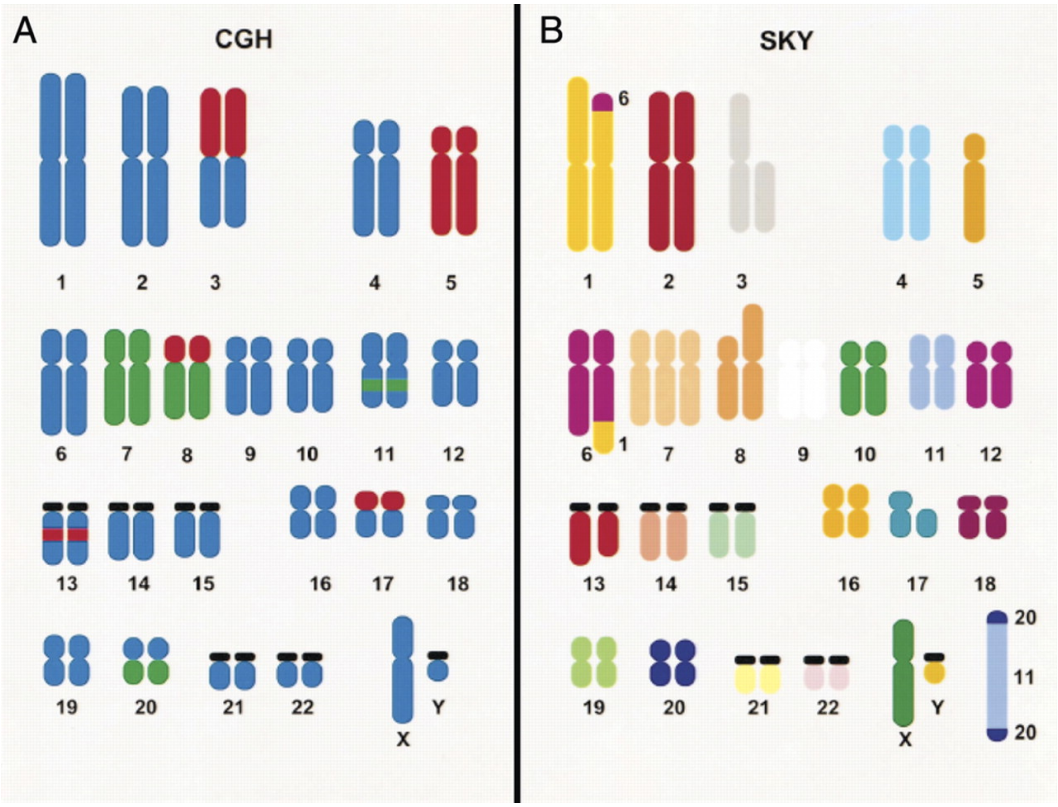


Figure 2.3: A schematic representation of the output from array CGH and FISH experiments. In FISH, chromosomes are dyed different colors, making it is possible to detect balanced translocations like the one between the 1p and 6q by looking for color rearrangements. As this does not result in a copy number change, it is not picked up by the CGH experiment. However, the CGH experiment does pick up the deletion on 3p. If a SNP array would be able to detect allelic imbalances and therefore CNLOH, which neither FISH nor CGH can detect. This figure originally appeared as Figure 8 in Bishop [2010].



methods that rely on the biochemistry developed by Frederick Sanger in the 1970's. Both conventional and next generation sequencing workflows require chopping genomic DNA into smaller fragments called reads, amplifying these fragments, and then resolving the resultant bases. The primary distinction between Sanger and next generation sequencing methods is the degree of parallelization: in Sanger sequencing, the bases of each read are resolved using capillary electrophoresis, so the amount of parallelization is limited by the number of independent capillaries that can practically be run simultaneously–typically only in the hundreds. On the other hand, in next generation sequencing, bases are resolved by performing several cycles of imaging or other detection method simultaneously across an array onto which different fragments have been bound and are separated spatially. This means parallelization is only limited by the number of fragments that can be accommodated on the array; in practice, these cyclic-array methods can produce millions of reads in one run

[Shendure et al., 2004, Shendure and Ji, 2008, Metzker, 2010]. Next- or second-generation methods have presently found widespread use in the scientific community, but for completeness we note the recent development of "third generation sequencing" technologies that allow sequencing of single molecules in real time, such as that developed by Pacific Biosciences. This is beyond the scope of this dissertation and we refer the reader to the overview by Schadt et al. [2010] for more details.

The technologies that fall under the umbrella of next-generation sequencing (e.g., 454 pyrosequencing, Illumina Solexa, Applied Biosystems SOLiD, Ion Torrent) differ in the type of PCR performed during library prep and how actual sequencing is done afterward. For instance, on each cycle, the Ion Torrent system identifies bases by detecting changes in pH; whereas for Illumina Solexa sequencers, bases are detected by imaging, with the four nucleotides emitting different colors. There are advantages and disadvantages unique to each platform: for instance, 454 pyrosequencing accommodates longer reads than the other methods mentioned, but because of its sequencing biochemistry, its dominant error type is insertion/deletion of bases in the final sequence, due to its inability to accurately resolve the number of repeated bases in reads with homopolymers, i.e. runs of a single nucleotide such as AAA or GGG [Shendure and Ji, 2008]. On the other hand, while Illumina sequencers do not have an insertion/deletion problem, they have a documented bias against GC rich regions [Nakamura et al., 2011]. A vast body of research focuses on the idiosyncrasies of each platform and how they should be handled in different applications; this is beyond the scope of this dissertation, and we refer the reader to Shendure et al. [2004], Shendure and Ji [2008] for specifics on the differences between platforms.

Experimenters can choose reads of two different types, and these types impact the kind of information that can be learned from the experiment: 1) single-end reads or 2) paired-end or mate-pair reads. In single-end sequencing, each fragment of DNA produced during library preparation corresponds to a read, and these are sequenced in the 3' to 5' direction. In paired-end sequencing, fragments are larger and two reads are produced from sequencing each end; the middle portion is not sequenced. This yields as final output one read in the 3' to 5' and another in the 5' to 3' direction for each DNA fragment, where the distance between the reads is approximately known from being fixed beforehand in the design of the experiment. Mate-pair sequencing results in the same output as paired-end sequencing, but the gap distance between the reads is typically much larger (e.g., on the order of kilobases rather than hundreds of bases). The naming difference is due to different underlying library preparation processes: whereas paired-end reads are produced by sequencing both ends of a single fragment, the longer gap distance between mate-pair reads requires fragments to be circularized before being sequenced. What paired-end or mate-pair reads offer above single-end reads is the ability to detect structural variants and better alignment ability in repetitive regions. As an example of the former, if one end maps uniquely to a region in chromosome 9 and the other maps uniquely to a region in chromosome 20, then this indicates that a rearrangement of those two chromosomes has occurred. Likewise, having one end that maps uniquely to a region of a chromosome can resolve the position of a second end that falls in a highly repetitive region, since the known gap distance between them limits the places the

second end could map to.

2.2.2.2 Processing pipeline for studying mutations with next generation sequencing data

A sequencing experiment yields as raw output a file with base calls for every read produced and usually additional metadata for each read, like a score for the quality of each base call. These reads must be aligned either *de novo* or against a known reference, the latter being the most common choice in situations where a reference exists–having a known reference means that the experiment can be designed to produce shorter reads at a high throughput, which is generally cheaper and easier to do than obtaining fewer but longer reads. Several algorithms have been developed for the alignment of next-generation sequencing reads to a reference; to name a few: *Bowtie* [Langmead et al., 2009], *BWA* [Li and Durbin, 2009], *MAQ* [Li et al., 2008a], *Novoalign* (http://novoalign.com), and *SOAP* [Li et al., 2008b]. Yet more exist for other specialized purposes. Alignment continues to be an active area of research as new sequencing technology is constantly being developed and consequently different kinds of sequencing data are produced.

Generally, for studying mutations and chromosomal aberrations in cancer, longer reads like those offered by 454 pyrosequencing are not required at the outset because reads can be mapped against a known human reference, as opposed to, say, the use case of performing *de novo* assembly for mapping the genome of a new organism, for which longer reads provide better alignment confidence. Paired-end or mate-pair sequencing of the entire genome is frequently done in cancer studies as a way to detect somatic chromosomal rearrangements, but it is also common to investigate any breakpoints found via single-end sequencing further with targeted sequencing of select regions, e.g. Sanger sequencing; the same is true for validating interesting mutations initially detected by single-end sequencing. In the case of somatic mutations, researchers are often interested primarily in mutations that occur in coding regions of the genome, a.k.a the exome; mutations in these regions can directly alter the proteins that are encoded and thereby disrupt proper gene function or expression. Therefore, some studies, including the one in which we applied our method [Durinck et al., 2011], will run targeted sequencing of just the exome, which comprises only 2% of the genome. Restricting the region sequenced to the exome offers the benefits of lower cost and greater coverage of the targeted regions.

2.2.2.3 Read depth, estimated allele frequency, and an overview of the effect of normal contamination

From this point, we assume that a sequencing experiment has been run and the reads have been aligned to the human reference. Then, for each base along the regions sequenced, we have its *read depth*, the total number of reads that overlap that base. If the cancer is not highly mutated, then most bases in the reads match those of the reference. However, in a position with a somatic mutation or a SNP, reads will show bases different from the

reference. Typically, unless there are sequencing errors or contamination with a third allele (e.g., from a subclonal population), the majority of reads for a location will show at most two nucleotides–those of the reference and the variant, if any. If m_i is the read depth at location i and X_i is the number of reads showing the variant, then the variant read allele frequency is defined as X_i/m_i. The reference allele frequency is defined analogously. These can be seen as estimates of the true frequencies of the alleles in that location. As mentioned in Section 2.2.1 on cytogenic techniques, true allele frequencies are discrete. Precisely, if a region has S copies, the only possible true variant allele frequencies are $0/S, 1/S, \ldots, S/S$. However, allele frequencies from reads are more accurately modeled as continuous-valued when the average read coverage for each base is sufficiently high. Note that although $0/S$ is a valid allele frequency for a mutation, we are unable to identify mutations with this allele frequency using sequencing data since a $0/S$ allele frequency implies that both alleles match the reference and thus would be indistinguishable from an unmutated position.

Several algorithms have been developed to analyze sequencing read depths and allele frequencies from matched tumor and normal samples and produce a list of potential somatic and non-somatic variants. The resultant list provides information on each allele and the reads supporting them. Roughly speaking, to obtain somatic mutations, one can disregard variants that correspond to known SNPs catalogued in COSMIC, and also select only those locations where an allele is homozygous in the normal but not in the tumor. The latter criterion is at the center of methodological research on variant calling: probabilistic methods, for instance, estimate the probability that a locus is mutated and those locations with probabilities above some threshold are labeled mutations. As with all of the other workflows described for genome sequencing, there is a great deal of nuance in implementation, the details of which are beyond the scope of this dissertation. We cite the papers describing two commonly used variant callers for additional references on the topic: the *MuTect* algorithm offered in the Broad Institute's *Genome Analysis Tool Kit (GATK)* [DePristo et al., 2011, Auwera et al., 2013] and the variant-caller offered in the *SAMtools* toolkit [Li, 2011].

Tumor samples extracted from tissue rather than a pure source of DNA such as a cell line usually contain noticeable contamination from normal DNA. These samples may also contain contamination with subclonal cells, but the effect of subclonal contamination is complex and can be difficult to detect if most of the majority of mutations in the subclones also appear in the dominant clone and/or the subclonal cells comprise only a small percentage of the sample. In loci where the subclone differs from the dominant clone, either of these situations would likely result in the loci being filtered out in the mutation-calling step as sequencing error. Positions where the tumor is mutated but the subclone is not cannot be distinguished from natural variation in reference allele coverage or normal contamination. The same is true of a position where both tumor and subclone share the same mutation.

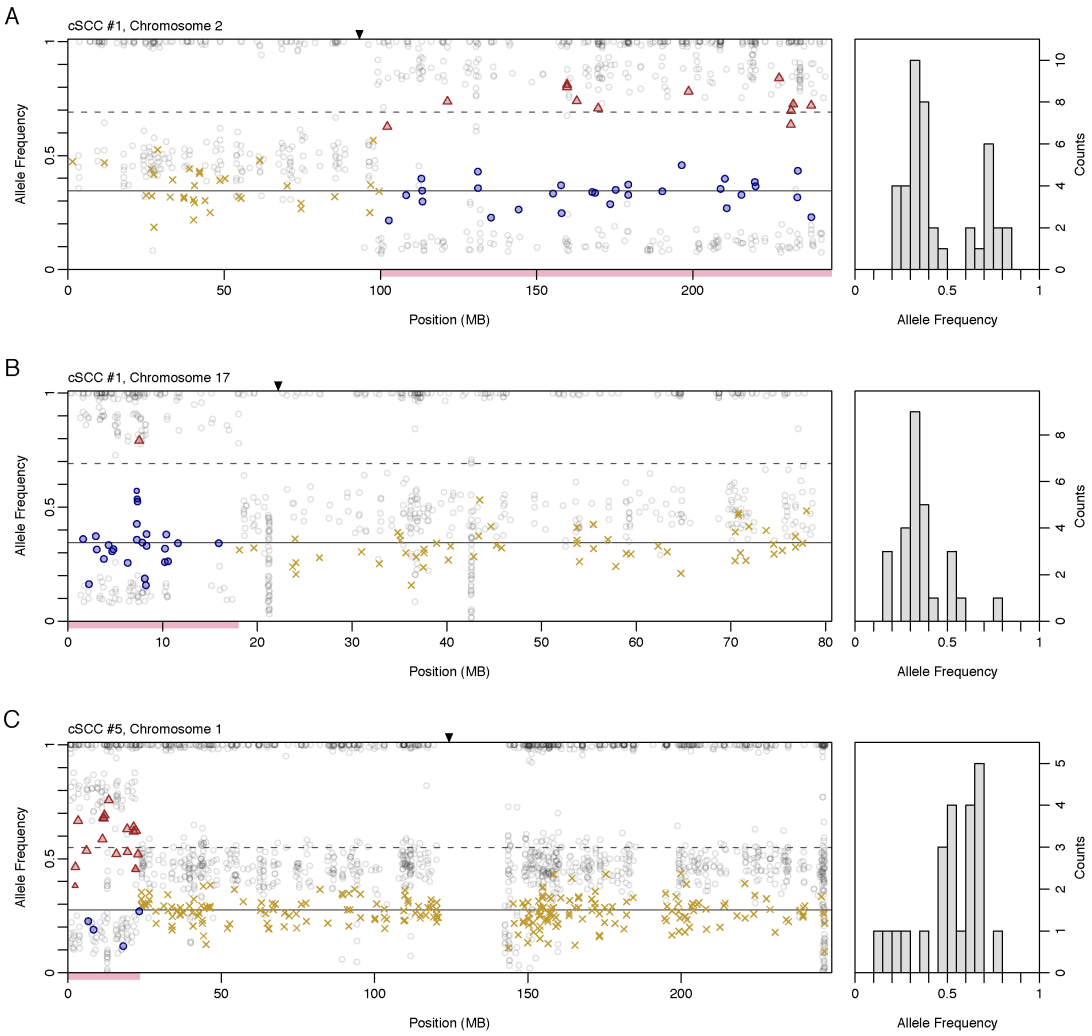
Genuine sequencing error incorrectly swaps a reference and variant allele, and in general affects only a small number of mutations (1-2% of mutations). This particular error becomes a problem only in situations where a single read base swap could result in classification to a different allele frequency. This occurs when both the overall read coverage and the true variant allele frequency are low. Most mutation calling algorithms set a minimum threshold

on read coverage for a position to be called a mutation, which essentially filters out these positions. In addition to genuine sequencing error, some situations, like the subclone scenario alluded to in the previous paragraph, can result in outcomes that are difficult to distinguish from the other kind of sequencing error that results in a small number of reads from a third allele. These reads are usually filtered out even earlier during the alignment step. As a result, positions with sequencing error or situations that produce similar outcomes to sequencing error largely do not make it into the final set for analysis.

On the other hand, the effect of normal contamination on sequencing data affects every mutation in the same way, which makes it easier to model and quantify. Specifically, for positions where the alleles in the tumor sample match the normal such as SNPs or unmutated bases, the resultant allele frequencies are unbiased estimates of the true allele frequency. For mutated positions in the tumor, the normal sample contributes more reference alleles than would be the case if the sample were pure. Because the estimated allele frequency is the proportion of variant alleles out of sequenced alleles, this increases the denominator and pushes the value toward 0.

This is illustrated further in Figure 2.4, which shows the estimated allele frequencies for variants on three chromosomes plotted by position. It is useful to think about the effect of normal contamination on SNPs and mutations separately because, unlike the latter, the former is present in the normal cells. In diploid regions, the estimated allele frequencies of SNPs are centered around 0.5, the true allele frequency, which agrees with how these positions are the same in both the tumor and the normal. However, the estimated allele frequencies for mutations are centered at a value less than 0.5, reflecting how the normal sample contributes additional reads from the reference allele. In regions where the tumor sample has CNLOH, the effect of the normal on SNPs and mutations is more nuanced, and it is useful to break down the four possibilities for reads for each variant: the variant allele from tumor, variant allele from normal, reference allele from tumor, and reference allele from normal. SNPs in tumor CNLOH regions are still heterozygous in the normal but in the tumor, they are now either homozygous in the variant or the reference. The only source of reference for a SNP homozygous in the variant in the tumor is from one chromosome copy in the normal cells. The converse is true for a SNP homozygous in the reference in the tumor. As the plot shows, a SNP homozygous in the variant in tumor tends to have higher estimated allele frequency than mutations homozygous in the variant in the CNLOH region: this is because homozygous mutations are "contaminated" by twice as many reference reads as homozygous SNPs, since both copies of the normal harbor the reference.

Figure 2.4: Three plots showing estimated allele frequency by position. Regions of the x-axis highlighted in pink indicate CNLOH. Mutations are indicated with colored points, whereas SNPs appear in the background in semi-transparent grey. Specifically, mutations in CNLOH regions are colored red or blue depending on whether they are likely to be homozygous or heterozygous, respectively. Mutations in the neighboring diploid regions are colored yellow. The first two plots show data from the same sample, but the third comes from a different sample. On the right of each scatterplot is a histogram of the allele frequencies for the CNLOH region. This figure originally appeared as Supplementary Figure 2 in Durinck et al. [2011].



2.3 Method for estimating temporal ordering

We assume that the tumors under study are comprised of a dominant clone and normal contamination, so that there is negligible contamination with divergent subclonal cells. Our model will also still work well for a tumor sample that possesses a small percentage of subclonal cells that share the vast majority of mutations with the dominant clone.

We take as our starting point a list of somatic mutations obtained from processing matched tumor and normal sequencing data through an alignment and mutation-calling algorithm. As such, we assume that we have counts for both the reference and variant allele for each mutation and that those counts are correct. In other words, we do not account for sequencing error in our model. We will continue to use the notation of Section 2.2.2.3 to describe the sequencing data: we will let X_i represent the reads of the variant allele, m_i the total read depth, and $P_i = X_i/m_i$.

We also assume that we are able to determine regions of the genome that either have chromosomal copy-number changes or copy-neutral changes resulting in allelic imbalance. As mentioned previously, these calls can be obtained from either using segmentation algorithms or manually inspecting both sequencing and array data.

We assume that each chromosomal aberration occurs as a series of K events to produce the final copy number S. These events split the lifetime of the tumor into $K + 1$ "stages": the 0th stage, the time before the first event; the 1st stage, the time between the first and second event; and so forth, ending with the Kth stage, the time after the Kth event. We further assume that point mutations accumulate at random on the genome at a constant rate, so that the proportion of mutations originating at each stage is a direct measure of the fraction of time spent in each stage.

In the following section, we provide a model for the relationship between the time spent in each stage and the number of mutations having each allele frequency. In this discussion, we assume that the tumor sample is free from contamination by normal cells, so that the allele frequencies are simply fractions of the number of copies present. We also assume that the true allele frequencies of each mutation can be determined, but in practice this must be estimated from the sequencing data. We account for both sample impurity and sequencing variability in the model later on in Section 2.3.4.

2.3.1 A model for the ideal setting: pure tumor sample and known allele frequencies

Let the components of the vector $\pi = (\pi_0, \ldots, \pi_K)$ give the probability that a mutation in that region originates in each stage. The goal of our method is to estimate π from the observed allele frequencies of the mutations.

In a particular region with a K chromosomal aberration resulting in S copies, we observe N point mutations, which we assume have only been mutated once in the history of the tumor so that a position is not being mutated repeatedly over time. We denote their true allele frequencies by p_i, $i = 1, \ldots, N$. As explained in Section 2.2.2.3, mutations where

all copies with the variant allele have been deleted have variant allele frequency 0, so we cannot distinguish them from unmutated positions. Therefore the N observed mutations only include those with true allele frequency $p_i > 0$. For now, to simplify explanations of allele frequency in the model, we will assume that the tumor sample does not contain any normal contamination; but we will adjust our model to accommodate contamination later in Section 2.3.3.2. In this setting, detectable mutations can be present on between 1 to all S final copies, resulting in the set of possible allele frequencies $1/S, \ldots, S/S$. Therefore, $p_i \in \{1/S, \ldots, S/S\}$ for all i in the pure tumor case.

The true mutation allele frequency p_i is completely determined by what stage the mutation occurred in and what copy the mutation was on. Since the mutations occur at random, we can model the number of mutations possessing each allele frequency as a multinomial random variable with the probability parameter $q = (q_1, q_2, \ldots, q_S)^T$, where $q_j = P(p_i = j/S | p_i > 0)$, i.e. the probability that an arbitrary observable mutation in the region has the allele frequency j/S.

If we know the region's event history well enough to identify the number of copies present at each stage of the event, we can write an expression that relates q to π, the parameter of interest. From there, it can be shown that under certain conditions on the event history, we can obtain a plug-in estimator for π in terms of an estimator of q. Precisely, we break down the probability $q_j = P(p_i = j/S | p_i > 0)$ by stage, so that we have

$$\begin{aligned} q_j &= \sum_{k=0}^{K} P(p_i = j/S \text{ and mutation } i \text{ originated in stage k} | p_i > 0) \\ &= \sum_{k=0}^{K} P(p_i = j/S \mid \text{mutation } i \text{ originated in stage } k,\, p_i > 0) \\ &\qquad \times P(\text{mutation } i \text{ originated in stage } k \mid p_i > 0) \end{aligned}$$

Let S_k denote the number of copies present at stage k, and let S_{jk} be the number of these copies that lead to a final true allele frequency of j/S in the tumor. We assume that each of these S_k copies is equally likely to receive a point mutation at stage k, so $P(p_i = j/S \mid \text{mutation } i \text{ originated in stage } k,\, p_i > 0) = S_{jk}/S_k$. The chance that a mutation originates in stage k depends directly on π_k and the number of copies present in that stage - intuitively, a stage with twice as many copies as another would have around twice the number of mutations. An application of Bayes' rule yields

$$P(\text{mutation } i \text{ originated in stage } k | P_i > 0) = \frac{S_k \pi_k}{\sum_{k=0}^{K} S_k \pi_k}.$$

Letting $c_\pi = \sum_{k=0}^{K} S_k \pi_k$ denote the normalizing constant in the expression above and putting these pieces together, we arrive at

$$q_j = \frac{1}{c_\pi} \sum_{k=0}^{K} \left(\frac{S_{jk}}{S_k} \times S_k \pi_k \right) \tag{2.1}$$

$$= \frac{1}{c_\pi} \sum_{k=0}^{K} S_{jk} \pi_k \tag{2.2}$$

This lends itself to a simple matrix representation: let A be the $S \times (K+1)$ matrix where $A_{jk} = S_{jk}$ as defined above, i.e. the number of copies at stage $k \in \{0, \ldots, K\}$ with true allele frequency j/S, $j \in \{1, \ldots, S\}$.

Then, the previous sum can be expressed as the linear relation $q = A\pi/c_\pi$. From this expression, it is evident that if q is known, a plug-in estimator for π is possible only if the matrix A is invertible. Events may result in only a subset of the possible $K+1$ allele frequencies. In this case, the rows corresponding to those allele frequencies will have all zeroes and can be removed from the matrix.

As an example to clarify these terms, consider the first event history in Figure 2.5: the maternal (M) copy is duplicated twice, followed by a duplication of the paternal (P) copy, resulting in three M copies and two P copies for an overall copy number of $S = 5$. The three duplications split the event into $3 + 1 = 4$ stages, corresponding to $k = 0$, the stage before any duplication has occurred; $k = 1$, stage after the first duplication; $k = 2$, stage after the second duplication; and $k = 3$, the stage after the third duplication, a.k.a the final, post-event stage. Since S_k corresponds to the number of copies present at each stage, we have $S_0 = 2$, $S_1 = 3$, $S_2 = 4$, and $S_3 = 5$. Then the normalization constant c_π in the relation $q = A\pi/c_\pi$ becomes $c_\pi = \sum_{k=0}^{K} S_k \pi_k = 2\pi_0 + 3\pi_1 + 4\pi_2 + 5\pi_3$. With five total copies, the allele frequencies $1/5, 2/5, \ldots, 5/5$ are possible. The term S_{13} represents the number of copies in stage 3 that have a final allele frequency of $1/5$. Note that any mutation acquired in stage $k = 3$ has final allele frequency $1/5$, because there are no further duplications. Therefore, $S_{13} = 5$ since there are 5 copies in $k = 3$. As another example, we find the nonzero values of S_{j2}. Mutations in stage $k = 2$ occur before the final duplication of the event, which involves the P copy. A mutation acquired on any of the three M copies would have allele frequency $1/5$ in the final stage because none of the M copies are duplicated. However, a mutation on the P copy would have allele frequency $2/5$ in the final stage because the duplication would make it present in two final copies. Therefore, $S_{22} = 1$ for the single P copy and $S_{12} = 3$ for the three M copies in stage $k = 2$, and no other allele frequencies are possible– mutations acquired in this stage cannot be present on more than two copies because only one duplication event remains after this stage. Using the same reasoning to obtain $A = [S_{jk}]$, we arrive at the matrix shown in Figure 2.5 for this event history. In Figure 2.5, rows for every possible allele frequency were included for completeness, even if the event history does not produce mutations with those allele frequencies (e.g., $4/5$ or $5/5$).

The vector q is not known, but can be estimated from the data. We describe our method for doing so in Section 2.3.4. The following sections delve deeper into various components of the model, leading up to the section on estimating π. They are organized as follows:

First, we highlight practical and theoretical considerations in model identifiability. Section 2.3.2 looks at which event histories are detectable from the data and can be timed. We divide the section into two parts: identifiability of the A matrix and identifiability of π. On the topic of the A matrix, our discussion addresses the questions "What kind of A matrices correspond to events that can be identified with data?" and "What histories result in unique A matrices?" Our discussion on the identifiability of π is centered around identifying event histories that result in invertible A, and therefore have estimable timing vectors. By the end, we restrict our attention to CNLOH events and a specific case of sequential gain, for which we can analytically show results in an identifiable event history.

Then, we turn our attention to addressing sequencing variability in the data: sample impurity and not knowing the allele frequency of the mutations. Sample impurity changes the expected allele frequencies from their pure values of $1/S, 2/S, \ldots, S/S$. First, we model the effect of normal contamination on the allele frequencies and provide an estimator for the degree of normal contamination based on point mutation data in regions without chromosomal aberration. Then, we explain how we correct the pure tumor sample allele frequencies when normal contamination is present.

Finally, in Section 2.3.4, we provide a model for the sequencing data from each mutation, taking into account sample impurity. To handle the fact that the true allele frequencies of mutations and therefore q are not known, we model the true mutation allele frequencies as latent variables in our model. From here, we are able to describe our estimation technique, which we refer to as full maximum likelihood (MLE) to distinguish it from the partial MLE technique of Greenman et al. [2012].

2.3.2 Identifiability

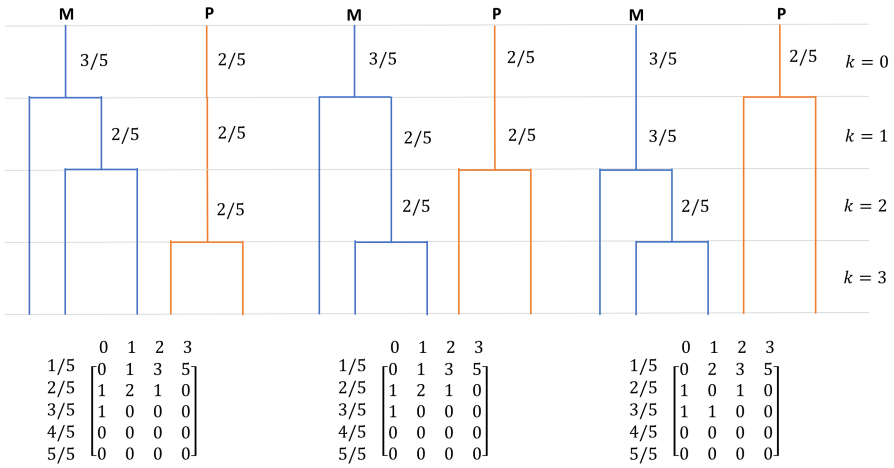
2.3.2.1 Determining the form of A

To write the matrix A, we must know the number of copies in each stage resulting in each allele frequency, which in turn requires knowledge of precisely how each gain occurred.

To illustrate by starting with the simplest case, events with only one stage ($K = 1$) have three possibilities–deletion of a copy, gain of a copy, or a CNLOH–which result in A matrices with different allele frequencies from each other. For example, the allele frequencies $\frac{2}{3}$ and $\frac{1}{3}$ are unique to the single copy gain; $\frac{1}{2}$ is unique to the CNLOH; and the deletion is characterized by having only one allele frequency, 1.

However, in events with more than one stage ($K > 1$), two different event progressions can result in the same A matrix. To illustrate, consider a $K = 3$ stage event that results in three copies of the maternal (M) copy and two of the paternal (P) copy. This requires that the M copy be duplicated twice, and the P copy once. This can happen in three ways, up to symmetry on the M copy: the duplication on the P copy could occur in stage $k = 1$, $k = 2$, or $k = 3$, as illustrated in Figure 2.5. For the two cases where the P duplication does not

Figure 2.5: The three possible event histories for a $K = 3$ stage event resulting in two maternal (M) copies and two paternal (P) copies. Written to the right of each copy in each stage is the final observed allele frequency of a mutation occurring on that copy; nothing is written if the final observed allele frequency is 1/5. Below each event history, the resultant A matrix is given, where the rows and columns have been labeled with the allele frequencies and stages they correspond to, respectively. Rows for every possible allele frequency were included for completeness, even if the event history does not produce mutations with those allele frequencies (e.g., 4/5 or 5/5). For the first two events where the M copy is duplicated first, the A matrices are the same. The A matrix is different for the case where the P copy is duplicated first.



occur first ($k = 1$) the final A matrices are the same and equal to

$$A = \begin{pmatrix} 0 & 1 & 3 & 5 \\ 1 & 2 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

where as in Figure 2.5 the rows correspond to the allele frequencies $\frac{1}{5}, \ldots, \frac{4}{5}, 1$ (the 0 allele frequency has been omitted) and the columns to stages 0, 1, 2, and 3. Rows for 4/5 and 1 were included for completeness although no mutations result in these allele frequencies. This example demonstrates that the A matrix is not unique to the event history.

Another possibility is that two event histories result in the same set of allele frequencies, but at different relative proportions. Mathematically, this means that the two events have A matrices with the same all-zero rows. Then, when comparing among a set of histories of interest, it is possible for the A matrices to be identifiable in the sense that there is a unique matrix for each history, but still present a challenge to practitioners trying to pinpoint which

particular history using the data. The same three-gain example of Figure 2.5 can be used to illustrate a trivial example of this: suppose now that a practitioner is trying to determine whether the stage 1 duplication occurred on the P copy or M copy. Letting the rows and columns indicate the same stages and allele frequencies, the resultant A matrix when the P duplication occurs in stage 1 is given by

$$A = \begin{pmatrix} 0 & 2 & 3 & 5 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

The resultant q vector for a first-stage P duplication is proportional to $(2\pi_1 + 3\pi_2 + 5\pi_3, \pi_0 + \pi_2, \pi_0 + \pi_1, 0, 0)$ up to normalization to make the vector sum to 1. Likewise, for non-first stage P duplication, the resultant q is proportional to $(\pi_1 + 3\pi_2 + 5\pi_3, \pi_0 + 2\pi_1 + \pi_2, \pi_0, 0, 0)$. Setting the components of both q vectors equal, we can see that if no time were spent in stage 1, i.e. $\pi_1 = 0$, both events would produce the same relative proportions of allele frequencies. This is a trivial case because $\pi_1 = 0$ means that the two event histories collapse into a single history where the P duplication and the M duplication occur simultaneously, but it is mathematically possible when $K < S - 1$ to find a set of π vectors that would produce the same q with two different A matrices. Notwithstanding, this result shows that the ability to tell the two events apart rests on observing mutations in the stage after the first duplication. If the tumor spent very little time in this stage (π_1 small), it would be difficult for a practitioner to determine whether the P or M duplication occurred first.

Methods have been developed to use whole-genome sequencing reads spanning breakpoints to help reconstruct more of the event history [Greenman et al., 2012]. Even so, not all regions with gain will have a unique construction. With exome sequencing, which only spans around 2% of the genome, the event histories of many gain regions will not be distinguishable.

2.3.2.2 Identifiability of π: invertibility of A

Even if we can precisely determine the copy changes that occur at each stage in an event, π could still be unidentifiable. Because of the relation $q = A\pi/c_\pi$, π is only identifiable when A is invertible. A minimal requirement for invertibility is that A be square, which means that an event that occurs in K steps must result in $K + 1$ observed allele frequencies. This automatically excludes events with deletions because they result in less than $K + 1$ copies in the final state, so it is impossible to have $K + 1$ observed allele frequencies. As a result, a requirement for identifiability of π is that the event be comprised only of gains.

One exception is the $K = 1$ case of CNLOH, where the deletion is assumed to occur simultaneously with a gain, resulting in two observable allele frequencies. We work out CNLOH case to illustrate how π can be written in terms of q. Without loss of generality, we assume the maternal (M) copy is duplicated, with simultaneous loss of the paternal (P) copy. Any mutations present on the M copy before the duplication appear on two

copies in the final state and therefore have an allele frequency of 1, whereas mutations on the P copy prior to deletion would be unobservable with allele frequency 0. Mutations occurring after the duplication event could occur on either copy, and therefore would have allele frequency 1/2. The normalizing constant c_π is $\sum_{k=0}^{1} S_k \pi_k = S_0\pi_0 + S_1\pi_1$, where S_0 and S_1 are the number of copies present in stage 0 and 1 that produce mutations with nonzero allele frequencies, respectively. Although there are two copies present in stage 0, only one produces mutations with a nonzero allele frequency in the final state. Therefore, $S_0 = 1$ rather than 2, and $c_\pi = \pi_0 + 2\pi_1$. This produces the two linear equations $q_{1/2} = 2\pi_1/(\pi_0 + 2\pi_1)$ and $q_1 = \pi_0/(\pi_0 + 2\pi_1)$. Together, these facts result in the system of equations

$$\begin{pmatrix} q_{1/2} \\ q_1 \end{pmatrix} = \begin{pmatrix} 0 & 2 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \pi_0/(\pi_0 + 2\pi_1) \\ \pi_1/(\pi_0 + 2\pi_1). \end{pmatrix}$$

Solving for π in terms of q, we have $\pi_0 = q_{1/2}/(2q_1 + q_{1/2})$ and $\pi_1 = 2q_1/(2q_1 + q_{1/2})$. From here, it is clear to see that plugging in empirical estimates for $q_{1/2}$ and q_1 would provide an estimate of π_0 and π_1.

Restricting our focus now to gain events, we can demonstrate that in the case of sequentially accumulated gains where each stage is the addition of only one copy, only one event history results in an invertible A: this is the event history where all of the gains occur on a single line of descent. This results in a simple form for A and its inverse. This also implies that the minor copy number, i.e. the smaller of the maternal or paternal copy numbers, must be 1. The converse is not true: a minor copy of 1 is not sufficient to guarantee identifiability.

We illustrate the math for the simplest sequentially accumulated gain, a single gain ($K = 1$), to make the abstraction in the proof that follows more accessible. Without loss of generality, we assume the gain occurs on the maternal (M) copy rather than the paternal (P) copy. The major copy number is then the maternal copy number, 2, and the minor copy number is the paternal copy number, 1. This trivially fits the sequential gain scenario mentioned in the previous paragraph.

If a mutation occurred on the M copy before the gain, it would be present on two of three copies in the final state, therefore resulting in allele frequency 2/3. Any other mutation would have allele frequency 1/3 because it would be present on only one of the three final copies. Thus, the A matrix is 2×2 and has rows corresponding to 1/3 and 2/3. Since all copies persist to the final tumor state, S_0 and S_1 in the normalizing constant $c_\pi = S_0\pi_0 + S_1\pi_1$ are simply the numbers of copies in stage 0 and 1, respectively. Together, these facts result in the following relationship:

$$\begin{pmatrix} q_{1/3} \\ q_{2/3} \end{pmatrix} = \begin{pmatrix} 1 & 3 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \pi_0/(2\pi_0 + 3\pi_1) \\ \pi_1/(2\pi_0 + 3\pi_1) \end{pmatrix}$$

Equivalently, we can say that $q_{1/3} = (\pi_0 + 3\pi_1)/(2\pi_0 + 3\pi_1)$ and $q_{2/3} = \pi_0/(2\pi_0 + 3\pi_1)$. Solving for π_0 and π_1, we obtain $\pi_0 = 3q_{2/3}/(q_{1/3} + 2q_{2/3})$ and $\pi_1 = (q_{1/3} - q_{2/3})/(q_{1/3} + 2q_{2/3})$.

Panels (a) and (b) of Figure 2.6 give some examples of events in the three-gain setting: panel (a) shows the identifiable sequential gain case, whereas panel (b) shows an unidentifiable case of three gains with a minor copy of 1. The time spent in each stage, π, is

unidentifiable in the latter case because only three allele frequencies are observed in the final tumor stage–1/5, 2/5, and 4/5–but there are four stages.

We note that the condition that the gains occur in a sequential, one-at-a-time way could be limiting in practice: at each stage, it is biologically possible for multiple copies to be duplicated simultaneously. In this case, there does not appear to be an obvious case that guarantees identifiability of π. Purdom et al. [2013] simulated events where a random set of the copies at each stage were chosen to be duplicated, and checked if the resultant A matrices were invertible. Only a small proportion of the simulated histories resulted in invertible A matrices. However, this simulation examined the set of possible gain events, assuming that all copies were equally likely to be duplicated; biologically, some gain events may be more likely than others.

We now formalize the claim that sequential gain with minor copy number of 1 results in an identifiable π, and present the proof.

Lemma 1. For a given copy number of $S = K + 2$, the only identifiable matrix A in the setting of sequential gain is given by a matrix of the form

$$A = \begin{pmatrix} 0 & 0 & 0 \cdots 0 & 0 & 1 \\ 0 & 0 & 0 \cdots 0 & 1 & 0 \\ 0 & 0 & 0 \cdots 1 & 0 & 0 \\ \vdots & & \cdots & & \\ 0 & 1 & 0 \cdots 0 & 0 & 0 \\ 1 & 0 & 0 \cdots 0 & 0 & 0 \end{pmatrix} + e_1(1, 2, \ldots, S-1)^T$$

where e_1 is the unit vector.

In this case, A^{-1} has a simple form and gives simple relationships between q and π:

$$A^{-1} = \begin{pmatrix} 0 & 0 & 0 \cdots 0 & 0 & 1 \\ 0 & 0 & 0 \cdots 0 & 1 & 0 \\ 0 & 0 & 0 \cdots 1 & 0 & 0 \\ \vdots & & \cdots & & \\ 0 & 1 & 0 \cdots 0 & 0 & 0 \\ 1 & 0 & 0 \cdots 0 & 0 & 0 \end{pmatrix} - \frac{1}{S} e_1(S-1, \ldots, 2, 1)^T$$

Then we have that

$$q_j = \begin{cases} \pi_{S-j-1}/c_\pi & j = 2, \ldots, S-1 \\ (\pi_K + \sum_k (k+1)\pi_k)/c_\pi & j = 1 \end{cases}$$

$$\pi_k = \begin{cases} q_{S-k-1}/d(q) & k = 0, \ldots, K-1 \\ q_1 - 1 + \sum_j \frac{j}{S} q_j & k = K \end{cases}$$

where $d(q) = \sum_j \frac{j}{S} q_j$ and $c_\pi = \sum_{k=0}^{K} S_k \pi_k$.

Proof. For this proof, the term allele frequency refers to the true allele frequency and not the estimated allele frequency. This is also evident from the context, but is mentioned here again for clarity.

Since we are assuming sequential gain, i.e. a single gain of the region at each stage, we begin with $S_0 = 2$ copies and have $S_k = 2+k$ copies for stages $1, 2, \ldots, K-1$. At the event's end, we have $S = S_K = K+2$ copies total. This event has $K+1$ possible allele frequencies– $\frac{1}{K+2}, \ldots, \frac{K+1}{K+2}$. The allele frequency $\frac{K+2}{K+2}$ is omitted, as this allele frequency implies that the mutation is present on all copies, which is impossible without a deletion at some stage to eliminate the reference allele. Thus, the matrix A is square with dimension $(K+1)\times(K+1)$. For A to be invertible, a necessary condition is that each allele frequency must be achieved, i.e. that no row in A is all zero. We demonstrate that the only possible event history is the one that corresponds to the A matrix given in the lemma statement.

Mutations acquired at each stage must attain the maximum allele frequency possible in order for A to be invertible, otherwise we would have $K+1$ distinct stages but less than $K+1$ allele frequencies in the final tumor state, meaning that at least two stages would share an allele frequency and be unidentifiable from each other.

The key insight is that at stage k, there are only $K-k$ gains remaining. Therefore, a mutation acquired in stage k can be on at most $K-k+1$ copies in the final tumor state. Equivalently, a mutation with allele frequency $j/(K+2)$ must have been acquired in stage $K-j+1$ or earlier.

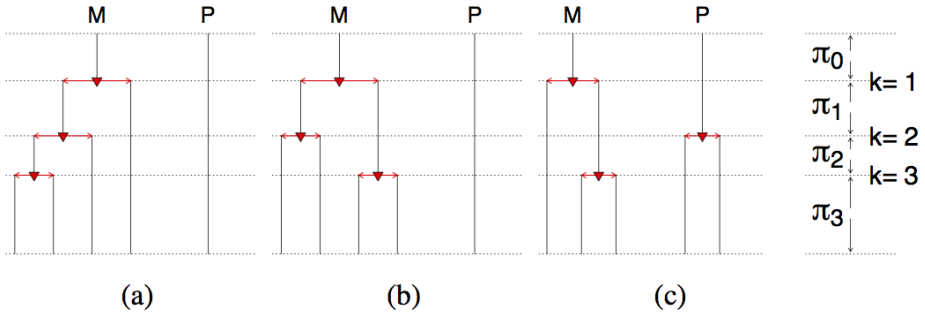
In order for a mutation acquired in stage k to be present in the maximum possible number of copies $K-k+1$, it must be the case that each stage after k involves a gain on a descendant of the same copy in stage k.

By recursion, this implies that the only history for a sequential gain that produces an invertible A is given by the one above: for a mutation to be present in $K+1$ copies, it must be acquired in stage 0 and K gain events that follow must be on descendants of the original copy carrying the mutation. Then, in stage 1, one of the two extant copies of the original carrying the stage 0 mutation must be gained. In stage 2, for the mutations in stage 1 to propagate, one of the two copies carrying both the stage 1 and stage 0 mutations must be gained, and so forth.

Equivalently, under this event timeline, mutations in stage 0 have two possible allele frequencies in the final tumor state: they were either on the copy which persisted to the end of the event, resulting in allele frequency $\frac{K+1}{K+2}$, or they were on the other copy, which was not duplicated at all, resulting in an allele frequency of $\frac{1}{K+2}$. By the same reasoning, mutations in stage k have the possible allele frequencies $\frac{K-k+1}{K+2}$ and $\frac{1}{K+2}$.

This implies that $A_{1,1} = 1$ and $A_{K+1,1} = 1$, while $A_{j,1} = 0$ for all other j; and columns corresponding to other stages of this event timeline would appear as in the matrix A given in the lemma statement. □

Figure 2.6: Three possible histories that result in a copy number of $S = 5$: the top representing the starting point with one copy each from maternal (M) and paternal (P). At each time point k there is a gain, until the tumor is removed after $k = 3$. The only identifiable history is (a) because all of its gains occur on one lineage. This figure originally appeared as Figure 1 in Purdom et al. [2013].



2.3.3 Modeling sequencing variability

2.3.3.1 Accounting for sample impurity

As we described in Section 2.2.2.3, real sequencing experiments on tumor samples are usually contaminated with DNA from neighboring normal cells or subclonal tumor populations. In describing our problem setting in Section 2.3.1, we mention that we ignore the effect of divergent subclonal cells and sequencing error, so we focus only on normal contamination.

For a particular mutation in the tumor, contamination by cells that do not have the same mutation results in an increase in the number of reference allele reads. The true variant allele frequencies are therefore diluted toward 0 from what they would have been in a pure tumor sample. We further make simplifying assumptions to limit the contamination to behave only in this predictable way, i.e. only by increasing the fraction of reference alleles over a location: we assume that at a location with a mutation in the tumor, the normal cells do not have the mutation at the same location and instead they are diploid in the reference. In practice, other outcomes are possible–for example the normal cells could have abnormal copy number. The latter is difficult to distinguish from sequencing error during mutation-calling unless it occurs sufficiently frequently and at high enough read coverage. The former would also require sufficient read coverage and mutations to detect, as this would manifest as the tumor showing uneven amounts of contamination in different regions. However, on the whole, these assumptions have not proven too limiting in practice: the other outcomes mentioned result in effects that would be too small or difficult to estimate with respect to the normal variation in read coverage.

To be precise, let w_N represent the fraction of the sample contaminated with normal

or subclonal cells. We no longer see the pure tumor allele frequency j/S, but the allele frequency for a mixture of two cell populations, $\rho_j = \frac{j(1-w_N)}{S(1-w_N)+2w_N}$. In the pure tumor, representing $1 - w_N$ of the sample, cells contribute S copies, of which j carry the mutation; in the contaminated portion comprising w_N of the sample, cells contribute 2 copies, none of which carry the mutation.

Given an estimate of w_N, we can estimate the observed true allele frequency with the plug-in estimator $\hat{\rho}_j = \frac{j(1-\hat{w}_N)}{S(1-\hat{w}_N)+2\hat{w}_N}$. In this next section, we provide a model for read coverage and use it to derive the estimator for w_N used in Durinck et al. [2011].

2.3.3.2 Estimating sample impurity

In Durinck et al. [2011], w_N was estimated from a Poisson model for the reads covering mutated locations, where the parameters of the Poisson modeled what proportion of reads came from the dominant tumor clone or the contaminating cells. Because read coverage increases with chromosomal copy number, we parameterized the model with a different mean for each continuous region of the genome with the same chromosomal copy number and allelic ratio–the same regions one finds from analyzing arrayCGH or other chromosomal copy number data.

Precisely, the read count covering a mutation at location i was modeled as

$$m_i \sim Poi(\lambda_{reg}[w^{(i)}_{ref,T} + w^{(i)}_{var,T} + w_N + \epsilon^{(i)}_N]),$$

where we constrain $w^{(i)}_{ref,T} + w^{(i)}_{var,T} + w_N + \epsilon^{(i)}_N = 1$ and $\sum_i \epsilon^{(i)}_N = 0$. This last condition on $\epsilon^{(i)}_N$ ensures that w_N is identifiable. Here, λ_{reg} represents the average read count over location i, and the expression in brackets splits the reads into three distinct sources: reads of tumor origin showing the reference ($w^{(i)}_{ref,T}$), reads of tumor origin showing the variant ($w^{(i)}_{var,T}$), and reads from normal or subclonal cells showing the reference ($w_N + \epsilon^{(i)}_N$). Note that although w_N represents the sample-wide average amount of contamination, $w_N + \epsilon^{(i)}_N$ represents the mutation-specific amount of contamination. This modeling choice was made since the amount of contamination per mutation changes depending on how many of the subclonal cells share the mutation with the dominant clone.

The key observation is that in unaltered regions of the tumor, i.e. diploid with one copy each of the maternal and paternal, all point mutations are heterozygous, so $w^i_{ref,T} = w^i_{var,T} \stackrel{def}{=} w^i_T$ for all locations i with point mutation. Under this model, the MLE for λ_{reg} is trivially m_i, since we observe only one Poisson instance for each location i. Likewise, X_i is the MLE estimate for $\lambda_{reg} w^{(i)}_{var,T}$. By the relation $\lambda_{reg}(w_N + \epsilon^{(i)}_N) = \lambda_{reg}(1 - 2w^{(i)}_T)$, and the fact that $\sum_i \epsilon^{(i)}_N = 0$, we have that an estimator for $w_N \sum_i \lambda_{reg}$ is $N\bar{m} - 2N\bar{X}$. Solving for w_N and plugging in $\bar{m}$ again as an estimator for λ_{reg}, we arrive at $\hat{w}_N = 1 - 2\frac{\bar{X}}{\bar{m}}$.

Intuitively, $\frac{\bar{X}}{\bar{m}}$ combines the read counts in the region to obtain a pooled allele frequency. Without contamination, variant reads are expected to be $\frac{1}{2}$ of all reads covering mutations in these unaltered regions. Then, the quantity $\frac{1}{2} - \frac{\bar{X}}{\bar{m}}$ measures how far the observed proportion

of variant reads deviates from expected. The estimator $\hat{w}_N$ is this expression multiplied by a factor of 2 since the variant allele accounts for only one of two copies in these unaltered regions.

2.3.4 Estimating π after accounting for sequencing variability

2.3.4.1 Accounting for sequencing variability

Earlier in Section 2.3.1 we laid out the problem setting and the data expected for our model and we provided a link between mutation allele frequency and the event timing vector π, but we delayed discussing the issue of how to determine the allele frequency of a mutation from sequencing data. In practice, the true allele frequencies must be estimated from the data. We now introduce a model for the variant reads for a mutation, which establishes a link between the observed sequencing data and the vector q.

For simplicity, we begin with the case where the sample does not contain normal contamination, so that the true allele frequencies p_i fall in the set $\{1/S, 2/S, \ldots, S/S\}$. We model the variant reads for each mutation X_i as Binomial(m_i, p_i), where m_i is the overall read depth m_i at locus i. This model essentially says that the chance that a read harbors the mutation is proportional to the the number of copies the mutation is on, i.e. that each copy is equally likely to be sequenced. In other words, we have that $P(X_i = x_i | p_i = j/S) = (j/S)^{x_i}(1 - j/S)^{m_i - x_i}$ from our Binomial model, and $P(p_i = j/S \mid p_i > 0) = q_j$ by definition of q.

To account for the fact that the tumor sample is a mix of the dominant clone and normal cells, we adjust the pure tumor allele frequencies of $1/S, \ldots, S/S$ to the mixed-sample allele frequencies of $\rho_1, \ldots, \rho_S$, where $\rho_j = \frac{j(1-w_N)}{S(1-w_N)+2w_N}$ and w_N is the known proportion of normal cells in the sample, as explained in Section 2.3.3.2. If this proportion is not known (as was the case in Durinck et al. [2011]), then it can be estimated from the sequencing data as described in Section 2.3.3.2, and the estimated mixed-sample allele frequencies $\hat{\rho}_j = \frac{j(1-\hat{w}_N)}{S(1-\hat{p}_N)+2\hat{w}_N}$ can be used instead. This gives us the model for variant allele coverage $P(X_i = x_i | p_i = \rho_j) = \rho_j^{x_i}(1 - \rho_j)^{m_i - x_i}$. No adjustment is needed for the q vector. The components of q simply correspond to different set of (adjusted) allele frequencies: now $q_j = P(p_i = \rho_j \mid p_i > 0)$ instead of $P(p_i = j/S \mid p_i > 0)$.

The true allele frequency p_i of each mutation is not known. We model it as a latent variable, which results in a Binomial mixture model for the variant reads X_i. Precisely, $X_i | p_i > 0 \ \sum_{j=1}^{S} Z_j \text{Binom}(m_i, \rho_j)$, where $Z_j = I\{\text{mutation } i \text{ has true allele frequency } \rho_j\}$.

This model results in the following expression for the log likelihood for X_i.

$$\begin{aligned}\log P(X_i|X_i > 0, q) &= \log\left(\frac{\sum_{j=1}^{S} P(X_i|p_i = \rho_j)q_j}{P(X_i > 0|q)}\right) \\ &= \log\left(\frac{\sum_{j=1}^{S} P(X_i|p_i = \rho_j)q_j}{1 - \sum_{j=1}^{S}(1-\rho_j)^{m_i}q_j}\right) \\ &= \log\left(\frac{\sum_{j=1}^{S} \rho_j^{x_i}(1-\rho_j)^{m_i - x_i}q_j}{1 - \sum_{j=1}^{S}(1-\rho_j)^{m_i}q_j}\right)\end{aligned}$$

From here, we can use maximum likelihood techniques to both infer q and the true allele frequencies for each mutation. As we explain in the next section, with an identifiable A matrix, this will also allow us to infer π.

2.3.4.2 Full maximum likelihood

In what follows, we assume that A corresponds to an event timeline with identifiable π, i.e. A has rank $K+1$. In section 2.3.1, we noted that allele frequencies which were not possible for an event history would appear in A as a row of all zeroes and could therefore be removed from the matrix. However, to simplify notation here by keeping the dimensions of A fixed, we assume that A has dimension $S \times (K+1)$, so that there is a row corresponding to each of the pure tumor allele frequencies of $1/S, 2/S, \ldots, S/S$, i.e. all of the pure tumor allele frequencies possible with S final copies.

As before, we have $q = A\pi/c_\pi$ where $c_\pi = \sum_k S_k\pi_k$ is a normalizing constant that ensures q sums to 1. Because A has rank $K+1$, we can find a matrix A^+ such that $A^+q = \pi/c_\pi$. For example, if by SVD $A = UDV^T$, then we can take $A^+ = VD_1^{-1}U^T$ where D_1 is the diagonal matrix that makes up the first $K+1$ rows of D and likewise for U_1 and U.

Because π must sum to 1, we have $1^T\pi = 1$. Multiplying both sides of $A^+q = \pi/c_\pi$ by 1^T, we can express the constant c_π in terms of A^+ and q: $1^TA^+q = 1^T\pi/c_\pi = 1/c_\pi$. This allows us to write π in terms of just A^+ and q as well: $\pi = \frac{A^+q}{1^TA^+q}$.

If $\hat{q}$ is the MLE for q, we have by the invariance property of MLEs that $\hat{\pi} = \frac{A^+\hat{q}}{1^TA^+\hat{q}}$ is the MLE for π; therefore, it is sufficient to find the maximum likelihood estimate for q.

Rather than directly maximizing the likelihood via a constrained optimization, we chose to maximize it using the Expectation-Maximization (EM) algorithm, treating the true allele frequencies for each mutation, p_i, are considered the latent or missing variables. Using the EM algorithm allowed us to use the same framework for implementing our method (full MLE) and the partial MLE method of Greenman et al. [2012] discussed in the next section. There is no other advantage to using the EM algorithm in this context, since the M-step already involves a constrained optimization so that the q vector produces valid probabilities. We expect that the results from either approach would be similar.

In our implementation, the E-step is given by

$$\begin{aligned}
Q(q|q^{(t)}) &= E_{p_i|X_i,q^{(t)}}[\log L(q;X,p)] = E\left\{\log\left(\prod_{i=1}\frac{P_q(X_i,p_i)}{P_q(X_i>0)}\right)|X_i,q^{(t)}\right\} \\
&= E\left\{\sum_i \log[P(X_i|p_i)P_q(p_i)]|X_i,q^{(t)}\right\} - \sum_i \log(1-\sum_j(1-\rho_j)^{m_i}q_j) \\
&= \sum_i\sum_j P(p_i=\rho_j|X_i,q^{(t)})\log q_j - \sum_i \log(1-\sum_j(1-\rho_j)^{(m_i)}q_j) + \text{constant} \\
&= \sum_j Y_j \log q_j - \sum_i \log(1-\sum_j(1-\rho_j)^{(m_i)}q_j) + \text{constant} \\
&\text{where } Y_j = \sum_i P(p_i=\rho_j|X_i,q^{(t)}).
\end{aligned}$$

The M-step then consists of maximizing $Q(q|q^{(t)})$ above with respect to q over the set $\Omega = \{q : A^+q \succeq 0,\ 1^Tq = 1\}$ because q represents a vector of probabilities and therefore must be non-negative and sum to 1. We parameterize the likelihood in terms of $q_S = 1 - \sum_{j=1}^{S-1} q_j$ so that the likelihood is proportional up to a constant to $\sum_j^{S-1} Y_j \log q_j + Y_s \log(1-\sum_{j=1}^{S-1} q_j)$. In terms of the vector $q_{-S} = (q_1,\ldots,q_{S-1})$ the constraint $A^+q \succeq 0$ is given as

$$A^+\begin{pmatrix} I_{S-1} \\ -1_{S-1}\end{pmatrix} q_{-S} + A^+e_S \succeq 0$$

where $e_S = (0,\ldots,0,1)^T$, I_m is the $m \times m$ identity matrix and 1_m is the m-length vector of ones. In the case of sequential gains, the constraint on q_j could be written more simply as $\sum_j \rho_j q_j \geq 1 - q_1$.

We implemented the M-step using the *constOptim* function in R which allows for constraints on a parameter θ in the form of $U\theta - c \succeq 0$. In our setting, the full set of constraints results in

$$U = \begin{pmatrix} A^+\begin{pmatrix} I_{S-1} \\ -1_{S-1}\end{pmatrix} \\ -I_{S-1} \\ I_{S-1}\end{pmatrix}, \text{ and } c = \begin{pmatrix} -A^+e_S \\ 1_{S-1} \\ 0_{S-1}\end{pmatrix}.$$

In Durinck et al. [2011], we used a semiparametric bootstrap to construct a confidence interval for π. The total number of mutations N and each mutation's read depth were held fixed, and the true allele frequencies and variant read counts were generated according to the multinomial mixture model with parameter $\hat{q}$.

2.3.4.3 Partial maximum likelihood: Greenman et al. [2012]

Greenman et al. [2012] also proposed the basic relation $q \sim A\pi$ of Section 2.3.1, but they did not account for sequencing variability in their model. Instead, they treat the maximum

likelihood assignments of mutations to their allele frequencies as the truth when performing ordering.

Their method can be framed as a modification of the E-step in our full maximum likelihood approach, which is why we refer to it as "partial" maximum likelihood. Our method uses a probabilistic assignment of mutations to allele frequencies based on the data X_i and the current choice of $q^{(t)}$. This is a well-known property of the EM algorithm with exponential families. If p_i were known, $Q(q|q^{(t)})$ would have the same form except Y_j would use the true allele frequency assignments, i.e. $Y_j = \sum_i I(p_i = \rho_j)$, and there would only be a single M-step since there would be no latent variables to estimate, and therefore no need to iterate.

The method of Greenman et al. [2012] is a hybrid approach: they use an allele frequency assignment that is estimated but treated as the truth, so $Y_j = \sum_i I(\hat{p}_i = \rho_j)$, where $\hat{p}_i$ is the MLE classification of the allele at location i. In other words, Greenman et al. [2012] assigns each mutation to its maximum likelihood allele frequency, and then uses the resultant counts to obtain q and estimate π.

Rather than using probabilistic assignments as in our full MLE approach, the partial MLE approach fails to account for the variability in estimating the allele frequencies because they treat the MLE assignments as ground truth in determining π. Intuitively, this makes a difference in edge cases where there is insufficient read support for a mutation to confidently determine its allele frequency among the other possibilities–cases where the maximum likelihood assignment could be incorrect. This could happen for a number of reasons: 1) there could simply be low read coverage; or there are several close possible allele frequencies because of 2) high copy number or 3) heavy normal contamination, or both. The results shown later in Section 2.4 confirm this intuition. For high sequencing depth, the maximum likelihood assignment tends to identify the correct allele frequencies and there is little difference between our method and that of Greenman et al. [2012]; but for lower levels of sequencing, explicitly accounting for the sequencing variability brings improved stability.

2.3.4.4 Bayesian estimation approach to mitigate instability when π_0 is small

We are often interested in finding the region with the smallest π_0, i.e. the region whose copy number change occurred first. However, the smaller the π_0, the less time that the region had to accumulate mutations prior to the first change. Since the robustness of the full MLE method depends on the number of mutations corresponding to each stage, estimates of π_0 are more unstable for earlier events.

One way to mitigate this instability is to try a Bayesian approach, placing a prior on the π vector. This introduces bias to the π_0 estimate but could result in decreased variance by "borrowing" information from stages with more mutations.

Purdom et al. [2013] explored placing a uniform prior on π, or equivalently, a Dirichlet(α) with $\alpha_i = 1$ for each $i = 1, \ldots, K$. Because the Dirichlet distribution is not a conjugate prior for the distribution of X_i, Purdom et al. [2013] sampled from the posterior distribution of π using sampling important resampling (SIR) to calculate the posterior mean and credible

intervals. We derive the log posterior distribution of π and explain the implementation in greater detail below.

For a region with N mutations and their variant read depths given by the usual notation $X = (X_1, \ldots, X_N)$, we can write the posterior distribution of π as

$$f(\pi|X, \alpha) = \frac{P(X|\pi, \alpha) f(\pi|\alpha)}{P(X|\alpha)} \tag{2.3}$$

where $f(\pi|\alpha) = \frac{1}{B(\alpha)} \prod_{k=0}^{K} \pi_k^{\alpha_k - 1}$, the density for a Dirichlet(α) distribution. $P(X|\pi, \alpha)$ can be expanded as $\prod_{i=1}^{N} P(X_i|\pi_i, \alpha)$ since each mutation is independent. Since X_i depends on allele frequency, we condition on the true allele frequency ρ_j and use the fact that $q_j = \frac{(A\pi)_j}{c_\pi}$ in order to obtain the following expression for $P(X_i|\pi, \alpha)$ in terms of π:

$$P(X_i|\pi, \alpha) = \sum_{j=1}^{S} P(X_i|\rho_j, \pi, \alpha) q_j = \sum_{j=1}^{S} P(X_i|\rho_j, \pi, \alpha) \frac{(A\pi)_j}{c_\pi}. \tag{2.4}$$

Substituting into equation 2.4 and setting $\alpha_i = \alpha$ constant for all i, we obtain the following expression for the log posterior distribution of π.

$$\log(f(\pi|X, \alpha)) = C + (\alpha - 1) \sum_{k=0}^{K} \log(\pi_k) + \sum_{i=1}^{N} \log \left(\sum_{j=1}^{S} P(X_i|\rho_j, \pi, \alpha) \frac{(A\pi)_j}{c_\pi} \right) \tag{2.5}$$

where $C = \frac{P(X|\alpha)}{B(\alpha)}$ is a constant that does not depend on π.

For the Bayesian estimation, we perform the change of variable $h : (\pi_0, \ldots, \pi_{K-1}, \pi_K) \to (\theta_0, \ldots, \theta_{K-1})$, where $\theta_j = \log(\frac{\pi_j}{\pi_K})$, where $j = 0, \ldots, K-1$. Rather than estimating π with the constraint $\sum_{i=0}^{K-1} \pi_i \leq 1$, this allows us to work with $\theta \in \mathbb{R}^K$. In working with the summations below, it will be convenient to define $\theta_K = 0$ so $e^{\theta_K} = 1$.

Then, this results in the change of variable

$$\pi_j = \frac{\pi_j}{\pi_K} \pi_K = \frac{e^{\theta_j}}{\sum_{k=0}^{K} e^{\theta_k}}$$

for $j = 1, \ldots, K$, and the Jacobian becomes

$$|J(\theta)| = \left| \frac{d\pi_0 \pi_1 \ldots \pi_{K-1}}{d\theta_0 \theta_1 \ldots \theta_{K-1}} \right| = \prod_{j=0}^{K-1} \frac{e^{\theta_j} (1 + \sum_{k=0}^{K-1} e^{\theta_k} - e^{\theta_j})}{(\sum_{k=0}^{K} e^{\theta_k})^2} = \prod_{j=0}^{K-1} \frac{e^{\theta_j} (\sum_{k \neq j}^{K} e^{\theta_k})}{(\sum_{k=0}^{K} e^{\theta_k})^2}.$$

For sequential gain, we begin with $S_0 = 2$ copies and increase by one at each stage, so $S_1 = 3, \ldots, S_K = 2 + K$. In this case, the normalizing constant c_π simplifies to

$$c_\pi = \sum_{k=0}^{K}(2+k)\pi_k = \frac{\sum_{k=0}^{K}(2+k)e^{\theta_k}}{\sum_{k=0}^{K}e^{\theta_k}}.$$

The log posterior distribution of π can now be expressed in terms of θ as follows.

$$\begin{aligned}
\log(f(\pi|X,\alpha)) &= C + (\alpha-1)\sum_{k=0}^{K}\log(\pi_k) + \sum_{i=1}^{N}\log\left(\sum_{j=1}^{S}P(X_i|\rho_j,\pi,\alpha)\frac{(A\pi)_j}{c_\pi}\right) + \log|J(\theta)| \\
&= C + (\alpha-1)\sum_{k=0}^{K-1}(\theta_k - \log(\sum_{l=0}^{K}e^{\theta_l})) \\
&\quad + \sum_{i=1}^{N}\log\left(\sum_{j=1}^{S}P(X_i|\rho_j,\pi,\alpha)\frac{(Ae^{\theta})_j}{\sum_{l=0}^{K}(2+l)e^{\theta_l}}\right) \\
&\quad + \sum_{k=0}^{K-1}\left(\theta_k + \log(\sum_{l\neq k}e^{\theta_l}) - 2\log(\sum_{l=0}^{K}e^{\theta_l})\right) \\
&= C + \alpha\sum_{k=0}^{K-1}\theta_k - K(\alpha+1)\log(\sum_{l=0}^{K}e^{\theta_l}) \\
&\quad + \sum_{i=1}^{N}\log\left(\sum_{j=1}^{S}P(X_i|\rho_j,\pi,\alpha)\frac{(Ae^{\theta})_j}{\sum_{l=0}^{K}(2+l)e^{\theta_l}}\right) + \sum_{k=0}^{K}\log(\sum_{l\neq k}e^{\theta_l}) \\
&= C + \alpha\sum_{k=0}^{K-1}\theta_k - K(\alpha+1)\log(\sum_{l=0}^{K}e^{\theta_l}) - N\log(\sum_{l=0}^{K}(2+l)e^{\theta_l}) \\
&\quad + \sum_{i=1}^{N}\log\left(\sum_{j=1}^{S}P(X_i|\rho_j,\pi,\alpha)(Ae^{\theta})_j\right) + \sum_{k=0}^{K}\log(\sum_{l\neq k}e^{\theta_l}) \\
&= C + \alpha\sum_{k=0}^{K-1}\theta_k - K(\alpha+1)\log(\sum_{l=0}^{K}e^{\theta_l}) - N\log(\sum_{l=0}^{K}(2+l)e^{\theta_l}) \\
&\quad + \sum_{i=1}^{N}\log\left(\sum_{m=0}^{K}e^{\theta_m}[\sum_{j=1}^{S}P(X_i|\rho_j,\pi,\alpha)A_{j(m+1)}]\right) + \sum_{k=0}^{K}\log(\sum_{l\neq k}e^{\theta_l})
\end{aligned}$$

Because the constant C is not explicitly calculable, Purdom et al. [2013] sampled from the posterior distribution of θ in order to estimate the posterior mean and posterior credible intervals. They used sampling importance resampling to sample from the posterior distribution, with the proposal distribution given by the multivariable t density with four degrees of freedom, the mean equal to the posterior mode, and the variance matrix given by the delta

method approximation, $V = (-h''(\hat{\theta}))^{-1}$) where h is the posterior distribution. They used the basic implementation given in the *LearnBayes* package in R available on CRAN.

Because terms like $\sum_{l=0}^{K} e^{\theta_l}$ can be unstable when θ is large, i.e. when π is near the boundary of the simplex, Purdom et al. [2013] stabilized the estimates by applying the standard technique of letting $m = \max_i(\theta_i)$ on $\theta_i > 10^{20}$: recall that for any constants c, a_i $(i = 1, \ldots, K)$, we have the approximation

$$\log(c + \sum_{i=1}^{K} a_i e^{\theta_i}) \approx \log(\sum_{i=1}^{K} a_i e^{\theta_i}) = \log(\sum_{i=1}^{K} a_i \frac{e^m}{e^m} e^{\theta_i}) = m + \log(\sum_{i=1}^{K} a_i e^{\theta_i - m}).$$

2.4 Simulation data results

We focus on the estimation of π_0, which indicates when the first change to the region occurred, which was the moment of greatest biological interest in Durinck et al. [2011]. Purdom et al. [2013] simulated mutation data for different histories under the model described above. The purpose of these simulations was to determine how estimates of π_0, the time of the first event, are affected by the total number of mutations in a region (N); the sequencing read coverage, and the true magnitude of π_0 itself. Bootstrap confidence intervals for each π_0 estimate in the same manner done for Durinck et al. [2011] to determine the accuracy of the desired coverage probability.

The simulation data was generated as follows: for several combinations of A and π, the probabilities of each allele frequency were computed by $q = A\pi/c_\pi$. These probabilities were used to generate allele frequencies for a set of N mutations from a multinomial distribution. Precisely, mutations have allele frequencies $p = (p_1, \ldots, p_N) \sim \text{Multinomial}(N, q)$, $i = 1, \ldots, N$. Each mutation was set to have the same read depth m, so variant read counts were generated from a Binomial(m, ρ_i) distribution. In the simulations, the total number of mutations N took values in 10, 25, 50, 75, and 125; while the mutation read depth took values 10, 30, or 75. These read depths correspond to low, moderate, and high coverage in practice–it is not common for sequencing experiments to have average read coverage far surpassing 75x. Multiple settings for N were chosen to encompass several plausible scenarios in cancer: low values of N could be the result of a low overall mutation rate or a short region of chromosomal aberration; high values of N are possible in the converse situation, long regions of chromosomal aberration in highly mutated cancers such as those of the skin.

Figures 2.7 and 2.4 show boxplots of $\hat{\pi}_0$ with true $\pi_0 \leq 0.10$ under several settings of N and read depth for event types CNLOH, single gain, and two sequential gains; the same for true $\pi_0 \geq 0.10$ appear in Figures 2.9 and 2.10. For convenience, one setting of the vector π was chosen for each setting of π_0 in the plots for two sequential gains after additional simulations showed that the results did not appear to depend on the values of the other components in the π vector. Regardless of event type, estimating very small π_0, e.g. $\pi_0 = 0.01$, is challenging even at high read coverage per mutation. In general, the full MLE method tends to underestimate π_0 when there are insufficient mutations. Even in the simple

cases of one-stage events, 200 or 300 mutations are required to produce unbiased estimates of $\pi_0 = 0.01$. This occurs because for all three event types, a single allele frequency is identified with and only with mutations occurring in stage 0; thus, when π_0 is this small, this allele will have a low probability of occurrence as given in q. Unsurprisingly, if no mutations are observed for the allele frequency corresponding to π_0, then $\hat{\pi}_0$ will be 0, which explains the tendency of the boxplots of Figures 2.7 and 2.4 to have median 0 when π_0 and N are low enough to make the chance of observing a mutation at the π_0 allele frequency low as well.

These simulations show that for small π_0, the more pressing difficulty is observing enough mutations to estimate q well rather than determining the correct allele frequencies of those observed. Having high read coverage helps with the latter but not the former; only having more mutations can mitigate the former. Further, although the two gain case results in the possible allele frequencies of $1/4, 2/4$, and $3/4$, which have less separation between them than those for CNLOH ($1/2$ and 1) or single gain ($1/3$ and $2/3$), the simulations in Figures 2.7, 2.4, 2.9, and 2.10 were done without normal contamination to further compress the range of allele frequencies. Doing so would have made identification of the correct allele frequency per mutation more difficult, thereby making the role of read depth more apparent–essentially, the allele frequencies for the event types shown are still too separated for read depth to make a meaningful difference. We find evidence for this in looking at the simulations for moderate values of π_0, where the same trend persists: estimation of π_0 is generally unbiased for all settings of N tried (consistently so for $N \geq 50$), with less variance for larger N. The role of read depth here is again minimal, which can be observed in Figures 2.9 and 2.10 by seeing that the white boxplots for each setting of N are comparable to the gray ones. It still stands that identification of the allele frequencies is a smaller issue than not having enough data.

While only a two-stage sequential gain ($K = 2$) is shown in these boxplots, these results would extend to gains of higher stages because additional copies would only serve to exacerbate the imbalance in q. In a sequential gain, mutations acquired in every stage contribute to the allele frequency corresponding to $1/S$ in the pure tumor, resulting in a relatively large value of q_1. In fact, it is possible to show that q_1 is guaranteed to be the most likely allele frequency in this case. The q probabilities are already imbalanced regardless of the value of π, thus an imbalance in π worsens the imbalance in q. This means that certain categories of mutations very difficult to observe, and also that estimates of π in this setting are unstable. Although the results shown in the boxplots of Figures 2.9 and 2.10 focus on estimating π_0, we note that observing and identifying mutations for all possible allele frequencies becomes more difficult in events with more stages. Estimating q with confidence requires that we have sufficient mutations for each possible allele frequency. Also, an increase in the number of observable allele frequencies means read coverage would play a more important role in estimation.

Purdom et al. [2013] examined the coverage accuracy of the semiparametric bootstrap confidence intervals for π in Figures 2.11. In theoretically well-behaved situations, a 95% confidence interval by definition would cover the true parameter 95% of the time on average. To examine whether this would hold true for the bootstrap confidence intervals constructed for π_0, Purdom et al. [2013] constructed confidence intervals for each $\hat{\pi}_0$ based on taking

$B = 500$ bootstrap samples each. They examined the $K = 1$ event types single gain and CNLOH with $\pi_0 = 0.01, 0.05, 0.10, 0.30$, and 0.50. The percentage of confidence intervals covering each possible value of π_0 was plotted in Figure 2.11 using a blue-to-red color scale, where red indicated $\geq 95\%$ coverage. Figure 2.11 shows only the results for data simulated with 30x read depth and no normal contamination, other simulation settings were tried and produced similar results. It was found that the true coverage probabilities of the confidence intervals are less than 95%, and that the confidence intervals tend to be biased toward lower values of π_0, with less bias for larger values of N.

2.4.1 Full MLE vs. partial MLE of Greenman et al. [2012]

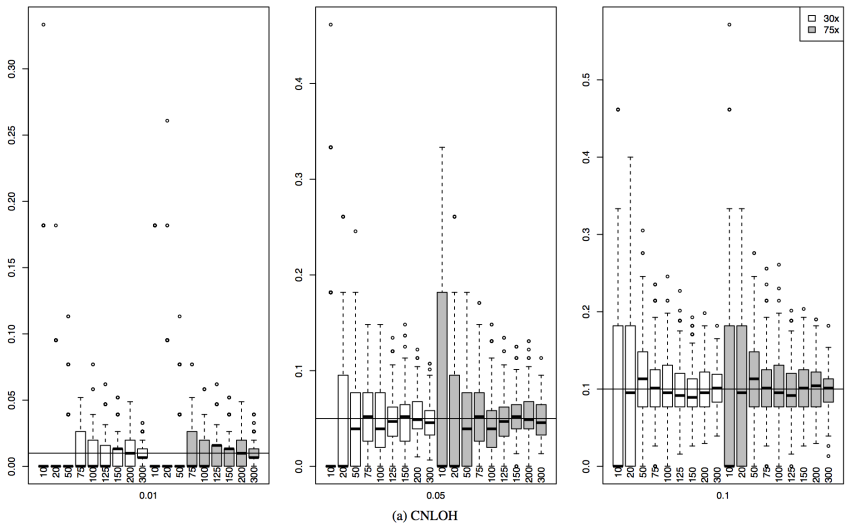
As explained in Section 2.3.4.3, we expect the greatest difference between the partial MLE of Greenman et al. [2012] and our full MLE to occur when there is the most uncertainty in classifying mutations to a particular allele frequency: this occurs when the mutation read coverage is low or the possible allele frequencies are close together, which can happen as a result of having several possible allele frequencies (i.e., high copy number in the final state) or high levels of normal contamination to compress the allele frequency range. To study the differential behavior of these methods carefully, Purdom et al. [2013] simulated data for each combination of these scenarios: small (0.10) vs. moderate (0.50) values of π_0; sequential gain in $K = 1, 2$ and 4 stages; 0% vs. 30$ normal contamination; and moderate (30x) vs. high (75x) read depth. The simulations for $K > 1$ encompass several possible π vectors with small and moderate π_0. For all of the simulations, the number of mutations was fixed at a high value of $N = 125$ so that the effect of the factors of interest could be isolated. The results from these simulations are shown in Figures 2.12 (single gain or $K = 1$), 2.13 ($K = 2$), and 2.14 ($K = 4$).

Figure 2.12 shows that the partial MLE method overestimates π_0 even in the simplest idealistic case of single gain with moderate read depth (30x) and no normal contamination. The bias is worse when π_0 is small. With sufficiently high read coverage, the bias disappears even with the addition of some normal contamination. The need for higher read coverage to mitigate the effect of normal contamination is true in general, and is observed as well in the simulations for $K = 2$ and $K = 4$.

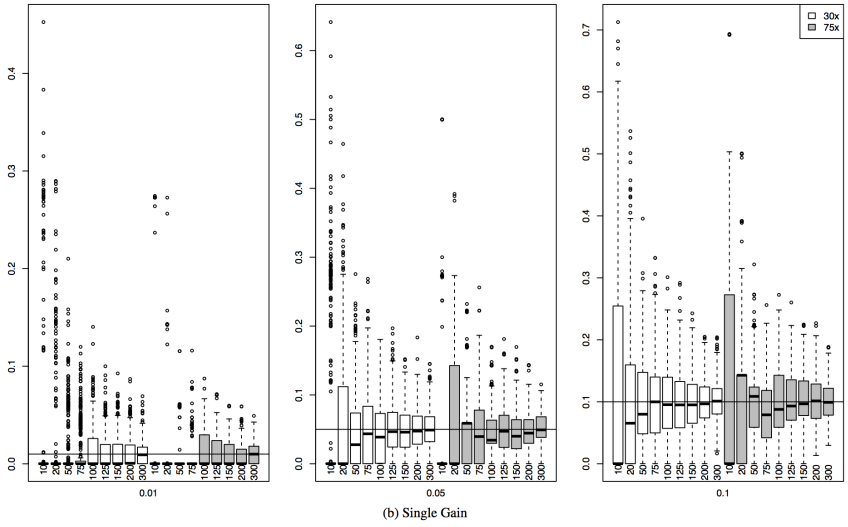
As expected, increasing the complexity of the event by increasing the number of stages exaggerates the bias observed in the simple setting. The general trend is for the partial MLE to overestimate small values of π_0 and underestimate large values, but the specific nature of the bias does depend on the other values in the π vector.

For example, in the $\pi_0 = 0.10$ examples for $K = 2$ shown in Figure 2.13, partial MLE actually underestimates the π_0 when $\pi = (0.1, 0.1, 0.8)$ but overestimates for $\pi = (0.1, 0.5, 0.4)$ and $\pi = (0.1, 0.8, 0.1)$. Intuitively, data generated from these last two π vectors are expected to have more mutations from $k = 1$. Recall that for the sequential gain setting, the allele frequency for the stage $k = 0$ event is the highest, corresponding to $(S - 1)/S$ in the pure tumor, and the allele frequencies uniquely corresponding to subsequent stages cascade from there: $(S - 2)/S, (S - 3)/S, \ldots, 1/S$. Thus, the allele frequencies from stage $k = 0$ and

Figure 2.7: Boxplots of estimates of π_0 for settings where $\pi_0 \leq 0.1$ and $K = 1$. The first row of plots correspond to three settings of π_0–0.01, 0.05, and 0.10–for CNLOH events and the second row to single gain. The horizontal lines indicate the true value of π_0. The white boxes correspond to a mutation read depth of 30x; grey to 75x. Each box represents a different setting of N, the total number of mutations in the region. No normal contamination is simulated. Note that the y-axis limits are different for each plot. This figure originally appeared as Supplementary Figure 2 in Purdom et al. [2013].



(a) CNLOH



(b) Single Gain

$k = 1$ are the most similar to each other, and having more mutations observed from stages adjacent to the one of interest (here $k = 0$) means greater chances of misclassification. For the π vectors $(0.1, 0.5, 0.4)$ and $(0.1, 0.8, 0.1)$, more mutations from stage $k = 1$ are expected than from $k = 0$, so the misclassifications will generally be of the type where a mutation from $k = 1$ is labeled $k = 0$, leading to an overestimate of the mutations belonging to $k = 0$, and therefore an overestimate of π_0.

In Figure 2.14, which examines the case of $K = 4$ sequential gains, we see how extreme the bias can get for the partial MLE. The second row represents an idealized situation of a pure tumor sample sequenced at high depth (75x). The value of π_0 was allowed to vary and the remaining probability was divided equally among the other components of π. The absolute difference from the true value of π_0 and the median estimate of π_0 increases as π_0 increases, reaching the largest amount of bias at $\pi_0 = 0.99$, meaning that the event occurred very late in the lifespan of the tumor. In fact, all of the partial MLE estimates in this setting were below the true value of π_0. The discussion so far has focused on how the partial MLE methods performs worse with normal contamination and on complex events because these increase the misclassification rate of mutations to allele frequencies. However, the difference between partial and full MLE methods depends on the complexity of the event in another way: in the partial MLE method, the estimated q changes in discrete increments because assignments of mutations to allele frequencies are taken as truth; in the full MLE method, q can change in a continuous manner because probabilistic assignments are used instead. For more complex events, small variations in the estimation of q result in larger perturbations of the estimated vector π. This can be seen in Figure 2.15, which shows the size of the gradient of π with respect to q. As a result, estimates from the partial MLE method are inherently more unstable than those from the full MLE method, and this problem is magnified as the complexity of the event increases.

2.4.2 Full MLE vs. Bayesian estimation

From a frequentist perspective, Bayesian estimates can result in less overall mean squared error (MSE) by trading an increase in bias for a decrease in variability. In simulations, Purdom et al. [2013] computed the relative MSE for each of the methods being compared (full MLE, partial MLE, and Bayesian method) for various settings of π_0. Relative MSE is the MSE scaled by the value of $\pi_0(1 - \pi_0)$ to reflect the size of the MSE relative to the size of π_0. The results from the simulations are plotted in Figures 2.16 (CNLOH), 2.17 (single gain), 2.18 (sequential gain in $K = 2$ and $K = 4$ stages). Additionally, the coverage of the full MLE bootstrap confidence intervals was compared to the Bayesian credible intervals, and blue-to-red color-scale plots with the same legend and markings as Figure 2.11 were generated for CNLOH and single gain in Figure 2.19.

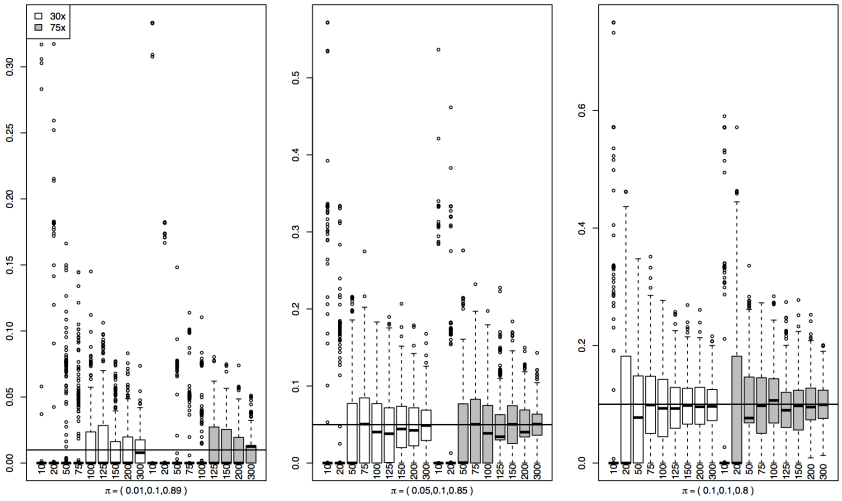
The performance of the Bayesian method against the full MLE is similar for all sequential gains, including the $K = 1$ stage single gain, but the behavior on CNLOH is very different, as we explain next.

In the case of sequential gain, the Bayesian estimates are similar to those from full MLE for all values of π_0. The original intention of the Bayesian estimation was to reduce estimation error for small values of π_0, where it was hoped that introducing a prior and thereby "borrowing information" from mutations from other stages would mitigate the problem of insufficient data from stage $k = 0$. The expectation was that the Bayesian estimates would shrink the estimates of π_0 closer to 0.5 as a result. While the shrinkage is observed in the simulation results, the Bayesian estimates actually have higher overall error than the full MLE for small values of π_0. Of the three methods, the full MLE offers the smallest relative MSE for small values of π_0.

Although the Bayesian estimates for sequential gain have higher error than those from full MLE, the credible intervals have better coverage than the bootstrap confidence intervals of the full MLE method for mid-range values of π_0. At either extreme–either very low or high π_0–neither method performs particularly well.

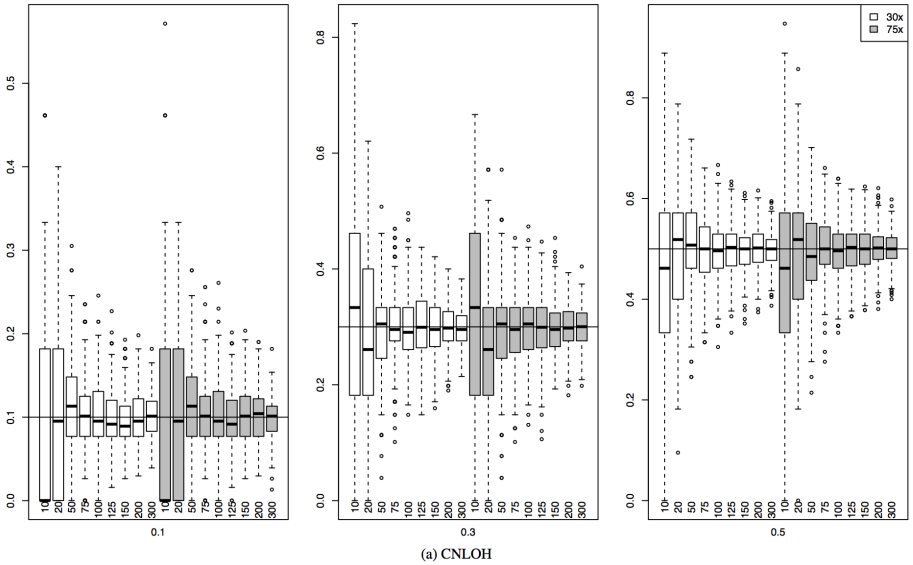
The opposite is true for CNLOH, where the Bayesian estimates do have smaller error than the full MLE method, as intended. The full MLE method still outperforms partial MLE. However, at around $\pi_0 = 1$, the relative MSE for the full MLE and the Bayesian estimates are equal, and thereafter the relative MSE for the Bayesian estimates are solidly higher. For $\pi_0 > 1$, the Bayesian estimates are severely biased downward, with relative MSE peaking at around $\pi = 0.5$; on the other hand, estimates from the full MLE method are improved as π_0 increases. As a result, even though the Bayesian method offers improvements for $\pi_0 < 0.10$, overall the full MLE method has less error across the range of π_0 values. In the region $\pi_0 < 0.10$ where the Bayesian estimation outperforms full MLE, the Bayesian credible intervals have a better coverage probability than the full MLE bootstrap confidence intervals and are less biased. This is a significant improvement over the full MLE confidence intervals for $\pi_0 < 0.10$, which tend to either produce very small or zero-width intervals.

Figure 2.8: Boxplots of estimates of π_0 for settings where $\pi_0 \leq 0.1$ and $K = 2$, i.e. two gains. The plots correspond to three settings of π_0–0.01, 0.05, and 0.10. For each π_0 a variety of values were compared for the remaining parts of the vector π, but results were similar, particularly for $N \geq 50$; the particular π shown was chosen for convenience. The horizontal lines indicate the true value of π_0. The white boxes correspond to a mutation read depth of 30x; grey to 75x. Each box represents a different setting of N, the total number of mutations in the region. No normal contamination is simulated. Note that the y-axis limits are different for each plot. This figure originally appeared as Supplementary Figure 3 in Purdom et al. [2013].

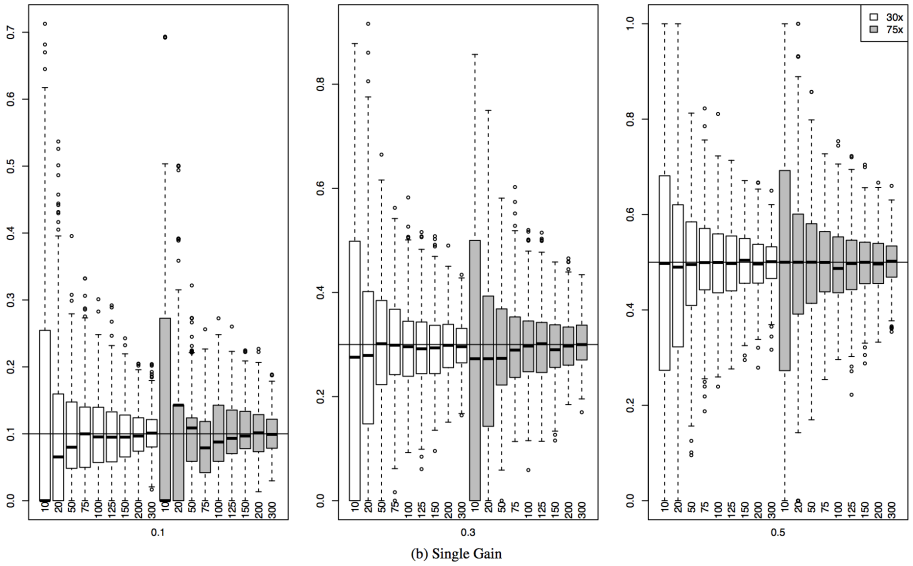


(a) Two Gains (K=2)

Figure 2.9: Boxplots of estimates of π_0 for settings where $\pi_0 \geq 0.1$ and $K = 1$. The first row of plots correspond to three settings of π_0–0.1, 0.3, and 0.5–for CNLOH events and the second row to single gain. The horizontal lines indicate the true value of π_0. The white boxes correspond to a mutation read depth of 30x; grey to 75x. Each box represents a different setting of N, the total number of mutations in the region. No normal contamination is simulated. Note that the y-axis limits are different for each plot. This figure originally appeared as Supplementary Figure 4 in Purdom et al. [2013].

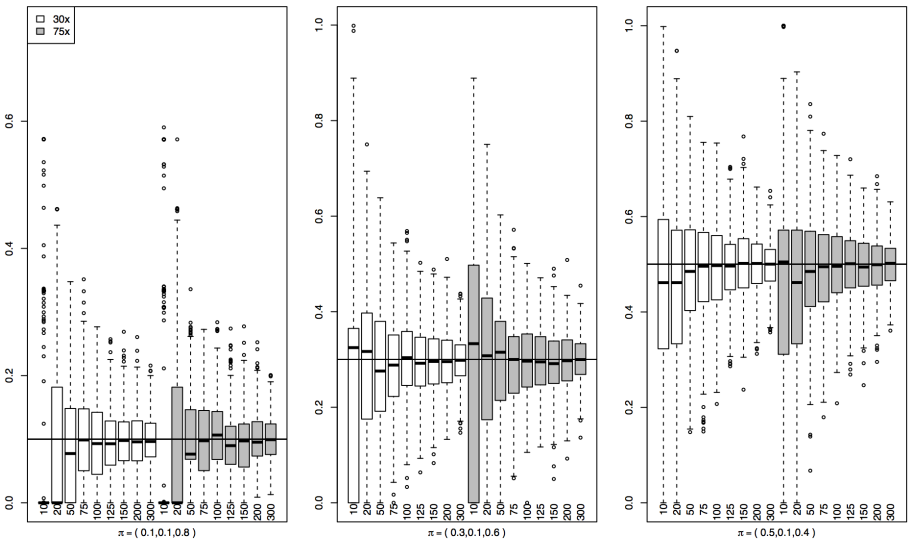


(a) CNLOH



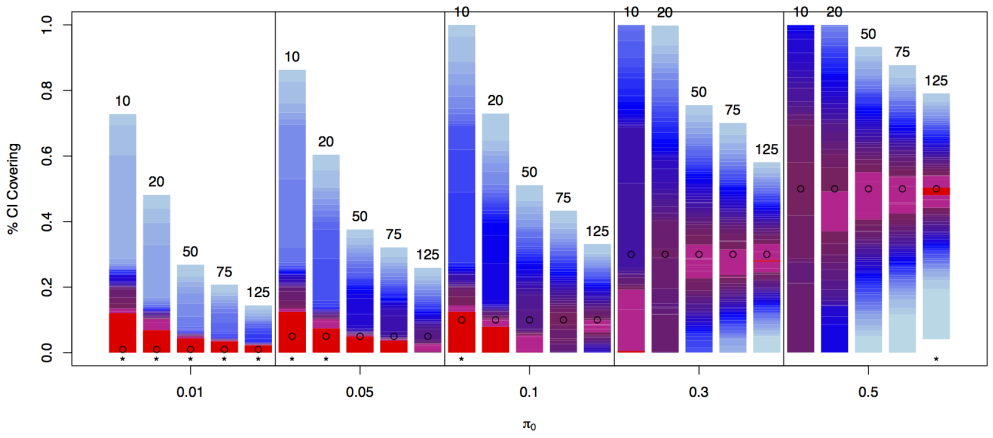
(b) Single Gain

Figure 2.10: Boxplots of estimates of π_0 for settings where $\pi_0 \leq 0.1$ and $K = 2$, i.e. two gains. The plots correspond to three settings of π_0–0.10, 0.30, and 0.50. For each π_0 a variety of values were compared for the remaining parts of the vector π, but results were similar, particularly for $N \geq 50$; the particular π shown was chosen for convenience. The horizontal lines indicate the true value of π_0. The white boxes correspond to an a mutation read depth of 30x; grey to 75x. Each box represents a different setting of N, the total number of mutations in the region. No normal contamination is simulated. Note that the y-axis limits are different for each plot. This figure originally appeared as Supplementary Figure 5 in Purdom et al. [2013].

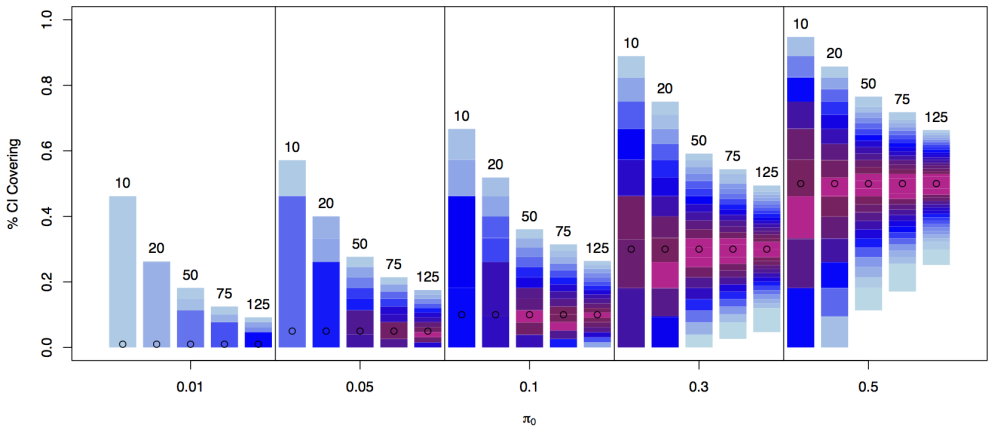


(a) Two Gains (K=2)

Figure 2.11: Coverage of 95% bootstrap confidence intervals on simulated data with a read depth of 30x and no normal contamination. For each simulation, a bootstrap confidence interval (B = 500) was constructed. The percentage of confidence intervals covering each value of π_0 is plotted using a blue to red color scale. Red indicates a coverage probability of $\geq 95\%$, and magenta $90\% - 95\%$. The true values of π_0 are indicated with black points. If a star appears underneath a plot, then the true value of π_0 has at least 95% coverage probability. This figure originally appeared as Supplementary Figure 6 in Purdom et al. [2013].



(a) Single Gain



(b) CNLOH

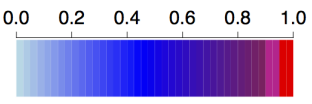
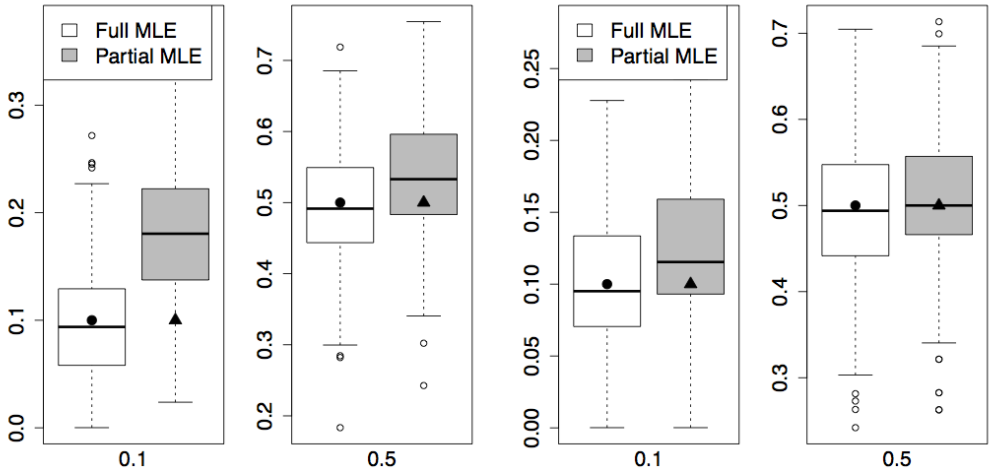


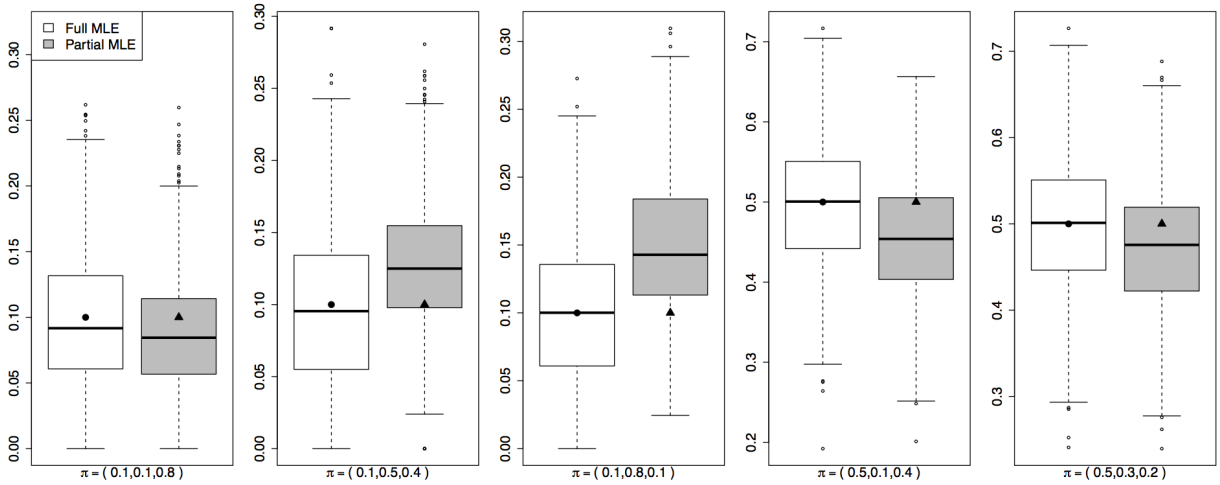
Figure 2.12: Boxplots of $\hat{\pi}_0$ from the full and partial MLE methods based on simulated data for single gain $\pi_0 = 0.1$ and $\pi_0 = 0.5$. In (a), reads were simulated with a depth of 30x and no normal contamination. In (b), reads were simulated with a depth of 75x and 30% normal contamination. For all simulations, $N = 125$. This figure originally appeared as Figure 2 in Purdom et al. [2013].



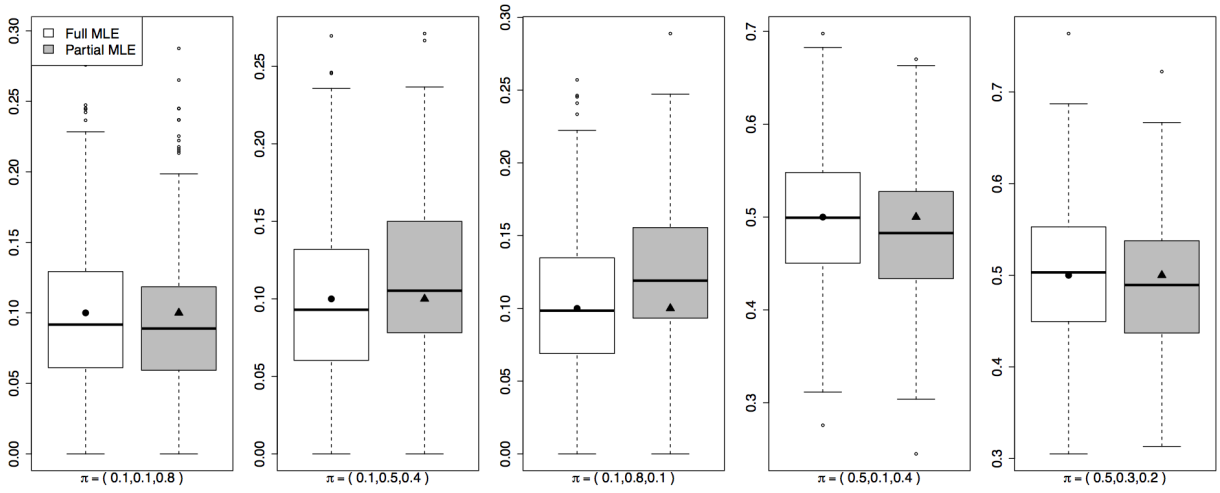
(a) Read depth 30x
$p_N = 0$

(b) Read depth 75x
$p_N = 0.3$

Figure 2.13: Boxplots of $\hat{\pi}_0$ from the full and partial MLE methods based on simulated data for two sequential gains ($K = 2$). In (a), reads were simulated with a depth of 30x and no normal contamination. In (b), reads were simulated with a depth of 75x and 30% normal contamination. For all simulations, $N = 125$. This figure originally appeared as Supplementary Figure 7 in Purdom et al. [2013].

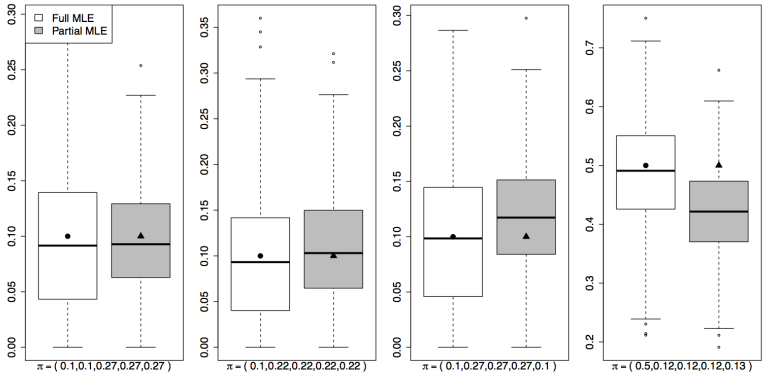


(a) Read depth 30x
$p_N = 0$

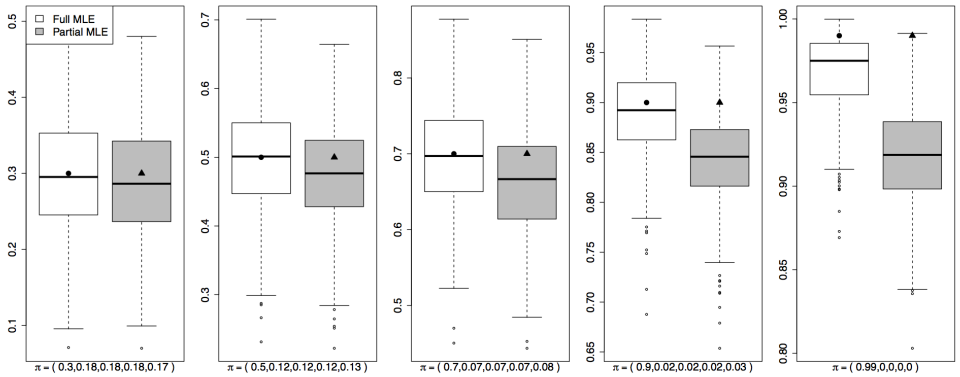


(b) Read depth 75x
$p_N = 0.3$

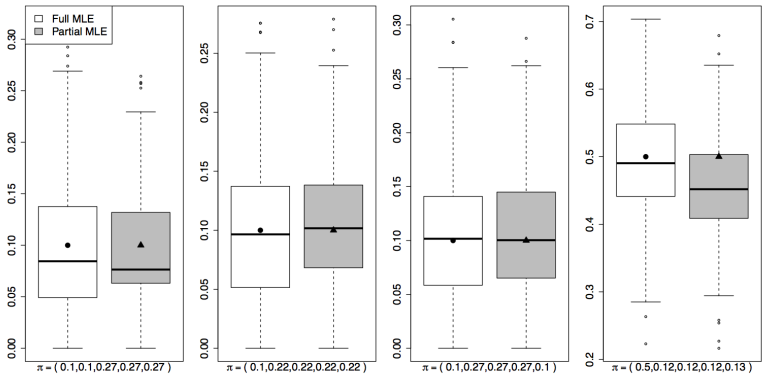
Figure 2.14: Boxplots of $\hat{\pi}_0$ from the full and partial MLE methods based on simulated data for four sequential gains ($K = 4$). In (a), reads were simulated with a depth of 30x and no normal contamination. In (b) and (c), reads were simulated at a depth of 75x, with (b) containing no normal contamination and (c) 30% normal contamination. For all simulations, $N = 125$. This figure originally appeared as Supplementary Figure 8 in Purdom et al. [2013].



(a) Read depth 30x
$p_N = 0$



(b) Read depth 75x
$p_N = 0$



(c) Read depth 75x
$p_N = 0.3$

Figure 2.15: Size of the gradient of π_0 with respect to q, plotted against the largest component of q corresponding to pure tumor allele frequency $1/S$, where S is the number of copies in the final state, as usual. The size of the gradient for events with one stage ($K = 1$) fall on a curve because the π vector is one dimensional. For $K > 1$, the size of the gradient can take a range of values, and is thus represented by polygons. The larger the magnitude of the gradient, the more rapidly the value of π changes as q changes. This figure originally appeared as Supplementary Figure 9 in Purdom et al. [2013].

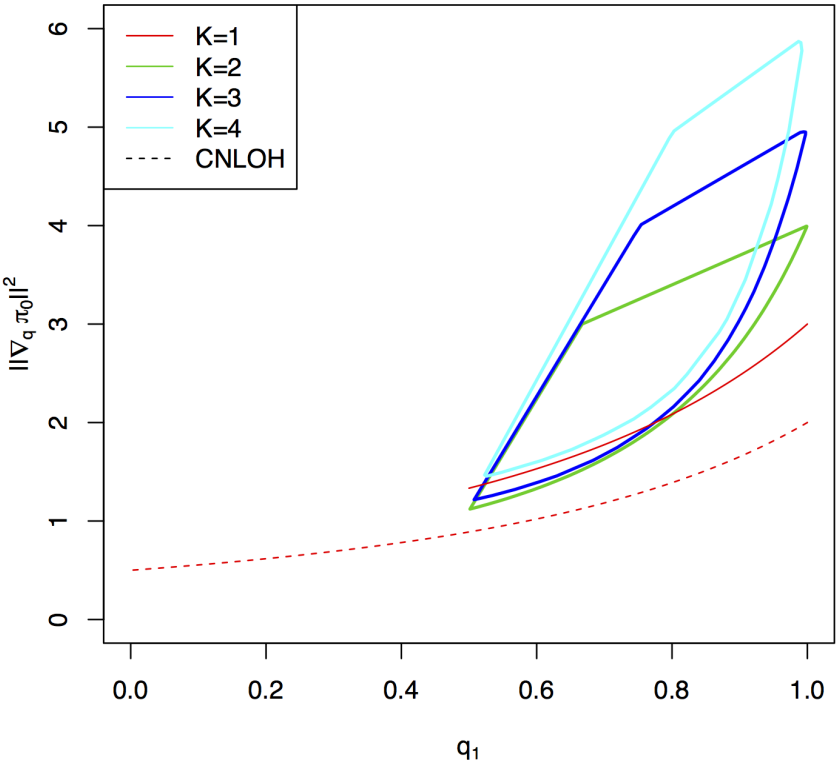
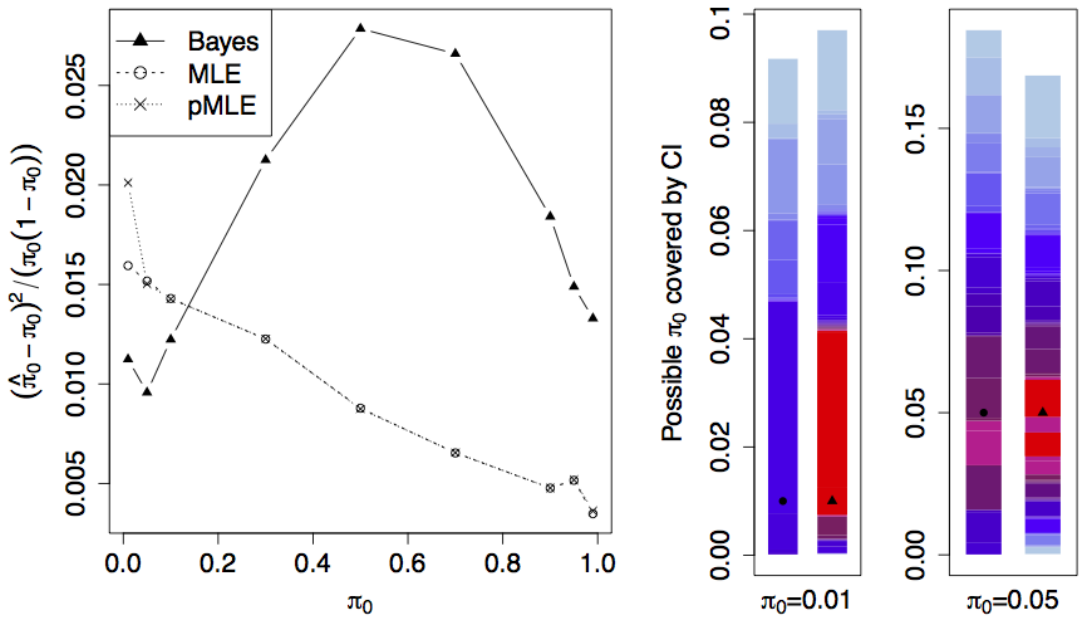
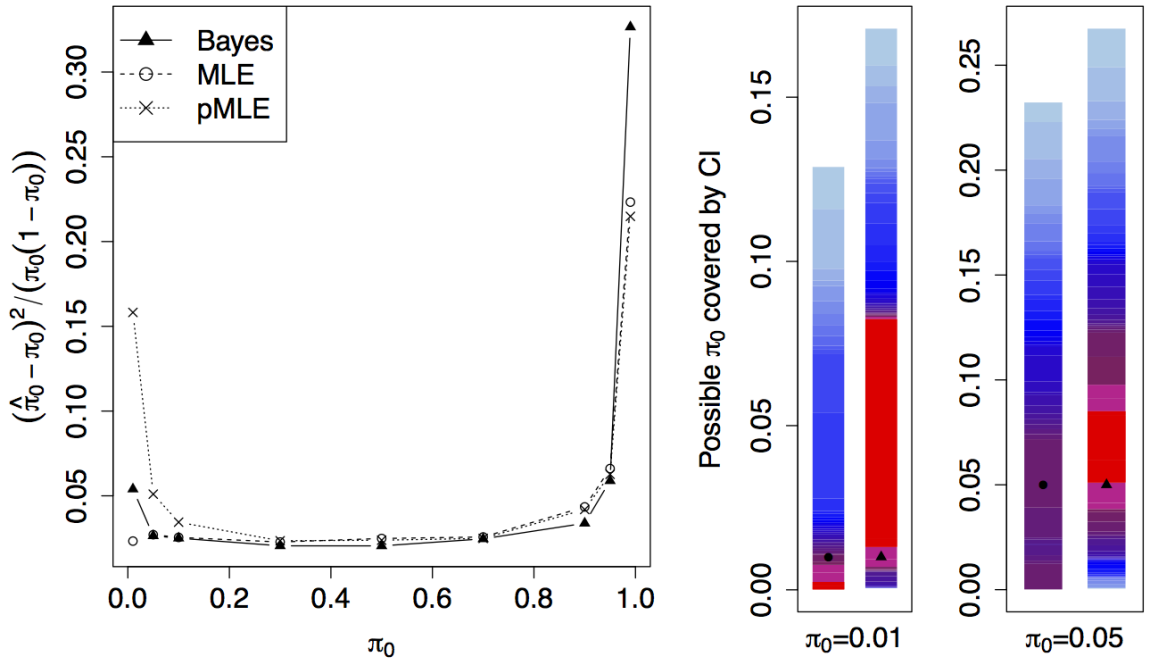


Figure 2.16: Comparison of Bayesian and full MLE estimates for CNLOH. Panel (a) shows the relative MSE for all three methods being compared. Panel (b) compares the bootstrap confidence intervals from the full method against the credible intervals obtained from the Bayesian method for two settings of small π_0 ($\pi_0 = 0.01$ and $\pi_0 = 0.05$), where the Bayesian estimates are not extremely biased. The color scale is the same used in Figure 2.11. The solid points in each bar indicate the true value of π_0. This figure originally appeared as Figure 3 in Purdom et al. [2013].



(a) Relative MSE (b) Coverage of CI

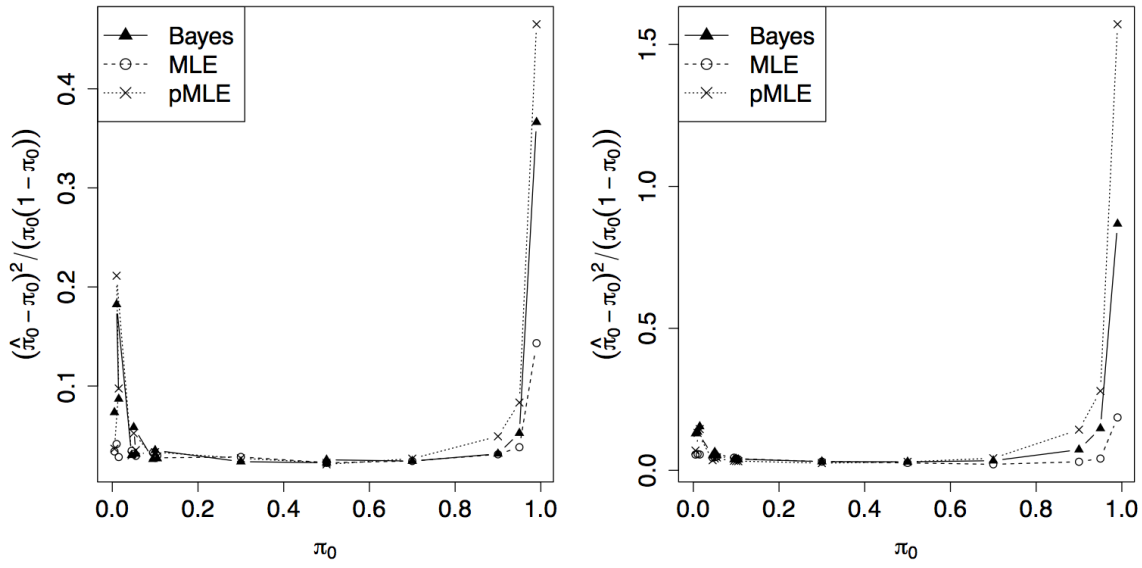
Figure 2.17: Comparison of Bayesian and full MLE estimates for single gain. Panel (a) shows the relative MSE for all three methods being compared. Panel (b) compares the bootstrap confidence intervals from the full method against the credible intervals obtained from the Bayesian method for two settings of small π_0 ($\pi_0 = 0.01$ and $\pi_0 = 0.05$). Additional credible interval plots can be found in Figure 2.19. The color scale is the same used in Figure 2.11. The solid points in each bar indicate the true value of π_0. This figure originally appeared as Supplementary Figure 10 in Purdom et al. [2013].



(a) Relative MSE

(b) Coverage of CI

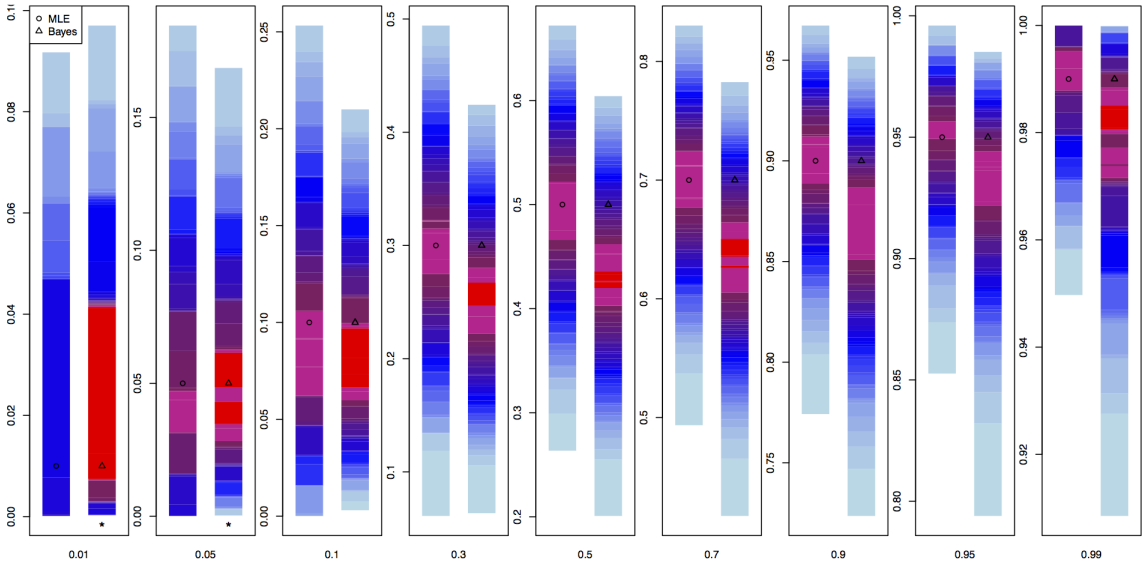
Figure 2.18: Comparison of Bayesian and full MLE estimates for sequential gain in (a) $K = 2$ and (b) $K = 4$ stages. This figure originally appeared as Supplementary Figure 11 in Purdom et al. [2013].



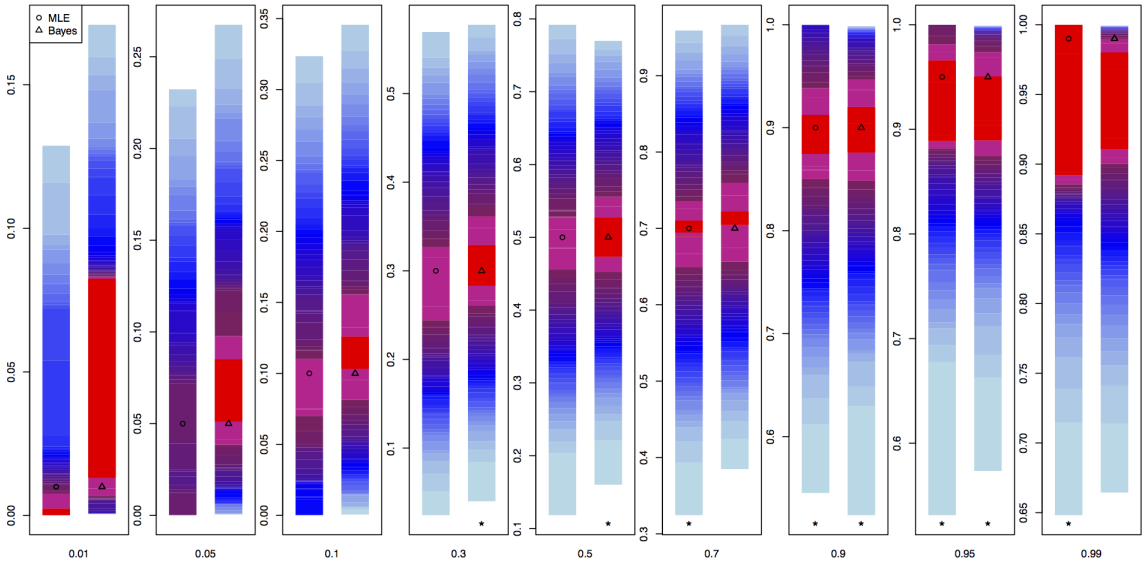
(a) Relative MSE, K=2

(b) Relative MSE, K=4

Figure 2.19: Comparison of Bayesian credible intervals and full MLE confidence intervals for CNLOH (a) and single gain (b). The color scale is the same used in Figure 2.11. The solid points in each bar indicate the true value of π_0. This figure originally appeared as Supplementary Figure 12 in Purdom et al. [2013].



(a) CNLOH



(b) Single Gain

2.5 Application to skin cancer

Primary cutaneous squamous cell carcinomas (cSCC) are among the most common human malignancies, with over 150 new cases out of every 100,000 Caucasians each year [Madan et al., 2010]. These tumors appear on the body and with a demographic proportional to sensitivity to sunlight exposure, and harbor mutations characteristic of UV radiation damage [Ziegler et al., 1994]. Another distinguishing feature of the UV radiation damage characteristic to cancers of the skin is that it usually results in over hundreds of thousands of point mutations across the genome, accumulated over the patient's lifespan. The model of Durinck et al. [2011] was originally developed to take advantage of this staggeringly high mutation density to produce insights into the tumor's history. We describe the application of our model to this cancer type in the sections that follow.

2.5.1 Data collection and preprocessing

For Durinck et al. [2011], eight matched tumor and normal samples from patients with cutaneous squamous cell carcinoma were obtained as part of a skin cancer study at the University of California, San Francisco. Roughly 40 megabases of coding region (exome) were isolated from each sample using oligonucleotide-based hybrid capture and sequenced using the Illumina sequencing-by-synthesis platform.

Point mutation calls were obtained as follows: reads were aligned to a human reference using BWA [Li and Durbin, 2009]. Duplicate reads were removed and read base quality scores recalibrated using *Picard* and *GATK* software, respectively [McKenna et al., 2010]. Mutations were then called from the reads using an early version of *MuTect* [Cibulskis et al., 2013], an algorithm which essentially identifies suspected mutations by looking for loci with sufficient support for non-reference alleles–as a minimum requirement on read coverage, only locations covered by at least 14 reads in the tumor sample and 10 reads in the normal sample were kept. Then, true somatic mutations were filtered from normal genetic variation: candidates that appeared in The Single Nucleotide Polymorphism Database, a.k.a. dbSNP [Sherry et al., 2001] were removed from the data set unless they were also present in COSMIC, a catalogue of somatic mutations in cancer. Although the name suggests only SNPs, the public archive dbSNP catalogs neutral polymorphisms more generally (i.e., not just single nucleotide) for several species (including human). The database is hosted by the NCBI and all of its entries are the result of submissions by members of the research community.

Allele-specific copy number analysis was performed on Affymetrix Genome-Wide Human SNP Array 6.0 chips. Major allele frequencies, minor allele frequencies, and copy number estimates were obtained by processing the tumor and normal pairs of SNP arrays via the CRMA v2 method provided in the R package *aroma.affymetrix*; the resultant allele fractions were normalized using *TumorBoost* [Bengtsson et al., 2008, 2010]. To determine regions of uniform copy number, a circular binary segmentation algorithm was applied using *aroma.affymetrix*, and the results were then checked by manually cross-examining plots of tumor-to-normal intensity ratios, normalized SNP allele B fractions, and mutation allele

frequency from sequencing data. Only unambiguous cases of CNLOH and copy gain events where we agreed with the algorithm results were used.

2.5.2 CNLOH with TP53 knockout occurs before others

In Durinck et al. [2011], we only estimated the temporal ordering of CNLOH events using the MLE method, although the analysis was performed on both CNLOH and single copy gain events using the MLE, partial MLE, and Bayesian methods in Purdom et al. [2013]. The results from Durinck et al. [2011] are shown in Figure 2.20, and the expanded results from Purdom et al. [2013] are shown in Figure 2.21.

Seven of the eight samples analyzed had at least one CNLOH event, and of those, four samples showed TP53 mutation in addition to CNLOH, resulting in double knock-out of the wild-type allele in the locus of the mutation. In total, we found 486 nonsynonymous mutations that were sequenced deeply enough to determine copy number (> 50 independent reads was the threshold used in Durinck et al. [2011]) and that fell at least once in a region of CNLOH. Figure 2.20 shows the estimates of π_0 obtained for each CNLOH event in the eight samples and their corresponding semiparametric bootstrap confidence intervals. CNLOH events that cover the TP53 gene are highlighted in red. For all four samples with TP53 CNLOH, we can observe that the magnitude of $\hat{\pi}_0$ is low, indicating that TP53 CNLOH occurs early in the history of the sample. In addition, for two samples–M01 and V07–we can say with 95% confidence that π_0 for the TP53 CNLOH is smaller than for other events, i.e. that under our model assumptions, the TP53 event was the first CNLOH to occur for that sample.

Based on our understanding of these estimates under simulation, we note that the value of $\hat{\pi}_0$ for M01 TP53 CNLOH could be an underestimate, since we observed under simulations for similarly small values of true π_0, there was a consistent tendency to underestimate unless the mutation counts were very high. This would make it difficult to determine whether the TP53 CNLOH or the Chr 2 event on M01 occurred first. The same could be said for the V07 TP53 CNLOH, but the next event has such as high value of π_0, it's unlikely that there is a different true underlying ordering after accounting for potential bias in the $\hat{\pi}_0$. If the true π_0 value for M45 and M19 is close to the estimate, then they lie within a region of π_0 where the full MLE method tends to produce unbiased estimates.

Figure 2.21 also shows estimates and confidence intervals for $\hat{\pi}_0$, but with the inclusion of single copy gain events. Results from the two other methods discussed–partial MLE and Bayesian–were included for completeness, though of the three methods evaluated under simulation, the full MLE produced the most reliable results across the widest range of π_0 values. Two samples M45 and M19 that were called to have TP53 CNLOH in Durinck et al. [2011] were not included in the figure because they had too few events that could be confidently called, due to the having too few events to begin with, the sample having a high amount of normal contamination, or a combination of both. For the two samples which had both TP53 CNLOH and other CNLOH events to compare against, we were able to identify single gains to add to the ranking: in both samples, TP53 CNLOH could still be called as the

earliest timed chromosomal aberration with 95% confidence. These results held regardless of the method used–MLE, partial MLE, or Bayesian. Figure 2.21 also includes temporal rankings for sample S128, which was sequenced after Durinck et al. [2011] was published but does not have a CNLOH over TP53.

2.5.3 Biological interpretation and significance of the TP53 finding

TP53 is often mutated in precursor lesions, which are growths that have not yet progressed to cancer. Prevailing models of tumor progression propose that inactivation of p53 is a late requirement, and postulate that the loss has a twofold function: it enables unlimited cycles of cell division by overcoming the senescence programs activated by other driver oncogenes, and also enables survival through telomere-crisis, a process characterized by genomic instability through the acquisition of chromosomal rearrangements and mutations [Fearon et al., 1990, Hruban et al., 2000, Chin et al., 1999]. Further, because p53 loss occurs frequently and is more commonly found in invasive disease, typical cross-sectional analyses of mutation frequency by stage usually interpret that it occurs late. However, some experimental evidence appears to contradict the temporal placement of p53 loss as a late event, showing that this mutation actually plays a dominant functional role in the development of phenotypes like tumor formation [Olive et al., 2004, Milner and Medcalf, 1991].

Our findings align with a progression model in which TP53 loss occurs early in tumorigenesis. This does not contradict the findings of cross-sectional mutation frequency by stage studies that place activation of key oncogenes prior to TP53 loss. An alternate model that would result in the same mutation frequencies found in the cross-sectional studies involves a temporal requirement that TP53 mutation precede driver oncogene mutation in precursor lesions that progress to cancer. In this model, precursor lesions that activate oncogenes first do not progress, but would still be detected in mutation frequency by stage surveys.

Extrapolating the average mutation rate observed in CNLOH regions before TP53 loss to the entire exome, we estimate that a patient would have acquired only about 100 total point mutations prior to TP53 loss. This indicates that the genome prior to TP53 loss is remarkably stable and capable of repairing itself. This finding also agrees with the common clinical observation of benign clonal patches of keratinocytes with heterozygous TP53 mutations [Jonason et al., 1996, Ren et al., 1997]: these patches are essentially "waiting" for the trigger of a second p53 allele loss, thereby disrupting the cell's normal repair processes and enabling a mass proliferation of mutations–based on the cSCCs we analyzed, the final mutation rate could reach approximately 50 per megabase or 150,000 per genome. Because DNA repair remains at least partially active, we believe that this vast mutation burden could be explained by the collaborative effects of ongoing DNA damage coupled with disabled DNA damage-induced cell death.

Figure 2.20: Plots of $\hat{\pi}_0$ and their corresponding bootstrap confidence intervals from the full MLE method for eight squamous cell carcinoma samples analyzed in Durinck et al. [2011]. Highlighted in red are CNLOH events that induce double-knockout of a wildtype allele on the TP53 gene, a well-known tumour suppressor gene implicated in several cancers. The number of mutations N for each region is indicated at the top of the plot.

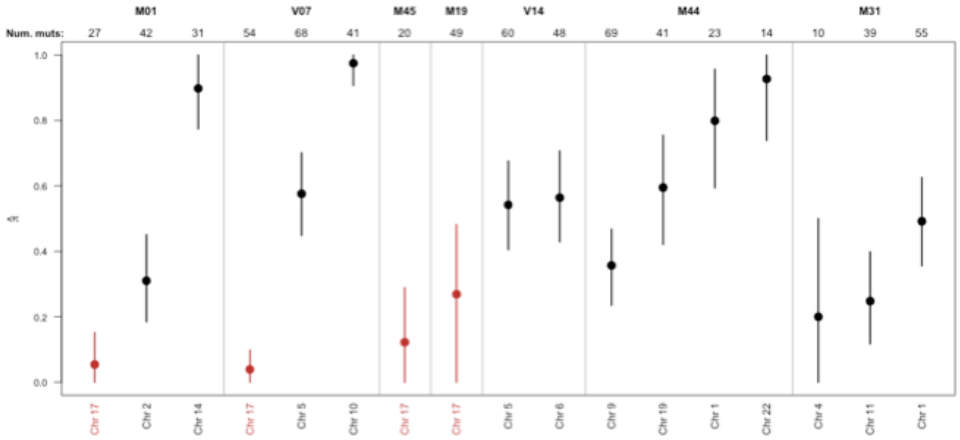
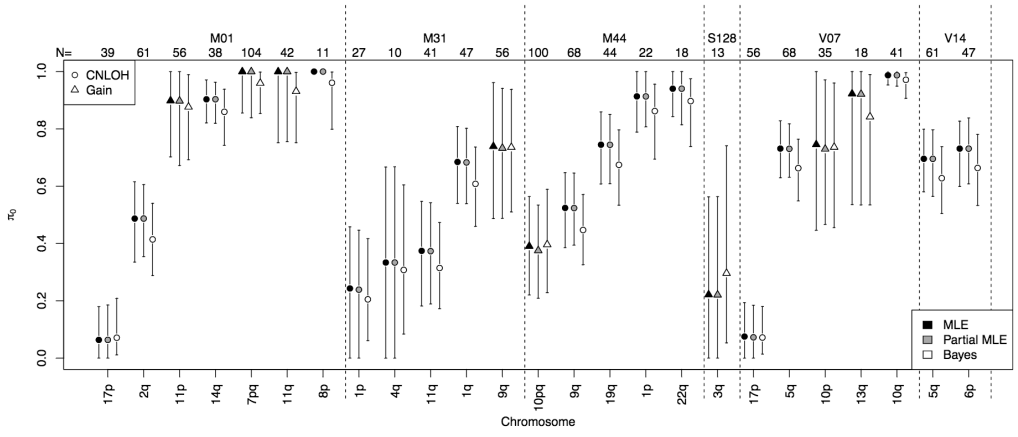


Figure 2.21: Extended version of the results in Figure 2.20, which appeared in Purdom et al. [2013]. The methodology was extended for sequential gain events, which allows several more events to be timed. In addition, two variations on the timing methodology–the partial MLE method of Greenman et al. [2012] and Bayesian estimation–were implemented as well. This figure originally appeared as Supplementary Figure 13 in Purdom et al. [2013].



2.5.4 Checking the constant mutation rate assumption

Our model uses the number of mutations acquired since aberration as a measure of time elapsed; this is only valid when the mutation rate during tumor development is constant. One way we attempted to check this assumption on real data was by comparing the heterozygous mutation rate in regions with and without CNLOH: if the mutation rate were constant, then late CNLOH events should have far less heterozygous mutations than regions without aberration, because those that were acquired before the event would have been knocked out. Likewise, a CNLOH event that occurs early is essentially accumulating mutations at the start of tumor development, and should have a heterozygous mutation rate similar to regions without aberration.

Sample M01 was chosen to look at more deeply because it had a representative mix of CNLOH events–early, medium, and late. Coding regions of the genome were then divided into bins of 0.18 megabases: each bin contains several concatenated exonic regions within one chromosome, and the width of the bin is the sum of the widths of these exonic regions. This corresponded to bins with median genomic width of between 6.1 MB (25th percentile) to 18.4 MB (75th percentile). For each bin, the heterozygous mutation rate, given by the number of mutations divided by the size of the bin, was computed; here size of the bin is taken to mean length of coding sequence, and not the genomic width. The typical number of mutations were bin were between 5 (25th percentile) to 10 (75th percentile).

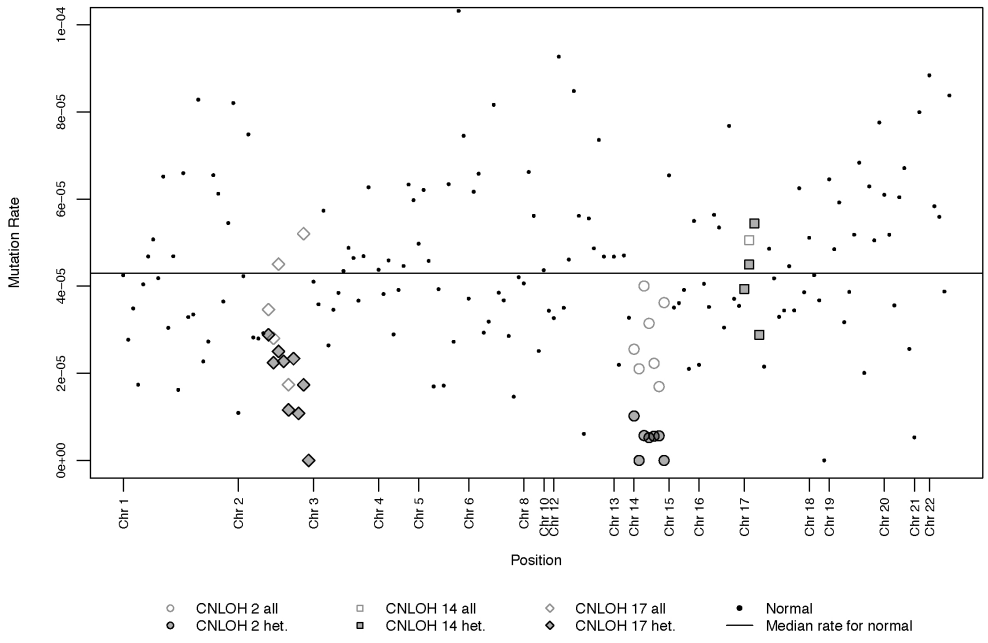
Because mutations in skin cancer differentially affect some nucleotides more than others, we did look into whether we would need to make an adjustment for GC content of the bin. However, we found that with bins of this size, the GC content of the bin was not correlated at all with the mutation rate. Therefore, the mutation-rate computation was not GC-adjusted.

Each bin is represented by a point in Figure 2.22, which shows the mutation rate as a function of position in the genome. Bins that overlap the CNLOH regions identified for M01 are indicated by special plotting characters, and the colors grey and white are used to denote heterozygous vs. total (heterozygous and homozygous) mutation rate. Outside of the CN-LOH events shown, only regions without chromosomal aberration were plotted; hence the lack of points from chromosome 7, 9, or 11, which showed chromosome-wide deletions/gains. Therefore, the x-axis in Figure 2.22 corresponds loosely to the genomic position of the bin, up to omission of aberrant regions and the variability in the genomic width of each bin.

As expected, the total mutation rate is not significantly different from the mutation rate for regions without aberration. However, for the event that had the latest timing, CNLOH on chromosome 14, we see that its rate of heterozygous mutations is visibly lower than what would be expected, whereas its total mutation rate is in line with that of a normal region.

These observations are consistent with what one would expect if there were a constant mutation rate and serve as indirect evidence for that assumption. Directly validating the assumption of a constant mutation rate would require collecting data at multiple time points from the same site on a tumor as it acquires more mutations, which could be impractical for ethical, cost, and patient compliance reasons.

Figure 2.22: Mutation rates in bins of approximately 0.18 Mb of coding sequence without chromosomal aberration are plotted against genomic position for sample M01. Each point corresponds to one bin. The same is plotted for the CNLOH regions of the sample, indicated by different plotting shapes. The color of the shape indicates whether all or just heterozygous mutations were being counted in the mutation rate calculation. The black horizontal line corresponds to the median mutation rate across regions without aberration. This figure originally appeared as Supplementary Figure 3 in Durinck et al. [2011].



2.6 Discussion and conclusions

In this chapter, we presented a method for temporally ordering certain copy number aberrations given allele frequency data from sequencing experiments on cancer. Specifically, we demonstrated that we could temporally order CNLOH events and sequential gain events. We discussed the simulation results of Purdom et al. [2013], where this method was evaluated against that of a similar competing approach by Greenman et al. [2012], which we called the partial MLE. Purdom et al. [2013] also introduced a Bayesian estimation technique to mitigate the instability of $\hat{\pi}_0$ for extreme-valued π_0, and we discussed their simulation results comparing this approach to the original full MLE method.

Overall, it was found that the true value of π and the number of mutations in the region have the largest impact on the ability to estimate π. For practitioners interested in using this method, the takeaway is that it is better to have more mutations at lower read depth rather the converse. However, the number of mutations in a region is out of the control of the practitioner, and is determined by the cancer type under study. We conclude therefore that this method will work best for highly mutated cancers such as skin.

Of the methods compared, the full MLE method produced the most reliable estimates across the widest range of π_0 values, which is typically the component of π of greatest biological interest. The Bayesian estimation technique improved the estimates of CNLOH for small π_0 and confidence interval coverage probabilities in many settings, but overall did not perform better than the full MLE–in fact, on CNLOH, the Bayesian estimates severely underestimated the true value of π_0. Because the partial MLE of Greenman et al. [2012] does not take into account sequencing variability, Purdom et al. [2013] was able to demonstrate through simulation several situations in which this would negative impact the accuracy of their results. Namely, partial MLE fares worse than full MLE as the amount of normal contamination or the complexity of the event increases.

We reviewed the application of the full MLE method to squamous cell carcinoma in Durinck et al. [2011], wherein it was used to show that TP53 double knockout occurred early in tumorigenesis. This was one of a few studies to report this finding, as the common wisdom at the time was that TP53 double knockout was a late event. Further, in a sample that had a mix of early, medium, and late CNLOH events, we were able to find some indirect empirical justification for the assumption of a constant mutation rate. The validity of the temporal ordering model rests upon this assumption.

Since Durinck et al. [2011], the method has been applied to a limited number of other cancer types, due to the fact that cancer type it was original developed for represented a special case where the mutation rate was incredibly high relative to other cancer types. We refer the reader to Purdom et al. [2013] for results from applying the method to sequencing data from whole genome sequencing of ovarian serous cystadenocarcinomas (matched blood and tumor samples) from five patients generated as part of The Cancer Genome Atlas pilot project [Network et al., 2011]. We also refer the reader to Greenman et al. [2012] and Nik-Zainal et al. [2012] where the partial MLE method was applied to breast cancer. Their results include an additional layer of depth since they employed paired-end sequencing, so they were

able to obtain more precise information on the nature of the chromosomal rearrangements that occurred.

Again, the primary limitations to the method being adopted more broadly are principally the need for regions of aberration to have a high mutation rate in order to obtain meaningful confidence intervals, and secondly the restrictions on the kinds of events that we can time. We had attempted to replicate our TP53 analysis on colorectal and lung squamous cell carcinoma, two cancer types with relatively high mutation rates, but found it challenging to find TP53 knockout that co-occurred with CNLOH on the region–in part, this is because it can be difficult to identify the exact nature of the chromosomal aberration from sequencing data alone, especially with normal contamination and few mutations.

To illustrate the mutation rate point more concretely, Figure 2.23, which originally appeared in Lawrence et al. [2013], shows the somatic mutation rate for 27 different cancer types summarized from a data set of over 3,000 matched tumor-normal samples. The plot shows that hematological and pediatric cancers incur the lowest mutation rates, whereas tumors induced by carcinogens like tobacco smoke or UV light have the highest. Across the set of 27 cancer types and even within a single tumor type, the mutation rates can differ by 1000-fold.

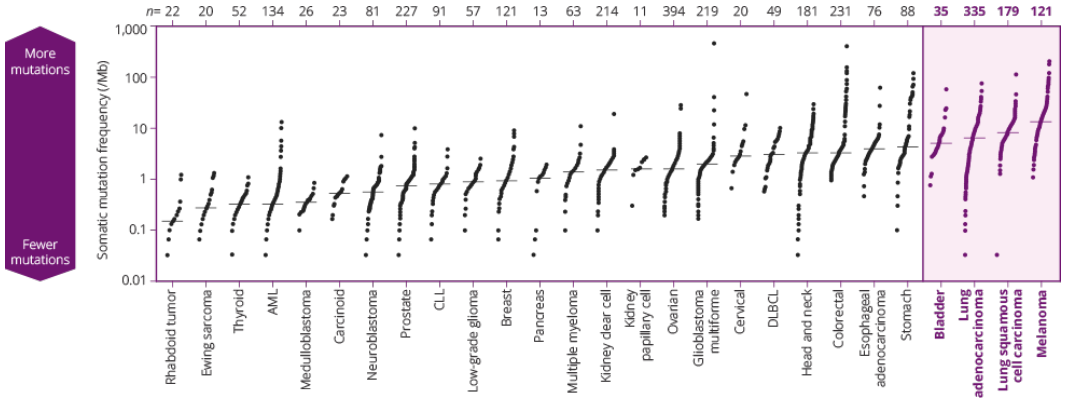
With projects like the TCGA [Network et al., 2011] and the International Cancer Genome Consortium [Zhang et al., 2011] underway, we can expect the amount of cancer genome data available to researchers to grow dramatically. Already, the TCGA implements standardized workflows so that the same type of data is available for each of its samples. In that vein, one can imagine being able to apply this method broadly to all of the cancer types that meet a minimum mutation rate threshold. Part of the challenge in contextualizing findings like those of Durinck et al. [2011] is understanding how they generalize to the population, since many of these cancer studies examine less than 10 samples at a time.

As a result, a future direction for the work presented in this chapter could be developing additional methodology for summarizing the temporal ordering across multiple samples. Because regions of chromosomal aberration can vary greatly in length and have many kinds of mutations, one of the first challenges is to define the granularity of the ranking. If the granularity is small, e.g. timing mutation at the gene-level, then it would be challenging to acquire enough data to draw meaningful conclusions.

Some work has already been done toward using large-scale genomic datasets such as those offered by the TCGA to infer tumor evolutionary pathways. As a recent example, Constantinescu et al. [2015] presents a waiting-time model for inferring mutually exclusive groups of genes which are altered in tumor evolution, which thereby allows inference of mutually exclusive evolutionary pathways. The main idea is that tumor evolution is constrained by partial orders of gene or pathway alterations, and these groups and their dependencies can be inferred jointly from large-scale data sets of mutational profiles.

In another direction, we note that our method has been developed to estimate π in general, though we focused in our applications–both simulated and on real data–on the timing of the first event in a chromosomal aberration, π_0. This was the event of greatest biological interest in Durinck et al. [2011] and Greenman et al. [2012]. However, as in the

Figure 2.23: Somatic mutation rates observed in exomes from 3,083 tumor-normal pairs across 27 cancer types, most of which were sequenced and processed at the Broad Institute. Each dot corresponds to the mutation rate for a single tumor-normal pair. The data originally appeared as Figure 1 in Lawrence et al. [2013].



case of oncogenes, it may not be the case that genes are activated or suppressed after the first stage of a multi-stage event. Therefore, developing a better understanding of the full sequence of events that comprise one aberration could provide a more nuanced look at the relationship between the activity of cancer-implicated genes and copy number.

Chapter 3

Hierarchical multilabel classification with local precision rates

3.1 Introduction

Hierarchical multilabel classification (HMC) refers to the classification problem in which instances can be assigned labels to multiple classes, and these classes follow a tree or directed acyclic graph (DAG) structure. The assignments can also follow multiple paths along the tree or DAG. In computational biology, HMC methods have been used primarily to categorize genes along the Gene Ontology (GO) directed acyclic graph or proteins along the MIPS FunCat rooted tree [Alves et al., 2010, Barutcuoglu et al., 2006, Blockeel et al., 2006, Clare, 2003, Kiritchenko et al., 2005, Valentini, 2009, 2011]. In computer science, HMC methods are commonly used to assign documents or music to categories that follow a hierarchical structure [Rousu et al., 2006, Kiritchenko et al., 2006, Mayne and Perry, 2009]. While functional genomics and text analysis represent the two most popular applications of HMC methods, hierarchical structures are common in other areas of research. For example, Huang et al. [2010] developed a classifier for disease diagnosis based on a patient's microarray gene expression profile: the disease labels were mined from the Unified Medical Language System (UMLS), a biomedical vocabulary organized as a directed acyclic graph.

Silla Jr and Freitas [2011] provide an overview of hierarchical multilabel classification (HMC) methods across several domains. Broadly, HMC methods fall into three categories: flat, local, and global classification. Most of these methods have come from the field of computer science, where local classification methods prevail. We review these three types of HMC methods below.

Flat classifiers perform classification at the leaf node level and force ancestors of positively called leaf nodes to also be positive; as such, flat classifiers do not take advantage of the full data set. Flat classification is equivalent to performing multilabel classification on the set of leaf nodes.

Local classification is a two-stage process: first classifiers are trained for each node or

group of nodes in the graph, then an adjustment is made so that the decisions produced by these classifiers respect the hierarchy. The simplest of these adjustments is a top-down procedure (Koller and Sahami [1997], Wu et al. [2005]): classification is performed sequentially and lower level decisions are only produced if their ancestors are classed as positive. This method suffers from a blocking problem: mistakes made at the top of the graph percolate to all of the descendants, so they can affect the performance of the classifier drastically. Sun and Lim [2001] proposed several other heuristics to eliminate the blocking problem, but none of these are motivated by statistical theory.

A more model-based approach is finding the *maximum a posteriori* classification via a Bayesian adjustment. Barutcuoglu et al. [2006] estimated $P(Q_1, \ldots, Q_p | S_1, \ldots, S_p)$, i.e. the probability of each graph outcome given the classifier scores from training a support vector machine for each node in the hierarchy. This method avoids the blocking issue of the first method and gives an optimal label assignment in some sense. The approach of Rousu et al. [2006] for text analysis involved training an extension of Maximum Margin Markov Network to perform joint multilabel classification, and performing a Bayesian correction via gradient descent to obtain labels consistent with the hierarchy. The primary drawback of these methods is that they do not scale well with the size of the graph and could incur numeric underflow issues because it requires estimating the joint density of the classifier scores for each node in the graph. One can reduce the complexity of estimation by using the generated labels, a binary classifier output, rather than the continuous scores but this has the downside of discarding what information is available in the magnitude of the classifier scores.

Unlike local classifiers, global classifiers jointly make decisions for the graph rather than on a node by node basis. Global classification does not require a second stage because the decisions produced inherently respect the hierarchy. The current state of the art in HMC is a global classifier called predictive clustering trees (PCTs), introduced in Vens et al. [2008]. Like CART, this algorithm finds optimal node splits to produce decisions. A more detailed explanation is given in section 3.4.1. Just as with the Bayesian adjustment for local classification, global classification methods do not scale well with the size of the graph because they require searching over the space of possible decision rules. However, Vens et al. [2008] argued that global classification should be more efficient because they require fewer decision rules overall than local classification methods.

The computational biology and medical imaging fields have begun to adopt global classification methods over local ones, whereas literature from computer science still seems dominated by local classification methods. Although global classifiers theoretically promise better performance and efficiency, in practice they can be difficult to train, and by nature do not scale well. Local classifiers are still used widely because they are flexible and are intuitive to construct.

In the following sections, we extend the work of Jiang et al. [2014] into a local classification method that eliminates the per-classifier tuning step required to produce initial calls before a second-stage adjustment as in Huang et al. [2010]. For the multilabel case without any hierarchy constraint, Jiang et al. [2014] developed methodology that uses the local precision

rate (LPR) to make classification decisions overall rather than node by node. For the HMC problem, we provide an algorithm to sort the LPRs that respects the hierarchical structure. Theoretically, decisions made by the LPR method maximize the expected pooled precision and recall, and the optimal performance of this method was demonstrated in Jiang et al. [2014] on simulated and real data settings. For the HMC problem, simply applying a cutoff to the estimated LPRs could produce calls inconsistent with the hierarchy. In this chapter, we introduce a sorting algorithm that, under certain conditions, can maximize the expected area under the hit curve for early calls, a related but weaker criterion than precision-recall.

Our method of sorting LPRs performs well compared to the global methods in the early part of the precision-recall curve, while retaining the efficiency and flexibility of a local classification method. This method is advantageous in that it can easily accommodate the addition of new nodes in the hierarchy: it is not necessary to retrain the entire classifier, as one would need to do with a global method. We are aware of only one other HMC method based on sorting classifier scores for each node, developed in Bi and Kwok [2011]; we discuss this in greater detail in Section 3.6.

In this next section, we review the LPR and compare methods for estimating it. Then, we introduce the sorting algorithm for LPRs to perform hierarchical multilabel classification. We evaluate its performance on simulated data and the disease diagnosis data of Huang et al. [2010]. In Section 3.6, we draw connections between this work, related methods, and statistical inference. We also highlight open problems for future research.

3.2 The local precision rate for multilabel classification

3.2.1 Problem setting and notation

For consistency, we use the same notation as Jiang et al. [2014]. Assume that classifiers have been learned for K labels connected in an acyclic graph and that there are M instances to be classified. We impose no requirements on class membership outside of being hierarchically consistent: an instance could belong to none of the classes, and those that do belong to a class are not required to have leaf-level membership.

We assume that each label's classifier was trained on $\widetilde{M}$ instances and produces a score $s_{k,m}$ $(m = 1, \dots M,\ k = 1 \dots, K)$ that can be thresholded to produce label assignments: without loss of generality we take larger scores to indicate the positive class, i.e. all instances with $s_{k,m} > \lambda_k$ are said to have label k. For example, if a logistic regression is used for predicting label k, a standard choice for $s_{k,m}$ is the estimated posterior probability that instance m belongs to label k.

Our classification framework begins with the scores for each instance, for which we assume the following generative model. If $Q_{k,m}$ is a binary indicator for whether instance m truly has label k, $Q_{k,m} = 1$ with probability π_k. We require that label membership implies membership in all of its ancestors: $P(Q_{Par(k),m} = 1 | Q_{k,m} = 1) = 1$, where $Par(k)$ is the parent of label k.

Also, we assume conditional independence of labels at the same hierarchical level: if labels k and j share a parent i, $Q_{k,m}$ and $Q_{j,m}$ are independent conditional on the parent status $Q_{i,m}$.

Given a threshold λ_k, the chance that the instance does not belong to the label k is given by $F_k(\lambda_k) = P(s_{k,m} \leq \lambda_k)$, the cumulative distribution function (cdf) for the scores of classifier k. This CDF can be expressed as a mixture of the score distributions for the two classes: $F_k = \pi_k F_{1,k} + (1 - \pi_k)F_{0,k}$, where $F_{1,k}$ is the CDF for those having the label k, and $F_{0,k}$ is the CDF for those without. Analogously, the respective density functions are denoted by $f_{1,k}$ and $f_{0,k}$, and the mixture density by f_k.

3.2.2 Definition and optimality result

Jiang et al. [2014] developed the local precision rate with the intention of maximizing precision with respect to recall in the multilabel setting. Specifically, they maximized an expected population version of the micro-averaged precision and recall rate given by Pillai et al. [2013]. The micro-averaged precision rate has the form $\frac{\sum_k TP_k}{\sum_k TP_k + FP_k}$, where TP_k and FP_k are the number of true and false positives for label k, respectively.

We can write expressions for the expected pooled precision and recall rate as follows. First, we can write the expected precision of the classifier for class k with threshold λ_k as

$$G_k(\lambda_k) = P(Q_{k,\cdot} = 1 | s_{k,\cdot} > \lambda_k) = \frac{\pi_k(1 - F_{1,k}(\lambda_k))}{1 - F_k(\lambda_k)}. \tag{3.1}$$

From rearranging we also have that the joint probability $P(s_{k,\cdot} > s \text{ and } Q_{k,\cdot} = 1)$ is $(1 - F_k(s))G_k(F_k(s))$.

Then, we can pool decisions across all K labels using the thresholds $\lambda_1, \ldots, \lambda_k$ to obtain the expected pooled precision rate (ppr).

$$ppr = \frac{\sum_k (1 - F_k(\lambda_k))G_k(\lambda_k)}{\sum_k 1 - F_k(\lambda_k)} \tag{3.2}$$

The denominator represents the *a priori* expected number of times a given instance will be assigned to a label if the decision thresholds $\lambda_1, \ldots, \lambda_k$ are used. The pooled recall rate (prr) has the same form, except with $\sum_k Q_{k,\cdot}$ as the denominator instead.

Jiang et al. [2014] observed that to maximize the expected pooled precision with respect to pooled recall, it was enough to maximize $\sum_k (1 - F_k(\lambda_k))G_k(\lambda_k)$ while holding $\sum_k 1 - F_k(\lambda_k)$ fixed since $\sum_k Q_{k,\cdot}$ was a constant. The local precision rate (LPR) was then defined as

$$LPR_k(s) = -\frac{d}{dF_k(s)}\{(1 - F_k(s))G_k(s)\} = G_k(s) - (1 - F_k(s))\frac{dG_k(s)}{dF_k(s)} \tag{3.3}$$

In their main theoretical result Theorem 2.1, they showed that if the LPR for each class is monotonic, then ranking the KM LPRs calculated for each instance/class combination and thresholding the result produces a classification that maximizes the expected pooled

precision with respect to a fixed recall rate. The monotonicity requirement is equivalent to having monotonicity in the likelihood of the positive class, and it is satisfied when higher classifier scores correspond to a greater likelihood of being from the positive class–this rules out poorly behaved classifiers, for example a multimodal case where the positive class scores lie in the range $[0, 0.3) \cup (0.7, 1]$, and the negative class scores in $[0.3, 0.7]$.

3.2.3 Connection to local true discovery rate

After substituting expressions for the derivatives $\frac{dG_k(s)}{dF_k(s)} = \frac{dG_k(s)}{ds}\frac{ds}{dF_k(s)}$, the LPR can be shown to be equivalent to the local true discovery rate, $l$tdr.

$$LPR_k(s) = G_k(s) - (1 - F_k(s))\frac{dG_k(s)}{dF_k(s)} \tag{3.4}$$

$$= G_k(s) - (1 - F_k(s))\left[\frac{\pi_k f_{1,k}(s)}{(1 - F_k(s))f_k(s)} + \frac{\pi_k(1 - F_{1,k}(s))}{(1 - F_k(s))^2}\right] \tag{3.5}$$

$$= G_k(s) - \frac{\pi_k f_{1,k}(s)}{f_k(s)} - G_k(s) \tag{3.6}$$

$$= \frac{\pi_k f_{1,k}(s)}{f_k(s)} = l\text{tdr} \tag{3.7}$$

The local false discovery rate, $l$fdr $= 1 - l$tdr is its more well known relative; it has been studied extensively for Bayesian large-scale inference. This connection between a statistic used for hypothesis testing and the LPR, which was developed for classification, suggests the possibility that methodological developments on the LPR in classification could have meaningful implications for statistical inference. We elaborate on this connection in Section 3.6.

3.2.4 Methods for estimating LPR

The optimality result in Jiang et al. [2014] was derived using true LPR values, which are generally unknown in practice. The authors discussed two methods for estimating the LPR. In the first method, estimates for $f_{0,k}$, f_k, and π_k are plugged in after expressing $LPR_k(s)$ as the local true discovery rate. In the second method, a local quadratic kernel smoother is used to simultaneously estimate $G_k(s)$ and $G'_k(s)$ in the definition of LPR.

Theoretically, Jiang et al. [2014] showed that under certain conditions, the first method converges to the true result faster than the second. However, in simulation studies the second method performed better than the first. The difference is due to the difficulty in estimating the densities $f_{0,k}$ and f_k on real data: any situation which would make kernel density estimation difficult would result in poor estimates of $l$tdr. For example, if the data are observed densely in one or two short intervals and sparsely elsewhere, the kernel density estimate of f_k would have bumps in the sparse regions, making $l$tdr unreliable. Further,

because $f_{0,k}$ and f_k are estimated separately, they have different levels of bias and variance; in particular $f_{0,k}$ has larger variance (since it is only estimated from the negative class cases. In comparison, the functions $G_k(u)$ and $G'_k(u)$ are estimated jointly in the second method and $G_k(u)$ is always densely observed, as its domain is score percentiles rather than the scores themselves.

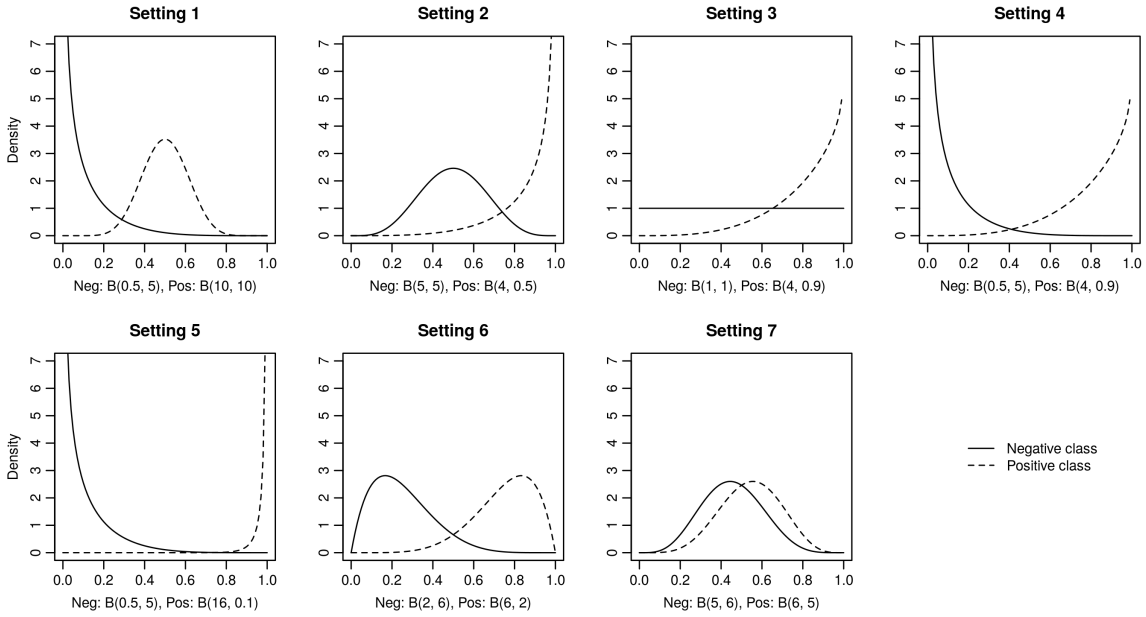
For both estimation methods, it is possible to obtain LPR estimates that are negative. Users must adopt a heuristic to handle these cases.

Lee [2013] suggested an alternative based on the second method that averages the estimates obtained via a weighted spline fit on bagged samples of the data. The addition of weights and bagging were introduced in order to estimate the precision function more robustly in regions supported by less data.

3.2.5 Comparison of estimation methods using simulated data

Although Lee [2013] used this estimation method on real data, he did not compare his method against that of Jiang et al. [2014] in a controlled setting. We examined the performance of the spline method of Lee [2013] and the second method of Jiang et al. [2014] under simulation.

Figure 3.1: Class distributions used in simulations comparing the LPR estimation methods of Jiang et al. [2014] and Lee [2013].



The same five combinations of positive/negative class distributions as the simulations in Jiang et al. [2014] were used, with two additional settings. All of the data were generated from Beta distributions. Figure 3.1 shows the parameters used for each simulation setting. These settings were chosen to represent a wide range of score distributions from classifiers, however

they all satisfy the monotonicity criterion given in Jiang et al. [2014] so that the asymptotc normality of the LPR holds and can be estimated well. This criterion essentially states that the likelihood of the positive class $f_{1,k}(s)/f_{0,k}(s)$ must increase and exceed 1 as s tends to 1. Settings 1 and 2 represent situations where either the positive or negative class score distribution is not concentrated around 0 or 1 as one would expect for posterior probabilities. Setting 3 explores the case when the classifier is able to detect the positive class reasonably well, but is no better than guessing at random on the negative class (uniform distribution). Settings 4 and 5 represent common score distributions for well trained classifiers, where the scores for the two classes are separated well–setting 5 has greater separation than setting 4. Settings 6 and 7 did not appear in Jiang et al. [2014] and are variations of settings 4 and 5 for when the classifiers are not as well trained.

For the second method of Jiang et al. [2014], quadratic polynomial smoothing was done with the R package `locpoly`. LPR estimates that fell outside of the range $[0, 1]$ were clipped.

Although bagging was proposed in Lee [2013] for the spline method, we also tried it with the kernel method. In particular, B bags of size M, i.e. the same size as the training sample, were taken and a spline or kernel smoother was fit to each bag. In our usage, the smoothing parameter for the spline fit was estimated via 5-fold cross validation; however, Lee [2013] modified the 5-fold cross validation procedure to produce larger smoothing parameters, which he found performed better in practice. Following Lee [2013], the weights in the spline fit were inversely proportional to the amount of data available to estimate $G_k(s)$: for example, the weight used at λ is $|\{s_{m,k} \geq \lambda\}|^{-1}$. Then, to estimate a new data point, the average of the B predictions from the bagged spline fits was taken.

Overall, we evaluated four methods: the kernel method, the kernel method with bagging, the spline method, and the spline method with bagging. In our simulations, we followed Lee [2013] and took 100 bags of the training sample.

As in Jiang et al. [2014], we examined performance with training sample sizes $n = 100$, 200, and 500 and positive class frequencies $\pi = 0.05$, 0.10, and 0.20. For the training set, π is exact: for example, when $n = 100$ and $\pi = 0.05$, exactly 5 positive cases and 95 negative cases were generated as a training sample. Each test set had $n = 1500$ cases and the number of positive instances was randomly generated from $\text{Binom}(1500, \pi)$. The training set's number of positive cases were held fixed to ensure consistency in training setting from simulation to simulation, whereas the test set's positive instances were allowed to vary to mimic a real life setting. The test set's size was set to be large so that an accurate estimate of our performance metric, the area under the precision-recall curve (AUPRC), could be obtained. Each simulation setting was replicated 25 times and the average and SD AUPRC were computed. These results are given in Tables 3.1 to 3.7.

Unsurprisingly, the AUPRC improves with an increase in either the number of positive cases or the total training set size. The quadratic kernel smoother consistently outperforms the spline method though this difference narrows as n increases for all simulation settings. Bagging modestly improves performance for both methods, but the gain is not significant and might not be worth the additional computing resources. At the best estimation setting tried ($n = 500$ and $\pi = 0.20$), the average gap in performance between kernel and spline

was largest for settings 3, 7, 1, 6, 2, 4, and finally 5. This roughly aligns with the degree of overlap between the two class distributions: if two classes are separated well, then there is little difference in performance between the two methods. In particular, both methods achieved nearly perfect classification on setting 5, which corresponds to the setting with the greatest degree of separation between the two classes; this was true even in the case of the least available data ($n = 100, \pi = 0.05$). On the other hand, the settings with the greatest overlap between the two classes–settings 3 and 7–also had the worst AUPRCs: at best, the AUPRC was approximately 0.45 for setting 3 and 0.35 for setting 7. This is low relative to the AUPRCs of around 0.9 or higher seen for the other settings.

Based on these results, we would recommend using the kernel method of Jiang et al. [2014] for estimating LPR on smaller data sets, and leave bagging as optional depending on the computing resources available. If the training sample size is large, however, there will be little difference in performance between the kernel and spline methods. At large enough training sample sizes, this becomes a meaningful difference: for example, the implementation of spline estimation in R (built-in with the `stats` package) is much faster than the kernel smoother (`locpoly` package). In that case, using the spline method without bagging may be the more appropriate choice.

Table 3.1: Estimation method comparison for setting 1: scores from the negative class follow a B(0.5, 5) distribution, and the positive class a B(10, 10) distribution.

n	π	Kernel		Spline	
		No bagging	With bagging	No bagging	With bagging
100	0.05	0.648 (0.052)	0.662 (0.053)	0.552 (0.099)	0.606 (0.090)
	0.10	0.798 (0.058)	0.803 (0.050)	0.715 (0.068)	0.750 (0.083)
	0.20	0.904 (0.023)	0.908 (0.015)	0.859 (0.043)	0.868 (0.044)
200	0.05	0.668 (0.066)	0.675 (0.062)	0.561 (0.117)	0.560 (0.093)
	0.10	0.816 (0.036)	0.821 (0.046)	0.740 (0.060)	0.768 (0.040)
	0.20	0.893 (0.026)	0.913 (0.020)	0.844 (0.034)	0.872 (0.025)
500	0.05	0.675 (0.063)	0.697 (0.052)	0.606 (0.078)	0.635 (0.074)
	0.10	0.813 (0.035)	0.821 (0.037)	0.765 (0.039)	0.787 (0.035)
	0.20	0.903 (0.022)	0.913 (0.014)	0.869 (0.025)	0.878 (0.021)

Table 3.2: Estimation method comparison for setting 2: scores from the negative class follow a B(5, 5) distribution, and the positive class a B(4, 0.5) distribution.

n	π	Kernel		Spline	
		No bagging	With bagging	No bagging	With bagging
100	0.05	0.817 (0.033)	0.829 (0.047)	0.753 (0.053)	0.768 (0.073)
	0.10	0.871 (0.024)	0.890 (0.022)	0.798 (0.070)	0.829 (0.065)
	0.20	0.913 (0.012)	0.918 (0.012)	0.843 (0.056)	0.870 (0.059)
200	0.05	0.842 (0.038)	0.851 (0.031)	0.786 (0.048)	0.799 (0.054)
	0.10	0.885 (0.021)	0.876 (0.016)	0.816 (0.050)	0.833 (0.051)
	0.20	0.917 (0.012)	0.921 (0.015)	0.861 (0.035)	0.892 (0.022)
500	0.05	0.838 (0.038)	0.841 (0.039)	0.792 (0.054)	0.810 (0.045)
	0.10	0.890 (0.025)	0.878 (0.021)	0.856 (0.033)	0.839 (0.032)
	0.20	0.916 (0.014)	0.919 (0.010)	0.883 (0.015)	0.898 (0.014)

Table 3.3: Estimation method comparison for setting 3: scores from the negative class follow a B(1, 1) distribution, and the positive class a B(4, 0.9) distribution.

n	π	Kernel		Spline	
		No bagging	With bagging	No bagging	With bagging
100	0.05	0.135 (0.031)	0.127 (0.042)	0.099 (0.031)	0.093 (0.031)
	0.10	0.245 (0.046)	0.238 (0.049)	0.201 (0.035)	0.190 (0.036)
	0.20	0.430 (0.047)	0.438 (0.044)	0.358 (0.051)	0.380 (0.048)
200	0.05	0.129 (0.036)	0.159 (0.033)	0.100 (0.038)	0.116 (0.027)
	0.10	0.249 (0.044)	0.268 (0.034)	0.197 (0.035)	0.211 (0.034)
	0.20	0.452 (0.037)	0.446 (0.042)	0.369 (0.040)	0.385 (0.038)
500	0.05	0.153 (0.032)	0.152 (0.033)	0.105 (0.032)	0.105 (0.025)
	0.10	0.278 (0.029)	0.279 (0.024)	0.220 (0.025)	0.232 (0.026)
	0.20	0.466 (0.030)	0.449 (0.036)	0.395 (0.035)	0.390 (0.031)

Table 3.4: Estimation method comparison for setting 4: scores from the negative class follow a B(0.5, 5) distribution, and the positive class a B(4, 0.9) distribution.

n	π	Kernel		Spline	
		No bagging	With bagging	No bagging	With bagging
100	0.05	0.963 (0.017)	0.955 (0.023)	0.935 (0.033)	0.898 (0.087)
	0.10	0.980 (0.007)	0.976 (0.013)	0.933 (0.042)	0.937 (0.046)
	0.20	0.991 (0.002)	0.990 (0.003)	0.965 (0.019)	0.968 (0.025)
200	0.05	0.967 (0.014)	0.968 (0.010)	0.929 (0.049)	0.945 (0.025)
	0.10	0.982 (0.007)	0.983 (0.005)	0.950 (0.029)	0.961 (0.020)
	0.20	0.992 (0.003)	0.991 (0.003)	0.964 (0.018)	0.964 (0.025)
500	0.05	0.972 (0.013)	0.968 (0.013)	0.942 (0.036)	0.946 (0.028)
	0.10	0.984 (0.005)	0.983 (0.005)	0.960 (0.020)	0.960 (0.020)
	0.20	0.991 (0.003)	0.992 (0.002)	0.980 (0.007)	0.975 (0.013)

Table 3.5: Estimation method comparison for setting 5: scores from the negative class follow a B(0.5, 5) distribution, and the positive class a B(16, 0.1) distribution.

n	π	Kernel		Spline	
		No bagging	With bagging	No bagging	With bagging
100	0.05	0.989 (0.014)	0.983 (0.027)	0.989 (0.014)	0.983 (0.027)
	0.10	0.994 (0.007)	0.996 (0.005)	0.994 (0.007)	0.996 (0.005)
	0.20	0.998 (0.003)	0.998 (0.003)	0.998 (0.003)	0.998 (0.003)
200	0.05	0.997 (0.003)	0.996 (0.006)	0.997 (0.003)	0.996 (0.006)
	0.10	0.999 (0.001)	0.999 (0.001)	0.999 (0.001)	0.999 (0.001)
	0.20	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)
500	0.05	0.999 (0.002)	0.999 (0.003)	0.999 (0.002)	0.999 (0.003)
	0.10	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)
	0.20	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)

Table 3.6: Estimation method comparison for setting 6: scores from the negative class follow a B(2, 6) distribution, and the positive class a B(6, 2) distribution.

n	π	Kernel		Spline	
		No bagging	With bagging	No bagging	With bagging
100	0.05	0.852 (0.049)	0.850 (0.048)	0.765 (0.095)	0.799 (0.087)
	0.10	0.909 (0.019)	0.915 (0.020)	0.849 (0.066)	0.874 (0.050)
	0.20	0.951 (0.011)	0.950 (0.010)	0.898 (0.045)	0.899 (0.039)
200	0.05	0.870 (0.032)	0.874 (0.031)	0.788 (0.085)	0.829 (0.048)
	0.10	0.914 (0.015)	0.917 (0.014)	0.830 (0.057)	0.864 (0.044)
	0.20	0.952 (0.011)	0.950 (0.009)	0.896 (0.035)	0.912 (0.024)
500	0.05	0.875 (0.029)	0.882 (0.021)	0.824 (0.039)	0.840 (0.030)
	0.10	0.921 (0.014)	0.921 (0.015)	0.882 (0.019)	0.888 (0.028)
	0.20	0.952 (0.009)	0.952 (0.008)	0.914 (0.023)	0.926 (0.015)

Table 3.7: Estimation method comparison for setting 7: scores from the negative class follow a B(5, 6) distribution, and the positive class a B(6, 5) distribution.

n	π	Kernel		Spline	
		No bagging	With bagging	No bagging	With bagging
100	0.05	0.080 (0.025)	0.075 (0.022)	0.068 (0.021)	0.060 (0.015)
	0.10	0.155 (0.044)	0.160 (0.027)	0.126 (0.040)	0.136 (0.025)
	0.20	0.308 (0.048)	0.300 (0.041)	0.267 (0.040)	0.258 (0.037)
200	0.05	0.089 (0.027)	0.077 (0.021)	0.063 (0.014)	0.063 (0.015)
	0.10	0.169 (0.035)	0.173 (0.032)	0.131 (0.022)	0.143 (0.030)
	0.20	0.316 (0.030)	0.320 (0.037)	0.256 (0.030)	0.269 (0.029)
500	0.05	0.104 (0.028)	0.093 (0.026)	0.075 (0.022)	0.069 (0.016)
	0.10	0.197 (0.028)	0.191 (0.029)	0.149 (0.030)	0.144 (0.021)
	0.20	0.336 (0.022)	0.330 (0.026)	0.273 (0.026)	0.274 (0.029)

3.3 Extension for hierarchical multilabel classification

The method of Jiang et al. [2014] is appropriate for multilabel classification when the labels do not have a hierarchical structure. In the hierarchical case, it is possible for the estimated LPR of a label to be greater than that of its parent label, since it only depends on each individual label classifier score. Regular sorting of LPRs can result in an instance being assigned to a label but not its ancestors, violating the hierarchy constraint.

We propose a sorting algorithm based on maximizing the expected area under the hit curve, a close relative to the ROC and precision-recall curves. The resultant sorting respects the hierarchy constraints, and the users can threshold the list to produce label assignments, e.g. taking the top k to be positive calls.

3.3.1 HierLPR

3.3.1.1 The hit curve

The hit curve plots the number of true positives against the number of positive calls. Figure 3.2 shows an example hit curve with 100 total instances, 50 of which are positive. A perfect decision rule would label all 50 positive instances as positive, then the remainder as negatives. The hit curve corresponding to the perfect decision rule would thus rise as $y = x$ until $x = 50$, then stay at $y = 50$ as there are no more positive instances left. Any hit curve is bounded by this ideal case. Precisely, if there are a total of n_{pos} positive cases, the area under the hit curve will always fall between $\frac{n_{pos}^2}{2}$ and $\frac{n_{pos}^2}{2} + (n - n_{pos})^2$. The lower bound is obtained with the worst case decision rule, which calls all of the negative instances first. For example, the simulated curve depicts a more realistic scenario, wherein the 21st to 25th positive calls were in truth negatives; thus, the hit curve stays level at 20 true positives up until $x = 25$.

The connection between the hit curve and the ROC curve and precision-recall curve is explained in detail in Su et al. [2013]. Hit curves have been explored in the information retrieval community as a useful alternative to these two other evaluation tools, particularly in situations where there is low positive class prevalence and the user is more interested in the first classified instances or the initial part of the curve–for example, this occurs in search engine page ranking where the top matches are the most important and the number of relevant pages is tiny relative to the size of the World Wide Web. The major flaw of the ROC curve in these situations is that its shape is prevalence independent [Davis and Goadrich, 2006, Hand, 2009]. The precision-recall curve accounts for prevalence, but Herskovic et al. [2007] provided a simple example where the shape of the hit curve was more informative: with only five positive cases out of 1000, the hit curve's shape clearly highlighted the call order of a method that had labeled 100 instances before the 5 true positives, whereas the corresponding precision-recall curve was flat and uninformative. As an example of their use in an applied setting, Bernstam et al. [2006] plotted hit curves to evaluate the effectiveness of different MEDLINE search algorithms.

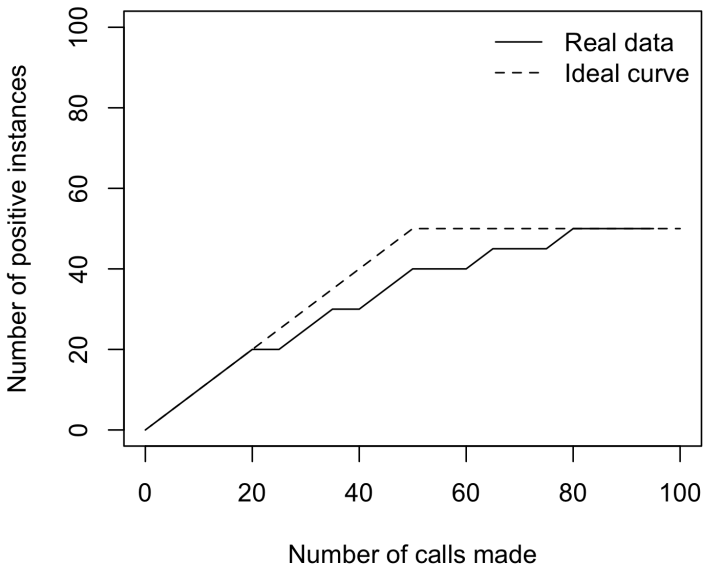


Figure 3.2: The dashed line shows a simulated hit curve, and the solid line the ideal hit curve. The ideal hit curve will follow the $y = x$ line until it reaches the total number of positive instances, and level off thereafter. This is equivalent to a decision rule that correctly calls all of the positive instances first, then all of the negative instances.

3.3.1.2 Maximizing the expected area under the hit curve

We now provide an algorithm for maximizing an objective function that is equivalent to the expected area under the hit curve under certain conditions. First, we derive the objective function.

Suppose the classifier scores are sorted so that $s_{(1)}, \ldots, s_{(KM)}$ represents the order in which they are called positive, and $Q_{(1)}, \ldots, Q_{(KM)}$ their true labels. Through a straightforward application of the trapezoid rule on a stepwise function, we obtain that the sum of height at each unit increment in the x-axis is the exact area under the hit curve up to a constant. The constant is the addition of the fixed unknown amount $P/2$ to the exact area, where P represents the true number of positive instances; this was done to arrive at the convenient form of the objective function below. Since the x-axis represents the number of calls made, the expression for area is equivalent to the sum of the number of true positives among the top k calls, for every k. To indicate the kth node as a true positive, we write $I\{Q_{(k)} = 1|s_1, \ldots, s_n\}$ since the sorting of the nodes depends on the value the classifier scores. This yields the convenient expression

$$\sum_{i=1}^{n}\sum_{k=1}^{i} I\{Q_{(k)} = 1|s_1, s_2, \ldots, s_n\} = \sum_{i=1}^{n}(n-i+1)I\{Q_{(i)} = 1|s_1, s_2, \ldots, s_n\} \tag{3.8}$$

Taking expected values, we arrive at

$$\sum_{i=1}^{n}(n-i+1)P(Q_{(i)} = 1|s_1, s_2, \ldots, s_n) \approx \sum_{i=1}^{n}(n-i+1)P(Q_{(i)} = 1|s_{(i)}) = \sum_{i=1}^{n}(n-i+1)LPR_{(i)} \tag{3.9}$$

The simplifying approximation relates the area under the hit curve to the local precision rate, but it implies that the nodes are independent, which is clearly violated with a hierarchy. However, this approximation will hold for top- or near top-level nodes with well trained classifiers. In practice, this situation occurs commonly: since the top-level nodes represent more general classes, they also tend to have well trained classifiers.

Initialize:
for *i in* L_{L-1} **do**
  Create supernodes from $Sub(i)$ with node i as the first in the list, then the remaining nodes sorted by decreasing LPR
end
Set *current_level* to L-2
while *current_level is not root* **do**
  for *i in* $L_{current\_level}$ **do**
    Create a supernode from $Sub(i)$ as follows.
    Place node i at the top (due to hierarchy constraint.) Set $nodes\_remaining = Sub(i)$
    while $|nodes\_remaining| > 0$ **do**
      Evaluate all LPR averages along each supernode from the step with $current\_level - 1$ that has node i as parent, i.e. within each supernode store the averages of the first value, first two values, etc.
      The supernode corresponding to the highest average is placed next in the ordering of $Sub(i)$.
      Remove the supernode from *nodes_remaining*.
    end
  end
end

Algorithm 1: The HierLPR algorithm for a single instance on tree-structured data.

Algorithm 1 provides a sorting of a single instance along a tree that maximizes the objective function given. We define a supernode here to mean an ordered set of nodes. This algorithm extends easily for multiple instances by assembling the supernodes discovered in the final merge step for each instance in decreasing mean LPR value.

We introduce some additional notation: Assume the tree has L levels. Let the nodes in the subtree with root i be given by the set $Sub(i)$, i.e. all nodes with i as an ancestor. The

algorithm starts from the bottom of the tree, ordering each subtree $Sub(i)$ for $i \in L_{L-1}$, the second to last level. This is the initialization point rather than the last level, since the last level consists only of leaf nodes.

3.3.1.3 Proof of optimality

We now demonstrate that the merging step in the algorithm above, where ordered subtrees are collated to sort a larger subtree, produces a sorting that maximizes the objective function while preserving the order of the original subtrees.

Base case. The simplest nontrivial case is a graph with two supernodes of size one, $X_{(1)}$ and $Y_{(1)}$. In this case, it is clear that the objective function is maximized when the two are sorted in decreasing order, and that the algorithm also produces this result.

Inductive step. Suppose there are two complete supernodes, one consisting of the nodes $X_{(1)}, X_{(2)}, \ldots, X_{(n)}$ and the other $Y_{(1)}, Y_{(2)}, \ldots, Y_{(m)}$. We want to merge these supernodes into a larger supernode, $Z_{(1)}, \ldots, Z_{(m+n)}$ ordering them so that the objective function $\sum_{i=1}^{m+n}(m+n-i+1)Z_{(i)}$ is maximized and the order of the nodes within the original two supernodes is preserved, i.e. $X_{(1)}$ still appears before $X_{(2)}$ after merging.

Suppose that $X_{(1)}, \ldots, X_{(k)}$ is the maximal supernode by the proposed algorithm, and let a denote the average of these k values. We show that the value of the objective function for a list where $X_{(1)}, \ldots, X_{(k)}$ occupies the top k positions is greater than that of a list where this is not the case. We will refer to this sorting with $X_{(1)}, \ldots, X_{(k)}$ at the top as the proposed sorting.

Consider an arbitrary sorting respecting the order constraint of the original supernodes. Let i_1 denote the position of $X_{(1)}$ in this list, i_2, $X_{(2)}$, and so on. Note that $i_{c+1} \geq i_c + 1$. Then, the difference in the value of the objective function between the proposed and arbitrary sorting can be written as follows:

$$\begin{aligned}
\text{proposed OF} - \text{arbitrary OF} &= \left[(i_1-1)X_{(1)} - \sum_{k=1}^{i_1-1} Y_{(k)}\right] + \ldots + \left[(i_n-n)X_{(n)} - \sum_{k=1}^{i_n-n} Y_{(k)}\right] \\
&= (i_1-1)\left[X_{(1)} - \frac{1}{i_1-1}\sum_{k=1}^{i_1-1} Y_{(k)}\right] + \ldots + (i_n-n)\left[X_{(n)} - \frac{1}{i_n-n}\sum_{k=1}^{i_n-n} Y_{(k)}\right] \\
&= (i_1-1)\left[\big(X_{(1)}-a\big) + \left(a - \frac{1}{i_1-1}\sum_{k=1}^{i_1-1} Y_{(k)}\right)\right] + \ldots \\
&\qquad + (i_n-n)\left[\big(X_{(n)}-a\big) + \left(a - \frac{1}{i_n-n}\sum_{k=1}^{i_n-n} Y_{(k)}\right)\right] \\
&= \big[(i_1-1)\big(X_{(1)}-a\big) + \ldots + (i_n-n)\big(X_{(n)}-a\big)\big] \\
&\qquad + \left[(i_1-1)\left(a - \frac{1}{i_1-1}\sum_{k=1}^{i_1-1} Y_{(k)}\right) + (i_n-n)\left(a - \frac{1}{i_n-n}\sum_{k=1}^{i_n-n} Y_{(k)}\right)\right]
\end{aligned}$$

Because $X_{(1)}, \ldots, X_{(k)}$ is the maximal supernode, each term in the expression

$$(i_1-1)\left(a - \frac{1}{i_1-1}\sum_{k=1}^{i_1-1} Y_{(k)}\right) + \ldots + (i_n-n)\left(a - \frac{1}{i_n-n}\sum_{k=1}^{i_n-n} Y_{(k)}\right) \tag{3.10}$$

must be nonnegative, and equality results only if there is a tie.

The first term on the right hand side must also be nonnegative. To see this, we can rewrite the sum

$$(i_1-1)\big(X_{(1)}-a\big) + \ldots + (i_n-n)\big(X_{(n)}-a\big) \tag{3.11}$$

as follows

$$(i_1-1)\sum_{k=1}^{n}(X_{(k)}-a) \;+\; (i_2-i_1-1)\sum_{k=2}^{n}\big(X_{(k)}-a\big) + \ldots + (i_n-i_{n-1}-1)(X_{(n)}-a). \tag{3.12}$$

The first sum $\sum_{k=1}^{n}(X_{(k)}-a) = 0$ since a is the average. The other sums being nonnegative follows from the fact that a must be at least as large as the smaller averages in the chain, i.e. $a \geq \frac{1}{j}\sum_{i=1}^{j} X_{(i)}$ where $1 \leq j \leq k$. In detail, we know that

$$\begin{aligned}
X_{(j+1)} + \ldots + X_{(n)} &= na - [X_{(1)} + \ldots + X_{(j)}], \quad 1 \leq j \leq n-1 \\
&\geq na - ka = (n-k)a, \quad \text{from the fact above} \\
(X_{(j+1)}-a) + \ldots + (X_{(n)}-a) &\geq 0
\end{aligned}$$

Therefore, each sum $\sum_{k=j}^{n} \big(X_{(k)} - a\big) \geq 0$, $j = 1, \ldots, n$. It is clear that the expression

$$(i_1 - 1) \sum_{k=1}^{n} (X_{(k)} - a) \; + \; (i_2 - i_1 - 1) \sum_{k=2}^{n} \big(X_{(k)} - a\big) + \ldots + (i_n - i_{n-1} - 1)(X_{(n)} - a) \quad (3.13)$$

is exactly zero only when each $X_{(j)} = a$.

This shows that the difference in the value of the objective function between the proposed and an arbitrary sorting is nonnegative. Equality only results when both 1) every term in the chain equals the average a, and 2) the supernodes in $Y_{(j)}$ in the expression above also attain the maximum average value a. As a result, a merged list with $X_{(1)}, \ldots, X_{(k)}$ at the top produces an objective function at least as large as a list without this property.

The specific case in which equality results is rare in practice because several conditions must be met, so it is almost always better to sort so that the entire supernode $X_{(1)}, \ldots, X_{(k)}$ appears before supernodes of lesser value.

This demonstrates that the sorting that maximizes the objective function is given by finding maximal supernodes and sorting these by decreasing value. To see this, note that for any arbitrary sorting of the nodes, it is possible to produce a series of permutations from the arbitrary sorting to the proposed optimal sorting by following the sequential arrangement into supernodes given by the proof. At each of these permutation steps toward the optimal sorting, the sorting by supernode will have the larger objective function value.

3.3.1.4 A faster variation with extension to DAGs

The original algorithm begins from the leaf nodes and works upward, searching for the best supernode at each step. This algorithm runs slowly because of the exhaustive search that must be conducted at each iteration, which often involves recomputing averages. A simple modification of this algorithm results in $O(n \log n)$ runtime, and in simulations yields the same ordering as the original algorithm, with differences only on supernodes that share the same value. It remains an open problem to demonstrate this equivalence theoretically. The modified algorithm is given in Algorithm 2. Rather than starting from the leaf nodes, supernodes are constructed outward from a starting point of the largest LPR in the graph. This version of the algorithm is implemented in later sections.

Result: A hierarchically consistent sorting of n LPR values
$Par(S_i)$ is the parent of supernode S_i, and $n(S_i)$ is the number of nodes in S_i. Ψ is a vector for holding sorted LPR values.
Initialize with one node per LPR value, and each node as its own supernode, $\Psi = []$ (empty vector).
while $|\Psi| < n$ **do**
  Find $i = \text{argmax}_i \ \frac{1}{n(S_i)} \sum_{j \in S_i} LPR_j$
  if $Par(S_i) \in \Psi$ **then**
    Append the nodes in S_i to Ψ.
  else
    Condense S_i and $Par(S_i)$ into a supernode.
  end
end

Algorithm 2: Modification of the original HierLPR algorithm for trees with faster runtime.

This faster sorting method is an extension of the Condensing Sort and Select Algorithm (CSSA) of Baraniuk and Jones [1994], which we discuss in the next section. This variation has also been applied previously to the HMC problem in Bi and Kwok [2011] using scores from ridge regression, but our contribution provides a theoretically motivated justification for sorting the LPR in particular. Although the algorithm given above is for tree-structured data, it can be modified easily to accommodate DAGs [Bi and Kwok, 2011]. An extension of the original HierLPR to DAG structures is still an open problem.

3.3.2 Relationship to Condensing Sort and Select Algorithm

The concept of evaluating average along a hierarchy to produce a sorting of the nodes that is hierarchically consistent actually has its origins in Baraniuk and Jones [1994] with the Condensing Sort and Select Algorithm. That algorithm was proposed as a solution to the more general linear programming problem

$$\max_{\Psi} \sum_{k \in \mathcal{T}} B(k)\Psi(k), \qquad B(k) \geq 0 \tag{3.14}$$

$$\text{subject to} \tag{3.15}$$

$$\Psi(k) \geq 0 \quad \forall k \tag{3.16}$$

$$\Psi(0) = d \tag{3.17}$$

$$\Psi(k) \text{ is } \mathcal{T}\text{-nonincreasing} \tag{3.18}$$

$$\sum_{k \in \mathcal{T}} \Psi(k) \leq \gamma, \quad \gamma > 0 \tag{3.19}$$

Their proposed solution was the same as Algorithm 2, except their focus was on finding the appropriate node weights Ψ for sorting $L \geq n$ nodes of a tree. They demonstrate in their theoretical results that the solution must always place weights 0 or 1 on supernodes, up to

end behavior. In the case when the next largest supernode would put the sorting at over L nodes, the algorithm splits what remains of the total weight to be allocated, γ, among the nodes in this last supernode. When taken to be a sorting of the full set of nodes ($L = n$), avoiding this end behavior issue, CSSA becomes equivalent to Algorithm 2.

3.3.3 An overview of other performance metrics

In contrast to simple classification where measures like AUC, precision-recall, and F-measure are widely used, standard metrics for evaluating the performance of HMC methods have not yet been established. Development of metrics remains an active research topic today, with papers like Cerri et al. [2015], Costa et al. [2007], and Kosmopoulos et al. [2015] outlining the pros and cons of different measures for hierarchical classification.

HierLPR was developed as a method for maximizing a measure involving the hit curve, which is closely related to precision. Our method is thus suited for problems where accuracy in the initial set of positive calls is desirable, rather than capturing all of the true positives in the dataset, i.e. recall. In our setting, the hit curve or precision are calculated by averaging across instances, i.e. micro-averaging. It has been common to adapt flat classification metrics like precision/recall to hierarchical problems by averaging in this way, though the literature varies in whether averaging is done by instance (micro-averaged) or by class (macro-averaging). For example, macro-averaged metrics are used in Valentini [2009, 2011].

The primary criticism of these micro- and macro-averaged metrics is that they do not take the hierarchy into account: for example, an error made closer to the root in the hierarchy may be considered more serious than one deeper down, since that represents failure to classify to a broader category. As another example, assignment to a label that shares many ancestors with the correct class may be considered a less serious mistake than assignment to one farther away on the graph, since labels that are close to each other tend to be more similar to each other. These two examples illustrate that there are many ways to assess the severity of an error; the "right" kind of error to look at is largely dependent on the user's end goal, which explains in large part why a standard evaluation metric has not yet been agreed upon [Costa et al., 2007].

Several research directions, including this dissertation, provide solutions for the optimization of error metrics tailored for specific needs. For example, Bi and Kwok [2015] provide an efficient greedy algorithm for minimizing an extension of Hamming loss to the hierarchical case. Ramírez-Corona et al. [2016] provide a local classification method based on path evaluation that is particularly well-suited for deep and populated hierarchies. They propose a new metric that avoids the bias that others have toward shallower, conservative calls, and perform a method comparison with this measure to demonstrate the unique strength of their proposed method.

A simple hierarchical extension of the micro- and macro-averaged precision and recall takes distance into account [Sun and Lim, 2001], assigning greater penalty to assignments farther from the correct class. Another common evaluation metric is the hierarchical loss or H-loss of Cesa-Bianchi et al. [2006], which penalizes at the first misclassification along the

hierarchy. Yet more commonly used are the hierarchical precision, recall, and F-measure proposed by Kiritchenko et al. [2005]: these measures count the number of correctly predicted classes, together with correct predicted ancestor classes. We refer the reader to Cerri et al. [2015] for an extensive analysis of these hierarchical evaluation measures and how discriminating they are on various global and local methods. While the authors provide a general recommendation for the hierarchical precision and recall measures of Kiritchenko et al. [2005], their full discussion is much more nuanced, explaining how different performance metrics favor different HMC methods.

3.4 Evaluating HierLPR performance via simulation

3.4.1 The state of the art: decision tree learners

Clare [2003] was the first to apply a decision tree-based method to a hierarchical multilabel classification problem. Their method is an extension of C4.5, replacing the splitting criterion of class entropy with the sum of entropies across all classes. They applied their method to the problem of predicting protein function along the MIPS FunCat graph. Blockeel et al. [2002] introduced ClusHMC, a decision tree learner for HMC with tree hierarchies based on the Clus algorithm a.k.a. predictive clustering trees (PCTs). Blockeel et al. [2006] later improved upon the method and Vens et al. [2008] extended it to apply to DAG-structured hierarchies. Presently, ClusHMC and its variants remain the prevailing global classification method for HMC [Cerri et al., 2015]. In a recent application, Dimitrovski et al. [2011] constructed ensembles of ClusHMC with bagging and random forests for an image classification problem. These variants of ClusHMC performed better than ClusHMC alone and several other HMC methods assessed.

ClusHMC is similar to other decision tree learners: it iteratively finds splits that minimize the group variances, subject to criteria that control the fineness of the splits. In the multilabel case, each instance is represented as a $K \times 1$ binary vector, with 1 indicating class membership and 0 otherwise. After creating the full decision tree, the mean vector at each leaf node gives the proportion of class members. These proportions are used like posterior probabilities to classify new instances. The splitting criteria suggested in Vens et al. [2008] checks that the number of instances in a partition and the reduction in variance after splitting are above preset minimums. For checking variance reduction, the usual F-statistic is used, comparing the average variance post-partition with the variance before splitting.

In Vens et al. [2008], weighted Euclidean distance is chosen as the metric so similarity between two nodes near the root level can be evaluated as more important than those farther down. The variance of a node is defined as usual. To be precise mathematically, the weighted Euclidean distance is given as $d(v_1, v_2) = \sqrt{\sum_{i=1}^{K} w_i (v_{1,i} - v_{2,i})^2}$ where $w_i \geq 0$, and $v_{j,i}$ refers to the ith component of the vector v_j. Then, for a set of vectors V, define the variance of this set as $Var(V) = \frac{\sum_{j=1}^{V} d(v_j, \bar{v})^2}{|V|}$, where $\bar{v}$ is the usual mean vector for V.

3.4.2 Comparing HierLPR to decision tree learners

We examined the performance of HierLPR against PCTs on a range of simple trees where the quality and hierarchical relationships of the nodes were varied. Intuitively, the quality of a node refers to the ability of the classifier to distinguish between the positive and negative classes. A bad quality node produces classifier scores that are not informative for distinguishing between the two classes, i.e. the score distributions for the two class are not well separated; the opposite is true for a good quality node.

A total of eleven simulation settings were tested. The first nine settings involve hierarchical structures, depicted in Figures 3.3 and 3.4. Settings 1 through 7 are comprised of simple three-node hierarchical structures with different mixes of good and bad quality nodes–some settings are re-runs of the same hierarchical structure under different levels of dependence, i.e. setting 2 and setting 4 share the same graph as setting 1 and 3, respectively. Settings 8 and 9 represent more complicated hierarchical structures on 25 nodes, the former with a mix of node qualities and the latter with only good quality nodes. Settings 10 and 11 were generated from a nonhierarchical setting consisting of three nodes. For setting 10, all three nodes were of good quality; for setting 11, there was one node each of good, medium, and bad quality.

For each simulation setting, 100 data sets were generated. Each simulation data set consisted of 50,000 training instances and 10,000 test instances. Data from good quality nodes were generated from a Beta(2, 6) negative class distribution and Beta(6, 2) positive class distribution. For bad quality nodes, the distributions were Beta(5.5, 6) and Beta(6, 5.5) for negative and positive classes, respectively. Settings 8 through 11 involve a third medium quality setting, which had the negative and positive class distributions Beta(4, 6) and Beta(6, 4). For each data set from a hierarchical setting, the conditional probabilities $P(Q_i = 1|Par(Q_i) = 1)$ were randomly generated from a uniform distribution, with the constraint that each data set had to have a minimum of 150 cases of the positive class in the training set, which amounts to minimum prevalence of 0.3% for any class. Settings 2 and 4 re-runs of the same hierarchical structure as 1 and 3 but with conditional probabilities fixed at 0.95 for each node, so that the effect of high dependence between nodes could be examined.

For PCTs, we used ClusHMC and followed Dimitrovski et al. [2011] by constructing bagged ensembles and used the original settings of Vens et al. [2008], weighting each node equally when assessing distance, i.e. $w_i = 1$ for all i. In addition to node weights, there are two parameters to establish split criteria: the minimum number of instances after splitting and a minimum percentage variance reduction. The minimum number of instances was set to 5 in keeping with Lee [2013], and the minimum variance reduction was tuned via 5-fold cross validation from the options 0.60, 0.70, 0.80, 0.90, and 0.95. Following the implementation of Lee [2013], a default of 100 PCTs were trained for each ClusHMC ensemble; each PCT was the result of running ClusHMC on a resampling with replacement of the training data.

The average hit curve areas are shown in Table 3.8 for all eleven simulation settings. We also plot the precision-recall curves, averaged over all 100 replications, for ClusHMC,

Figure 3.3: Three-node graphs tested under simulation. Dark gray indicates that the node had class distributions corresponding to a low quality classifier, whereas light grey indicates high quality. Graph A corresponds to settings 1 and 2; B, setting 3 and 4; C, setting 5; D, setting 6; and E, setting 7.

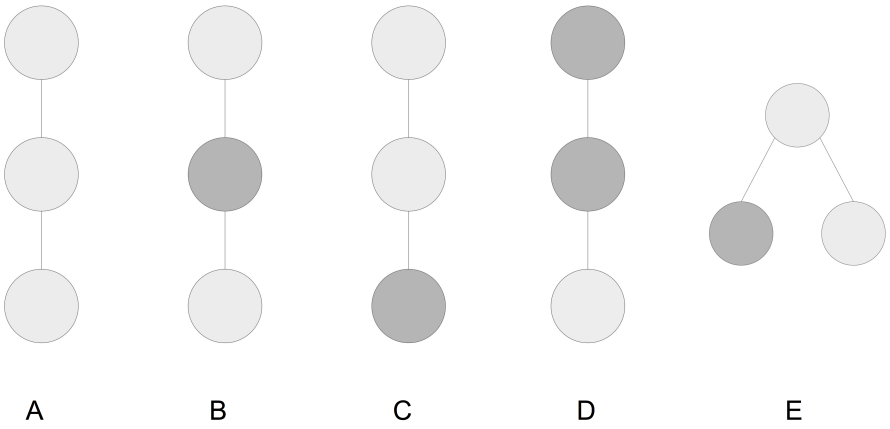
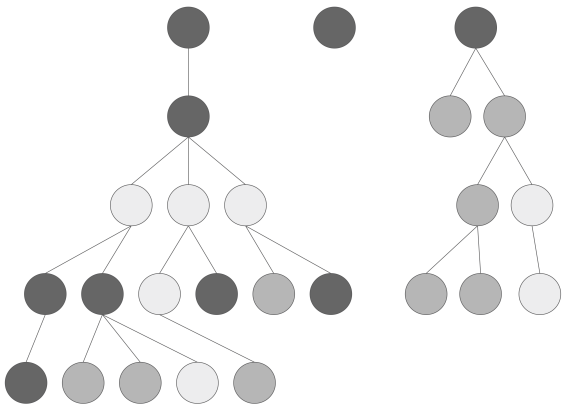


Figure 3.4: Graph structure with 25 nodes. The coloring indicates node quality for simulation setting 8: light, medium, and dark grey correspond to high, medium, and low quality, respectively. In simulation setting 9, all of the nodes have high quality class distributions.



HierLPR, and LPR on each setting. These are given in Figures 3.4.2 to 3.14.

For the hierarchical settings, HierLPR has better accuracy in its initial calls, as shown

in the inset graphs in Figures 3.4.2 to **??** that plot recall up to 0.01 along the x-axis. Most data sets had around 10,000 positive cases, so this corresponds to HierLPR having better performance on the first 100 or so calls. On these data sets, ClusHMC outperforms HierLPR eventually, although HierLPR tracks the performance of ClusHMC closely on settings with good quality nodes at the top of the hierarchy like settings 1, 2, 5, and 7. The more bad quality nodes occupy the top levels of the hierarchy, as tested in setting 6 and 8, the worse HierLPR's comparative performance–this occurs because the more low LPRs for positive cases there are, the stronger the evidence required of descendent labels to propagate a positive call upward. These results hold true under the more complex settings of 8 and 9. Increasing the level of dependence between the nodes also does not change the results.

In the non-hierarchical settings 10 and 11, HierLPR outperforms ClusHMC. This makes sense by design, since ClusHMC finds splits assuming dependence; when the nodes are independent as in the non-hierarchical case, any rules or associations it finds are due to random noise. On the other hand, HierLPR defaults in this setting to regular sorting, and is equivalent to the method of Jiang et al. [2014]. ClusHMC's performance suffers more noticeably when nodes of poor quality are included, as can be seen in the result for setting 11.

The areas under the hit curve (AUHC) shown in Table 3.8 confirm these results. On the hierarchical settings, HierLPR has similar performance to ClusHMC, but does slightly worse. On the nonhierarchical settings, HierLPR outperforms ClusHMC slightly. In all of the cases, the differences in performance are within an SD of each other, so none of these methods stand out as clear winners for any setting. For example, one difference from the analysis based on precision-recall is that HierLPR achieves the best AUHC on the setting with two bad nodes at the top, but we believe this is likely due to random noise since the difference in AUHC is small relative to the SD.

Altogether, our results suggest that HierLPR would perform best on mixed classification problems where graphs are either shallow with top level classifiers of good quality or standalone nodes. While global classification methods do better at leveraging information within a connected graph, they would perform poorly in situations where the hierarchy given contains a high proportion of standalone nodes.

Table 3.8: The average area under the hit curve over 100 replications for ClusHMC, LPR, and HierLPR under each simulation setting tested. The SD is given in parentheses. Each value has been divided by 1×10^6, so a value of 1 in the table actually corresponds to an area under the hit curve of 1×10^6.

	Area Under Hit Curve ($\times 10^6$)		
Setting	ClusHMC	LPR	HierLPR
1	188.366 (99.924)	187.604 (99.599)	188.044 (99.763)
2	188.392 (99.900)	187.625 (99.621)	188.023 (99.752)
3	88.537 (45.844)	86.231 (44.886)	87.852 (45.777)
4	185.612 (98.526)	179.760 (96.330)	184.712 (98.480)
5	186.837 (98.585)	182.248 (96.521)	185.234 (98.406)
6	165.624 (99.289)	164.439 (94.263)	170.938 (95.887)
7	100.709 (51.887)	98.290 (51.083)	99.446 (51.701)
8	11161.03 (3512.758)	10845.04 (3431.810)	10953.09 (3458.724)
9	11445.38 (3326.868)	11420.26 (3328.185)	11445.82 (3334.034)
10	323.374 (63.315)	324.770 (63.301)	324.770 (63.301)
11	303.221 (64.353)	308.272 (64.155)	308.272 (64.155)

Figure 3.5: Precision recall curves comparing ClusHMC, HierLPR, and LPR under simulation setting 1.

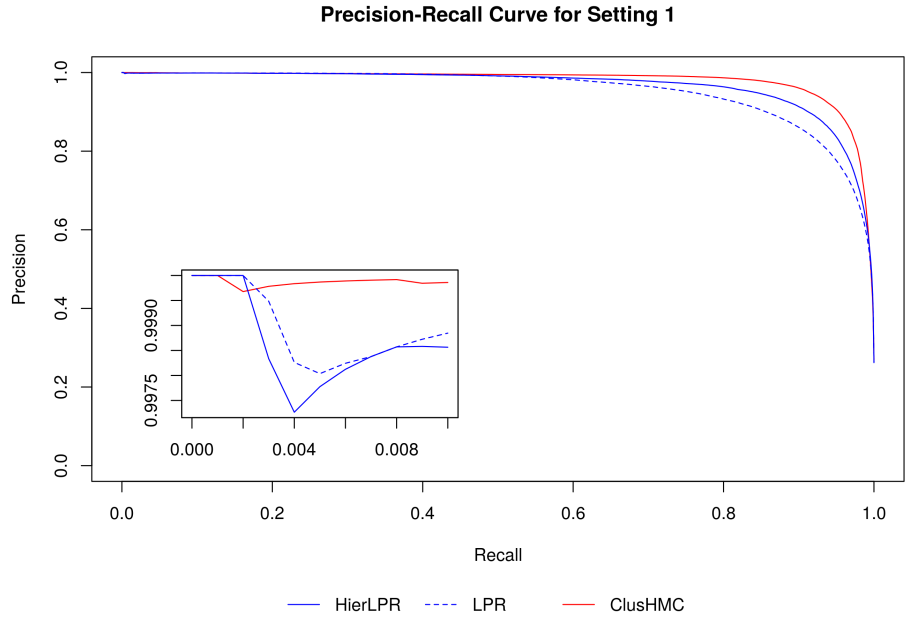


Figure 3.6: Precision recall curves comparing ClusHMC, HierLPR, and LPR under simulation setting 2.

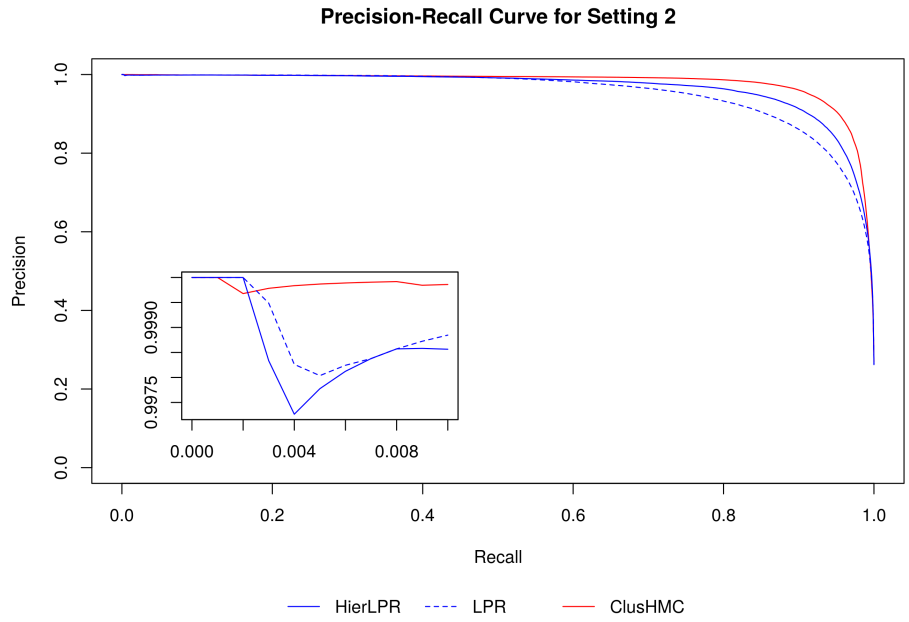


Figure 3.7: Precision recall curves comparing ClusHMC, HierLPR, and LPR under simulation setting 3.

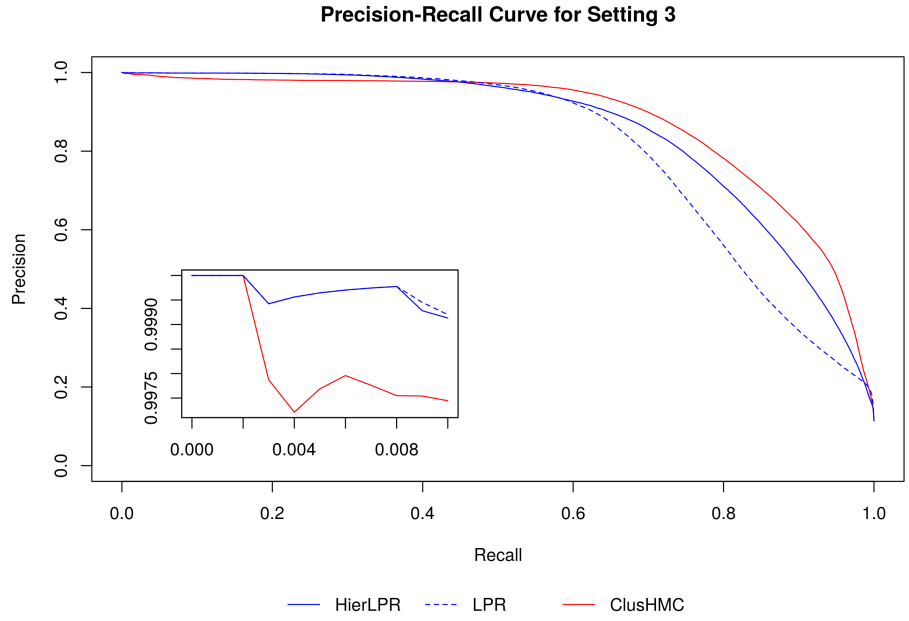


Figure 3.8: Precision recall curves comparing ClusHMC, HierLPR, and LPR under simulation setting 4.

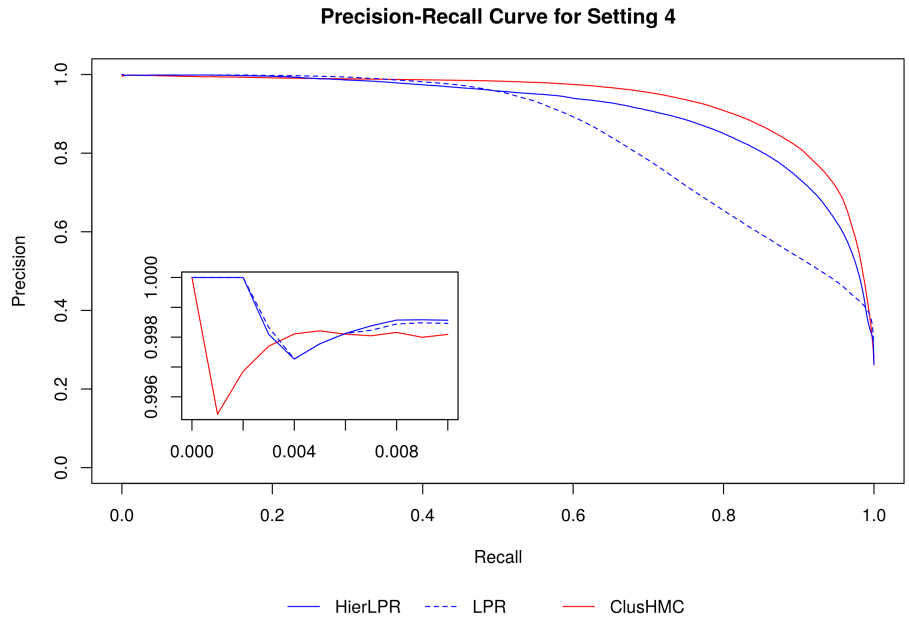


Figure 3.9: Precision recall curves comparing ClusHMC, HierLPR, and LPR under simulation setting 5.

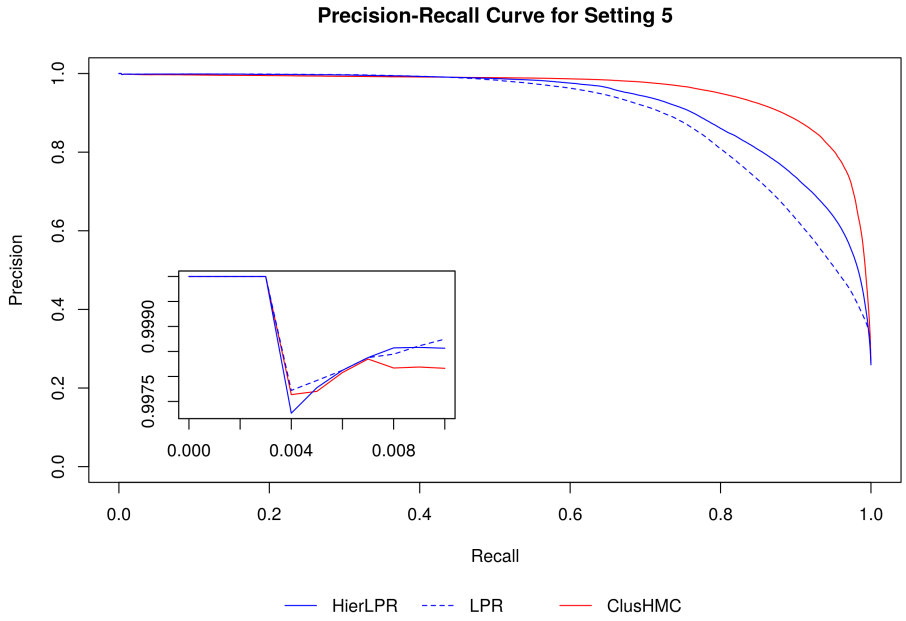


Figure 3.10: Precision recall curves comparing ClusHMC, HierLPR, and LPR under simulation setting 6.

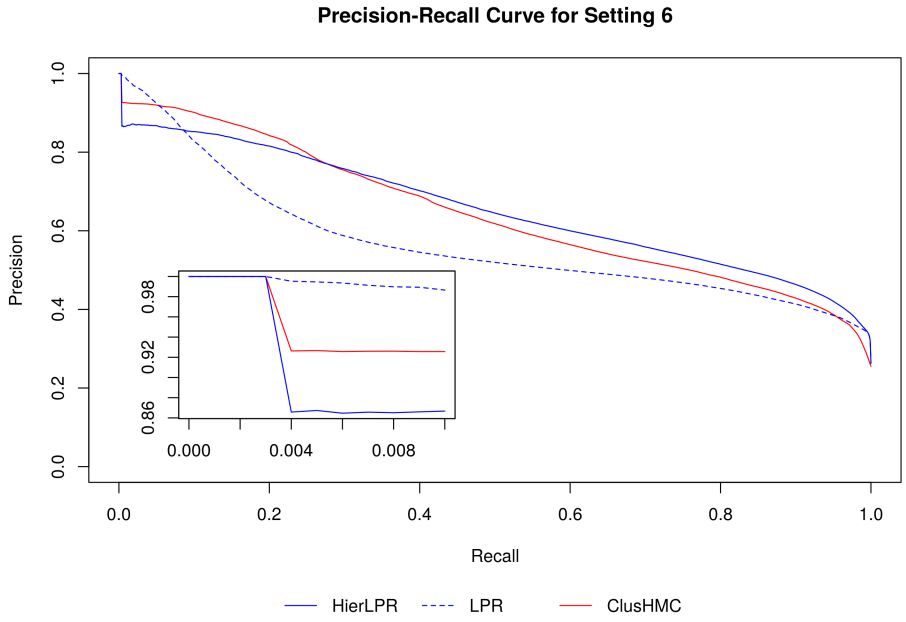


Figure 3.11: Precision recall curves comparing ClusHMC, HierLPR, and LPR under simulation setting 7.

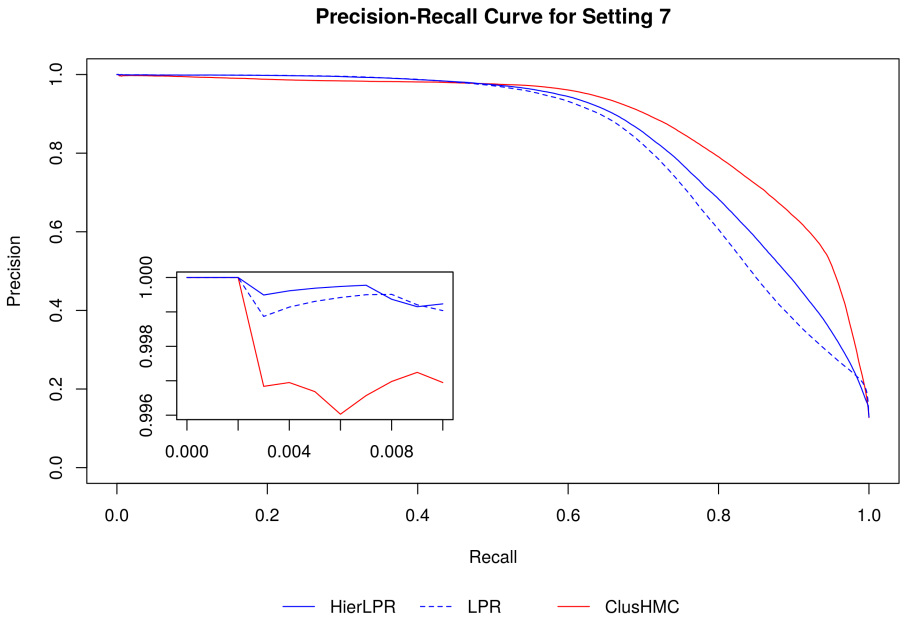


Figure 3.12: Precision recall curves comparing ClusHMC, HierLPR, and LPR under simulation setting 8.

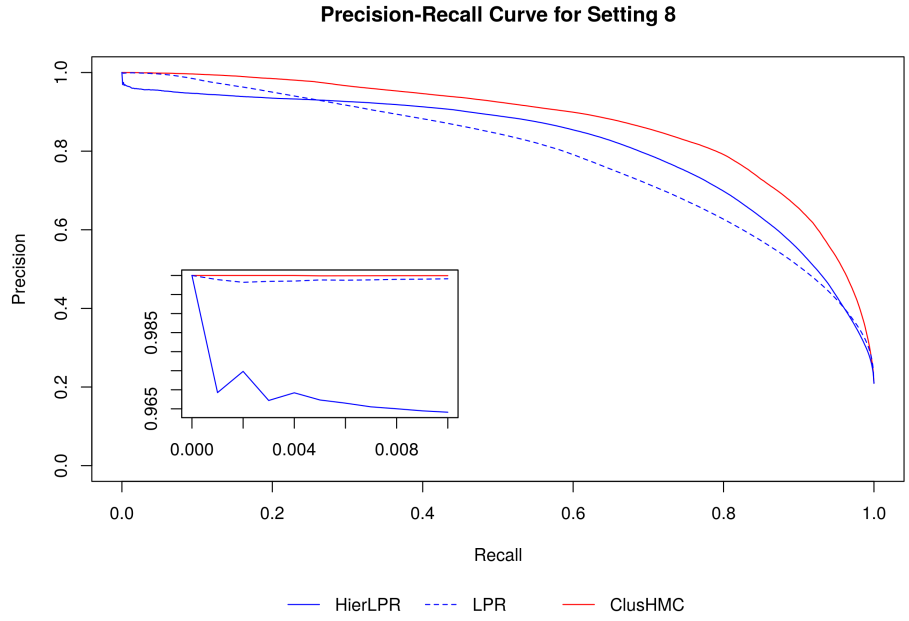


Figure 3.13: Precision recall curves comparing ClusHMC, HierLPR, and LPR under simulation setting 9.

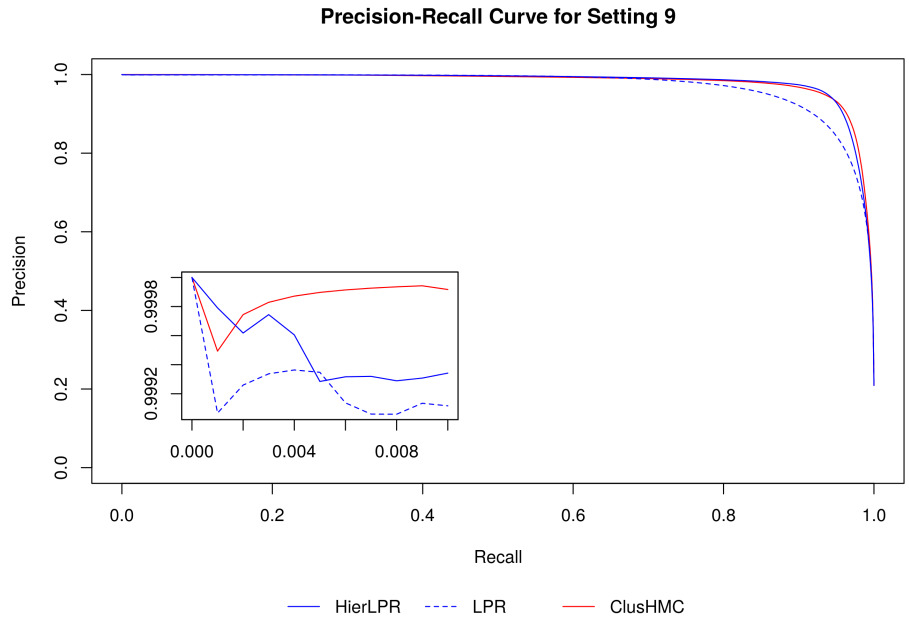


Figure 3.14: Precision recall curves comparing ClusHMC, HierLPR, and LPR under simulation setting 10.

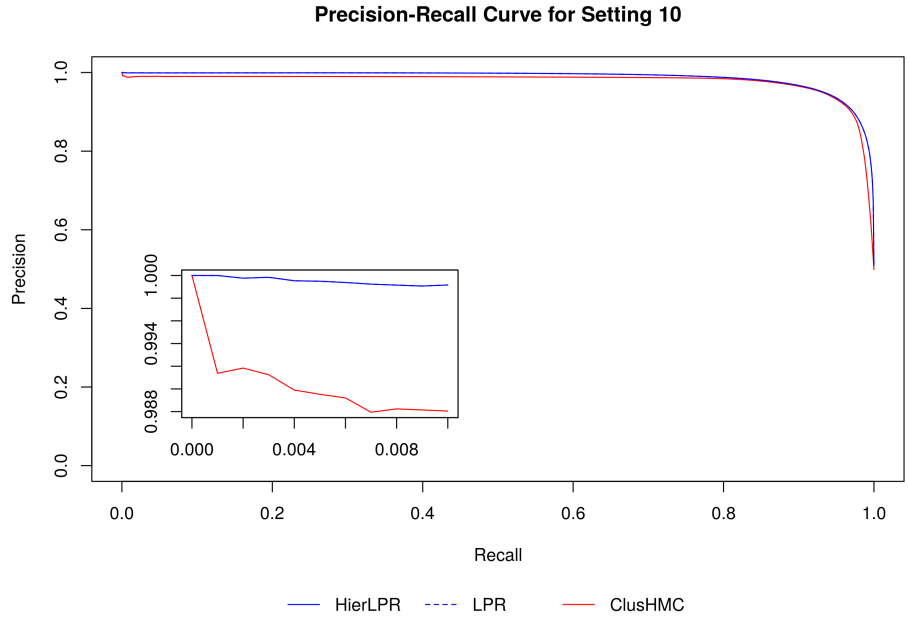
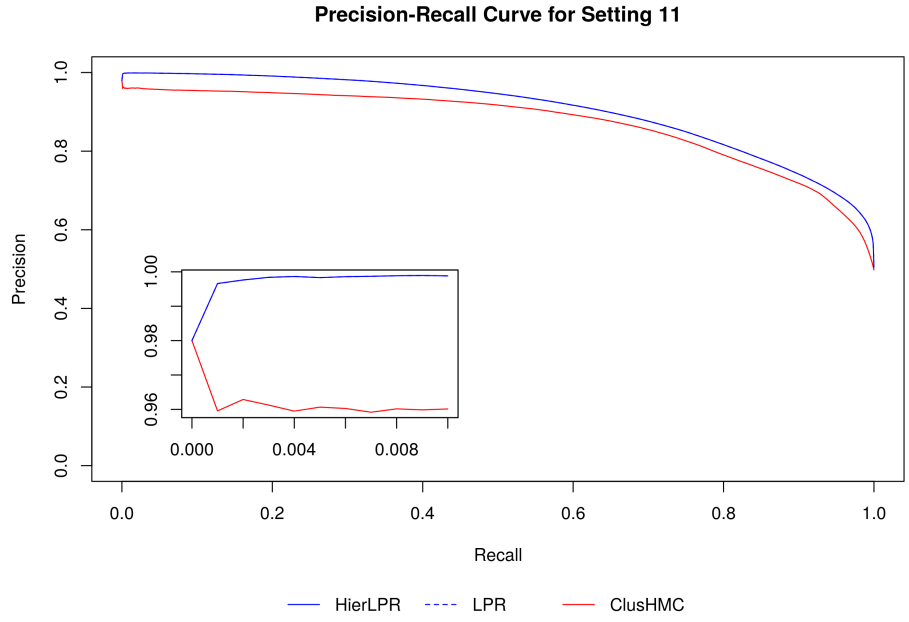


Figure 3.15: Precision recall curves comparing ClusHMC, HierLPR, and LPR under simulation setting 11.



3.5 An application to disease diagnosis with public microarray expression data

Huang et al. [2010] developed a classifier for predicting disease along the UMLS directed acyclic graph, trained on public microarray data sets from the National Center for Biotechnology Information (NCBI) Gene Expression Omnibus (GEO). As mentioned previous, the authors used a two-step local classification technique: they trained classifiers for each label in the UMLS graph, and corrected inconsistencies in the initial calls via a Bayesian adjustment. At its heart, the problem is hierarchical multilabel classification problem along a directed acyclic graph, where instances are not required to have leaf-level memberships. In the paragraphs below, we explain the problem setting in greater detail and compare the performance of our algorithm on their data set.

3.5.1 Data collection and classifier training

GEO was originally founded in 2000 to systematically catalog the growing volume of data produced in microarray gene expression studies. The large majority of data on GEO are these studies, and they continue to be the most common kind of study submitted today [Barrett et al., 2013]. The data in GEO represent research experiments submitted by scientists usually in compliance with grant or journal guidelines requiring that the data be made available in a public repository.

At the time of data retrieval, July 2008, GEO contained 421 human gene expression studies on the three microarray platforms that were selected for analysis (Affymetrix HG-U95A (GPL91), HG-U133A (GPL96), and HG-U133 Plus 2 (GPL570)). After filtering studies that contained data for both the disease and non-disease (normal) state, 100 studies yielding a total of 196 data sets remained. These were used for training the classifier in Huang et al. [2010].

Although each experiment had been done on one of the three microarray platforms, standardization or normalization of the data was necessary for removing study-specific technical effects. Because each data set contained replicated samples from a normal patient, the authors employed the following nonparametric procedure: for a data set with d disease and n normal replicates, the gene expression values were replaced with their ranks, and the logarithm of the ratio of ranks was computed for each of the $d \times n$ disease/normal pairs. The Pearson correlations between the log-rank-ratio vectors were used as raw data to train the classifier. If a new query were to be given, its expression values would have to be standardized against a known normal sample to produce a log-rank-ratio vector so that its Pearson correlations against the log-rank-ratio vectors in the training set could be used as input to the classifier. The Pearson correlations are called "similarity scores" as shorthand in the article.

Labels for each data set were obtained by mapping text from descriptions on GEO to concepts in the Unified Medical Language System (UMLS), an extensive vocabulary of con-

cepts in the biomedical or health field organized as a directed acyclic graph. The mapping resulted in a directed acyclic graph of 110 concepts matched to the 196 data sets with three degrees of similarity: 0 to indicate no match; 1, a GEO submission match; and 2, a data set match. These concepts, along with their GEO submission matches, are listed in Table S2 in the supplementary information for Huang et al. [2010]. As an example, the study with GEO identifier GDS2649 was matched to the unrelated concepts "female urogenital diseases" and "infections." Because of the hierarchical structure, it was also matched to a parent node of "female urogenital diseases", the broader label "female urogenital diseases and pregnancy complications."

Training a classifier for each label was a complex multi-step process, and is described in detail in the Supplementary Information of Huang et al. [2010]. For each node, the negative instances were taken to be the profiles in the 196 that did not have that label. The principal modeling step involved expressing the posterior probability of belonging to a label in terms of the log likelihood ratio and some probabilities that have straightforward empirical estimates. The log likelihood ratio was then modeled with a log-linear regression. A posterior probability estimate was then obtained for each of the 110×196 instances in the data by leave-one-out cross-validation, i.e. estimating the ith posterior probability based on the remaining 195 instances, and this was used as the first-stage classifier score. An initial label assignment for the first-stage was then obtained by finding the optimal score cutoffs for each classifier.

3.5.2 Characteristics of the disease diagnosis data and hierarchy

The full graph is given in Figures 3.16 and 3.17. The values inside the nodes indicate the number of positive cases, and the percentages underneath give the maximum percentage of cases shared with a parent. The colors indicate the quality of the nodes: the AUCs of each classifier were computed and grouped into three categories, ranging from (0.9, 1] (white), (0.7, 0.9] (light gray), and ≤ 0.7 (dark gray).

As the figure shows, the 110 nodes are grouped into 24 connected sets. In general, the graph is shallow rather than deep: the maximum node depth is 6, though the median is 2. However, only 10 nodes have more than one child. This occurs because 11 of the connected sets are standalone nodes, while six are simple two-node trees. The two largest sets consist of 28 and 30 nodes, respectively.

The graph nearly follows a tree structure. Most nodes have only one parent or are at the root level. Only 15 nodes have 2 parents, and 2 nodes have the maximum of 3 parents.

Most nodes do not have a high positive case prevalence. The largest number of instances belonging to a label is 62, or a 32.63% positive case prevalence. On average, the prevalence is 5.89%, with a minimum number of 1.53%, corresponding to 3 cases for a label.

Data redundancy occurs as an artifact of the label mining: because the most specific label for a data set was used for the hierarchy, sometimes no or few additional distinct cases were processed for ancestor nodes. Twenty six nodes or 23.64% of all nodes share the same data as their parents, so they have the same classifier, and therefore the same classifier scores

Figure 3.16: Structure of the disease diagnosis data set, part 1 of 2. The colors correspond to node quality: white indicates that a node's base classifier has AUC between $(0.9, 1]$; light grey, $(0.7, 0.9]$, dark grey, $(0, 0.7]$. The values inside the circles indicate the number of positive cases, while the value underneath gives the maximum percentage of cases shared with a parent node.

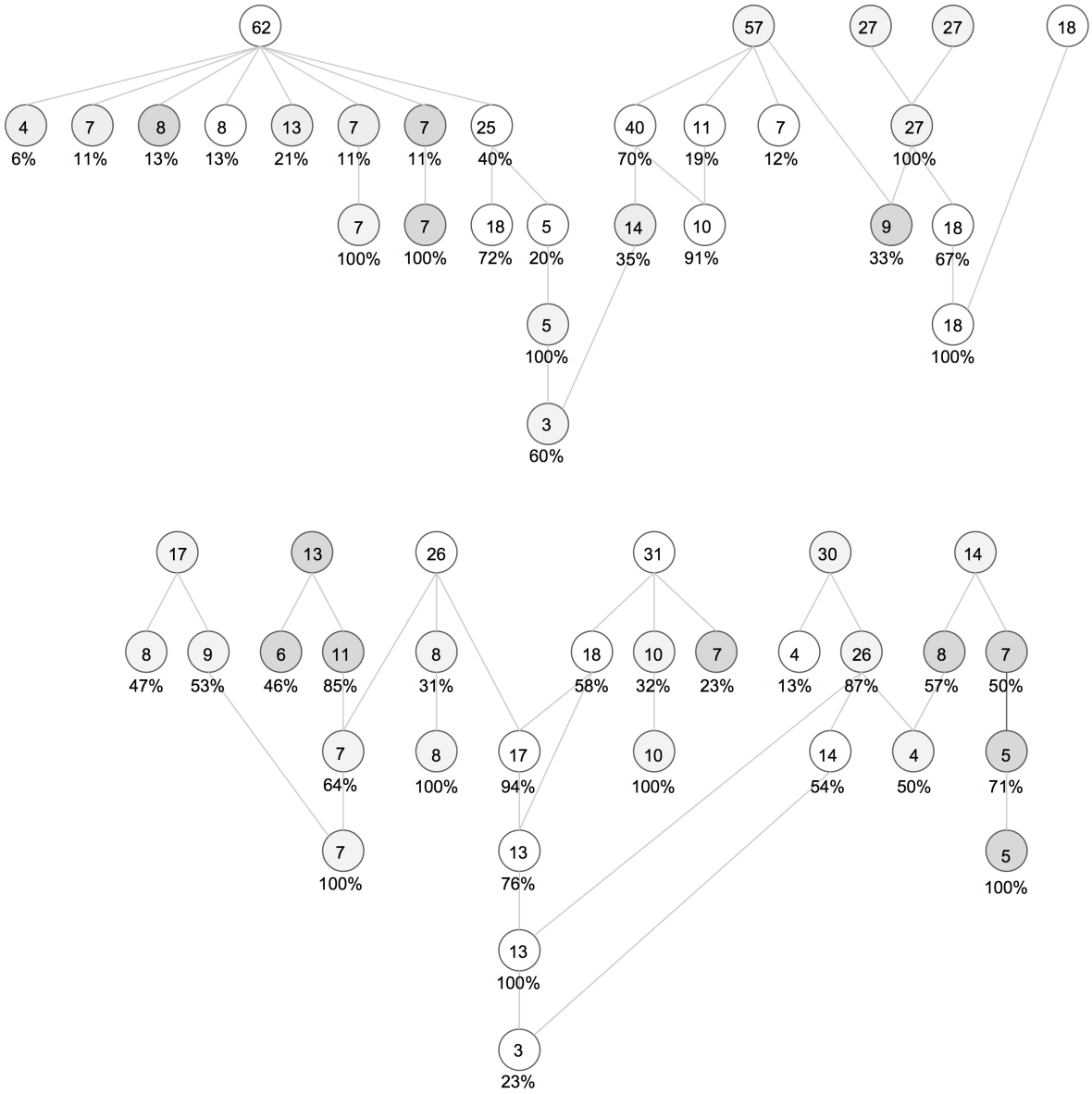
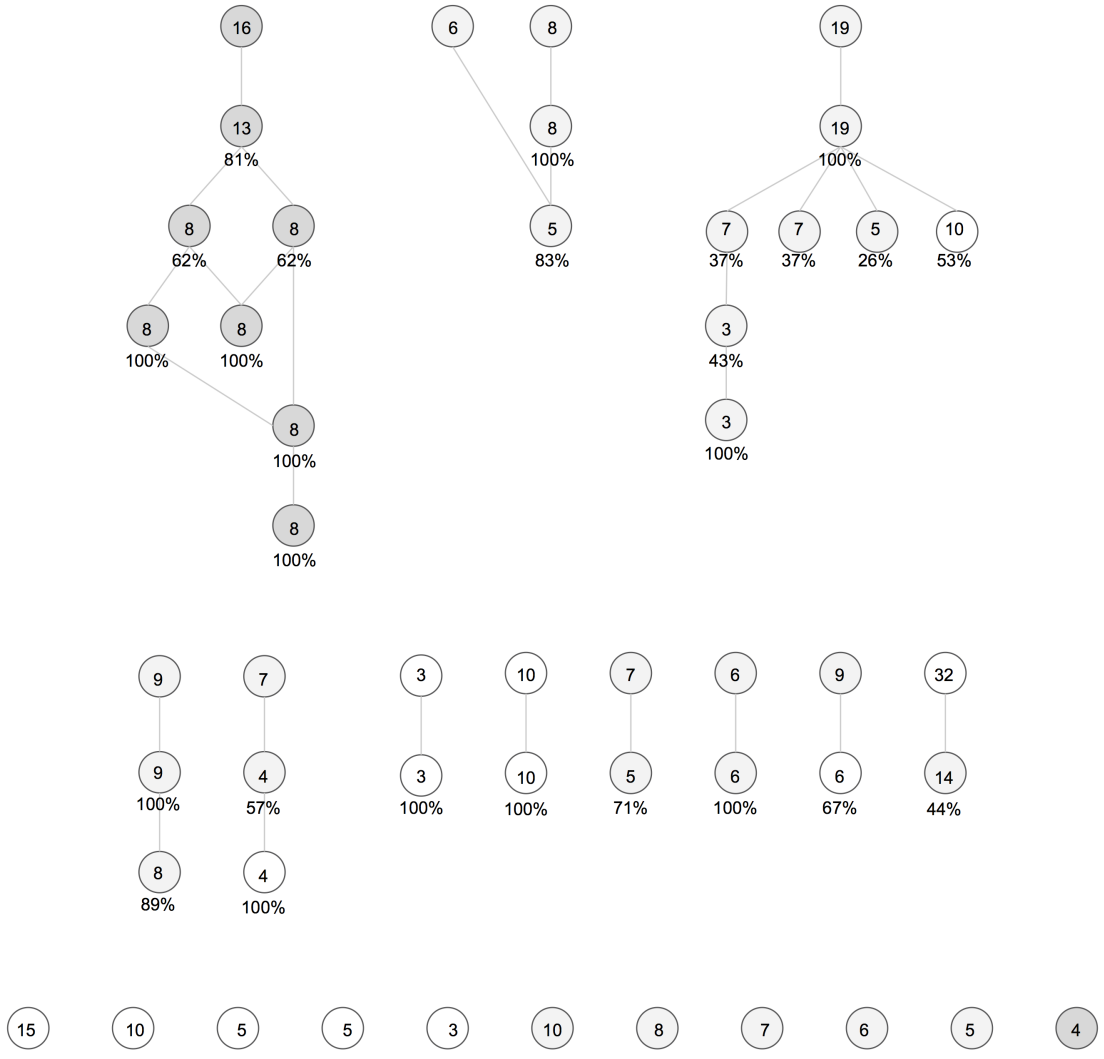


Figure 3.17: Structure of the disease diagnosis data set, part 2 of 2. The colors correspond to node quality: white indicates that a node's base classifier has AUC between $(0.9, 1]$; light grey, $(0.7, 0.9]$, dark grey, $(0, 0.7]$. The values inside the circles indicate the number of positive cases, while the value underneath gives the maximum percentage of cases shared with a parent node.



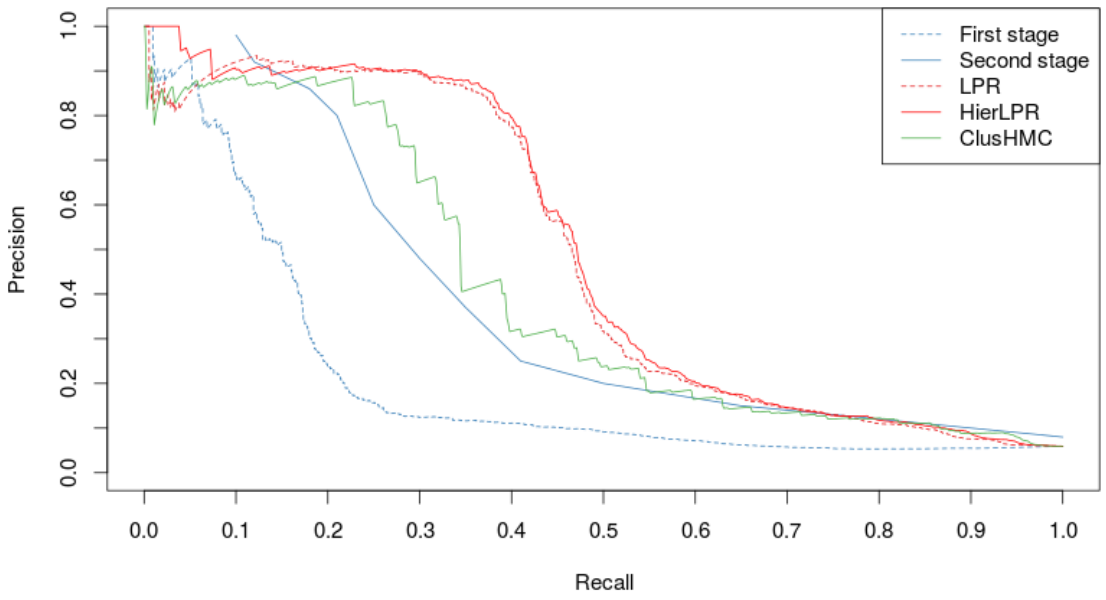
or LPRs as their parents. If we take the number of nodes that share more than half of their data with their parent, this statistic rises to 50%. A consequence of this redundancy is that

the graph is shallower than appears in the figure: for example, the first connected set in the top left of Figure 3.17 appears to have six levels, but actually only has three because the last three levels do not contain any new information.

3.5.3 Comparing HMC methods for disease diagnosis

We compared the performance of HierLPR against ClusHMC, the first- and second-stage classifier calls of Huang et al. [2010], and the LPR multilabel method of Jiang et al. [2014]. ClusHMC was run on the first-stage local classifier scores and the same parameter tuning options were used as in the simulations of Section 3.4.2. HierLPR was run with the same estimated LPRs as the method of Jiang et al. [2014], so the only difference between the two is the order in which the LPRs are sorted, and thus the order in which calls are made. The resulting precision-recall curve is shown in Figure 3.18.

Figure 3.18: Precision recall curve for several classifiers run on the public microarray disease data set of Huang et al. [2010].



HierLPR performs better than all of the other methods overall, although it performs significantly better in the initial portion of the precision-recall curve, as we expected from both our theoretical and simulation results. The precision-recall curve for HierLPR eventually tracks the one for the LPR method: this is because of the shallow graph structure, which makes the problem similar to the multilabel case. As discussed earlier in our analysis of the simulation results, HierLPR achieves the best performance here because over 20% of the

nodes are standalone, while the rest are in shallow hierarchical graphs. Further, the disease diagnosis data contains a mix of nodes of different qualities. As we saw in the simulation results earlier, the performance of ClusHMC on the nonhierarchical case is affected by the quality of the nodes. A closer inspection of the call order of the positive cases from the standalone labels is revealing: for HierLPR, these are some of the first instances to be called; in contrast, ClusHMC calls most of them after over half of the other positive instances have been called, illustrating the difficulty ClusHMC has in detecting these particular cases.

3.6 Discussion

3.6.1 Related methods

Rather than using LPRs, Bi and Kwok [2011] apply Algorithm 2 to the scores from ridge regression using transformed labeled. They apply their method to the same gene and protein function prediction data set as Clare [2003] and demonstrate improved precision-recall over ClusHMC, but we were unable to follow their method nor reproduce their results, and therefore we did not include it in our classifier comparisons. Their method involves performing kernel principal components analysis (PCA) on the labels as a pre-processing step. The ridge regression scores must be transformed back to the original label space, but the paper does not give details on how they do this since an exact back-projection is not possible with kernel PCA. The paper also does not describe how they imputed the missing values in the public data sets they used, nor do they make their data publicly available. Thus, even after applying common techniques for missing value imputation and approximations for projecting back to the original label space, we were unable to reproduce the AUPRC values they provided.

Although we could not compare against Bi and Kwok [2011], their method was the first to apply the sorting of Algorithm 2 to the HMC problem. Like this work, they were motivated by a need for a hierarchically consistent ordering of scores. However, our method provides theoretical justification for using the local precision rate over any other classifier score.

Valentini [2009] and Valentini [2011] provide an alternative algorithm for making calls beyond the common flat and sequential approaches given in Sun and Lim [2001]. Their algorithm is motivated by the true path rule property of the Gene Ontology and their specific application is gene function prediction on this taxonomy. Like our sorting method, their algorithm takes the weighted average of local classifier scores starting from the leaf level in the hierarchy, and is motivated in part by propagating evidence from child nodes up the hierarchy. They do not compare their method against other HMC methods and in other contexts.

3.6.2 Connection to statistical inference

The key distinction between inference and classification is the presence of training data, which allows users to estimate distributions that are assumed unknown in statistical inference. If we choose to ignore the available class distribution information, one can reframe a two-class classification problem as a hypothesis testing problem where the null corresponds to membership in the positive class. The classifier score could be used as a statistic, although this means that one would need to train the classifier on the available data and thereafter assume that they cannot estimate the class distributions. This approach clearly fails to take full advantage of the available data, but is meant to highlight the connection between these two problems.

As mentioned in Section 3.2.3, the local precision rate is closely related to another statistic used in Bayesian large-scale inference, the local false discovery rate. The local false discovery rate was motivated by the insight that in large-scale inference, enough data is available to estimate class distributions with some accuracy. As a result, it is possible to use pointwise statistics based on $f_0(s)/f_1(s)$, which may contain more information than their more popular tail-probability counterparts.

Most of the literature on this statistic has come from Bradley Efron, who laid the groundwork theory and provided interesting applications of the local false discovery rate in microarray gene expression experiments in Efron [2005, 2007, 2012]. Cai and Sun [2012] proved an optimality result similar to that of Jiang et al. [2014] for a multiple inference procedure for grouped hypotheses that uses local false discovery rates: their procedure minimizes the false nondiscovery rate subject to a constraint on the false discovery rate.

Research on hierarchical hypothesis testing is limited but growing. Yekutieli et al. [2006] first defined different ways to evaluate FDR when testing hypotheses along a tree, and gave a simple top-down algorithm for controlling these error types in Yekutieli [2008]. In that work, the hypotheses at each level of the tree were assumed independent. More recently, Benjamini and Bogomolov [2014]'s work on selective inference provided an algorithm for testing on hypotheses arranged in families along a two-level tree where the parent and child are permitted to be highly dependent, although in so doing they give up control on the global FDR.

Beyond the connection with the local false discovery rate, remains to be seen whether other concepts from classification with LPRs can also be applied to inference. Most of the literature is concentrated on theoretical results that show that certain testing procedures can effectively bound a measure of Type I error. One possibility is that sorting algorithms with origins in computer science, like the one presented in this work, could have meaningful applications as testing procedures.

3.6.3 Conclusions and open research directions

In this chapter, we present a method for performing hierarchical multilabel classification using local precision rates that is particularly well suited for graphs that contain a mix of

broad (rather than deep) hierarchies and standalone nodes. Our method was developed with the intent of maximizing the expected area under the hit curve, a measure closely related to precision that is used primarily in information retrieval, where the first few classifier calls are typically the most important for users. The proposed method is a local classification method, which means it is flexible: users have the option to build their own base classifiers; it is also fast: the algorithm has an $O(n \log n)$ runtime and it is not necessary to retrain the entire classifier when new nodes are added to the hierarchy.

A broad research direction is the extension of these ideas from classification to hypothesis testing along hierarchies. In addition, some smaller theoretical aspects of the proposed method have not been addressed. We presented two versions of our algorithm that gave the same results in simulation up to the ordering of supernodes that have the same value, but we have not yet demonstrated this equivalence theoretically. Likewise, the faster version of the algorithm has an easy extension for DAG structures, but an extension for DAGs was not developed for the original version from which the hit curve result was derived.

Bibliography

Ash A Alizadeh, Victoria Aranda, Alberto Bardelli, Cedric Blanpain, Christoph Bock, Christine Borowski, Carlos Caldas, Andrea Califano, Michael Doherty, Markus Elsner, et al. Toward understanding and exploiting tumor heterogeneity. *Nature medicine*, 21(8):846–853, 2015.

Roberto Teixeira Alves, MR Delgado, and Alex Alves Freitas. Knowledge discovery with artificial immune systems for hierarchical multi-label classification of protein functions. In *Fuzzy Systems (FUZZ), 2010 IEEE International Conference on*, pages 1–8. IEEE, 2010.

Camille Stephan-Otto Attolini, Yu-Kang Cheng, Rameen Beroukhim, Gad Getz, Omar Abdel-Wahab, Ross L Levine, Ingo K Mellinghoff, and Franziska Michor. A mathematical framework to determine the temporal sequence of somatic genetic events in cancer. *Proceedings of the National Academy of Sciences*, 107(41):17604–17609, 2010.

Geraldine A Auwera, Mauricio O Carneiro, Christopher Hartl, Ryan Poplin, Guillermo del Angel, Ami Levy-Moonshine, Tadeusz Jordan, Khalid Shakir, David Roazen, Joel Thibault, et al. From fastq data to high-confidence variant calls: the genome analysis toolkit best practices pipeline. *Current protocols in bioinformatics*, pages 11–10, 2013.

Richard G Baraniuk and Douglas L Jones. A signal-dependent time-frequency representation: Fast algorithm for optimal kernel design. *Signal Processing, IEEE Transactions on*, 42(1): 134–146, 1994.

Tanya Barrett, Stephen E Wilhite, Pierre Ledoux, Carlos Evangelista, Irene F Kim, Maxim Tomashevsky, Kimberly A Marshall, Katherine H Phillippy, Patti M Sherman, Michelle Holko, et al. Ncbi geo: archive for functional genomics data sets, update. *Nucleic acids research*, 41(D1):D991–D995, 2013.

Zafer Barutcuoglu, Robert E Schapire, and Olga G Troyanskaya. Hierarchical multi-label prediction of gene function. *Bioinformatics*, 22(7):830–836, 2006.

Ali Bashashati, Gavin Ha, Alicia Tone, Jiarui Ding, Leah M Prentice, Andrew Roth, Jamie Rosner, Karey Shumansky, Steve Kalloger, Janine Senz, et al. Distinct evolutionary trajectories of primary high-grade serous ovarian cancers revealed through spatial mutational profiling. *The Journal of pathology*, 231(1):21–34, 2013.

Sílvia Beà, Rafael Valdés-Mas, Alba Navarro, Itziar Salaverria, David Martín-Garcia, Pedro Jares, Eva Giné, Magda Pinyol, Cristina Royo, Ferran Nadeu, et al. Landscape of somatic mutations and clonal evolution in mantle cell lymphoma. *Proceedings of the National Academy of Sciences*, 110(45):18250–18255, 2013.

Niko Beerenwinkel, Jörg Rahnenführer, Martin Däumer, Daniel Hoffmann, Rolf Kaiser, Joachim Selbig, and Thomas Lengauer. Learning multiple evolutionary pathways from cross-sectional data. *Journal of computational biology*, 12(6):584–598, 2005.

Niko Beerenwinkel, Nicholas Eriksson, and Bernd Sturmfels. Evolution on distributive lattices. *Journal of theoretical biology*, 242(2):409–420, 2006.

Henrik Bengtsson, Ken Simpson, James Bullard, and Kasper Hansen. aroma. affymetrix: A generic framework in r for analyzing small to very large affymetrix data sets in bounded memory. Technical report, tech report, 2008.

Henrik Bengtsson, Pierre Neuvial, and Terence P Speed. Tumorboost: Normalization of allele-specific tumor copy numbers from a single pair of tumor-normal genotyping microarrays. *BMC bioinformatics*, 11(1):1, 2010.

Yoav Benjamini and Marina Bogomolov. Selective inference on multiple families of hypotheses. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76(1): 297–318, 2014.

Elmer V. Bernstam, Jorge R. Herskovic, Yindalon Aphinyanaphongs, Constantin F. Aliferis, Madurai G. Sriram, and William R. Hersh. Using citation data to improve retrieval from medline. *Journal of the American Medical Informatics Association*, 13(1):96–105, 2006. ISSN 1067-5027. doi: 10.1197/jamia.M1909.

Wei Bi and Jame T Kwok. Bayes-optimal hierarchical multilabel classification. *IEEE Transactions on Knowledge and Data Engineering*, 27(11):2907–2918, 2015.

Wei Bi and James T Kwok. Multi-label classification on tree-and dag-structured hierarchies. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 17–24, 2011.

Sven Bilke, Qing-Rong Chen, Frank Westerman, Manfred Schwab, Daniel Catchpoole, and Javed Khan. Inferring a tumor progression model for neuroblastoma from genomic data. *Journal of clinical oncology*, 23(29):7322–7331, 2005.

Ryan Bishop. Applications of fluorescence in situ hybridization (fish) in detecting genetic aberrations of medical significance. *Bioscience Horizons*, page hzq009, 2010.

Hendrik Blockeel, Maurice Bruynooghe, Saso Dzeroski, Jan Ramon, and Jan Struyf. Hierarchical multi-classification. In *Proceedings of the ACM SIGKDD 2002 workshop on multi-relational data mining (MRDM 2002)*, pages 21–35, 2002.

Hendrik Blockeel, Leander Schietgat, Jan Struyf, Sašo Džeroski, and Amanda Clare. *Decision trees for hierarchical multilabel classification: A case study in functional genomics*. Springer, 2006.

Niccolo Bolli, Hervé Avet-Loiseau, David C Wedge, Peter Van Loo, Ludmil B Alexandrov, Inigo Martincorena, Kevin J Dawson, Francesco Iorio, Serena Nik-Zainal, Graham R Bignell, et al. Heterogeneity of genomic evolution and mutational profiles in multiple myeloma. *Nature communications*, 5, 2014.

Garrett M Brodeur, Anastasios A Tsiatis, Dorothy L Williams, Fred W Luthardt, and Alexander A Green. Statistical analysis of cytogenetic abnormalities in human cancer cells. *Cancer genetics and cytogenetics*, 7(2):137–152, 1982.

T Tony Cai and Wenguang Sun. Simultaneous testing of grouped hypotheses: Finding needles in multiple haystacks. *Journal of the American Statistical Association*, 2012.

Peter J Campbell, Erin D Pleasance, Philip J Stephens, Ed Dicks, Richard Rance, Ian Goodhead, George A Follows, Anthony R Green, P Andy Futreal, and Michael R Stratton. Subclonal phylogenetic structures in cancer revealed by ultra-deep sequencing. *Proceedings of the National Academy of Sciences*, 105(35):13081–13086, 2008.

Benilton Carvalho, Henrik Bengtsson, Terence P Speed, and Rafael A Irizarry. Exploration, normalization, and genotype calls of high-density oligonucleotide snp array data. *Biostatistics*, 8(2):485–499, 2007.

Ricardo Cerri, Gisele L Pappa, André Carlos PLF Carvalho, and Alex A Freitas. An extensive evaluation of decision tree–based hierarchical multilabel classification methods and performance measures. *Computational Intelligence*, 31(1):1–46, 2015.

Nicolò Cesa-Bianchi, Claudio Gentile, and Luca Zaniboni. Incremental algorithms for hierarchical classification. *Journal of Machine Learning Research*, 7(Jan):31–54, 2006.

Lynda Chin, Steven E Artandi, Qiong Shen, Alice Tam, Shwu-Luan Lee, Geoffrey J Gottlieb, Carol W Greider, and Ronald A DePinho. p53 deficiency rescues the adverse effects of telomere loss and cooperates with telomere dysfunction to accelerate carcinogenesis. *Cell*, 97(4):527–538, 1999.

Kristian Cibulskis, Michael S Lawrence, Scott L Carter, Andrey Sivachenko, David Jaffe, Carrie Sougnez, Stacey Gabriel, Matthew Meyerson, Eric S Lander, and Gad Getz. Sensitive detection of somatic point mutations in impure and heterogeneous cancer samples. *Nature biotechnology*, 31(3):213–219, 2013.

Amanda Clare. *Machine learning and data mining for yeast functional genomics*. PhD thesis, The University of Wales, 2003.

Eric A Collisson, Raymond J Cho, and Joe W Gray. What are we learning from the cancer genome? *Nature Reviews Clinical Oncology*, 9(11):621–630, 2012.

Simona Constantinescu, Ewa Szczurek, Pejman Mohammadi, Jörg Rahnenführer, and Niko Beerenwinkel. Timex: a waiting time model for mutually exclusive cancer alterations. *Bioinformatics*, page btv400, 2015.

E Costa, A Lorena, ACPLF Carvalho, and A Freitas. A review of performance evaluation measures for hierarchical classifiers. In *Evaluation Methods for Machine Learning II: papers from the AAAI-2007 Workshop*, pages 1–6, 2007.

Alexander Davis and Nicholas E Navin. Computing tumor trees from single cells. *Genome biology*, 17(1):1, 2016.

Jesse Davis and Mark Goadrich. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on Machine learning*, pages 233–240. ACM, 2006.

Elza C de Bruin, Nicholas McGranahan, Richard Mitter, Max Salm, David C Wedge, Lucy Yates, Mariam Jamal-Hanjani, Seema Shafi, Nirupa Murugaesu, Andrew J Rowan, et al. Spatial and temporal diversity in genomic instability processes defines lung cancer evolution. *Science*, 346(6206):251–256, 2014.

Mark A DePristo, Eric Banks, Ryan Poplin, Kiran V Garimella, Jared R Maguire, Christopher Hartl, Anthony A Philippakis, Guillermo Del Angel, Manuel A Rivas, Matt Hanna, et al. A framework for variation discovery and genotyping using next-generation dna sequencing data. *Nature genetics*, 43(5):491–498, 2011.

Richard Desper, Feng Jiang, Olli-P Kallioniemi, Holger Moch, Christos H Papadimitriou, and Alejandro A Schäffer. Distance-based reconstruction of tree models for oncogenesis. *Journal of Computational Biology*, 7(6):789–803, 2000.

Ivica Dimitrovski, Dragi Kocev, Suzana Loskovska, and Sašo Džeroski. Hierarchical annotation of medical images. *Pattern Recognition*, 44(10):2436–2449, 2011.

Li Ding, Timothy J Ley, David E Larson, Christopher A Miller, Daniel C Koboldt, John S Welch, Julie K Ritchey, Margaret A Young, Tamara Lamprecht, Michael D McLellan, et al. Clonal evolution in relapsed acute myeloid leukaemia revealed by whole-genome sequencing. *Nature*, 481(7382):506–510, 2012.

Steffen Durinck, Christine Ho, Nicholas J Wang, Wilson Liao, Lakshmi R Jakkula, Eric A Collisson, Jennifer Pons, Sai-Wing Chan, Ernest T Lam, Catherine Chu, et al. Temporal dissection of tumorigenesis in primary cancers. *Cancer discovery*, 1(2):137–143, 2011.

Bradley Efron. *Local false discovery rates*. Division of Biostatistics, Stanford University, 2005.

Bradley Efron. Size, power and false discovery rates. *The Annals of Statistics*, pages 1351–1377, 2007.

Bradley Efron. *Large-scale inference: empirical Bayes methods for estimation, testing, and prediction*, volume 1. Cambridge University Press, 2012.

Jan B Egan, Chang-Xin Shi, Waibhav Tembe, Alexis Christoforides, Ahmet Kurdoglu, Shripad Sinari, Sumit Middha, Yan Asmann, Jessica Schmidt, Esteban Braggio, et al. Whole-genome sequencing of multiple myeloma from diagnosis to plasma cell leukemia reveals genomic initiating events, evolution, and clonal tides. *Blood*, 120(5):1060–1066, 2012.

Eric R Fearon, Bert Vogelstein, et al. A genetic model for colorectal tumorigenesis. *Cell*, 61 (5):759–767, 1990.

S Forbes, J Clements, E Dawson, S Bamford, T Webb, A Dogan, A Flanagan, J Teague, R Wooster, PA Futreal, et al. Cosmic 2005. *British journal of cancer*, 94(2):318–322, 2006.

Dan Frumkin, Adam Wasserstrom, Shalev Itzkovitz, Tomer Stern, Alon Harmelin, Raya Eilam, Gideon Rechavi, and Ehud Shapiro. Cell lineage analysis of a mouse tumor. *Cancer Research*, 68(14):5924–5931, 2008.

Marco Gerlinger, Andrew J Rowan, Stuart Horswell, James Larkin, David Endesfelder, Eva Gronroos, Pierre Martinez, Nicholas Matthews, Aengus Stewart, Patrick Tarpey, et al. Intratumor heterogeneity and branched evolution revealed by multiregion sequencing. *New England journal of medicine*, 366(10):883–892, 2012.

Moritz Gerstung, Michael Baudis, Holger Moch, and Niko Beerenwinkel. Quantifying cancer progression with conjunctive bayesian networks. *Bioinformatics*, 25(21):2809–2815, 2009.

Geoffrey S Ginsburg and Huntington F Willard. Genomic and personalized medicine: foundations and applications. *Translational research*, 154(6):277–287, 2009.

Chris D Greenman, Erin D Pleasance, Scott Newman, Fengtang Yang, Beiyuan Fu, Serena Nik-Zainal, David Jones, King Wai Lau, Nigel Carter, Paul AW Edwards, et al. Estimation of rearrangement phylogeny for cancer genomes. *Genome research*, 22(2):346–361, 2012.

Douglas Hanahan and Robert A Weinberg. Hallmarks of cancer: the next generation. *cell*, 144(5):646–674, 2011.

David J Hand. Measuring classifier performance: a coherent alternative to the area under the roc curve. *Machine learning*, 77(1):103–123, 2009.

Stefan Heinrichs and A Thomas Look. Identification of structural aberrations in cancer by snp array analysis. *Genome biology*, 8(7):1, 2007.

Jorge R Herskovic, M Sriram Iyengar, and Elmer V Bernstam. Using hit curves to compare search algorithm performance. *Journal of biomedical informatics*, 40(2):93–99, 2007.

Marcus Hjelm, Mattias Höglund, and Jens Lagergren. New probabilistic network models and algorithms for oncogenesis. *Journal of Computational Biology*, 13(4):853–865, 2006.

Ralph H Hruban, Michael Goggins, Jennifer Parsons, and Scott E Kern. Progression model for pancreatic cancer. *Clinical cancer research*, 6(8):2969–2972, 2000.

Haiyan Huang, Chun-Chi Liu, and Xianghong Jasmine Zhou. Bayesian approach to transforming public gene expression repositories into disease diagnosis databases. *Proceedings of the National Academy of Sciences*, 107(15):6823–6828, 2010.

Hanwen Huang, Fei Zou, and Fred A Wright. Bayesian analysis of frequency of allelic loss data. *Journal of the American Statistical Association*, 2012.

Ci-Ren Jiang, Chun-Chi Liu, Xianghong J Zhou, and Haiyan Huang. Optimal ranking in multi-label classification using local precision rates. *Statistica Sinica*, 24(4):1547–1570, 2014.

Alan S Jonason, Subrahmanyam Kunala, Gary J Price, Richard J Restifo, Henry M Spinelli, John A Persing, David J Leffell, Robert E Tarone, and Douglas E Brash. Frequent clones of p53-mutated keratinocytes in normal human skin. *Proceedings of the National Academy of Sciences*, 93(24):14025–14029, 1996.

Anne Kallioniemi, Olli P Kallioniemi, Damir Sudar, Denis Rutovitz, Joe W Gray, Fred Waldman, and Dan Pinkel. Comparative genomic hybridization for molecular cytogenetic analysis of solid tumors. *Science*, 258(5083):818–821, 1992.

Kenneth W Kinzler and Bert Vogelstein. Lessons from hereditary colorectal cancer. *Cell*, 87 (2):159–170, 1996.

Svetlana Kiritchenko, Stan Matwin, and Fazel Famili. Functional annotation of genes using hierarchical text categorization. 2005.

Svetlana Kiritchenko, Fazel Famili, S Matwin, and R Nock. Learning and evaluation in the presence of class hierarchies: Application to text categorization. 2006.

Daphne Koller and Mehran Sahami. Hierarchically classifying documents using very few words. 1997.

Aris Kosmopoulos, Ioannis Partalas, Eric Gaussier, Georgios Paliouras, and Ion Androutsopoulos. Evaluation measures for hierarchical classification: a unified view and novel approaches. *Data Mining and Knowledge Discovery*, 29(3):820–865, 2015.

Dan A Landau, Scott L Carter, Petar Stojanov, Aaron McKenna, Kristen Stevenson, Michael S Lawrence, Carrie Sougnez, Chip Stewart, Andrey Sivachenko, Lili Wang, et al. Evolution and impact of subclonal mutations in chronic lymphocytic leukemia. *Cell*, 152 (4):714–726, 2013.

Ben Langmead, Cole Trapnell, Mihai Pop, and Steven L Salzberg. Ultrafast and memory-efficient alignment of short dna sequences to the human genome. *Genome biology*, 10(3): 1, 2009.

Michael S Lawrence, Petar Stojanov, Paz Polak, Gregory V Kryukov, Kristian Cibulskis, Andrey Sivachenko, Scott L Carter, Chip Stewart, Craig H Mermel, Steven A Roberts, et al. Mutational heterogeneity in cancer and the search for new cancer-associated genes. *Nature*, 499(7457):214–218, 2013.

Wayne T. Lee. *Bayesian Analysis in Problems with High Dimensional Data and Complex Dependence Structure*. PhD thesis, University of California, Berkeley, 2013.

Heng Li. A statistical framework for snp calling, mutation discovery, association mapping and population genetical parameter estimation from sequencing data. *Bioinformatics*, 27 (21):2987–2993, 2011.

Heng Li and Richard Durbin. Fast and accurate short read alignment with burrows–wheeler transform. *Bioinformatics*, 25(14):1754–1760, 2009.

Heng Li, Jue Ruan, and Richard Durbin. Mapping short dna sequencing reads and calling variants using mapping quality scores. *Genome research*, 18(11):1851–1858, 2008a.

Ruiqiang Li, Yingrui Li, Karsten Kristiansen, and Jun Wang. Soap: short oligonucleotide alignment program. *Bioinformatics*, 24(5):713–714, 2008b.

Jun Liu, Nirmalya Bandyopadhyay, Sanjay Ranka, Michael Baudis, and Tamer Kahveci. Inferring progression models for cgh data. *Bioinformatics*, 25(17):2208–2215, 2009.

Vishal Madan, John T Lear, and Rolf-Markus Szeimies. Non-melanoma skin cancer. *The Lancet*, 375(9715):673–685, 2010.

Xueying Mao, Bryan D Young, and Yong-Jie Lu. The application of single nucleotide polymorphism microarrays in cancer research. *Current genomics*, 8(4):219–228, 2007.

Andriy Marusyk, Vanessa Almendro, and Kornelia Polyak. Intra-tumour heterogeneity: a looking glass for cancer? *Nature Reviews Cancer*, 12(5):323–334, 2012.

Andrew Mayne and Russell Perry. Hierarchically classifying documents with multiple labels. In *Computational Intelligence and Data Mining, 2009. CIDM'09. IEEE Symposium on*, pages 133–139. IEEE, 2009.

Nicholas McGranahan and Charles Swanton. Biological and therapeutic impact of intratumor heterogeneity in cancer evolution. *Cancer cell*, 27(1):15–26, 2015.

Aaron McKenna, Matthew Hanna, Eric Banks, Andrey Sivachenko, Kristian Cibulskis, Andrew Kernytsky, Kiran Garimella, David Altshuler, Stacey Gabriel, Mark Daly, et al. The genome analysis toolkit: a mapreduce framework for analyzing next-generation dna sequencing data. *Genome research*, 20(9):1297–1303, 2010.

Roger McLendon, Allan Friedman, Darrell Bigner, Erwin G Van Meir, Daniel J Brat, Gena M Mastrogianakis, Jeffrey J Olson, Tom Mikkelsen, Norman Lehman, Ken Aldape, et al. Comprehensive genomic characterization defines human glioblastoma genes and core pathways. *Nature*, 455(7216):1061–1068, 2008.

Michael L Metzker. Sequencing technologiesthe next generation. *Nature reviews genetics*, 11(1):31–46, 2010.

Jo Milner and EA Medcalf. Cotranslation of activated mutant p53 with wild type drives the wild-type p53 protein into the mutant conformation. *Cell*, 65(5):765–774, 1991.

Kensuke Nakamura, Taku Oshima, Takuya Morimoto, Shun Ikeda, Hirofumi Yoshikawa, Yuh Shiwa, Shu Ishikawa, Margaret C Linak, Aki Hirai, Hiroki Takahashi, et al. Sequence-specific error profile of illumina sequencers. *Nucleic acids research*, page gkr344, 2011.

Nicholas E Navin and James Hicks. Tracing the tumor lineage. *Molecular oncology*, 4(3): 267–283, 2010.

Cancer Genome Atlas Research Network et al. Integrated genomic analyses of ovarian carcinoma. *Nature*, 474(7353):609–615, 2011.

Michael A Newton. Discovering combinations of genomic aberrations associated with cancer. *Journal of the American Statistical Association*, 2012.

Michael A Newton and Yoonjung Lee. Inferring the location and effect of tumor suppressor genes by instability-selection modeling of allelic-loss data. *Biometrics*, 56(4):1088–1097, 2000.

Michael A Newton, Shi-Qi Wu, and Catherine A Reznikoff. Assessing the significance of chromosome-loss data: Where are suppressor genes for bladder cancer? *Statistics in medicine*, 13(8):839–858, 1994.

Michael A Newton, Michael N Gould, Catherine A Reznikoff, and Jill D Haag. On the statistical analysis of allelic-loss data. *Statistics in Medicine*, 17(13):1425–1445, 1998.

Serena Nik-Zainal, Peter Van Loo, David C Wedge, Ludmil B Alexandrov, Christopher D Greenman, King Wai Lau, Keiran Raine, David Jones, John Marshall, Manasa Ramakrishna, et al. The life history of 21 breast cancers. *Cell*, 149(5):994–1007, 2012.

Takafumi Nishizaki, Sandy DeVries, Karen Chew, William H Goodson, Britt-Marie Ljung, Ann Thor, and Frederic M Waldman. Genetic alterations in primary breast cancers and their metastases: direct comparison using modified comparative genomic hybridization. *Genes Chromosomes and Cancer*, 19(4):267–272, 1997.

Kenneth P Olive, David A Tuveson, Zachary C Ruhe, Bob Yin, Nicholas A Willis, Roderick T Bronson, Denise Crowley, and Tyler Jacks. Mutant p53 gain of function in two mouse models of li-fraumeni syndrome. *Cell*, 119(6):847–860, 2004.

Adam B Olshen, ES Venkatraman, Robert Lucito, and Michael Wigler. Circular binary segmentation for the analysis of array-based dna copy number data. *Biostatistics*, 5(4): 557–572, 2004.

Ignazio Pillai, Giorgio Fumera, and Fabio Roli. Threshold optimisation for multi-label classifiers. *Pattern Recognition*, 46(7):2055–2065, 2013.

Daniel Pinkel and Donna G Albertson. Array comparative genomic hybridization and its applications in cancer. *Nature genetics*, 37:S11–S17, 2005.

Karin J Purdie, Sally R Lambert, Muy-Teck Teh, Tracy Chaplin, Gael Molloy, Manoj Raghavan, David P Kelsell, Irene M Leigh, Catherine A Harwood, Charlotte M Proby, et al. Allelic imbalances and microdeletions affecting the ptprd gene in cutaneous squamous cell carcinomas detected using single nucleotide polymorphism microarray analysis. *Genes, Chromosomes and Cancer*, 46(7):661–669, 2007.

Elizabeth Purdom, Christine Ho, Catherine S Grasso, Michael J Quist, Raymond J Cho, and Paul Spellman. Methods and challenges in timing chromosomal abnormalities within cancer samples. *Bioinformatics*, page btt546, 2013.

Yi Qiao, Aaron R Quinlan, Amir A Jazaeri, Roeland GW Verhaak, David A Wheeler, and Gabor T Marth. Subcloneseeker: a computational framework for reconstructing tumor clone structure for cancer variant interpretation and prioritization. *Genome biology*, 15 (8):1, 2014.

Jörg Rahnenführer, Niko Beerenwinkel, Wolfgang A Schulz, Christian Hartmann, Andreas Von Deimling, Bernd Wullich, and Thomas Lengauer. Estimating cancer survival and clinical outcome based on genetic tumor progression scores. *Bioinformatics*, 21(10):2438–2446, 2005.

Mallinali Ramírez-Corona, L Enrique Sucar, and Eduardo F Morales. Hierarchical multilabel classification based on path evaluation. *International Journal of Approximate Reasoning*, 68:179–193, 2016.

Richard Redon, Shumpei Ishikawa, Karen R Fitch, Lars Feuk, George H Perry, T Daniel Andrews, Heike Fiegler, Michael H Shapero, Andrew R Carson, Wenwei Chen, et al. Global variation in copy number in the human genome. *nature*, 444(7118):444–454, 2006.

Zhi-Ping Ren, Afshin Ahmadian, Fredrik Ponten, Monica Nistér, Cecilia Berg, Joakim Lundeberg, Mathias Uhlen, and Jan Ponten. Benign clonal keratinocyte patches with p53 mutations show no genetic link to synchronous squamous cell precancer or cancer in human skin. *The American journal of pathology*, 150(5):1791, 1997.

Charles W Ross, Peter D Ouillette, Chris M Saddler, Kerby A Shedden, and Sami N Malek. Comprehensive analysis of copy number and allele status identifies multiple chromosome defects underlying follicular lymphoma pathogenesis. *Clinical Cancer Research*, 13(16): 4777–4785, 2007.

Edith M Ross and Florian Markowetz. Onconem: inferring tumor evolution from single-cell sequencing data. *Genome biology*, 17(1):1, 2016.

Andrew Roth, Jaswinder Khattra, Damian Yap, Adrian Wan, Emma Laks, Justina Biele, Gavin Ha, Samuel Aparicio, Alexandre Bouchard-Côté, and Sohrab P Shah. Pyclone: statistical inference of clonal population structure in cancer. *Nature methods*, 11(4):396–398, 2014.

Juho Rousu, Craig Saunders, Sandor Szedmak, and John Shawe-Taylor. Kernel-based learning of hierarchical multilabel classification models. *Journal of Machine Learning Research*, 7(Jul):1601–1626, 2006.

Eizaburo Sasatomi, Sydney D Finkelstein, Jeffrey D Woods, Anke Bakker, Patricia A Swalsky, James D Luketich, Hiran C Fernando, and Samuel A Yousem. Comparison of accumulated allele loss between primary tumor and lymph node metastasis in stage ii non-small cell lung carcinoma implications for the timing of lymph node metastasis and prognostic value. *Cancer research*, 62(9):2681–2689, 2002.

Eric E Schadt, Steve Turner, and Andrew Kasarskis. A window into third-generation sequencing. *Human molecular genetics*, 19(R2):R227–R240, 2010.

Anna Schuh, Jennifer Becq, Sean Humphray, Adrian Alexa, Adam Burns, Ruth Clifford, Stephan M Feller, Russell Grocock, Shirley Henderson, Irina Khrebtukova, et al. Monitoring chronic lymphocytic leukemia progression by whole genome sequencing reveals heterogeneous clonal evolution patterns. *Blood*, 120(20):4191–4196, 2012.

Roland F Schwarz, Anne Trinh, Botond Sipos, James D Brenton, Nick Goldman, and Florian Markowetz. Phylogenetic quantification of intra-tumour heterogeneity. *PLoS Comput Biol*, 10(4):e1003535, 2014.

Roland F Schwarz, Charlotte KY Ng, Susanna L Cooke, Scott Newman, Jillian Temple, Anna M Piskorz, Davina Gale, Karen Sayal, Muhammed Murtaza, Peter J Baldwin, et al. Spatial and temporal heterogeneity in high-grade serous ovarian cancer: a phylogenetic analysis. *PLoS Med*, 12(2):e1001789, 2015.

Sohrab P Shah, Andrew Roth, Rodrigo Goya, Arusha Oloumi, Gavin Ha, Yongjun Zhao, Gulisa Turashvili, Jiarui Ding, Kane Tse, Gholamreza Haffari, et al. The clonal and mutational evolution spectrum of primary triple-negative breast cancers. *Nature*, 486 (7403):395–399, 2012.

Jay Shendure and Hanlee Ji. Next-generation dna sequencing. *Nature biotechnology*, 26(10): 1135–1145, 2008.

Jay Shendure, Robi D Mitra, Chris Varma, and George M Church. Advanced sequencing technologies: methods and goals. *Nature Reviews Genetics*, 5(5):335–344, 2004.

Stephen T Sherry, M-H Ward, M Kholodov, J Baker, Lon Phan, Elizabeth M Smigielski, and Karl Sirotkin. dbsnp: the ncbi database of genetic variation. *Nucleic acids research*, 29(1):308–311, 2001.

Marwan Shinawi and Sau Wai Cheung. The array cgh and its clinical applications. *Drug discovery today*, 13(17):760–770, 2008.

Kimberly D Siegmund, Paul Marjoram, Yen-Jung Woo, Simon Tavaré, and Darryl Shibata. Inferring clonal expansion and cancer stem cell dynamics from dna methylation patterns in colorectal cancers. *Proceedings of the National Academy of Sciences*, 106(12):4828–4833, 2009.

Carlos N Silla Jr and Alex A Freitas. A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery*, 22(1-2):31–72, 2011.

Richard Simon, Richard Desper, Christos H Papadimitriou, Amy Peng, David S Alberts, Raymond Taetle, Jeffrey M Trent, and Alejandro A Schäffer. Chromosome abnormalities in ovarian adenocarcinoma: Iii. using breakpoint data to infer and test mathematical models for oncogenesis. *Genes, Chromosomes and Cancer*, 28(1):106–120, 2000.

Kathleen Sprouffske, John W Pepper, and Carlo C Maley. Accurate reconstruction of the temporal order of mutations in neoplastic progression. *Cancer Prevention Research*, 4(7): 1135–1144, 2011.

Johan Staaf, Johan Vallon-Christersson, David Lindgren, Gunnar Juliusson, Richard Rosenquist, Mattias Höglund, Åke Borg, and Markus Ringnér. Normalization of illumina infinium whole-genome snp data improves copy number estimates and allelic intensity ratios. *BMC bioinformatics*, 9(1):1, 2008.

W. Su, Y. Yuan, and M. Zhu. Threshold-free evaluation of medical tests for classification and prediction: Average precision versus area under the roc curve. *ArXiv e-prints*, 2013.

Aixin Sun and Ee-Peng Lim. Hierarchical text classification and evaluation. In *Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on*, pages 521–528. IEEE, 2001.

Barry S Taylor, Jordi Barretina, Nicholas D Socci, Penelope DeCarolis, Marc Ladanyi, Matthew Meyerson, Samuel Singer, and Chris Sander. Functional copy-number alterations in cancer. *PloS one*, 3(9):e3179, 2008.

Giorgio Valentini. True path rule hierarchical ensembles. In *International Workshop on Multiple Classifier Systems*, pages 232–241. Springer, 2009.

Giorgio Valentini. True path rule hierarchical ensembles for genome-wide gene function prediction. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 8 (3):832–847, 2011.

Celine Vens, Jan Struyf, Leander Schietgat, Sašo Džeroski, and Hendrik Blockeel. Decision trees for hierarchical multi-label classification. *Machine Learning*, 73(2):185–214, 2008.

J Craig Venter, Mark D Adams, Eugene W Myers, Peter W Li, Richard J Mural, Granger G Sutton, Hamilton O Smith, Mark Yandell, Cheryl A Evans, Robert A Holt, et al. The sequence of the human genome. *science*, 291(5507):1304–1351, 2001.

Bert Vogelstein and Kenneth W Kinzler. Cancer genes and the pathways they control. *Nature medicine*, 10(8):789–799, 2004.

Bert Vogelstein, Eric R Fearon, Stanley R Hamilton, Scott E Kern, Ann C Preisinger, Mark Leppert, Alida MM Smits, and Johannes L Bos. Genetic alterations during colorectal-tumor development. *New England Journal of Medicine*, 319(9):525–532, 1988.

Kai Wang, Mingyao Li, Dexter Hadley, Rui Liu, Joseph Glessner, Struan FA Grant, Hakon Hakonarson, and Maja Bucan. Penncnv: an integrated hidden markov model designed for high-resolution copy number variation detection in whole-genome snp genotyping data. *Genome research*, 17(11):1665–1674, 2007.

Jamie MJ Weaver, Caryn S Ross-Innes, Nicholas Shannon, Andy G Lynch, Tim Forshew, Mariagnese Barbera, Muhammed Murtaza, Chin-Ann J Ong, Pierre Lao-Sirieix, Mark J Dunning, et al. Ordering of mutations in preinvasive disease stages of esophageal carcinogenesis. *Nature genetics*, 46(8):837–843, 2014.

Feihong Wu, Jun Zhang, and Vasant Honavar. Learning classifiers using hierarchically structured class taxonomies. In *Abstraction, Reformulation and Approximation*, pages 313–320. Springer, 2005.

Lucy R Yates and Peter J Campbell. Evolution of the cancer genome. *Nature Reviews Genetics*, 13(11):795–806, 2012.

Daniel Yekutieli. Hierarchical false discovery rate–controlling methodology. *Journal of the American Statistical Association*, 103(481):309–316, 2008.

Daniel Yekutieli, Anat Reiner-Benaim, Yoav Benjamini, Gregory I Elmer, Neri Kafkafi, Noah E Letwin, and Norman H Lee. Approaches to multiplicity issues in complex research in microarray analysis. *Statistica Neerlandica*, 60(4):414–437, 2006.

Ke Yuan, Thomas Sakoparnig, Florian Markowetz, and Niko Beerenwinkel. Bitphylogeny: a probabilistic framework for reconstructing intra-tumor phylogenies. *Genome biology*, 16 (1):1, 2015.

Junjun Zhang, Joachim Baran, Anthony Cros, Jonathan M Guberman, Syed Haider, Jack Hsu, Yong Liang, Elena Rivkin, Jianxin Wang, Brett Whitty, et al. International cancer genome consortium data portala one-stop shop for cancer genomics data. *Database*, 2011: bar026, 2011.

Nancy R Zhang. Dna copy number profiling in normal andátumor genomes. In *Frontiers in Computational and Systems Biology*, pages 259–281. Springer, 2010.

Annemarie Ziegler, Alan S Jonason, David J Leffellt, Jeffrey A Simon, Harsh W Sharma, Jonathan Kimmelman, Lee Remington, Tyler Jacks, and Douglas E Brash. Sunburn and p53 in the onset of skin cancer. *Nature*, 372(6508):773–776, 1994.